



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 VEDENÍ ATLETICKÝCH TRÉNINKŮ

MOBILE SPORT ADVISOR FOR ATHLETICS TRAINING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ SCHMELCER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN HRUBÝ, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Schmelcer Lukáš**
Program: Informační technologie
Název: **Mobilní aplikace pro vedení atletických tréninků**
Mobile Sport Advisor for Athletics Training
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte programování pro iOS a knihovnu HealthKit. Prostudujte literaturu o vedení atletického výcviku se zaměřením na běh. Zvolte si metodiku vedení tréninku.
2. Navrhněte aplikaci implementující tréninkovou metodiku. Jako zdroj dat předpokládejte systémovou knihovnu HealthKit.
3. Implementujte aplikaci optimalizovanou pro iPhone. Zaměřte se na spolupráci s Apple Watch pro sběr dat.
4. Aplikaci testujte v horizontu delšího provozu. Ukažte naměřená data.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání.

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

Vedoucí práce: **Hrubý Martin, Ing., Ph.D.**

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

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Cielom tejto bakalárskej práce je navrhnúť, implementovať a otestovať tréningovú mobilnú aplikáciu pre platformu iOS. Aplikácia by užívateľovi mala umožniť rýchlejšie splnenie bežeckých cieľov, pričom by malo byť minimalizované riziko zranenia. Aplikácia bola navrhnutá tak, aby spĺňala požiadavky cieľovej skupiny. Vďaka častej spätnej väzbe užívateľov výsledná aplikácia disponuje zrozumiteľným prostredím v ktorom má užívateľ prehľad o svojom bežekom tréningu.

Abstract

This bachelor thesis aims to design, implement and test a mobile training application for the iOS platform. The application should allow the user to meet running goals faster, while the risk of injury should be minimized. The application was designed to meet the requirements of the target group. Due to user frequent feedback, the final application has a very easy to understand user interface, in which the user has an overview of the planned running training.

Klíčové slová

bežecký tréning, HealthKit, SwiftUI, mobilná aplikácia, iOS, CoreData, športová aplikácia

Keywords

running training, HealthKit, SwiftUI, mobile application, iOS, CoreData, sports application

Citácia

SCHMELCER, Lukáš. *Mobilní aplikace pro vedení atletických tréninků*. Brno, 2021. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Hrubý, Ph.D.

Mobilní aplikace pro vedení atletických tréninků

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Martina Hrubého Ph.D. .Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Lukáš Schmelcer
10. mája 2021

Podakovanie

Chcem sa poďakovať pánovi profesorovi Hrubému za jeho odborné rady a pozitívny prístup počas vedenia tejto práce a za jeho čas, ochotu a správne nasmerovanie pri riešení problému. Rovnako by som sa chcel poďakovať všetkým jednotlivcom ktorí mi pomohli s testovaním výslednej aplikácie a rodine, za podporu a dodávanie energie počas celého štúdia.

Obsah

1	Úvod	2
2	Teória vedenia bežeckého výcviku a vymedzenie cieľovej skupiny	3
2.1	Spôsoby odhadu športovej výkonnosti	3
2.2	Spôsoby štruktúrovania bežeckého tréningu	5
2.3	Tréningové intenzity	7
2.4	Prevenca zranenia	8
3	Východiskový stav technológií	11
3.1	Technológie tvorby iOS aplikácií	11
3.2	Návrhový vzor Model–View–Viewmodel	12
3.3	CoreData	12
3.4	CloudKit	14
3.5	HealthKit	15
3.6	MapKit	16
3.7	Analýza existujúcich riešení	17
4	Návrh aplikácie	20
4.1	Vymedzenie cieľovej skupiny	20
4.2	Prípad použitia	20
4.3	Užívateľské rozhranie	22
4.4	Systém spätnej väzby	25
4.5	Návrh databázy	26
5	Implementácia a testovanie	30
5.1	Použité technológie a postupy	30
5.2	Výsledné užívateľské prostredie	31
5.3	Spôsob implementácie	32
5.4	Testovanie	35
6	Záver	40
	Literatúra	41
A	Obsah priloženého média	43

Kapitola 1

Úvod

Od roku 2001 do roku 2018 sa počet ľudí ktorí sa rozhodli postaviť na štartovú čiaru svojho vytúženého bežeckého závodu znásobil takmer 4 krát¹. Náruživí bežci sa pridali do lokálnych bežeckých klubov, niektorí sa rozhodli pre súkromnú formu tréningu s osobným trénerom, avšak zvyšná väčšina ľudí nevie, ako sa správne pripraviť na ich nadchádzajúci veľký deň.

Bežci ktorí sa rozhodli trénovať sami, sa veľmi často na štartovú čiaru ani nepostavia. A to nie z dôvodu nedostatku motivácie, veď športových aplikácií s cieľom motivovať a zdieľať výsledky s komunitou je veľké množstvo. Tieto športové aplikácie sa v dnešnej dobe zameriavajú hlavne na sociálny aspekt športu – stretnúť nových ľudí, porovnať si výsledky, vytvárať výzvy. Chýba v nich vzdelávanie o športe, nevysvetlia ako správne dávkovať tréningovú záťaž a ani neberú do úvahy individuálne potreby športovcov. Preto veľké množstvo bežcov počas tréningu vyradí zranenie², odradí ich buď potrebná dĺžka a náročnosť tréningu, alebo veľké množstvo informácií ktoré by si mali naštudovať. Aké cviky by mali cvičiť, ako by si mali správne preťahovať svalstvo, ako dávkovať nabehnutú kilometráž. Preto je cieľom tejto aplikácie poskytnúť formu bežeckého trénera, ktorý užívateľa dovedie k jeho športovému cieľu bez zranení a plne pripraveného. Kapitola 2 uvedie do problematiky vedenia bežeckého tréningu. Kapitola 3 obsahuje informácie o teórií tvorby aplikácií pre zariadenia s operačným systémom iOS a preskúma momentálne existujúce riešenia. Kapitola 4 sa zameriava na návrh samotnej aplikácie. Opisuje vymedzenú cieľovú skupinu, rozoberá návrh užívateľského rozhrania, databázy a systém spätnej väzby ktorý slúži na určenie ďalšieho postupu pri tréningu. Kapitola 5 popisuje implementáciu aplikácie, popisuje zvolené postupy a technológie. V ďalšej časti obsahuje informácie k testovaniu aplikácie a rozoberá výsledky testovania.

¹<https://runrepeat.com/state-of-running>

²<https://runnersconnect.net/why-runners-get-hurt/>

Kapitola 2

Teória vedenia bežeckého výcviku a vymedzenie cieľovej skupiny

Dôležitým aspektom tejto práce je mať podrobný prehľad nad rôznymi metódami vedenia bežeckého tréningu. Musí sa zväžiť efektívnosť jednotlivých spôsobov stimulácie indikátorov výkonnosti, pričom je vhodné zamerať sa hlavne na metódy ktoré umožňujú dlhodobú udržateľnosť tréningovej záťaže. Rovnako dôležitým aspektom je aj spôsob odhadu zdravotného stavu a aktuálnej športovej výkonnosti užívateľa.

2.1 Spôsoby odhadu športovej výkonnosti

V dnešnej dobe existuje rozmanité množstvo indikátorov ktoré sú zamerané na odhad miery fyzickej zdatnosti[19]. Testy založené na metabolickom ekvivalente ktoré majú za úlohu odhadnúť príjem kyslíka počas určenej intenzity cvičenia pri nešportovcoch[20], alebo takzvané dvanásť minútové testy počas ktorých človek musí ubehnúť čo najväčšiu vzdialenosť. Avšak je nutné sa zamerať na osvedčené a uznávané metódy, podľa ktorých sa dá odhadnúť fyzická zdatnosť a kondícia.

Srdcový tep

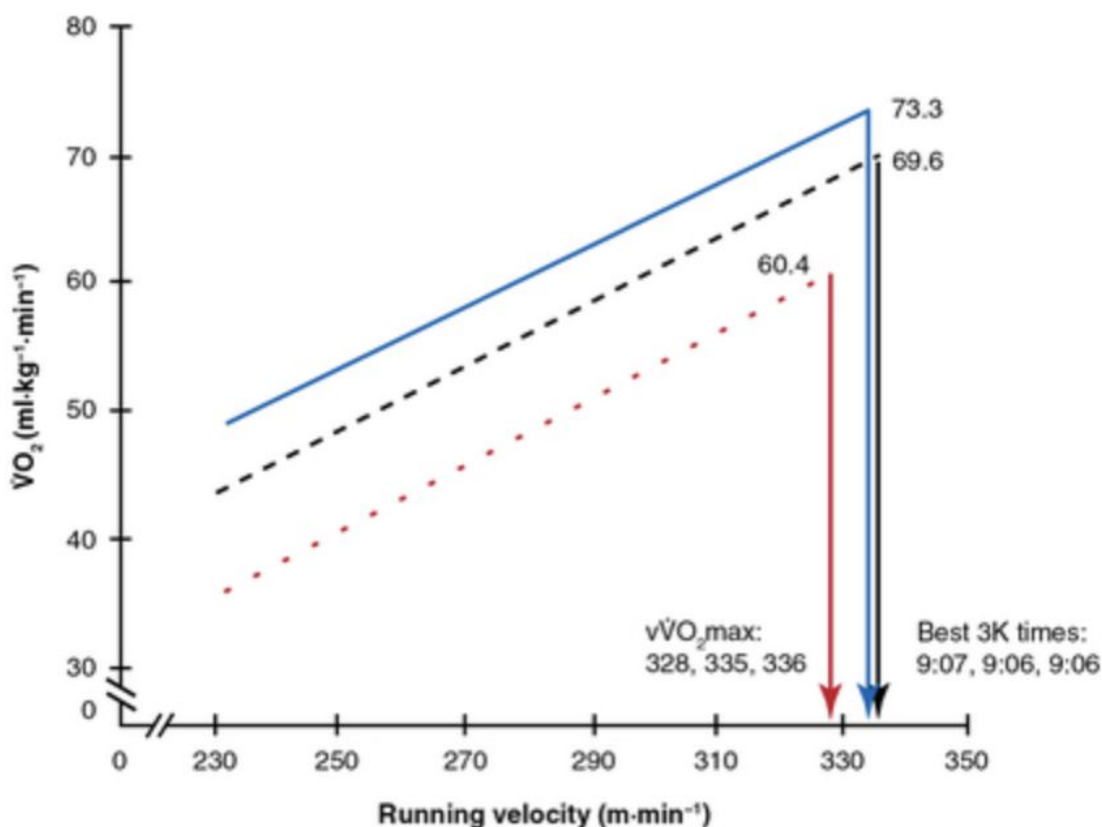
Výborným indikátorom aeróbnej výkonnosti človeka je srdcový tep a to špecificky kľudový srdcový tep. Ten sa najlepšie meria ihneď po zobudení, ešte pred tým ako človek vstane z postele. Pre väčšinu ľudí je v normálnom rozpätí tep od 60 do 100 úderov za minútu. Kľudový tep môže byť ovplyvnený najrôznejšími faktormi ako napríklad stres, hormóny, alebo to ako je človek fyzicky zdatný [2]. Mnoho profesionálnych atlétov máva kľudový tep v rozmedzí od 40 úderov za minútu a viac. Kľudový srdcový tep sa tiež znižuje pri pravidelnom vykonávaní fyzickej aktivity, znížení krvného tlaku a znížení váhy¹. Platí, že čím je kľudový srdcový tep nižší, tým by mal byť človek fyzicky zdatnejší. Srdcový tep však nie je vhodným indikátorom výkonnosti počas tréningov. Človek síce môže mať cieľový srdcový tep, avšak ten nereflektuje momentálnu výkonnosť športovca. Srdcový tep sa mení prudko už pri malej zmene teploty vzduchu, ktorá spôsobuje zvýšenie teploty tela, kde zmena môže dosahovať až desať úderov za minútu na jeden stupeň Celzia [7], pri zmene vlhkosti, alebo pri zmene nadmorskej výšky. Brať srdcový tep ako vhodný indikátor výkonnosti je možné vtedy, ak by sme sa zamerali na zotavovací srdcový tep. Ten je možné pozorovať ihneď

¹<http://heart.bmj.com/content/99/12/882.full?sid=90e3623c-1250-4b94-928c-0a8f95c5b36b>

po ukončení fyzickej aktivity. Srdcový tep by mal klesať rýchlejšie bezprostredne po ukončení aktivity práve vtedy, keď sa človek stáva viac fyzicky zdatným a výkonnejším. Bežný pokles srdcovej frekvencie je v rozmedzí 25 až 30 úderov za minútu. Čím vyšší a rýchlejší je pokles, tým je človek výkonnejší. Toto pozorovanie je vhodné vykonať dve až tri minúty bezprostredne po ukončení cvičenia.[9]

VO2Max

Značí maximálnu rýchlosť spotreby kyslíka a je jeden z najlepších ukazateľov aeróbnej výkonnosti. Udáva sa v mililitroch spotrebovaného kyslíka za minútu na jeden kilogram ($\text{ml}\cdot\text{kg}^{-1}\cdot\text{min}^{-1}$). Testuje sa v laboratóriách, avšak väčšia hodnota $\dot{V}O_{2max}$ nutne neznamená lepší maximálny výkon. Túto skutočnosť je možné vidieť na obrázku 2.1, ktorý porovnáva tri bežkyne s rôznymi aeróbnymi profilmi, pričom dosahujú veľmi podobné najlepšie výsledky pri behu na vzdialenosť 3000 metrov.



Obr. 2.1: Obrázok zobrazujúci to, že rozdiel v aeróbnom profile nutne neznamená rozdielny maximálny výkon. Obrázok je prebratý z publikácie [6]

Laktátový prah

Reprezentuje maximálnu hodnotu kyseliny mliečnej v krvi, pri ktorej ju telo vie odstraňovať rýchlejšie ako tvoriť. Teda reprezentuje maximálnu intenzitu cvičenia, ktorú by mal bežec vydržať približne hodinu. S vyššou koncentráciou kyseliny mliečnej v krvi klesá výkonnosť bežca. Uvádza pocit tuhnutia svalstva.

VDOT systém

Odhad správneho tempa bude vykonaný priradením hodnôt veličiny nazývanej VDOT [6]. Toto označenie zaviedol Jack Daniels a Jimmy Gilbert, keď skúmali jednotlivé výsledky z testovania, a $\dot{V}O_2max$ hodnoty elitných bežcov so zameraním na stredne dlhé a dlhé trate. Slúži na elimináciu nepresnosti hodnoty $\dot{V}O_2max$ nakoľko ako už bolo vyššie popísané, bežci ho môžu mať rôzne, ale bežci mohli dosiahnuť rovnakých výkonov — zaradili ich k rovnakému aeróbnemu profilu. Hodnoty tohto profilu znázorňuje hodnota VDOT, alebo "pseudo $\dot{V}O_2max$ ". Z dôvodu limitácie laboratórneho vybavenia u bežných bežcov sa ako maximálny výkon berie nedávny výsledok závodu, alebo vrcholového behu. Na základe týchto hodnôt sa dá odhadnúť predpokladaný čas, ktorý by mal bežec dosiahnuť na inej vzdialenosti. Musí sa však brať do úvahy fakt, že tento odhad platí práve vtedy, ak sa aplikuje tréning na danú vzdialenosť. Napríklad ak najlepší výkon bude zaznamenaný z behu na päť kilometrov, pričom bežec chce zabehnúť maratón. V tomto prípade sa nasledujúci tréning musí zamerať na maratónsku vzdialenosť.

2.2 Spôsoby štruktúrovania bežekého tréningu

Jack Daniels

Jack Daniels prvý krát predstavil svoj tréningový plán v roku 1998. Niekoľko krát bol menovaný za najlepšieho bežekého trénera². Uviedol koncept špecifikácie tréningového tempa na základe fyzickej zdatnosti, pričom fyzickú zdatnosť meria na základe výsledkov z bežekých závodov. VDOT je systém na odhad fyzickej zdatnosti. Na základe tejto hodnoty je možné dopočítať doporučenú rýchlosť pre každý druh bežekého tréningu a to doplnením cieľovej vzdialenosti a cieľového času. Vo svojich tréningových plánoch špecifikuje päť rôznych druhov bežekých tréningov. Tieto sú podrobnejšie rozobrané v podkapitole 2.3 a systém VDOT je popísaný v podkapitole 2.1

Furman Institute of Running and Scientific Training

Prezentuje evolúciu bežekého plánu podľa Jacka Danielsa, kde prezentuje filozofiu "behaj menej, behaj rýchlejšie". Je jedinečný v tom, že plán je štruktúrovaný tak, aby v jednom týždni bežec praktizoval bežeký tréning iba počas troch dní a dva dni využil na posilňovanie, alebo na cyklistiku či plávanie. Medzi doporučený tréning patrí dlhý beh – mal by tvoriť aspoň 45 percent týždennej vzdialenosti. Tréningy zamerané na rozvoj tempa a intervalový tréning.

Jeff Galloway

Tréningový plán založený na pridaní páuz do bežekých tréningov a to vo forme chôdze. Týmto spôsobom je možné navýšiť celkovú vzdialenosť tréningov. Dlhé behy by mali byť vykonané v pomalom tempe. Zameriava sa na ľudí ktorí nie sú bežeky skúsení, alebo sa vracajú po zranení.

²<http://www.rrm.com/archive05/032905n1.htm>

Hansonova maratónová metóda

Metóda ktorá limituje dĺžku dlhých behov na 16 míľ, čo reprezentuje vzdialenosť približne 26 kilometrov. V prieskume však bolo uvedené, že veľké množstvo bežcov ktorí trénovali podľa tohto bežeckého plánu prevýšili maximálnu doporučenú vzdialenosť[24].

Pfitzinger

Tréningový plán pre skúsených bežcov, špecificky pre bežcov ktorí už odbehli maratón a ide im o zlepšenie celkového času. Prezentuje veľké množstvo dlhých a stredne dlhých behov, s občasným zameraním na rozvoj rýchlosti. Bežcom poskytuje kratšiu dobu odpočinku, čím sa zvýši riziko zranenia. V závislosti na týždennom množstve odbehnutých kilometrov odporúča od štyroch až po päť behov za týždeň. Pre viac skúsených bežcov až po sedem a viac behov za týždeň, kedy privádza koncept dvojitého bežeckého tréningu v jednom dni – ráno a večer. V takomto prípade je však dôležité si dávať pozor na pretrénovanie³.

Porovnanie plánov

Z dôvodu veľkého množstva plánov a faktorov je nutné porovnať jednotlivé plány a to ako vyhovujú rôznym úrovňam bežcov. V tabuľke 2.1 je vhodnosť tréningového plánu označená od nuly po číslo päť, kde nula reprezentuje najhoršiu možnosť, a päť najlepšiu. V tabuľke sú uvedené úrovne:

- **začiatočník** reprezentuje bežca, ktorý neubehne viac ako 3 kilometre bez zastavenia, alebo behá menej ako 2 mesiace
- **nováčik** bežec, ktorý behá medzi 2 až 12 mesiacmi, alebo pravidelne behá vzdialenosti aspoň 10 kilometrov
- **pokročilý** reprezentuje bežca, ktorý behá viac ako rok, behá dlhšie behy v približnej vzdialenosti 16 kilometrov, alebo dosahuje zľahka nadpriemerné výsledky
- **semi-pro** reprezentuje bežca, ktorý je takmer na profesionálnej úrovni, pravidelne sa umiestňuje medzi najlepšími, avšak ešte stále nedosiahol na čas, ktorý je potrebný na to, aby mohol byť považovaný za elitného bežca
- **elitný** reprezentuje kategóriu bežcov, pre ktorých je beh živobytie, alebo dosahuje časy ktoré sú na národnej úrovni

	FIRST	Jack Daniels	Pfitzinger	Hanson	Galloway
začiatočník	0	1	0	0	5
nováčik	1	3	0	1	3
pokročilý	2	3	3	2	2
semi-pro	4	3	3	3	2
elitný	3	4	3	2	1

Tabuľka 2.1: Modifikovaná tabuľka porovnávajúca jednotlivé tréningové plány – pôvodná tabuľka pochádza z článku J. Savage [23]

³moment keď človek príliš zaťažie telo a prekročí jeho schopnosť zotaviť sa zo stresovej aktivity

2.3 Tréningové intenzity

K správnej stimulácii jednotlivých indikátorov výkonnosti ktoré boli popísané v kapitole 2, dochádza pri rôznej záťaži a pri rôznej dĺžke záťaže na organizmus jednotlivých bežcov. Informácie k tréningovým intenzitám pochádzajú z publikácie Jacka Danielsa [6].

Lahká intenzita

Typicky je to tréning zodpovedajúci intenzite približne 59 až 74 percent $\dot{V}O_2max$, alebo približne 65 až 78 percent maximálnej hodnoty srdcovej frekvencie. Bežecké tréningy pri tejto intenzite majú veľký význam hlavne z pohľadu prevencie zranení. Sú vhodné na budovanie základnej kilometráže bežca, alebo pri návrate z dlhšej pauzy a to napríklad po zranení. Tento typ tréningu zohráva veľkú úlohu pri tréningu srdcového svalstva, nakoľko maximálna sila pri jednom údere srdca dosahuje maximum, ak srdcová frekvencia dosahuje približne 60 percent maxima srdcovej frekvencie daného užívateľa. Jedným z ďalších pozitívnych účinkov bežeckého tréningu pri tejto intenzite je otvorenie menších krvných ciev, ktoré sú využívané na privádzanie výživy do fungujúceho svalstva. Ďalšie vhodné využitie ľahkej intenzity prichádza v podobe menšej náročnosti, ktorá umožňuje zvládnuť beh na dlhšiu trať a tak ešte viac stimulovať svalové tkanivá až do takej miery, aby prijímali viac kyslíka a menili ho na energiu. Optimálna dĺžka behu strávená pri tejto intenzite je aspoň 30 minút a v tréningovom režime by nemala presiahnuť 150 minút. Ak užívateľ počas tréningu kombinuje beh s chôdzou, je nutné túto dobu jemne predĺžiť.

Maratónová intenzita

Úlohou bežeckých tréningov pri takzvanej maratónovej intenzite je pomôcť telu zvyknúť na záťaž potrebnú pri behaní maratónu. V prípade ak sa bežec neplánuje zúčastniť závodu s maratónskou dĺžkou, majú tieto tréningy skôr psychické účinky. Bežcovi pridávajú mieru dôvery vo svoje výkony a rovnako zapôsobia ako oživenie tréningového plánu. Ak sa bežec nezotavuje zo zranenia, alebo nepocituje veľkú únavu, je vhodné aby svoje dlhé behy behal pri tejto intenzite. A to hlavne preto, lebo sa približuje intenzite pri ktorej by mal bežať závody nad vzdialenosť 15 kilometrov. Počas týchto tréningov je vhodné telo naučiť prijímať tekutiny, poprípade živiny vo forme rôznych gélov tak, aby bežcov počas ich cieľového výkonu nezastihla žalúdočná nevoľnosť. Rýchlosť behu by mala spadať medzi ľahké behy a behy pri prahovej intenzite, čo zodpovedá približne 76 až 82 percent $\dot{V}O_2max$, alebo 79 až 85 percent maximálnej hodnoty srdcovej frekvencie užívateľa.

Prahová intenzita

Reprezentuje tréning pri intenzite ktorú je oddýchnutý bežec schopný vydržať približne 60 minút. Cieľom tréningu pri tejto intenzite, je dovoliť telu zlepšiť schopnosť odstraňovania kyseliny mliečnej z krvi. Počas tréningov pri prahovej intenzite by sa mala hladina kyseliny mliečnej udržiavať pod maximálnou znášanou hladinou (laktátový prah). Účelom týchto tréningov je zvýšiť rýchlosť ktorú je bežec počas behu schopný udržať bez toho, aby množstvo kyseliny mliečnej v krvi nepresiahlo maximálny prah. Táto intenzita zodpovedá asi 86 až 88 percentám $\dot{V}O_2max$, alebo 88 až 90 percentám maximálnej srdcovej frekvencie. Bežecké tréningy v tejto kategórii sa dajú rozdeliť na dva druhy. Buď do stáleho behu, ktorý bude trvať 20 až 30 minút, alebo do niekoľkých behov – typicky dvoch až šiestich – rozdelených

na trvanie 5 až 10 minút, s odpočinkom trvajúcim asi 2 minúty. Oba typy by mali byť vykonávané pri rovnakej rýchlosti.

Intervalová intenzita

Hlavným cieľom tréningov spadajúcich do tejto kategórie je maximalizovať aeróbnu silu, teda zvýšiť $\dot{V}O_2max$. Tempo týchto tréningov by malo byť veľmi blízke tempu, kedy sa dosiahne maximálna hodnota $\dot{V}O_2max$, alebo tempu ktoré sa bude približovať maximálnej hodnote srdcovej frekvencie. Z dôvodu veľkej fyzickej náročnosti, je potrebné tieto tréningy vykonávať spôsobom beh – oddych – beh. Narozdiel od testovania hladiny $\dot{V}O_2max$, kde sa dĺžka testov blíži k 11 minútám, je doporučené aby sa tréningy v tejto kategórii nevykonávali dlhšie ako 3 až 5 minút. To zodpovedá dobe mierne dlhšej, za akú sme sa na danú hodnotu $\dot{V}O_2max$ vedeli dostať (90 až 120 sekúnd). Dlhšia doba behu pri tejto intenzite nie je žiadúca. Nemenej dôležité je dodržiavať dĺžku oddychového intervalu. Ten by mal byť dostatočne dlhý na to, aby sa hodnoty zvýšené z predchádzajúceho behu dostali do normálnych hladín. Táto doba predstavuje čas dvoch až piatich minút. Môže pozostávať z ľahkého poklusu, alebo rýchlejšej chôdze.

Intenzita opakovaní

Tento typ tréningu slúži na rozvoj rýchlych svalových vlákien, ktoré sú dôležitým aspektom zvýšenia tempa. Okrem toho slúži na rozvoj anaeróbnej sily. Rovnako by si bežec mal všimnúť svoju bežeckú formu – vzpriamený postoj, neprehnutý krk, švih rúk, dopad na prednú, alebo strednú časť chodidla, správna kadencia a tak ďalej. Tak ako pri tréningu v intervalovej intenzite, je dôležité dovoliť dostatočnú dobu na oddych, tak aby došlo k úplnej rekonvalescencii. Je dôležité, aby sa táto doba neskracovala, nakoľko by sa mohla zvyšovať únava, čo by malo za následok celkový úpadok formy. V tom prípade by tréning strácal zmysel. Je vhodné vyhnúť sa poklusu medzi opakovaniami, zaradená by mala byť skôr chôdza. Typická forma tréningu pozostáva z 10 niekedy až 20 opakovaní, kde sa jedná o úseky 200 až 400 metrov. Trvanie intervalu by malo byť 40 až 90 sekúnd.

2.4 Prevencia zranenia

V súvislosti so zvýšenou záťažou svalstva a šliach dochádza k väčšiemu riziku zranenia. Je preto vhodné svalstvo pravidelne natahovať, uvoľňovať a masírovať. Rovnako by sa nemala zanedbávať ani svalová príprava a to hlavne nôh a brušného svalstva.

Statický strečing

Cieľom je pretiahnuť svalstvo, uvoľniť ho a zabrániť stuhnutiu. Je nutné ho aplikovať po bežeckej aktivite, nikdy nie pred a to z dôvodu zvýšenia flexibility svalov. To nie je pred behom vhodné. V súčasnej dobe sa uvažuje, že by statický strečing mohol byť kontraproduktívny a nevhodný^[5]. Je dobré sa statickému strečingu vyhnúť počas zranenia, hlavne počas zranenia šliach. Naopak je vhodný v prípade stuhnutého svalstva po tréningu, na zabránenie zranení akým je na príklad bežecké koleno⁴. Dôležité je strečing nepraktizovať príliš dlho, nakoľko by mohlo dôjsť k zraneniu. Ideálna je doba v trvaní 45 sekúnd.

⁴<https://www.hopkinsmedicine.org/health/conditions-and-diseases/patellofemoral-pain-syndrome-runners-knee>

Dynamický strečing

Vhodný doplnok zahriatia svalstva pred samotným bežeckým tréningom. Dynamický strečing má za následok zvýšený tok krvi do svalstva, čím podporuje rýchlosť kontrakcií svalstva, prenos kyslíka z krvi do svalstva. Je vhodné ho doplniť o ľahký poklus nakoľko so zvýšenou teplotou tela sa zlepši svalový metabolizmus a nervový prenos.[1]

Silové cvičenie

Silový tréning je neoddeliteľnou súčasťou prípravy na bežecký závod. Dá sa rozdeliť na dve etapy a to na etapu pred sezónou – doba kedy nie sú naplánované žiadne závody a etapu v hlavnej sezóne. V predsezónnej etape je vhodné silové cvičenia praktizovať približne dva krát za týždeň, kedy príde k rozvoju nielen maximálnej sily. V hlavnej sezóne potom prichádza redukcia na jeden silový tréning za týždeň [3]. Tento tréning má za úlohu udržanie pozitívnych efektov z predsezónnej časti prípravy. Je vhodné ho komponovať z cvikov pri ktorých sa rozvíja sila v krátkych časových intervaloch, ide o takzvané plyometrické cviky. Je rovnako vhodné zaradenie cvikov ktoré sa zameriavajú na brušné a chrbtové svalstvo. Naberanie svalovej hmoty v hornej časti tela je nežiadúce, nakoľko pridáva váhu, pričom je však dôležité toto svalstvo úplne neignorovať.

Iné športy ako beh

Z dôvodu častých zranení pohybového aparátu pri bežeckom tréningu je vhodné aspoň raz za týždeň absolvovať iný tréning zameraný na získanie aeróbnej sily. Z toho pohľadu sa naskytuje niekoľko alternatív a to menovite plávanie, cyklistika a box. Zaradenie plaveckých tréningov má vplyv na srdcové svalstvo bez toho, aby trpeli kĺby a šlachy, čím predchádza nepríjemným bežeckým zraneniam. Zmena bežeckých tenisiek za bazén má pozitívny vplyv na rozvoj svalovej hmoty na chrbte a bruchu. Zaradenie cyklistických tréningov vplyva na budovanie svalovej hmoty na nohách. Je veľmi účinné v prípadoch, keď bežca začne trápiť bolesť kolien. Jedným z dôvodov týchto bolestí je aj nedostatok svalovej hmoty v tejto oblasti. Rovnako vplyva na srdcové svalstvo a má tak pozitívny vplyv na celkovú výdrž bežca. Avšak z cyklistického tréningu vyplýva aj negatívum, podľa publikácie [4] dochádza k zhoršeniu bežeckej formy. Dochádza k problémom držania tela a to z dôvodu pozície v ktorej na bicykli trávime čas. Dochádza k náklonu tela do predu. Box je vhodný ako náhrada za silové cvičenie a aeróbne cvičenie zároveň. Rozvíja brušné, chrbtové svalstvo, rýchlosť, rovnováhu a zvyšuje výdrž. Podľa článku z populárneho bežeckého magazínu [11] môže byť nápomocný aj z mentálneho hľadiska, nakoľko zvyšuje mentálnu odolnosť ktorú potrebuje každý bežec.

Možná štruktúra tréningov

Zvolená metodika Jacka Danielsa 2.2 umožňuje vhodne kombinovať tréningy v jednotlivých intenzitách tak, aby vznikol tréning vhodný pre každého bežca. V tabuľkách 2.2, 2.3, 2.4 a 2.5 sú uvedené ukážky možných tréningov.

Typ cvičenia	Trvanie
10 min. E , 40 min. M , 5 min. E	55 minút
15 min. E , 50 min. M , 10 min. E	75 minút
30 min. E , 50 min. M , 10 min. E	90 minút

Tabuľka 2.2: Tabuľka popisujúca možný tréning — E značí ľahký beh, M značí maratónsku intenzitu

Typ cvičenia	Trvanie
15 min. E , 30 min. M , 5 min. T , 5 min. E	55 minút
15 min. E , 5 min. T , 40 min. M , 5 min. T , 10 min. M , 5 min. T , 10 min. E	90 minút

Tabuľka 2.3: Tabuľka popisujúca možný tréning — E ľahká intenzita, M maratónska intenzita, T tempo intenzita

Typ cvičenia	Trvanie
20 minútový beh v T intenzite	20 minút
6 x 5 min. beh v T intenzite s 2 min. oddychom	38 minút
2 x 15 min. beh v T intenzite s 3 až 4 min. oddychom	až 40 minút minút

Tabuľka 2.4: Tabuľka popisujúca možný tréning v tempovej T intenzite

Typ cvičenia	Trvanie
8 x 200m R s 300 m E	18 minút
4 x 400m R s 400 až 500m E	20 minút
4 x 200m R s 200m E , 2x 400m R s 400m E , 4 x 200m R a 200m E	až 27 minút

Tabuľka 2.5: Tabuľka popisujúca možný tréning v tempe pre opakovania R

Kapitola 3

Východiskový stav technológií

Kapitola popisuje problematiku tvorby mobilných aplikácií pre zariadenia s operačným systémom iOS. Stručne popisuje nástroje a postupy používané v dnešnej dobe – programovací jazyk Swift, lokálne úložisko CoreData, knižnicu CloudKit, HealthKit a MapKit. Posledná sekcia prezentuje analýzu existujúcich riešení.

3.1 Technológie tvorby iOS aplikácií

Pre tvorbu mobilných natívnych aplikácií pre platformu iOS sa využívajú primárne dva programovacie jazyky. Objective-C ktorý je v súčasnej dobe využívaný menej ako jeho nástupca Swift. V roku 2019 spoločnosť predstavila framework SwiftUI ktorý zrevolucionizoval spôsob ktorým sa tvorili užívateľské rozhrania pre všetky zariadenia firmy Apple, pričom sa do istej miery snaží nahradiť framework UIKit.

Swift

Programovací jazyk Swift bol predstavený firmou Apple v roku 2014. Bol vytvorený ako nástupca Objective-C s cieľom rozšíriť jeho možnosti a urobiť nový jazyk viac atraktívny novým programátorom, nakoľko kód jazyka Objective-C bol ťažko čitateľný. Je to multiparadigmatický, objektovo orientovaný programovací jazyk, ktorý prišiel s novou programovacou paradigmou – protokolovo orientované programovanie. Narozdiel od svojho predchodcu Swift obsahuje silnú typovú kontrolu a upúšťa od používania ukazateľov. V dôsledku uvedených zmien sa snaží umožniť jednoduchšie hľadanie a odstraňovanie chýb. Okrem iného je Swift približne trikrát rýchlejší ako Objective-C, pričom sa v niektorých testoch približuje výkonu C++^[8]. V roku 2018 sa stal preferovaným jazykom pre tvorbu aplikácie pre zariadenia od firmy Apple.

UIKit

UIKit je imperatívny framework ktorý definuje základné komponenty iOS aplikácie ako sú tlačítka, text, obrázky, navigácia. Na implementáciu rozhrania aplikácie poskytuje architektúru zostavenú z okien (anglicky window) a pohľadov (anglicky view). Zabezpečuje spracovanie udalostí pre rôzne typy vstupov a obstaráva hlavnú slučku behu programu ktorá je zodpovedná za správu interakcií medzi užívateľom, systémom a aplikáciou. Mimo iné podporuje animácie, kreslenie, vyhľadávanie. Poskytuje podporu pre správu zdrojov

a informácie o súčasnom zariadení. UIKit sa využíva na tvorbu grafického, udalosťami riadeného užívateľského rozhrania pre aplikácie na platforme iOS, alebo tvOS

SwiftUI

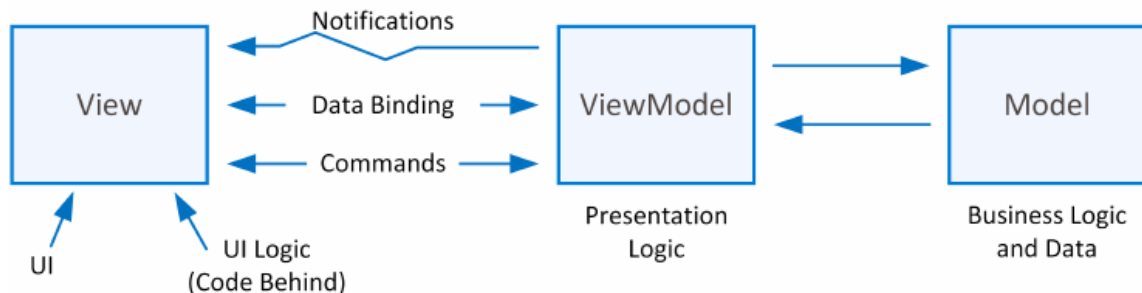
SwiftUI je framework predstavený na konferencii WWDC v roku 2019. Poskytuje nový spôsob vytvárania aplikácií, ktorý umožňuje aplikácie dizajnovat deklarativným spôsobom za použitia syntaxe jazyka Swift. Umožňuje teda jednoduchšiu a rýchlejšiu tvorbu užívateľského rozhrania na všetky dostupné platformy od spoločnosti Apple.

3.2 Návrhový vzor Model–View–Viewmodel

Návrhové vzory slúžia na reprezentáciu niektorých vhodných postupov pri návrhu a implementácii aplikácie. Návrhový vzor pomenúva a vysvetľuje všeobecný dizajn, ktorý má za úlohu riešiť niektoré problémy pri návrhu aplikácií a softvéru. Popisuje problém, prezentuje jeho riešenie spolu s informáciou kedy návrhový vzor použiť.

Je doporučený návrhový vzor pre použitie s frameworkom SwiftUI[21], ktorý oddeluje objekty v aplikácií do troch skupín:

- **Model** – drží aplikačné dáta, zvyčajne sa jedná o štruktúru alebo triedu
- **View** – zobrazuje vizuálne elementy a ovládacie prvky na obrazovke
- **View model** – zaobstaráva funkčnú časť ktorú prezentuje view, obsahuje funkcie a metódy



Obr. 3.1: Obrázok znázorňujúci návrhový vzor Model-View-Viewmodel. Obrázok prebratý z publikácie [22]

3.3 CoreData

CoreData je framework, ktorý poskytuje nástroje na správu objektov modelovej vrstvy v aplikácií. Zovšeobecňuje a automatizuje riešenia bežných úloh ktoré sú úzko spojené so životným cyklom objektov, správou objektových grafov. Rovnako zabezpečuje perzistenciu, čo znamená, že dáta sú uložené na fyzickej pamäti na užívateľovom zariadení. Dáta teda zostávajú zachované aj po ukončení aplikácie. CoreData zabezpečuje monitorovanie zmien jednotlivých objektov, vyhľadávanie, tvorbu, editáciu a mazanie objektov. Rovnako zabezpečuje kontrolu hodnôt objektov. Pri zmene schémy databázy ponúka nástroj na jednoduchú migráciu starej databázy na novú. Týmto spôsobom sa podľa oficiálneho stanoviska

spoločnosti Apple zredukujú percento kódu, ktoré je nutné napísať pre modelovú vrstvu až o 70 percent[12]. Pre vytvorenie databázy framework CoreData používa vstavaný databázový systém SQLite, ktorý sa nachádza v každom zariadení iOS. Framework sa skladá z niekoľkých hlavných komponentov ktoré sú dôležité pre jeho správnu funkciu.

Entita

Entita je trieda ktorá je reprezentovaná v objektovom grafe. Každá entita by mala mať meno a atribúty, ktoré reprezentujú stav danej entity. Avšak niektoré entity môžu byť vo vzťahu s inými entitami, pričom tento vzťah reprezentuje referenciu na iný záznam v CoreData. Medzi podporované typy vzťahov medzi entitami patrí vzťah jeden k jednému (anglicky one to one), jeden k mnohým (anglicky one to many), mnoho k mnohým (anglicky many to many). Každý vzťah môže mať určité atribúty, pričom najdôležitejším z nich je pravidlo mazania. Toto pravidlo určuje čo sa stane s entitami, ktoré sú v momentálnom vzťahu s mazanou entitou.

Managed Object Model

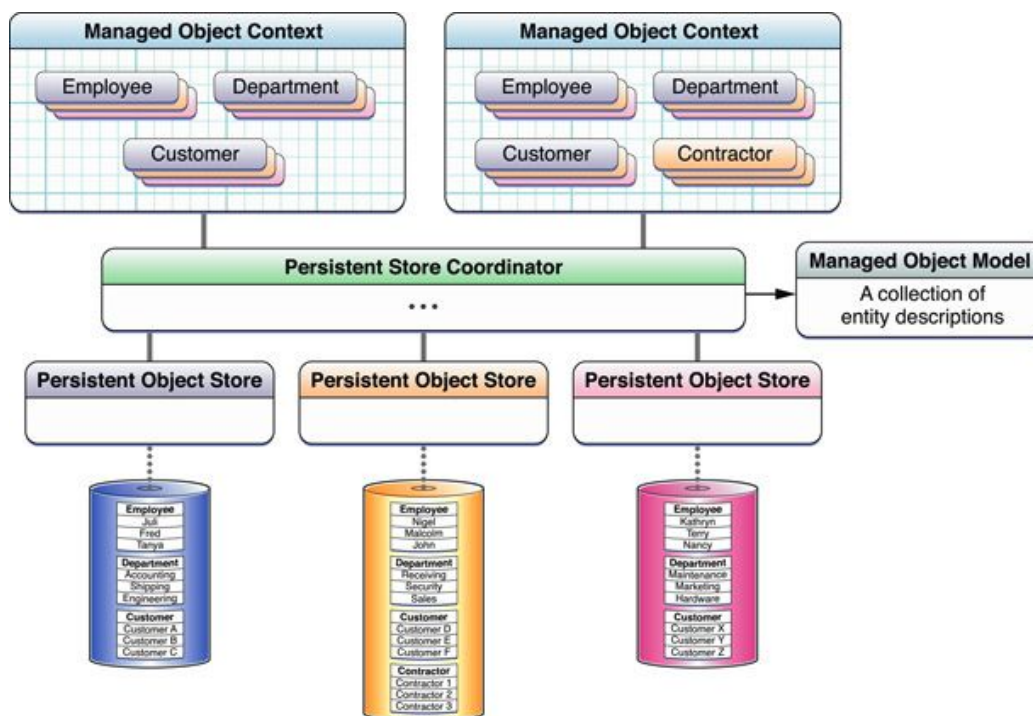
Tvorí schému ktorá definuje všetky objekty s ktorými sa bude pracovať. Schéma obsahuje názvy jednotlivých objektov, ich atribúty a vzťahy s ostatnými objektami modelu (anglicky managed object). Objekt modelu reprezentuje záznam uložený v trvalom úložisku. Obsahuje referenciu na popis entity objektu, ktorý určuje akú entitu reprezentuje. Týmto spôsobom objekt môže predstavovať ktorúkoľvek entitu.

Managed Object Context

Predstavuje jediný objektový priestor v aplikácii s CoreData. Jeho primárnou úlohou je spravovanie kolekcie jednotlivých spravovaných objektov (anglicky managed objects). Tieto spravované objekty predstavujú interne konzistenté zobrazenie jedného, alebo viacerých perzistentných úložísk. Z perspektívy programátora je kontext hlavný objekt v CoreData. Je to objekt, pomocou ktorého sa vytvárajú, modifikujú a vyhľadávajú jednotlivé spravované objekty. V danom kontexte existuje najviac jeden spravovaný objekt, ktorý predstavuje akýkoľvek daný záznam v trvalom úložisku[13].

Persistent Store Coordinator

Prepojuje trvalé úložiská objektov s managed object modelom. Predstavuje skupinu kontextov spravovaných objektov tak, že skupina trvalých úložísk sa javí ako jeden agregovaný ukladač priestor. Koordinátor trvalého úložiska obsahuje odkaz na model spravovaného objektu, ktorý popisuje jednotlivé entity v úložiskách ktoré spravuje. Vo väčšine aplikácií sa nachádza iba jeden koordinátor, avšak pri robustných aplikáciách môže koordinátorov existovať viac. Pri dotazovaní sa na záznamy, CoreData vráti záznamy zo všetkých koordinátorov, pokiaľ nie je špecifikované inak. Teda koordinátor slúži na získanie dát z databázy ktoré prezentuje ich managed object context.



Obr. 3.2: Obrázok znázorňujúci funkciu persistent store coordinatoru. Obrázok prebratý z dokumentácie spoločnosti Apple [14]

3.4 CloudKit

CloudKit je systém od spoločnosti Apple, ktorý umožňuje ukladanie dát na iCloud jednotlivých užívateľov a následné čítanie týchto dát. Ponúka jednoduchosť nastavenia. Vývojárom sa stačí zaregistrovať do programu iOS Developer Program, nie je nutné sťahovať dodatočné knižnice a nie je nutné míňať veľké množstvo peňazí[16]. Pre užívateľa je výhodné z dôvodu ochrany osobných údajov, nakoľko svoje dáta neukladá na serveroch vývojára, ale na serveroch spoločnosti Apple. Ponúka dva druhy databáz. Verejnú – je zdieľaná medzi všetkých užívateľov kontajnera. Každý užívateľ kontajnera má prístup k dátam ktoré obsahuje a vo východnom stave do nej môže aj každý zapisovať. Súkromnú – slúži iba danému užívateľovi, čo znamená že užívateľ musí byť prihlásený na svoj iCloud účet. Dáta zo súkromnej databázy nie je možné zdieľať a prístup k nim nemá ani vývojár.

Základné objekty

CloudKit obsahuje 7 základných objektov ktoré sú pre jeho funkciu elementárne. Sú to:

- **CKDatabase** – je súčasťou kontajnera a rozlišuje dva typy databázy. Súkromná a verejná, pričom slúži na samotné ukladanie všetkých záznamov
- **CKContainer** – každá aplikácia má iba jeden kontajner. Reprezentuje ohraničené územie, mimo ktoré aplikácia nemôže zasahovať
- **CKRecord** – jedná sa o časť dát uložených v databáze vo forme kľúča a hodnoty, pričom sa dajú uložiť hodnoty typu NSString, NSNumber, NSDate, CLLocation, CKAsset, CKReference

- **CKRecordZone** – záznamy nie sú roztrúsené po celej databáze, sú v takzvaných záznamových zónach. Každá aplikácia má hlavnú zónu a môže mať aj vytvorené ďalšie vedľajšie zóny
- **CKRecordIdentifier** – jedinečný identifikátor záznamu, slúžiaci na jednoduché vyhľadávanie
- **CKReference** – je vzťah medzi entitami, slúži na pridelenie kontextu
- **CKAsset** – reprezentuje rôzne druhy zdrojov, avšak primárne by sa malo jednať o binárne súbory, fotky užívateľov

3.5 HealthKit

HealthKit bol predstavený spoločnosťou Apple v roku 2014. Ponúka riešenie pre ukladanie, zdieľanie a editáciu zdravotných dát užívateľa. Vývojárom umožňuje čítať, alebo zapisovať nové zdravotné dáta do tejto databázy, ktorá je spravovaná operačným systémom.

Kľúčovou vlastnosťou je spomínaná možnosť zdieľať dáta, a to nielen s jednou aplikáciou. HealthKit je navrhnutý tak, aby umožnil kombinovať a agregovať dáta z rôznych aplikácií. Teda jedna aplikácia dáta nahrá, druhá ich analyzuje a vyhodnotí. Užívateľ musí jednotlivým aplikáciám povoliť prístup k daným zdravotným dátam a záznamom, ktoré sa dajú spravovať jednotlivo[15]. Samotný užívateľ rozhoduje o tom, ktoré dáta pre ktorú aplikáciu sú viditeľné. Samotné aplikácie aj v tomto prípade nebudú vedieť, že užívateľ im zamietol prístup a to z toho dôvodu, že pri zamietnutí prístupu sa HealthKit chová rovnako ako keby bol prístup povolený, ale neboli v ňom uložené žiadne dáta. Okrem toho stanovisko spoločnosti Apple uvádza, že aplikácie ktoré budú ukladať zdravotné dáta na iCloud nebudú prijaté do AppStore, nakoľko porušujú ochranu súkromia. To ukazuje veľké zameranie na zabezpečenie dát.

Dátové typy

HealthKit používa podtriedy triedy HKObjectType na úspešnú identifikáciu jednotlivých dát uložených v tomto úložisku. Jedná sa o dáta ktoré sa typicky nemenia, až po komplexné dáta ktoré sa menia v čase. Ponúka typy ktoré kombinujú dáta z niekoľkých zdrojov, poprípade počítajú priemernú alebo maximálnu hodnotu z niekoľkých dátových zdrojov. Medzi najpoužívanejšie dátové typy patria:

- **HKWorkoutType** – reprezentuje dáta ktoré obsahujú informácie o tréningoch
- **HKQuantityType** – typ reprezentujúci dáta ktoré ukladajú číselné hodnoty
- **HKClinicalTypeIdentifier** – reprezentuje dáta ktoré obsahujú informácie o zdravotnom zázname

Dotazy na záznam

Pre získanie dát zo záznamov sa využívajú dotazy. Môžu byť využité na získanie jednoduchých dát, na získanie zdrojov konkrétnych dát – či sa jedná o záznam z mobilného zariadenia, hodínok, hrudného pásu. Možné je tvoriť dotazy ktorých výsledok reprezentuje výsledok štatistického výpočtu cez veľké množstvo dát. Napríklad pre výpočet celkového

počtu krokov za týždeň. Okrem dotazov ktoré vrátia hodnotu záznamu okamžite, je možné použiť dotazy ktoré neustále naslúchajú na zmeny dát. Tieto zmeny sa následne okamžite prenesú do výsledku dotazu. Dovoľuje výsledky jednotlivých dotazov triediť podľa takzvaných deskriptorov radenia (anglicky sort descriptor)

Existuje veľké množstvo dotazov, avšak medzi tie najvyužívanejšie patria:

- **HKSampleQuery** – najpoužívanejší dotaz ktorý vráti záznam všetkých dát, ktoré odpovedajú špecifikátoru a sú uložené v HealthKite
- **HKQuantitySeriesSampleQuery** – dotaz ktorý vráti zoznam hodnôt záznamu asociovaného s dátami reprezentujúcimi množstvo
- **HKAnchoredObjectQuery** – dotaz vracajúci zmeny, ktoré boli vykonané v HealthKite, pokračuje v monitorovaní dát
- **HKStatisticsCollectionQuery** – dotaz ktorý vykoná niekoľko štatistických dotazov nad sériou dát v pevne danom časovom pásme

3.6 MapKit

MapKit je natívny framework vytvorený spoločnosťou Apple, ktoré je dostupné na zariadeniach iOS. Umožňuje jednoducho zobrazovať mapu, alebo lokácie na mape doplnené o dáta o cestách. Umožňuje vykresľovať prejdenú trasu a pridať anotácie k jednotlivým bodom záujmu – v prípade tejto práce, začiatok a koniec bežeckého tréningu.

Pridanie anotácií

MapKit umožňuje pridávať na mapu rôzne objekty. Jedným z týchto objektov je anotácia. Je to určitý bod na mape, ktorý sa nachádza na presne určených súradniciach. V praxi sa používajú tri druhy anotácií na vyznačenie určitého miesta.

Najjednoduchšou variantou je použitie `MKPlacemark`[18]. Dovoľuje na mapu vložiť takzvaný špendlík. Všetko čo je na vloženie špendlíka na mapu potrebné, sú len súradnice vo formáte `CLLocationCoordinate2D`. Druhý spôsob akým vytvoriť anotáciu, je použitie triedy `MKPointAnnotation`, ktorá umožňuje okrem vyznačenia pomocou jednoduchého špendlíka, aj špecifikáciu názvu, poprípade aj podnázvu. Posledný spôsob ktorým je anotáciu možné vytvoriť, je použitie rozhrania `MKAnnotation`. Toto rozhranie spojuje nami vytvorené anotácie s miestami na mape. Všetky tieto anotácie môžu byť následne zlúčené a to v závislosti od okolia, ktoré si užívateľ prezerá – pri prezeraní mapy štátu sa nad mestom zobrazí anotácia s detailným počtom anotácií v tom danom meste.

Jednotlivé typy máp a spôsoby ich zobrazenia

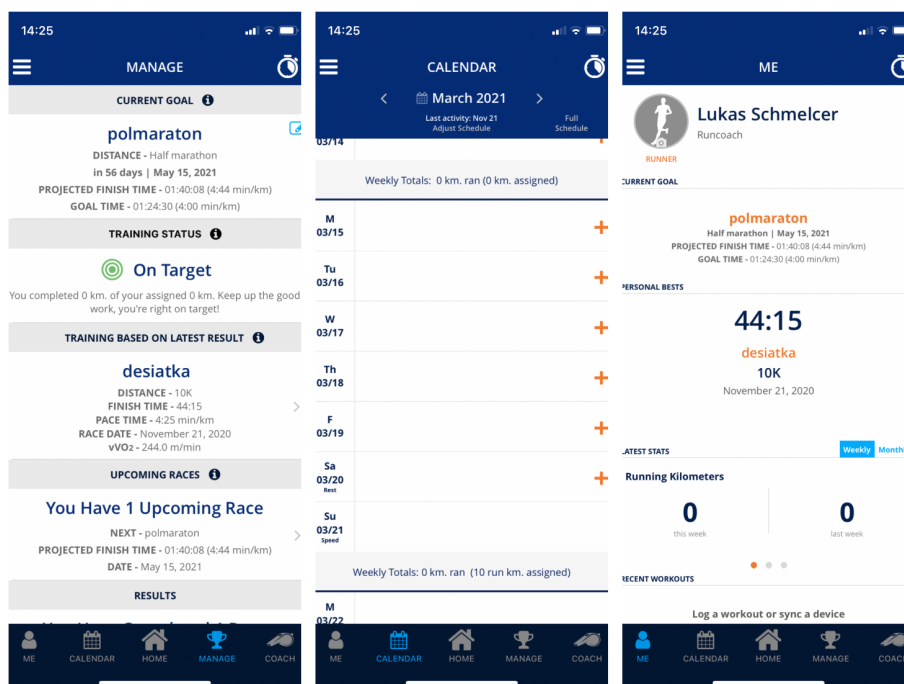
MapKit ponúka štyri štandardné druhy máp. Každý z týchto druhov má špecifické vlastnosti. Štandardné mapy – `.standard`. Sú to mapy, ktoré ukazujú pozície všetkých ciest a ulíc, pričom pri niektorých uliciach, alebo cestách zobrazuje aj ich názov. Hybridné mapy – `.hybrid`. Hybridný typ máp prezentuje satelitnú snímku špecifikovaného okolia. Tento snímok je obohatený o informácie o cestách a uliciach. Ďalej MapKit poskytuje mapy, ktoré sú satelitným snímkom špecifikovaného okolia – `.satellite`. Posledným typom máp je `.mutedStandard`. Jedná sa o štandardné mapy, ktoré zvyrazňujú elementy ktoré boli pridané programátorom.

3.7 Analýza existujúcich riešení

Celkom boli vybrané dve riešenia. Riešenia boli vyberané tak, aby sa čo najvernejšie približovali cieľu aplikácie, teda umožnili sledovať ukončené bežecké tréningy, získať vedomosti o bežekom tréningu, ukladať si spätnú väzbu a plánovať tréning. Systém musí tréning ponúkať automaticky, s možnosťou užívateľovej interakcie.

RunCoach

RunCoach je aplikácia zameraná na bežcov všetkých bežeckých kategórií, či sa jedná o začiatočníkov, alebo profesionálnych atlétov. Umožňuje špecifikovať cieľ, nahrať výsledky závodov. Vytvorí tréningový plán na základe posledného závodu a postupne zvyšuje záťaž až po cieľový závod. Má jednoduchý systém spätnej väzby – tréning sa mi páčil, tréning sa mi nepáčil. Mimo bežeckých tréningov, do tréningového cyklu vkladá tréning na svalstvo a zaraďuje oddych. Umožňuje si jednoducho pozrieť týždenný, alebo mesačný prehľad nabehaných kilometrov. Rovnako má funkciu zaznamenania behu, avšak ukazuje veľmi limitované informácie. Unikátnou funkciou RunCoach aplikácie je možnosť kontaktovať certifikovaného trénera, ktorý užívateľovi vysvetlí plán, poprípade ho pozmení.

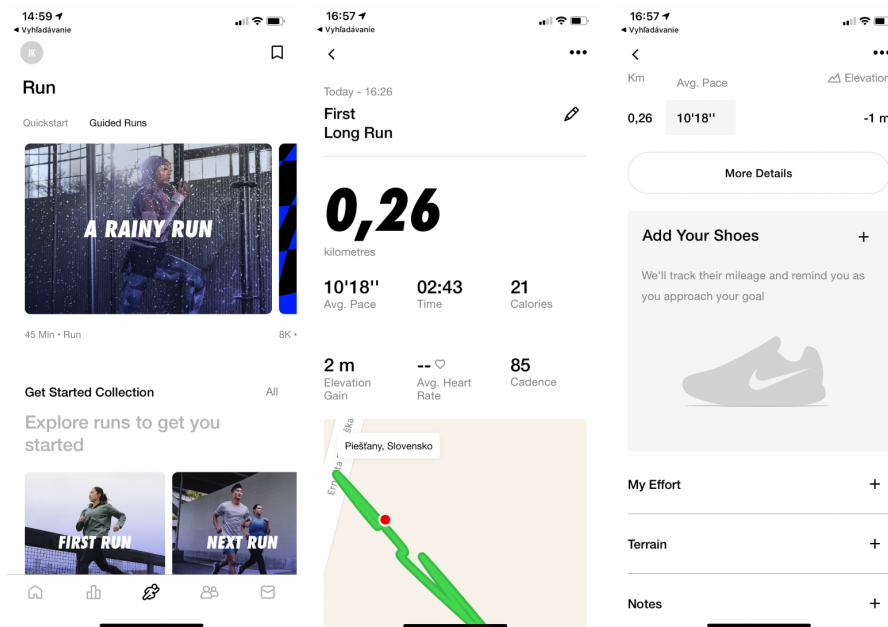


Obr. 3.3: Aplikácia RunCoach – zľava: Prehľad cieľov, kalendár tréningov, užívateľský profil

Nike Run Club

Aplikácia vytvorená spoločnosťou Nike. Po vyplnení základného profilu užívateľovi ponúkne takzvaný riadený beh (anglicky guided run). Na výber ich je niekoľko, avšak nie sú usporobené pre každého bežca osobitne. Po začatí behu užívateľa sprevádza vopred nahratá audio nahrávka od trénera Nike. Na vopred definovaných úsekoch, vždy oznámi v akej fáze tréningu sa človek nachádza, povie čo čakať od nasledujúceho úseku. Po ukončení behu poskytuje možnosť zobraziť si detail o dokončenom tréningu. Zobrazí mapu, odbehnutú

vzdialenosť, kalórie a iné základné štatistiky, avšak nie vo forme grafov, ale iba v písomnej forme. Umožňuje dokončenú aktivitu priradiť k bežeckým teniskám, aby sa ľahko sledovali nabehané kilometre. K tréningu rovnako ponúka možnosť priradiť poznámku. Okrem spomínanej funkcionality aplikácia umožňuje vytvárať a spĺňať výzvy vytvorené iným užívateľom alebo firmou Nike. Pri splnení výzvy užívateľ dostane odznak (anglicky achievement) za jej splnenie.



Obr. 3.4: Aplikácia Nike Run Club – zľava: Prehľad riadených behov, prehľad po ukončení tréningu, detaily tréningu

Nedostatky existujúcich riešení a návrh na ich vypelšenie

V súčasnej dobe sa medzi aplikáciami ako problém javí povinnosť registrácie užívateľa. V dobe, kedy má každý užívateľ veľké množstvo účtov na mnohých službách, sa každý užívateľ zamyslí nad tým, či potrebuje práve danú službu, ktorá od neho tieto údaje požaduje. Nejde len o to, že užívateľovi vznikne ďalší účet, ale aj o to, že tieto v súčasnej dobe citlivé informácie poskytuje ďalšiemu prevádzkovateľovi. Stačí jeden bezpečnostný incident u ktoréhokolvek poskytovateľa a tieto údaje sa zrazu stanú verejné. Vo výslednej aplikácii je tento problém riešený tak, že sa užívateľove informácie synchronizujú na ďalšie zariadenia pomocou úložiska iCloud. Užívateľ sa tak nemusí prihlasovať do svojho účtu, nezabudne heslo a informácie o ňom sú stále skutočne jeho.

Druhým problémom teraz už len hore spomenutých aplikácií je nedostatok personalizácie tréningov pre užívateľa. Ak vôbec tréningy aplikácia generuje, vygeneruje ich tak, že sa až do zmeny cieľa nemenia. Nezáleží teda na tom, či sa užívateľ počas tréningu cíti dobre, alebo nie. Kludne si môže privodiť zranenie a aplikácia na túto skutočnosť nemá ako zareagovať. Prakticky v týchto aplikáciách neexistuje možnosť ako zaznamenať zranenie, teplotu, bolesť kĺbov a iných neprijemných kondícií. Tento nedostatok je vo výslednej aplikácii riešený tak, že ku každému dňu si užívateľ môže zadať spätnú väzbu. Podľa tejto spätnej väzby sa následne automaticky upraví tréning na ďalší deň. Ďalším spôsobom spätnej väzby

vo výslednej aplikácii je možnosť zadať zranenie. Po zadaní zranenia sa automaticky upraví tréning na nasledujúcich štrnásť dní, pričom sa toto zranenie berie do úvahy.

Posledným nedostatkom spomenutých aplikácií je slabá informovanosť užívateľa o ponúkaných tréningoch. Aplikácie okrem bežeckých tréningov odporúčajú vykonať silový tréning, alebo strečing. Avšak tieto odporúčania neobsahujú žiadne konkrétne cviky. Užívateľ teda nevie ktoré cviky by mal zaradiť do svojho tréningu tak, aby sa dosiahol požadovaný efekt. Vo výslednej aplikácii je tento problém riešený tak, že sa po rozkliknutí odporúčaného tréningu užívateľovi zobrazia všetky cviky, ktoré by mal do daného tréningu zaradiť. Nezáleží na tom, či sa jedná o silové cvičenie, preťahovanie svalov, alebo bežcom špecifické cviky. Pri každom cviku je obrázok, alebo video ktoré znázorňuje ako má byť cvik vykonaný. Pri niektorých je navyše uvedený popis a možné varianty. Užívateľ si tak môže celý tréning ešte sťažiť, poprípade uľahčiť a to podľa toho ako sa dnes cíti.

Kapitola 4

Návrh aplikácie

Kapitola popisuje kompletný návrh aplikácie. Definuje cieľovú skupinu, predstavuje prípady užitia a popisuje návrh užívateľského rozhrania aplikácie. Ďalej popisuje vznik názvu aplikácie a vznik maskota. Obsahuje podkapitoly, ktoré rozoberajú návrh systému spätnej väzby – teda to ako aplikácia reaguje na zranenia a pocity užívateľa. Poslednou témou tejto kapitoly je popísanie databázovej schémy.

4.1 Vymedzenie cieľovej skupiny

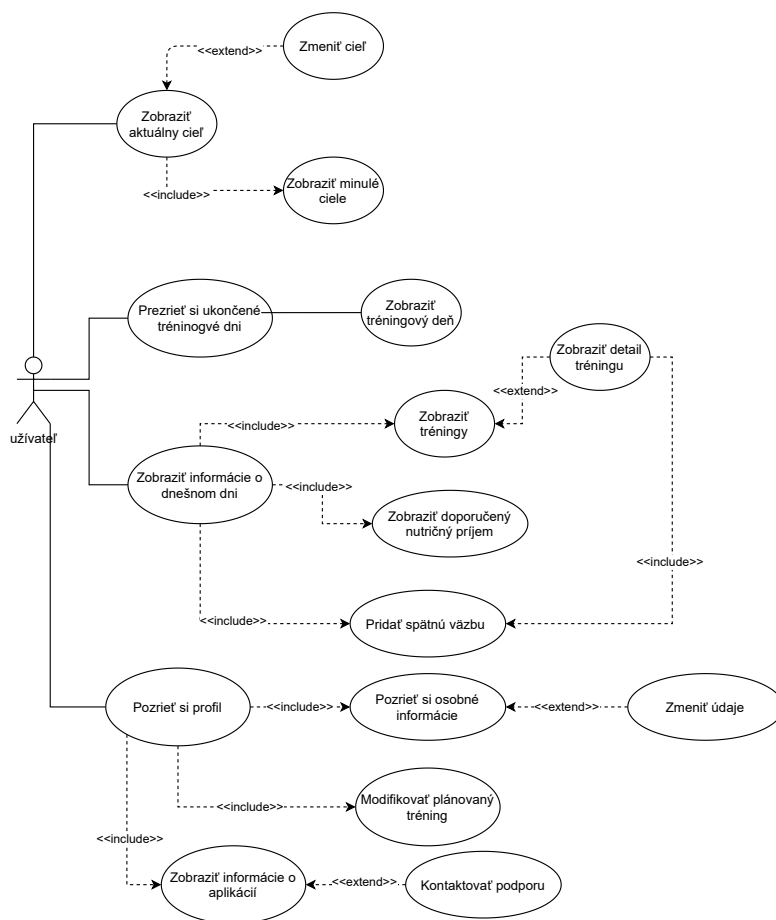
Cieľová skupina užívateľov ktorí by mali aplikáciu používať sú bežci, ktorí dokážu zabehnúť aspoň tri kilometre. Teda pokročilí začiatočníci a skúsenejší, ktorí sa chcú pripraviť na závod, zlepšiť svoj čas, alebo predĺžiť maximálnu odbehnutú vzdialenosť. Títo bežci zväčša nemajú členstvo v klube, alebo si neplatia osobného trénera.

Modelový cieľový užívateľ číslo jedna je žena. Posledný mesiac kombinuje rýchlu chôdzu s behom. Z dôvodu absencie bežeckého klubu v jej lokalite a nemožnosti navštevovať súkromný kurz sa rozhodne trénovať sama. Po čase zistí, že jej výkonnosť nestúpa a cíti sa viac unavená. Po stiahnutí aplikácie zistí, že jej tréning nebol optimálny – nezaraďovala silové cviky na rozvoj svalstva spodnej a brušnej časti tela. Intenzita jej tréningov bola priveľká. Po mesiaci užívania a praktizovaní cvikov a tréningov sa jej zlepši výkonnosť a minimalizuje pocit únavy a teda aj riziko zranenia.

Modelový cieľový užívateľ číslo dva je muž. Aktívny bežec, ktorí sa pravidelne zúčastňuje súťaží vo svojom meste. Navštevuje klub, avšak z dôvodu nutnosti prispôbiť sa ostatným, nedostáva pozornosť ktorú by potreboval. Aplikácia mu pomôže zlepšiť jeho momentálny najlepší čas, pričom využije funkcie spätnej väzby, aby dal najavo ktoré cviky sa mu páčia, prípadne ktoré mu spôsobujú bolesť. Pravidelne si vedie záznamy o tréningových dňoch ku ktorým sa spätne vracia. Pri zranení alebo bolesti vie okamžite ako to ovplyvní jeho tréning a aké kroky je vhodné podstúpiť.

4.2 Prípady užitia

Na základe vymedzenej skupiny mohli byť analyzované požiadavky na aplikáciu. Po vytriedení protichodných alebo nezmyselných požiadaviek vznikol diagram prípadu užitia. Diagram obsahuje aktéra, teda užívateľa aplikácie a prípady užitia.



Obr. 4.1: Diagram prípadov užitia aplikácie

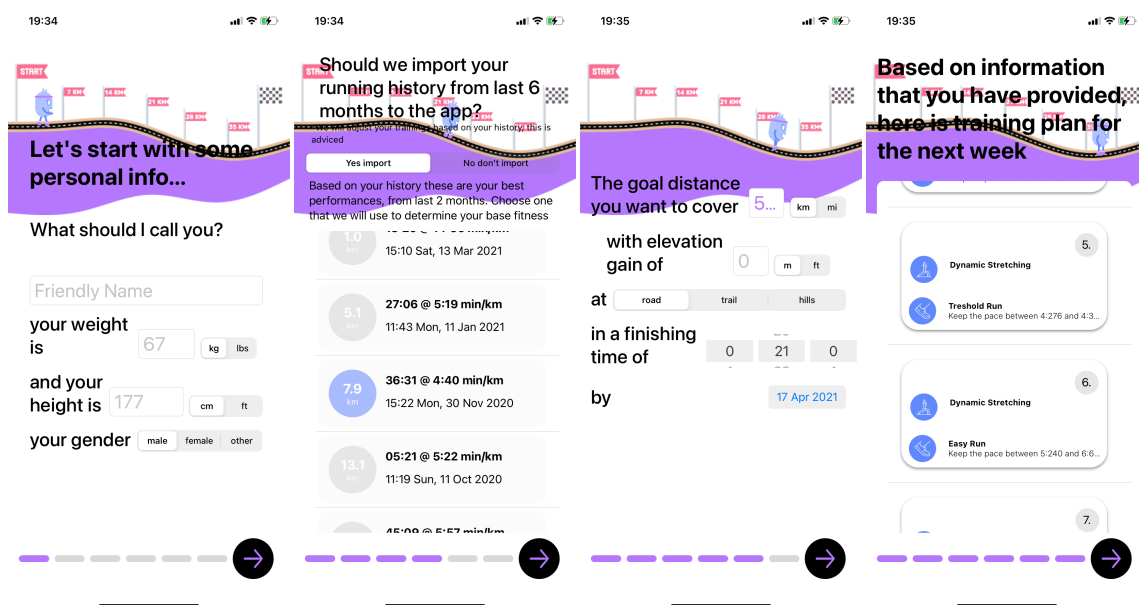
Po spustení aplikácie si užívateľ zobrazí informácie o dnešnom dni, ktoré pozostávajú z jednotlivých tréningov ktoré sú naplánované. Zobrazením detailu tréningu sa užívateľovi zobrazia dodatočné informácie o tréningu. Prípadne ak tréning obsahuje silové cviky, alebo cviky na pretiahnutie svalstva, budú v tejto sekcii zobrazené videá, alebo fotografie správneho výkonu. V prípade ak by bol bežecký tréning už ukončený, tak sa zobrazia štatistiky o tréningu. Je možné si prezrieť kadenciu, tempo, srdcový tep a mapu. K ukončenému tréningu môže užívateľ priradiť spätnú väzbu. Tá bude slúžiť na zaznamenanie pocitov z tréningu. Ak by mal užívateľ záujem pozrieť si staršie vykonané tréningy, alebo si iba spätne skontrolovať ako sa v daných dňoch cítil, môže využiť možnosť pozrieť si pre-

hľad dní. Po rozkliknutí konkrétneho dňa sa ukážu informácie o vybranom dni. Užívateľ si môže zobrazíť jeho aktuálny bežecký cieľ. Ten si môže zmeniť podľa toho ako sa cíti, alebo ak sa zmení dátum cieľového podujatia. Užívateľ má taktiež možnosť prezrieť si svoje splnené ciele. Ako jeden z posledných prípadov užívania je možnosť pozrieť si svoj profil, osobné informácie ktoré sa dajú zmeniť, tréningový plán na celý týždeň, alebo si pozrieť informácie o aplikácii. V aplikácii sa nachádza aj možnosť kontaktovať podporu v prípade záujmu nahlásiť chybu.

4.3 Užívateľské rozhranie

Po vytýčení požadovanej funkcionality aplikácie a dokončení diagramu prípadov užívania vznikol predbežný návrh užívateľského prostredia. Pri návrhu bol kladený dôraz na prehľadnosť a jednoduchosť použitia prostredia. Pri návrhu bol vynechaný proces vytrárania drátenných modelov, kde bol drátenný model nahradený menej detailným prototypom – prototypom ktorý neumožňoval interakciu. Ešte pred začatím tvorby prototypu bolo nutné vymedziť aj farebnú schému aplikácie. Tá bola zvolená tak, aby v cieľovej skupine vytvárala pocit sily, spoľahlivosti a úprimnosti¹.

Obrázok 4.2 ukazuje vstupné obrazovky, kde si užívateľ vytvorí svoj lokálny profil, vyberie si svoj najlepší bežecký výkon, definuje svoj cieľ a dostane prehľad vygenerovaného tréningu. Jedná sa o obrazovku ktorá sa užívateľovi zobrazí iba pri prvom spustení aplikácie.



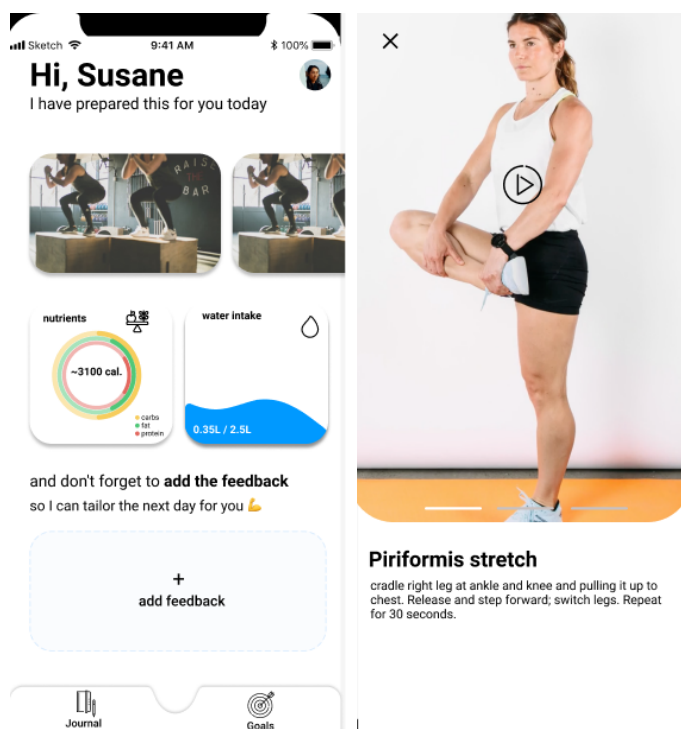
Obr. 4.2: Obrázok prezentujúci obrazovky pri prvom prihlásení

Na obrázku 4.3 je vidieť úvodná obrazovka aplikácie ktorá už bola inicializovaná užívateľom – nejedná sa o prvé otvorenie aplikácie a potrebné informácie od užívateľa už boli zadané. Užívateľ na úvodnej obrazovke vidí svoj denný prehľad, to aké tréningy boli na dnes pre neho navrhnuté, prehľad nutričných hodnôt a prehľad doporučeného príjmu tekutín.

¹<https://www.toptal.com/designers/ux/color-in-ux>

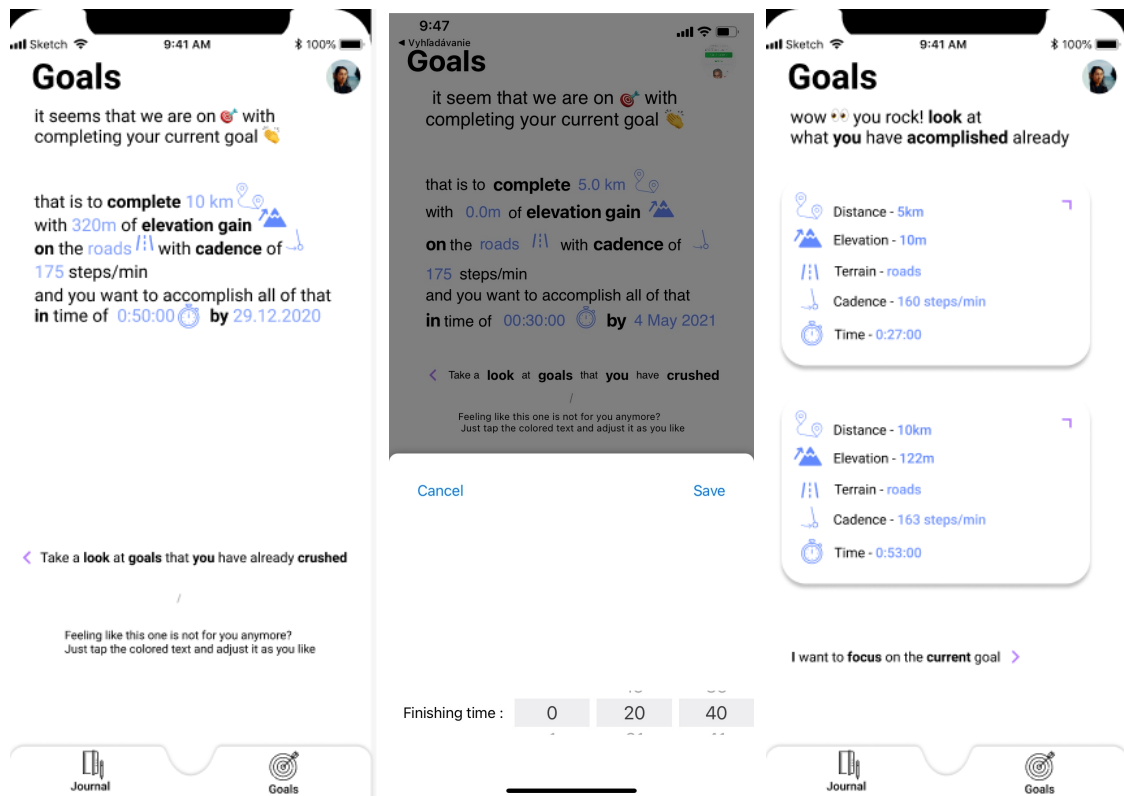
Posledná funkcia tejto obrazovky je možnosť pridať spätnú väzbu, kde v sa v prípade už pridenej spätnej väzby na daný deň aj zobrazí. Ak by užívateľ chcel zistiť viac informácií o dnešných tréningoch, môže na niektorý z tréningov kliknúť, čo ho dovedie na obrazovku s detailom tréningu.

Ďalej je na obrázku 4.3 zobrazená obrazovka prezentujúca detail tréningu, ktorá zobrazuje video, alebo fotografiu cviku ktorý sa má vykonať aj s popisom zvoleného cviku a inštrukciami k jeho vykonaniu. Pre uzavretie detailnej obrazovky tréningu je nutné kliknúť na tlačítko krížik v ľavom hornom rohu.



Obr. 4.3: Obrázok prezentujúci obrazovky zľava: úvodná obrazovka, obrazovka detailu tréningu

Na obrázku 4.4 sú zľava zobrazené obrazovky aktuálneho cieľa, editácie súčasného cieľa a obrazovka splnených cieľov. Obrazovka aktuálneho cieľa sa skladá z informácií o ňom prezentované v textovom formáte. Časti ktoré užívateľ môže meniť sú odlíšené od zvyšku inou farbou. Po kliknutí na vyznačenú informáciu sa užívateľovi zobrazí obrazovka editácie daného údaju. Ďalej obrazovka obsahuje tlačítko ktoré užívateľa presmeruje na obrazovku už splnených cieľov. Pri editácii cieľa zmenený údaj užívateľ môže buď uložiť kliknutím na tlačítko Save, alebo zrušiť kliknutím na tlačítko Cancel.



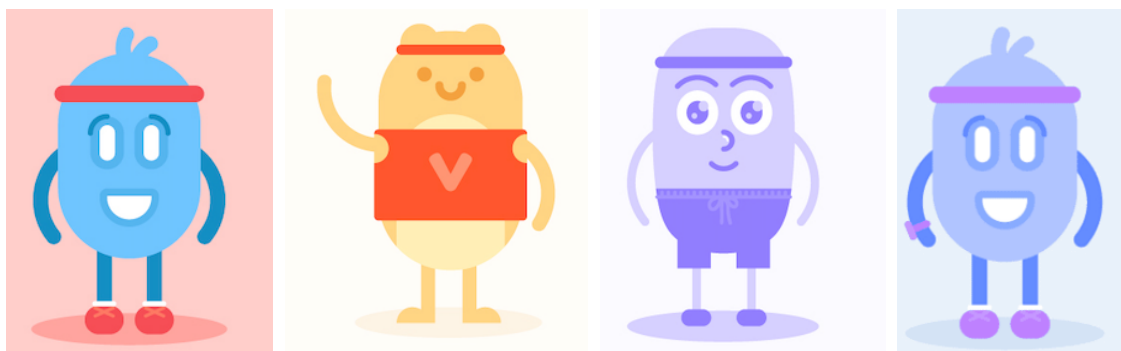
Obr. 4.4: Obrázok prezentujúci obrazovky užívateľových cieľov

Na všetkých hlavných obrazovkách je zobrazená spodná lišta, ktorá sa chová ako hlavná navigácia aplikácie a obsahuje dve tlačítka. Tlačítko **Journal** slúži na zobrazenie kalendára. Ten vedie na spätné zobrazenie tréningových dní. Tlačítko **Goals** vedie na obrazovku s vyznačeným cieľom. Druhým navigačným prvkom aplikácie je profilová ikonka, alebo fotka v pravom hornom rohu aplikácie. Toto tlačítko vedie na obrazovku s prehľadom nastavení profilu.

Voľba názvu aplikácie a vznik maskota

Počas návrhu aplikácie vznikol nápad na vytvorenie maskota. Maskot by mal slúžiť na vytvorenie lepšieho puta s užívateľom. Mal by vytvárať známe prostredie a osobnosť ktorej môže užívateľ dôverovať. Cieľom teda bolo vytvoriť priateľsky vyzerajúceho maskota, ktorý bude znázorňovať virtuálneho trénera.

Obrázok 4.5 zobrazuje rôzne varianty návrhu maskota. Nakoniec bol vybraný prvý maskot zľava. Boli upravené farby tak, aby ladili s farebnou schémou aplikácie. Výsledného maskota je taktiež možné vidieť na obrázku 4.5, kde je to prvý maskot z prava. Maskot by mal mať aj ľahko zapamätateľné a výstižné meno. Meno ktoré bolo pre maskota vybrané je VITO – Virtual Trainer Only (po slovensky iba virtuálny tréner).



Obr. 4.5: Obrázok prezentujúci tri návrhy maskota. Návrhy sú prvé tri postavy zľava. Finálny maskot je maskot najviac v pravo.

4.4 Systém spätnej väzby

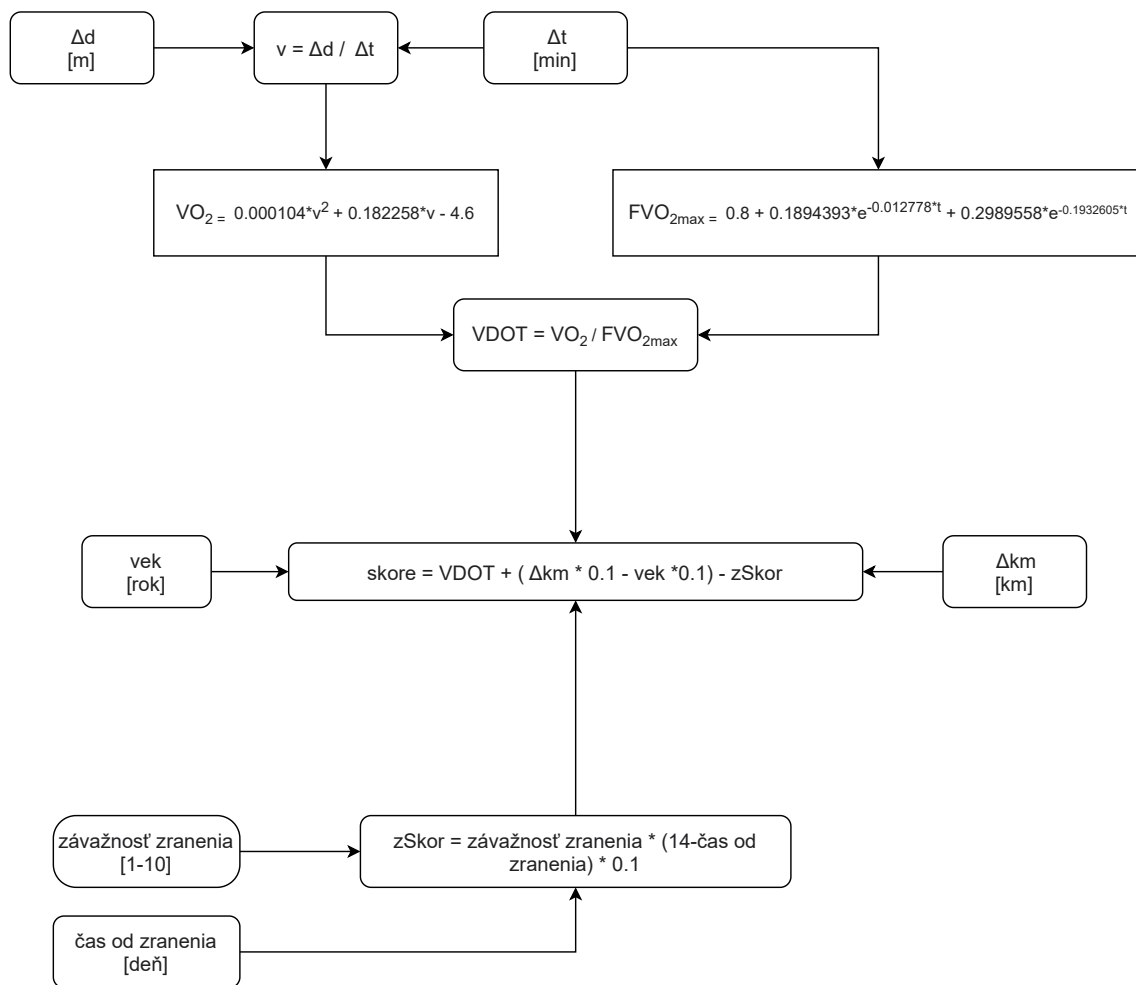
Z dôvodu meniť jednotlivé tréningové tempá podľa toho, ako sa užívateľ cíti, bolo nutné zaviesť určitý systém, ktorý bude brať tieto informácie do úvahy. Tieto informácie sú zbierané v podobe spätnej väzby. Budú automaticky uvážené v plánovaní tréningových temp na ďalšie dni, poprípade týždne.

Ako počiatočný bod výpočtu tempa je zvolený systém VDOT. Užívateľ má v aplikácii možnosť zadať zranenia, pocity z behu, pocity z jednotlivých dní. Vznikol tak systém spätnej väzby, ktorého výsledkom je skóre, ktoré bere do úvahy zranenia, vek bežca, jeho najlepší výkon a priemerný počet nabehaných kilometrov za posledný mesiac. Pre výpočet hodnoty VDOT je nutné poznať parametre najlepšieho bežeckého výkonu, ideálne nie staršieho ako 6 mesiacov. Je nutné poznať zabehnutú vzdialenosť v metroch, a čas v ktorom bola vzdialenosť zabehnutá. Ďalej bol vytvorený systém klzavého okna pre zranenia. Každé zranenie má priradenú závažnosť a deň kedy bol užívateľ zranený. Nakoniec stačí poznať počet nabehaných kilometrov za posledný mesiac a vek bežca. Výsledné skóre sa vnesie do rovnice na výpočet tempa 4.6, ktoré odzrkadluje tempo pre jednotlivé tréningové záťaže, ktoré boli popísané v podkapitole 2.3

$$tempo = \frac{(1000*2*0.000104)}{(-0.182258 + \sqrt{(0.182258^2 - 4*0.000104*(-4.6 - hrMax*skore)})}$$

Obr. 4.6: Rovnica na výpočet tempa bežeckého tréningu v minútách

- *hrMax* vyjadruje maximálne percento srdečnej frekvencie podľa podkapitoly 2.3
- *skore* je hodnota vypočítaná zohľadnením VDOT, veku, priemernej kilometráže podľa obrázku 4.7
- ostatné sú vhodne zvolené konštanty



Obr. 4.7: Spôsob výpočtu skóre, slúžiaceho na výpočet tempa

4.5 Návrh databázy

Perzistentné informácie o ukončenom tréningu sú uchované na zariadení. K prístupu k nim je použitá systémová knižnica HealthKit, pričom je prísne zakázané akékoľvek informácie o užívateľovi z HealthKitu ukladať cez CloudKit. Okrem informácií o vykonanom tréningu, je však nutné uchovávať informácie o samotnom užívateľovi, plánovaných tréningoch, alebo spätnej väzbe. Tieto informácie sú ukladané lokálne pomocou knižnice CoreData, pričom na synchronizáciu so zariadeniami bude využitá knižnica CloudKit.

Framework CoreData bol popísaný v podkapitole 3.3 a framework CloudKit bol popísaný v podkapitole 3.4. Obrázok 4.8 znázorňuje navrhnutý dátový model. Pričom jednotlivé entity s atribútami a reláciami sú:

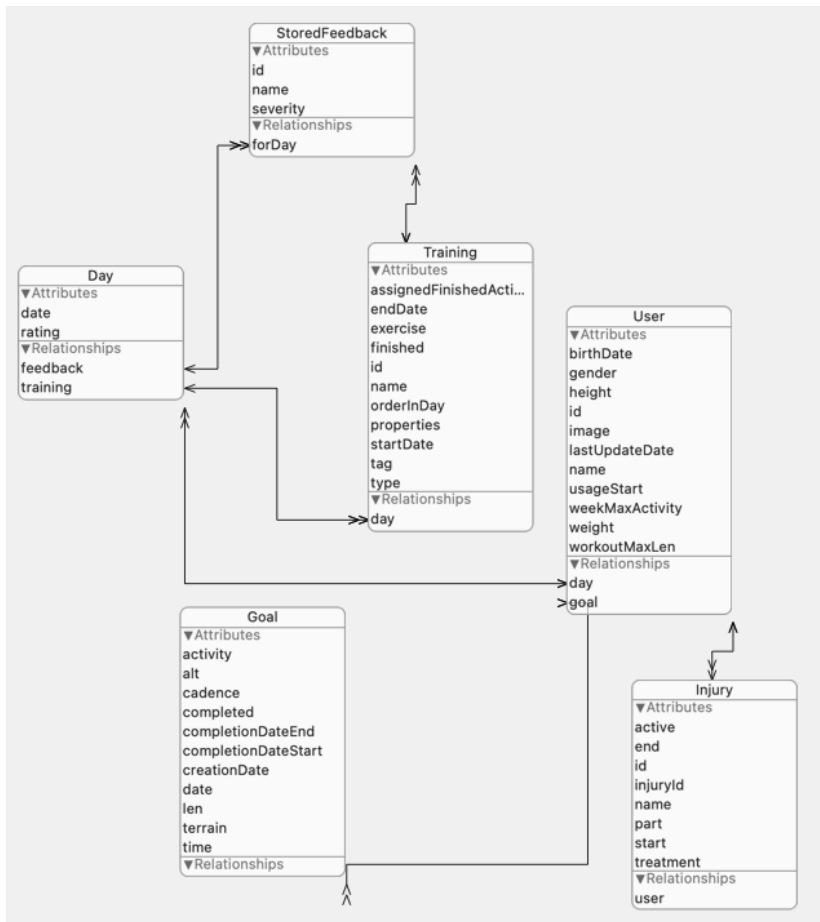
- Entina StoredFeedback reprezentuje spätnú väzbu užívateľa
 - `id` (UUID) – Unikátny identifikátor spätnej väzby.
 - `name` (String) – Názov spätnej väzby, napríklad vyklbený členok.
 - `severity` (Int) – Závažnosť, alebo negativita spätnej väzby. Čím vyššie číslo, tým viac negatívne ovplyvní nasledujúce tréningy.

- `forDay (Day)` – Relácia typu `M:1` ktorá reprezentuje deň pre ktorý je spätná väzba ukladaná
- `of (Training)` – Relácia `M:1` ktorá reprezentuje tréning pre ktorý je spätná väzba ukladaná
- Entita `Day` reprezentuje jeden deň používania aplikácie
 - `date (Date)` – Jedinečný identifikátor v podobe dátumu
 - `rating ([Int])` – Pole ktoré drží hodnotenia pre ten daný deň
 - `feedback (StoredFeedback)` – inverzná relácia k relácii `forDay (Day)` entity `StoredFeedback`
 - `training (Training)` – Relácia typu `1:M` reprezentujúca dni ku ktorým je tréning priradený
 - `user (User)` – Relácia `M:1`, ktorá reprezentuje užívateľa ku ktorému je tento deň priradený
- Entita `Training` reprezentujúca naplánovaný tréning
 - `assignedFinishedActivity (Int)` – Hash aktivity z `HealthKitu` ktorá reprezentuje tento naplánovaný tréning
 - `endDate (Date)` – Dátum kedy bola aktivita ukončená
 - `exercise ([Int])` – Obsahuje unikátne identifikátory cvikov ktoré majú byť pri tréningu vykonané
 - `finished (Boolean)` – Značí, či je tréning skončený
 - `id (UUID)` – Unikátny identifikátor naplánovaného tréningu
 - `name (String)` – Názov tréningu
 - `orderInDay (Int)` – Poradie tréningu v danom dni
 - `properties ([Int])` – Metadáta tréningu, ak sa jedná o bežecký tréning ukladá informácie o doporučenom tempe, vzdialenosti.
 - `startDate (Date)` – Začiatok tréningu
 - `tag (Int)` – Označenie tréningu, kde môže ísť o plánovaný tréning, ukončený tréning, alebo tréning počas ktorého bol zdolaný doterajší najlepší výkon
 - `type (Int)` – Typ tréningu, môže ísť o preťahovanie svalov, silový tréning, alebo bežecký tréning
- Entita `Goal` je cieľ užívateľa ktorý chce splniť
 - `alt (Float)` – Prevýšenie ktoré cieľový závod má
 - `cadence (Int)` – Kadencia ktorú chce užívateľ dosiahnuť počas behu
 - `completed (Boolean)` – Značí či je cieľ úspešne ukončený
 - `completionDateEnd (Date)` – Dátum ukončenia tréningu ktorý cieľ dosiahol
 - `completionDateStart (Date)` – Dátum začatia tréningu ktorý cieľ dosiahol

- `creationDate` (Date) – Dátum vytvorenia cieľa
 - `date` (Date) – Dátum kedy by mal byť cieľ splnený
 - `len` (Int) – Cieľová vzdialenosť
 - `terrain` (Int) – Typ terénu na ktorom chce užívateľ behať
 - `time` (Int) – Cieľový čas
 - `user` (User) – Relácia typu M:1, ktorá značí ku ktorému užívateľovi je cieľ priradený
- Entita User znázorňuje samotného užívateľa aplikácie
 - `birthDate` (Date) – Dátum narodenia užívateľa
 - `gender` (Int) – Pohlavie užívateľa
 - `height` (Float) – Výška užívateľa
 - `id` (UUID) – Unikátny identifikátor užívateľa
 - `image` (Binary Data) – Profilový obrázok užívateľa
 - `lastUpdateDate` (Date) – Dátum posledného prihlásenia užívateľa
 - `name` (String) – Meno užívateľa
 - `usageStart` (Date) – Kedy bol vytvorený profil
 - `weekMaxActivity` (Int) – maximálny počet tréningov ktoré chce užívateľ vykonať za jeden týždeň
 - `weight` (Float) – Výška užívateľa
 - `workoutMaxLen` (Int) – Maximálna dĺžka tréningu ktorú je užívateľ ochotný trénovať počas jednej aktivity
 - `day` (Day) – Relácia typu 1:M ktorá reprezentuje dni ktoré sú priradené k užívateľovi. Je inverzná k relácií `user` entity `Day`
 - `goal` (Goal) – Relácia typu 1:M ktorá reprezentuje ciele priradené k užívateľovi. Táto relácia je inverzná k relácií `user` entity `Goal`
 - `injury` (Injury) – Relácia typu 1:M ktorá značí zranenia ktorými užívateľ trpel, alebo trpí
 - Entita Injury reprezentuje zranenia užívateľa
 - `active` (Boolean) – Značí, či je zranenie ešte prítomné
 - `end` (Date) – Dátum kedy príznaky zranenia odzneli
 - `id` (UUID) – Unikátny identifikátor zranenia
 - `injuryId` (Int) – Identifikátor zranenia v lokálnom zozname zranení
 - `name` (String) – Názov zranenia
 - `part` (Int) – Identifikátor zranenej časti
 - `start` (Date) – Dátum vzniku zranenia
 - `treatment` ([Int]) – Pole ktoré udržiava identifikátory cvikov ktoré slúžia

na liečenie daného zranenia

- user (User) – Relácia typu M:1. Je inverzná s reláciou injury entity User a značí, ku ktorému užívateľovi je zranenie priradené



Obr. 4.8: Navrhnutý dátový model aplikácie

Kapitola 5

Implementácia a testovanie

Kapitola na začiatku predstaví niektoré zvolené technológie a spôsob implementácie zaujímavých oblastí aplikácie. Následne zoznámi so zmenami užívateľského prostredia ktoré bolo navrhnuté v protype v podkapitole 4.3. Nakoniec kapitola popisuje priebeh testovania a vyhodnotenie jeho výsledkov.

5.1 Použité technológie a postupy

Behom implementácie aplikácie boli využité knižnice na prácu s animáciami, videami a grafami, pričom ich táto podkapitola predstaví podrobnejšie.

Lottie

Lottie je knižnica pre zariadenia zo systémom iOS, Android a bola vyvinutá spoločnosťou Airbnb¹. Lottie podporuje aj React Native a slúži na vykreslenie animácií z After Effects² animácií v reálnom čase. Tieto exportované animácie potom dovoľuje vývojárom využívať rovnako ako iné statické zdroje. Rieši problém komplexných animácií na spomínaných zariadeniach, čím efektívne znižuje veľkosť aplikácie ako aj množstvo napísaného kódu. Lottie funguje na základe JSON³ súboru, z rozšírenia nad aplikáciou After Effects ktorá sa nazýva Bodymovin⁴. Okrem silnej spoločnosti ktorá Lottie spravuje, má aj širokú komunitu nadšencov na platforme Github⁵.

SwiftGif

SwiftGif⁶ je vysokovýkonná a jednoducho použiteľná knižnica na vkladanie Gif⁷ obrázkov do aplikácie pre systém iOS. Funguje ako rozšírenie komponentov z knižnice UIKit. Umožňuje jednoducho ovládať spôsob prehrávania – nekonečná slučka, prehranie iba jedenkrát. Je uspôsobená na načítanie webových zdrojov. Teda obrázky nemusia byť stiahnuté priamo v aplikácií, ale môžu byť načítané zo serveru. Okrem toho dovoľuje špecifikovať animácie

¹<http://airbnb.com>

²www.adobe.com/products/aftereffects

³JavaScriptový objektový zápis

⁴<https://exchange.adobe.com/creativecloud.details.12557.bodymovin.html>

⁵<https://github.com/airbnb/lottie-ios>

⁶<https://github.com/kirualex/SwiftyGif>

⁷grafický formát určený pre rastrovú grafiku

pri načítaní zdroja, špecifikovať kvalitu prehrávaného zdroja, alebo špecifikovať limit pamäti, ktorá môže byť využitá. Tabuľka 5.1 porovnáva knižnicu SwiftyGif s jej najbližšou konkurenciou. Ako je z tabuľky vidieť knižnica využije procesor o niekoľko desiatok percent menej, avšak obetuje využitie pamäti zariadenia.

	Priemerné využitie CPU	Priemerné využitie pamäti
FLAnimatedImage	35%	9,5Mb
SwiftyGif	2%	18,4Mb
SwiftyGif(memoryLimit:10)	34%	9,5Mb

Tabuľka 5.1: Tabuľka popisujúca rozdiel medzi zvolenou knižnicou a konkurenčnou knižnicou

SwiftUICharts

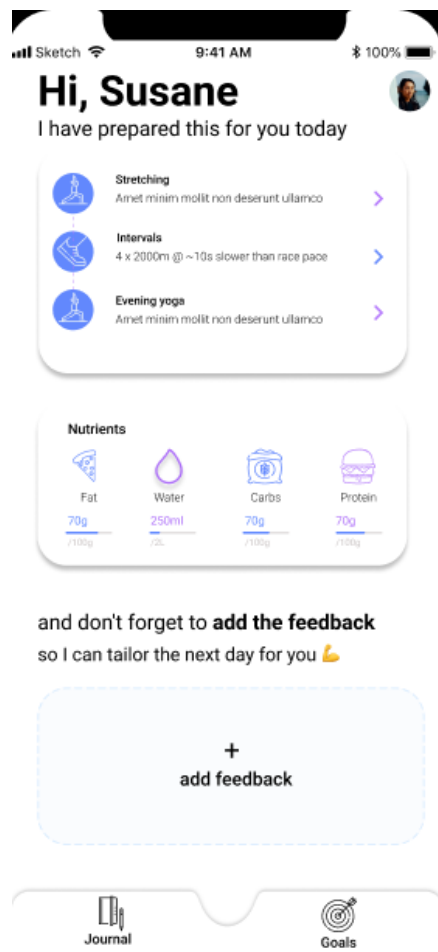
Knižnica na vykresľovanie grafov pre framework SwiftUI. Podporuje čiarový, koláčový, alebo stĺpcový graf. Umožňuje veľkú variabilitu vlastností grafov. Knižnica má v súčasnosti problém s efektívnosťou, ale momentálne sa jedná o jedinú knižnicu dostupnú pre SwiftUI. Tento problém bude vyriešený publikáciou novej verzie, ktorá by mala prísť v priebehu niekoľkých týždňov⁸.

5.2 Výsledné užívateľské prostredie

Užívateľské prostredie vychádza z prototypu uvedeného v podkapitole 4.3. Počas implementácie, prakticky všetky obrazovky aplikácie zostali bez zmien. Jediná zmena nastala pri úvodnej časti obrazovky aplikácie. Pôvodný stav je možné vidieť na obrázku 4.3. Zmenený stav je zobrazený na obrázku 5.1.

Ku zmene užívateľského prostredia tejto obrazovky došlo z dôvodu spätnej väzby užívateľov. Pri implementácii pôvodného konceptu sa nenačítavali obrázky, prostredie sa spomalovalo. Nebola žiadna možnosť rozoznať ukončený a naplánovaný tréning. Rovnako ku zmene došlo k informáciám o nutričných hodnotách. Pôvodný návrh sa užívateľom v praxi nepozdával. Vznikali problémy so zistením presných doporučených hodnôt. Všeobecne úvodná obrazovka bola z užívateľského hľadiska menej prehľadná a viac rušivá ako zvyšok aplikácie. Zmena teda prebehla tak, aby bola zapracovaná každá relevantná spätná väzba. Výsledné prostredie je prehľadnejšie a viac zapadá do témy aplikácie.

⁸<https://github.com/AppPear/ChartViewV2Demo>



Obr. 5.1: Zmenená úvodná obrazovka

5.3 Spôsob implementácie

Táto podkapitola predstaví spôsob implementácie niektorých častí aplikácie, pričom tiež popíše dôvod voľby spôsobu tejto implementácie.

Použitie UIViewControllerov vo SwiftUI

SwiftUI je stále veľmi mladý framework, z toho dôvodu ešte stále nie je taký flexibilný ako jeho predchodca UIKit. Je veľké množstvo vecí ktoré sa vo SwiftUI nedajú spraviť a preto je nutné mať možnosť použiť funkcionality a bloky z frameworku UIKit. V tomto prípade sa jednalo napríklad o implementáciu zobrazenia mapy. Našťastie sa prvky z UIKit frameworku dajú obaliť pre použitie vo SwiftUI. Pred začatím procesu previazania prvkov z pôvodného frameworku do nového frameworku je nutné vedieť tri základné veci. UIKit obsahuje triedu ktorá sa nazýva `UIView` a reprezentuje rodičovskú triedu všetkých komponentov – text, tlačítka, formulár a ďalšie. `UIViewController` je druhá trieda, ktorá obsahuje všetok kód nutný na fungovanie jednotlivých komponentov. UIKit používa návrhový vzor delegácie. To znamená, že existuje samostatná trieda ktorá rozhoduje o tom, ako by sa mali jednotlivé komponenty chovať, napríklad pri zmene hodnoty. Avšak pri použití frameworku

SwiftUI sa kód musí štrukturovať odlišne. Veľkým rozdielom je tiež použitie štruktúr namiesto tried, rovnako aj to, že triedy `UIView` a `UIViewController` kombinuje do jedného protokolu. Pre obalenie komponentu pochádzajúceho z `UIKit` je nutné vytvoriť štruktúru, ktorá bude vyhovovať `UIViewControllerRepresentable`, to znamená, že treba implementovať dve metódy – `makeUIViewController()` a `updateUIViewController()` ktoré nám dovoľia reagovať na zmenu stavu štruktúry[17]. Po splnení týchto krokov si v aplikácii už môžeme zobraziť samotný komponent, avšak pri väčšine komponentov toto nie je dostačujúci stav. Preto sa musí implementovať trieda zodpovedajúca triede `Coordinator` a metóda `makeCoordinator()` ktorá vracia implementovaný koordinátor. Koordinátor má vo frameworku podobnú úlohu ako delegát vo frameworku `UIKit`. Úryvok kódu aplikácie 5.1 ukazuje spôsob previazania komponentu `MKMapView`, ktorý slúži na zobrazenie mapy v aplikácii. Úryvok je skrátenej o metódu `updateUIViewController()`.

```

1 struct MapView: UIViewRepresentable {
2     @Environment(\.managedObjectContext) var moc
3     @EnvironmentObject var mainViewModel: MainViewModel
4     class Coordinator: NSObject, MKMapViewDelegate { // delegat pre mapView
5         var parent: MapView
6
7         init(parent: MapView) {
8             self.parent = parent
9         }
10
11        func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer {
12            let renderer = MKPolylineRenderer(overlay: overlay)
13            renderer.strokeColor = UIColor.orange
14            renderer.lineWidth = 3
15            return renderer
16        }
17    }
18
19    func makeCoordinator() -> MapView.Coordinator {
20        Coordinator(parent:self)
21    }
22
23    func makeUIView(context: Context) -> MKMapView {
24        let mapView = MKMapView()
25        mapView.delegate = context.coordinator
26
27        let route = self.mainViewModel.route
28        if let thisWorkout = route.first{
29            let locations = thisWorkout.locations
30
31            let polyline = MKPolyline(coordinates: locations, count: locations.count) // nakreslime
32                ciaru medzi koordinatmi
33            mapView.addOverlay(polyline, level: .aboveRoads)
34            return mapView
35        }
36        return mapView
37    }
38
39    func updateUIView(_ uiView: MKMapView, context: Context) {
40        ...

```

Výpis 5.1: Ukážka obalenia `UIViewController` pre použitie vo SwiftUI

Získanie dát o rýchlosti a kadencií zo systémovej knižnice `HealthKit`

Systémová knižnica `HealthKit` bola uvedená v podkapitole 3.5. Samotný `HealthKit` neponúka možnosť získať informácie o rýchlosti a kadencií užívateľa z dát o jeho tréningu za pomoci dotazu. Na získanie týchto dát je nutné uskutočniť viacero dotazov, v ktorých je nutné manuálne vykonať štatistické výpočty. Prvý dotaz ktorý musí smerovať na knižnicu `HealthKit` je `HKSampleQuery` na všetky bežecké tréningy. Potom môžeme smerovať druhý dotaz, so špecifikáciou konkrétneho tréningu. Tento dotaz je typu `HKSampleQuery` na zaznamenané zmeny lokácie počas tréningu. Každá zaznamenaná zmena má pridelenú

prejdenú vzdialenosť od poslednej zmeny a pridelenú časovú pečiatku. Postupným prejdenním týchto záznamov dokážeme zistiť rýchlosť užívateľa a jeho kadenciu. Na obrázku 5.2 je ukážka výpočtu rýchlosti užívateľa počas tréningu.

```

1   func getPaceAndCadence(for workout: HKWorkout, handler: @escaping ([Double], ...) -> Void){
2       guard let healthStore = self.healthStore else { return }
3       // ake data chceme ziskat
4       let distanceType = HKObjectType.quantityType(forIdentifier:
5           HKQuantityTypeIdentifier.distanceWalkingRunning)
6       let workoutPredicate = HKQuery.predicateForObjects(from: workout)
7       let startDateSort = NSSortDescriptor(key: HKSampleSortIdentifierStartDate, ascending: true)
8       getCadence(for: workout){ cadenceArray, avgCadence, minCadence, maxCadence in
9           let query = HKSampleQuery(sampleType: distanceType!, predicate: workoutPredicate, limit:
10              HKObjectQueryNoLimit, sortDescriptors: [startDateSort]) { (query, samples, error) in
11              if error != nil {return}
12              if let distance = samples as? [HKQuantitySample] {
13                  //mozeme vypocitat tempo
14                  var count = 0.0, drift = 0.0, driftTime = 0.0, time = 0.0, totalTime = 0.0,
15                      totalDistance = 0.0, avgPace = 0.0, minPace = 0.0, maxPace = 0.0
16                  var counter = 0
17                  var pace = [Double]()
18                  var smooth = 0
19                  var lapData = [Double: Double]()
20                  distance.forEach{sample in
21                      if ((count + sample.quantity.doubleValue(for: HKUnit.meter())) > 1000){
22                          drift = count + sample.quantity.doubleValue(for: HKUnit.meter()) - 1000
23                          count = 1000
24                      }
25                      if drift > 0.001 {
26                          // vypocitame cas
27                          driftTime = drift / (sample.quantity.doubleValue(for: HKUnit.meter()) /
28                              sample.endDate.timeIntervalSince(sample.startDate))
29                          time = time + (sample.endDate.timeIntervalSince(sample.startDate) -
30                              driftTime)
31
32                          totalTime += time
33                          totalDistance += count
34                          if count == 1000 { // ulozime udaje ku kolu
35                              lapData[count - Double(counter)] = time
36                          }else {
37                              lapData[1000 + count] = time
38                          }
39
40                          count = drift
41                          time = driftTime
42                          drift = 0
43                          driftTime = 0
44                          counter += 1
45                      }
46                      count += sample.quantity.doubleValue(for: HKUnit.meter())
47                      time += sample.endDate.timeIntervalSince(sample.startDate)
48                      ...
49                      let total = pace.reduce(0,+)
50                      avgPace = total / Double(pace.count)
51                      totalTime += time - Double(counter * 2)
52                      totalDistance += count - Double(counter * 8)
53                      handler(cadenceArray, avgCadence, minCadence, maxCadence, pace, 60 / avgPace,
54                          minPace, maxPace, lapData)
55                  }
56              }
57              healthStore.execute(query)
58          }
59      }
60  }

```

Výpis 5.2: Ukážka výpočtu rýchlosti užívateľa z tréningových dát

Výpočet doporučených nutričných hodnôt

Užívateľovi je každý deň doporučený určitý nutričný príjem. Ten je počítaný na základe znalostí o ňom – výška, váha, pohlavie a vek. Ďalej sa do výpočtu zaradia všetky cviky ktoré má na daný deň užívateľ doporučené. Najrprv sa vypočíta potrebný kalorický príjem a to tak, že sa spočíta odporúčaný kalorický príjem pomocou metódy Mifflin-St Jeor,

metódy Katch-McArdle a metódy Harris-Benedict, informácie k týmto metódam boli čerpané z publikácie[10]. Z týchto hodnôt sa urobí priemer. K tomuto priemeru sa pripočítajú odhadované spálené kalórie z plánovaných tréningov. Tieto hodnoty sa odhadujú pomocou metabolických ekvivalentov. Úryvok kódu z aplikácie 5.3 ukazuje ako tento výpočet prebiehal.

```

1  func calculateCaloriesForUser() -> Double{
2      var age = 0.0
3      if let birthDate = self.userDataManager.localUser?.birthDate {
4          age = (Date().timeIntervalSince(birthDate) / 60 / 60 / 24 / 365 )
5      }else{
6          age = (Date().timeIntervalSince(Calendar.current.date(byAdding: .year, value: -22, to:
7              Date())!) / 60 / 60 / 24 / 365 )
8      }
9      let height = Double(self.localUser.height)
10     let weight = Double(self.localUser.weight)
11     var result = 0.0
12     if localUser.gender == 0{ // muz
13         let helper = 10 * weight + 6.25 * height
14         let MSJE = helper - 5 * age + 5
15         let other = 13.397 * weight + 4.799 * height
16         let RHBE = other - 5.677 * age + 88.362
17
18         result = (MSJE + RHBE) / 2
19     }else{ // zena
20         let helper = 10 * weight + 6.25 * height
21         let MSJE = helper - 5 * age - 161
22         let other = 9.247 * weight + 3.098 * height
23         let RHBE = other - 4.330 * age + 447.593
24
25         result = (MSJE + RHBE) / 2
26     }
27
28     var MET = 1.0
29     var exerciseTotalCaloriesEstimate = 0
30     for workout in self.todaysWorkouts{
31         var duration = 0.0
32         if workout.type == .easy {
33             duration = Double(Double(workout.duration.min() ?? 0) / 1000.0 * 60.0)
34             MET = 10
35         }else if workout.type == .cross {
36             duration = 35 * 60 // priblizne 30 minut pre cross trening
37             MET = 4.0
38         }else if workout.type == .dynamicStretching || workout.type == .staticStretching {
39             duration = 20 * 60 // 20 minut pre dynamicky a staticky strecing
40             MET = 3.0
41         }else {
42             duration = Double(Double(workout.duration.min() ?? 0) / 1000 * 60)
43             MET = 13.0
44         }
45         exerciseTotalCaloriesEstimate += Int(duration * (MET * 3.5 * weight)) / (200*60)
46     }
47     return result + Double(exerciseTotalCaloriesEstimate)
48 }

```

Výpis 5.3: Ukážka výpočtu nutričných hodnôt

Následne sa kalórie rozdelili medzi proteín (17 percent), karbohydráty (53 percent) a tuky (30 percent) v pomere vhodnom pre bežcov.

5.4 Testovanie

Testovanie sa stalo veľmi dôležitým aspektom vývoja aplikácie a to ešte pred začiatkom samotnej implementácie. Otestovať grafické rozhranie ešte pred začatím implementácie bolo kľúčovým aspektom zabezpečenia konzistentnosti grafického rozhrania. Neskôr mnohonásobne prebiehalo testovanie jednotlivých funkcií aplikácie. Testy užívateľského rozhrania prebiehali pomocou sady nástrojov od firmy Figma, testy funkčných blokov aplikácie prebiehali na niekoľkých mobilných zariadeniach iPhone. Neskôr po overení funkcionality, bola

aplikácia testovaná na skupine reálnych bežcov. V tejto sekcii sú popísané jednotlivé časti testovania a ich vyhodnotenie.

Testovanie užívateľského rozhrania

Ešte pred začiatkom implementácie bolo nutné sa uistiť, že užívateľské rozhranie je jednoducho pochopiteľné, nie je rušivé a bude obsahovať všetky informácie ktoré by od podobnej aplikácie mohli užívatelia očakávať. Na začiatku testovania boli pripravené dva základné prototypy aplikácie. Oba boli spracované v nástroji Figma⁹, ktorý umožňuje jednotlivé bloky užívateľského rozhrania previazať aj funkčne. Prototyp sa následne preniesol do mobilného zariadenia, pomocou nástroja Mirror¹⁰. Testovanie v tejto fáze prebiehalo na siedmich užívateľoch, pričom z tejto skupiny boli aktívni bežci traja. Každý užívateľ dostal za úlohu:

1. Vytvoriť si profil bežca
2. Prezrieť si tréningový plán na nadchádzajúci týždeň
3. Nájsť ukončené tréningy
4. Prezrieť si ukončený tréning
5. Nájsť informácie k dnešnému tréningu
6. Zadať spätnú väzbu k tréningu
7. Zmeniť bežecký cieľ

Po zadaní úloh a spustenia prototypu boli jednotliví užívatelia pozorovaní, pričom boli vyzvaní k tomu, aby svoje myšlienky a pocity prezentovali nahlas. Po ukončení úlohy bolo testovanie prerušené, za účelom ohodnotenia tej danej časti testovania. Po zaznamenaní hodnotenia sa pokračovalo v testovaní. Výsledky testovania sú prezentované v tabuľke 5.2, pričom každá úloha dostala určitý počet bodov v jednej z troch kategórií. Kategórie boli -vzhlad, jednoduchosť použitia a kompletnosť očakávaných funkcií. Maximálny počet bodov v každej kategórii je 10.

Meno	1. úloha	2. úloha	3. úloha	4. úloha	5.úloha	6. úloha	7. úloha
Beáta	10/10/8	8/6/7	9/5/9	8/9/8	10/10/10	9/9/9	10/10/10
Martina	9/9/9	6/6/6	10/7/9	10/9/9	10/10/10	10/9/9	10/10/10
Braňo	10/10/9	9/8/8	10/8/10	10/10/10	9/10/10	9/8/9	10/10/10
Juraj	7/9/10	5/7/7	8/6/9	8/8/8	9/9/9	10/10/9	10/10/10
Samuel	9/10/10	9/9/10	8/8/9	9/10/10	10/10/9	9/8/8	10/10/10
Amália	10/8/9	10/7/9	10/6/10	10/10/9	10/10/10	8/9/9	10/10/10
Jozef	8/8/9	8/8/8	8/6/8	9/9/10	9/7/7	9/7/7	10/10/10

Tabuľka 5.2: Tabuľka s hodnotením testovania užívateľského rozhrania

Okrem ohodnotenia bola spätná väzba zozbieraná aj z pocitov a myšlienok ktoré užívatelia prezentovali. Z nich vyplynulo niekoľko nedostatkov, medzi ktoré najviac patrila

⁹<https://www.figma.com/>

¹⁰<https://www.figma.com/mirror>

absencia vyznačenia ukončeného tréningu – užívatelia neboli schopní rozoznať naplánovaný tréning od ukončeného. Niektorým užívateľom vadila absencia štatistík v prehľade ukončeného bežeckého tréningu. Okrem uvedených boli spomenuté aj iné menšie grafické nedostatky.

Testovanie funkčných blokov aplikácie

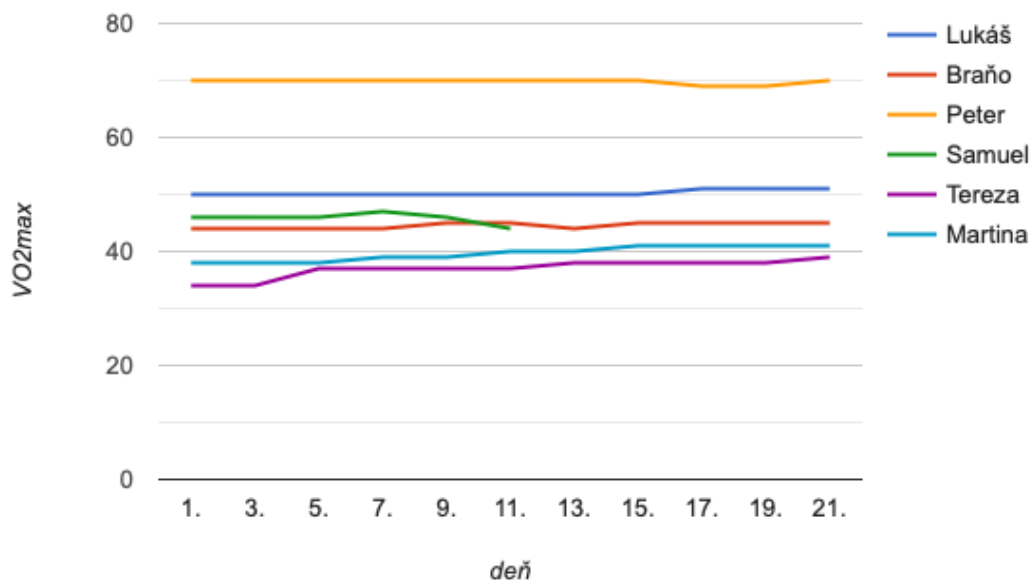
Počas implementácie boli na testovanie využívané iba mobilné zariadenia iPhone, konkrétne sa jednalo o iPhone 7, Xr a 11Pro s verziou systému iOS 14.2 a 14.3. Týmto spôsobom bola postupne otestovaná funkčnosť a vzhľad aplikácie, pričom zariadenia boli vybrané tak, aby disponovali rôznou veľkosťou obrazoviek. Počas testovania bol špecificky vynechaný simulátor iOS zariadení, z dôvodu nemožnosti správne testovať požadovanú funkcionálnosť. Pri objavení chýb bolo použité vývojové prostredie XCode¹¹. Toto prostredie umožňuje nastaviť na špecifickú funkciu takzvaný breakpoint, teda zastaví aplikáciu v danom okamihu a zobrazí stav aktuálnych premenných, spolu s ich hodnotou v pamäti. Pri riešení fatálnych chýb je k dispozícii spätná väzba, pričom nie vždy je poskytnutá v zrozumiteľnej forme. Na zobrazenie podrobnej správy sa využíva špeciálna funkcia¹².

Testovanie na užívateľoch

Po testovaní užívateľského rozhrania a po dokončení implementácie jednotlivých blokov, bola aplikácia testovaná ako celok. Test prebiehal počas troch týždňov, pričom aplikácia bola testovaná na skupine šiestich ľudí. Traja pochádzajú z pôvodnej skupiny bežcov, ktorí boli zaradení do testovania užívateľského rozhrania a traja sú noví užívatelia, ktorí aplikáciu ešte nevideli. Cieľom testovania bolo zistiť stabilitu aplikácie a či v reálnom prostredí funguje funkcionálnosť generovania tréningov. Rovnako boli užívatelia požiadaní, aby v priebehu testovania dbali na zadávanie spätnej väzby a dodržiavanie tréningov, ktoré im boli vygenerované. Po testovaní s každým užívateľom prebehla diskusia o aplikácii a boli skontrolované dáta o hodnote $\dot{V}O_2max$ a o poklese srdcového tepu ihneď po ukončení tréningu. Zaznamenaný rozdiel v hodnotách $\dot{V}O_2max$ je zaznamenaný na obrázku 5.2. Konkrétne hodnoty sú zaznamenané v tabuľke 5.3. Vyššie čísla indikujú lepší výsledok.

¹¹<https://developer.apple.com/xcode/>

¹²<https://www.natashatherobot.com/xcode-debugging-trick/>



Obr. 5.2: Graf znázorňujúci namerané hodnoty $\dot{V}O_2max$ počas doby 21 dní

	1. deň	3. deň	5. deň	7. deň	9. deň	11. deň	13. deň	15. deň	17. deň	19. deň	21. deň
Lukáš	50	50	50.1	50.1	50.1	50.6	50.6	50.6	51	51	51.1
Braňo	44	44.6	44.6	44.6	45.1	45	44.9	45.2	45.2	45.1	45.2
Peter	70.1	70.1	70.1	70	70.1	70.1	70	70.2	69.9	69.9	70.0
Samuel	46.2	46.4	46.9	47	46.4	44.1					
Tereza	34.45	34.49	37.68	37.77	37.58	37.81	38.91	38.95	38.51	38.93	39.02
Martina	38.7	38.7	38.8	39.1	39.5	40.1	40.7	41.3	41.6	41.7	41.7

Tabuľka 5.3: Tabuľka zobrazujúca namerané hodnoty $\dot{V}O_2max$ počas doby testovania

Zo šiestich sledovaných subjektov sa štyrom hodnota $\dot{V}O_2max$ mierne zvýšila. Subjekt (Samuel) u ktorého došlo k zníženiu hodnoty $\dot{V}O_2max$ počas testovania musel testovanie prerušiť, nakoľko bol infikovaný vírusovým ochorením. Toto ochorenie vysvetľuje veľmi zníženú hodnotu $\dot{V}O_2max$ v deň posledného uskutočneného merania. Tento subjekt nebol ďalej monitorovaný. Druhý subjekt (Peter) nezaznamenal takmer žiadne zlepšenie ani zhoršenie sledovanej hodnoty. Táto skutočnosť sa prisudzuje relatívne veľkej trénovanosti subjektu. Pri vysokých hodnotách $\dot{V}O_2max$, už nie je troj-týždňová fyzická stimulácia dostačujúca. Ostatné sledované subjekty reagovali na vygenerovaný tréning priaznivo, pričom priemerná zmena hodnoty $\dot{V}O_2max$ činí zvýšenie o $1.96 \text{ min}^{-1} \cdot \text{kg}^{-1}$.

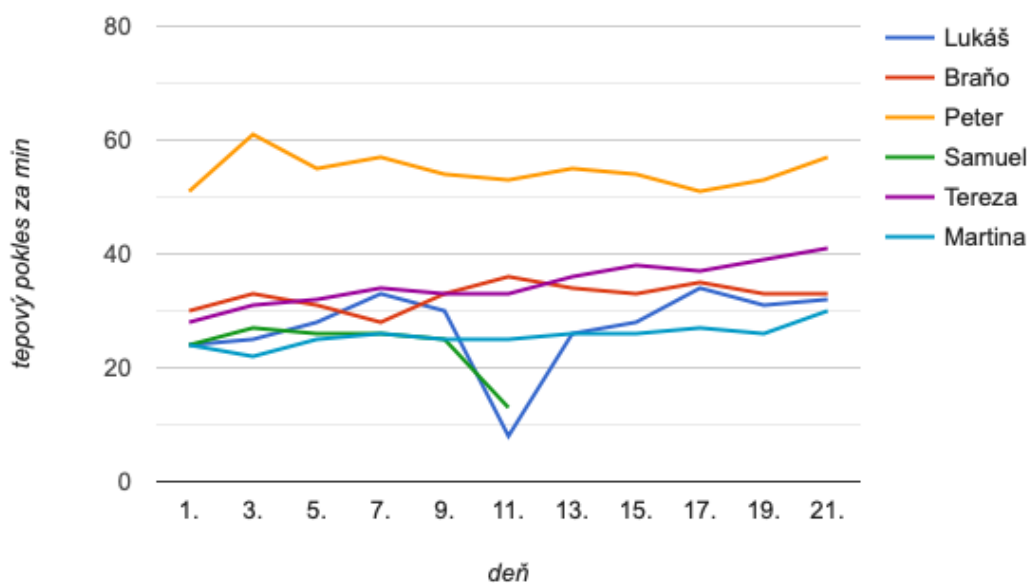
Druhý spôsob ktorým prebiehalo testovanie zlepšenia výkonnosti jednotlivých subjektov je sledovanie regenerácie srdcového tepu okamžite po ukončení aktivity. Merania prebiehali pomocou zariadenia Apple Watch. Pozorované hodnoty boli namerané počas prvých dvoch minút po ukončení aktivity. V tabuľke 5.4 sú zaznamenané hodnoty poklesu srdcového rytmu za minútu. V tabuľke 5.5 sú uvedené spriemerované hodnoty z tabuľky 5.4. Tieto spriemerované hodnoty boli následne vynesené do grafu, ktorý je vidieť na obrázku 5.3. Meranie bolo uskutočnené po každom bežekom tréningu. Vyššie čísla indikujú lepší výsledok.

	1. deň	3. deň	5. deň	7. deň	9. deň	11. deň	13. deň	15. deň	17. deň	19. deň	21. deň
Lukáš	22 / 26	31 / 19	34 / 23	37 / 30	29 / 31	11 / 6	32 / 21	27 / 30	37 / 31	35 / 28	29 / 35
Braňo	32 / 28	30 / 37	29 / 33	28 / 29	30 / 36	37 / 35	34 / 34	32 / 34	36 / 34	31 / 36	35 / 36
Peter	48 / 55	60 / 62	59 / 52	58 / 57	56 / 52	50 / 56	53 / 57	56 / 53	49 / 53	55 / 52	55 / 60
Samuel	24 / 25	26 / 28	29 / 23	27 / 25	26 / 24	15 / 11					
Tereza	26 / 31	30 / 32	30 / 35	37 / 32	29 / 37	34 / 33	36 / 37	39 / 37	30 / 45	41 / 38	44 / 39
Martina	23 / 25	23 / 21	26 / 24	25 / 27	25 / 26	24 / 27	25 / 28	25 / 27	29 / 25	27 / 26	30 / 31

Tabuľka 5.4: Tabuľka ukazujúca nameraný pokles srdcového tepu – prvá minúta / druhá minúta

	1. deň	3. deň	5. deň	7. deň	9. deň	11. deň	13. deň	15. deň	17. deň	19. deň	21. deň
Lukáš	24	25	28.5	33.5	30	8.5	26.5	28.5	34	31.5	32
Braňo	30	33.5	31	28.5	33	36	34	33	35	33.5	33.5
Peter	51.5	61	55.5	57.5	54	53	55	54.5	51	53.5	57.5
Samuel	24.5	27	26	26	25	13					
Tereza	28.5	31	32.5	34.5	33	33.5	36.5	38	37.5	39.5	41.5
Martina	24	22	25	26	25.5	25.5	26.5	26	27	26.5	30.5

Tabuľka 5.5: Tabuľka s priemerným poklesom srdcového tepu v daný deň



Obr. 5.3: Graf znázorňujúci zaznamenaný pokles srdcového tepu za jednotlivé dni

Ani jeden pozorovaný subjekt sa počas sledovanej doby nezranil. Aplikáciu užívatelia hodnotili aj zo svojich skúseností, pričom niektorým užívateľom sa nepáčilo anglické užívateľské prostredie a pomalá odozva pri prehlade ukončeného tréningu. Celkovo však užívatelia hodnotia samotnú aplikáciu kladne, pričom sa im páči hlavne zapracovanie spätnej väzby.

Z pozorování subjektov a merania uvedených hodnôt vyplýva, že použitie mobilnej aplikácie ktorá je predmetom tejto práce, má pozitívny vplyv na výkonnosť koncových užívateľov.

Kapitola 6

Záver

Cieľom bakalárskej práce bolo vytvoriť aplikáciu na vedenie atletických tréningov so špecifickým zameraním na beh. Počas vývoja bola aplikácia testovaná na cieľovej skupine užívateľov. Na základne zozbieraných podnetov bola aplikácia doladovaná a vylepšovaná. Dôvodom bolo vytvorenie jednoduchého a zrozumiteľného prostredia.

Pre správny návrh aplikácie bolo najskôr nutné naštudovať už existujúce riešenia, ktoré sa zaoberali podobnou problematikou. Neskôr bolo dôležité preskúmať rôzne tréningové metodiky a vybrať z nich tú najvhodnejšiu. Mimo skúmaných metodík bol kladený dôraz na prevenciu zranení. Z tohto dôvodu bol navrhnutý systém spätnej väzby, ktorý reflektuje pocity užívateľa. Posledným krokom pred začatím implementácie bolo preskúmanie jednotlivých techník prevencii zranenia a zozbieranie zodpovedajúcich tréningových cvikov. Informácie pre úspešnú implementáciu aplikácie boli získané z dokumentácie k jazyku Swift, SwiftUI a konzultáciami s vedúcim bakalárskej práce.

Testovanie prebiehalo na menšej skupine užívateľov, ktorí spadajú do cieľovej kategórie. Podnety na zmenu grafického prostredia boli zvážené a zapracované. Z praktického hľadiska boli porovnané hodnoty $\dot{V}O_2max$ pred začatím testovania a po jeho ukončení. Tiež boli porovnané dáta o poklese srdcového tepu bezprostredne po tréningu. Aplikácia funguje spoľahlivo, avšak stále necháva priestor na budúce rozšírenia. Po publikácii v obchode s aplikáciami mám v pláne vytvoriť konverzačného robota, ktorý bude slúžiť na priame každodenné vedenie užívateľa. Rovnako plánujem rozšíriť databázu tréningových cvikov, ktorá už teraz obsahuje viac ako sto variant.

Literatúra

- [1] ACTIVEPT. *RUN FASTER & MORE EFFICIENTLY WITH A DYNAMIC WARM-UP* [online]. 2020 [cit. 2021-01-04]. Dostupné z: <https://www.activeptandsports.com/dynamic-warm-ups/>.
- [2] ASSOCIATION, A. H. *Target Heart Rates Chart* [online]. 2021 [cit. 2021-30-03]. Dostupné z: <https://www.heart.org/en/healthy-living/fitness/fitness-basics/target-heart-rates>.
- [3] BEATTIE, K., CARSON, B., LYONS, M., ROSSITER, A. a KENNY, I. The Effect of Strength Training on Performance Indicators in Distance Runners. *Journal of Strength and Conditioning Research*. 1. vyd. Apríl 2016, zv. 31, s. 1. DOI: 10.1519/JSC.0000000000001464.
- [4] BONACCI, J., GREEN, D., SAUNDERS, P. U., BLANCH, P., FRANETTOVICH, M. et al. Change in running kinematics after cycling are related to alterations in running economy in triathletes. *Journal of Science and Medicine in Sport*. Elsevier BV. júl 2010, zv. 13, č. 4, s. 460–464. DOI: 10.1016/j.jsams.2010.02.002. Dostupné z: <https://doi.org/10.1016/j.jsams.2010.02.002>.
- [5] CUCUZZELLA, M. *Run for your life*. New York: Alfred A. Knopf, 2018. ISBN 978-1101946305.
- [6] DANIELS, J. *Daniels' running formula*. Champaign, IL: Human Kinetics, 2014. ISBN 9781450431835.
- [7] DAVIES, P. a MACONOCHE, I. The relationship between body temperature, heart rate and respiratory rate in children. *Emergency Medicine Journal*. 1. vyd. BMJ. aug 2009, zv. 26, č. 9, s. 641–643. DOI: 10.1136/emj.2008.061598. Dostupné z: <https://doi.org/10.1136/emj.2008.061598>.
- [8] ELENA. *Swift vs. Objective-C: What language to Choose in 2021?* [online]. 2021 [cit. 2021-02-04]. Dostupné z: <https://www.activeptandsports.com/dynamic-warm-ups/>.
- [9] FREY, M. *How to Use Recovery Heart Rate for Fitness and Health* [online]. Wiki FIT VUT v Brně, 2021 [cit. 2021-04-17]. Dostupné z: <https://www.verywellfit.com/what-is-recovery-heart-rate-3495557>.
- [10] GUPTA, R., JOSEPH, M., PREMA, L., INBAKUMARI, M. a THOMAS, N. Are predictive equations for estimating resting energy expenditure accurate in Asian Indian male weightlifters? *Indian Journal of Endocrinology and Metabolism*. Medknow. 2017, zv. 21, č. 4, s. 515. DOI: 10.4103/ijem.ijem_563_16. Dostupné z: https://doi.org/10.4103/ijem.ijem_563_16.

- [11] HUTCHESON, A. *Cross-Training: Should you Take a Swing at Boxing?* [online]. 2018 [cit. 2021-01-04]. Dostupné z: <https://www.activeptandsports.com/dynamic-warm-ups/>.
- [12] INC, A. *Core Data Programming Guide - What Is Core Data?* [online]. 2018 [cit. 2021-01-04]. Dostupné z: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/index.html#//apple_ref/doc/uid/TP40001075-CH2-SW1.
- [13] INC, A. *Managed object context* [online]. 2018 [cit. 2021-12-04]. Dostupné z: https://developer.apple.com/library/archive/documentation/DataManagement/Devpedia-CoreData/managedObjectContext.html#//apple_ref/doc/uid/TP40010398-CH24-SW1.
- [14] INC, A. *Persistent store coordinator* [online]. 2018 [cit. 2021-23-04]. Dostupné z: https://developer.apple.com/library/archive/documentation/DataManagement/Devpedia-CoreData/persistentStoreCoordinator.html#//apple_ref/doc/uid/TP40010398-CH27-SW1.
- [15] INC, A. *Authorizing Access to Health Data* [online]. 2021 [cit. 2021-25-04]. Dostupné z: https://developer.apple.com/documentation/healthkit/authorizing_access_to_health_data.
- [16] INC, A. *CloudKit* [online]. 2021 [cit. 2021-25-04]. Dostupné z: <https://developer.apple.com/icloud/cloudkit/>.
- [17] INC, A. *Interfacing with UIKit* [online]. 2021 [cit. 2021-13-04]. Dostupné z: <https://developer.apple.com/tutorials/swiftui/interfacing-with-uikit>.
- [18] INC, A. *MKPlacemark* [online]. 2021 [cit. 2021-12-04]. Dostupné z: <https://developer.apple.com/documentation/mapkit/mkplacemark>.
- [19] MANAGEMENT, U. O. O. P. *Physical Ability Tests* [online]. 2019 [cit. 2021-31-03]. Dostupné z: <https://www.opm.gov/policy-data-oversight/assessment-and-selection/other-assessment-methods/physical-ability-tests/>.
- [20] MGR. IGOR DUVAČ, a. k. a. *Výzkum ve sportovním tréninku II*. 1. vyd. Masarykova univerzita, 2013. ISBN 978-80-210-6278-8.
- [21] NAUMOV, A. *Clean Architecture for SwiftUI* [online]. 2019 [cit. 2021-01-04]. Dostupné z: <https://nalexn.github.io/clean-architecture-swiftui/>.
- [22] PATRZYK, J., PATRZYK, B., RYCERZ, K. a BUBAK, M. Towards A Novel Environment For Simulation Of Quantum Computing. *Computer Science*. 1. vyd. Január 2015, zv. 16, č. 1, s. 103. DOI: 10.7494/csci.2015.16.1.103.
- [23] SAVAGE, J. *A comparison of the best marathon training plans* [online]. 2016 [cit. 2021-27-03]. Dostupné z: https://fellrnr.com/wiki/A_Comparison_of_Marathon_Training_Plans.
- [24] SAVAGE, J. *Hanson's Marathon Method (second edition)* [online]. 2019 [cit. 2021-30-03]. Dostupné z: <https://fellrnr.com/wiki/Hanson>.

Príloha A

Obsah priloženého média

xschme00-root	
├── xschme00-thesis.pdf	Text bakalárskej práce
├── Source	Súbory aplikácie
│ ├── README.md	Inštrukcie
│ ├── RunSmartBP	Zdrojový kód aplikácie
│ └── RunSmartBP.xcodeproj	Xcode projekt aplikácie
└── Thesis	Zdrojový kód textu bakalárskej práce