



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**MOBILNÍ APLIKACE PRO KONTROLU DIETY
PACIENTŮ NA DIALÝZE**

MOBILE APPLICATION TO CONTROL THE DIET OF PATIENTS ON DIALYSIS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ PAVLACKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ GOLDMANN

BRNO 2021

Zadání bakalářské práce



Student: **Pavlacký Ondřej**
Program: Informační technologie
Název: **Mobilní aplikace pro kontrolu diety pacientů na dialýze**
Mobile Application to Control the Diet of Patients on Dialysis
Kategorie: Uživatelská rozhraní

Zadání:

1. Seznamte se s požadavky na dietu, kterou musí dodržovat pacienti na dialýze. Popište na základě konzultace s odborníkem základní parametry, které lékaři při těchto dietách u pacientů sledují.
2. Prostudujte a seznamte se s dostupnými technologiemi pro vývoj mobilních aplikací. Sumarizujte informace o dostupných relačních databázových systémech pro operační systém Android.
3. Navrhněte mobilní aplikaci, která bude sloužit pro evidenci požitých potravin a vypitých tekutin. Aplikace bude na základě složení potravin evidovat zkonsumované množství vybraných minerálů v krvi (především pak draslíku a fosforu). Denní přehledy zasílejte na server.
4. K aplikaci navrhněte intuitivní uživatelské a řešení implementujte pro operační systém Android.
5. Výslednou aplikaci otestujte s databází potravin dodanou vedoucím.

Literatura:

- LEVIN, Adeera, et al. Guidelines for the management of chronic kidney disease. *Cmaj*, 2008, 179.11: 1154-1162.
- MEIER, Reto. *Professional Android 4 application development*. John Wiley & Sons, 2012.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Goldmann Tomáš, Ing.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Cílem této práce je vytvořit intuitivní mobilní aplikaci pro operační systém Android, která bude pomáhat pacientům na dialýze zobrazovat přehled přibližných hodnot sledovaných látek v těle a nahradit klasické zapisování výsledných hodnot na papír. Díky této mobilní aplikaci budou mít pacienti lepší přehled o zkonsumovaných potravinách a pomůže jim to s dodržováním dietních opatření. Při zadávání budou vybírat z databáze potravin a aplikace jim spočítá hodnoty ze zadaného množství. Tuto databázi mohou pacienti rozšiřovat vlastními potravinami. Aplikace také bude evidovat mezidialyzační období (interval, který je mezi jednotlivými dialýzami) a z jejich délky, váhy pacienta a množství diurézy počítat maximální hodnoty sledovaných prvků (voda, draslík a fosfor). Pacienti uvidí také hodnoty za již uplynulé období, které budou zobrazeny ve statistikách. Data budou perzistentně uložena v lokální databázi i na serveru. Aplikace je vyvíjena na platformě Xamarin.

Abstract

The goal of this thesis is to create an intuitive mobile application for Android operation system which will help dialysis patients by displaying approximate values of monitored substances in their body and replace classic paper records. Thanks to this mobile application patients are going to have better overview of consumed foodstuffs a help them stick to their diet. When inserting the foodstuff patients will select from the database of foodstuffs and the application will calculate values that the food contains. Patients can also expand the database with their own food. Application will also register periods between dialysis and from the length of the period, patients' weight and their diuresis calculate maximal values of monitored substances (water, potassium and phosphorus). Patients will also see values from passed periods, which will be shown in statistics. Data will be persistently saved in local and remote database. Application is developed on Xamarin platform.

Klíčová slova

dialýza, dieta, pitný režim, pacient, mobilní aplikace, Android, Azure, Firebase, REST API, Xamarin, databáze

Keywords

dialysis, diet, drinking habits, patient, mobile application, Android, Azure, Firebase, REST API, Xamarin, database

Citace

PAVLACKÝ, Ondřej. *Mobilní aplikace pro kontrolu diety pacientů na dialýze*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann

Mobilní aplikace pro kontrolu diety pacientů na dialýze

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, pod vedením pana Ing. Tomáše Goldmanna. Další informace mi poskytl externí konzultant pan Vladimír Pleva. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Ondřej Pavlacký
11. května 2021

Poděkování

Děkuji panu Ing. Tomáši Goldmannovi za věnovaný čas, konzultace a odborné rady k implementaci. Také bych chtěl poděkovat panu Vladimíru Plevovi za věnovaný čas a odborné rady a konzultace ohledně léčby a potřeb pacientů na dialýze. V neposlední řadě bych chtěl poděkovat všem respondentům, kteří mi pomohli s vyplňováním dotazníků a uživatelském testování.

Obsah

1	Úvod	2
2	Dialýza	3
2.1	Uvedení do lékařské terminologie	5
2.2	Jídelníček	6
2.3	Sledované chemické prvky a molekuly	7
2.4	Použité výpočty	8
2.5	Analýza potřeb uživatele	9
2.6	Průzkum aktuálních řešení	11
3	Návrh aplikace a databáze	16
3.1	Návrh datového modelu	16
3.2	Návrh serverové části	18
3.3	Návrh uživatelského rozhraní	19
4	Implementace	22
4.1	Použité technologie	22
4.2	Ukládání dat	27
4.3	Mobilní aplikace	29
4.4	Generování intervalů	32
4.5	Implementace výpočtů hodnot	33
4.6	Synchronizace mezi zařízeními	33
4.7	Architektonický vzor MVVM	33
4.8	XAML	34
4.9	Entity Framework	34
4.10	LINQ	35
4.11	Server	35
5	Testování	37
5.1	Uživatelské testování	37
5.2	Automatické testy	38
5.3	Nalezené chyby	39
5.4	Nové nápady na vylepšení aplikace	39
6	Závěr	41
	Literatura	42

Kapitola 1

Úvod

Chronické onemocnění ledvin (*chronic kidney disease*), celosvětově postihující 9–10 % dospělé populace [12]. Toto onemocnění je závažným zdravotnickým problémem z hlediska zvýšeného rizika kardiovaskulárních onemocnění a předčasného úmrtí. Jedná se o dlouhodobé onemocnění, které vyžaduje speciální léčbu a dietu zaměřenou na nepřekročení přijatých hodnot z potravy. Léčba hemodialýzou (proces, při němž jsou z krve filtrovány škodlivé látky, které by při správném fungování ledvin vyloučilo tělo močí) je velmi náročná, nejen pro zdravotnický personál (pacienti musí dojíždět většinou 3× týdně do nemocnice nebo dialyzačních center a to většinou sanitkami), ale také i finančně pro stát. Proto prevence anebo dodržování stanovené diety může velmi usnadnit léčbu i náklady spojené s léčbou.

Mezi nejčastější pacienty se řadí lidé s věkem nad 50 let [12, str. 724, Obrázek 4]. Většina z nich musí podstupovat hemodialýzu. Mnoho z těchto pacientů si nezaznamenává údaje o svém životním stylu a proto je pro lékařský personál těžší diagnostikovat příčiny zhoršujícího se zdravotního stavu pacienta. Pouze malému procentu pacientů se daří dodržovat dietu kvůli její komplikovanosti a náročnosti. Tyto potíže můžou mít velmi vážné následky a vést až ke smrti. Pokud by pacienti měli možnost využít aplikaci, velmi by jim to usnadnilo dodržování diety a zjednodušilo hledání a výpočet sledovaných hodnot z potravin.

Za poslední roky se počet pacientů se selháním ledvin a nutností dialýzy stále zvyšuje následkem špatného životního stylu jako je obezita, kouření, alkohol, diabetes [12]. V neposlední řadě také kvalita potravin, v nichž často najdeme škodlivé chemikálie. Pacienti přibývají už od 20 let. U těchto pacientů je jednodušší sledovat zdravotní stav dlouhodobě, protože využívání mobilních zařízení je jim velmi dobře známo a práce s aplikacemi je pro ně jednoduchá. Navíc se hodí také dovednost najít si cenné informace na internetu, což může být pro starší pacienty mnohdy problém.

Cílem je vytvořit intuitivní a pro uživatele jednoduchou aplikaci, která bude počítat denní příjem hodnot fosforu, draslíku a tekutin z potravy a následně uživatele přibližně informovat o jeho zdravotním stavu a limitech, které má nastavené mezi jednotlivými dialýzami. Aplikace bude data také dlouhodobě uchovávat v lokální i vzdálené databázi, a proto bude možné vést dlouhobý přehled, který bude také zobrazovat. Aplikace bude fungovat i bez připojení k internetu, aby uživatel mohl provádět výpočty a sledovat hodnoty i bez internetového připojení. Hlavní cíl je rozšířit počet pacientů, kteří si evidují svůj příjem potravy a přispět ke snížení nárůstu úmrtí, kterých v posledních letech rapidně přibývá. Proto by tato aplikace měla zjednodušit pacientům sledování jejich diety a s nimi spojených opatření.

Kapitola 2

Dialýza

Pojem dialýza je lékařský proces, při němž se pacientovi trpícímu selháním ledvin, odstraní z těla látky, které by se při správném chodu ledvin odstranily samovolně močí. Mezi hlavní sledované hodnoty se řadí draslík, fosfor, ale také i množství vody. Pokud se v těle zadrží tyto látky ve vysokých hodnotách, může to vést k velkým zdravotním komplikacím [22]. Maximální hodnoty sledovaných hodnot většinou určuje nefrolog. Dají se ovšem dohledat v literatuře vzorce, které jsou schopné správně určit maximální a optimální hodnoty prvků, které pacient přijímá.

Léčba se dělí na několik druhů: peritoneální dialýza, hemodialýza a transplantace ledvin. Méně uváděnou možností léčby selhání ledvin je konzervativní léčba, kdy lékař diagnostikuje onemocnění a snaží se léky a správnou stravou obnovit správnou funkci ledvin. Tato metoda může pacienta vyléčit nebo aspoň oddálit nutnost dialýzy. Ovšem tento způsob léčby je neprokázaný a některé studie jej vyvrací [4].

Peritoneální dialýza

Peritoneální dialýza je méně praktikovaný způsob léčby. Pacienti nemusí dojíždět tak často do nemocnice a proces čištění látek z krve probíhá z pohodlí domova, pomocí vaku a roztoku viz obrázek 2.1. Přes chirurgicky zavedený katetr do dutiny břišní je napouštěn dialyzační roztok, který filtruje z krve odpadní látky a také přebytečnou vodu. Objem roztoku může být 2–3 litry. Po zhruba 5 hodinách se tekutina s odfiltrovanými látkami vypustí. Provádí se většinou v noci ve spánku. Do těla pacienta je také vpuštěna látka, které pomáhá k stabilizování acidózy.

Podle celosvětového průzkumu z roku 2004 se tato metoda používá zhruba u 11 % pacientů. Na druhou stranu v zemích jako je Mexiko je tato metoda aplikována u 75 % všech pacientů [13]. V České republice se používá pouze u 8 %. Jedním z důvodů proč tomu tak je, je fakt, že pacienti, kteří už začnou z hemodialýzou u ní také zůstávají. Dalším faktorem může být fakt, že plno zdravotnického personálu není s tímto způsobem léčby příliš seznámena a proto jej nedoporučuje [5].

Výhodou tohoto přístupu léčby je skutečnost, že může probíhat nepřetržitě a tudíž se může přirovnat k funkci opravdové ledviny. Tato metoda se dlouhodobě jeví jako nejlepší, protože jsou s hemodialýzou stejně bezpečné, ale navíc je méně nákladná. Nevýhodou může být snazší přenos infekcí do těla pacienta.



Obrázek 2.1: Ukázka zavedeného katetru při peritoneální dialýze [24, str. 7, Obrázek 1]

Hemodialýza

Hemodialýza se vykonává zhruba třikrát týdně a trvá v rozmezí dvou až osmi hodin. V roce 2009 bylo v České republice léčeno hemodialýzou 5 763 pacientů [5]. Pacienti dojíždí do nemocnic nebo speciálních středisek kam jsou dopravováni hlavně sanitkami. Tam za pomoci speciálního filtru (dialyzátoru) jsou jim z krve odebírány přebytečné látky, které nejsou schopni vyloučit sami. Jedná se o postup, při němž se krev pacienta čistí mimotělně pomocí dialyzátoru neboli umělé ledviny. Jak je vidět na obrázku 2.2, krev je z těla pacienta odváděna do dialyzačního přístroje, kde je zbavena odpadních látek (močoviny, kreatininu, fosforu a nadbytečných tekutin). Poté, co krev projde dialýzou, dostává se zpět do cévního řečiště a vrací se pacientovi do oběhu [20].

O času a četnosti hemodialýzy rozhoduje ošetřující nefrolog. Ten na základě výsledků pacienta rozhodne, kolikrát týdně je nutné navštívit dialyzační centra a jak dlouho bude dialýza trvat. Při hemodialýze probíhá veškerá práce v dialyzačním centru, nebo nemocnici pod pohledem proškoleného personálu. Tudíž můžou mít pacienti jistotu, že je o ně dobře postaráno.



Obrázek 2.2: Ukázka průběhu hemodialýzy a dialyzátoru [28, Obrázek 1]

Transplantace ledvin

K transplantaci ledvin se přistupuje pouze v případě, že jsou pacientovy ledviny v kritickém stavu a hrozí vážné zdravotní komplikace. Takových pacientů strmě přibývá a jen ve Spojených státech amerických je na seznamu čekatelů přes 75 745 pacientů, ale v roce 2020 jich bylo odoperováno pouze kolem 21 000 [14]. Proto se lékaři snaží léky a pravidelnou dialýzou snížit nutnost transplantace. Pacienti se po transplantaci většinou musí vrátit k dialýze, ale jejich zdravotní stav se velmi zlepšil.

2.1 Uvedení do lékařské terminologie

Definice níže uvedených pojmů vychází z literatury [33][6][19].

Dialýza Lékařský proces, při kterém jsou z těla pacienta odstraňovány škodlivé látky hromaděné v těle kvůli nesprávné funkci ledvin a nemohou být samovolně vyloučeny. Existují dva typy dialýz: hemodialýza a peritoneální dialýza.

Hemodialýza Procedura, která se provádí v nemocnici nebo dialyzačním centru. Pacient je připojen ke speciálnímu zařízení obsahující filtr nazývaný dialyzátor neboli umělé ledviny. Krev prochází přes filtr, je čištěna od odpadních látek, které nemůže sám vyloučit (např. fosfor, močovina, kreatinin, voda). Tento proces se provádí obvykle 3× týdně a zabere kolem 3-5 hodin, podle zdravotního stavu pacienta.

Peritoneální dialýza Procedura, kdy se pacientovi provádí čištění z krve přes výstelku dutiny břišní. Tekutina je zavedena peritoneálním katetrem. Pacient si napouští a vypouští dialyzační roztok, který se v peritoneální dutině nechává kolem 6ti hodin. Použitý roztok se poté vypustí a místo něj se napustí nový.

Suchá váha Normální nebo ideální tělesná hmotnost pacienta při optimálním obsahu vody v těle (hydrataci). Průměrný obsah vody se může velmi lišit na základě věku, svaloviny nebo onemocnění. U mužů průměrně činí 35-45 litrů a u žen 25-33 litrů.

Dialyzátor Dialyzátor je speciální filtr používaný při hemodialýze pro odstraňování látek a přebytečné vody z krve. Bývá někdy také označován jako „umělá ledvina“. Skládá se z filtru, který je složen z vláken, kterým se říká membrána. Tuto membránu si můžeme představit jako propustnou vrstvu, kterou prochází krev a filtrují se částice, které by mohli pacientovi uškodit.

Peritonální výstelka Velmi tenká výstelka dutiny břišní, která ji vymezuje a zároveň přechází na orgány, které jsou zde umístěny.

Cholesterol Nejvýznamnější živočišný sterol, který je přijímán výhradně v živočišné potravě. Jedná se o nezbytnou komponentu buněčných membrán, zdroj pro syntézu žlučových kyselin a steroidních hormonů.

VLDL cholesterol Patří mezi škodlivé typy cholesterolu, který je často označován jako „zlý“ cholesterol. Zanáší se do krevního oběhu plakem (mazlavá látka, je spojení cholesterolu, tuku a vápníku), který časem tuhne a usazuje se v krevním řečišti a ucpává cévy. To může mít za následek cévní příhody.

2.2 Jídelníček

Dietní opatření

Z konzultací s externím konzultantem a z literatury [34][9] jsem čerpal znalosti pro dietní opatření pacientů.

Dietní opatření se liší podle stupně selhání ledvin a metody léčby. Adekvátní výživa je velmi důležitá a musí být nedílnou součástí léčby spolu s dialýzou a léčebnými přípravky. Dialyzovaný pacient by měl dodržovat doporučení lékaře nebo nutričního terapeuta. Ten vypočítá a navrhne optimální stravovací plán [30].

U onemocnění ledvin je zejména důležité dbát na omezený příjem některých látek (např. bílkovin) a některé potraviny obsahující velké množství sodíku (obsaženého zejména v kuchyňské soli), fosforu a draslíku. Hlavním zdrojem energie by měly být sacharidy a tuky [30]. Pacient by se měl vyvarovat příjmu nasycených tuků a potravin s vysokým obsahem cholesterolu. Hlavní složku potravy by pak měly tvořit potraviny s vyšším obsahem nenasycených vyšších mastných kyselin. Mezi potraviny bohaté na tyto složky patří palmový olej, olivový olej, rybí tuk a vlašské ořechy. Dalším doporučením je dodržovat nízký příjem rychlých jednoduchých cukrů, které by měly tvořit maximálně 10 % z celkového denního energetického příjmu. V ideálním případě by tyto cukry měly být nahrazeny polysacharidy (škroboviny, zelenina).

Velice důležitý je přísun fosfátů, které se vyskytují hlavně v mléčných výrobcích, rybách, hořké čokoládě a mandlích.

Při nadměrném příjmu bílkovin vzniká řada nežádoucích látek (například dusíkaté metabolity) v krvi. Ty pak mohou mít negativní vliv na průběh léčby a následně vyvolat další zdravotní komplikace. Potraviny obsahující velké množství cholesterolu způsobují zvýšení tvorby VLDL cholesterolu, který zvyšuje riziko vzniku kardiovaskulárních onemocnění (infarkt myokardu, ateroskleróza). Cholesterol je často v potravě doprovázen nasycenými mastnými kyselinami, které taktéž zvyšují hladinu VLDL cholesterolu a přispívají k inzulinové rezistenci, tedy k diabetes mellitus.

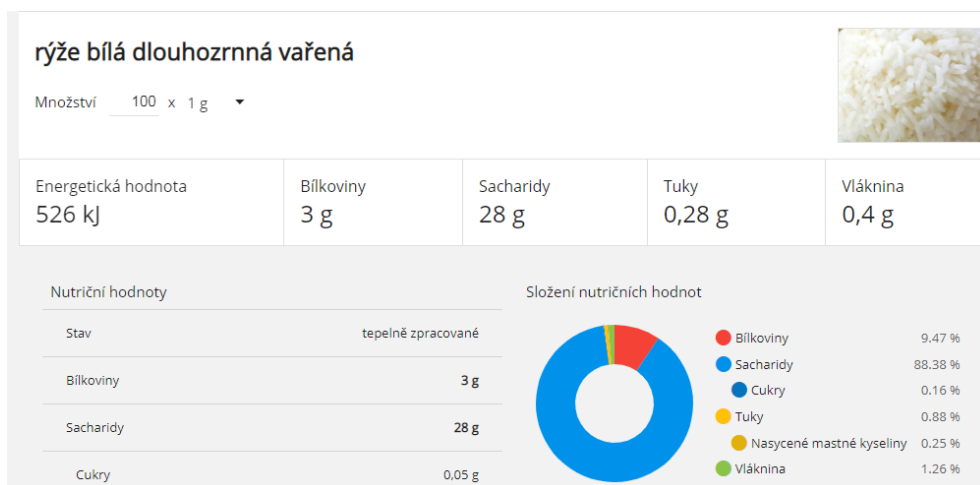
Pacient s chronickým onemocněním ledvin by se měl vyvarovat přesolování. Kuchyňská sůl totiž obsahuje poměrně velké množství sodíku. Doporučený denní příjem soli by neměl přesáhnout cca 5g pro dospělého člověka. V dnešní době se doporučené denní dávky dá dosáhnout i bez dosolování vlivem skrytých zdrojů soli, mezi které patří například sýry, pečivo, uzeniny. Hlavní problém zvýšeného příjmu sodíku spočívá v tom, že sodík váže molekuly vody. Pacient pak potřebuje přijímat větší množství tekutin, které musí ledviny vyfiltrovat.

Při dialýze je také velmi důležité dbát na příjem tekutin. U každého pacienta se přísun tekutin liší. Počítá se na základě diurézy a konstanty pro denní limit. Hodnotu většinou určuje nefrolog. Obecně platí, že přísun tekutin by měl být o něco vyšší než pacient vymočí [10]. U pacientů, kteří mají nulovou diurézu se jako denní limit udává pouze minimum. Při dialýze se pacientovi odebere přebytečná voda, ale může nastat situace, že není možné pacientovi odebrat veškerou vodu. Tudíž do dalšího období vstupuje s „dluhem“, který se mu od maxima odečte.

Kalorické tabulky

Soubor druhů potravin a ingrediencí, se kterými se nejčastěji setkáváme, informují o zastoupení bílkovin, sacharidů, tuků, chemických prvků a vitamínů pro předem určenou dávku. Každé kalorické tabulky jsou pro určité účely. Některé sledují kalorie a bílkoviny a

jsou určeny převážně pro sportovce. Jiné uvádí sacharidy, sůl, tuky a mnoho dalších pro obecné použití, jak je vidět na obrázku 2.3. Pacienti na dialýze sledují specifickou kombinaci hodnot, které se mnohdy v tabulkách nenachází a musí hledat ve více zdrojích. Proto nefrologové pacientům předávají základní seznam potravin se sledovanými hodnotami. Na základě své rešerše a tabulky dodané vedoucím, jsme vytvořili databázi potravin, která je v aplikaci použita. Principem kalorických tabulek je spočítat pomocí matematických operací množství látek v námi vyžadovaném množství. Příklad kalorických tabulek: <https://www.kaloricketabulky.cz/>



Obrázek 2.3: Ukázka z výživových hodnot rýže [21].

2.3 Sledované chemické prvky a molekuly

Draslík (K) Hlavní intracelulární prvek. Podporuje činnost trávicího traktu, udržování stálé tělesné teploty. V krvi je jeho koncentrace za normálních podmínek velmi nízká. U pacientů s chronickým onemocněním ledvin bývá jeho koncentrace zvýšená, a proto se také jeho přítomnost v krvi sleduje. Není tomu však pravidlem, u některých pacientů (například u pacientů podstupujících pravidelnou peritoneální dialýzu) může být hladina dokonce nižší. Výkyvy v koncentraci draslíku v krvi mají vždy negativní důsledek například: srdeční obtíže, bolesti hlavy a křeče. Hodnota draslíku v krvi má velmi úzké optimální rozmezí, v extrémních případech může dojít až k zástavě srdce [8].

Fosfor (P) Fosfor patří mezi tzv. biogenní prvky. Tvoří přibližně 1 % hmotnosti lidského těla. Společně s vápníkem přispívá k mineralizaci kostí a zubů, zodpovídá za jejich tvrdost. Jeho koncentrace v krvi je přísně regulována prostřednictvím činnosti ledvin. U pacientů s poruchou ledvin může nadměrná hladina fosforu v krvi zvýšit riziko vzniku osteoporózy (odvápňování kostí) anebo aterosklerózy (kornatění cév) [7].

Voda (H₂O) Voda tvoří asi 60 % hmotnosti lidského těla. Doporučená denní dávka je pro zdravého jedince kolem 2-2,5l. Při onemocnění ledvin se může pitný režim neustále měnit v závislosti na aktuálním stavu.

Močovina ($\text{CH}_4\text{N}_2\text{O}$) Odpadní látka v krvi, která vzniká rozkladem bílkovin. Je vylučována z lidského těla močí. Její koncentrace v krvi se bude zvyšovat právě při chronickém selhání ledvin. Je jedním z důležitých indikátorů při jejich selhání.

Kreatinin ($\text{C}_4\text{H}_7\text{N}_3\text{O}$) Odpadní produkt vznikající při svalové práci. Jeho zvýšení koncentrace může signalizovat selhání ledvin.

2.4 Použité výpočty

Pro výpočet maximálních a optimálních hodnot jsem vycházel ze vzorců, které jsem našel v literatuře. Pro všechny sledované míry se vzorce liší a je nutné, aby použité vzorce reflektovaly realitu a měly za následek co nejvíce se přiblížit reálným hodnotám z přijaté potravy. U některých vzorců se konstanty mění podle různých faktorů, jako je například suchá váha nebo diuréza.

V literatuře se pro maximální denní hodnoty používá tabulka 2.1, která počítá maximální hodnoty na jeden den. Pro větší přesnost měření maximálních hodnot jsou dny, ve kterých probíhá dialýza rozděleny i na hodiny, abychom předešli nepřesnostem.

Prvek	Hodnota
Energie	150 kJ/kg/den
Bílkoviny	1,2 g/den
Draslík	2000 mg/den
Fosfor	do 1000 mg/den
Sodík	5 g/den
Tekutiny	500ml + množství diurézy

Tabulka 2.1: Tabulka maximálních denních hodnot pro pacienta na dialýze [27]

Tekutiny

Příjem vody u dialyzovaných pacientů se řídí několika faktory. Prvním z nich je množství diurézy, kterou si pacient měří zhruba jednou za měsíc. Obecně platí, že maximální denní příjem tekutin je konstanta 500ml, která je uvedena v literatuře [27] + množství diurézy viz tabulka 2.1. Ovšem u pacientů, u kterých je diuréza nulová se jedná pouze o konstantu 500-700ml na jeden den. Dále by maximální příjem tekutin v mezidialyzačním období neměl překročit 3 % suché hmotnosti pacienta (v průměru činí zhruba 2kg) [16].

Fosfor

Maximální denní dávka fosforu se určuje konstantou, která dosahuje až 1000mg/den. Pacienti mohou pro snížení hladiny fosforu použít léky, takzvané vazače fosfátů. Jsou jim vydávány na předpis jejich ošetřujícího nefrologa. Existuje mnoho druhů a každý je schopen navázat různé množství fosforu. Více než 70 % pacientů tyto vazače nevyužívá navzdory jejich pozitivním účinkům [11]. Bohužel u těchto pacientů je složení těchto vazačů nějakým způsobem závadné a mohly by jim více uškodit, než pomoci.

Draslík

Množství draslíku se určuje konstantou, která podle literatury [27] dosahuje až 2000 mg/-den. Od konzultanta jsem se dozvěděl, že většina pacientů na tuto konstantu stejně nedosáhne. Ovšem je důležité řídit se tím, co vychází z literatury.

2.5 Analýza potřeb uživatele

Od svého externího konzultanta pana Plevy jsem se dozvěděl potřeby pacientů na dialýze. Velmi důležité je pro ně dodržování diety a tato aplikace má sloužit jako pomůcka při počítání hodnot minerálů z potravy a dodržování doporučených dávek. Dále jsem také provedl průzkum mezi pacienty, abych lépe pochopil jejich potřeby a na druhou stranu do aplikace nedával funkcionalitu, která by nebyla potřebná.

Seznámení se s požadavky uživatele

Z konzultací jsem se dozvěděl, že pro uživatele je zejména důležité mít na výběr z širokého sortimentu potravin. Dále uživatelé vyžadují možnost textového vyhledávání mezi potravinami a výpočet hodnot pro určité množství. Aplikace by neměla pouze staticky ukazovat hodnoty pro předem definované množství. Uživatelé si chtějí vést záznamy a statistiky, které aplikace bude uchovávat a vést přehledy o potravinách, které přijali mezi dialýzami. Budou si tedy moci lépe spočítat a upravovat stravu na základě statistik uložených v aplikaci. Nebude nutné získané hodnoty uchovávat mimo aplikaci a bude k nim snadný přístup. Velký zájem je také o rozšiřitelnost databáze potravin, kde by mohli uživatelé přidávat nové, nezohledněné potraviny. Uživatelé by tedy mezi sebou mohli sdílet jejich přidání potravin a mohlo by to vést ke zkvalitnění aplikace. Pro uživatele není tak důležité uvedení všech minerálů a výživových hodnot. Některé tabulky jsou podle nich až moc složité, a proto mohou uživatele zmást, nebo jej od užívání odradit. Chtěné jsou pouze hodnoty, které jsou relevantní pro pacienty na dialýze (draslík, fosfor, tekutiny) a základní míry jako jsou energetické hodnoty, bílkoviny, tuky, sacharidy. Dále jsem se dozvěděl, že mnohým uživatelům se termíny dialýz občas mění, proto je dobré je moci upravit. Ať už jednorázově jedno období, tak i celý rozpis.

Průzkum mezi pacienty

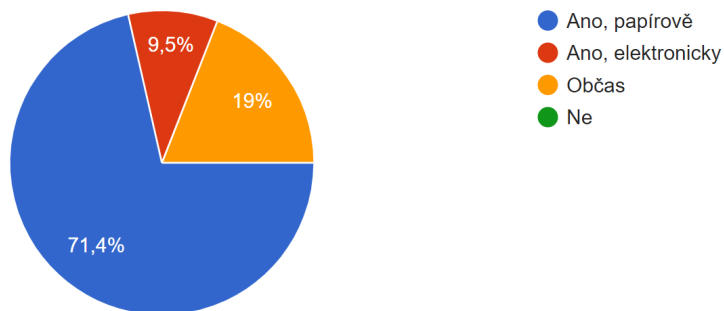
Pro přehlednost jsem také udělal průzkum mezi pacienty, kteří odpovídali na jednoduché otázky v dotazníku, který jsem pro ně připravil. Uživatelský průzkum je cennou součástí návrhu a vytváření aplikace. Dotazník byl vyplněn 21 pacienty na dialýze.

Z průzkumu na obrázku 2.4 vyplývá, že 80,9 % pacientů si eviduje zkonsumované potraviny. Bohužel pouze necelých 10 % si je eviduje elektronicky (Excel, Word). Těmto uživatelům by přechod na aplikaci mohl být nejsnazší. Z průzkumu také vyplývá, že všichni pacienti si aspoň občas evidují hodnoty, tudíž by aplikace měla aspoň základní využití mezi všemi vrstvami pacientů. Pro pacienty, kteří si hodnoty uchovávají jen občas (19 %) by mohla aplikace změnit jejich laxnější přístup a pomoci jim k lepší životosprávě a dodržování diety.

Další otázka v dotazníku 2.5 byla směřovaná na průměrný věk uživatelů. Je obecně známé, že s rostoucím věkem se jemná motorika uživatelů zhoršuje a může dojít i ke zhoršení zraku. Z dotazníku vyplývá, že 66,7 % pacientů je starších 50 let. Můžeme tedy předpokládat, že bude vhodné aplikaci držet v jednoduchém stylu a zvětšit velikost písma a

Evidujete si hodnoty z potravin?

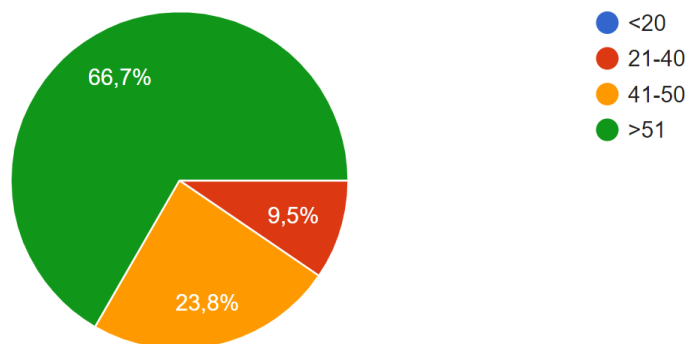
21 odpovědí



Obrázek 2.4: Odpovědi pacientů na otázku evidence potravin.

V jakém věkovém rozmezí se pohybujete ?

21 odpovědí



Obrázek 2.5: Odpovědi pacientů na otázku věkového rozmezí.

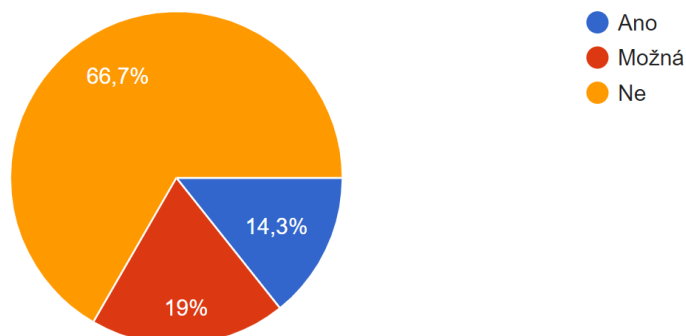
tlačítek. Dále 9,5 % uživatelů je ve věku od 21-40 let, tudíž s prací s telefonem jsou velmi dobře seznámení. A také 23,8 % uživatelů ve věku 41-50 telefon používají na denní bázi.

Předposlední otázka v dotazníku 2.6 byla položena na využití online synchronizace. Většina uživatelů (66,7 %) odpověděla, že tuto funkci nepotřebuje. Hlavním důvodem byl omezený přístup k internetu a fakt, že používají aktivně pouze jedno zařízení, tudíž tato funkce pro ně pozbývá smysl. Na druhou stranu 14,3 % uživatelů by ji aktivně využilo a 19 % ji vyzkoušela. Tudíž je možnost k využití této funkce přimět až 33,3 % uživatelů.

Poslední otázka dotazníku 2.7 byla položena na téma sledovaných hodnot. Valná většina uživatelů uvedla, že sleduje hodnoty, které jsme již dříve vybrali. Tudíž můžeme uvažovat, že rozsah sledovaných hodnot je dostatečný a můžeme bílkoviny a kalorie vynechat z důvodu nízkého zájmu.

Využili byste online synchronizaci?

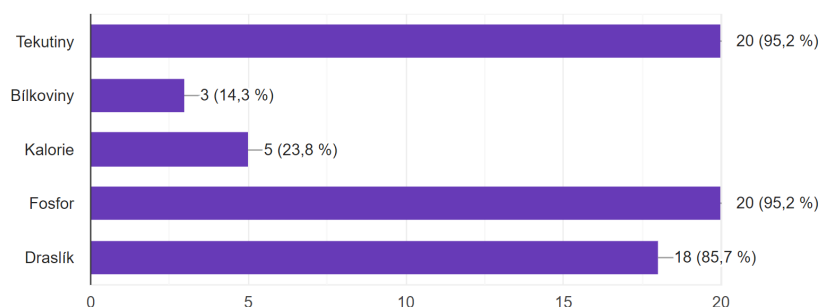
21 odpovědí



Obrázek 2.6: Odpovědi pacientů na otázku online synchronizace.

Které hodnoty sledujete ?

21 odpovědí



Obrázek 2.7: Odpovědi pacientů na otázku sledovaných hodnot.

2.6 Průzkum aktuálních řešení

V současné době existuje pouze velmi omezené množství řešení. Většina pacientů i lékařských dialyzačních zařízení si záznamy uchovává v písemné formě. Mezi hlavní nevýhody písemných záznamů patří jejich omezená životnost (vliv času na inkoust a papír) a možnost snadné ztráty a nepřesností podávaných informací pacientem lékaři. V neposlední řadě může také dojít k odcizení a zneužití osobních informací. Možnost odcizení identity pacienta může nastat také i v elektronické podobě, ale při správném použití aktuálních zabezpečovacích algoritmů (například hašování hesel a uchovávání identity pod číslem na databázové úrovni, nebo zabezpečená internetová komunikace například pomocí protokolů SSL nebo TLS) se tato možnost minimalizuje. Téměř všechny aplikace jsou určeny pro světový trh, tudíž jsou v anglickém jazyce bez možnosti překladu do češtiny. Dalším nepříznivým faktorem může být také zpoplatnění aplikace nebo její velmi strohá až neúplná funkčnost.

Český trh

Na českém trhu se můžeme setkat pouze s jedinou aplikací myCompanion¹, a to od společnosti NephroCare², která je určena pouze pro pacienty navštěvující jejich dialyzační centra, tudíž není dostupná pro širokou veřejnost. Jak lze vidět na obrázku 2.8 aplikace zobrazuje výsledky laboratorních testů, záznamy předepsaných léků a postupu léčby. Nabízí sekci s tipy a radami k dietě a dialýze společně s kuchařskými recepty, z kterých může při přípravě pokrmů uživatel čerpat a lépe zvládat speciální dietu. Chybí jí možnost výpočtu přibližných výživových hodnot z přijaté potravy.

Z hodnocení na Google Play³ vyplývá, že není uživatelsky přívětivá. Kritizovanými faktory jsou zejména nepřehlednost a chybějící vysvětlivky u použitých zkratk a pojmů. Víceero uživatelů není spokojeno s funkčností aplikace a aktualizacemi výsledků.

Aplikaci jsem neměl možnost vyzkoušet, protože je určena pouze pro klienty navštěvující střediska NephroCare.



Obrázek 2.8: Aplikace Renal Diet Adviser - ukázka rozhraní

Globální trh

Průzkum

Na světovém trhu se vyskytuje víceero aplikací určených ke kontrole dialýzy včetně již dříve zmiňované myCompanion¹ od společnosti NephroCare².

¹https://play.google.com/store/apps/details?id=com.fresenius.MyCompanionNephroCare&hl=cz_IE

²<https://www.nephrocare.cz/>

³<https://play.google.com/store/apps/details?id=com.fresenius.MyCompanionNephroCare&hl=cs&gl=US&showAllReviews=true>

Z průzkumu provedeného na Google Play⁴ vyplynulo jen několik možných řešení, které se přibližují dané tématice. Tyto aplikace ovšem fungují pouze jako klasické kalorické tabulky. Hodnoty, které z potravin počítají, jsou relevantní pro pacienty na dialýze, nicméně denní, nebo dlouhodobé přehledy nejsou uchovávány. Pro uživatele jsou tato řešení vhodná spíše pro jednorázový výpočet. Většina aplikací je zaměřena na rady a tipy při léčbě. Dále ukazují příklady pro dietu a recepty, pro zdravý životní styl. V neposlední řadě také počítají látky, které byly po provedení dialýzy vyloučeny z těla nebo pro určení dávky dialyzační směsi.

Aplikace Renal Diet Adviser⁵

Tuto aplikaci jsem zvolil kvůli pozitivním uživatelským hodnocením. Navíc se tato aplikace částečně přibližuje mé vyvíjené aplikaci. Byla vyvinuta Mahmoudem AbuZnaidem a publikována v roce 2019, má více než 5000 stažení.

Aplikaci jsem vyzkoušel z uživatelského pohledu. Mezi hlavní výhody patří distribuce zdarma, možnost volby ze 2 jazyků. Dalším pozitivem je intuitivní navigace a jednoduchost ovládání viz obrázek 2.9. Navigace v sekci potravin je logicky členěna podle druhů potravin. Uživatel také uvítá zobrazení pouze relevantních hodnot minerálů sledovaných při chronickém onemocnění ledvin. Uživatel si může zobrazit nabídku doporučeného denního příjmu minerálů podle stupně závažnosti onemocnění. V neposlední řadě ocení také rady s popisem speciální diety včetně tipů, kterým potravinám se vyhnout, případně které do svého jídelního plánu zařadit častěji.

Největší nevýhody jsou velice časté reklamy, které uživateli zneprůjemňují práci s aplikací. Téměř při každém kliknutí na jakoukoli položku se zobrazí alespoň jeden reklamní panel. Nabídka potravin je velice obecná a poměrně strohá. Složení potravin se může zásadně lišit a tudíž zkreslovat výsledné hodnoty. Jako příklad bych uvedl mléko. V této aplikaci je pouze jeden typ mléka s průměrnými hodnotami, což je v dnešní době absolutně nedostačující. Na trhu je široká nabídka různých typů mléka (například nízkotučné, bezlaktózové, plnotučné, apod.). Každý typ má jiné výživové hodnoty, které se mohou markantně lišit. Další nevýhodou je nemožnost uchovávání dlouhodobějších dat. Překážkou ve snadném používání může být nemožnost vyhledávání jednotlivých potravin podle názvu a indispozicí možností výpočtu sledovaných hodnot na specifické množství potravy (vše je určeno pro dávku na 100 gramů).

Dialysis Calculator⁶

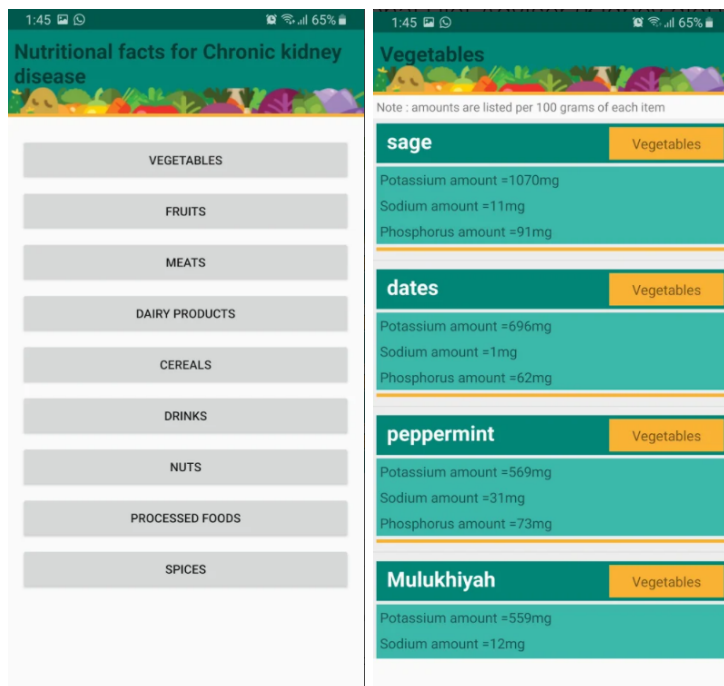
Tato aplikace se nejlépe podobá tématu této práce. V obchodě Google Play je již delší dobu a nasbírala přes 10 000 stažení. Aplikace je kladně hodnocena a aktivně využívána převážně lékařským personálem. Aplikace byla vytvořena za spolupráce s oddělením nefrologie jedné z univerzit v Chile.

Pacient zadává typ a frekvenci dialýz. Zadává také svoji váhu a diurézu. Aplikace se ovšem zaměřuje na výpočty hodnot, které jsou během dialýzy vyloučeny z těla pacienta, hlavně tedy kreatininu. Více než pro pacienty je tato aplikace směřovaná zdravotnickému personálu, který řídí množství odvedených látek z těla pacienta. Jak je vidět na obrázku 2.10 aplikace disponuje obecnými výpočty jako je například BMI (*Body mass index*) k posouzení váhy pacienta. Při používání můžeme využít i výpočet dávkování dialyzačního roztoku a maximální ultrafiltraci, kterou ovšem běžný pacient nereguluje. Bohužel tato aplikace

⁴https://play.google.com/store/search?q=dialysis%20calculator&c=apps&hl=en_IE

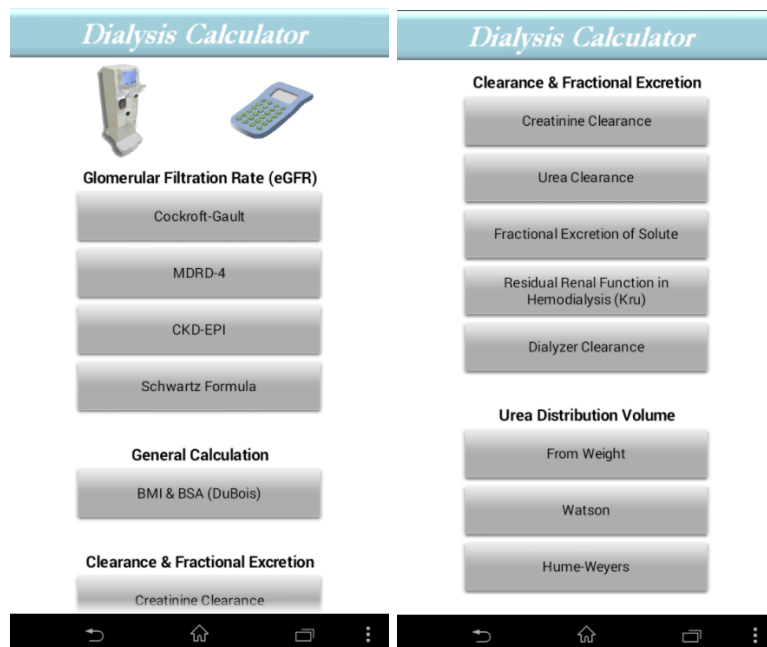
⁵<https://play.google.com/store/apps/details?id=com.mahapps.renaldietadviser>

⁶https://play.google.com/store/apps/details?id=appinventor.ai_medicina_android.Dialysis_Calculator



Obrázek 2.9: Aplikace Renal Diet Adviser - ukázka rozhraní

nedisponuje kalorickými tabulkami a pacient si nemůže vypočítat hodnoty z potravy. Dalším negativním bodem je také nepřívětivé uživatelské rozhraní, které může značně znepříjemnit práci s aplikací.



Obrázek 2.10: Aplikace Dialysis Calculator - ukázka rozhraní

Zhodnocení existujících řešení

Z průzkumu vyplývá, že existují aplikace, které jsou určeny pro pacienty na dialýze. Některé slouží spíše jako odborné znalosti pro lékařský personál a pomáhá jim například s určením maximální ultrafiltrace nebo množství dialyzačního roztoku. Některé aplikace slouží pouze jako kalorické tabulky rozšířené o sledované hodnoty (fosfor, draslík, tekutiny), ale neumožňují jejich výpočet ani uložení. Také existuje aplikace, která je určena pouze pro klienty speciálních dialyzačních center a jsou do ní nahrávány výsledné hodnoty po konci dialýzy. Ovšem ani jedna aplikace se nepřibližuje řešené tématice. Na druhou stranu jsem čerpal z jejich uživatelského rozhraní a ze zpětné vazby uživatelů, kteří některé prvky příliš neocenili a snažil jsem se jim vyhnout.

V aplikaci, která vzniká je hlavní předností fakt, že aplikace si bude uchovávat mezidialyzační období a bude sama pomocí předdefinovaných vzorců počítat maximum sledovaných hodnot fosforu, draslíku a tekutin. Uživatelé budou moci z databáze potravin vybrat položku a aplikace jim spočítá hodnoty pro dané množství a persistentně je uloží. Dále si také budou moci vést dlouhodobé statistiky a ukládat data na server. Což žádná z již existujících aplikací nenabízí.

Kapitola 3

Návrh aplikace a databáze

V této fázi je nutné navrhnout uživatelské rozhraní a datovou strukturu. Byly zahrnuty poznatky z literatury a poznámky konzultanta a uživatelů. Návrh aplikace byl prováděn společně s konzultantem. Tento přístup mi přišel vhodný hlavně z důvodu, že i on je sám pacientem na dialýze a je možné s ním koncepci uživatelského rozhraní prodiskutovat a lépe vystihnout potřeby uživatelů.

3.1 Návrh datového modelu

Pod pojmem datový model si můžeme představit popis objektů reality, reprezentujících dat, vztahů mezi nimi, sémantiky a integritních omezení [36]. Při vytváření datového modelu byly zohledněny dotazníky a také rady, které jsem během průzkumu získal. Z nich byl vytvořen diagram případů užití.

Diagram případů užití se využívá hlavně pro vystihnutí chování systému z uživatelského pohledu. Dá se tedy říct, že jde o specifikaci případů, které bude uživatel provádět a které by systém měl zvládat. Díky tomuto diagramu, jsme schopni jako tvůrci aplikace lépe pochopit potřeby uživatele.

Z dřívějších zkušeností implementace databáze jsem pro jednoznačný identifikátor použil GUID¹ (*Globally Unique Identifier*), kvůli jeho velmi rozsáhlé škále hodnot (128 bitové číslo). Dá se říct, že GUID je shodný s UUID² (*Universally Unique Identifier*), ale je spojován s technologií od společnosti Microsoft.

Při vytváření návrhu byl zhotoven ERD (*Entity Relationship Diagram*). Entitně vztahový diagram se využívá pro návrh databáze. Můžeme na něj referovat jako na model, který zohledňuje reálné objekty a vztahy, které se mezi nimi vyskytují. Při správném posouzení a návrhu nám může velmi zjednodušit návrh databáze.

Diagram případů užití

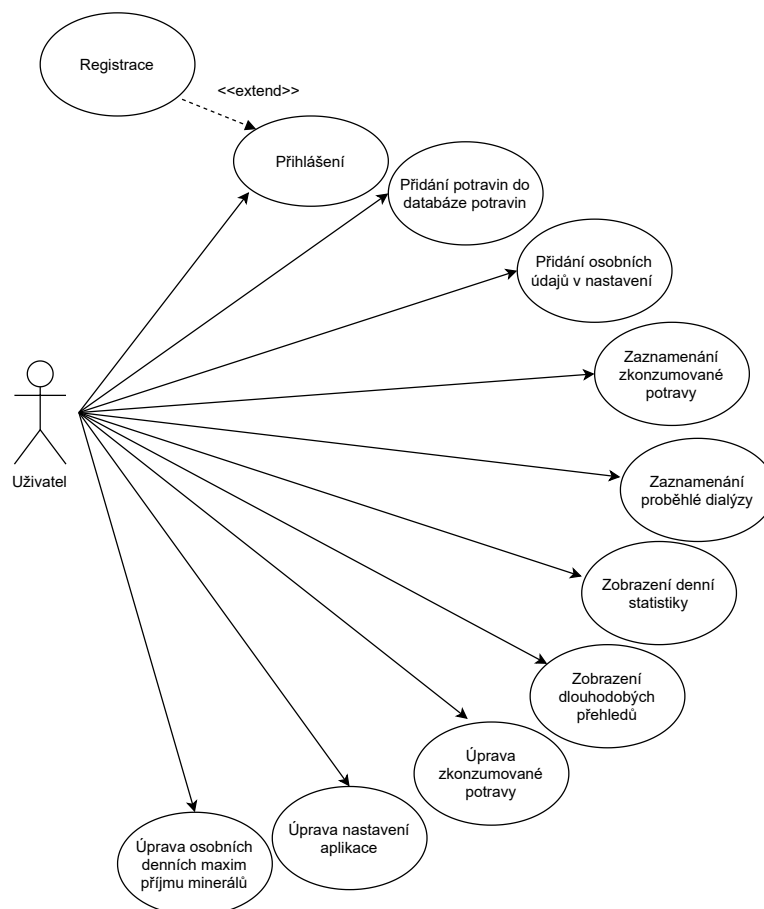
Diagram případů užití reflektuje vztahy mezi aktéry a případy užití systému. Tyto případy popisují posloupnost akcí, které poskytují měřitelnou hodnotu pro aktéra. Příklad užití je nakreslen jako vodorovná elipsa na diagramu [1].

V této aplikaci je v roli aktéra pacient na dialýze viz obrázek 3.1. Pacient si sám spravuje svůj uživatelský účet. Dále si také nastavuje dialyzační intervaly a osobní údaje. Jeho hlavním zájmem je přidávat potraviny a kontrolovat živiny, které z nich přijal. Je možné také

¹<https://docs.microsoft.com/en-us/dotnet/api/system.guid?view=net-5.0>

²<https://tools.ietf.org/html/rfc4122>

zpětně upravit množství potravin, které v rámci současného období zkonsumoval. Může se také podívat na již uběhlá období, kde budou nejen přijaté hodnoty z potravin, ale také všechny zkonsumované potraviny.



Obrázek 3.1: Diagram případů užití

Návrh databáze

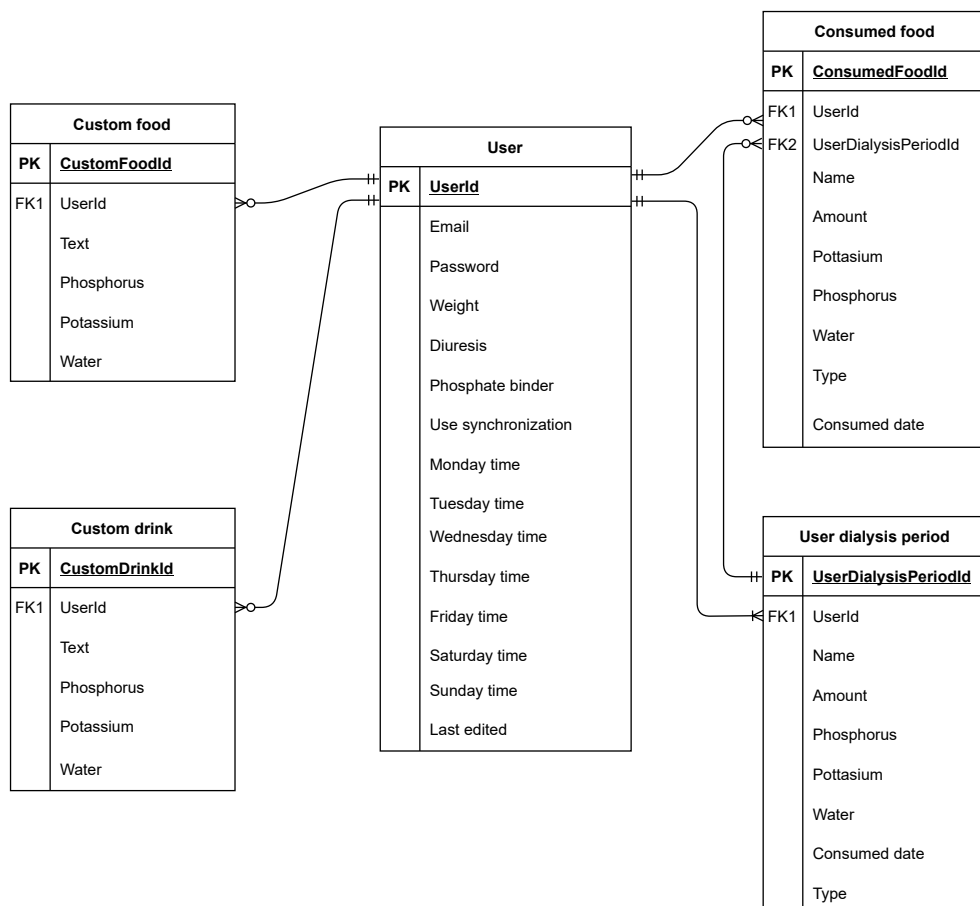
Při vytváření návrhu databáze byl zhotoven ERD diagram ve volně dostupném nástroji draw.io³. Pro snadnější pochopení je dobré si vyjasnit základní pojmy z oblasti modelování databáze [36]:

- **Entita** - věc nebo objekt reálného světa rozlišitelný od jiných objektů, o níž chceme mít informace v databázi. V souvislosti této práce je entita například: uživatel nebo potravina.
- **Atribut** - vlastnost entity, která nás v kontextu daného objektu zajímá a chceme si ji uchovat. Například váha, datum narození a množství diurézy uživatele.
- **Vztah** – asociace vícero entitami. Například zkonsumovaná potravina náleží k uživateli a dialyzačnímu období.

³<https://www.diagrams.net/> – Webový nástroj pro tvorbu diagramů.

- **Klíč (primární klíč)** – z hlediska databáze je nutné, aby entity byly jednoznačně identifikovatelné. To samé platí i pro vztahy mezi nimi. Tudiž musí existovat jednoznačný identifikátor.

Ze sekce případů užití už známe všechny úkony, které bude uživatel s aplikací vykonávat. Všechny tyto akce vedly k navrhnutí entit **3.2**, které budeme v databázi vytvářet. Lokální i vzdálená databáze budou používat stejné definice tabulek, abychom předešli přemapování a ztrátě informací. Při vytváření návrhu databáze jsem se snažil vytvořit takový návrh, aby byl dobře rozšiřitelný při dalším navazujícím vývoji aplikace.



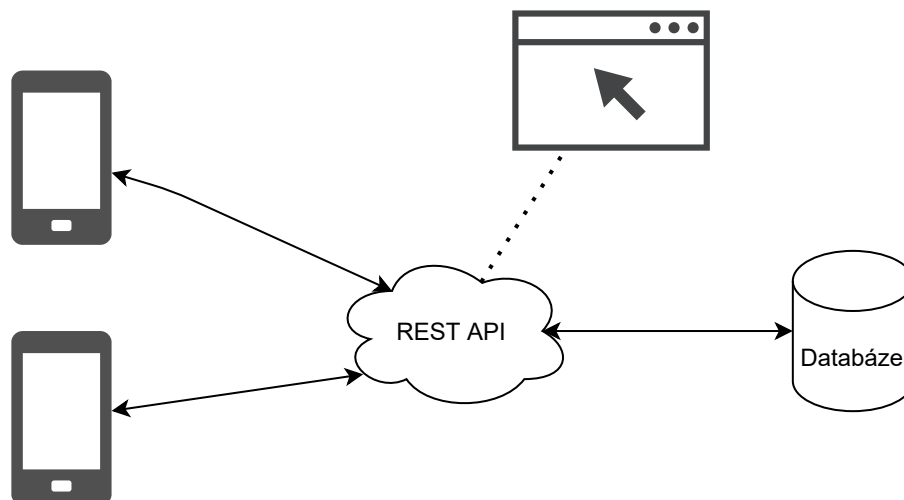
Obrázek 3.2: ER diagram

3.2 Návrh serverové části

Z průzkumu mezi uživateli vyšlo, že o serverovou část uživatelé jeví menší zájem. Ovšem je dobré myslet na rozšiřitelnost aplikace, a proto je potřeba zavést online synchronizaci, kterou bude zastřešovat REST API⁴, které během vývoje aplikace vznikne. Uživatelé budou muset být při registraci připojeni k internetu, abychom byli schopni zachytit jejich emailovou adresu a také jednoznačný identifikátor kvůli možným duplicitám. Podle schématu **3.3** je bráno v úvahu možné rozšíření i na webový prohlížeč. Tudiž by koncové body byly

⁴<https://www.redhat.com/en/topics/api/what-is-a-rest-api>

částečně připraveny na propojení s webovou stránkou a také možné rozšíření v průběhu implementace.



Obrázek 3.3: Návrh funkce serveru

3.3 Návrh uživatelského rozhraní

Uživatelské rozhraní v kontextu Human-computer interaction představuje součást počítačového systému, se kterou provádí uživatel interakce s cílem vykonat určitý úkol a která mu umožňuje dosáhnout jeho cílů [15]. Při návrhu je vhodné řídit se pravidly designu [25]:

- Snížit využívání paměti koncového uživatele
- Umístit uživatele do centra všeho dění
- Udělat přívětivé rozhraní napříč všemi skupinami uživatelů
- Přizpůsobení rozhraní pro potřeby uživatelů
- Snažit se o konzistenci ovládacích prvků
- Snažit se ošetřit všechny případy neočekávaných chyb
- Otestovat aplikaci co nejvíce uživateli

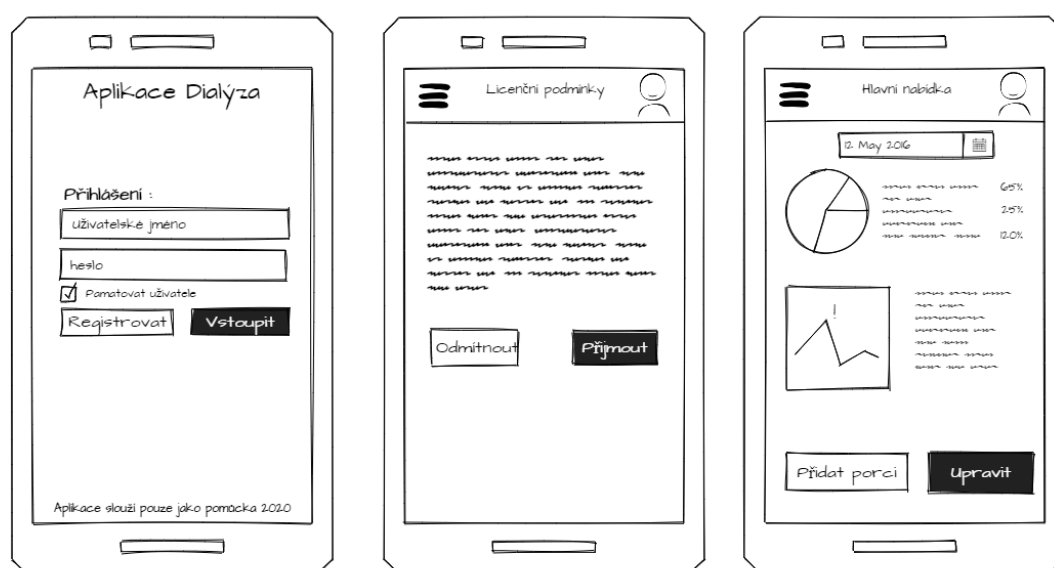
V této fázi řešení jsem začal navrhovat, jak bude uživatelské rozhraní vypadat, abych mohl hrubě začít s implementací a také mohl vytvořit mockup návrh, od kterého bude možné se odrazit. Pro vytvoření mockup návrhu jsem použil službu Mockflow⁵.

Při návrhu jsem začal přihlašovací obrazovkou na obrázku 3.4 vlevo, která bude obsahovat pouze název aplikace v horní části. Uživatel zadá své přihlašovací údaje do dvou oken, jedno pro přihlašovací jméno a druhé pro heslo. Po zadání informací uživatel klikne na tlačítko pro přihlášení a bude přesměrován na hlavní nabídku. Uživatel může zaškrtnout políčko pro zapamatování uživatele. Pokud není registrovaný, klikne na tlačítko registrovat a bude přesměrován na obrazovku registrace.

⁵<https://mockflow.com/app>

Po zaregistrování musí uživatel přijmout licenční podmínky na obrázku 3.4 uprostřed, které se vztahují k používání aplikace a hlavně odkazuje na fakt, že aplikace je pouze pomůcka a nikoli certifikovaná lékařská aplikace. Uživatel má na výběr ze dvou tlačítek. Pokud licenční podmínky přijme je přeměrován na hlavní stránku. Pokud s licenčními podmínkami nesouhlasí je registrace zrušena.

Přihlášený uživatel je odkázán na hlavní obrazovku na obrázku 3.4 vpravo, kde hlavní část tvoří údaje o hodnotách minerálů a živin, které během dne přijal. Hodnoty a jejich hladina budou zobrazeny grafy a ukazateli pro snazší orientaci. V dolní části obrazovky budou dvě tlačítka. Tlačítko přidat porci přeměruje uživatele na obrazovku databáze potravin, kde si vybere potraviny, které pozřel a následně se spočítají hodnoty a výsledek se promítne na hlavní obrazovce. Tlačítko upravit bude sloužit k upravení již zadaných hodnot, protože uživatel může udělat chybu a je žádoucí, aby mohl své již zadané hodnoty měnit.

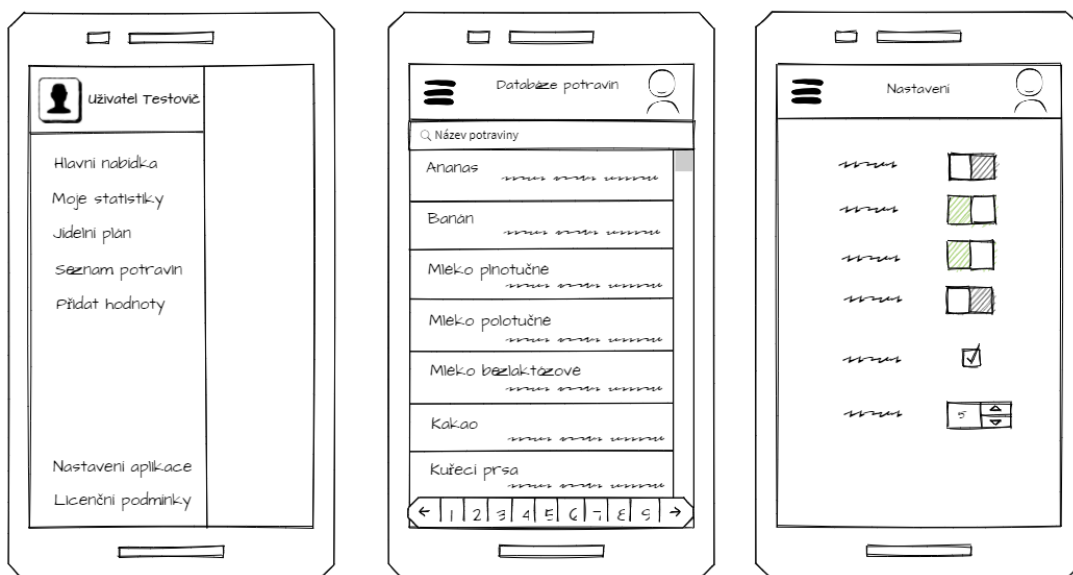


Obrázek 3.4: Návrh přihlašovacího okna, licenční smlouvy a hlavní strany

Na obrázku 3.5 vlevo je návrh bočního menu, kde bude moci uživatel vybírat z oken, která jsou mu dostupná a ulehčí mu navigaci mezi obrazovkami. Bude moci vybrat nejen ze stránek, které slouží pro kontrolu diety a výběru potravy, ale také na stránku s nastavením a licenčními podmínkami. Nastavení bude dvojího typu. Uživatel si bude moci přizpůsobovat aplikaci nastavením zobrazení. Tyto úpravy budou mít dopad na vzhled aplikace. Druhým způsobem úpravy bude možnost změnit maximální hodnoty minerálů a živin a také změnit živiny, které jsou pro něj relevantní a chce je sledovat.

Databáze potravin je na obrázku 3.5 uprostřed. Jedná se o jednu z nejrozsáhlejších částí aplikace. Na obrazovce je zobrazen seznam potravin. Pro jeho rozsáhlost bude členěn na několik stránek. Navigace mezi stránkami je umístěna v dolní části obrazovky. Jednotlivé potraviny mají uvedené relevantní hodnoty vždy na 100g/ml. Při kliknutí na řádek potraviny se zobrazí okno detailu potraviny a následně bude možné vybrat hodnoty pro specifické množství. Pro zjednodušení vyhledávání uživatele je potraviny možné vyhledávat podle názvu.

Nastavení je na obrázku 3.5 vpravo. Uživatel si může nastavit a personalizovat aplikaci podle jeho preferencí. Návrh tohoto okna bude podobný, jak při nastavení aplikace, tak i při nastavení měřených hodnot uživatelem a jejich množství.



Obrázek 3.5: Návrh bočního panelu, databáze potravin a nastavení aplikace

Kapitola 4

Implementace

Tato kapitola je zaměřena na implementační detaily spojené s vývojem mobilní aplikace a REST API. Na začátku kapitoly se seznámíme s výběrem vhodné technologie (Xamarin.Forms a .NET Core) a následují principy, kterými jsem tuto problematiku řešil.

Po uvedení do použitých technologií a výběru způsobu uložení dat se podíváme, jak bylo implementováno uživatelské rozhraní a na koncepci mobilní aplikace, jako je například registrace, přihlašování a komunikace se serverem. V další části se zaměříme na koncepci a implementaci rozhraní REST API. Všechny tyto implementační kroky vedly k vytvoření výsledného řešení.

4.1 Použité technologie

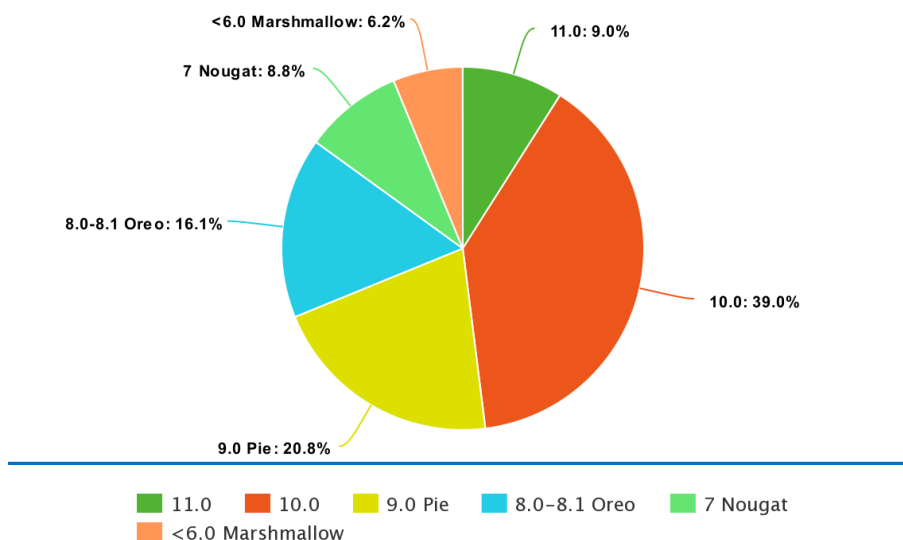
Tato sekce slouží jako znalostní báze pro ucelení použitých technologií. Aplikace je určena pro operační systém Android. Pro vývoj byla použita platforma Xamarin, která je vhodná pro vývoj mobilních aplikací a také umožňuje rozšíření aplikace na operační systém iOS a Windows phone, který se v dnešní době téměř nepoužívá. V této kapitole bude popsán výběr zvolených technologií, jejich porovnání a důvod výběru. Bude blíže specifikován jazyk C#, který Xamarin a .NET používá. V neposlední řadě bude popsána databáze i prostředí včetně serveru, kde bude nasazena.

Operační systém Android

Operační systém Android je v dnešní době nejpoužívanější operační systém mezi mobilními telefony. Android je primárně určen pro chytré zařízení jako je například: telefon, tablet, hodinky nebo televize. První komerční vydání vyšlo v roce 2008 v telefonu od společnosti HTC. Uživatelské rozhraní bylo velmi strohé, ale zanedlouho dosáhlo obrovské komunitní oblibě. Mezi hlavní výhody rozhodně patří za zmínku jeho open-source distribuce. Operační systém je možné volně stáhnout a upravit podle potřeb. Jedním z hlavních důvodů je jeho rozšířenost mezi většinu mobilních výrobců. Každý výrobce si svoji verzi upraví podle potřeb, a proto je velmi těžké tvořit stabilní verze, kompatibilní se všemi zařízeními. V roce 2005 byl koupen organizací Google. Ovšem na distribuci to nic nezměnilo. Google využívá operační systém hlavně pro svá zařízení, jejichž počet v dnešní době stále roste [23].

Verze Androidu

Většina verzí toho operačního systému je vydávána s přezdívkami. Tato jména se odvíjejí od sladkostí. Nejnovější verze Androidu v současné době je verze 11, která společně s verzí 10 po dlouhé době nenese žádnou přezdívku. Na obrázku 4.1 si můžeme všimnout, že Android 10.0 vede s 39 % zařízení, druhým nejoblíbenějším je Android 9.0, který využívá 20,8 %, na třetím místě Android 8.0-8.1, který využívá 16,1 % zařízení a již dříve zmiňovaný Android 11.0, který využívá už 9 % všech zařízení [2].



Obrázek 4.1: Graf rozložení verzí Androidu na trhu

Technologie pro vývoj mobilních aplikací pro Android

Pro vývoj aplikací pro Android se dá použít dvojího přístupu. Je možné vyvíjet nativní aplikaci pomocí specifických nástrojů. Druhým přístupem je vývoj multiplatformní aplikace. Hlavní výhodou multiplatformních aplikací je paralelní vývoj aplikace pro více platform a může to zrychlit a zjednodušit jejich vývoj.

Kotlin

Kotlin je open-source staticky napsaný programovací jazyk, který podporuje jak objektově orientované, tak funkční programování. Byl vytvořen firmou JetBrains¹. Kotlin poskytuje podobnou syntaxi a koncepty z jiných jazyků, včetně C#, Java a Scala. Kotlin nemá za cíl být jedinečný - místo toho čerpá inspiraci z desetiletí vývoje jazyka. Existuje ve variantách, které cílí na JVM (Kotlin / JVM), JavaScript (Kotlin / JS) a nativní kód (Kotlin / Native).

Mezi hlavní výhody se řadí hlavně jeho snadné použití s ohledem na funkční programování. Syntaxe je velmi podobná jazyku Java. Při správném strukturování kódu je velmi dobře čitelný a přehledný. Pokud se zaměříme pouze na funkční programování, bude nám stačit méně kódu. Je optimalizovaný na Android zařízení, tudíž jeho doba zpracování a běhu je velmi dobrá. Umožňuje deklarovat funkce na nejvyšší úrovni, což může vést ke zmatku při volání stejné funkce.

¹<https://www.jetbrains.com/>

React Native

React Native je JavaScript framework pro psaní nativních mobilních aplikací pro iOS a Android. Je založen na knihovně JavaScriptu. Vytvořený Facebookem pro vytváření uživatelských rozhraní na mobilní platformy. Jinými slovy: weboví vývojáři nyní mohou psát mobilní aplikace, které vypadají a působí skutečně „nativně“, a to z pohodlí knihovny JavaScript. Navíc, protože většinu kódu, který napíšete, lze sdílet mezi platformami, React Native usnadňuje simultánní vývoj pro Android i iOS. Podobně jako React for the Web, React Native aplikace jsou napsány kombinací JavaScriptu a JSX. Nativní renderování je uskutečněno za pomoci Objective-C pro iOS nebo Javy pro Android. Aplikace se tedy bude vykreslovat pomocí skutečných mobilních komponent uživatelského rozhraní, nikoli pomocí webview, a bude vypadat jako každá jiná mobilní aplikace [26].

Výhody můžeme vidět hlavně v rychlosti, protože jsou využity nativní komponenty, kde můžeme ocenit i úpravy vzhledu při vývoji. Předností může také být multiplatformní vývoj. Při vývoji rozsáhlých aplikací React ztrácí na rychlosti. Při použití náročných animací a vizualizace aplikace není plynulá.

Flutter

Flutter je multiplatformní sada nástrojů pro vývoj multiplatformních aplikací, která je navržena tak, aby umožňovala opětovné použití kódu napříč operačními systémy jako jsou iOS a Android, a zároveň umožňuje aplikacím využívat rozhraní přímo se základními službami platformy. Plánované rozšíření by mohlo přinést vývoj pro web a desktop. Cílem je umožnit vývojářům dodávat vysoce výkonné aplikace, které se na různých platformách cítí přirozeně a přijímat rozdíly tam, kde existují, a zároveň sdílet co nejvíce kódu. Aplikace jsou psány v jazyce Dart.

Během vývoje běží aplikace Flutter na virtuálním počítači, který nabízí stavové rychlé načtení změn bez nutnosti úplné kompilace. Pro vydání jsou aplikace Flutter kompilovány přímo do strojového kódu, ať už podle pokynů Intel x64, ARM nebo do JavaScriptu, pokud cílí na web. Vzhledem k tomu, že je framework open-source je v něm mnoho balíčků rozšiřujících funkčnost technologie [31].

Jako výhodu můžeme brát rychlé načtení bez nutnosti úplné kompilace a velmi rychlý vývoj. Při složitějších aplikacích ztrácí na výkonu a koncové aplikace jsou větší než bývá zvykem.

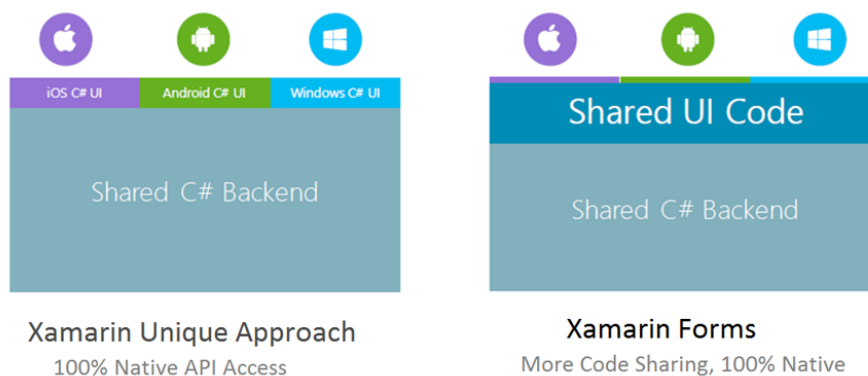
Xamarin

Xamarin je open-source platforma od společnosti Microsoft pro vytváření moderních a výkonných aplikací pro iOS, Android a Windows pomocí .NET. Xamarin je abstraktní vrstva, která spravuje komunikaci sdíleného kódu se základním kódem platformy. Xamarin běží ve spravovaném prostředí, které poskytuje pohodlí jako přidělování a uvolňování paměti. Xamarin umožňuje vývojářům sdílet průměrně 90 % svých aplikací napříč platformami. Tento model umožňuje vývojářům psát veškerou obchodní logiku v jednom jazyce (nebo znovu použít existující kód aplikace), ale dosáhnout nativního výkonu, vzhledu a chování na jednotlivých platformách [18]. Při vývoji aplikací se na této platformě využívá koncept MVVM (*Model-view-viewmodel*). Obrovskou výhodou Xamarinu je velmi rozsáhlá možnost používání nativních funkcí, tudíž je možné aplikaci velmi zrychlit. Aplikace Xamarin mohou být napsané na počítačích PC nebo Mac a kompilovat do nativních balíčků aplikací, jako je soubor `.apk` v Androidu nebo soubor `.ipa` v systému iOS [18].

Platforma Xamarin má mnoho výhod na základě kterých jsem se rozhodl využít tuto platformu. Jeden z hlavních důvodů je fakt, že aplikace vytvořené v Xamarinu jsou srovnatelně rychlé, jako nativní aplikace. Dále s technologií .NET jsem byl již dříve seznámen a práce s ní mi není cizí. Dále také vývojové prostředí Visual Studio² mi je velmi dobře známo a nebylo nutné je měnit. Zkušenosti jsem načerpal při vytváření projektů v předmětech ICS a IW5, které mi poskytly cenné zkušenosti a rady.

Xamarin.Forms

Xamarin.Forms je open-source framework, který využívá sdílenou logiku pro platformy Android, iOS i Windows Phone. Tento framework rozšiřuje základní řešení o sdílení téměř všeho obsahu. Hlavní výhodou je fakt, že sdílí i návrh uživatelského rozhraní. Klasický Xamarin sdílí pouze kód řízení. Tento koncept umožňuje sdílet veškeré definice uživatelského rozhraní včetně navigace, ale také i vyskakovací okna a animace. Na obrázku 4.2 je znázorněn hlavní rozdíl.



Obrázek 4.2: Rozdíl mezi klasickým Xamarinem a Xamarin.Forms [3]

NuGet

Vzhledem k tomu, že Xamarin rozšiřuje platformu .NET je možné využít všechny NuGet³ balíčky. Tyto balíčky rozšiřují funkcionalitu o nové funkce, které jsou s těmito nadstavbovými balíčky spojeny. Pro technologii Xamarin bylo vytvořeno mnoho balíčků pro rozšíření uživatelského rozhraní, zjednodušení, ale také zefektivnění funkcionality kódu, který aplikaci řídí. Během vývoje jsem využil více než 20 balíčků, které mi mimo jiné zjednodušili i přístup k databázi i přístup k objektům a práci s nimi.

Při vývoji aplikace a REST API jsem nejvíce ocenil balíky:

- **Entity Framework Core** - umožňuje vytvořit databázi přístupem code first, tudíž jsem vytvořil definici tabulek, které jsem pomocí migrací nasadil na server.
- **Microcharts.Forms** - tento balík jsem využil pro vytváření grafů.
- **Newtonsoft.Json** - při vytváření požadavků jsem využil serializaci i deserializaci.

²<https://visualstudio.microsoft.com/cs/>

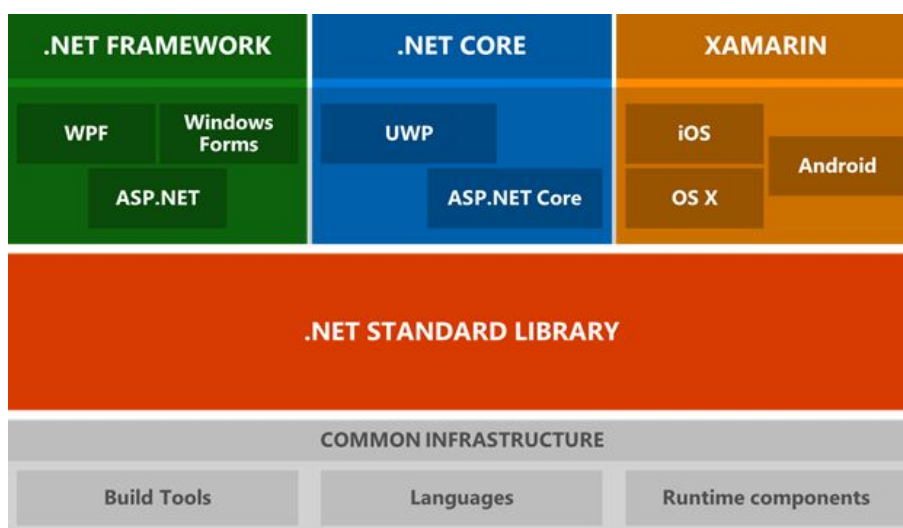
³<https://www.nuget.org/>

- **sqlite-net-pcl** - Tento balík byl použit pro vytvoření SQLite databáze v telefonu a přístupu k datům.
- **Swashbuckle.AspNetCore** - Tento balík byl použit pro vytvoření grafického znázornění koncových bodů na REST API.

.NET core

ASP.NET core je open-source framework, který byl vytvořen a využíván společností Microsoft. Je založený na technologii .NET Framework. Na obrázku 4.3 lze vidět, že architektura je velmi podobná standardnímu .NETu i Xamarinu. Slouží k vytváření aplikací pomocí technologie .NET. Používá se pro vývoj multiplatformních aplikací, ale také i pro využití v Dockeru⁴. Využívá principy MVC (Model View Controller), které velmi dobře od sebe oddělují jednotlivou logiku aplikace, a proto je dobře škálovatelná. Při vytváření aplikací můžeme využít mnoho knihoven a balíčků, které rozšiřují jeho funkcionalitu a zpříjemňují práci s touto technologií. Využívá jazyk C#.

Byl využit pro vývoj REST API, které používám pro přístup k datům do databáze a také pro jejich ukládání. Hlavní výhodou je fakt, že jsem jej ještě před nasazením mohl využít v rámci své lokální sítě a pomocí toho otestovat funkcionalitu aplikace.



Obrázek 4.3: Znázorněná architektura technologií .NET Framework, .NET Core a Xamarinu [17].

Jazyk C#

Je objektově orientovaný a typovaný jazyk, který v dnešní době nabývá velké oblibě. Je tomu tak kvůli kořenům vycházejícím z jazyka C, který je již velmi dobře známý vývojářské komunitě. Poprvé byl představen v roce 2002.

Jazyk C# spojuje základy C obohacené o rozšířenou funkcionalitu. Svým syntaxem je možné jej částečně přirovnat k jazykům jako je Java, C++ nebo i částečně k jazyku Python. Tudíž na sebe může přilákat vývojáře z různých odvětví. Od klasického jazyka C se například liší tím, že využívá automatický nástroj pro správu paměti (*garbage collector*) a

⁴<https://www.docker.com/>

to velmi zpříjemňuje práci s ním. Při deklaraci proměných lze využívat typ `var`, který sám určí typ proměnné. Dále také provádí typovou kontrolu a nelze přiřazovat různé typy. V C# neexistuje vícenásobná dědičnost, tudíž každá třída může být potomkem pouze jedné třídy. K tomuto rozhodnutí se přistoupilo kvůli obrovské složitosti, kterou by vícenásobná dědičnost přinesla. Na druhou stranu jedna třída může implementovat libovolný počet rozhraní.

Firestore

Firestore je platforma, která umožňuje vytváření a nasazení mobilních i webových aplikací. Společnost vznikla v roce 2011, ale v roce 2014 jej společnost Google koupila a vytvořila z ní hlavní nástroj pro vývoj aplikací. Firestore zprostředkovává širokou škálu funkcionalit. Mezi hlavní přednosti patří autentizace uživatelů, která probíhá přes API. Další vlastností je hostování aplikací a také jejich automatické testování. Dále také umí rozesílat uživatelům oznámení, třeba na jejich emailové adresy. Někteří vývojáři mohou ocenit také možnost vytvoření databáze. Nejedná se ovšem o relační databázi, ale pouze speciální typ ukládání dat na serverech společnosti Google.

V rámci bakalářské práce jsem využil tuto technologii pro registraci uživatelů a dále kontrolu jejich přístupu k REST API. Díky této technologii si také můžu vést anonymní statistiky o využívání přihlašování a registrace. Možnost ukládání dat jsem nevyužil kvůli nestandardnímu ukládání dat do „balíků“, namísto relační databáze.

4.2 Ukládání dat

Při používání aplikace uživatelem je nutné ukládat data, která se při používání shromažďují. Data nebudou obsahovat citlivé informace o uživateli. V záznamech budou pouze výsledky měření a kalorické tabulky. Tyto hodnoty není potřeba nijak zvlášť zabezpečovat. Pokud uživatel nebude mít přístup k internetu, budou se data ukládat lokálně na jeho zařízení. Při synchronizaci se data odešlou na server, kde se uloží a budou přístupná i z dalších zařízení.

Lokální úložiště

Při výběru vhodné metody bylo zvažováno několik přístupů. Data se mohla ukládat v textové podobě jako obyčejný formátovaný text, který by je dělil do sekcí. Tato metoda by vyžadovala určení specifické syntaxe a přístupu. Nebyla by efektivní a zbytečně složitá pro implementaci. Další z možných variant bylo data ukládat v JSON⁵ formátu. Pro posílání po síti by tento formát byl vhodný, ale cílem je použít relační model. Pro zjištění vhodného úložiště byl proveden průzkum možných relačních databází pro operační systém Android.

Realm DB

Realm DB je relační databáze vytvořená společností Realm. Funguje na stejném principu jako jsou klasické databáze. Data se dají šifrovat a tím pádem jsou bezpečně uložena, jak na serveru, tak i lokálně. Velkou předností je synchronizace dat na pozadí se serverem. Databáze si sama zjistí, zda je uživatel připojen k internetu a následně zkusí synchronizovat data. V tomto typu databáze je také možné přidělovat uživatelům práva aby měli přístup

⁵<https://www.json.org/json-cz.html>

pouze k určitým datům. Tento typ databáze se jeví jako velmi dobrý, ovšem ve spolupráci s Xamarinem není ještě úplně prozkoumán a vyskytují se v něm chyby.

Berkeley DB

Berkeley DB je open-source vysoce výkonná databáze, která dovoluje ukládat data různými způsoby. Data je možné ukládat do relační databáze pomocí dodané knihovny nebo také pojena pouze metodou klíč-hodnota jako bajtového pole. Dovoluje ukládat vícero dat pod stejným klíčem. Databázi vyvinula společnost Sleepycat Software, ale byla v roce 2006 koupena firmou Oracle. Berkeley DB je v určitých ohledech rychlejší než SQLite, na druhou stranu je výpočetně a paměťově náročnější. Bohužel pro tuto aplikaci je velikost důležitým faktorem a budou se provádět poměrně jednoduché zápisy, tudíž se tento typ databáze nejvíce jeví jako příliš vhodný.

SQLite

Je relační databázový systém, vyvinut D. Richardem Hippem. Databáze je uložena v samostatném souboru .dbm (Database Manager). Data jsou tam uložena za pomoci jednoduchého primárního klíče a pomocí hašovací funkce se k nim přistupuje. Systém je vyvíjen v jazyku C a je šířen pod licencí public domain (v oblasti angloamerické právní kultury nazývána díla, včetně počítačových programů, která vůbec, kupříkladu na základě rozhodnutí nositele práv, nepodléhají ochraně autorským právem) [29]. Xamarin při použití lokálního úložiště dobře spolupracuje s tímto typem databáze a tím padem se také nejvíce používá. Byl tedy vybrán jako způsob lokálního úložiště.

Uložení na serveru

Při výběru vhodné relační databáze byl zohledněn fakt, že databáze bude nasazena na platformě Azure⁶. Z tohoto důvodu bylo přistoupeno k použití Azure SQL⁷. Navíc se dobře používá ve spolupráci s Entity Frameworkem⁸, který se bude používat při implementaci databáze.

Azure SQL Database

Azure SQL Database je škálovatelná relační databázová služba vytvořená pro cloud provozovaná společností Microsoft. Je součástí řady Azure SQL, který je spravovaný databázový stroj typu PaaS (*Platform as a service*). Zpracovává většinu funkcí správy databází, jako je upgrade, opravy, zálohování a monitorování. Pomocí Azure SQL Database můžete vytvořit vysoce výkonnou vrstvu datového úložiště pro aplikace. Obrovskou výhodou této databáze je fakt, že umožňuje zpracovávat jak relační, tak i nerelační struktury. To se může hodit při zpracovávání XML, JSON nebo třeba i obrázky. Ovšem záleží na vrstvě předplatného, které si uživatel nebo společnost zakoupí [32].

⁶<https://portal.azure.com/#home>

⁷<https://azure.microsoft.com/cs-cz/services/sql-database/>

⁸<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

4.3 Mobilní aplikace

V sekci výběru technologie jsme se dozvěděli, který nástroj využijeme pro implementaci aplikace a v návrhu jsme si na skicách ukázali návrh uživatelského rozhraní. Během implementace se vzhled aplikace měnil a byl konzultován s pacienty a přizpůsoben jejich konkrétním potřebám. Také z poznatků při implementaci se měnila koncepce REST API a bylo upraveno nebo přidáno plno nových koncových bodů.

Jedním z nejdůležitějších prvků je vzhled aplikace. Vzhledem k tomu, že pokud uživatel bude aplikace nesympatická, bude mít odpor k tomu ji používat. Při konzultacích s externím konzultantem a jak z dotazníku 2.5 vyplynulo, je většina uživatelů starších 40 let. Z různých průzkumů je pro tento typ uživatelů nejpřívětivější jednoduché a zároveň světlé pozadí a vysoký kontrast. Při vývoji jsem vyzkoušel také tmavší styl, který ovšem uživatelům přišel spíše nepřehledný a komplikoval jim práci s aplikací. Dále jsem také zkoumal styl grafů. Ze zpětné vazby vyšlo, že pro zobrazení hodnot uživatelé nejvíce ocení koláčové grafy.

Obrazovka přihlášení a registrace

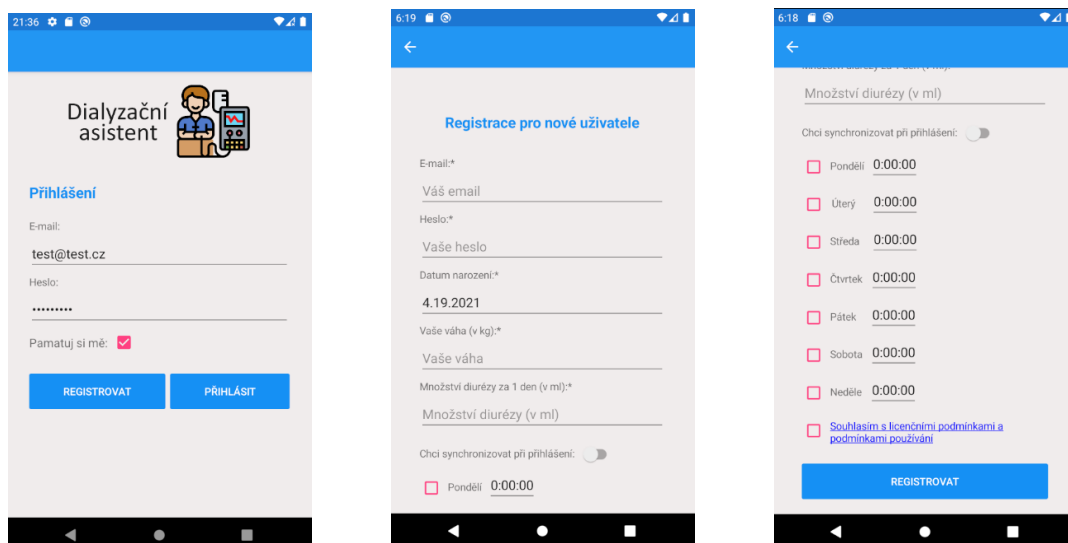
Při zapnutí aplikace se objeví obrazovka pro přihlášení, jak je vidět na obrázku 4.4(a). Na obrazovce jsou vidět dvě tlačítka (jedno pro přechod na stránku registrace a druhé pro přihlášení). Pro přihlášení uživatel zadá svůj email a svoje heslo. Pokud je uživatel online a má povolenou synchronizaci v nastavení aplikace, synchronizují se mu data se serverem. Dále také může vybrat možnost zapamatování si uživatele.

Na obrazovce pro registraci, která je znázorněna na obrázcích 4.4(b) a 4.4(c) uživatel musí vyplnit všechna povinná pole, zadat aspoň jeden termín dialýzy, protože dialýza vždy probíhá aspoň jednou za týden a také souhlasit s licenčními podmínkami. Pokud uživatel nevyplní některé z povinných polí zobrazí se mu dialogové okno, které ho informuje o tom, že některé povinné pole zůstalo nevyplněné. Při registraci je nutné mít internetové připojení. Pokud uživatel nedisponuje připojením k internetu, nemůže se registrovat.

Hlavní obrazovka, statistiky a boční navigace

Na hlavní obrazovce na obrázku 4.5(a) se uživateli zobrazí grafy sledovaných hodnot (voda, fosfor, draslík). Tyto grafy mají nastavené maximální a optimální hodnoty, kterých uživatel může v jedno období dosáhnout. Grafy mění barvu podle intervalu, ve kterém se nachází. Dále v horní liště na pravé straně může přidat zbytkovou vodu, která mu při dialýze nebyla z těla odstraněna a vstupuje tedy do daného intervalu s „dluhem“. Z uživatelské zpětné vazby vyplynulo, že je pro ně jednodušší mít možnost přidávat potraviny a nápoje hned na hlavní stránce. Proto byly ve spodní části stránky přidány tlačítka pro vložení zkonsumované potravy. Po kliknutí je uživatel přesměrován na stránku, kde si může vybrat, kterou potravinu nebo nápoj chce přidat.

Na obrazovce statistik na obrázku 4.5(b) můžou uživatelé vidět statistiky z období, která již proběhla. V horní části obrazovky kliknou na časový interval. Po kliknutí se jim objeví seznam již uplynulých období mezi kterými si můžou vybírat. Pro odlišení od statistik aktuálního období jsem zvolil sloupcové grafy, které svoji funkcionalitou jsou totožné s hlavní stránkou. Uživatel vidí hodnoty fosforu, draslíku a tekutin, které přijal. Pokud ho také zajímají konkrétní potraviny, které přijal klikne v dolní části obrazovky na tlačítko zobrazit zkonsumovaná jídla a je přesměrován na seznam již zkonsumovaných potravin. Tyto potraviny již nemůže měnit, protože změna hodnot v již uběhlém období není žádoucí.



(a) Přihlašovací obrazovka

(b) První část stránky registrace

(c) Druhá část stránky registrace

Obrázek 4.4: Obrázky z přihlašovací stránky aplikace a stránky pro registraci

Po rozkliknutí navigace uživatel uvidí všechna okna, kam se může přesměrovat, jak lze vidět na obrázku 4.5(c). Položky v seznamu navigace byly seřazeny podle uživatelských potřeb.

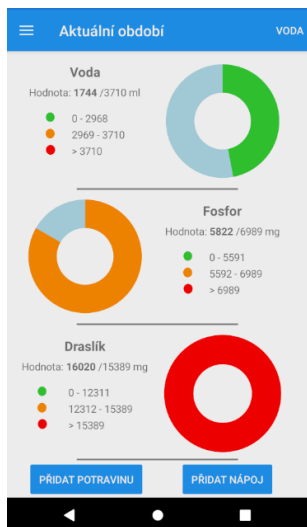
Zkonzumované potraviny, seznam potravin a nápojů

Na stránce zkonzumovaných potravin na obrázku 4.6(a) vidí všechny potraviny, které během aktuálního období zkonzumoval. Vidí zde množství a hodnoty vody, fosforu a draslíku které dané množství obsahovalo. Pokud chce uživatel změnit zkonzumované množství, klikne na danou potravinu a objeví se mu dialogové okno. Do tohoto okna zadá nové, upravené množství a po potvrzení se změny projeví v celé aplikaci.

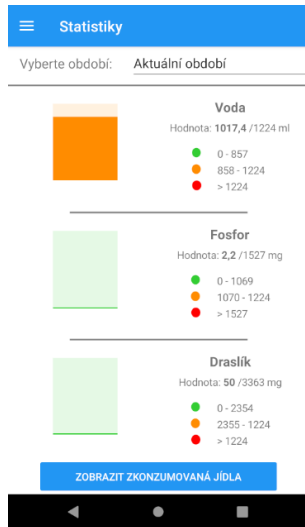
Seznamy potravin na obrázku 4.6(b) a nápojů na obrázku 4.6(c) byly rozděleny, aby se uživatelům s nimi lépe pracovalo. Z funkcionálního hlediska obě stránky fungují velmi podobně. Tyto stránky reflektují databázi potravin, kterou jsem od svého vedoucího dostal. Pokud uživatel chce přidat potravinu, klikne na ni a objeví se mu dialogové okno. Do něj zadá množství, které si přeje zadat a uloží je. Po uložení se mu položka ukáže v seznamu zkonzumovaných potravin. Pro větší přehlednost a zjednodušení vyhledávání mezi potravinami bylo přidáno fulltextové vyhledávání. Pokud uživatel bude chtít přidat vlastní potravinu pro rozšíření své databáze potravin klikne v horním pravém rohu na tlačítko přidat. Je přesměrován na stránku, kde zadá název potraviny a její hodnoty. Pokud jsou všechna pole vyplněná, má možnost potravinu přidat a bude se mu dále zobrazovat v seznamu všech potravin nebo nápojů.

Nastavení profilu a mezidialyzační období

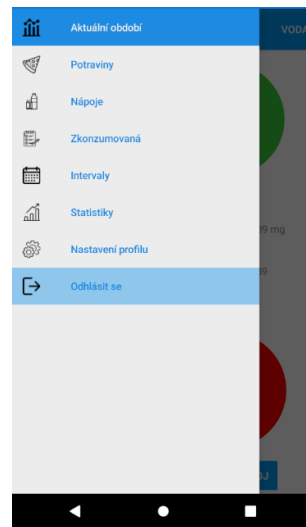
V sekci nastavení profilu na obrázku 4.7(a) může uživatel nastavit jeho suchou váhu, množství diurézy nebo také úplně změnit intervaly, ve kterých se dialýza bude konat. Tyto hodnoty uživatel téměř nemění, někteří uživatelé dokonce vůbec. Ovšem je žádoucí, aby



(a) Hlavní obrazovka



(b) Stránka statistik



(c) Boční navigace

Obrázek 4.5: Ukázka hlavní stránky aplikace, stránky statistik a boční navigace



(a) Stránka se zkonzumovanými potravinami

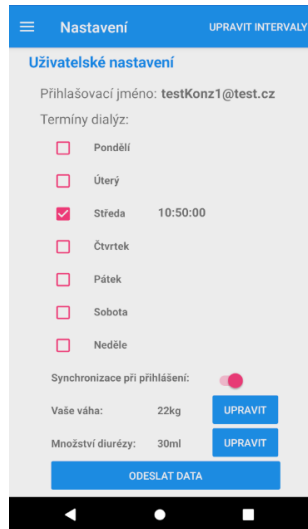


(b) Seznam potravin



(c) Seznam nápojů

Obrázek 4.6: Ukázka vzhledu stránky se seznamem potravin, mezidialyzačními intervaly a zkonzumovaných potravin



(a) Stránka nastavení profilu



(b) Stránka s mezidialyzačními intervaly

Obrázek 4.7: Ukázka obrazovky nastavení profilu a mezidialyzačních období

bylo možné tyto hodnoty změnit. Ve spodní části obrazovky je tlačítko na odeslání dat na server. Pokud uživatel změnil hodnoty špatně nebo není připojen k internetu, vyskočí mu dialogová hláška o neúspěšnosti akce. Pokud uživatel bude aktivně využívat možnost synchronizace, je možné si pomocí posuvného tlačítka zapnout nebo vypnout synchronizaci po přihlášení. Pokud se uživatel se zapnutou synchronizací přihlásí a není připojen k internetu, akce se neprovede a další pokus se vykoná až při dalším přihlášení.

Na stránce mezidialyzačních intervalů na obrázku 4.7(b) vidí uživatel následující období měsíc dopředu. Jsou generovány s předstihem kvůli faktu, že je možné, že se uživateli jednorázově změnil termín dialýzy a je žádoucí, aby měl možnost tento termín změnit. K provedení změny klikne na interval a po zobrazení dialogového okna změnil dobu, kdy se dialýza bude konat. Tuto funkcionalitu jsem do aplikace přidal po zájmu pacientů. Změny intervalů jsou ošetřeny podmínkami, aby se nestalo, že se jednotlivé intervaly kryjí.

Dialogová okna

Při implementaci rozhraní je důležité používat také dialogová okna, která uživatele upozorňují na důležité akce, které při běhu aplikace nastávají. S touto prací jsou dialogová okna použita z několika důvodů. Prvním z nich jsou informativní výpisy, které uživatele informují o zdaření, nebo také i neúspěchu akce, kterou chtěli provést. Dalším jsou vyskakovací okna, která nabízejí uživateli výběr z několika možností. Poslední částí jsou vyskakovací okna, kam uživatel zadává vstup. Příkladem může být dialogové okno při vložení nebo úpravě zkonsumovaných potravin.

4.4 Generování intervalů

K využívání aplikace je důležité správným způsobem uživateli generovat nadcházející intervaly. Intervaly se generují při registraci, ale také i při přihlášení uživatele. Získá se aktuální datum a k datu se přidá měsíc (intervaly se generují měsíc dopředu). Na začátku se přes

iterátor⁹ projde všemi dny, které se v intervalu nachází a probíhá kontrola, zda se den neshoduje se dnem ve který má proběhnout dialýza. Nový interval se uloží do listu nových intervalů. Po nalezení intervalu se nastaví začátek dalšího intervalu přesně za konec nalezeného a pokračuje v iteraci. Po nalezení všech možných intervalů je uloží do databáze.

4.5 Implementace výpočtů hodnot

Během implementaci výpočtu jsem se řídil konstantami, které jsme si již dříve definovali při teoretické části 2.4.

Při vytváření intervalů se vypočítá jeho délka. Z hodin, které se vypočítají se pronásobí konstantami a dalšími omezeními jako je množství diurézy. Na základě těchto výpočtů dostaneme maximální hodnoty, které může pacient v mezidialyzačním období přijmout. Při změně délky intervalů se přepočítají maximální hodnoty, které může pacient přijmout.

Při zadávání zkonsumovaných potravin se z přijatého množství vypočítají hodnoty, které potravina obsahuje. Uloží se do databáze a vypočítá se, zda uživatel nepřekročil denní limity. Při překročení denních limitů aplikace pacientovi ukáže dialogové okno s oznámením o tom, které hodnoty překračuje.

4.6 Synchronizace mezi zařízeními

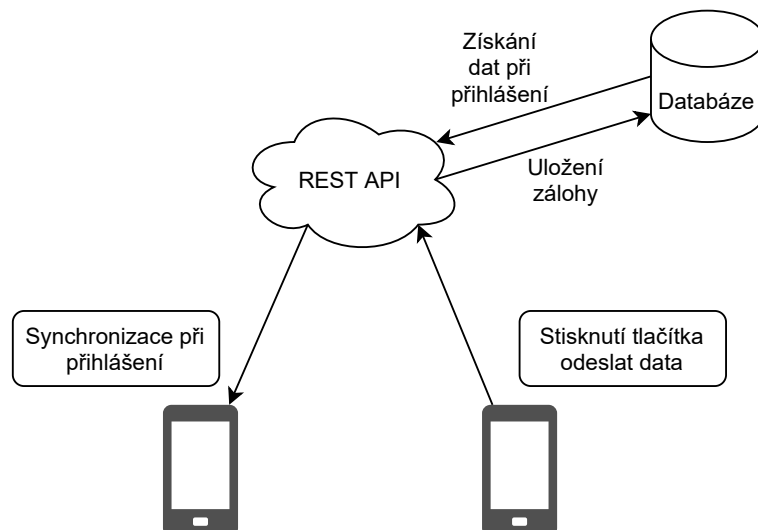
Během implementace bylo vytvořeno schéma synchronizace. Díky tomuto schématu jsem byl schopný synchronizovat informace mezi vícero zařízeními jak je vidět na obrázku 4.8. Uživatel si povolí synchronizaci při přihlášení a poté stiskne tlačítko pro odeslání dat v nastavení aplikace a pokud je připojen k internetu, data se odešlou na server. Pokud má uživatel na druhém zařízení přístup k internetu a povolenou synchronizaci, jsou data odeslána ze serveru na zařízení uživatele. Pokud uživatel nemá na tomto zařízení vytvořený účet, jsou mu odeslány všechny záznamy, včetně uživatelského účtu, tudíž může na druhém zařízení pracovat bez ztráty dat.

Uživatel si synchronizaci může kdykoli vypnout a není nutné ji využívat. Tato funkcionality byla vytvořena hlavně pro migraci z jednoho zařízení na druhé.

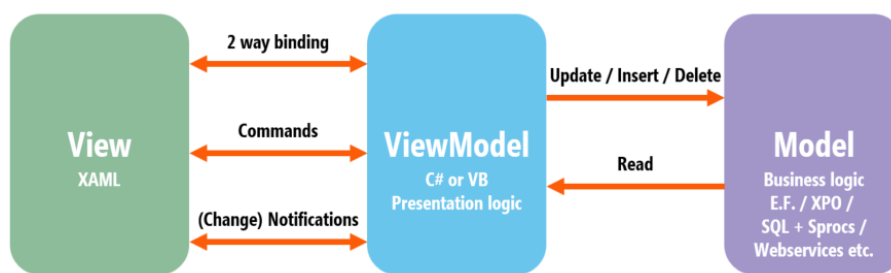
4.7 Architektonický vzor MVVM

Architektonický vzor MVVM (*Model-View-ViewModel*) rozděluje části aplikace do logických celků a odděluje od sebe jednotlivé části, jak lze vidět na obrázku 4.9. Využívání tohoto konceptu má za důsledek větší přehlednost vytvořeného kódu a také snazší hledání chyb, spojených s nepřehledným a rozsáhlým kódem. Model nám specifikuje jaké typy objektů pro danou obrazovku budeme používat a následně do nich ukládat hodnoty nebo z nich číst. View je zobrazovač jednotlivých elementů a používá se pro něj značkovací jazyk XAML. Viewmodel obsahuje logickou vrstvu. V této části přistupujeme k objektům a jejich vlastnostem a plníme je.

⁹<https://stackoverflow.com/a/1847601/14687208> - (Den návštěvy - 10.11.2020)



Obrázek 4.8: Schéma principu synchronizace mezi zařízeními.



Obrázek 4.9: Grafické znázornění konceptu MVVM [35].

4.8 XAML

Při implementaci uživatelského rozhraní se využívá jazyk XAML (*Extensible Application Markup Language*), který jsem využil pro deklaraci celého uživatelského rozhraní. Je to jazyk, který používá značky pro definici rozhraní. Velkou výhodou může být fakt, že tato technologie se využívá delší dobu ve WPF (*Windows Presentation Foundation*), která slouží k vytváření uživatelského rozhraní desktopových aplikací na platformě Windows. S touto technologií jsem byl již dříve seznámen a proto se mi s ním dobře pracovalo.

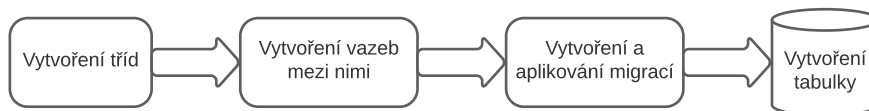
Všechny soubory asociované s tímto jazykem mají koncovku `.xaml`, kde se definuje vzhled rozhraní a v sdruženém souboru, který má koncovku `.cs` je vypracovaná logika, která například vkládá hodnoty do textových polí, nebo řídí co se stane po kliknutí na tlačítka.

4.9 Entity Framework

Entity Framework je open-source multiplatformní framework využívající se jako vrstva, kterou je možné přistoupit z kódu do databáze a pomocí speciálních funkcí získávat, ale také vkládat a upravovat data v databázi. Díky této vrstvě abstrakce je pro uživatele jednodušší

získávat data z databáze a nemusí provádět složité dotazy. Dalo by se říct, že se používá jako takový mapovač.

Při vytváření databáze jsem využil přístupu Code-First, který spočívá na principu definování tříd a vazeb mezi nimi v kódu viz obrázek 4.10. Po nadefinování se vytvoří migrace, které se aplikují na databázi a vytvoří příslušné tabulky. Tento přístup je zpravidla rychlejší a lépe aplikovatelný při definování tabulek v databázi.



Obrázek 4.10: Obrázek znázorňující princip přístupu Code-First.

4.10 LINQ

Je to integrovaný jazyk, který se využívá v .NET kvůli přístupu k objektům a práci s nimi. Díky tomuto přístupu jsem mohl dobře pracovat s entitami a upravovat jejich obsah. Také ve spojení s Entity Frameworkem bylo jednodušší dotazovat databázi. Při používání LINQ se používají lambda výrazy.

Tento jazyk obsahuje různé metody pro práci s objekty. Při vývoji aplikace jsem využil hlavně tyto metody:

- **Select** - Jedná se o metodu, kterou můžeme vybírat pouze určité atributy objektů. Také tyto objekty můžeme při vybrání dále upravovat.
- **Where** - Pomocí této metody je možné vybírat objekty podle podmínky, kterou nastavíme. Jde o takzvanou filtrovací metodu. Využití můžeme najít například při výběru objektu, jehož atribut má určitou hodnotu.
- **Sum, Max** - Řadí se do skupiny agregačních metod. Pomocí těchto metod můžeme na datovém zdroji provádět operace jako je suma nebo největší hodnota atributu z datového zdroje.
- **First, FirstOrDefault** - Díky těmto metodám můžeme vybrat pouze první prvek datového zdroje. Také je základní metoda **First** obohacena o rozšířenou verzi, která navrácí `null`, ovšem základní varianta by skončila výjimkou.

4.11 Server

Úlohou serveru ve výsledné aplikaci je především autentizace uživatele a také ukládání, ale i posílání dat uživateli zpět do telefonu. V rámci této bakalářské práce jsem přemýšlel nad rozšíření aplikace o webové rozhraní. Z tohoto důvodu byly přidány koncové body, které by mohly sloužit i pro webovou aplikaci, kterou by mohl využívat ošetřující nefrolog. Server jsem implementoval pomocí .NET Core, jak již bylo zmiňováno v kapitole o použitých technologiích.

Pro autentizaci uživatele a přístup ke koncovým bodům se uživatel musí autentizovat a na základě autentizace a vydání tokenu může využívat jemu příslušné koncové body.

Koncové body

Při vytváření REST API jsem naimplementoval zhruba 20 koncových bodů. Jak jsem dříve zmínil, některé jsou ovšem jen experimentální a jsou připraveny na rozšíření například pro přístup z webového rozhraní.

Při implementaci jsem se řídil zkratkou CRUD (*Create, Read, Update, Delete*). Tato zkratka symbolizuje základní operace, které se používají nad uloženými záznamy z databáze. Všechny entity mají koncové body, které tyto operace vykonávají.

Pro chod aplikace jsou v současné době nejdůležitější tyto koncové body:

- **/Login (POST)** - přijímá uživatelskou emailovou adresu a heslo. Na základě těchto údajů ověří jeho identitu a vrátí token, který uživatel použije k autentizaci na dalších koncových bodech.
- **/Register (POST)** - přijímá uživatelskou emailovou adresu a heslo. V těle požadavku jsou specifikovány všechny uživatelské údaje. Registruje uživatele do databáze služby Firebase a také do uživatelské databáze aplikace s jeho informacemi.
- **/api/UserSync/ButtonPost (POST)** - přijímá strukturu, která obsahuje všechny entity (uživatel, intervaly, zkonsumované potraviny a vlastní přidání potravin a nápoje) a uloží je do databáze jako poslední zálohu.
- **/api/UserSync/LoginSync (GET)** - vybere všechny informace, které se s uživatelem pojí v databázi a vrátí je v těle odpovědi zpět do telefonu pro synchronizaci.

Firestore autentizace

Pro autentizaci jsem použil technologii Firestore Authentication¹⁰. Díky této technologii jsem byl schopen autentizovat uživatele a následně jim vydat token, kterým se přihlásí na další koncové body, které tento token vyžadují. Pomocí speciálního API klíče, který jsem dostal při registraci aplikace na Firestore a předem dané adresy se určuje, kam se požadavek zašle. Pokud je požadavek úspěšný, vrátí se token, který se při dalších požadavcích přibalí do těla zprávy. Tato technologie také zprostředkovává registraci a rušení uživatelských účtů.

Asynchronní programování

Při přístupu k REST API nebo celkově k databázi, je vhodné používat asynchronní operace kvůli časově nákladným úkonům. Pokud bychom použili synchronní operace při dotazování vytvořeného REST API, mohl by dotazovat pouze jeden klient a ostatní by museli čekat.

Tuto funkcionalitu v .NET zajišťuje práce s vlákny. Pro inicializaci asynchronní operace se používají operátory `async` a `await`. Pomocí těchto metod jsem byl schopen optimalizovat výkon aplikace i REST API.

¹⁰<https://firebase.google.com/docs/auth>

Kapitola 5

Testování

V této části se podíváme na testování aplikace. Testování je velmi důležitou součástí vývoje aplikací. Díky testování jsme schopní zjistit chyby, které programátor může přehlédnout, a také zjistit slabá místa, která uživatelům nevyhovují. Také je dobré otestovat intuitivnost aplikace a její používání v praktickém životě. Pokud by se aplikace jevila jako příliš složitá a neintuitivní je pravděpodobné, že tyto faktory budou mít špatný vliv na používání aplikace. Při testování jsem rozdělil fáze na krátkodobé a dlouhodobé.

Při testování je také dobré použít automatické testy, které jsem připravil. Těmito testy jsem byl schopen otestovat změny, které jsem v průběhu vývoje provedl na koncových bodech. Díky automatickým testům můžeme například odhalit i nekonzistenci koncových bodů na REST API. Důležitou součástí vytváření automatických testů je fakt, že pokud je test špatně koncipován, může přinést více škody než užitku.

5.1 Uživatelské testování

Při testování uživateli jsem se zaměřil na distribuci aplikace mezi několik uživatelů, kteří aplikaci používali na denní bázi. Toto testování mi pomohlo otestovat, zda je aplikace uživatelsky přívětivá a uživatelé ji rádi používají. V první fázi jsem dával uživatelům jednoduché úkony, které vyplynuly z diagramu případu užití 3.1 a jak dlouho jim tyto úkony zaberou. Na základě zpětné vazby jsem zkoušel problémové sekce zjednodušit, nebo je upravit tak, aby byly jednodušší pro použití.

Aplikaci jsem dal k otestování lidem napříč rozmanitým věkovým spektrem. Ovšem zaměřil jsem se na uživatele starších 51 let, jak vychází z dotazníku 2.5, protože jejich podíl mezi respondenty je více než poloviční.

Krátkodobé testování

V této fázi byly provedeny základní krátkodobé testy, které se zaměřily na běžné užívání aplikace. Uživatelé vycházeli z testovacích scénářů, které jsem pro ně vytvořil. Díky těmto scénářům jsem schopný uživatele otestovat na základních činnostech a vyplnit zpětnou vazbu.

Testovací scénáře:

- Vytvořte uživatelský účet (registrujte se) a proveďte první přihlášení.
- Zadejte svou první potravinu a nápoj.

- Změňte množství zadaných potravin.
- Prozkoumejte seznam potravin a nápojů.
- Přidejte vlastní potravinu a nápoj.
- Změňte jednorázově délku intervalu.
- Nahlédněte do sekce statistik a prozkoumejte je.
- V nastavení změňte dny dialýz.
- V nastavení proveďte změnu uživatelských údajů.
- Použijte tlačítko odeslat data pro odeslání dat na server.
- Proveďte odhlášení a znovu se přihlašte.

Uživatelé začali s registrací a následně se svým prvním přihlášením. Hned poté jsem čerpal zpětnou vazbu, jak na ně aplikace působí na první pohled. Uživatelé pokračovali s nejběžnějším využitím - evidování zkonsumovaných potravin. Po zkonsumování potravin zadají množství a zkontrolují, zda aplikace pracuje správně. Také jsem je požádal, aby občas změnil zadané množství a dali mi vědět, zda se změny projeví. Také byla otestována intuitivnost přidání vlastních potravin a nápojů. Jedním z neposledních úkonů bylo změnit si uživatelské nastavení. Tato akce neprobíhá příliš často, ale je nutné ověřit funkcionalitu a zároveň jednoduchost provedení této akce.

Uživatelé vždy měli krátkou chvíli na seznámení se s aplikací. Ze zpětné vazby jsem se snažil evidovat průměrné časy úkonů, které prováděli. Na základě průměrů těchto časů jsem byl schopen vyhodnotit, zda tyto úkony netrvaly zbytečně dlouho a jestli se náhodou nedají zjednodušit.

Dlouhodobé testování

Po provedení základních testů jsem uživatele nechal testovat aplikaci zhruba tři týdny. Díky dlouhodobému testování vyplynuly chyby, které jsem při krátkodobém testování nebyl schopen odhalit. Také jsem trochu více pronikl do potřeb uživatelů a jejich nápady mi pomohly dále rozvíjet aplikaci. Po skončení doby jsem jim aplikaci nechal nainstalovanou v telefonu a většina těchto testovacích uživatelů ji dále používá. Dále jsem s nimi také provedl konzultaci, kde jsem čerpal jejich zkušenosti a probírali jsme další možný směr aplikace a návrhy.

Během této doby jsem dostal mnoho nápadů na rozvoj aplikace. Mimo jiné jsem si vyzkoušel i komunikaci s uživateli a snažil se přizpůsobovat jejich potřebám.

5.2 Automatické testy

Aby byla zajištěna konzistence koncových bodů na REST API, implementoval jsem automatické testy, které při spuštění kontrolují koncové body. Tyto testy fungují na principu provolání metody obsluhující koncový bod a přímého přístupu do databáze. Vyhodnocení se provede porovnáním výsledků, které tyto dva úkony vrátí.

Při automatickém testování přistoupíme k obslužné metodě, která vrátí objekt nebo seznam objektů. V následující části přistoupíme do databáze a dotážeme se se stejnými parametry a očekáváme navrácení stejného objektu nebo listu objektů.

Při úpravách koncových bodů jsem vždy spustil testy, ze kterých jsem mohl vyhodnotit konzistenci jak koncových bodů, tak databáze. Díky tomuto přístupu jsem byl schopen zamezit chybám, které by jinak mohly velmi znatelně omezit fungování aplikace.

5.3 Nalezené chyby

V této sekci se podíváme na nalezené chyby, které z uživatelského testování vyplynuly. Díky jejich intenzivnímu testování jsem byl schopný chyby opravit a dokonce občas přepracovat logiku vzniklého problému.

Registrace bez připojení k internetu

Jeden z testerů zaznamenal situaci, že se snažil registrovat bez připojení k internetu, což jak je již zmíněno v návrhu by nemělo být proveditelné. Ovšem tato situace nebyla ošetřena a bylo nutné aplikaci vypnout a zapnout, protože se stále točila animace při registraci. Tento problém byl ošetřen za pomoci funkce z rozhraní `Xamarin.Essentials` a konkrétně jednoduché podmínky, která kontroluje zda je zařízení připojeno k internetu a pokud ne, je zobrazeno dialogové okno.

Tuto chybu jsem stejným způsobem také opravil u použití tlačítka k odeslání dat. Bylo zvláštní, že chybu nezaregistroval ani jeden z testerů. Na základě zpětné vazby jsem se dozvěděl, že tuto funkcionalitu používají vyjíměčně a není pro ně až tak důležitá.

Přehled statistik

Dalším problémem, na který jsem byl upozorněn jedním z testerů, se stal při prvním přihlášení. Statistiky jsou uloženy v databázi a dotaz do databáze byl stavěn tak, že hledal statistiky z minulých období. Ovšem při prvním přihlášení statistiky pravděpodobně ještě neexistují. Problém jsem vyřešil pomocí podmínky, která kontrolovala, zda uživatel již nějaké statistiky má. Pokud nejsou v databázi nalezeny žádné záznamy, zobrazí se uživateli přehled aktuálního období.

Přidání vlastních potravin

V sekci potravin je možnost vytvoření vlastní potraviny a její uložení do databáze. Tuto položku pak může uživatel vybírat ze svého seznamu potravin. Chyba byla v tom, že při ukládání do databáze telefonu jsem zapomněl vložit všechny atributy a při výběru potraviny ke zkonzumování se aplikace chovala nekonzistentně a ukazovala špatné hodnoty.

5.4 Nové nápady na vylepšení aplikace

Velmi důležitou součástí vývoje aplikací, ať už webových či mobilních, je zpětná vazba uživatelů. Vývojáři a studia, kteří naslouchají komunitním požadavkům a nápadům, bývají obvykle velmi oblíbení mezi uživateli a také tato vlastnost může vést ke zkvalitnění aplikace. Kvůli faktu, že na aplikaci mám zájem dále pokračovat a stále ji vyvíjet i po ukončení této práce, je nedílnou součástí naslouchat uživatelům. Chtěl bych, aby se aplikace zapsala v dobrém světle do povědomí komunity pacientů s onemocněním ledvin, nebo také lidem, kteří by ocenili vyhodnocování příjmu těchto specifických hodnot z potravin a nápojů.

Vytvořené webové aplikace

Někteří uživatelé vyjádřili zájem o vývoj webové aplikace, kvůli snazšímu přístupu k statistikám a kvůli tomu, že by jejich ošetřující lékař mohl také snadněji nahlédnout do hodnot, které uživatelé naměřili.

Díky tomu, že mám zájem celkové řešení dále rozšiřovat, jsem přesvědčen, že tato součást je realizovatelná a mohla by celkové řešení velmi obohatit. Proto zůstanu v kontaktu s uživateli, kteří tento názor vyjádřili.

Jazyková podpora

S tímto nápadem přišel můj konzultant. Nad touto možností jsem také přemýšlel, ovšem až v pokročilé části práce a proto jej v rámci práce neřeším. Ovšem tento nápad se mi velmi zamlouvá a mám v plánu aplikaci rozšířit o multijazyčnost.

Kapitola 6

Závěr

Cílem práce bylo vytvořit mobilní aplikaci na operační systém Android, která kontroluje dietu pacientů na dialýze a pomáhá jim s dodržováním dietních opatření. Aplikace by měla být schopná ukládat data lokálně, tak i na serveru. Tudíž bylo potřeba vytvořit serverovou část kvůli propojení aplikace s databází. V aplikaci se nachází databáze potravin, kterou uživatelé mohou libovolně rozšiřovat. Práce se zaměřuje na seznámení se s požadavky diety pacientů na dialýze a jejich specifickými potřebami. Vývoj aplikace a důležité informace jsem konzultoval s odborníkem, kterého mi dělal externí konzultant doporučený vedoucím práce. Bylo také nutné vybrat vhodné hodnoty pro sledování.

Díky řešení této práce jsem načerpal cenné zkušenosti z oblasti onemocnění ledvin. Pronikl jsem do dané problematiky a také získal znalosti z lékařské terminologie. Také jsem se seznámil s procesem dialýzy, o kterém jsem před řešením práce věděl velmi málo. Z hlediska technické odbornosti jsem získal znalosti s vývojem uživatelského rozhraní a celkového konceptu vývoje aplikací počínaje návrhem, vytváření diagramů případů užití, ale také i testování a vyhodnocování uživatelských dotazníků. Dále jsem si vyzkoušel vývoj mobilních aplikací v prostředí Xamarin. Prohloubil jsem zkušenosti s rozhraním API, se kterým jsem měl pouze základní zkušenosti. S touto problematikou se úzce váže vytváření webových požadavků, autorizace a komunikace klient-server, kterou jsem implementoval. V neposlední řadě jsem komunikoval s uživateli a řešil jejich návrhy a požadavky.

V dalším kroku vývoje, který se bude odehrávat po dokončení práce, bych chtěl aplikaci ještě zdokonalit o novou funkcionalitu. Hlavním motivem je přidání poznatků od uživatelů. Tudíž bych chtěl implementovat webové rozhraní a také rozšířit aplikaci na operační systém iOS. Aplikace je v současné době distribuována mezi testery a je jimi aktivně využívána. Na základě dlouhodobého testování a referencí zvažuji aplikaci zpřístupnit veřejně na Google Play.

Literatura

- [1] AMBLER, S. W. UML Use-Case Diagrams. In: *The Elements of UML™ 2.0 Style*. Cambridge University Press, 2005, s. 33–46. DOI: 10.1017/CBO9780511817533.005.
- [2] APPBRAIN. *Top Android OS versions* [online]. (Den návštěvy: 1.3.2021), 2021. Dostupné z: <https://www.appbrain.com/stats/top-android-sdk-versions>.
- [3] APPTECHBLOGGER. *Comparative Analysis Of The Merits Of Xamarin.Forms And Xamarin Native* [online]. 2017. (Den návštěvy: 25.3.2021). Dostupné z: <https://medium.com/@techblogger/comparative-analysis-of-the-merits-of-xamrin-forms-and-xamarin-native-9400ea934e1c>.
- [4] BECKWITH, S. Last acts task forces, resource committees focus on changing care at the end of life. *Journal of palliative medicine*. Mary Ann Liebert, Inc. 2 Madison Avenue Larchmont, NY 10538 USA. 1999, sv. 2, č. 4, s. 377–381.
- [5] BEDNÁROVÁ V, HRUSKOVÁ Z, MOTÁN V, NEPRASOVÁ M.. Peritoneální dialýza a její modifikace v léčbě chronického selhání ledvin. *Vnitřní Lékařství*. 2011, sv. 57, 7 a 8, s. 635–639.
- [6] CHUMLEA, W. C., GUO, S. S., ZELLER, C. M., REO, N. V. a SIERVOGEL, R. M. Total body water data for white adults 18 to 64 years of age: The Fels Longitudinal Study. *Kidney International*. 1999, sv. 56, č. 1, s. 244–252. ISSN 0085-2538. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0085253815462829>.
- [7] DIALYZA.CZ. *Fosfor u dialyzovaných* [online]. (Den návštěvy: 25.12.2020), 2019. Dostupné z: <https://www.dialyza.cz/cs/porozumet/zivot-s-dialyzou/fosfor-u-dialyzovanych/>.
- [8] DIALYZA.CZ. *Živiny a jejich význam* [online]. (Den návštěvy: 2.1.2020), 2019. Dostupné z: <https://www.dialyza.cz/cs/porozumet/zivot-s-dialyzou/vyziva/ziviny-a-jejich-vyznam/>.
- [9] DRAŠAR, P. a FUSEK, M. Recenze: Koolman J., Röhms K.-H.: Barevný atlas biochemie. *Chemické listy*. 2012, sv. 106, č. 8.
- [10] FLASHHEALTH. *Pitný režim* [online]. (Den návštěvy: 25.12.2020), 2020. Dostupné z: <https://www.dialyza.cz/cs/vyziva/stravovaci-a-pitny-rezim/pitny-rezim/>.
- [11] FLOEGE, J. Phosphate binders in chronic kidney disease: a systematic review of recent data. *Journal of nephrology*. Springer. 2016, sv. 29, č. 3, s. 329–340.

- [12] GBD CHRONIC KIDNEY DISEASE COLLABORATION. Global, regional, and national burden of chronic kidney disease, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017. *The Lancet*. 1. vyd. 2020, sv. 395, č. 10225, s. 709–733.
- [13] GRASSMANN, A., GIOBERGE, S., MOELLER, S. a BROWN, G. ESRD patients in 2004: global overview of patient numbers, treatment modalities and associated trends. *Nephrology dialysis transplantation*. Oxford University Press. 2005, sv. 20, č. 12, s. 2587–2593.
- [14] HEALTH, N. I. o. D. National Institutes of, DIGESTIVE a DISEASES, K. *United States Renal Data System. 2019 USRDS annual data report: Epidemiology of kidney disease in the United States*. [online]. (Den návštěvy: 20.12.2020), 2019. [ADR Executive Summary, p. 46]. Dostupné z: <https://usrds.org/media/2371/2019-executive-summary.pdf>.
- [15] HOŘAVA, M. a ČERVENKOVÁ, A. *Uživatelsky přívětivá rozhraní*. 1. vyd. HORAVA & ASSOCIATES, 2009. ISBN 978-802-5452-950.
- [16] HRUBÝ, M. a MENGEROVÁ, O. *Výživa při pravidelném dialyzačním léčení*. Forsapi, 2009. ISBN 978-80-87250-06-8.
- [17] JAMES NEWTON-KING. *Cílení na více platforem* [online]. 2021. (Den návštěvy: 28.4.2021). Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/standard/library-guidance/cross-platform-targeting>.
- [18] JOHNSON, J. *What is Xamarin?* [online]. (Den návštěvy: 30.12.2020), květen 2020. Dostupné z: <https://docs.microsoft.com/cs-cz/xamarin/get-started/what-is-xamarin>.
- [19] KABI, F. *Dialyzátor* [online]. (Den návštěvy: 25.12.2020), 2015. Dostupné z: <https://www.nizkobilkovinnadieta.cz/dialyzator.html>.
- [20] KALLENBACH, J. *Review of Hemodialysis for Nurses and Dialysis Personnel 9th Edition*. Elsevier health sciences, 2015. ISBN 9780323310192.
- [21] KALORICKÉ TABULKY. *Kalorické tabulky* [online]. (Den návštěvy: 15.10.2020). Dostupné z: <https://www.kaloricketabulky.cz/potraviny/ryze-bila-dlouhozrnnna-varena>.
- [22] LEE a KYUNG YIL. A unified pathogenesis for kidney diseases, including genetic diseases and cancers, by the protein-homeostasis-system hypothesis. *Kidney research and clinical practice*. Korean Society of Nephrology. 2017, sv. 36, č. 2, s. 132.
- [23] MACLEAN, D. *Pro Android 5*. 1. vyd. Springer-Verlag Berlin and Heidelberg GmbH & Co. KG, 2014. ISBN 9781430246800.
- [24] MS EAVAN FLYNN. Commencing Peritoneal Dialysis. *Children's Health Ireland* [online]. 2018. Dostupné z: <https://www.olchc.ie/Healthcare-Professionals/Nursing-Practice-Guidelines/Peritoneal-Dialysis-Commencement-Guideline-2018.pdf>.
- [25] NIELSEN, J. *Designing Web usability*. 1. vyd. New Riders, 1999. ISBN 156205810X.

- [26] O'REILLY MEDIA. *Chapter 1. What Is React Native?* [online]. (Den návštěvy: 5.1.2021). Dostupné z: <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>.
- [27] POKOROVÁ, P. *Výživa dialyzovaných pacientů*. Forsapi, 2013. ISBN 9788087250235.
- [28] SA MATHIESON. *Kidney dialysis – from the comfort of your sitting room* [online]. 2016. (Den návštěvy: 20.2.2021). Dostupné z: <https://www.theguardian.com/healthcare-network/2016/feb/02/kidney-dialysis-hospital-machine-home>.
- [29] SAMUELS, E. *The Public Domain in Copyright Law, 1993, 41 Journal of the Copyright Society* [online]. (Den návštěvy: 8.1.2021), 1993. Dostupné z: <http://www.edwardsamuels.com/copyright/beyond/articles/public.html#fn109>.
- [30] SKUPINA B. BRAUN. *Vše o dialýze* [online]. (Den návštěvy: 23.3.2021). Dostupné z: <https://www.bbraun.cz/cs/spolecnost/b-braun-avitum/vse-o-dialyze.html#peritoneln-dialza>.
- [31] SNEATH, T., MULLER, H. a PIERRE LOUIS. *Flutter architectural overview* [online]. (Den návštěvy: 5.1.2021), říjen 2020. Dostupné z: <https://flutter.dev/docs/resources/architectural-overview>.
- [32] STEIN, S. *Co je Azure SQL Database?* [online]. (Den návštěvy: 9.1.2021), září 2020. Dostupné z: <https://docs.microsoft.com/cs-cz/azure/azure-sql/database/sql-database-paas-overview>.
- [33] TESAŘ, V., SCHÜCK, O. a TEPLAN, V. *Klinická nefrologie*. Grada, 2006. ISBN 9788024705033.
- [34] TOMANDL, J., DOSTÁL, J., PAULOVÁ, H. a TÁBORSKÁ, E. *Základy lékařské chemie a biochemie*. Masarykova univerzita, 2020.
- [35] WIBIER, D. *Webinar: Using DevExpress MVVM Magic with WPF* [online]. 2016. (Den návštěvy: 15.3.2021). Dostupné z: <https://community.devexpress.com/blogs/donw/archive/2016/08/25/webinar-using-devexpress-mvvm-magic-with-wpf.aspx>.
- [36] ZENDULKA, J. a BARTÍK, V. *Úvod, základní pojmy databázových technologií* [online]. (Den návštěvy: 18.2.2020), 2020. Dostupné z: https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIDS-IT%2Flectures%2Fcz%2F1_uvod.pdf.