

# Mobilní aplikace pro podporu osobního růstu

Bc. Radovan Račák

---

Diplomová práce  
2020



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Radovan Račák**  
Osobní číslo: **A16230**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **Kombinovaná**  
Téma práce: **Mobilní aplikace pro podporu osobního růstu**  
Téma práce anglicky: **A Mobile Application to Support Personal Growth**

### Zásady pro vypracování

1. Prostudujte aktuální nástroje pro tvorbu mobilní aplikace sestavitelné pro obě hlavní platformy Android a iOS.
2. Sestavte funkční a nefunkční požadavky, navrhnete jednotlivé obrazovky, případy užití a způsoby interakce pro mobilní aplikaci na podporu osobního růstu.
3. Aplikace bude uživateli nabízet několik vybraných psychologických testů – zaměřte se na vhodný způsob uložení dat jednotlivých testů a způsob jejich zobrazení v rámci uživatelského rozhraní.
4. Dle návrhu aplikaci prakticky implementujte, otestujte na mobilním zařízení a popište možnosti sestavení pro obě zmíněné platformy.
5. Popište možné způsoby rozšíření aplikace o další sady testů pomocí standardní aktualizace aplikace, nebo přes síť, uveďte možné způsoby monetizace.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Ionic Framework. [online]. 2019. Dostupné na internetu: <https://ionicframework.com/docs>
2. Angular. [online]. 2019. Dostupné na internetu: <https://angular.io/docs>
3. Design & Quality. [online]. 2019. Dostupné na internetu: <https://developer.android.com/design>
4. Singh, I. 2018. Ionic Cookbook: Recipes to create cutting-edge, real-time hybrid mobile apps with Ionic. 3. vyd. Packt Publishing, 2018. 390s. ISBN 978-1788623230
5. Griffith, Ch. 2017. Mobile App Development with Ionic, Revised Edition: Cross-Platform Apps with Ionic, Angular, and Cordova. O'Reilly Media, 2017. 292s. ISBN 978-1491998120.
6. Griffiths, D. 2017. Head First Android Development: A Brain-Friendly Guide. 2. vyd. O'Reilly Media, 2017. 928s. ISBN 978-1491974056
7. Menendez Ph.D., Diane S., Williams Ed.D., P. 2015. Becoming a Professional Life Coach: Lessons from the Institute of Life Coach Training. 2. vyd. W. W. Norton & Company, 2015. 480s. ISBN 978-0393708363

Vedoucí diplomové práce:

**Ing. Radek Vala, Ph.D.**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: 28. listopadu 2019  
Termin odevzdání diplomové práce: 15. května 2020



---

**doc. Mgr. Milan Adámek, Ph.D.**  
děkan

---

**prof. Mgr. Roman Jašek, Ph.D.**  
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....  
podpis diplomanta

## **ABSTRAKT**

Táto práca popisuje celý životný cyklus implementácie mobilnej aplikácie pre podporu osobnostného rastu. Okrem hlavnej aplikácie je výstupom aj teoretická časť, ktorá popisuje nájdenie technického riešenia, pomocou ktorého môže klient vykonávať jednotlivé techniky životného koučingu interaktívnym spôsobom bez prítomnosti mentora. V práci sa venujem aj analýze, prieskumu a porovnaniu vybraných nástrojov pre vývoj mobilných aplikácií. Po tejto časti sa práca zaoberá technickým popisom riešenia (analýza, dizajn, implementácia a testovanie).

**Kľúčové slová:** mobilná aplikácia, Android, Ionic framework, Angular, multiplatformová aplikácia, WebView, životný koučing, osobnostný rast

## **ABSTRACT**

This thesis describes a full development process of application for personal development process. In addition to the final application, the output is also theoretical part that describes technical solution by which the client can perform individual life coaching techniques in an interactive way without the presence of a mentor. The thesis also deals with the analysis, research and comparison of selected tools for the development of mobile applications. After this part, the work deals with the technical description of the solution (analysis, design, implementation and testing).

**Keywords:** mobile application, Android, Ionic framework, Angular, multiplatform application , WebView, life coaching, personal development

V prvom rade by som chcel poďakovať vedúcemu mojej diplomovej práce, pánovi Ing. Radekovi Valovi Ph.D. za dovolenie pracovať na tejto práci pod jeho dohľadom a za všetky cenné rady, ktoré mi s ochotou poskytol. Ďalej by som chcel poďakovať Dr. Petre Tomašovič a jej manželovi Ing. Robertovi Tomašovičovi za všetky konzultácie, počas ktorých mi objasnili problematiku životného koučingu.

Veľká vďaka patrí mojej manželke za porozumenie, trpezlivosť a neutíchajúcu podporu z jej strany. Taktiež veľmi pekne ďakujem mojím rodičom a starým rodičom za možnosť študovať a neúnavnú podporu počas celej doby môjho štúdia, ktorú som od nich dostával.

Prehlasujem, že odovzdaná verzia diplomovej práce a elektronická verzia nahraná do IS/STAG sú totožné.

# Obsah

Úvod.....	10
<b>I. Teoretická část .....</b>	<b>11</b>
<b>1 Možnosti přístupu vývoje mobilných aplikací .....</b>	<b>12</b>
1.1 Natívne aplikácie .....	12
1.2 Hybridné aplikácie .....	13
1.3 Multiplatformové mobilné aplikácie.....	14
1.4 Progresívne webové aplikácie .....	15
<b>2 Porovnanie nástrojov pre vývoj mobilných aplikácií.....</b>	<b>16</b>
2.1 Výber frameworkov pre porovnanie.....	16
2.2 Spoločné vlastnosti .....	16
2.3 Špecifiká frameworkov .....	17
2.3.1 Flutter .....	17
2.3.2 NativeScript.....	18
2.3.3 Ionic.....	19
2.4 Podpora natívnych API.....	19
2.5 Popularita a komunita.....	20
2.5.1 Stack Overflow .....	20
2.5.2 Twitter.....	21
2.5.3 Github .....	21
2.5.4 Analýza Google Trends .....	22
2.5.5 Prieskum JetBrains .....	23
2.5.6 Populárne aplikácie .....	24
2.6 Licencovanie .....	25
2.7 Porovnanie výkonu .....	25
2.7.1 Čas spustenia aplikácie .....	25
2.7.2 Veľkosť balíčkov zostavených aplikácií.....	26
2.7.3 Záverečné zhodnotenie.....	26
<b>II. Praktická časť .....</b>	<b>27</b>
<b>3 Návrh mobilnej aplikácie .....</b>	<b>28</b>

3.1	Identifikácia biznis procesu.....	28
3.1.1	Motivácia vytvorenia aplikácie.....	29
3.1.2	GROW model .....	29
3.1.3	Koleso životnej rovnováhy .....	30
3.1.4	Technika magickej reality.....	30
3.1.5	Energia .....	30
3.2	Funkčné požiadavky.....	31
3.3	Nefunkčné požiadavky.....	31
3.4	Prípady použitia .....	32
3.5	Návrh používateľského rozhrania .....	33
3.5.1	Tok obrazoviek aplikácie .....	33
3.5.2	Hlavná ponuka aplikácie .....	34
3.5.3	Technika Koleso životnej rovnováhy .....	35
3.5.4	Technika Energia .....	36
3.5.6	Technika Magická realita .....	37
3.5.7	Technika GROW .....	38
3.6	Spôsoby monetizácie aplikácie .....	39
3.6.1	Návrh stratégie monetizácie .....	39
3.6.2	Porovnanie s obdobnými aplikáciami .....	39
3.6.3	Návrh monetizačnej stratégie .....	40
<b>4</b>	<b>Implementácia .....</b>	<b>42</b>
4.1	Architektúra aplikácie.....	42
4.2	Popis použitých knižníc vo vývoji aplikácie .....	43
4.3	Implementácia metódy GROW .....	43
4.3.1	Motivácia vytvorenia dynamického formulára .....	43
4.3.2	Popis implementácie.....	44
4.3.3	Záver .....	46
4.4	Ukladanie dát .....	46
4.4.1	Dátový model.....	46
4.4.2	Popis implementácie prístupu k databáze.....	47
4.5	Zostavenie aplikácie .....	50
4.5.1	Testovanie a vývoj.....	50
4.5.2	Produkcia .....	50
4.5.3	Požiadavky .....	50



---

4.6	<i>Testovanie aplikácie</i> .....	51
4.6.1	Príprava .....	51
4.6.2	Vykonanie .....	51
4.7	<i>Internacionalizácia aplikácie</i> .....	53
4.7.1	Popis funkčnosti .....	53
4.8	<i>Spôsob rozširovania aplikácie</i> .....	53
<b>Záver</b> .....		<b>54</b>
<b>Zoznam použitej literatúry</b> .....		<b>55</b>
<b>Zoznam použitých skratiek a symbolov</b> .....		<b>57</b>
<b>Zoznam obrázkov</b> .....		<b>58</b>
<b>Zoznam tabuliek</b> .....		<b>59</b>
<b>Zoznam príloh</b> .....		<b>60</b>

## ÚVOD

V dnešnej dobe je množstvo ľudí, ktorí sa snažia dosiahnuť vyvážený pracovný a osobný život a snažia sa napredovať dopredu vo viacerých životných oblastiach. Chcú získať zo života viac a vyťažiť zo seba čo najviac. Ponechať si takúto motiváciu spočiatku nie je vôbec jednoduché a tak sa ľudia začali obracať na mentorov životného koučingu, ktorí im vedia pomôcť dosahovať vyššiu mieru úspechu a spokojnosti v ich osobnom a profesionálnom živote.

Hlavným cieľom tejto práce je vytvorenie aplikácie, ktorá slúži k podpore osobnostného rastu prostredníctvom techník životného koučingu. Táto aplikácia má za účel v istej miere suplovať aktivity mentora a interaktívnym spôsobom ponúka vykonanie rôznych psychologických testov, ktoré predstavujú rôzne techniky životného koučingu a bežne sa používajú pri sedeniach s klientami.

Tento dokument popisuje implementáciu mobilnej aplikácie v celom životnom cykle jej vývoja - teda od teoretického základu samotnej problematiky, cez analýzu požiadaviek, až po technický popis implementácie aplikácie. V teoretickej časti sa venujem prieskumu a analýze aktuálnych nástrojov pre vývoj mobilných aplikácií. Tieto nástroje sú medzi sebou porovnávané rôznymi metrikami, vďaka ktorým bol vykonaný výber technológie pre implementáciu. V praktickej časti sa sústreďujem na analýzu biznis procesu, analýzu požiadaviek na aplikáciu, technický popis riešenia a testovanie aplikácie.

## **I. TEORETICKÁ ČASŤ**

## 1 MOŽNOSTI PRÍSTUPU VÝVOJA MOBILNÝCH APLIKÁCIÍ

Táto kapitola popisuje spôsoby ako pristupovať k vývoju mobilných aplikácií. Aktuálne je možné použiť množstvo frameworkov a nástrojov k vývoju pre rôzne mobilné platformy. V závislosti od vybraného prístupu k vývoju budú vyplývať spôsoby distribúcie aplikácie. Na základe zdrojov [1], [2] a [3] môžeme jednotlivé typy mobilných aplikácií rozdeliť na natívne, hybridné, multiplatformové a progresívne aplikácie.

Spôsob vývoja jednotlivých typov aplikácií sa líši, a tak výber typu aplikácie resp. spôsob jej vývoja má podstatný vplyv na

- Výkon
- Čas dodania aplikácie na trh
- Rozpočet
- Náročnosť implementácie
- Dostupnosť aplikácie pre rôzne platformy
- Podporu natívnych rozhraní (API) zariadenia

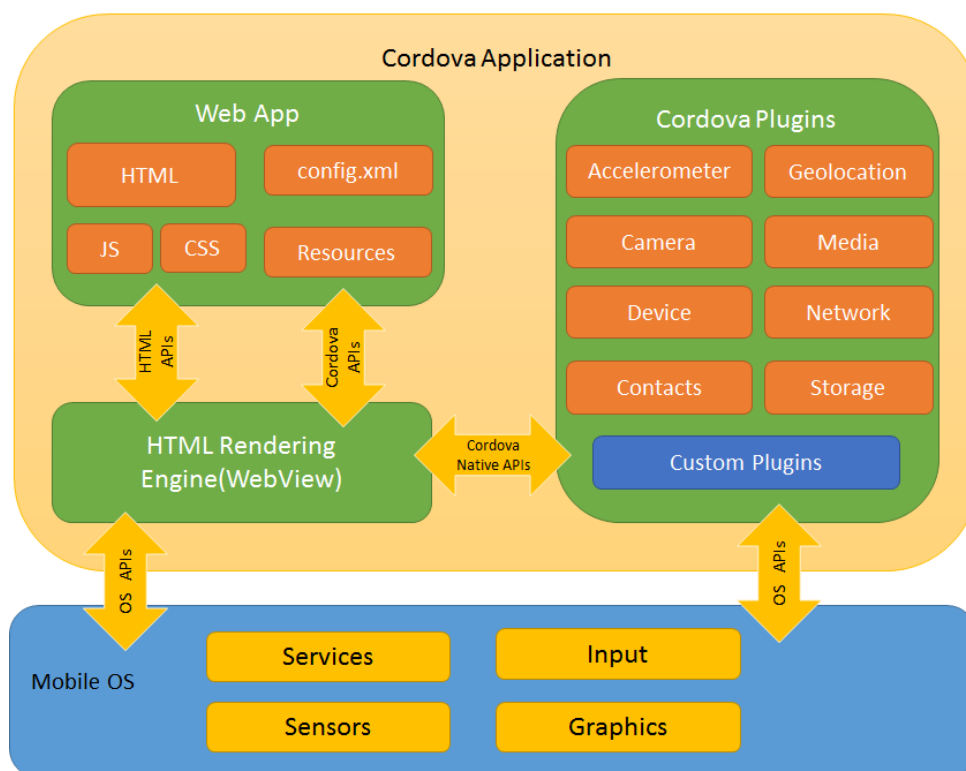
V nasledujúcich podkapitolách sú vysvetlené špecifické vlastnosti z pohľadu vývoja jednotlivých typov mobilných aplikácií a aktuálne možnosti nástrojov pre ich vývoj.

### 1.1 Natívne aplikácie

Vytvárať aplikáciu natívnym spôsobom vývoja mobilnej aplikácie hlavne znamená, že aplikácia bude naprogramovaná jazykmi špecifickými pre danú platformu, ktorými sú Objective-C alebo Swift pre iOS a Java, Kotlin alebo C++ pre Android [1]. Pomocou nich sa pristupuje priamo k natívnemu SDK. Z toho vyplýva, že ak potrebujeme aplikáciu dodať pre iOS a zároveň Android, tak potrebujeme vytvoriť dve unikátne aplikácie naprogramované v rôznych programovacích jazykoch. Výsledkom však bude, že aplikácia je na zariadení spustená a vykresľovaná priamo natívnym vykresľovacím jadrom zariadenia, bez ďalších medzivrstiev kódu, ktoré môžu ovplyvniť výkon [1]. Čo môže byť výhodou, ak chod aplikácie vyžaduje vysoko náročné CPU a GPU spracovanie s vysokou spotrebou pamäte [4].

## 1.2 Hybridné aplikácie

Hybridné aplikácie sú natívne aplikácie, v ktorých je spustená webová aplikácia pomocou objektu WebView. Webová aplikácia je vo WebView schopná komunikovať s rozhraním zariadenia pomocou „Web to Native Bridge“ nástrojov, akými sú Apache Cordova alebo Capacitor. Tieto nástroje sprístupňujú webovej aplikácii spustenej vo WebView Javascript API na komunikáciu s natívnym API. Výhodou je, že kód je napísaný len raz a väčšina na-programovanej logiky je zdieľaná medzi rôznymi platformami ako Android, iOS, Web alebo desktop. [1]. Vývoj je pomocou štandardných webových technológií ako HTML, CSS a JavaScript, tým pádom sa skúsení webový vývojári nemajú potrebu naučiť kompletne nový programovací jazyk. Medzi populárne frameworky na tvorbu hybridných mobilných aplikácií patria frameworky Ionic, Onsen UI alebo Kendo UI. Hybridnú mobilnú aplikáciu je tak tiež možné zostrojiť aj pomocou štandardného JavaScriptu.



Obrázok 1 Ukážka architektúry hybridnej aplikácie [5]

### 1.3 Multiplatformové mobilné aplikácie

Výstupom vývoja multiplatformovej aplikácie je aplikácia programovaná ako jeden projekt v jednom programovacom jazyku a spustiteľná vo viacerých platformách ako Android, iOS, Web alebo Desktop. Aplikácia pri spustení taktiež využíva natívne vykresľovacie jadro zariadenia bez použitia medzivrstiev kódu ako WebView [2], v čom je aj najpodstatnejší rozdiel oproti hybridným aplikáciám. Kód aplikácie je kompilovaný do natívneho kódu pomocou takzvaných „Bridges“. Najpopulárnejšie nástroje pre vývoj multiplatformových mobilných aplikácií sú Flutter, NativeScript, React Native a Xamarin.

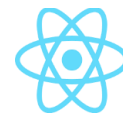
Tabuľka 1 Prehľad frameworkov pre vývoj multiplatformových aplikácií



**Xamarin**



**NativeScript**



**React Native**



**Flutter**

<i>Rok vydania</i>	2011	2014	2015	2018
<i>Spoločnosť</i>	Microsoft	Telerik	Facebook	Google
<i>Tvorba GUI</i>	XAML alebo xamarin.forms	XML, CSS	JSX, CSS	Dart
<i>Programovací jazyk</i>	C#	JavaScript	JavaScript	Dart

## 1.4 Progresívne webové aplikácie

Progresívne webové aplikácie (PWA) sú novým typom webových aplikácií a kombinujú benefity natívnych mobilných a webových aplikácií. Sú spustené na mobilnom zariadení ako bežná webová stránka, ale vyzerajú ako bežná natívna mobilná aplikácia. PWA sa spúšťa z domovskej obrazovky užívateľa a vďaka funkčnostiam ako Service Workers, Cache API sa načíta okamžite bez ohľadu na funkčné internetové pripojenie. Výhoda je v ich prenositeľnosti, keďže nie je potrebné meniť spôsob nasadzovania a balíčkovania aplikácie vzhľadom na typ fyzického zariadenia. Sú inštalovateľné, pridané na domovskú obrazovku priamo z webu a bez potreby návštevy obchodu s aplikáciami. [6]

## 2 POROVNANIE NÁSTROJOV PRE VÝVOJ MOBILNÝCH APLIKÁCIÍ

Z predošlej kapitoly vyplýva, že existujú viaceré prístupy k vývoju mobilných aplikácií a taktiež množstvo nástrojov, pomocou ktorých môžeme aplikácie vytvárať. Táto kapitola sa bude venovať porovnaniu niektorých konkrétnych nástrojov k vývoju. Sú porovnávané v rôznych oblastiach, aby objektívny výstup z ich porovnania pomohol k správne rozhodnutiu výberu.

### 2.1 Výber frameworkov pre porovnanie

Do užšieho zoznamu nástrojov pre porovnanie boli zvolené frameworky Flutter, Ionic, NativeScript. Boli vybrané z dôvodu ich popularity, výkonu, množstva vzdelávacieho materiálu na internete a jednoduchosti implementácie.

### 2.2 Spoločné vlastnosti

Spoločné vlastnosti týchto frameworkov môžeme zhrnúť do viacerých bodov:

- Zostaviteľnosť aplikácie pre platformy Android a iOS.
- Opakované použitie kódu. Väčšina kódu je spoločná pre obidve platformy, z toho dôvodu nie je potrebné vyvíjať rovnakú funkčnosť v inom jazyku.
- Programovanie bez platformovo závislých jazykov ako Swift, Objective-C, Java alebo Kotlin.
- Integrovaný prístup na natívne API ku funkciám zariadenia, akými sú napríklad geolokácia, kamera, súborový systém a iné.
- Jednoduchá udržiavateľnosť. V prípade opráv alebo zmien v aplikácii nie je potrebné projekt spravovať pre každú platformu zvlášť.
- Sú Open Source.
- Vývoj je možný vo Windows, Linux a MacOS.

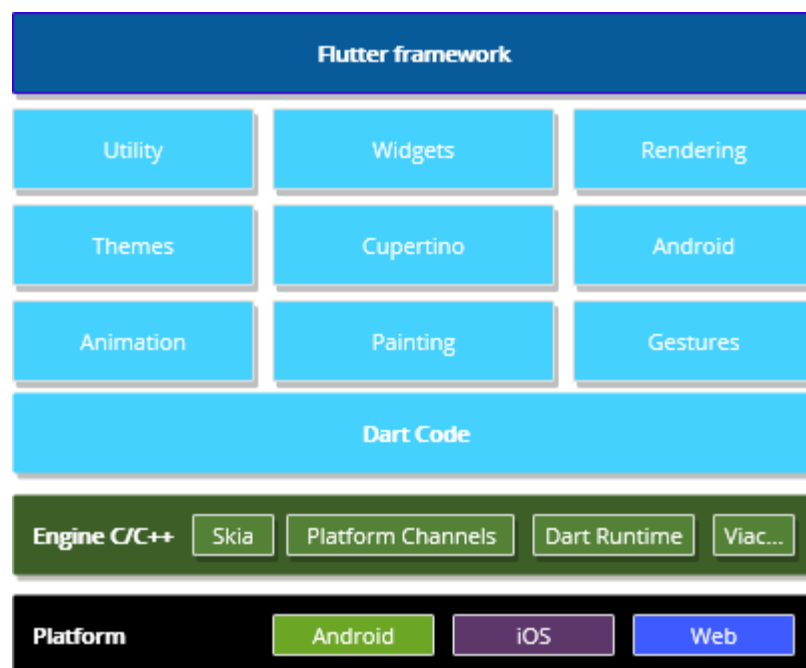


## 2.3 Špecifiká frameworkov

Na úvod je ešte potrebné vyzdvihnúť špecifiká jednotlivých frameworkov. Tieto frameworky sa líšia vo svojej architektúre, ktorá bude vysvetlená v nadchádzajúcich podkapitolách..

### 2.3.1 Flutter

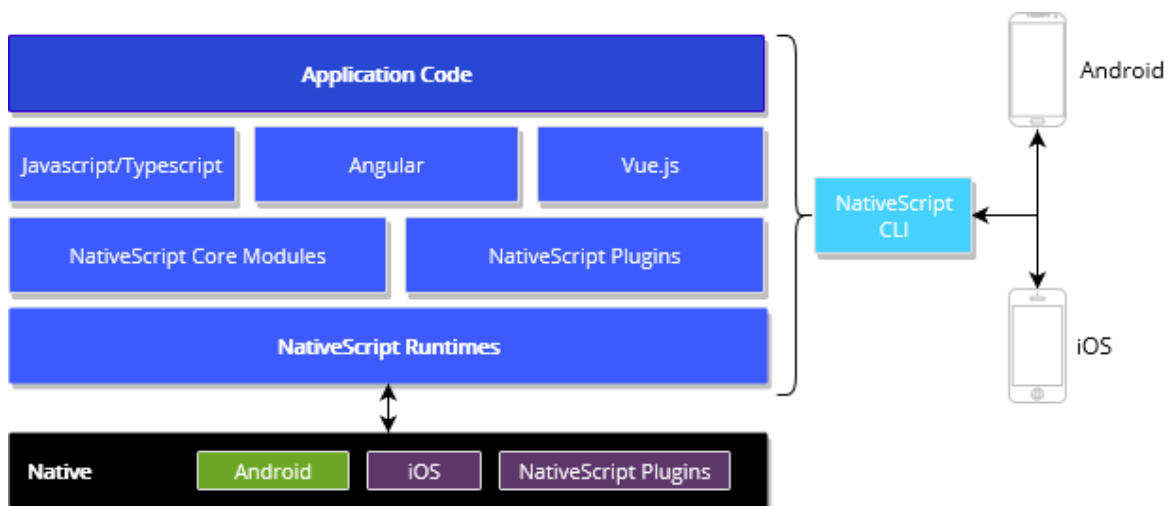
Flutter je framework vytvorený spoločnosťou Google. Pre vývoj sa používa programovací jazyk Dart. Flutter SDK v sebe zahŕňa funkčnosť vlastného vykresľovacieho jadra, a preto sa Dart kód nekompiluje ďalej do natívneho kódu platformy, ale v podstate vlastným jadrom kontroluje každý pixel zobrazený na užívateľskej obrazovke. Urýchliť vývoj aplikácie dokáže funkčnosť „Hot reload“, čo zabezpečí pri zmenách v kóde uchovať predošlý stav a prekresliť len súčasti v ktorých nastala zmena. Veľmi špecifické pre Flutter je to, akým spôsob sa vytvára užívateľské rozhranie aplikácie. Ide o princíp “UI as Code”, kde sa kód pre užívateľské rozhranie vytvára pomocou Dart kódu a každý prvok užívateľského rozhrania sa chápe ako Widget [7].



Obrázok 2 Architektúra Flutter [7]

### 2.3.2 NativeScript

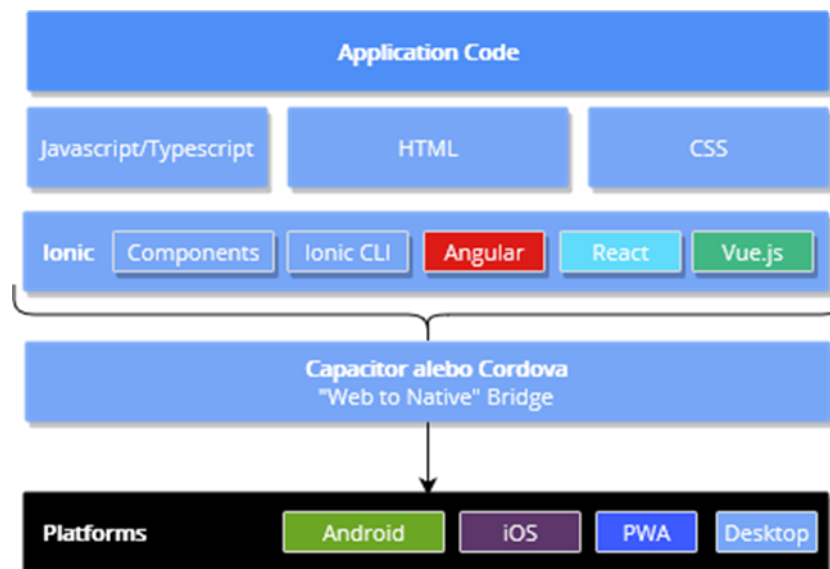
NativeScript používa k vývoju mobilných aplikácií JavaScript. Podporuje Typescript a frameworky Angular a Vue.js. Podpora pre tieto nástroje je dosiahnutá samostatným vývojom verzii NativeScriptu. Napriek tomu, že vývoj aplikácie je pomocou štandardných webových technológií Javascript, XML a CSS, tak aplikácia nie je spustená vo WebView objekte, ale je spustená priamo ako natívna aplikácia. Toto je dosiahnuté pomocou súčasti NativeScript Runtimes, ktorá poskytuje rozhranie na komunikáciu s natívnym Android alebo iOS API [8]. Vďaka NativeScript súčasti Core Modules je Javascript kód a kód pre UI preložený do natívneho kódu.



Obrázok 3 Architektúra NativeScript [8]

### 2.3.3 Ionic

Ionic je typický hybridný framework. Na rozdiel od NativeScriptu a Flutteru je mobilná aplikácia spustená vo WebView objekte a ide prakticky o webovú aplikáciu. Rozhranie od nástrojov ako Cordova alebo Capacitor umožňujú pristupovať ku natívnym vlastnostiam platformy ako napríklad kamera, geolokácia a iné. Vývoj je pomocou štandardných webových technológií HTML, CSS, Javascript/Typescript. Ionic podporuje React, Angular, Vue.js alebo len natívny Javascript. Pochopiteľne je Ionic veľmi populárna voľba pre webových vývojárov.






Obrázok 4 Architektúra Ionic framework [3]

## 2.4 Podpora natívnych API

Mobilné aplikácie bežne pristupujú ku funkciám zariadenia akými sú kalendár, kontakty, kamera, kontrola oprávnení a rôzne iné. Všetky tri frameworky ponúkajú pre vývojárov balíčky, pomocou ktorých je možné použiť natívne funkcie zariadenia. Taktiež podporujú tvorbu vlastných balíčkov. Pre všetky tri platí, že balíčky vytvára priamo vývojový tím frameworku a taktiež množstvo z nich sú vytvorené komunitou. V databázach Flutter a NativeScript balíčky obsahujú aj také, ktoré slúžia ako podpora pri vytváraní UI, a tak počty balíčkov, ktoré publikovala komunita prevyšujú počty komunitne dodaných balíčkov pre Ionic.

Tabuľka 2 Porovnanie počtu dodatočných balíčkov

	 Flutter	 NativeScript	 Ionic
<i>Oficiálne</i>	46	41	33
<i>Neoficiálne</i>	6 000+	1 300+	260




## 2.5 Popularita a komunita

Jedným z faktorov, ktorý môže vplývať na výber je popularita frameworkov a komunita, ktorá za danými nástrojmi stojí. V tejto časti popíšem prieskum, ktorý skúma popularitu a komunitu vybraných frameworkov v rôznych kanáloch internetu k januáru roku 2020.

### 2.5.1 Stack Overflow

Web Stack Overflow je komunitná stránka s obrovskou komunitou určená na pomoc vývojárom každého druhu. Funguje spôsobom otázka – odpoveď, kde otázku pokladajú ľudia, ktorí sa stretli s nejakým špecifickým problémom a komunita na ňu odpovedá. V prieskume nás bude zaujímať koľko otázok bolo vytvorených v téme skúmaných frameworkov.




Tabuľka 3 Porovnanie počtu vytvorených otázok v Stackoverflow pre jednotlivé frameworky

	 Ionic	 Flutter	 NativeScript
<i>Počet otázok</i>	42 100	34 400	6 100

### 2.5.2 Twitter

Twitter je sociálna sieť, v ktorej majú jednotlivé projekty vytvorené oficiálne profily, prostredníctvom ktorých informujú komunitu o svojich novinkách alebo radách k vývoju. Nasledujúca tabuľka porovnáva počty sledovateľov oficiálnych profilov jednotlivých frameworkov.




Tabuľka 4 Porovnanie počtu Twitter sledovateľov

	 <b>Ionic</b>	 <b>Flutter</b>	 <b>NativeScript</b>
<i>Počet otázok</i>	109 000	80 000	75 000

### 2.5.3 Github

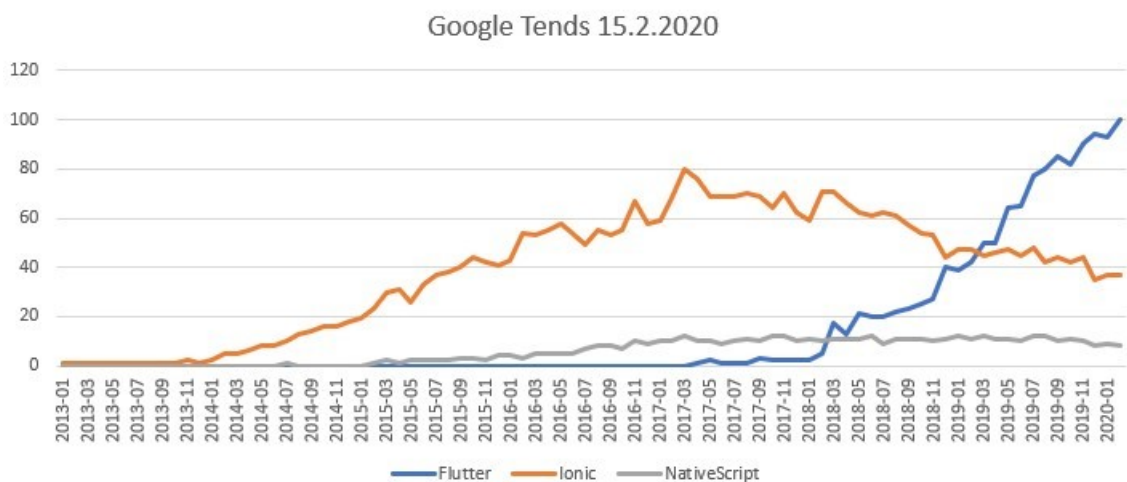
Vybrané frameworky sú Open Source a ich kód je voľne dostupný v službe Github. Na tejto platforme sa nachádza veľká komunita vývojárov z celého sveta a Github komunita umožňuje zapojiť sa do vývoja Open Source softvéru. Aktuálnu popularitu jednotlivých projektov v komunite Github môžeme určovať podľa počtu „Github Stars“ a počtu ľudí, ktorí sa rozhodli prispievať svojou prácou do projektov. Nasledujúca tabuľka predstavuje aktuálne hodnoty „Github Stars“ a počtu prispievateľov do jednotlivých frameworkov.

Tabuľka 5 Porovnanie počtu Github Stars

	 <b>Flutter</b>	 <b>Ionic</b>	 <b>NativeScript</b>
<i>Github Stars</i>	85 000	40 000	18 000
<i>Prispievatelia</i>	525	354	159

### 2.5.4 Analýza Google Tends

Vďaka nástroju Google Tends môžeme porovnať, aké bolo populárne vyhľadávanie názvov jednotlivých frameworkov vo vyhľadávači Google. Pre porovnanie boli zvolené kľúčové slová „Flutter“, „Ionic“, „NativeScript“. Dátum štatistík bol zvolený od dňa 1.1.2013 vzhľadom na rok vydania najstaršieho zo skúmaných frameworkov, Ionicu. Parameter oblasti bol zvolený na hodnotu „celosvetovo“ a téma bola zvolená na hodnotu „programovanie“. Nasledujúci graf znázorňuje trend relatívnej popularity v priebehu času pre jednotlivé frameworky.



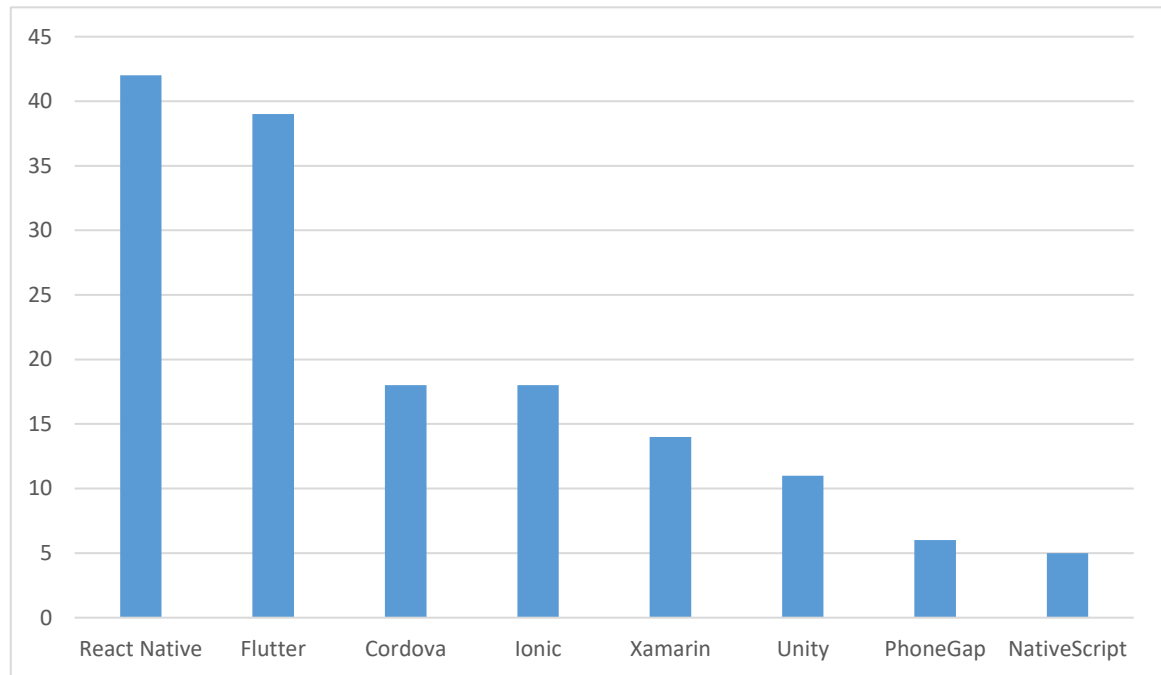
Obrázok 5 Porovnanie počtu vyhľadávaní v Google

Pri pohľade na krivky je jasné, že Flutter sa teší raketovému vzostupu od roku 2018, kdežto pre Ionic začal platiť od roku 2018 mierne klesajúci trend vo vyhľadávaní. Záujem o vyhľadávanie NativeScript je najmenej populárny, avšak stály počet vyhľadávaní si dlhodobo drží.

Tento prieskum však reálne neodzrkadľuje skutočnú popularitu, keďže klesajúca krivka relatívnej popularity môže v jednoduchosti znamenať, že ľudia na stránku prístupujú priamo, a nie cez vyhľadávač Google.

### 2.5.5 Prieskum JetBrains

Prieskum od spoločnosti JetBrains zachytáva na vzorke 7 000 respondentov použitie frameworkov vývojármi.



Obrázok 6 Výsledok prieskumu používania frameworkov vývojármi od spoločnosti JetBrains v roku 2020 [18]

Podľa grafu sa Flutter umiestňuje na druhej priečke, Ionic na štvrtej a NativeScript na ôsmej priečke. Taktiež si môžeme všimnúť, že celkové použitie hybridných riešení (WebView) ako Cordova, Ionic, PhoneGap je medzi vývojármi veľmi časté.

## 2.5.6 Populárne aplikácie

Na webových stránkach jednotlivých frameworkov môžeme nájsť sekciu „Showcase“, ktorá prezentuje populárne aplikácie, ktoré boli pomocou jednotlivých nástrojov vytvorené.

### 2.5.6.1 *Flutter*

Medzi najpopulárnejšie aplikácie vytvorené pomocou frameworku Flutter patria aplikácie Alibaba, Reflectly, Google Ads a aplikácie pre Google Assistant.

### 2.5.6.2 *Ionic*

Aplikácie MarketWatch, Sworkit Fitness, Pacifica, McDonald patria medzi najpopulárnejšie aplikácie vytvorené vo frameworku Ionic. Ionic taktiež uvádza použitie v spoločnostiach McDonald a Diesel.

### 2.5.6.3 *NativeScript*

Aplikácie Yape, Sennheiser Smart Control, MyPuma patria medzi najpopulárnejšie vytvorené v NativeScripte. Na stránke NativeScript môžeme nájsť informácie, že spoločnosti ako napríklad Deloitte Digital, Verizon, SAP a iné NativeScript používajú.



## 2.6 Licencovanie

Vybrané frameworky sú voľne dostupné Open Source nástroje. Licenciami sa však mierne líšia. Flutter je dostupný pod novou BSD licenciou, NativeScript pod Apache 2.0 a Ionic je dostupný pod MIT licenciou.

## 2.7 Porovnanie výkonu

Výkon aplikácií je jeden z kľúčových parametrov pre užívateľov. Užívatelia neradi čakajú a z tohto dôvodu bola porovnaná rýchlosť spustenia aplikácií vytvorených vo vybraných frameworkoch. Pre požiadavky testu boli vytvorené 4 rôzne aplikácie s jednou základnou obrazovkou s textom „Hello World“. Boli vytvorené nasledujúce zostavy:

- Natívna Android aplikácia pomocou nástroja Android Studio v jazyku Java
- Flutter aplikácia
- NativeScript aplikácia
- Ionic aplikácia

### 2.7.1 Čas spustenia aplikácie

Na vytvorených aplikáciách bol meraný čas spustenia – tzn. „Cold start“. Čas spustenia je možné zistiť z logov aplikácie pomocou nástroja Logcat v aplikácii Android Studio. Každá aplikácia bola spustená 5-krát a výsledný čas je priemerom nameraných časov.

Tabuľka 6 Porovnanie času spustenia základných aplikácií

	<b>Android</b>	<b>Flutter</b>	<b>Ionic</b>	<b>NativeScript</b>
<i>Čas spustenia</i>	420 ms	577 ms	700 ms	1 300 ms

### 2.7.2 Veľkosť balíčkov zostavených aplikácií

Tabuľka 7 Porovnanie veľkosti zostavených aplikácií

	Android	Flutter	Ionic	NativeScript
Čas spustenia	3.1 MB	15 MB	9.1 MB	13 MB

### 2.7.3 Záverečné zhodnotenie

Z vykonaného porovnania síce vyplývajú technické špecifiká jednotlivých nástrojov, ale taktiež o žiadnom z nich nemôžeme povedať, že sa jedná o nevhodný nástroj pre vývoj mobilnej aplikácie, keďže všetky predmetné technológie sú aktuálne populárne, podporované, vyvíjané a používané komunitou vývojárov.

Výber nástroja pre vývoj multiplatformovej mobilnej aplikácie z pomedzi technológii Ionic, Flutter a NativeScript považujem za veľmi subjektívny. Výber správneho závisí od mojich predchádzajúcich pracovných skúseností, ochoty učiť sa konkrétnu technológiu a jej krivka učenia, vyhliadky do budúcnosti vývoja technológie a samozrejme od spomenutých ukazovateľov vyššie.

Keďže som fanúšik webových technológií zvolil som si nástroj Ionic. Na mňa ako webového front-end vývojára tento framework zapôsobil jednoduchosťou a rýchlym štartom spôsobu vývoja mobilnej aplikácie – vďaka tomu, že som mohol využiť množstvo mojich doterajších znalostí. Taktiež ma oslovila komunita vyvíjajúca framework, množstvo preddefinovaných UI komponentov, doplnkových a ľahko použiteľných knižníc a samozrejme licenčná politika. Naproti tomu ma NativeScript neoslovil spôsobom definície šablón, malou komunitou a množstvom rozdielneho kódu pre Android a iOS. Flutter vyzerá veľmi zaujímavo, ale keďže by som nemohol využiť moje doterajšie znalosti z webových technológií, tak by bola práca s týmto nástrojom časovo náročnejšia, a tým by sa predĺžil vývoj aplikácie.

## **II. PRAKTICKÁ ČASŤ**

### 3 NÁVRH MOBILNEJ APLIKÁCIE

Aplikácia slúži ako pomocný nástroj pre podporu životného koučingu. Implementuje interaktívnym spôsobom rôzne techniky, ktoré mentori životného koučingu používajú pri sedeniach s klientami.

Cieľom úvodnej analýzy je popísanie procesu sedenia mentora životného koučingu s klientom. Výstup z tejto analýzy bude použitý k nájdeniu prípadov použitia, ktoré by aplikácia mohla implementovať. Následne bude možné zdefinovať funkčné a nefunkčné požiadavky na aplikáciu. V ďalšom kroku budú zdefinované diagramy prípadov použitia, ktoré detailne popíšu jednotlivé funkčnosti aplikácie.

Analýza predchádzajúcich krokov bude následne použitá pre návrh užívateľského rozhrania, zdefinovanie kvalitatívnych požiadaviek a definovanie architektúry aplikácie.

#### 3.1 Identifikácia biznis procesu

Typická motivácia ľudí kontaktovať mentora životného koučingu je z dôvodu dosiahnutia alebo zlepšenia niečoho konkrétneho v ich životoch. Požadovaný výstup zo sedenia vo všeobecnosti spadá troch hlavných kategórií [9]:

1. Výkonnostné ciele: napríklad zlepšenie výkonu v podnikaní
2. Učebné ciele: byť trpezlivejší rodič, zlepšenie argumentácie
3. Ciele napĺňania: dosiahnuť lepšiu rovnováhu medzi osobným a pracovným životom

Mentor ľuďom zvyčajne pomáha s nasledujúcimi situáciami:

- Pomáha s napredovaním tam, kde to je pre jednotlivca náročné
- Výzva, o ktorej si nie sme istý
- Plán osobnostného rozvoja
- Brainstorming riešenia problému
- Nová stratégia alebo usmernenie v určitých situáciách

### 3.1.1 Motivácia vytvorenia aplikácie

Existuje veľké množstvo nástrojov pre výsluch klienta a techník, pomocou ktorých mentor pomáha klientovi získať viac informácií o jeho živote, problémoch, okolnostiach a pomôcť mu vo výzvach, ktoré tieto oblasti prinášajú. Avšak niektoré z nich je možné použiť aj na účely seba-koučovania, čiže koučingu bez prítomnosti mentora [10]. Klient teda tieto nástroje môže používať z pohodlia domova. Sledované techniky v tejto analýze môžu pomôcť klientovi v nasledujúcich veciach:

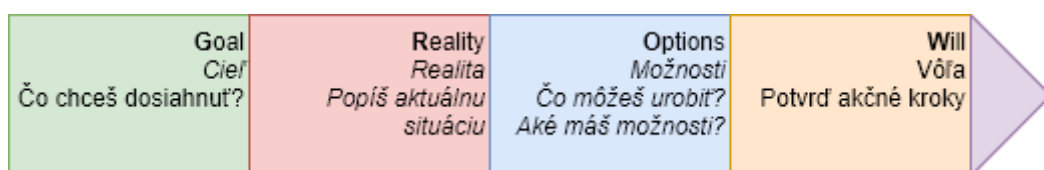
- Poznať lepšie seba samého, svoju osobnosť, motívy a ciele
- Uvedomiť si svoje silné a slabé stránky
- Pochopiť a predvídať možné zámery svojho správania, čo môže predchádzať nerozumieniam a konfliktom

Na základe osobných konzultácií s mentorom životného koučingu vyplynulo, že nástroje ako GROW model, Koleso životnej rovnováhy, Technika magickej reality a technika Energia sú vhodné pre seba-koučovanie, a teda z môjho pohľadu je ich možné aj interaktívne implementovať.

Aplikácia, ktorá by implementovala mnohé nástroje životného koučingu by poskytla pridanú hodnotu k sedeniam kouča s klientom, taktiež by v istých prípadoch dokázala kouča z časti nahradiť alebo doplniť. Klienta by motivovalo aplikáciu používať kvôli jednoduchému prístupu k nástrojom, ktoré mu pomôžu k dosiahnutiu vytúžených cieľov, naplánovaniu akčných krokov, nájdenia lepšej životnej rovnováhy alebo pomoci pri nájdení ich silných alebo slabých stránok. A to všetko vo chvíľach, keď klientovi stačí vziať svoj smartfón a spustiť aplikáciu.

### 3.1.2 GROW model

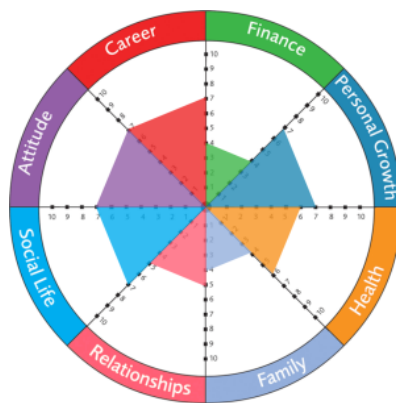
GROW model patrí medzi najpoužívanejšie nástroje používané v koučingu. Jedná sa o efektívnu techniku pre stanovenie cieľov, ktorá klientovi umožní vypracovať konkrétny plán akcií pre dosiahnutie vytúženého cieľa [10].



Obrázok 7 Fázy GROW modelu

### 3.1.3 Koleso životnej rovnováhy

Technika Koleso životnej rovnováhy sa bežne používa pri prvom alebo druhom sedení s mentorom. Používa sa na účely vizuálneho hodnotenia úrovne spokojnosti klienta v rôznych životných oblastiach, akými sú napr. financie, kariéra, zdravie, priatelia a iné. Táto vizuálna reprezentácia pomáha koučovi a klientovi porozumieť rozdielu medzi aktuálnou a požadovanou realitou klienta.



Obrázok 8 Príklad Kolesa životnej rovnováhy

Táto technika obsahuje nasledujúce fázy:

1. Premyslenie životných oblastí – 6 až 8 najdôležitejších oblastí pre klienta
2. Ohodnotenie spokojnosti s oblasťami od 0 do 10
3. Stanovenie akčného plánu k zlepšeniu oblastí s menším ohodnotením

### 3.1.4 Technika magickej reality

Použitím techniky magickej reality musí klient zapojiť svoju kreativitu a predstavivosť. Na základe premyslenia ideálnej reality mu pomáha zostaviť akčné kroky k jej dosiahnutiu [9].

Technika obsahuje nasledujúce fázy:

1. Stanovenie ideálneho stavu v budúcnosti
2. Stanovenie akčných krokov a ich termínov

### 3.1.5 Energia

Technika Energia pomáha k eliminovaniu vecí, ktoré klientovi „vysávajú“ energiu. Medzi takéto veci, ktoré klienta stoja energiu spadajú aktivity, zvyky, vzťahy alebo neporiadok v živote.

### 3.2 Funkčné požiadavky

Funkčné požiadavky definujú funkcionality aplikácie.

- REQ 1. Systém implementuje techniku Koleso životnej rovnováhy.
- REQ 2. Systém implementuje techniku Energia.
- REQ 3. Systém implementuje techniku GROW.
- REQ 4. Systém implementuje techniku Magickej reality.
- REQ 5. Systém umožní podporu lokalizácie v jazykoch slovenčina a angličtina.
- REQ 6. Systém zobrazuje sekciu s propagačnými článkami a zobrazenie ich detailu.
- REQ 7. Systém eviduje dokončené procesy jednotlivých techník.
- REQ 8. Systém umožní rozšírenie aplikácie o ďalšie sady nástrojov prostredníctvom štandardnej aktualizácie alebo cez sieť.

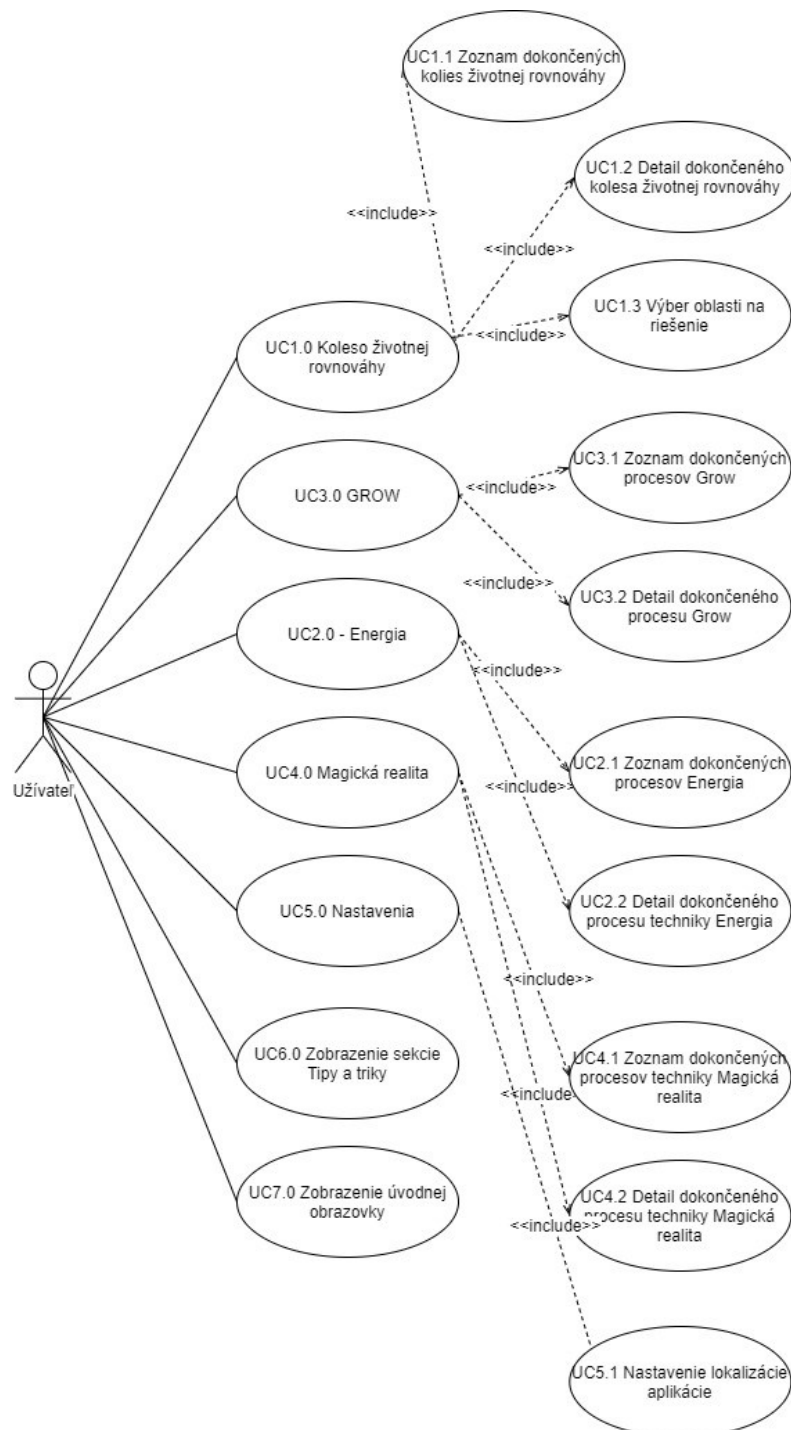
### 3.3 Nefunkčné požiadavky

Nefunkčné požiadavky definujú nároky a obmedzenie systému

- NR 1. Aplikácia bude implementovaná ako hybridná a bude spustiteľná na platformách Android a iOS.
- NR 2. Systém bude podporovať adaptívne užívateľské rozhranie pre mobil a tablet.
- NR 3. Systém nebude ukladať citlivé užívateľské dáta.
- NR 4. Systém bude pristupovať k nástrojom zariadenia ako Kalendár a Email.
- NR 5. Systém ukladá dáta na lokálne úložisko.
- NR 6. Rýchla odozva aplikácie.
- NR 7. Aplikácia bude podporovať internacionalizáciu – anglický a slovenský jazyk.

### 3.4 Případy použitia

Diagram prípadov použitia znázorňuje konkrétne funkcie aplikácie. V nasledujúcej sekcii sa nachádza špecifikácia jednotlivých prípadov použitia.



Obrázok 9 Diagram prípadov použitia

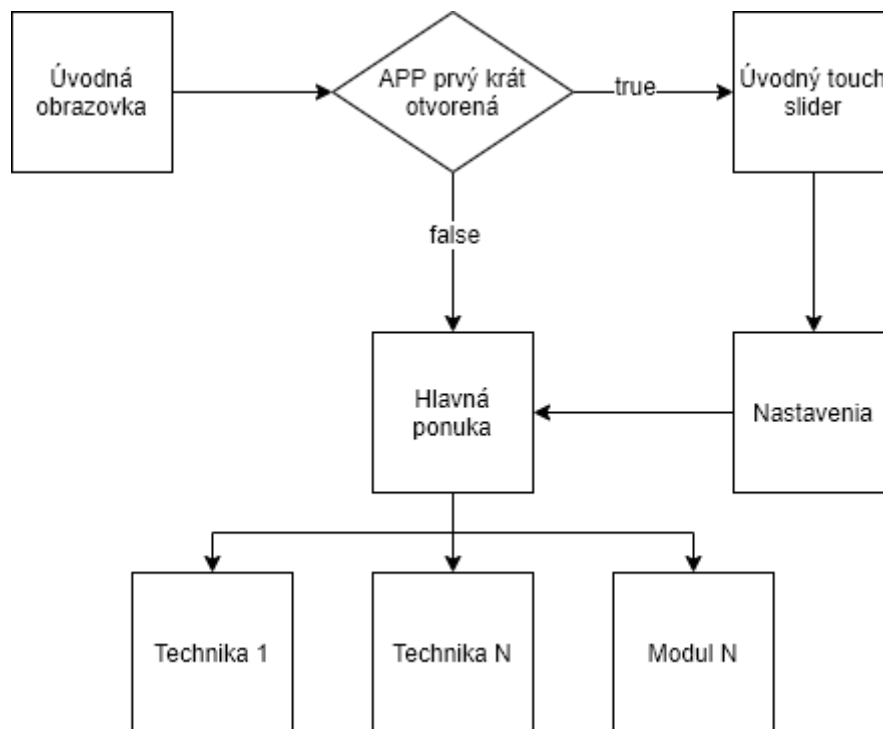


### 3.5 Návrh používateľského rozhrania

Návrh používateľského rozhrania popisuje hrubý návrh ako by aplikácia mala vizuálne pôsobiť, ako jednotlivé funkčnosti ponúknuť používateľovi, prípadne vizuálnym spôsobom vysvetlí spôsob interakcie medzi používateľom a aplikáciou.

#### 3.5.1 Tok obrazoviek aplikácie

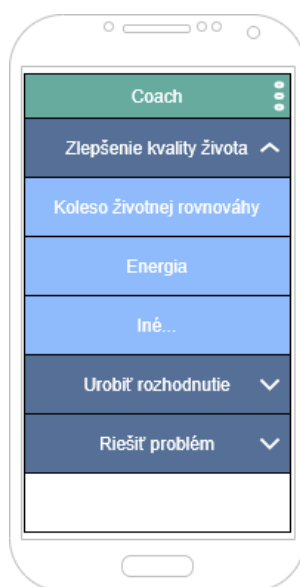
Nasledovný diagram popisuje v akom slede sa zobrazujú jednotlivé obrazovky v aplikácii. Taktiež popisuje prípad prvého spustenia aplikácie, kde nastáva zobrazenie obrazovky Úvodný touch slider. Na tejto obrazovke je používateľ oboznámený s funkčnosťami aplikácie, následne je zobrazená obrazovka Nastavenia.



Obrázok 10 Tok obrazoviek

### 3.5.2 Hlavná ponuka aplikácie

Tento návrh bol výsledkom brainstormingu o tom, ako by v aplikácii mala vyzerat' hlavná ponuka, z ktorej si používateľ spustí vybrané techniky životného koučingu v mobile a tablete. Pre zlepšenie používateľského zážitku bolo navrhnuté vytvorenie adaptívneho rozhrania, ktoré sa prispôsobuje podľa toho či je aplikácia spustená na mobile alebo tablete.



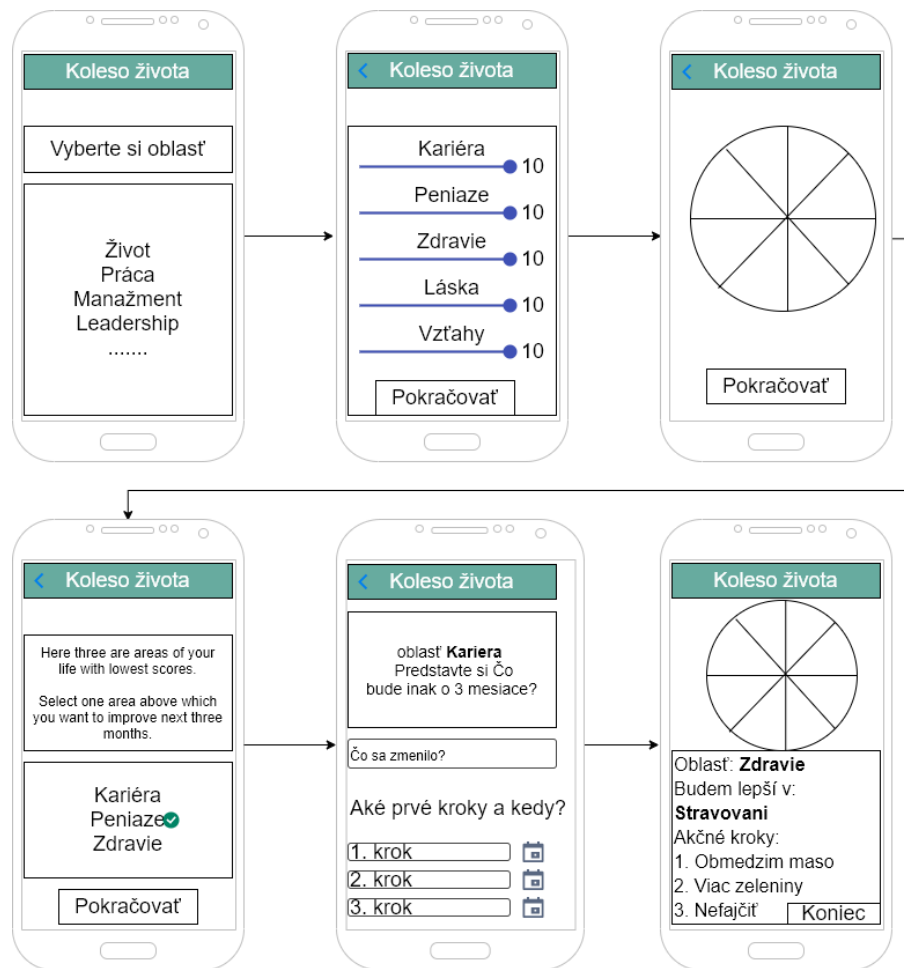
Obrázok 11 Návrh hlavnej ponuky pre mobil



Obrázok 12 Návrh hlavnej ponuky pre tablet

### 3.5.3 Technika Koleso životnej rovnováhy

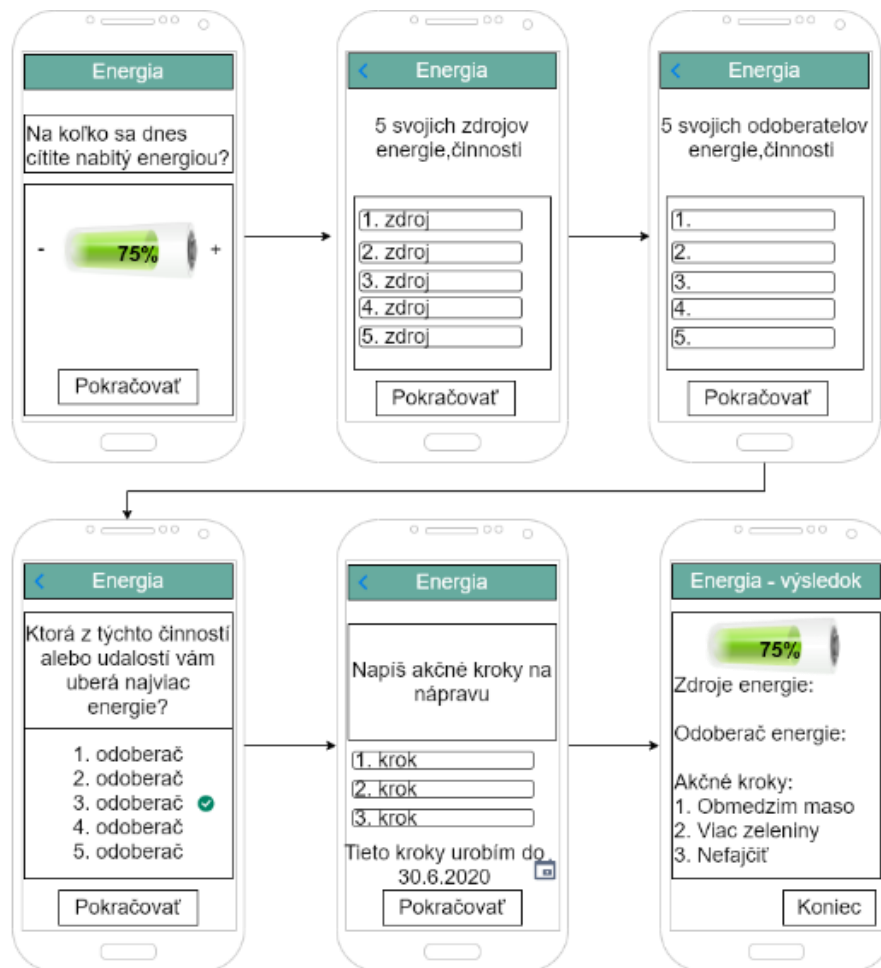
Nasleduje hrubý návrh obrazoviek pre implementáciu techniky Koleso života. Tento 6-krokový proces prevedie používateľa procesom nájdania lepšej životnej rovnováhy. Po dokončení procesu systém presmeruje do obrazovky Hlavná ponuka. Proces prechádza krokmi popísanými v hlavnom toku prípadu použitia UC1.0.



Obrázok 13 Návrh procesu techniky Koleso životnej rovnováhy

### 3.5.4 Technika Energia

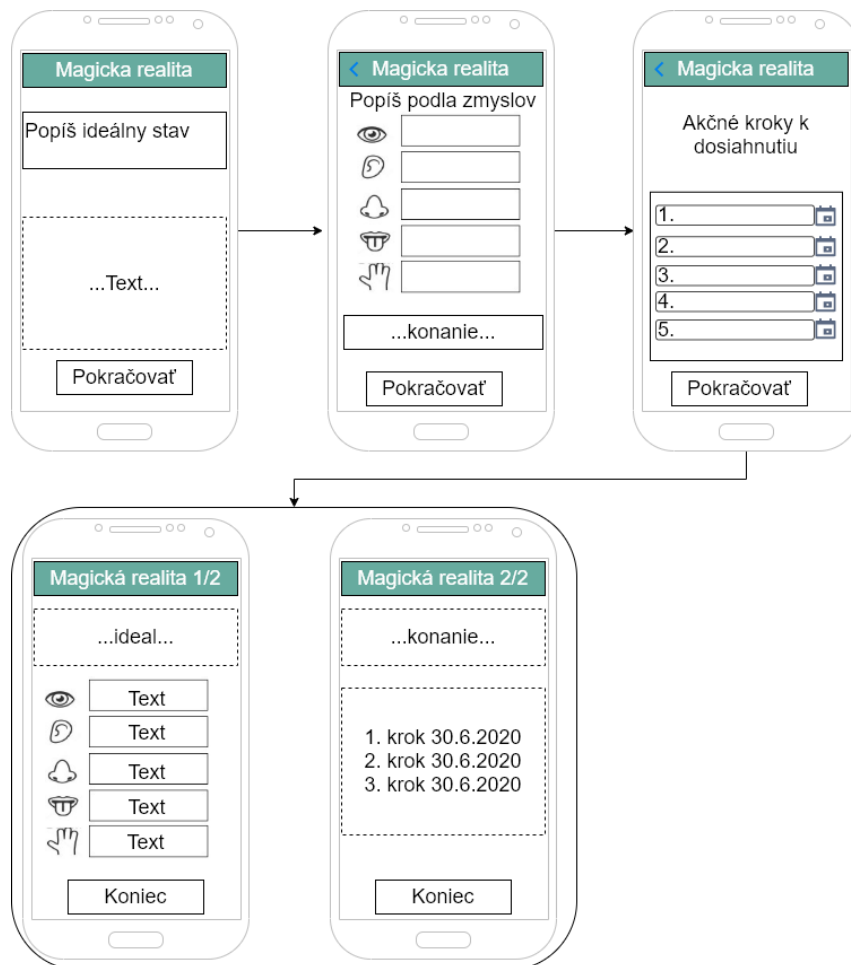
Návrh techniky Energia, v ktorom sú znázornené kroky procesu, ktorý pomôže klientovi vysporiadať sa s činnosťami, ktoré mu odoberajú energiu a so stanovením akčných krokov na nápravu. Výstupom tejto implementácie je aj evidencia predošlých hodnôt z procesov.



Obrázok 14 Návrh procesu techniky Energia

### 3.5.6 Technika Magická realita

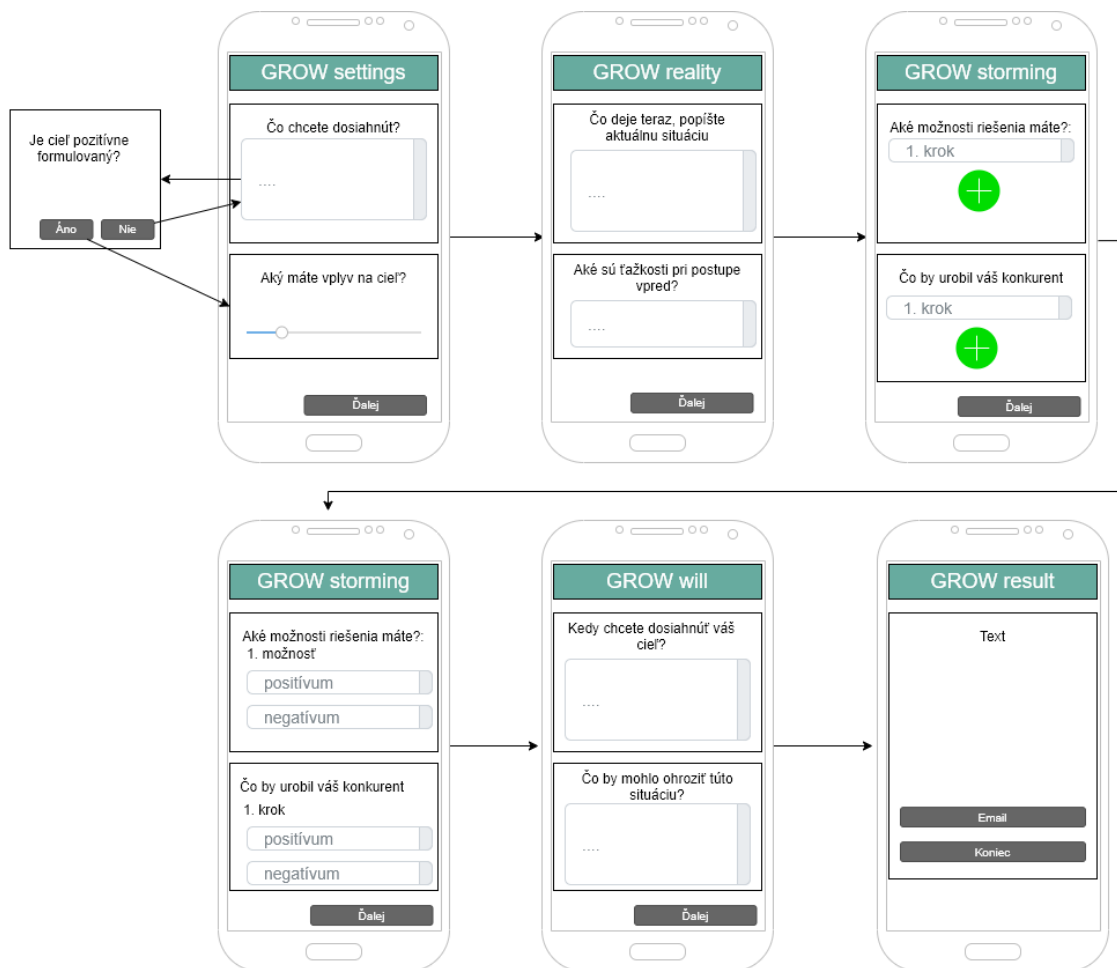
V tomto návrhu sú zachytené kroky techniky Magickej reality. Pomocou jednotlivých krokov tohto procesu používateľ v jednotlivých krokoch zapája svoju tvorivosť a uvažuje o svojom ideálnom stave v budúcnosti. Na základe svojej ideí zostavuje akčné kroky k dosiahnutiu vytúženého stavu.



Obrázok 15 Návrh techniky Magická realita

### 3.5.7 Technika GROW

Návrh zachytáva hlavné obrazovky na zadanie potrebných údajov v jednotlivých fázach GROW modelu. V každej fáze bude potrebné zadať väčšie množstvo údajov ako je uvedené v návrhu a tak sa navrhuje, že nasledujúce vstupné polia sa budú zobrazovať dynamicky až po zadaní predošlého poľa. Pre isté polia je potrebné zobrazit' modálne okno na potvrdenie, resp. vynútiť klienta zamyslieť sa nad svojou odpoveďou. Vzhľadom na akciu v modálnom okne bude možné klientovi nanútiť odpovedať na predošlú otázku alebo bude môcť pokračovať ďalej.



Obrázok 16 Návrh procesu techniky GROW

### 3.6 Spôsoby monetizácie aplikácie

Monetizácia mobilnej aplikácie je termín pre spôsob ako generovať z aplikácie peňažný zisk. Keď sa tvorcovia aplikácií snažia zostať ziskoví, otázka výberu správneho obchodného modelu a marketingového mixu sa stáva naliehavejšou. [11]

Vykonáva sa nasledujúcimi spôsobmi [12]:

1. Stratégiou mikro transakcií, po zaplatení ktorých budú ďalšie funkčnosti aplikácie odomknuté.
2. Predplatným.
3. Predávaním reklamného priestoru v užívateľskom rozhraní aplikácie.
4. Predávaním platenej verzie aplikácie.
5. Predávaním tovaru a služieb cez aplikáciu.

#### 3.6.1 Návrh stratégie monetizácie

K maximalizácii ziskov, je potrebné pre aplikáciu zvážiť niekoľko modelov monetizácie. Pretože rôzne typy publika, majú rôzne preferencie pre mikro transakcie, predplatné, zobrazovanie reklám alebo platené aplikácie [12].

#### 3.6.2 Porovnanie s obdobnými aplikáciami

Pre nastavenie počiatočnej monetizačnej stratégie, som vykonal prieskum stratégie na obdobných aplikáciách. Vybral som si úspešnejšie aplikácie zamerané na osobnostný rozvoj a s počtom stiahnutí aspoň 50 000. Do užšieho výberu som vybral Android aplikácie Remente, Improve My Life a Success Life Coach. Tieto aplikácie aspoň čiastočne implementujú niektoré techniky životného koučingu, takže ich považujem za porovnateľné s témou tejto práce. Informácie o daných aplikáciách a ich monetizáciách som čerpal po stiahnutí a ich testovaní na smartfóne Honor 9.

##### **Remente**

Táto aplikácia ponúka základný balík funkcií zdarma a bez reklám, avšak vo veľmi obmedzenom režime. Naproti tomu ponúka prémiový balík, na ktorý láka množstvom dodatočných funkcií a obsah, ktorý údajne vytvárajú experti pre psychológiu. V ponuke sú tri balíčky predplatného a to: mesačné za 10 €, ročné za 34 € a doživotné za 140 €. Zákazníka ale často oslovujú reklamnou kampaňou kde je zľava na balíčky až 50%.

**Improve My Life**

Aplikácia ponúka taktiež základný a prémiový balík. Avšak v základom balíku neobmedzuje užívateľa v ponuke techník osobnostného rozvoja, len ich prémiovým balíkom rozširuje o ďalšie funkčnosti ako neobmedzené množstvo zadaných akčných krokov, úloh alebo prispôbenie segmentov v technike Strom života. Ponúka dve možnosti predplatného a to: mesačné za 2 € a ročné za 21 €.

**Success Life Coach**

V tejto aplikácii je trochu komplexnejší model monetizácie. Ponúkajú tiež základný balík služieb Free, ale taktiež tu sú aj balíky Pro a Premium. Balík Pro nadväzuje na balík Free a ponúka neobmedzené zadanie cieľov, aktivít, poznámok a ďalších iných, kdežto v balíku Free je možné zadávať len obmedzené množstvo týchto funkčností. Balík Premium navyše pridáva funkčnosť exportu dát a rôzne webové služby. Je možné zaplatiť si predplatné buď pre balík Pro mesačne za 5.50 € Premiuma celoročné za 27 €. Pri balíku Premium je možné predplatiť si aplikáciu na pol roka za 49 €.

Prieskum monetizačných stratégií ma oboznámil s aktuálnymi biznis trendami v danej kategórii aplikácií a mohol som preskúmať akú politiku poplatkov za obsah zvolili rôzni vývojárari. Vďaka tomu môžem tieto modely zvážiť pri návrhu monetizácie pre aplikáciu v tejto práci.

**3.6.3 Návrh monetizačnej stratégie**

Pri návrhu som postupoval podľa rád od spoločnosti Google, ktorá publikuje online sprievodcu pre monetizáciu aplikácii [13].

V prieskume aplikácii v danej kategórii podľa mojich kritérií som nenarazil na žiadnu, ktorá by mala stratégiu zameranú na predávanie reklamného priestoru.

**Reklamný priestor**

Stratégiu, ktorá využíva reklamný priestor v aplikácii som vylúčil z dôvodu vplyvu na dojem z aplikácie, a taktiež si myslím, že aplikácia v takejto kategórii nebude mať tak široké publikum, aby mohla generovať nejaký významný zisk prostredníctvom takejto monetizačnej techniky.

**Freemium**

V tomto biznis modeli používatelia neplatia nič za inštaláciu aplikácie, ale je im ponúknutý voliteľný obsah s dodatočnými funkciami, ponuka ďalšieho obsahu alebo digitálny tovar [14].



Môj návrh spočíva v použití modelu Freemium, vďaka ktorému bude základná funkčnosť ponúknutá každému záujemcovi o obsah aplikácie s cieľom dať zákazníkovi čas na zoznámenie s aplikáciou a jej funkčnosťami. Z tejto stratégie vyplýva, že aplikácia bude ponúkať funkčnosť základného balíka (Free) a prémiového balíka (Pro) funkčností. Taktiež by som do biznis modelu zaviedol kredit model, ktorý by slúžil na zakúpenie dodatočného obsahu prostredníctvom mikro transakcií. Pomocou kreditového modelu by bolo možné zakúpenie napríklad nových psychologických testov, zobrazenie článkov alebo videí od expertov.

Všetko čo by hypoteticky mohol ponúkať základný a prémiový balík uvádzam v Tabuľka 8.

Tabuľka 8 Prehľad o funkciách v jednotlivých balíkoch

	<i>Free</i>	<i>Pro</i>
Koleso života	Áno	Áno
Energia	Áno	Áno
GROW	Áno	Áno
Magická realita	Áno	Áno
História jednotlivých procesov	Max 5 posledných	Neobmedzené
Push notifikácie	-	Áno
Kalendár	-	Áno
Synchronizácia dát v cloude	-	Áno
Export dát výsledkov do PDF/CSV	-	Áno

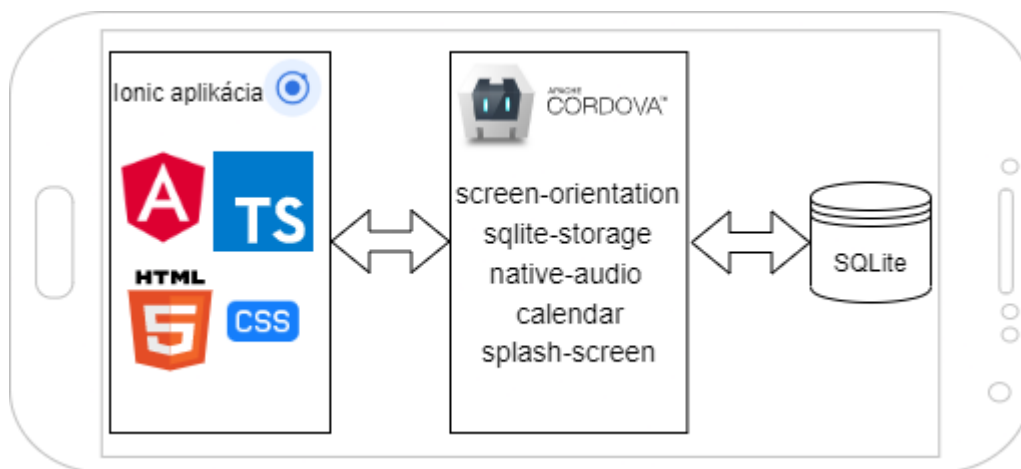
Pre nastavenie cien by sa vykonal obsiahlejší prieskum vzhľadom na ekonomickú povahu a rôzne iné faktory cieľového trhu. Predaje by som navrhoval stimulovať prostredníctvom časovo obmedzených cenových ponúk za predplatné a pre nových používateľov by som zacielenil kampaň s časovo obmedzeným balíčkom Pro zdarma.

## 4 IMPLEMENTÁCIA

V tejto sekcii je popísaný postup a spôsob implementácie mobilnej aplikácie. Mobilná aplikácia bola zhotovená ako hybridná s použitím Javascript frameworkov Ionic, Angular a Apache Cordova.

### 4.1 Architektúra aplikácie

Architektúra aplikácie je pomerne jednoduchá, keďže sa z aplikácie nedopytujú dáta z externého rozhrania. Vrstva užívateľského rozhrania je tvorená pomocou Javascript frameworkov Ionic a Angular, a taktiež pomocou HTML a CSS. Dáta z aplikácie sa ukladajú lokálne na úložisko do databázového systému SQLite, ktorý je prístupný v programovom kóde pomocou Apache Cordova modulu SQLite-Storage. Programový kód Ionic aplikácie ďalej používa rôzne iné moduly Apache Cordova, ktoré sú znázornené na Obrázok 17. Detailnejší popis použitých závislostí sa nachádza v sekcii 4.2.



Obrázok 17 Diagram architektúry aplikácie

## 4.2 Popis použitých knižníc vo vývoji aplikácie

Okrem hlavných vývojových nástrojov Ionic, Angular a Cordova boli použité dodatočné knižnice uvedené v Tabuľka 9.

Tabuľka 9 Zoznam použitých knižníc

#	Knižnica	Využitie
1	@ionic-native/native-audio	Použitie pri spúšťaní rôznych zvučiek v aplikácii.
2	@ionic-native/screen-orientation	Zamknutie orientácie obrazovky – Landscape/Portrait.
3	@ionic-native/sqlite	Prístup do SQLite databázy na zariadení.
4	@ionic-native/calendar	Umožňuje pridávať udalosti do aplikácie kalendár.
5	@ionic-native/splash-screen	Riadenie zobrazovania obrazovky pre načítanie aplikácie.
6	@ngx-translate	Knižnica pre internacionalizáciu aplikácie.
7	Moment.js	Knižnica na prácu s dátumom a časom v Javascript.
8	Rxjs	Knižnica pre reaktívne programovanie v Javascript.
9	Tslint	Knižnica pre statickú analýzu kódu.

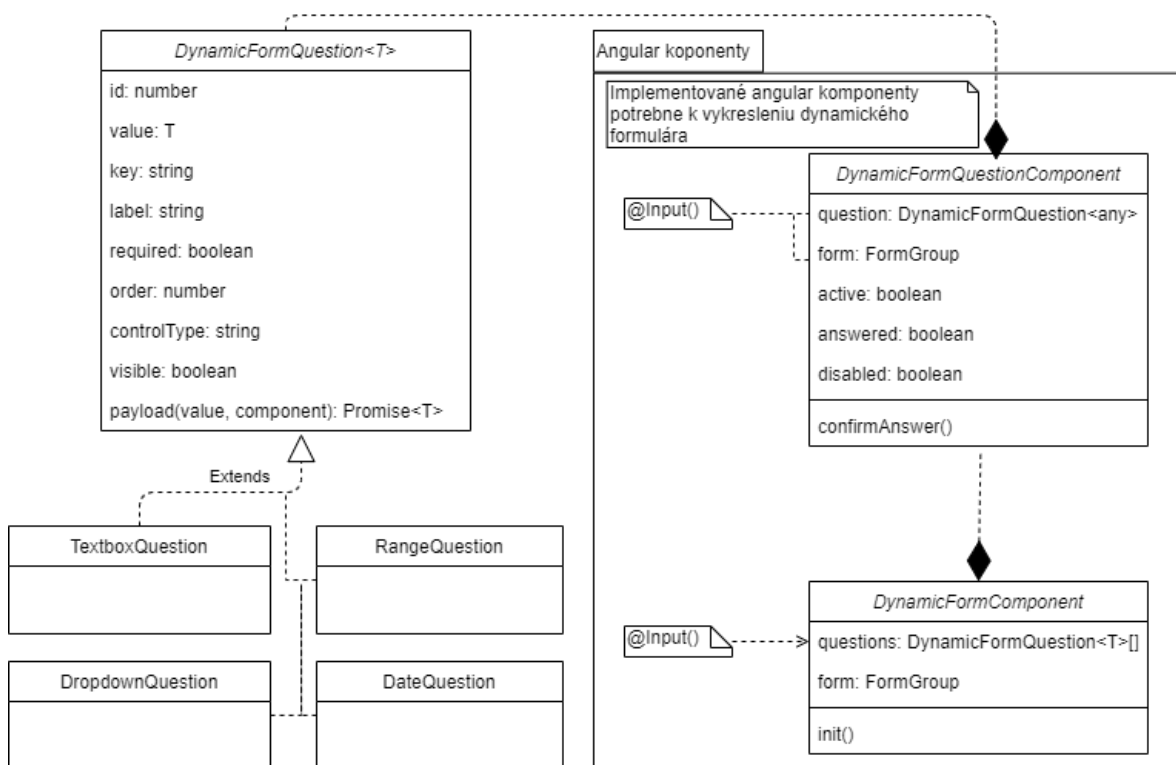
## 4.3 Implementácia metódy GROW

Popis zadania pre techniku GROW je popísaný v sekcii 3.1.2, 3.5.7 a konkrétny UC v sekcii 6.3. V nasledujúcich podkapitolách je priblížené technické riešenie sekcie techniky GROW, keďže pri jej implementácii bol využitý prístup vytvorenia dynamického formulára, ktorý všetky vstupné formulárové polia v tejto technike vypisuje podľa konfigurácie.

### 4.3.1 Motivácia vytvorenia dynamického formulára

Z analýzy a návrhu vyplynulo, že implementácia tejto techniky bude vyžadovať množstvo vstupných formulárových polí v jednotlivých krokoch ako Settings, Reality, Brainstroming a Will, keďže sa typy vstupných polí budú zobrazovať postupne po vyplnení predošlého polia a pod sebou. Ďalšia otázka musí byť vždy viditeľná až po vyplnení predošlej. Usúdil som, že by bolo najefektívnejšie mať polia zadané v konfigurácii a vykresľovať ich dynamicky. Z čoho nám vzniká dynamický formulár, ktorý je ľahko rozšíriteľný o ďalšie otázky, resp. vstupné polia. Tento prístup sa môže zovšeobecniť a implementovať obdobným spôsobom aj v iných sekciách. Veľké plus je, že pre takto definovaný formulár môžu tiecť dáta aj z rozhrania API, súboru alebo skriptu.

### 4.3.2 Popis implementácie



Obrázok 18 Diagram tried riešenia dynamického formulára

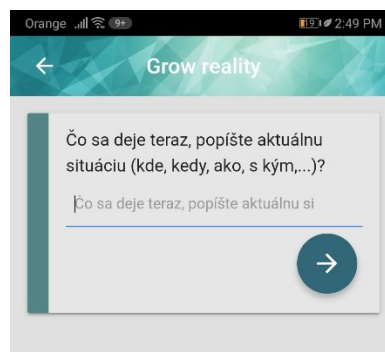
Diagram tried v obrázku vyššie popisuje spôsob implementácie a závislostí medzi základnými triedami a Angular komponentami. Pri implementácii v technike GROW sa používa komponent `DynamicFormComponent`, do ktorej naplníme konfiguráciu dynamického formulára prístupom cez Angular `Input`<sup>1</sup> dekorátor. Komponent `DynamicFormComponent` je vložená do jednotlivých krokov `Settings`, `Reality`, `Brainstroming` a `Will`.

<sup>1</sup> Angular Input decorator <https://angular.io/api/core/Input>

```
1 {
2   "questions": [
3     {
4       "id": 1,
5       "key": "situation",
6       "label": "Čo sa deje teraz, popíšte aktuálnu situáciu",
7       "value": "",
8       "required": true,
9       "order": 1,
10      "controlType": "textbox"
11    }
12  ]
13 }
```

Obrázok 19 Definícia položky dynamického formuláru v JSON formáte

Obrázok 19 je ukážkou definovania položky pre dynamický formulár formátom JSON. Dáta v takomto formáte je následne možné mapovať na typ `DynamicFormQuestion`. Následne sa takto definovaná otázka vykreslí s textovým poľom ako zachytáva Obrázok 20.



Obrázok 20 Vykreslenie položky z dynamického formulára

Spracovanie konfigurácie s definovanými otázkami a následné vykreslenie do UI sa vykonáva v komponente `DynamicFormQuestionComponent`. Tento komponent rozhoduje o vykreslení aktuálne podporovaných formulárových vstupných polí typu posuvník<sup>2</sup>, text, textarea<sup>3</sup> a dátum. Pre tieto vstupné polia boli použité preddefinované Ionic komponenty `IonRange`, `IonInput`, `IonTextArea`, `IonDatetime`. To, aký typ poľa sa zobrazí sa riadi hodnotou

<sup>2</sup> Ionic komponent `Ion-Range` <https://ionicframework.com/docs/api/range>

<sup>3</sup> Ionic komponent `Ion-textarea` <https://ionicframework.com/docs/api/textarea>

z atribútu `controlType`. Prístupovanie k dátam formulára je možné cez inštanciu `FormGroup` uchovanej v atribúte `form` v komponente `DynamicFormComponent`.

### 4.3.3 Záver

Implementácia dynamického formulára otvorila možnosti ako aplikáciu doplniť o sadu ďalších testov, napr. z externého zdroja prístupom REST API alebo zo súboru. Taktiež funkčnosť jednotlivých dynamických vstupných polí je možné jednoducho rozšíriť vďaka aktuálnej modulárnej implementácii a návrhu.

## 4.4 Ukladanie dát

Dôležitou časťou aplikácie je ukladanie výsledkov dokončených testov z dôvodu historického prehľadu o výsledkoch. Taktiež sa ukladajú užívateľské nastavenia aplikácie ako napr. jazyk užívateľského rozhrania. Dáta, ktoré aplikácia ukladá sú uložené len na klientskom zariadení pomocou databázového systému SQLite. V procese návrhu boli identifikované nasledujúce entity, ktoré vystupujú v špecifikácii štruktúry databázy.

Tabuľka 10 Typy entít v štruktúre databázy

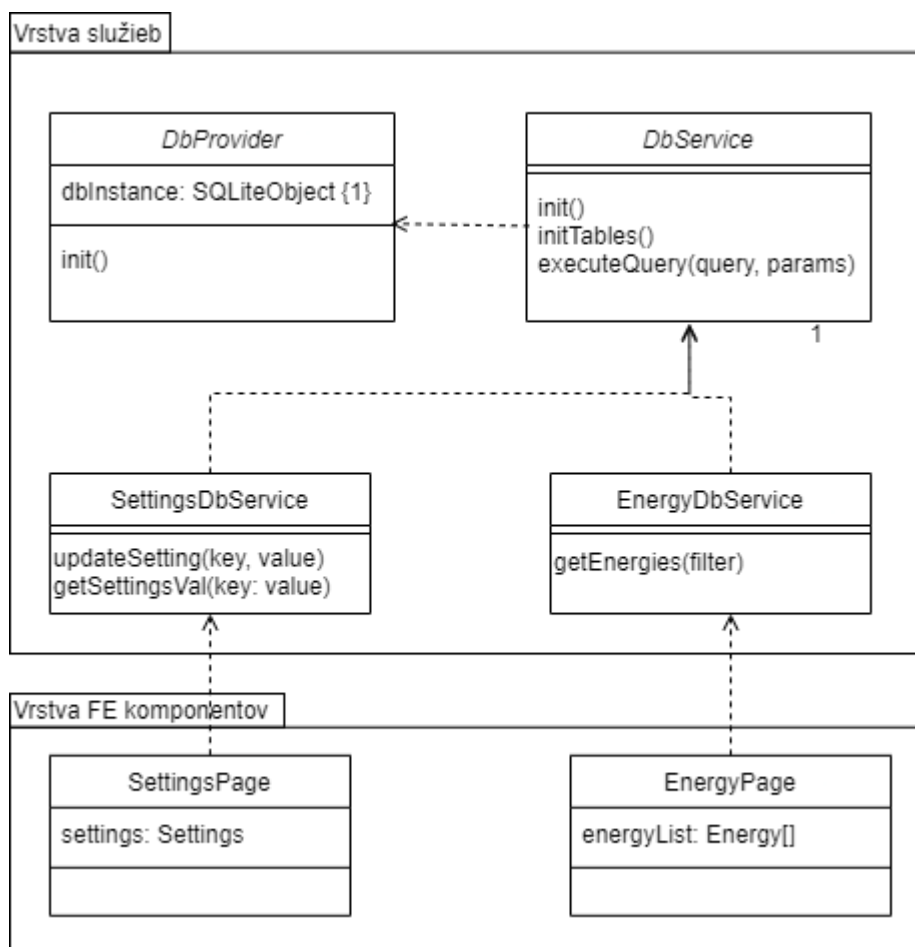
<b>Entita</b>	<b>Popis</b>
USER_SETTINGS	<i>Ukladanie nastavení aplikácie. Využíva sa pre uloženie prvej návštevy aplikácie, nastavenie jazyka.</i>
WHEEL_OF_LIFE	Ukladanie výsledkov z techniky Koleso životnej rovnováhy.
ENERGY	Ukladanie výsledkov z techniky Energia.
GROW	Ukladanie výsledkov z techniky GROW.
MAGIC_REALITY	Ukladanie výsledkov z techniky Magická realita.

### 4.4.1 Dátový model

Kompletný fyzický dátový model sa nachádza v prílohe P I. Zachytáva všetky tabuľky, pomocné tabuľky, vzťahy medzi entitami a špecifiká dátových typov potrebných k realizácii v systéme SQLite.

#### 4.4.2 Popis implementácie prístupu k databáze

Riešenie používa k prístupu k databáze databázový systém SQLite. Používa sa rozšírenie SQLite<sup>4</sup> zo štandardnej ponuky Ionic modulov. Prístupovanie jednotlivých služieb z programového kódu k databáze je implementované pomocou prístupu Angular Dependency injection<sup>5</sup>. Implementácia a použitie databázy spočíva v dvoch logických vrstvách, ktoré sú znázornené v diagrame tried Obrázok 21 Diagram tried implementácie a použitia.



Obrázok 21 Diagram tried implementácie a použitia databázy SQLite

Pri implementácii prístupu k databáze som mal v úmysle rozdeliť úlohy a prístup k databáze tak, aby sa z programového kódu jednotlivých UI komponent nevykonávali žiadne SQL príkazy. Konkrétne databázové operácie vyvoláva vrstva služieb, ktorá sa viaže na kontext UI

<sup>4</sup> Docs Ionic SQLite <https://ionicframework.com/docs/native/sqlite/>

<sup>5</sup> Angular Dependency injection <https://angular.io/guide/dependency-injection>

komponent, teda napr. SettingsDbService, EnergyDbService, atď. Táto vrstva služieb prístupuje pomocou dependency injection k službe DbService a vyvoláva jej metódu executeQuery. Metóda executeQuery pristúpi k inštancii SQLiteObject uchovanej v službe DbProvider a vykoná nad databázou potrebnú operáciu a vráti výsledky formou triedy Promise.

Tabuľka 11 Popis jednotlivých služieb v implementácii prístupu k databáze.

<b>Trieda</b>	<b>Popis</b>
DbProvider	Prístup k rozhraniu rozšírenia SQLite. Vyvoláva metódu create nad službou SQLite, čím sa zabezpečí otvorenie alebo vytvorenie SQLite súboru na zariadení. Uchováva inštanciu SQLiteObject pre prístup k DB, ktorá sa ďalej distribuuje pomocou dependency injection.
DbService	Pristupuje k inštancii SQLiteObject. Vykonáva vytvorenie celej štruktúry databázy metódou init(). Pomocou verejnej metódy executeQuery, umožňuje vykonávať príkazy nad databázovým súborom.
SettingsDbService	Implementuje konkrétne operácie nad databázou potrebné v sekcii Nastavenia aplikácie. Ukladá do databázy a získava z nej nastavenia.
Modul <názov>DbService	Obdobné služby ako SettingsDbService, ktoré implementujú konkrétne operácie k čítaniu a zápisu nad databázou, potrebné v konkrétnych moduloch.



Ukážka použitia prístupu k databázovej položke je predstavená v ukážke kódu v Obrázok 22 Ukážka kódu metódy služby SettingsDbService. Táto ukážka zachytáva metódu `getSettingVal` slúžiacu pre načítanie uloženého nastavenia aplikácie. Táto metóda vyvoláva `executeQuery` zo služby `DbService` a spracúva výsledok vykonania SQL príkazu nad tabuľkou `USER_SETTINGS`.

```
52  /**
53   * Metoda vrati hodnotu ulozeného nastavenia.
54   * @param key predstavuje kluc k ulozenej hodnote nastavenia.
55   */
56  getSettingVal(key: string): Promise<string> {
57    const dbQuery = `select * from user_settings where setting_key = ?`;
58    return this.dbService.executeQuery(dbQuery, [key]).then(response => {
59      let result = null;
60      if (response != null && response.rows.length > 0) {
61        result = response.rows.item(0)['SETTING_VALUE'];
62      }
63      return result;
64    });
65  }
```

Obrázok 22 Ukážka kódu metódy služby SettingsDbService

## 4.5 Zostavenie aplikácie

Aplikáciu je možné zostaviť pre platformy Android, iOS, a taktiež je možné aplikáciu spúšťať vo webovom prehliadači. Vývoj aplikácie bol v prvej fáze zameraný na funkčnosť na Androide a tak je možné, že niektoré funkčnosti nie sú správne odladené na platforme iOS.

### 4.5.1 Testovanie a vývoj

Počas vývoja sa aplikácia spúšťala prevažne v internetovom prehliadači príkazom *ionic serve*. Pre vývoj to znamená jednoduché a rýchle otestovanie aktuálne vyvíjanej funkčnosti. Avšak množstvo Ionic Native modulov v tomto režime nepracujú a je nutné aplikáciu spustiť v mobilnom emulátore alebo priamo na zariadení pomocou módu USB ladenie. [3]

### 4.5.2 Produkcia

Produkčné zostavenie Ionic aplikácie pre Android je možné pomocou príkazu *ionic cordova build android --prod --release*. Tento príkaz minimalizuje programový kód aplikácie, odstráni funkčnosť ladenia aplikácie a vygeneruje súbor aplikácie formátu APK. [3]

Produkčné zostavenie pre iOS je možné pomocou príkazu *ionic cordova build ios --prod*. Následne sa vygenerujú potrebné súbory do priečinka *platforms/ios* a otvorením súboru *.xcodeproj* v programe Xcode spustíme aplikáciu na zariadení iOS. [3]

### 4.5.3 Požiadavky

Pre zostavenie produkčného Android zostavenia sú potrebné nasledujúce závislosti. [3]

1. Java JDK 8
2. Android Studio
3. Aktualizované Android SDK nástroje

## 4.6 Testovanie aplikácie

Testovanie aplikácie prebiehalo manuálnym funkčným testovaním počas vývoja aplikácie, a taktiež počas demo stretnutí formou tzv. Usability testingu s mentorom životného koúčingu. Testovanie už počas vývoja bolo zamerané na odhalenie rôznych nedostatkov aplikácie, akými sú napr. náročnosť pri používaní aplikácie, správnosť textov, navigácia medzi sekciami, dizajn a funkčnosť aplikácie.

### 4.6.1 Príprava

Počas demo stretnutí s mentorom sa pri testovaní vychádzalo z vopred špecifikovaných testovacích scenárov, pričom príklad jedného z nich je znázornený v Tabuľka 12. Testovacích scenárov bolo omnoho viac, spomenuté sú teda iba niektoré.

Tabuľka 12 Tabuľka s testovacími scenármi.

#	Scenár	Predmet testovania
1	Funkčnosť hlavnej navigácie v mobile	Navigácia
2	Funkčnosť hlavnej navigácie v tablete	Navigácia
3	Nastavenie jazyka SK/EN	Nastavenia
4	Funkčnosť flowu Koleso života	Koleso životnej rovnováhy
5	Historický prehľad z prejdených flowov Kolesa života	Koleso životnej rovnováhy
6	Správnosť lokalizačných textov SK/EN	Celá aplikácia
7	Zobrazenie interaktívneho obrázku batéria	Sekcia Energia

### 4.6.2 Vykonanie

Vykonávanie testov bolo zamerané na otestovanie kompatibility aplikácie na rôznych zariadeniach, a preto sa jednotlivé testovacie scenáre vykonávali manuálne na viacerých zariadeniach.

### Otestovanie kompatibility aplikácie

Testovanie kompatibility prebiehalo na smartfónoch, a taktiež na tabletoch. Hoci vo fáze vývoja aplikácie bolo rozhodnuté, že aplikácia bude v prvej fáze uvoľnená najprv na platformu Android, aplikácia sa taktiež testovala na zariadeniach so systémom iOS. Bolo dôležité otestovať ako sa jednotlivé funkčnosti budú správať na rôznych zariadeniach. Taktiež

vzhľadom na adaptívny dizajn hlavnej navigácie aplikácie, sa pri testovaní kládol dôraz na otestovanie rôzneho zobrazenia menu na mobile a tablete.

Tabuľka 13 Zoznam otestovaných zariadení

<i>Zariadenie</i>	<i>Verzia</i>
Honor 9	Android 9.0 Pie
Xiaomi MI Note 2	Android 8.0 Oreo
Lenovo M10	Android 9.0 Pie
iPad Pro	iOS 9.3.5
iPhone 7 Plus	iOS 10.0.1

### **Odhalené problémy z testovania**

Ako dôkaz užitočnosti testovania aplikácie uvádzam niektoré defekty, ktoré sa počas testovania odhalili. Po odhalení sa jednotlivé defekty evidovali v systéme Trello<sup>6</sup> a následne sa opravili.

Tabuľka 14 Popis nájdených chýb počas testovania

<i>#</i>	<i>Popis defektu</i>	<i>Riešenie</i>
1	Pád aplikácie pri nedostatku miesta v pamäti telefónu.	Ošetrovanie, zobrazenie upozornenia v aplikácii.
2	Rozhádzané zobrazenie kola života v iOS.	Úprava CSS štýlov.
3	Nefunkčný fokus na ďalšie vstupné pole po vyplnení poľa dynamického formulára.	Oprava správania Angular komponenty
4	Aplikácia zobrazuje slovenský preklad, aj keď je nastavená angličtina.	Doplnenie anglických textácií do lokalizačného súboru.
5	Pre užívateľa nebolo jasné, čím má potvrdiť formulár.	Naštýlovanie niektorých potvrdzovacích tlačidiel v pulzujúcom štýle.

<sup>6</sup> Trello – webová aplikácia na správu projektov <https://trello.com/>

## 4.7 Internacionalizácia aplikácie

Jednou z nefunkčných požiadaviek pre aplikáciu bola internacionalizácia, teda podpora rôznych lokalizácií v aplikácii. Pre prvú fázu aplikácie bolo rozhodnuté o podpore angličtiny a slovenčiny. Z pohľadu klienta je možné si jazyk aplikácie nastaviť jednoducho v sekcii Nastavenia/Settings, avšak štandardne sa predvolí angličtina pri prvom spustení aplikácie.

### 4.7.1 Popis funkčnosti

Pre implementáciu bola použitá knižnica ngx-translate<sup>7</sup>. Táto knižnica bola vybratá z dôvodu podpory pre framework Angular a možnosti jednoduchej zmeny lokalizácie počas spustenia aplikácie.

Lokalizačné texty sú umiestnené v priečinku assets/i18n. Tie sa načítavajú pomocou služby TranslateService. Následne sa v programovom kóde HTML šablón používa pipe<sup>8</sup> TranslatePipe, tento operátor zaistí preloženie kľúča na lokalizačný text.

## 4.8 Spôsob rozširovania aplikácie

Aplikáciu je možné rozširovať o ďalšie sady testov štandardnou aktualizáciou, presnejšie doprogramovaním ďalších nových funkčnosti a následné nové zostavenie aplikácie. Riešenie dynamického formulára popísané v sekcii 4.3.1, nám otvára možnosti správy testov pomocou API, ktoré by hypoteticky mohli byť dostupné on-line a pomocou REST rozhrania by sa načítavala konfigurácia pre jednotlivé testy v JSON formáte bez potreby štandardnej aktualizácie. Momentálne sa táto konfigurácia definuje v staticky uložených súboroch, pre budúcnosť je ale možné tento prístup vylepšiť pomocou prístupu do API pomocou HTTP protokolu.

---

<sup>7</sup> Ngx-translate <http://www.ngx-translate.com/>

<sup>8</sup> Angular Pipe - <https://angular.io/guide/pipes>

## ZÁVER

Aplikáciu pre podporu osobnostného rastu sa mi podarilo úspešne vytvoriť s funkčným zostavením pre Android a iOS. Vytvoreniu samotnej aplikácie predchádzali rôzne fázy prieskumu procesov v danej problematike. Osobné konzultácie s mentorom životného koučingu boli veľmi prínosné pre pochopenie problematiky a následné stanovenie požiadaviek pre aplikáciu bolo prienikom mojich technických znalostí a mentorových znalostí z oblasti koučingu. Po špecifikovaní požiadaviek na aplikáciu bolo následne vykonaná porovnanie aktuálnych nástrojov pre tvorbu mobilných aplikácií, ktoré dokážu zostaviť aplikáciu pre platformy Android a iOS z jedného programového kódu. S mojimi skúsenosťami webového vývojára som inklinoval k použitiu nástrojov, ktoré využívajú webové technológie, a tak som sa rozhodol pre použitie frameworkov Ionic a Angular. Porovnanie jednotlivých technológií v rôznych ukazovateľoch sa mi podarilo zachytiť v sekcii Návrh mobilnej aplikácie. Pred programovaním aplikácie bola veľmi dôležitá fáza prototypovania jednotlivých obrazoviek a návrh dizajnu aplikácie. Na základe teórie o jednotlivých technikách boli špecifikované prípady použitia, vďaka ktorým som vedel ako implementovať jednotlivé techniky, ktoré môže užívateľ v aplikácii vykonávať. V tejto fáze boli jednotlivé prípady použitia komunikované s mentorom a po odsúhlasení implementované do aplikácie. Výstupom je aplikácia, ktorá môže pomôcť užívateľovi dosiahnuť vysnívané ciele, plánovať akčné kroky, nájsť životnú rovnováhu alebo pomôcť pri nájdení jeho silných alebo slabých stránok.

Aktuálny stav aplikácie predstavuje prvú fázu vývoja a je plánované jej vylepšovanie o ďalšie funkčnosti. Aplikácia by sa mohla v budúcnosti rozšíriť o nasledujúce funkčnosti:

- Spôsob rozšírenia aplikácie o ďalšie testy, navrhnutý v sekcii 4.3.3.
- Spôsobu monetizácie aplikácie, navrhnutý v sekcii 3.6.3.
- Export dát do PDF/CSV
- Synchronizácia dát v cloude
- Sekcia s článkami od expertov
- Podpora push notifikácií

**ZOZNAM POUŽITEJ LITERATÚRY**

- [1] C. Griffith, Mobile App Development with Ionic, 1st Edition, 2017.
- [2] R. Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, 2019.
- [3] „Ionic Framework,“ Január 2020. [Online]. Available: <https://ionicframework.com/docs>.
- [4] S. Indermohan a P. Hoc, Ionic Cookbook Third Edition, Packt Publishing Ltd., 2018.
- [5] T. A. S. Foundation, „Apache Cordova - Overview,“ [Online]. Available: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>. [Cit. 7 2020].
- [6] G. Saini, Hybrid Mobile Development with Ionic, Packt Publishing Ltd., 2017.
- [7] „Flutter Docs,“ Január 2020. [Online]. Available: <https://flutter.dev/docs>.
- [8] N. Docs, 2020. [Online]. Available: <https://docs.nativescript.org>.
- [9] D. S. Menendez a P. Williams, Becoming a Professional Life Coach: Lessons from the Institute of Life Coach Training, 2015.
- [10] D. Amory, Personal Coaching, Edgard Adriaens, Belgium, 2011.
- [11] G. e. a. APPEL, „On the monetization of mobile apps,“ 2019. [Online]. Available: <https://msbfile03.usc.edu/digitalmeasures/gappel/intellcont/AppelLibaiMullerShachar2019-1.pdf>. [Cit. 7 2020].
- [12] „Earn more revenue with the right monetization options,“ 6 2020. [Online]. Available: <https://developer.android.com/distribute/best-practices/earn/monetization-options>. [Cit. 6 2020].
- [13] Google, „Monetization options,“ [Online]. Available: <https://developer.android.com/distribute/best-practices/earn/monetization-options>. [Cit. 6 2020].

- [14] Apple, „Freemium Business Model,“ [Online]. Available: <https://developer.apple.com/app-store/freemium-business-model/>. [Cit. 7 2020].
- [15] Google, „Design & Quality,“ Január 2020. [Online]. Available: <https://developer.android.com/docs/quality-guidelines>.
- [16] Apple, „iOS Design Themes,“ Január 2020. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>.
- [17] „Ionic plugin SQLite,“ [Online]. Available: <https://ionicframework.com/docs/v3/native/sqlite/>.
- [18] JetBrains, „The State of Developer Ecosystem 2020,“ 2020. [Online]. Available: <https://www.jetbrains.com/lp/devecosystem-2020/>. [Cit. 7 2020].



**ZOZNAM POUŽITÝCH SKRATIEK A SYMBOLOV**

Angular	Aplikačný rámec s kolekciou HTML/CSS/JS nástrojov na tvorbu webových aplikácií.
Angular komponent	Najzákladnejší stavebný blok UI v Angular aplikácii.
Cloud	Internetová služba dostupná odkiaľkoľvek.
CPU	Základná jednotka počítača - procesor.
Dekorátor	Štrukturálny návrhový vzor.
Dependency injection	Návrhový vzor k poskytnutiu externých závislostí softvérového komponentu.
Framework	Aplikačný rámec.
Github Stars	Indikátor popularity Open source projektov na platforme Github.
GPU	Procesor grafickej karty.
IOS	Operačný systém firmy Apple pre mobilnú platformu.
JSON	Javascript Object Notation je spôsob zápisu dát, ktorý je platformovo nezávislý.
Open source	Open source sa nazývajú programy, ktorých zdrojový kód je voľne šíriteľný.
Push notifikácie	Správy, ktoré sa generujú z nainštalovaných aplikácií.
Rest API	Representational State Transfer je dátovo orientovaná architektúra.
SDK	Sada nástrojov pre vývoj software.
SQL	Structured Query Language - štrukturovaný dotazovací jazyk používaný v relačných databázach.
UI	User Interface - užívateľské rozhranie.
WebView	Android komponent, ktorý slúži na zobrazenie webového obsahu v nainštalovanej aplikácii.

**ZOZNAM OBRÁZKOV**

Obrázok 1 Ukážka architektúry hybridnej aplikácie .....	13
Obrázok 2 Architektúra Flutter .....	17
Obrázok 3 Architektúra NativeScript .....	18
Obrázok 4 Architektúra Ionic framework .....	19
Obrázok 5 Porovnanie počtu vyhľadávani v Google .....	22
Obrázok 6 Výsledok prieskumu používania frameworkov vývojármi od spoločnosti JetBrains v roku 2020 .....	23
Obrázok 7 Fázy GROW modelu .....	29
Obrázok 8 Príklad Kolesa životnej rovnováhy .....	30
Obrázok 9 Diagram prípadov použitia .....	32
Obrázok 10 Tok obrazoviek .....	33
Obrázok 11 Návrh hlavnej ponuky pre mobil .....	34
Obrázok 12 Návrh hlavnej ponuky pre tablet .....	34
Obrázok 13 Návrh procesu techniky Koleso životnej rovnováhy .....	35
Obrázok 14 Návrh procesu techniky Energia .....	36
Obrázok 15 Návrh techniky Magická realita .....	37
Obrázok 16 Návrh procesu techniky GROW .....	38
Obrázok 17 Diagram architektúry aplikácie .....	42
Obrázok 18 Diagram tried riešenia dynamického formulára .....	44
Obrázok 19 Definícia položky dynamického formuláru v JSON formáte .....	45
Obrázok 20 Vykreslenie položky z dynamického formulára .....	45
Obrázok 21 Diagram tried implementácie a použitia databázy SQLite .....	47
Obrázok 22 Ukážka kódu metódy služby SettingsDbService .....	49

**ZOZNAM TABULIEK**

Tabuľka 1 Prehľad frameworkov pre vývoj multiplatformových aplikácií.....	14
Tabuľka 2 Porovnanie počtu dodatočných balíčkov .....	20
Tabuľka 3 Porovnanie počtu vytvorených otázok v Stackoverflow pre jednotlivé frameworky .....	20
Tabuľka 4 Porovnanie počtu Twitter sledovateľov .....	21
Tabuľka 5 Porovnanie počtu Github Stars.....	21
Tabuľka 6 Porovnanie času spustenia základných aplikácií.....	25
Tabuľka 7 Porovnanie veľkosti zostavených aplikácií.....	26
Tabuľka 8 Prehľad o funkciách v jednotlivých balíkoch.....	41
Tabuľka 9 Zoznam použitých knižníc .....	43
Tabuľka 10 Typy entít v štruktúre databázy .....	46
Tabuľka 11 Popis jednotlivých služieb v implementácii prístupu k databáze.....	48
Tabuľka 12 Tabuľka s testovacími scenármi.....	51
Tabuľka 13 Zoznam otestovaných zariadení .....	52
Tabuľka 14 Popis nájdených chýb počas testovania .....	52

## ZOZNAM PRÍLOH

Príloha P I: Fyzický dátový model .....	I
Príloha P II: Prípady použitia.....	II

## 5 PRÍLOHA P I: FYZICKÝ DÁTOVÝ MODEL



## **6 PRÍLOHA P II: PRÍPADY POUŽITIA**

### **6.1 UC 1.0 Vykonanie techniky Kolesa životnej rovnováhy**

#### **6.1.1 Stručný popis**

Prípad použitia popisuje funkčnosť techniky Koleso životnej rovnováhy.

#### **6.1.2 Podmienky pre spustenie**

Aplikácia musí byť spustená

#### **6.1.3 Hlavný tok**

1. Používateľ zvolí v menu položku Koleso života.
2. Systém zobrazí obrazovku SCR1-01 Výber oblasti na riešenie.
3. Používateľ zvolí z ponuky oblasť, ktorú chce riešiť.
4. Systém zobrazí obrazovku SCR1-02 Nastavenie kolesa.
5. Používateľ priradí jednotlivým oblastiam hodnotu 1 až 10.
6. Používateľ potvrdí priradené hodnoty tlačidlom Pokračovať.
7. Systém zobrazí obrazovku SCR1-03 Koleso životnej rovnováhy.
8. Používateľ potvrdí nastavenie kolesa tlačidlom Pokračovať.
9. Systém zobrazí obrazovku SCR1-04 Oblasti s najmenším skóre.
10. Používateľ jednu zvolí oblasť, ktorú chce zlepšiť a potvrdí.
11. Systém zobrazí obrazovku SCR1-05 Akčné kroky.
12. Používateľ vyplní vstupné polia a potvrdí.
13. Systém zobrazí obrazovku SCR1-06 Koleso životnej rovnováhy – Výsledok.
14. Používateľ obrazovku potvrdí tlačidlom Koniec.

#### **6.1.4 Alternatívne toky**

##### **1a – Používateľ zvolí možnosť zaslať email**

1. Bod 13 hlavného toku.
2. Používateľ zvolí možnosť E-mail.
3. Systém spustí funkčnosť lokálneho email klienta s funkciou nový email a telom emailu, ktorý bude obsahovať zadané dáta a obrázkom kolesa.

4. Používateľ vyplní vstupné polia a potvrdí odoslanie emailu.
5. Prípád použitia pokračuje krokom 13 hlavného toku.

## **6.2 UC 2.0 Vykonanie techniky Energia**

### **6.2.1 Stručný popis**

Prípád použitia popisuje funkčnosť techniky Energia.

### **6.2.2 Podmienky pre spustenie**

Aplikácia musí byť spustená.

### **6.2.3 Hlavný tok**

1. Používateľ zvolí možnosť v menu Energia.
2. Systém zobrazí obrazovku SCR2-01 Nastavenie energie.
3. Používateľ nastaví hodnotu 0 až 100 na slideri a potvrdí.
4. Systém zobrazí obrazovku SCR2-02 Zdroje energie.
5. Používateľ vyplní vstupné polia pre zadanie zdrojov energie a potvrdí.
6. Systém zobrazí obrazovku SCR2-03 Vysávače energie.
7. Používateľ vyplní vstupné polia pre zadanie vysávačov energie a potvrdí.
8. Systém zobrazí obrazovku SCR2-04 Zvolenie činnosti.
9. Používateľ zvolí činnosť, ktorá mu odoberá energiu najviac.
10. Systém zobrazí modálne okno SCR2-05 Modálne okno rozdelenie oblasti.
11. Používateľ zvolí možnosť pokračovať.
12. Systém zobrazí obrazovku SCR2-06 Nastavenie akčných krokov.
13. Používateľ vyplní akčné kroky.
14. Používateľ nastaví termín vykonania krokov a potvrdí.
15. Systém zobrazí obrazovku SCR2-07 Energia výsledok.
16. Systém uloží zadané údaje.
17. Používateľ obrazovku potvrdí tlačidlom Koniec

### **6.2.4 Alternatívne toky**

#### **2a. Rozdelenie oblasti na menšie**

1. Bod 10 hlavného toku.

2. Používateľ zvolí možnosť rozdeliť.
3. Systém zobrazí obrazovku SCR2-08 Rozdelenie oblasti.
4. Používateľ vyplní a potvrdí.
5. Prípád použitia pokračuje krokom 11 hlavného toku

## **2b. Nastavenie termínu v kalendári zariadenia**

1. Bod 14 hlavného toku.
2. Používateľ klikne na ikonku kalendára.
3. Systém spustí natívnu funkčnosť Pridanie udalosti do kalendára a vopred vyplní s dátumom, ktorý užívateľ zvolil ako termín vykonania krokov.
4. Používateľ nastaví udalosť do kalendára a potvrdí.
5. Systém zobrazí natívnu funkčnosť Kalendár s nastavenou udalosťou.
6. Používateľ zvolí tlačidlo na návrat späť.
7. Prípád použitia pokračuje krokom 14 hlavného toku



## **6.3 UC 3.0 GROW**

### **6.3.1 Stručný popis**

Prípád použitia popisuje funkčnosť techniky GROW.

### **6.3.2 Podmienky pre spustenie**

Aplikácia musí byť spustená.

### **6.3.3 Hlavný tok**

1. Používateľ zvolí možnosť v menu GROW.
2. Systém zobrazí obrazovku SCR3-01 GROW úvod.
3. Používateľ potvrdí obrazovku tlačidlom Ďalej.
4. Systém zobrazí obrazovku SCR3-02 Nastavenie cieľa.
5. Používateľ vyplní postupne vstupné polia a potvrdí nastavenie cieľa.
6. Systém zobrazí obrazovku SCR3-03 GROW Realita.
7. Používateľ vyplní postupne vstupné polia a potvrdí nastavenie reality.
8. Systém zobrazí obrazovku SCR3-04 GROW Možnosti.
9. Používateľ vyplní postupne formulár a potvrdí nastavenie možností.
10. Systém zobrazí obrazovku SCR4-05 GROW prehľad možností.
11. Používateľ obrazovku potvrdí tlačidlom Ďalej.
12. Systém zobrazí obrazovku SCR4-06 GROW Brainstorming
13. Používateľ ku každej zadanej hodnote z obrazovky SCR4-04 vyplní prínosy a nedostatky a potvrdí.
14. Systém zobrazí obrazovku SCR4-07 GROW Vôľa.
15. Používateľ vyplní postupne formulár a potvrdí.
16. Systém zobrazí modálne okno SCR4-08 GROW Rozdelenie úlohy potvrdenie.
17. Používateľ zvolí možnosť Áno.
18. Systém zobrazí obrazovku SCR4-08 GROW Výsledok.
19. Systém uloží zadané dáta.
20. Používateľ zvolí možnosť Koniec.

### **6.3.4 Alternatívne toky**

#### **3a. Preformulovanie cieľa**

1. Bod 4 hlavného toku.
2. Systém zobrazí po vyplnení textového poľa Čo chcete dosiahnuť modálne okno SCR4-09 Preformulovanie cieľa.
3. Používateľ zvolí možnosť Nie.
4. Systém vyzve používateľa znovu vyplniť textové pole Čo chcete dosiahnuť.
5. Prípád použitia pokračuje krokom 4 hlavného toku.

#### **3b. Vplyv na cieľ**

1. Bod 4 hlavného toku.
2. Používateľ nastavil pre pole Vplyv na cieľ hodnotu menšiu ako 8.
3. Systém zobrazí po nastavení hodnoty poľa Vplyv na cieľ modálne okno SCR4-10 Preformulovanie problému.
4. Používateľ zvolí možnosť Prehodnotiť vplyv na cieľ.
5. Prípád použitia pokračuje krokom 4 hlavného toku vyplňaním poľa Vplyv na cieľ.

#### **3c. Vplyv na cieľ**

1. Bod 3 alternatívneho toku 3b
2. Používateľ zvolí možnosť Preformulovať cieľ.
3. Prípád použitia pokračuje krokom 4 hlavného toku vyplňaním textového Čo chcete dosiahnuť.

#### **3d. Prehodnotenie ochoty**

1. Bod 14 hlavného toku.
2. Používateľ nastavil pre pole Ochota hodnotu menšiu ako 8.
3. Systém zobrazí modálne okno SCR4-11 Prehodnotenie ochoty.
4. Používateľ zvolí možnosť Prehodnotiť svoju ochotu.
5. Prípád použitia pokračuje krokom 14 hlavného toku vyplňaním poľa Ochota.

#### **3e. Prehodnotenie ochoty**

1. Bod 3 alternatívneho toku 3d.
2. Používateľ zvolí možnosť Preformulovať čo chcem urobiť.

3. Prípád použitia pokračuje krokom 14 hlavného toku vyplňaním textového poľa Čo hodláte urobiť.

### **3f. Rozdelenie riešenia na viac úloh**

1. Bod 18 hlavného toku.
2. Používateľ zvolí možnosť Nie.
3. Systém zobrazí obrazovku SCR4-12 Rozdelenie riešenia na časti.
4. Používateľ vyplní formulár na rozdelenie riešenia do viacerých častí.
5. Používateľ potvrdí formulár.
6. Prípád použitia pokračuje krokom 18 hlavného toku.

### **3g. Odoslať email**

1. Bod 18 hlavného toku.
2. Používateľ zvolí možnosť Email.
3. Systém spracuje zadané dáta a otvorí natívneho emailového klienta a vopred vyplní telo emailu zadanými dátami z procesu techniky GROW.
4. Používateľ vyplní potrebné dáta na odoslanie e-mailu a potvrdí.
5. Systém zatvorí emailového klienta.
6. Prípád použitia pokračuje krokom 18 hlavného toku.