



**ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE**

**F3**

**Fakulta elektrotechnická  
Katedra počítačů**

**Bakalářská práce**

# **Pomoc se zvládnáním depresí**

**Lukáš Novák**

**Softwarové inženýrství a technologie**

**květen 2023**

**Vedoucí práce: Ing. Božena Mannová, Ph.D.**



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novák** Jméno: **Lukáš** Osobní číslo: **484961**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Pomoc se zvládním depresí**

Název bakalářské práce anglicky:

**Help with depression**

Pokyny pro vypracování:

1. Seznamte se s problematikou depresí a možnostmi s nimi bojovat.
2. Proveďte analýzu dostupných IT projektů, které se zabývají problematikou depresi a pomáhají ji zvládat.
3. Na základě provedené analýzy navrhnete základní funkcionality navrhované aplikace, která se bude využívat principů kognitivní terapie, která se bude zaměřovat na myšlenkové pochody.
4. Zvolte architekturu aplikace a vyberte nejvhodnější technologie pro implementaci. Výběr technologií zdůvodněte.
5. Aplikaci implementujte a otestujte včetně uživatelských testů.
6. Zhodnoťte výsledky a navrhnete případné další funkcionality nebo jiná zlepšení.
7. Při řešení využijte vhodných prostředků softwarového inženýrství.

Seznam doporučené literatury:

- [1] Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach , ISBN-10: 9780078022128
- [2] U.S. Department of Health and Human Services, National Institutes of Health, National Institute of Mental Health. (2021). Depression (NIH Publication No. 21-MH-8079). Bethesda, MD: U.S. Government Printing Office. From: <https://www.nimh.nih.gov/health/publications/depression>
- [3] Robert E. Rakei, Depression, Primary Care: Clinics in Office Practice, Volume 26, Issue 2, 1999, Pages 211-224, ISSN 0095-4543, [https://doi.org/10.1016/S0095-4543\(08\)70003-4](https://doi.org/10.1016/S0095-4543(08)70003-4).
- [4] Depresivní porucha a jak ji překonat, prof. MUDr. Praško Jan, CSc. a kol., ISBN: 978-80-7262-656-4, první vydání, © Galén, 2009

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Božena Mannová, Ph.D. kabinet výuky informatiky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Božena Mannová, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování / Prohlášení

Tímto děkuji paní Ing. Boženě Mannové, Ph.D. za její podporu a zkušenosti, které mi poskytla při vedení mé bakalářské práce. Rovněž děkuji své rodině za jejich neutuchající podporu během celého studia.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 1. 5. 2023.

.....

## Abstrakt / Abstract

Tato práce je zaměřena na vytvoření webové aplikace na pomoc lidem se zvládáním příznaků deprese. Byla provedena analýza existujících IT projektů, které se zabývají problematikou depresí. Na základě této analýzy byly navrženy základní funkcionality IT systému. Byl proveden návrh aplikace. Na základě analýzy implementačních technologií byl systém implementován a otestován. Výsledkem je funkční aplikace pomáhající uživateli zvládat deprese.

**Klíčová slova:** aplikace na zvládání deprese, deprese, kognitivní terapie, funkční požadavky aplikace, Spring Boot, React

This work is aimed at creating a web application to help people cope with the symptoms of depression. An analysis of existing IT projects dealing with the issue of depression was carried out. Based on this analysis, the basic functionalities of the IT system were designed. The design of the application was carried out. Based on the analysis of implementation technologies, the system was implemented and tested. The result is a functional application helping the user to manage depression.

**Klíčová slova:** applications for coping with depression, depression, cognitive therapy, application functional requirements, Spring Boot, React

# Obsah /

<b>1 Úvod</b>	<b>1</b>	<b>8 Rešerše technologií</b>	<b>30</b>
<b>2 Co je deprese?</b>	<b>2</b>	8.1 Backend . . . . .	30
2.1 Představení . . . . .	2	8.2 Databázový systém . . . . .	31
2.2 Popis . . . . .	2	8.3 Frontend . . . . .	32
2.3 Příznaky . . . . .	3	8.4 Výstup z rešerše technologií . . . . .	32
2.4 Léčba . . . . .	3	<b>9 Implementace</b>	<b>34</b>
<b>3 Duševní nemoci spojené s depresí</b>	<b>4</b>	9.1 Backend . . . . .	34
3.1 Dystymie . . . . .	4	9.1.1 Spring Boot . . . . .	34
3.2 Smíšená úzkostně depresivní porucha . . . . .	4	9.2 Frontend . . . . .	34
3.3 Bipolární afektivní porucha . . . . .	4	9.3 Zabezpečení . . . . .	35
3.4 Sezónní afektivní porucha . . . . .	5	9.3.1 Zabezpečení uživatelských hesel . . . . .	35
<b>4 Analýza současných IT projektů pro pomoc s depresí</b>	<b>6</b>	9.3.2 Zabezpečení serverových požadavků . . . . .	36
4.1 Zkoumané aplikace . . . . .	6	9.4 Grafické zpracování . . . . .	37
4.1.1 Nepanikař . . . . .	6	9.5 Závěr implementace . . . . .	41
4.1.2 VOS: Plán Mentálního Zdraví . . . . .	7	<b>10 Testování</b>	<b>42</b>
4.1.3 MindDoc: Your Companion . . . . .	9	10.1 Jednotkové testování . . . . .	42
4.2 Závěr analýzy . . . . .	10	10.2 Uživatelské testování . . . . .	42
<b>5 Zvolené řešení</b>	<b>11</b>	10.2.1 Výběr testerů . . . . .	42
5.1 Bezpečné a uklidňující prostředí . . . . .	11	10.2.2 Prostředí . . . . .	42
5.2 Stav deprese . . . . .	11	10.2.3 Testovací scénáře . . . . .	42
5.3 Plánovač aktivit s výsledkem . . . . .	12	10.2.4 Závěr uživatelského testování . . . . .	44
5.4 Uklidňovač myšlenek . . . . .	12	<b>11 Závěr</b>	<b>45</b>
5.5 Hlášení chyb a připomínek . . . . .	12	11.1 Budoucí vylepšení . . . . .	45
<b>6 Softwarová analýza</b>	<b>13</b>	<b>Literatura</b>	<b>46</b>
6.1 Definice rolí . . . . .	13	<b>A Zkratky</b>	<b>49</b>
6.2 Systémové požadavky . . . . .	14	A.1 Soubory, které jsou součástí práce . . . . .	49
6.2.1 Funkční požadavky . . . . .	14		
6.2.2 Kvalitativní požadavky a omezení . . . . .	16		
6.3 Případy užití . . . . .	17		
6.4 Hierarchická úkolová analýza . . . . .	22		
6.4.1 Deník myšlenek . . . . .	22		
6.4.2 Plánovač aktivit . . . . .	23		
<b>7 Návrh řešení</b>	<b>24</b>		
7.1 Doménový model . . . . .	24		
7.1.1 Entita Uživatele (User) . . . . .	24		
7.2 Vnitřní hierarchie dat . . . . .	26		
7.2.1 Třídy Helper a UserItem . . . . .	26		
7.2.2 Třída HelperItem . . . . .	26		
7.3 Prototyp aplikace . . . . .	28		

## Tabulky / Obrázky

<b>4.1</b> Srovnání vlastností zkoumaných aplikací .....	10
<b>4.1</b> Aplikace Nepanikař – Přehled a pomůcky deprese .....	6
<b>4.2</b> Aplikace Nepanikař – Úzkost a panika .....	7
<b>4.3</b> Aplikace VOS: Plán Mentálního Zdraví .....	8
<b>4.4</b> Aplikace MindDoc: Your Companion .....	9
<b>6.1</b> Katalog Aktérů .....	13
<b>6.2</b> Případy užití: Správa účtu .....	17
<b>6.3</b> Případy užití: Test deprese .....	18
<b>6.4</b> Případy užití: Plánovač aktivit .....	19
<b>6.5</b> Případy užití: Deník myšlenek .....	20
<b>6.6</b> Případy užití: Hlášení .....	21
<b>6.7</b> HTA diagram – Deník myšlenek .....	22
<b>6.8</b> HTA diagram – Plánovač aktivit .....	23
<b>7.1</b> Doménový model .....	25
<b>7.2</b> Vnitřní reprezentace tříd Helper a UserItem .....	26
<b>7.3</b> Vnitřní reprezentace třídy HelperItem .....	27
<b>7.4</b> Prototyp Plánovače aktivit .....	28
<b>7.5</b> Prototyp Deníku myšlenek .....	29
<b>9.1</b> Uživatelské rozhraní pro přihlášení .....	37
<b>9.2</b> Uživatelské rozhraní Plánovače aktivit .....	38
<b>9.3</b> Uživatelské rozhraní Deníku myšlenek .....	39
<b>9.4</b> Administrátorské rozhraní se seznamem uživatelů .....	40



# Kapitola 1

## Úvod

Téma zaměřené na pomoc zvládání symptomů deprese jsem zvolil z toho důvodu, že se jedná o duševní onemocnění, které je mezi lidmi dobře známé. Jde o nežádoucí nepříjemný stav myslí, se kterým se alespoň jednou setká mnoho z nás ať už kvůli vrozené tendenci k těmto stavům, kvůli poruše nálady, kvůli reakci na těžkou životní událost nebo bez zjevného důvodu. Zároveň především vlastní zkušeností jsem vyzoroval, že ač množství lidí trpící skutečnou lékařsky diagnostikovatelnou depresí není závratně vysoké, tak množství lidí, kteří mají alespoň některé nežádoucí příznaky přítomné u deprese, je enormní. Myslím si proto, že bychom se měli více zaměřit na pomoc lidem s těmito potížemi.

Deprese je v České republice většinou řešena s lékaři pomocí terapií a podáváním psychofarmak jako jsou antidepressiva. Po telefonní komunikaci s řadou českých psycho-terapeutů a psychiatrů jsem zjistil, že kapacita lékařů, kteří jsou hrazení pojišťovnami, je velmi vytížena. Někteří lékaři poskytující psychotherapeutické služby hlásí plnou kapacitu na měsíce dopředu a již nenabírají nové pacienty. Tyto problémy jsou nejvíce znát ve velkých městech jako je Praha nebo Brno. Stejná situace je u zařízení specializujících se na psychické zdraví. Pokud nejde o naléhavý případ, je zde dlouhá čekací lhůta trvající i měsíce. V menších městech a vesnicích zase naopak tyto specialisté a zařízení často zcela chybí.

Současný stav navíc není pouze trend poslední doby, ale trvá již dlouhé roky. Mnozí terapeuté rovněž podotýkají, že po začátku pandemie viru SARS-CoV-19 a kvůli vy-  
hrocené politické situaci a rostoucí inflaci ve světě zaznamenali ještě větší zájem o jejich odborné služby.

Když se pak člověku podaří k této péči dostat, může jít o léčbu trvající roky, neboť léčba deprese není v mnoha případech jednoduchá a člověk nemá žádnou záruku, že zvolený typ léčby bude úspěšný, protože výsledek závisí na mnoha faktorech.

# Kapitola 2

## Co je deprese?

### 2.1 Představení

Každý z nás jistě zažil smutné období ať už ze zřejmého důvodu nebo bez něj. Do určité míry je pro náš život takové období prospěšné, neboť nás to přiměje se pozastavit a zamyslet se nad tím, co je pro nás opravdu důležité a k čemu chceme směřovat. Někdy však tyto pocity přetrvávají i měsíce a ovlivňují náš každodenní život. V takovém případě nás možná napadne termín deprese. Toto slovo je dnes rozhodně více skloňované než například před sto lety. Pro některé lidi je toto označení pouze výmluva a přehánění a nerozumí tomu, proč se depresivní lidé nad touto náladou “prostě nepřenesou”. Pro jiné je to velká překážka, přes kterou je těžké se sám dostat.

Slavný anglický filozof Francis Bacon napsal: „*Člověk věří tomu, co chce považovat za pravdivé.*“[1] Mínil tím proces známý jako konfirmační zkreslení, což je fakt, že mnozí z nás věnují zvýšenou pozornost pouze věcem, které je v jejich přesvědčení utvrzují a ať už vědomě či nevědomě ignorují nebo bagatelizují opačné názory a utvrzují se tak v těch svých.

Ať je pravda jakákoli, deprese a negativní myšlenky jsou nepříjemné a lidé v nich setrvalovat nechtějí. Dovolte mi tuto část ukončit citátem dalšího známého člověka, a to norského spisovatele a malíře Jens Bjørneboe: „*Ten, kdo nezažil těžkou depresi po dlouhé období – ten zůstává dítětem.*“[2]

### 2.2 Popis

Deprese je závažná duševní porucha nálady, která se vyznačuje především dlouhým stavem smutku a znatelně sníženou radostí z aktivit, které danou osobu běžně činily šťastnou. Existují různé formy deprese, přičemž někdy je deprese brána i jako samotný příznak jiné nemoci. Délka jedné depresivní epizody je dva týdny až několik let. Průměrně však trvá kolem šesti měsíců. Epizody se mohou opakovat.[3] Depresi má v jednu chvíli přibližně 3,8% světové populace.[4] Průzkumy z roku 2020 také ukazují, že depresí za předchozí rok trpělo 17% lidí ve Spojených Státech Amerických ve věku 18 až 25 let.[5]

Oficiálním kritériem pro diagnózu deprese je zažívání příznaků deprese soustavně každý den minimálně po dobu dvou týdnů. Zároveň jedním z těchto příznaků musí být depresivní skleslá nálada, nebo ztráta zájmu či radosti z denních aktivit.[3]

Vědecké průzkumy naznačují, že 59–87% lidí, kteří se pokusili o sebevraždu, trpělo depresivní poruchou. Podle studií ukončilo svůj život 15% pacientů trpících poruchou nálady. To je 500krát více než u běžné populace.[6]

## 2.3 Příznaky

Příznaků indikujících depresi je velké množství a nemusí se u každého člověka projevat stejným způsobem. Přestože je deprese známá především jako psychický stav, může se stát, že pacient nejdříve pozoruje spíše fyzické příznaky a duševní zdraví z těchto potíží nepodezírá. Záleží potom na správné diagnostice lékařem.

Následuje výčet nejčastějších **příznaků deprese**[3]:

- přetrvávající smutek a skleslá nálada
- pocity beznaděje a tendence k pesimistickému myšlení
- přehnané pocity viny
- snížené sebevědomí
- ztráta zájmu či radosti z vlastních zájmových aktivit (anhedonie)
- snížená motivace, lhostejnost a netečnost (apatie)
- únava a vyčerpání (letargie)
- nerozhodnost, potíže se soustředit a potíže s pamětí
- potíže usnout (insomnie) nebo naopak přílišné spaní (hypersomnie)
- přejídání či naopak ztráta chuti k jídlu a s ní spojené změny tělesné váhy
- somatické (tělesné) potíže jako bolesti částí těla, křeče a zažívací problémy bez zjevné fyzické příčiny
- myšlenky na smrt a sebevražedné pokusy
- součástí těžkých forem deprese mohou být i vizuální a sluchové halucinace

## 2.4 Léčba

U **lehčích forem** deprese je často doporučována změna životosprávy a návštěva psychoterapie, na které se pacient svěří se svými problémy a učí se s nimi pracovat a rozvíjet své myšlení a osobnost. Existují různé druhy psychoterapií. Dělí se většinou na individuální (které se odehrávají pouze mezi pacientem a terapeutem), rodinné a skupinové terapie. Mezi známější typy specializovaných psychoterapií patří například kognitivně behaviorální terapie (KBT) nebo Gestalt terapie.

Pro **závažnější formy** potíží související s depresí je vedle změny životosprávy a návštěvování psychoterapeuta ještě doporučována psychiatrická péče, při které může lékař rozhodnout o léčbě psychofarmaky, tedy léky ovlivňujících mozkovou činnost. K léčbě depresivní nálady se nejčastěji předepisují různé druhy antidepresiv. Známým zástupcem antidepresiv jsou takzvané SSRI, tedy antidepresiva se selektivní inhibicí zpětného vychytávání serotoninu (selective serotonin reuptake inhibitor). Plný nástup účinků antidepresiv je u většiny z nich očekáván v rozmezí 4 až 6 týdnů. K léčbě psychofarmaky se přistupuje pouze v těžších případech, neboť mohou způsobovat nežádoucí účinky a být tak další zátěží na zdraví pacienta.[3]

Užitečnou pomůckou ke správné léčbě je **psychodiagnostika**, jejíž funkcí je pomocí současných poznatků odborníků provést analýzu duševního stavu pacienta a pokusit se určit správnou diagnózu. Pacient se tak dozví více o svém duševním zdraví a jeho lékaři mohou efektivněji směřovat léčbu. Může se například dozvědět, že jeho duševní nebo fyzické potíže spočívají v odlišném problému, než se iniciálně domníval.

## Kapitola 3

### Duševní nemoci spojené s depresí

K tomu, abychom pochopili, jak můžeme lidem s depresivními příznaky co nejlépe pomoci, nám pomůže seznámit se s typy poruch souvisejícími s depresí a s nimi spojeného prožívání lidí, kteří jimi trpí.

#### 3.1 Dystymie

Dlouhotrvající typ poruchy nálady s méně závažnými příznaky deprese se nazývá dystymie. Lidé trpící touto poruchou zažívají depresi obvykle několik let. Tento stav může být buď souvislý, nebo nálada takového člověka může kolísat mezi mírnou depresí a běžným stavem. Člověk je často dlouhodobě unavený, má potíže se spánkem, každodenní činnosti jsou pro něj obtížné a žádná aktivita (ani volnočasová) mu nečiní radost. Obvykle však projevy dystymie nejsou natolik omezující, aby člověk nezvládal běžný život jako například lidé v těžké depresivní epizodě. Neznamena to ale, že tato porucha negativně neovlivňuje jeho život.

Případy dystymie mají ve většině případů začátek v mládí. Mohou být také následkem depresivní epizody.

Dystymie je někdy lidmi označována jako *chronická deprese*. Lidé s touto poruchou nálady se svému okolí mohou jevit pouze jako pesimisté. Pesimismus je však osobnostní rys.[7][8]

#### 3.2 Smíšená úzkostně depresivní porucha

Tato porucha opět bývá mylně považována za povahový rys. Jedná se totiž o poruchu, při níž člověk vykazuje jak příznaky deprese, tak příznaky úzkosti, avšak tyto příznaky nejsou natolik závažné, aby splňovaly diagnózu pro úzkostnou poruchu nebo deprese. Mezi příznaky patří například přetrvávající smutek, snížené sebevědomí a katastrofické uvažování o budoucnosti.[7]

Zajímavým poznatkem je, že mezi pacienty s touto poruchou jsou často rozšířeny poruchy spánku (potíže usnout, časté probouzení), psychosomatické potíže (bolesti žaludku, hlavy) a poruchy imunitního systému.

Lidé s touto poruchou jsou většinou schopni hodnotně fungovat v běžném životě, avšak v důsledku této poruchy se často stahují do sebe a snaží se splnit pouze to nejnnutnější.[9]

#### 3.3 Bipolární afektivní porucha

Dalším příbuzným duševním onemocněním je bipolární afektivní porucha, v minulosti označována jako maniidepresivní psychóza. Toto onemocnění se vyznačuje střídáním období deprese a nadměrně radostné nálady – mánie. Mezi těmito fázemi může a nemusí být delší nebo kratší bezpříznakové období. Známé je i ultra-rychlé cyklování, kdy se

fáze mění i dvakrát denně. Bipolární afektivní porucha se v psychiatrii dělí na dva hlavní typy. Bipolární porucha I. typu, při které se deprese střídá s plně rozvinutou mánií a bipolární porucha II. typu, kdy se střídá deprese s hypománií. Hypománie je slabší formou mánie. U druhého typu bipolární afektivní poruchy se depresivní stavy vyskytují častěji.

Pacienti ve většině případů vyhledávají pomoc právě při depresivních stavech, neboť v manických a hypomanických stavech mají kromě dobré nálady i více energie, zvýšenou chuť k jídlu, k odpočinku potřebují méně spánku a necítí se nemocní. Tyto stavy však mohou být stejně nebezpečné, neboť u těžších forem může dojít k bludům a halucinacím, nebo člověk může ztratit sociální zábrany.[10]

Lidé trpící bipolární afektivní poruchou mají dvojnásobné riziko úmrtí v důsledku sebevražedného pokusu ve srovnání s jinými typy depresivní poruchy. Podle odhadů se pokusí o sebevraždu alespoň jednou během svého života 25–50% bipolárních pacientů a 8–19% ji úspěšně dokoná.[6]

### 3.4 Sezónní afektivní porucha

Pokud člověk pocítuje příznaky deprese každoročně s nástupem podzimu či zimy a vymizí u něj pravidelně až na jaře, je vysoce pravděpodobné, že trpí sezónní afektivní poruchou také označovanou jako *zimní deprese*. Tato porucha souvisí se změnou počasí v zimních měsících.

Mezi nejčastější příznaky patří zvýšená spavost, málo energie, zvýšená chuť k jídlu a s ní spojené přibírání na váze.

Dobrou zprávou je, že tento typ deprese je velmi efektivně léčen fototerapií. Fototerapie je metoda vystavení se zdroji záření potřebné vlnové délky v bezpečném prostředí. Tato procedura je nejúčinnější při provádění v ranních hodinách.[7]

# Kapitola 4

## Analýza současných IT projektů pro pomoc s depresí

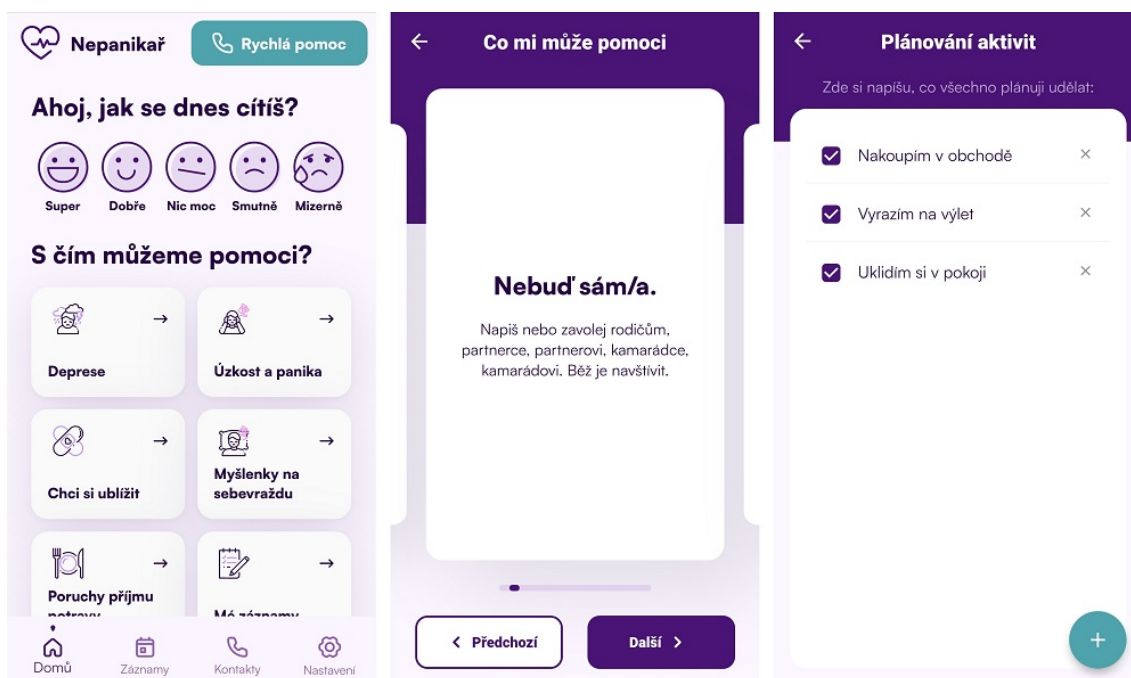
Na následujících stranách zdokumentuji, jaké IT projekty, zabývající se problematikou deprese a psychické nepohody, v současné době již existují a jaké jsou jejich přednosti nebo naopak slabiny.

### 4.1 Zkoumané aplikace

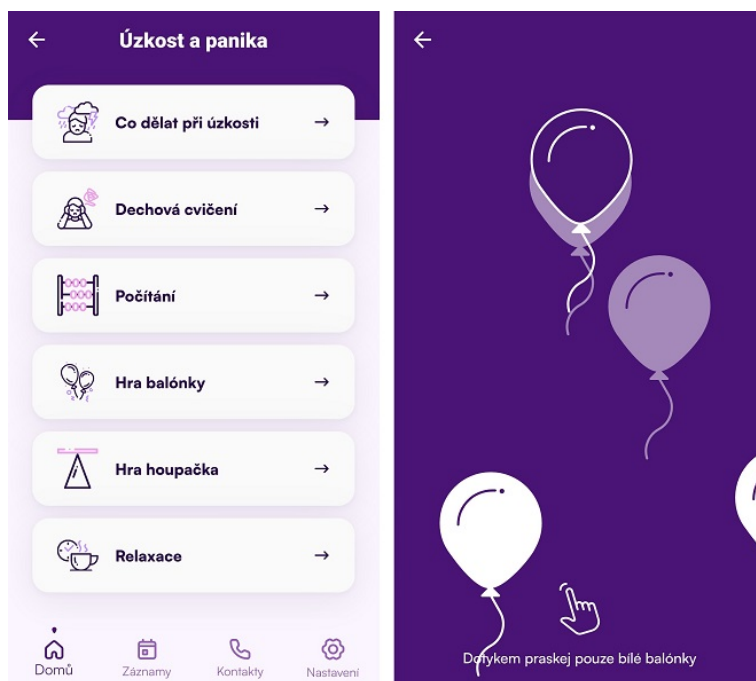
#### 4.1.1 Nepanikař

Pravděpodobně neznámější českou terapeutickou aplikací je aplikace Nepanikař[11], která se do povědomí občanů dostala především díky velké publicitě českých médií. Aplikace je rozhodně krok správným směrem, neboť psychoterapie na pojišťovnu jsou dnes v Česku často přeplněné[12] a člověk potřebuje pomoc co nejdříve.

Aplikace je rychlou pomocí pro lidi s depresemi, úzkostmi, sebepoškozováním, myšlenkami na sebevraždu a pro lidi s poruchami příjmu potravy, tedy těmi nejčastějšími psychickými potížemi. Možností jsou zde dechová cvičení pro uklidnění, jednoduché hry pro odreagování mysli a plánování či zaznamenávání aktivit pro zlepšení nálady. Aplikace je dostupná v deseti jazycích a je možno měnit barevný vzhled.



Obrázek 4.1. Aplikace Nepanikař – Přehled a pomůcky deprese



**Obrázek 4.2.** Aplikace Nepanikař – Úzkost a panika

Web organizace Nepanikař<sup>1</sup> propojuje širokou veřejnost se službami psychologů a psychiatrů po celém Česku. Na samotném webu je také možnost zaplatit si on-line terapii.

#### Výhody:

- Přehlednost
- Pomoc pro různé psychické potíže
- Originální nápady pomoci uživatelům s úzkostí
- Možnost přizpůsobení vzhledu
- Zdarma
- Odkaz na on-line chatovou poradnu
- Mnoho kontaktů na odbornou pomoc

#### Nevýhody:

- Je zde sice přítomen deník a průběžné sledování nálady, ale nenarazil jsem na průběžnou komplexní práci za účelem zbavení se zmíněných potíží
- Prostředků pro pomoc s depresí je poměrně málo

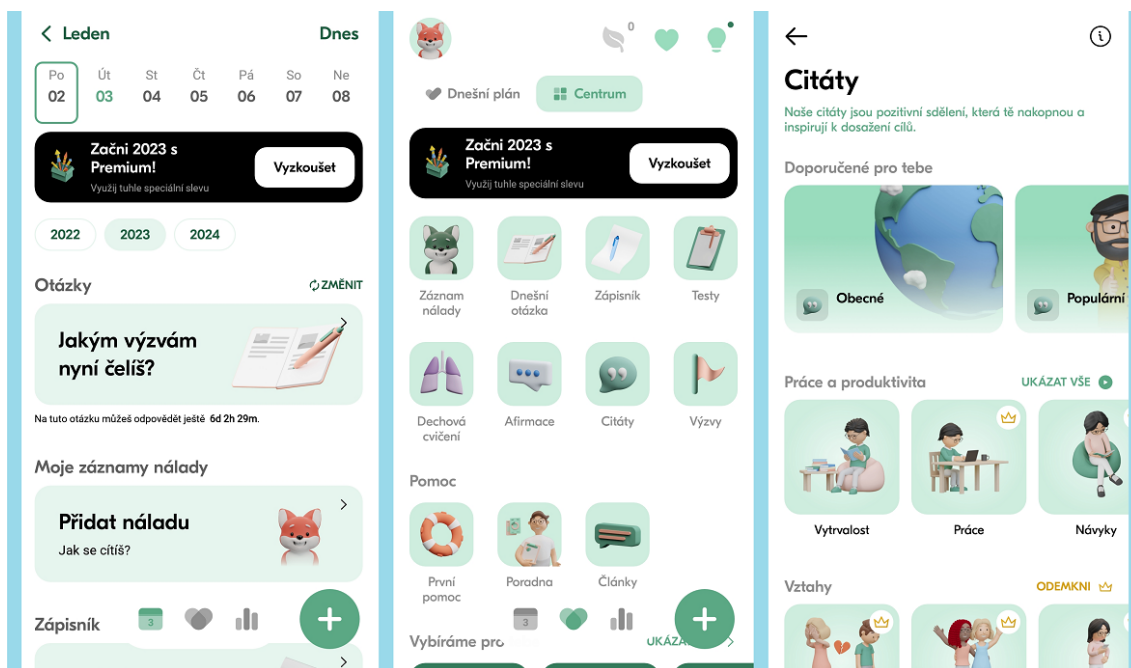
### 4.1.2 VOS: Plán Mentálního Zdraví

VOS[13] je poměrně nová aplikace, kterou vytvořilo české vývojářské studio Qusion. Jejím cílem je pomoci vám dosáhnout duševní pohody. Po spuštění aplikace dostanete k vyplnění dotazník, díky kterému aplikace zjistí, na jakých oblastech vašeho života potřebujete zapracovat. Mezi tyto oblasti patří spánek, vztahy, produktivita, stravování, cvičení a zvládání stresu. Velké množství funkcí v této aplikaci je placených, avšak jak zmiňují – roční předplatné stojí méně než návštěva u soukromého psychoterapeuta.

Aplikace umožňuje uživatelům sledovat jejich dlouhodobé změny zaznamenáváním si aktuální nálady, které si mohou zpětně zobrazit. V přehledu je vidět průměrná nálada,

<sup>1</sup> <https://nepanikar.eu/>

souvislost množství vašeho spánku a nálady, souvislost fyzických aktivit a nálady, stabilita nálady a průměrný počet ušlých kroků. Uživatel si v aplikaci také může vyplnit psychologické testy, které mají za úkol zjistit, zda má člověk duševní potíže jako depresi nebo generalizovanou úzkostnou poruchu.



**Obrázek 4.3.** Aplikace VOS: Plán Mentálního Zdraví

Jedna z možností, jak odreagovat své myšlenky, je funkce s názvem Chytrý deník. Zde je uživateli zobrazena určitá otázka k zamyšlení, na kterou může libovolně odpovědět. Jedná se však o jedinou otázku denně, a proto ji mezi celkové hodnocení jako aktivity pro odreagování neřadím. Další funkce, kterou aplikace disponuje jsou dechová cvičení. Tato sekce je v aplikaci opravdu velmi bohatá, protože nabízí 48 různých druhů dechových cvičení, které se liší způsobem provádění a účelem. Dechová cvičení jsou zde rozdělena do sekcí pro zklidnění, dobrý spánek, nabrání energie, zlepšení výkonu a pro lepší zdraví.

Oblíbenou částí aplikace některých uživatelů jsou afirmace a motivační citáty. Afirmace jsou pozitivní fráze, jejichž účelem je lidem pomoci při překonávání životních výzev a vypořádat se s negativními myšlenkami. Motivační citáty lidem dávají jiný úhel pohledu na důležité věci v životě a motivují je k dosažení jejich cílů. Součástí aplikace je také on-line poradna, ve které mají uživatelé možnost konzultovat své potíže s reálným psychologickým poradcem.

#### Výhody:

- Dlouhodobé statistiky souvislosti spánku a aktivity s náladou
- Mnoho dechových cvičení
- Afirmace a motivační citáty
- On-line poradna

#### Nevýhody:

- Placená verze může uživatele odradit
- Rozumím, že hlavní placenou částí aplikace je on-line poradna a že se tvůrci snažili, aby aplikace nebyla příliš velká, ale očekával bych od aplikace více funkcí – například

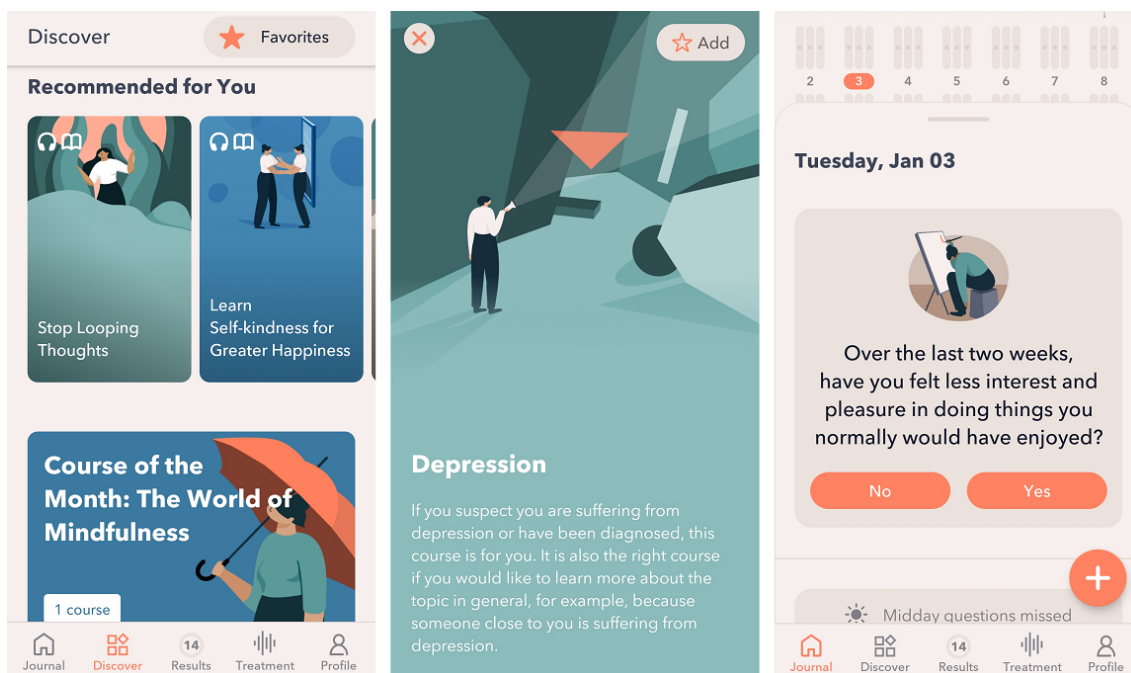


jednoduché hry na odreagování, jiné relaxační techniky než dechová cvičení nebo relaxační hudbu

### 4.1.3 MindDoc: Your Companion

Aplikací, která volí mírně odlišný přístup k pomoci s duševním zdravím, je MindDoc[14]. Na začátku vám aplikace položí otázky týkající se vašeho života a na jejich základě vám doporučí možnosti pomoci.

Pomoc v aplikaci se skládá z různých kurzů, které si uživatel pročítá, nebo je poslouchá. Tyto kurzy se týkají například klidu mysli, relaxace, posílení sebevědomí, způsobu přemýšlení, zvládnání emocí, spánku, cvičení, vztahů a porozumění duševním nemocem. Jednotlivé kurzy jsou šikovně navrženy, esteticky přívětivé a čtivé.



**Obrázek 4.4.** Aplikace MindDoc: Your Companion

Většina těchto kurzů je dostupná pouze v prémiové verzi, která stojí osmnáct set korun českých. Aplikace je navíc pouze v angličtině.

#### Výhody:

- Analýza statistik
- Originální způsob pomoci
- Velké množství obsahu

#### Nevýhody:

- Většina kurzů je placená
- Ke kvalitní analýze statistik je třeba vyplňovat dotazník o náladě třikrát denně. To může některé uživatele mírně obtěžovat.
- Dostupná pouze v angličtině

## 4.2 Závěr analýzy

<i>vlastnost</i>	Nepanikař	VOS	MindDoc
dechová cvičení	+	+	+
aktivity pro odreagování	+	–	+
motivační texty	–	+	+
statistiky	+	prémium	+
kontakt na specialisty	+	+	+
plně bezplatná	+	–	–
v českém jazyce	+	+	–

**Tabulka 4.1.** Srovnání vlastností zkoumaných aplikací

Z uvedených aplikací je vidět, že existuje mnoho způsobů, které jsou určeny ke zlepšení duševního zdraví a tím i příznaků deprese.

Jednou z častých funkcí těchto aplikací je sledování nálady, díky kterému je možné rozpoznat například nastupující depresivní epizodu, nebo jak uživatelova nálada přímo souvisí s množstvím a pravidelností spánku či se zdravým pohybem. V této funkci exceluje například VOS nebo Daylio. (Aplikace Daylio nebyla zmíněna v analýze, neboť se přímo nejedná o terapeutickou aplikaci) Neshledávám tedy za nutné, aby má aplikace touto funkcí disponovala, neboť je v existujících aplikacích dovedena téměř k dokonalosti.

Další oblíbenou funkcí je instruované dechové cvičení. Určité druhy dechového cvičení mají prokazatelný účinek na snížení napětí při stresu. Aplikací pro dechová cvičení existuje mnoho. Zároveň jsou namířena spíše na zvládnutí právě uvedeného stresu a úzkostí, ne deprese.

Zkoumané aplikace obsahují kontakty na rychlou lékařskou pomoc. Tato funkce je zde velmi důležitá, neboť nevíme, zda uživatel, který aplikaci zrovna používá, nemá suicidalní myšlenky, nebo nezažívá těžké psychické potíže. Bude tedy správné zakomponovat tuto funkci i do mé aplikace.

# Kapitola 5

## Zvolené řešení

Příčiny symptomů deprese mají z velké části počátek v mozku. Tedy, i když by se změnila nebo udála určitá externí akce, která by nám od deprese dlouhodobě pomohla, musí zákonitě mezi těmito dvěma událostmi dojít ke vlivu na mozek. (Už jen z toho důvodu, že odtud depresi pociťujeme.) Jedním ze způsobů, kterým je možné zbavit se pocitů deprese nebo je alespoň zmírnit, je změnit způsob uvažování a tím změnit myšlenky, kterými depresi prohlubujeme, nebo udržujeme v chodu.

Lidský mozek funguje na základě asociací – například pokud slyšíme konverzaci na nějaké téma, vybaví se nám často informace, kterou máme s daným tématem spojené. V tomto případě můžeme přirovnat funkci mozku k umělým neuronovým sítím, které se učí získáním vstupů a následnou úpravou vnitřní struktury na základě porovnání svého konečného výstupu a očekávaného výstupu. Podobným způsobem si však může lidský mozek vytvořit negativní vzorce chování například tak, že se učí svým předpokládaným výsledkem a ne tím reálným.

Aplikace bude tedy využívat principů kognitivní terapie[15][16], která se právě na myšlenkové pochody zaměřuje. Kognitivní terapie je předchůdcem terapie kognitivně behaviorální, což je psychologická léčba, která je mezi odborníky poměrně známá. Některé základní myšlenky kognitivní terapie jsou v ní však často opomíjeny.

Následující funkce a vlastnosti výsledného řešení byly zvoleny na základě analýzy existujících softwarových řešení, připomínek jejich uživatelů, konzultací specialistů a studia dané oblasti.

### 5.1 Bezpečné a uklidňující prostředí

Aplikaci je nutné vytvořit vzhledem k její terapeutické povaze. Přispívající k její hlavní funkci, a to k psychickému uvolnění. Aplikace bude mít uklidňující vzhled a bude příjemná k používání.

- Některé barvy a fonty jsou u lidí oblíbenější než jiné a mají na lidi různý efekt. Je potřeba prozkoumat poznatky v této oblasti a použít uklidňující barvy a styly písma.
- Aplikace nesmí být frustrující k používání – musí být přehledná a rychle reagovat.
- Jelikož se budou do databáze ukládat citlivá data uživatelů, budou se ukládat v zašifrované podobě.

### 5.2 Stav deprese

Někteří lidé se necítí dobře, ale považují to za běžný stav. Aby uživatelé zjistili, zda tomu tak opravdu je, budou mít v aplikaci možnost vyplnit automatický dotazník na stav deprese.

- K rozpoznání depresivní nálady je dobrým ukazatelem mezi odborníky známá Beckova stupnice, proto bude použita v aplikaci.

### 5.3 Plánovač aktivit s výsledkem

Způsoby pro plánování aktivit obsahovaly v určité formě i některé ze zkoumaných aplikací, avšak pouze jako seznam poznámek. Má aplikace bude využívat zmíněné myšlenky kognitivní terapie. Konkrétně pracuje s předpokladem, že budoucí očekávaná činnost je méně příjemná, než jaká je doopravdy.

Nejdříve si člověk zapíše aktivity, které má v plánu daný den vykonat. Dále v procentech uvede hodnotu vyjadřující očekávanou radost z konání těchto činností a následně tyto aktivity provede. Kdykoli se pak může ke svému plánovači vrátit a zapsat si reálnou radost z každé aktivity a vidět, zda ji při hodnocení předem podcenil.

Myšlenkou je, že uživatel, který trpí depresí, uvidí, že jeho předchozí hodnocení je často zkreslené a nemusí se činností vyhýbat, čímž by se mu měla časem zlepšit nálada, neboť nemusí trávit čas tak nepříjemnými činnostmi, za jaké je považoval.[15]

### 5.4 Uklidňovač myšlenek

V tíživých psychických situacích, do kterých se lidé s depresí dostávají, si budou moci uživatelé zaznamenat, co se stalo, své současné myšlenky a pocity a následně se zamyslet, zda jsou tyto myšlenky založené na reálných faktech a případně si zapsat racionální myšlenky.

Cílem je uvědomit si, že člověk má v depresi tendenci podceňovat situace a mysl ho nabádá přemýšlet o těch nejhorších možných scénářích. Je potřeba, aby si člověk tyto myšlenkové pochody v mysli neupevňoval, neboť tím se pouze zvýší jejich frekvence.[16] Dále tuto funkci nazývám jako deník myšlenek.

### 5.5 Hlášení chyb a připomínek

Uživatel bude mít možnost nahlásit případné chyby, se kterými se v aplikaci setká, a informovat administrátory o svých názorech a připomínkách k jejímu používání.

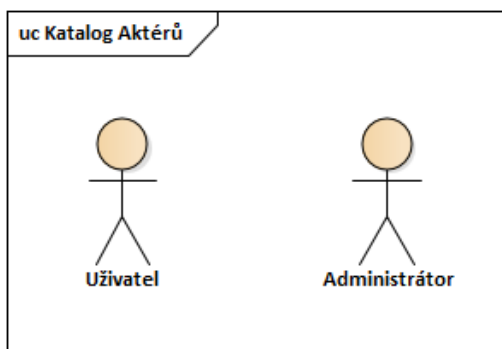
# Kapitola 6

## Softwarová analýza

Tato kapitola definuje role uživatelů systému, systémové požadavky aplikace, případy užití aplikace a hierarchickou analýzu úkolů.

### 6.1 Definice rolí

Aplikace se sestává ze dvou různých skupin uživatelů. Vytvoří se tedy dvě různá rozhraní s rozdílnými privilegii:



Obrázek 6.1. Katalog Aktérů

#### Běžný uživatel

- Běžný uživatel (dále označován pouze jako uživatel, pokud nebude specifikováno jinak), je role, pro kterou bude aplikace cíleně vyvíjena. Jde o uživatele, od kterého se neočekávají předchozí odborné znalosti v oblasti informatiky nebo psychologie.
- Uživatel bude mít možnost používat terapeutické nástroje aplikace a zobrazit si pouze data spojená s jeho vlastním účtem včetně zaslaných hlášení. Nebude mít možnost nijak ovlivňovat účty ostatních uživatelů.

#### Administrátor

- Administrátoři jsou správci aplikace. Opět od nich nebudou očekávány odborné znalosti z oblasti psychologie, ale očekává se od nich alespoň pokročilá znalost softwarových aplikací, aby mohli vývojářům komunikovat problémy a připomínky z hlášení uživatelů. Navíc budou muset být předem seznámeni s obsluhou aplikace.
- Administrátoři si mohou zobrazovat seznam ostatních uživatelů a informace o jejich účtech, zároveň mohou spravovat tyto účty a generovat aktivační kódy pro nové uživatele. Kromě toho si mohou zobrazit zasláná hlášení chyb a návrhů cizích uživatelů na zlepšení aplikace.

## 6.2 Systémové požadavky

Funkční požadavky (FR) popisují funkce, které tato aplikace bude svým uživatelům umožňovat. Kvalitativní požadavky (NFR – z anglického termínu “non-functional requirements”) pak popisují nároky na výkon, bezpečnost, škálovatelnost apod.

### 6.2.1 Funkční požadavky

#### FR 1 – Registrace nových uživatelů

- Systém umožní novým uživatelům registrovat se použitím aktivačního kódu, který získají od některého ze správců.

#### FR 2 – Vyplnění údajů při registraci

- Systém bude pro registraci uživatele vyžadovat následující údaje:
  - unikátní uživatelské jméno,
  - heslo a
  - rok narození. (S tím, že rok narození nebude povinný údaj.)

#### FR 3 – Přihlášení uživatele

- Systém umožní uživatelům přihlásit se zadáním správné kombinace uživatelského jména a hesla.

#### FR 4 – Změna osobních a přihlašovacích údajů

- Systém umožní přihlášeným uživatelům změnu:
  - uživatelského jména,
  - hesla a
  - roku narození.

#### FR 5 – Suspendování účtu

- Systém umožní uživatelům suspendovat vlastní účet. Takový účet nebude ze systému odstraněn, ale nebude se k němu moci nikdo přihlásit.

#### FR 6 – Suspendování uživatelských účtů administrátorem

- Systém umožní administrátorům suspendovat cizí účty a zapsat důvod tohoto kroku.

#### FR 7 – Obnovení uživatelských účtů administrátorem

- Systém umožní administrátorům obnovit suspendované účty s platným odůvodněním jeho uživatele.

#### FR 8 – Vyplnění testu deprese

- Systém umožní uživatelům vyplnit si test deprese a zvolit, zda chce odpovědi uchovat. Pokud byl předchozí test vyplněn před méně než 24 hodinami, upozorní uživatele, že by měl kvůli efektivitě s dalším testem počkat, ale dovolí mu si další test vyplnit.

#### FR 9 – Zobrazení seznamu předchozích testů deprese

- Systém umožní uživatelům zobrazit si seznam předchozích vyplněných testů deprese.

#### FR 10 – Zobrazení detailu předchozího testu deprese

- Systém umožní uživatelům zobrazit si detail předchozího vyplněného testu deprese.

**FR 11 – Smazání odpovědí testu deprese**

- Systém umožní uživatelům smazat uchované odpovědi předchozího vyplněného testu deprese. (Bodový výsledek uchován zůstane.)

**FR 12 – Bodový vývoj testů deprese**

- Systém umožní uživatelům zobrazit si bodový vývoj testů deprese za období posledního:
  - týdne,
  - měsíce,
  - roku.

**FR 13 – Podrobné instrukce plánovače aktivit**

- Systém poskytne uživatelům instrukce pro používání plánovače aktivit, aby uživatel pochopil jeho princip.

**FR 14 – Vytvořit plánovač aktivit**

- Systém umožní uživatelům vytvořit nový plánovač aktivit.

**FR 15 – Zobrazit seznam předchozích plánovačů aktivit**

- Systém umožní uživatelům zobrazit si seznam předchozích vytvořených plánovačů aktivit.

**FR 16 – Vytvořit aktivitu v plánovači aktivit**

- Systém umožní uživatelům přidat novou aktivitu v plánovači aktivit.

**FR 17 – Upravit aktivitu v plánovači aktivit**

- Systém umožní uživatelům upravit data aktivity v plánovači aktivit i po opětovném otevření plánovače.

**FR 18 – Odstranit aktivitu v plánovači aktivit**

- Systém umožní uživatelům odstranit existující aktivitu v plánovači aktivit.

**FR 19 – Vytvořit nový deník myšlenek**

- Systém umožní uživatelům vytvořit nový deník myšlenek.

**FR 20 – Zobrazit seznam předchozích deníků myšlenek**

- Systém umožní uživatelům zobrazit si seznam předchozích deníků myšlenek.

**FR 21 – Přidat novou položku v deníku myšlenek**

- Systém umožní uživatelům přidat novou položku v deníku myšlenek.

**FR 22 – Upravit položku v deníku myšlenek**

- Systém umožní uživatelům upravit data položky v deníku myšlenek i po jeho opětovném otevření.

**FR 23 – Odstranit položku v deníku myšlenek**

- Systém umožní uživatelům odstranit existující položku v deníku myšlenek.

**FR 24 – Nahlásit chybu nebo připomínku**

- Systém umožní uživatelům nahlásit administrátorům chybu, či připomínku k aplikaci.

#### **FR 25 – Zobrazit si historii hlášení**

- Systém umožní uživatelům zobrazit si historii svých nahlášených připomínek.

#### **FR 26 – Zobrazit si všechna hlášení**

- Systém umožní administrátorům zobrazit si všechny nahlášené připomínky s tím, že odděleně zobrazí:
  - nahlášené chyby v aplikaci a
  - připomínky.

#### **FR 27 – Zobrazit si seznam ostatních účtů uživatelů**

- Systém umožní administrátorům zobrazit si seznam existujících uživatelů.

#### **FR 28 – Zobrazit si detailní informace o účtech ostatních uživatelů**

- Systém umožní administrátorům zobrazit si detailní informace účtů existujících uživatelů.

#### **FR 29 – Vyhledávání uživatelských účtů**

- Systém umožní administrátorům vyhledávat účty podle jména a identifikátoru.

#### **FR 30 – Přístup ke kontaktům specialistů**

- Systém umožní uživatelům zobrazit si kontakty na subjekty specializující se na pomoc s duševním zdravím.

### **6.2.2 Kvalitativní požadavky a omezení**

#### **NFR 1 – Zákonné nakládání s osobními údaji**

- Systém bude při nakládání s údaji uživatelů následovat zákon č. 110/2019 Sb. o zpracování osobních údajů.

#### **NFR 2 – Šifrování citlivých údajů**

- Jelikož se jedná o terapeutickou aplikaci a uživatelé očekávají jistou míru diskrétnosti, bude systém uchovávat citlivá data v databázi zašifrovaná.

#### **NFR 3 – Klidné barvy uživatelského prostředí**

- Uživatelské prostředí aplikace bude obsahovat barvy, které jsou z hlediska psychologické teorie barev považovány za barvy navozující klid.

#### **NFR 4 – Tmavý motiv uživatelského prostředí**

- Uživatelské prostředí aplikace bude mít možnost přepnutí vzhledu na tmavý motiv.

#### **NFR 5 – Responzivní design**

- Design aplikace bude optimalizovaný pro mobilní zařízení a nejužívanější webové prohlížeče (Chrome a Safari).

#### **NFR 6 – Český a anglický jazyk rozhraní**

- Aplikace bude v českém a anglickém jazyce.

#### **NFR 7 – Zálohování**

- Pro případ selhání hardware, nebo vnitřní chyby, budou data každý měsíc zálohovány.

#### **NFR 8 – Rychlá odezva**

- Doba odezvy aplikace musí být maximálně dvě sekundy.

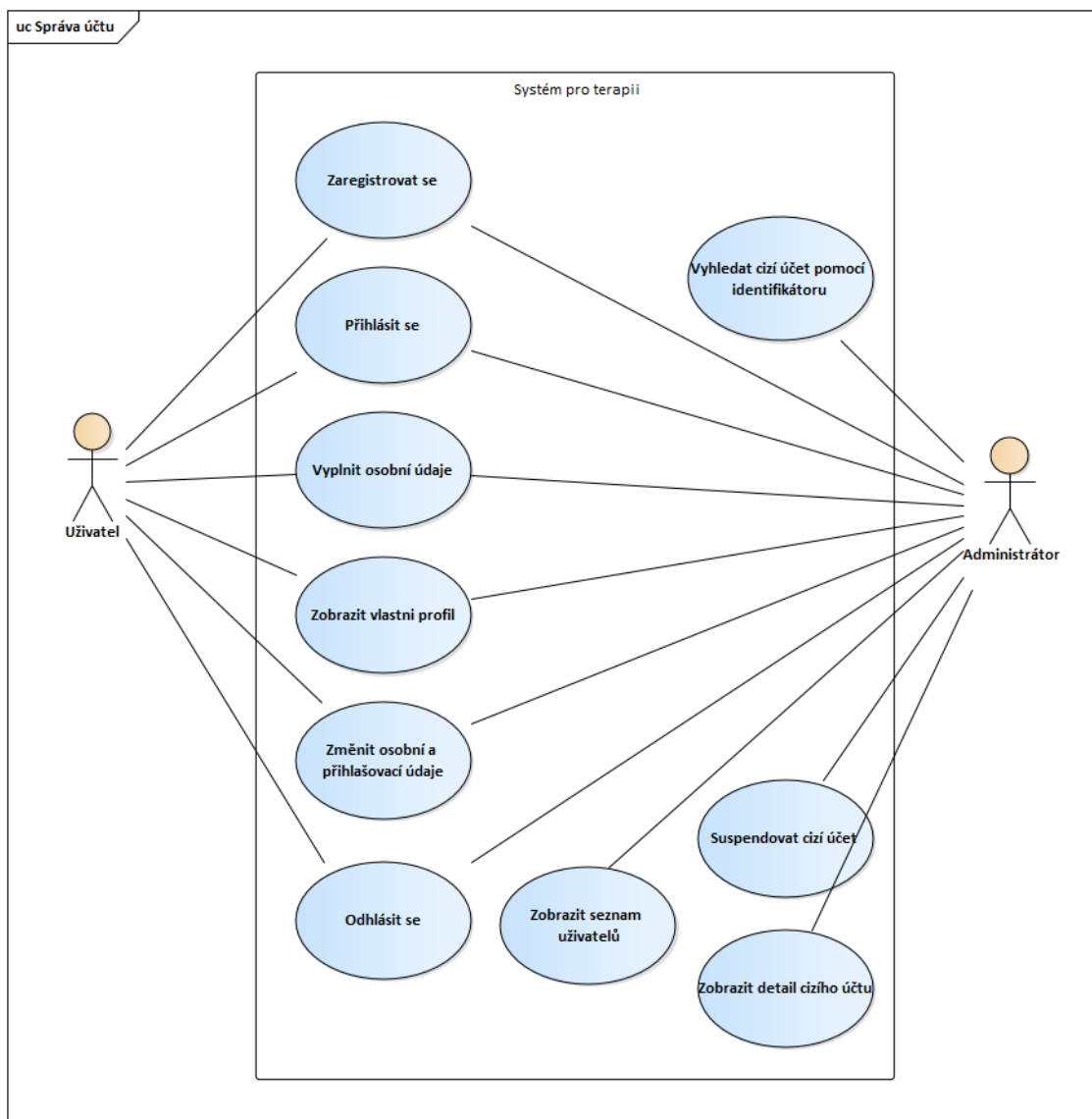


## 6.3 Případy užití

Následují případy užití (Use Cases) aplikace.

### Správa účtu:

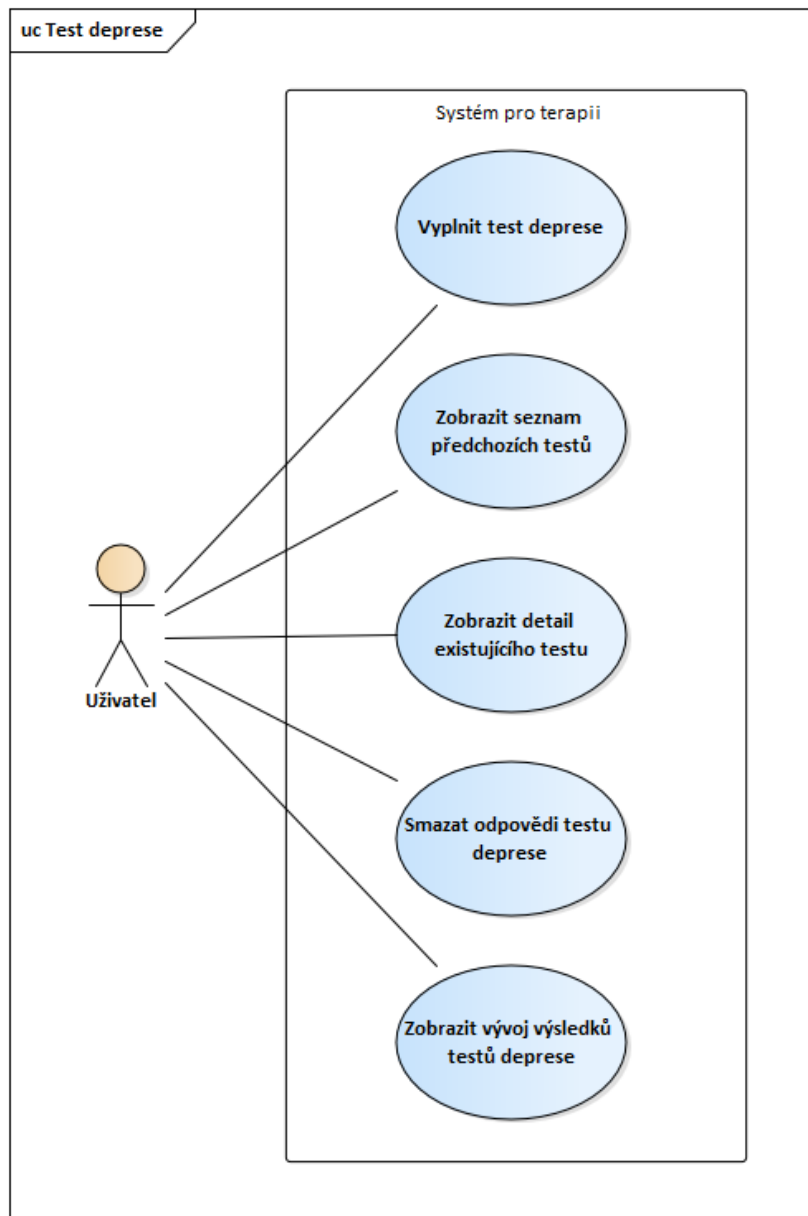
- UC 1 – Zaregistrovat se
- UC 2 – Přihlásit se
- UC 3 – Vyplnit osobní údaje
- UC 4 – Zobrazit vlastní profil
- UC 5 – Změnit osobní a přihlašovací údaje
- UC 6 – Odhlásit se
- UC 7 – Zobrazit seznam uživatelů
- UC 8 – Suspendovat cizí účet
- UC 9 – Vyhledat cizí účet pomocí identifikátoru
- UC 10 – Zobrazit detail cizího účtu



**Obrázek 6.2.** Případy užití: Správa účtu

**Test deprese:**

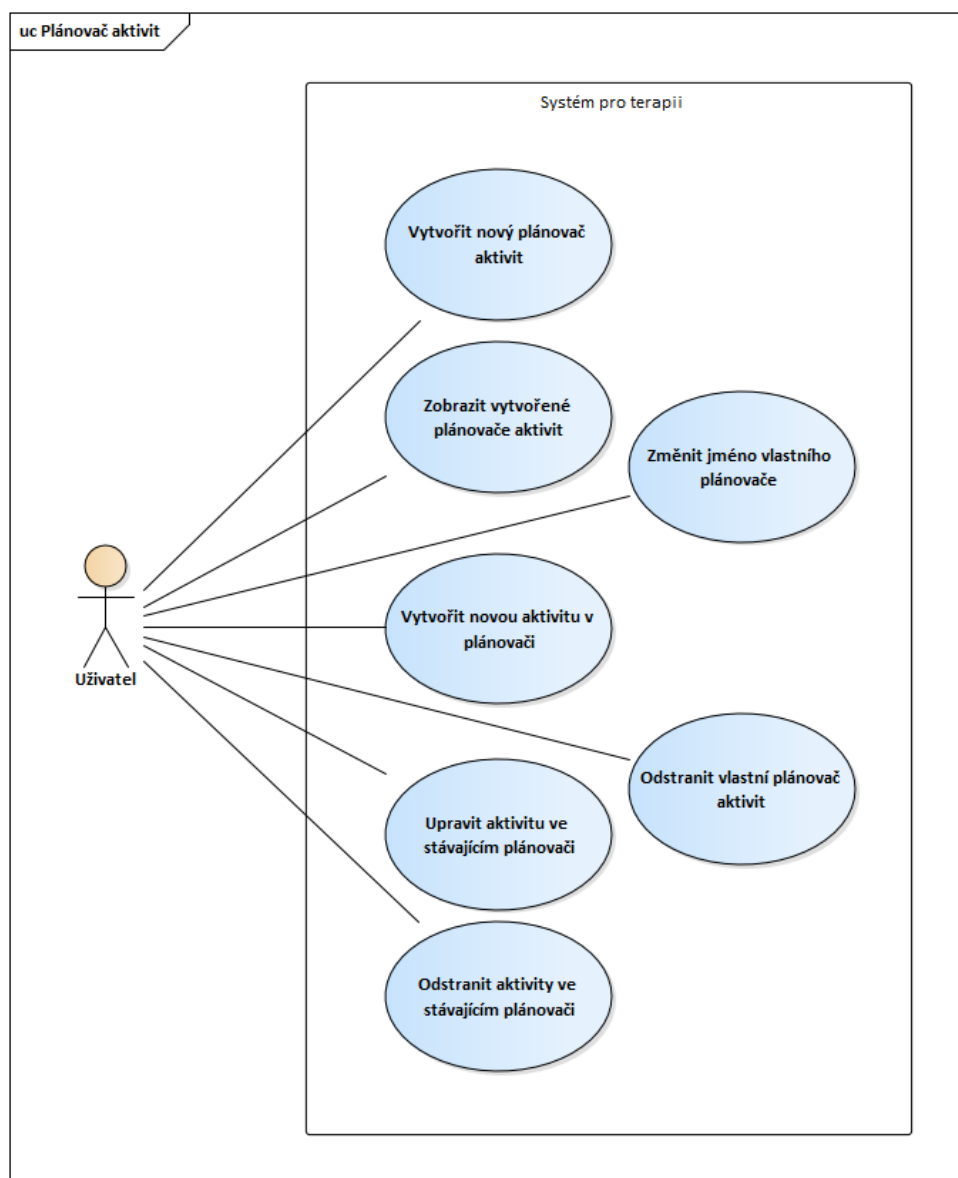
- UC 11 – Vyplnit test deprese
- UC 12 – Zobrazit seznam předchozích testů
- UC 13 – Zobrazit detail existujícího testu
- UC 14 – Smazat odpovědi testu deprese
- UC 15 – Zobrazit vývoj výsledků testů deprese



**Obrázek 6.3.** Případy užití: Test deprese

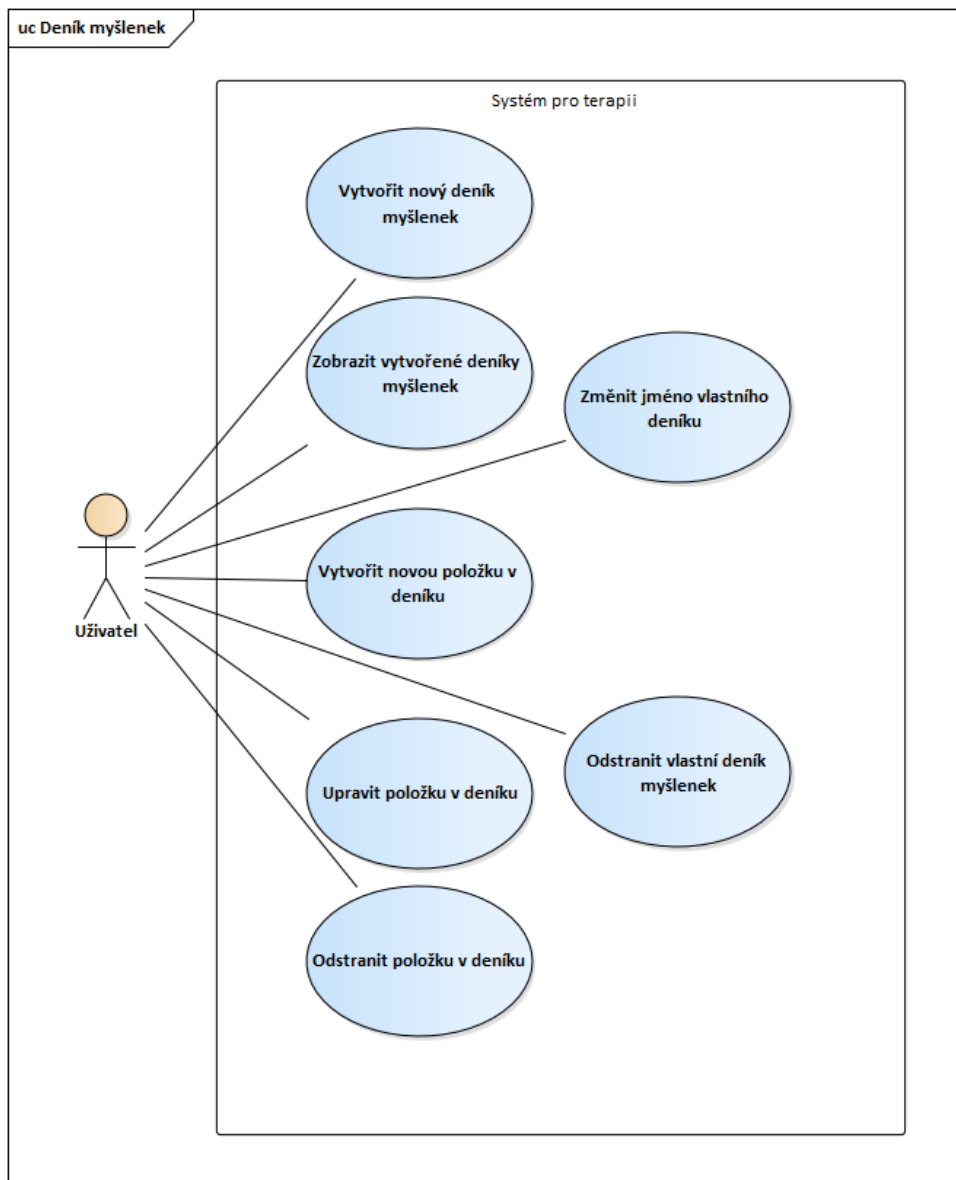
**Plánovač aktivit:**

- UC 16 – Vytvořit nový plánovač aktivit
- UC 17 – Zobrazit vytvořené plánovače aktivit
- UC 18 – Vytvořit novou aktivitu v plánovači
- UC 19 – Upravit aktivitu ve stávajícím plánovači
- UC 20 – Odstranit aktivity ve stávajícím plánovači
- UC 21 – Změnit jméno vlastního plánovače
- UC 22 – Odstranit vlastní plánovač aktivit

**Obrázek 6.4.** Případy užití: Plánovač aktivit

**Deník myšlenek:**

- UC 23 – Vytvořit nový deník myšlenek
- UC 24 – Zobrazit vytvořené deníky myšlenek
- UC 25 – Vytvořit novou položku v deníku
- UC 26 – Upravit položku v deníku
- UC 27 – Odstranit položku v deníku
- UC 28 – Změnit jméno vlastního deníku
- UC 29 – Odstranit vlastní deník myšlenek



**Obrázek 6.5.** Případy užití: Deník myšlenek

**Hlášení:**

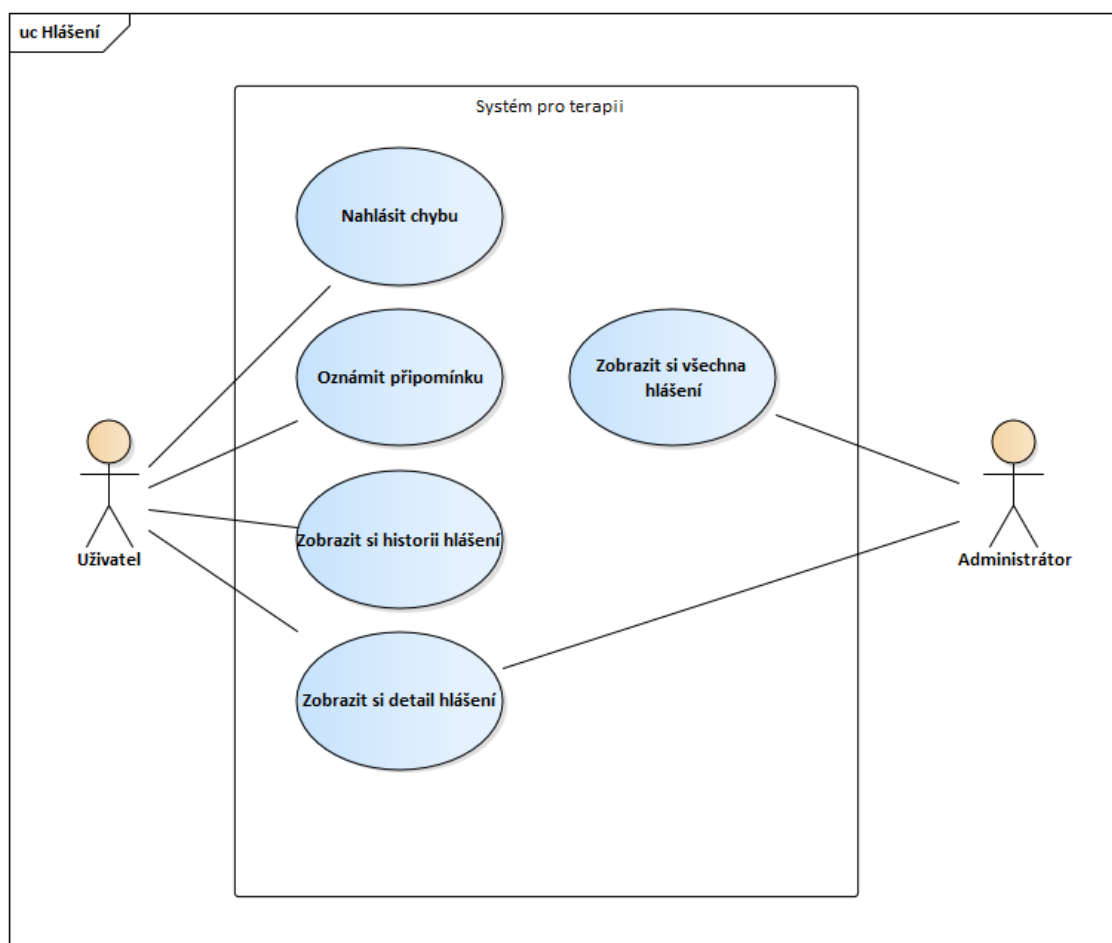
UC 30 – Nahlásit chybu

UC 31 – Oznámit připomínku

UC 32 – Zobrazit si historii hlášení

UC 33 – Zobrazit si detail hlášení

UC 34 – Zobrazit si všechna hlášení

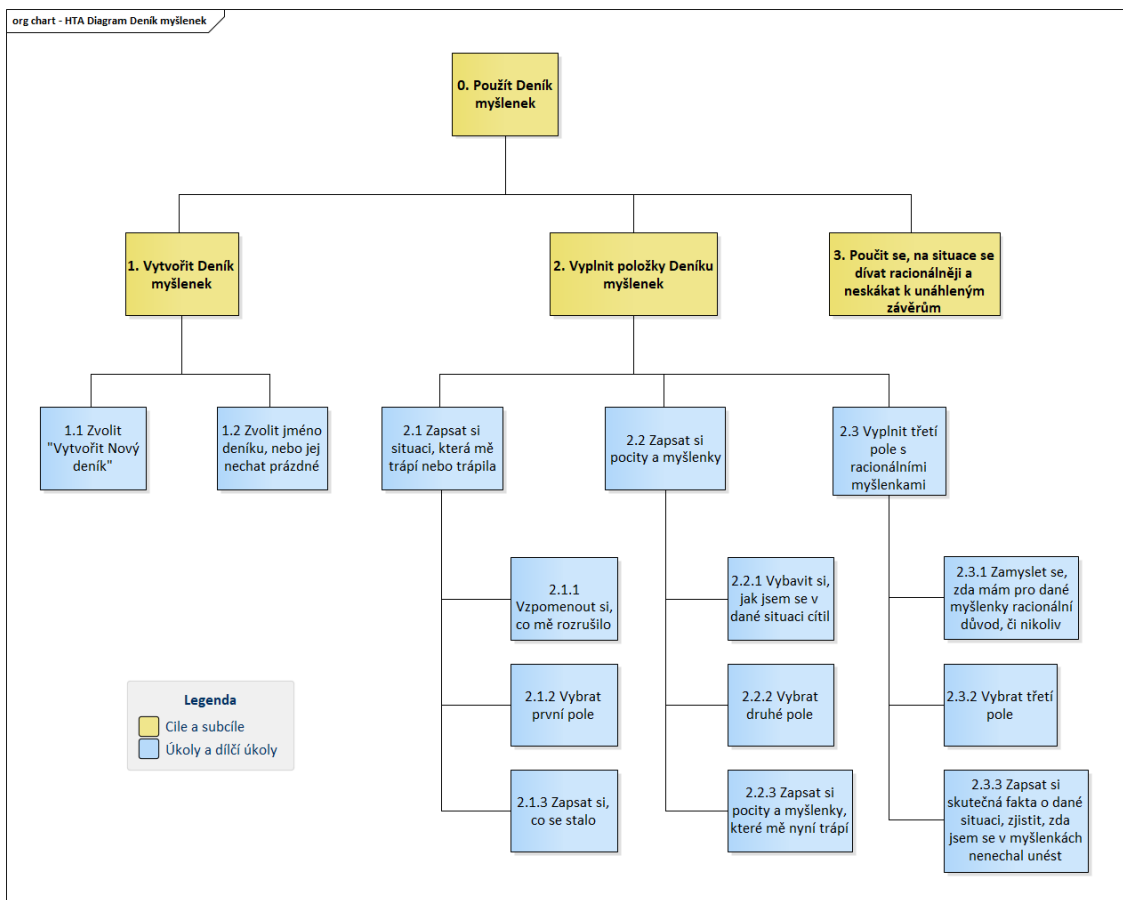
**Obrázek 6.6.** Případy užití: Hlášení

## 6.4 Hierarchická úkolová analýza

Hierarchical Task Analysis (HTA) neboli hierarchická úkolová analýza je popis úkolů k dosažení určitých cílů. Tato analýza nám pomůže jednoduše si vizualizovat posloupnost nutných úkolů. Zpracoval jsem takto dva dva komplexnější cíle v mé aplikaci.

### 6.4.1 Deník myšlenek

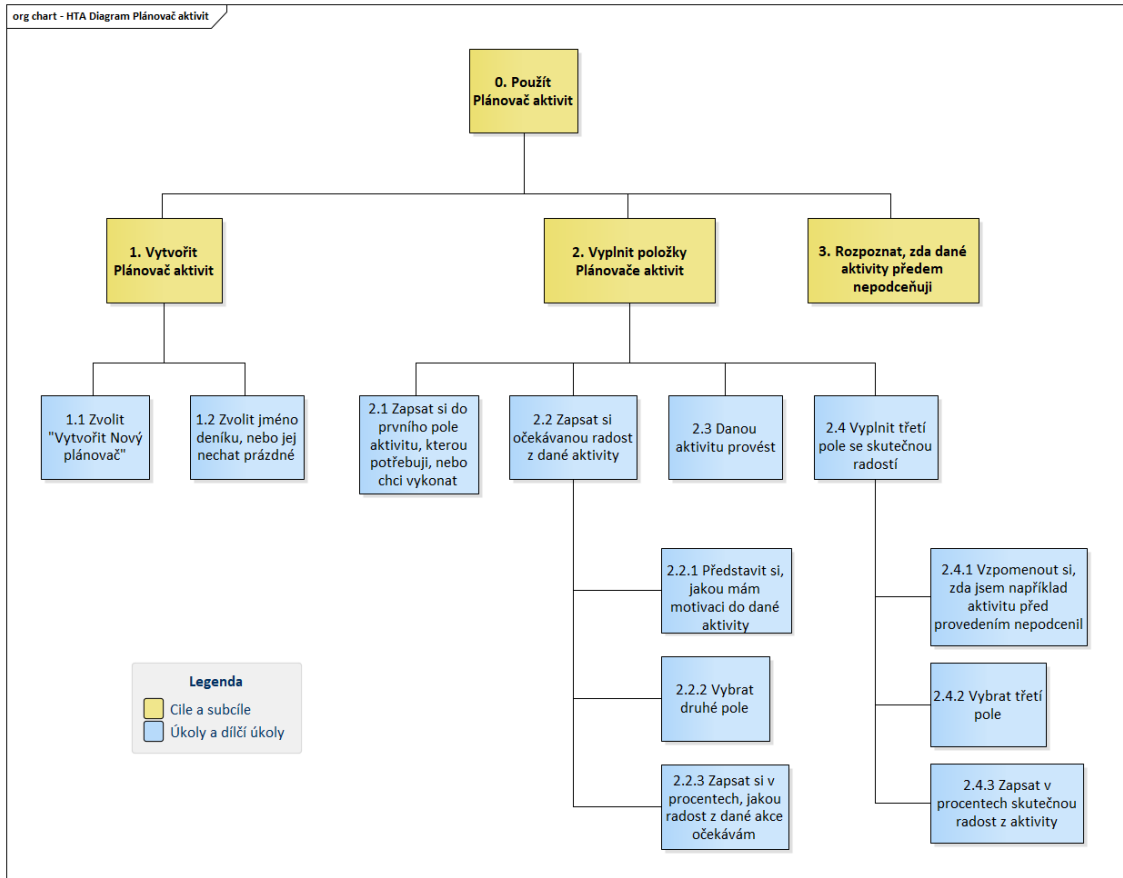
Žlutá pole v tomto diagramu znázorňují cíle jako tvorbu deníku myšlenek a jeho vyplnění, zatímco modrá pole popisují jednotlivé dílčí úkoly pro splnění těchto cílů.



Obrázek 6.7. HTA diagram – Deník myšlenek

## 6.4.2 Plánovač aktivit

Podobně je strukturován i diagram pro použití plánovače aktivit, kde hlavní odlišnost v používání spočívá v nutnosti provedení zvolené aktivity mezi vyplněním polí (úkol 2.3 na diagramu).



**Obrázek 6.8.** HTA diagram – Plánovač aktivit

# Kapitola 7

## Návrh řešení

### 7.1 Doménový model

Na obrázku 7.1 se nachází mnou navržený doménový model, který znázorňuje datovou strukturu systému. Z modelu je patrné, že ústřední entitou doménového model je `Uživatel (User)`, který interně představuje jak běžného uživatele, tak administrátory. K `Uživateli` se přímo vztahuje pět různých entit – `Test deprese (DepressionTestResult)`, `Plánovač aktivit (LethargyHelper)`, `Deník myšlenek (AutomaticThoughtsHelper)`, `Hlášení (Report)` a `Důvody suspendování účtu (BanReason)`.

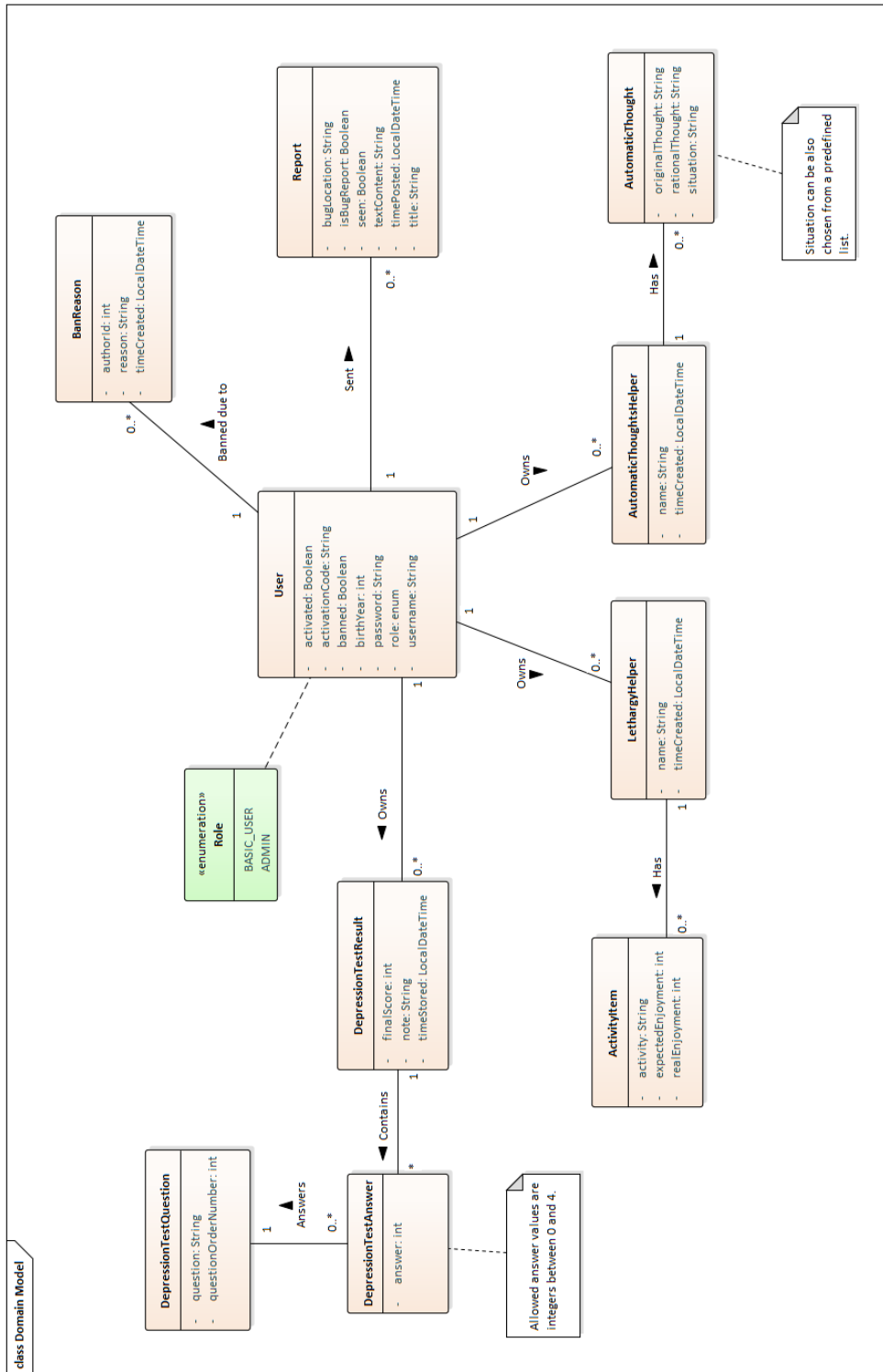
#### 7.1.1 Entita `Uživatele (User)`

Životní cyklus entity uživatele nemusí být na první pohled zřejmý. Považuji tedy za vhodné jej zde podrobněji popsat.

Tato entita vzniká při vytvoření aktivačního kódu, který následně uchovává. Při registraci obdrží následující informace: uživatelské jméno, heslo v hašované podobě a volitelně i rok narození. Rovněž jsou v této entitě údaje o tom, zda byl účet již aktivován a zda je účet v současné době suspendovaný (zablokovaný). V neposlední řadě také obsahuje roli uživatele v systému – tedy zda jde o běžného uživatele aplikace nebo o administrátora. Uživatelé vytvoření pomocí vygenerování nového aktivačního kódu jsou vždy běžní uživatelé.

Pokud bude účet uživatele již aktivovaný nebo zablokovaný, nemůže se daný uživatel s příslušným aktivačním kódem zaregistrovat. Pokud bude účet zablokovaný nebo nebude aktivovaný, nemůže se uživatel přihlásit.





Obrázek 7.1. Doménový model

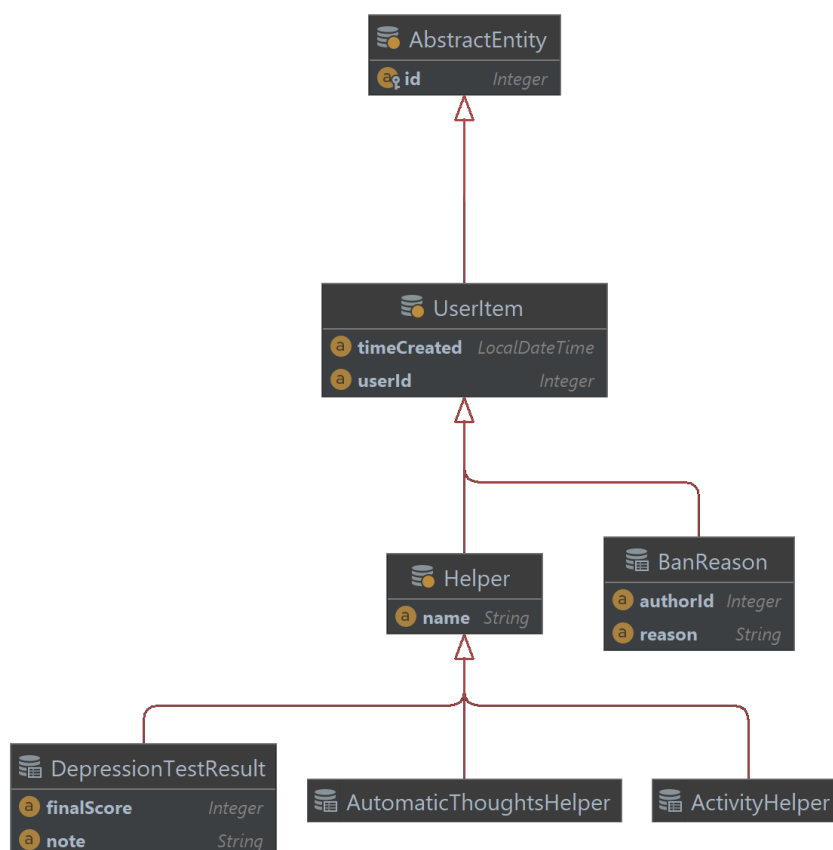
## 7.2 Vnitřní hierarchie dat

Správná volba dědičnosti tříd datového modelu je důležitou součástí k následné implementaci. Zabrání se tím duplikaci kódu a sníží se tím i množství možných chyb.

### 7.2.1 Třídy Helper a UserItem

Při pohledu na doménový model lehce zpozorujeme, že se v našem systému objevuje více objektů, které jsou vlastněny uživatelem a samy obsahují soubor dat, ke kterým bude uživatel potřebovat přistupovat. Konkrétně jde o objekty `DepressionTestResult`, `LethargyHelper` a `AutomaticThoughtsHelper`. Tyto objekty rozvíjí abstraktní třídu, kterou jsem nazval **Helper**.

Samotná abstraktní třída `Helper` obsahuje informaci o jméně objektu, ale mnohem podstatnější pro nás bude ve fázi implementace dědičnost těchto objektů. Samozřejmě nám ještě zbývá ještě informace o vlastnictví objektů. Tuto abstraktní třídu jsem nazval **UserItem** a zařadil mezi ně kromě třídy `Helper` ještě `BanReason`. viz obr. 7.2 Zde přítomná hodnota `userId` obsahuje identifikátor uživatele, kterému objekt patří.

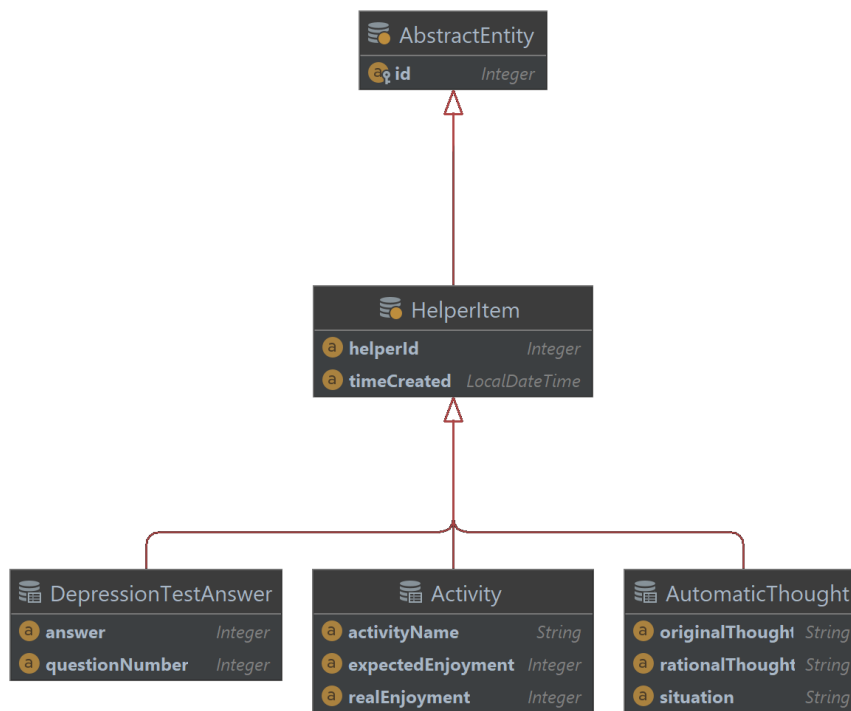


Obrázek 7.2. Vnitřní reprezentace tříd `Helper` a `UserItem`

### 7.2.2 Třída `HelperItem`

Nyní můžeme obdobně určit společnou abstraktní třídu pro již zmíněné soubory dat, které jsou „obsaženy“ v objektech typu `Helper`. Nazval jsem je originálně **HelperItem**.

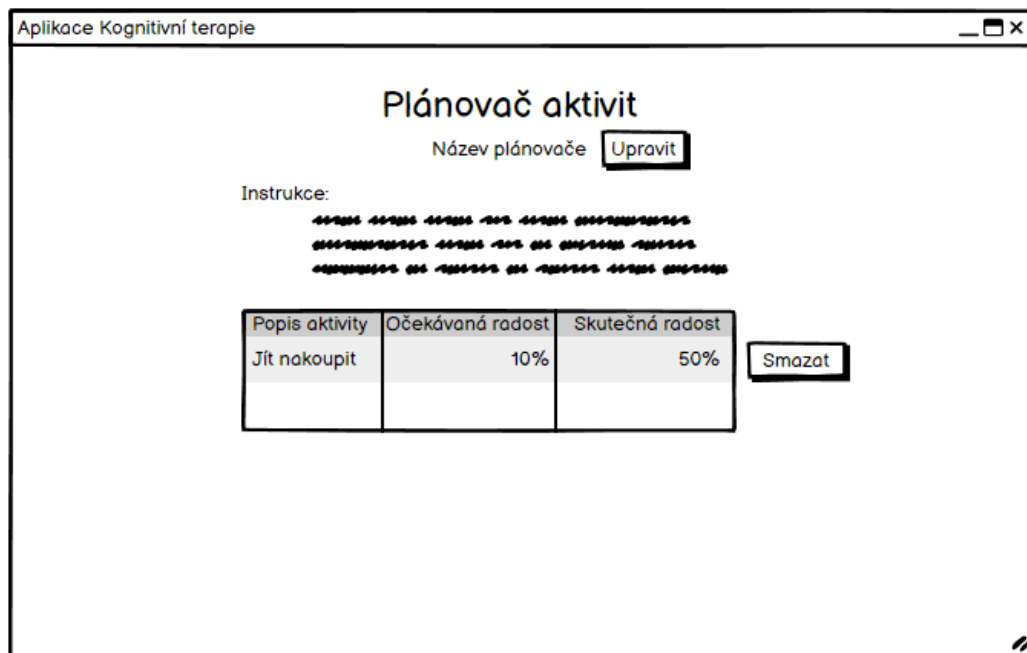
Jak je asi zřejmé, hodnota `helperId` je identifikátor příslušného objektu typu `Helper`, pod který spadá. Opět pro nás tato dědičnost bude užitečná při implementaci.



**Obrázek 7.3.** Vnitřní reprezentace třídy `HelperItem`

## 7.3 Prototyp aplikace

V rámci návrhu jsem vytvořil low-fidelity (lo-fi) prototyp hlavních částí aplikace v nástroji Balsamiq.



**Obrázek 7.4.** Prototyp Plánovače aktivit

The screenshot shows a window titled "Aplikace Kognitivní terapie" with a close button. The main heading is "Deník myšlenek". Below it is a text input field "Název deníku" followed by an "Upravit" button. Underneath is the label "Instrukce:" followed by three lines of placeholder text represented by asterisks. The central part of the form is a table with three columns: "Situace", "Původní myšlenka", and "Racionální myšlenka". The first row contains the prompts "Co se stalo?", "Co tě napadalo?", and "Co můžeš o situaci říci?". The second row contains three empty input fields. To the right of the table is a "Smazat" button. A small icon is visible in the bottom right corner of the window.

Situace	Původní myšlenka	Racionální myšlenka
Co se stalo?	Co tě napadalo?	Co můžeš o situaci říci?

**Obrázek 7.5.** Prototyp Deníku myšlenek

# Kapitola 8

## Rešerše technologií

Následuje analýza technologií, které pro mne byly z hlediska vývoje webových aplikací nejzajímavější.

### 8.1 Backend

U serverové části (také nazývané backendem) pro mě hraje hlavní roli především bezpečnost aplikace a rychlost vývoje webové aplikace.

- **Laravel (PHP)** – Jedná se o volně dostupný framework pro vývoj serverové strany aplikací. Laravel je oblíbeným frameworkem, neboť jednoduše řeší mnoho problémů, kterým vývojáři čelí jako například vestavěná autentizace příchozích požadavků, práce s databází a testování aplikace. Jelikož se jedná o framework založený na technologii jazyka PHP, není velkým překvapením, že může generovat i samotný kód HTML, pro který má Laravel připravené i základní šablony. Nic však nebrání tomu, aby implementoval pouze endpointy API. Velkou výhodou jak pro začátečníky tak pro zkušené vývojáře je přehledná dokumentace. Laravel je především určený pro aplikace s architekturou MVC (model–view–controller). Hojně se využívá pro single-page aplikace. Naopak nevýhodou může být pomalý vývoj v porovnání s ostatními technologiemi a nekompatibilita některých aktualizací s předchozími verzemi, což ztěžuje vývojářům držet krok s moderními trendy.[17]
- **Spring Boot (Java)** – Spring Boot je open-source rozšíření frameworku Spring, které zjednodušuje konfiguraci potřebnou pro vytvoření aplikace a má zabudovaný webový server pro rychlé spuštění. Mezi frameworky pro jazyk Java je Spring velmi oblíbený. Funguje na principu Dependency Injection (DI) a Inversion of Control (IoC), které vývojářům umožní jednoduché testování aplikace. Spring Boot si však vytvoří velké množství nepoužívaných závislostí, které pak zbytečně zvětšují výsledný soubor určený k nasazení. Umožňuje použití velkého množství modulů jako Spring JDBC, Security, objektově relační mapování nebo Test. Spring Boot není vhodný pro velké projekty a je potřeba mít alespoň základní znalosti frameworku Spring pro jeho efektivní využití.[18]
- **Django (Python)** – Django je full-stack framework napsaný v oblíbeném programovacím jazyce Python. Jeho cílem je zajistit vývojářům rychlý vývoj jejich webových aplikací. Jednou z jeho výhod je možnost znovupoužití již existujících komponent, a tedy předejít opakování kódu a umožnit jednoduchou škálovatelnost aplikace. Rovněž je považován za vcelku bezpečný, neboť vývojářům pomáhá předejít častým chybám jako jsou SQL injection, XSS nebo CSRF. Nevýhodou frameworku Django jako celku je jeho robustnost a složitost. Není tedy velmi intuitivní a vhodný pro jednoduché aplikace.[19–20]

- **Flask (Python)** – Dalším webovým frameworkem napsaným v jazyce Python je Flask, jehož popularita především kolem roku 2018 velmi vzrostla.[21] Na rozdíl od frameworku Django se jedná o tzv. micro-framework, čímž je myšleno, že je velmi minimalistický. Obsahuje pouze základní nástroje pro tvorbu webových aplikací a vše ostatní musí být přidáno externími moduly. Umožňuje tak vývojářům velkou nezávislost a flexibilitu. Framework Flask je oproti Django sice jednodušší, ale vzhledem k jeho modularitě může být například práce s databázemi pro začátečníky poměrně nepřehledná. Zároveň neobsahuje objektově relační mapování dat. Jedná se tedy o základní framework vhodný především pro menší webové aplikace.[19–20]

## 8.2 Databázový systém

- **MySQL** – MySQL je jedním z nejoblíbenějších open-source databázových systémů na světě. Vlastníkem systému MySQL je známá mezinárodní technologická společnost Oracle Corporation, která je zároveň jeho hlavním současným vývojářem. Systém MySQL je dostupný bezplatně, avšak společnost Oracle rovněž poskytuje placenou komerční verzi, která zahrnuje například pokročilé zabezpečení a technickou podporu a údržbu. Placenou verzi systému většinou využívají podniky, kterým tato služba může ušetřit čas a zvýšit tak efektivitu jejich práce. Díky své popularitě je široce podporovaný mnohými programovacími jazyky a platformami. Jde o velmi rychlý a bezpečný databázový systém vhodný i pro rozsáhlé objemy dat.

MySQL je relační databázový systém a uchovává tedy data v tabulkách, které mezi sebou mohou být logicky propojeny pomocí tzv. cizích klíčů.[22]

- **PostgreSQL** – Dalším oblíbeným zástupcem relačního open-source databázového systému je PostgreSQL. Tento systém je rovněž velmi rychlý a spolehlivý. Dává vývojářům poměrně vysokou míru svobody, neboť kromě běžných datových typů umožňuje i tvorbu vlastních datových typů. Nechybí zde ani předdefinované funkce a podpora cizojazyčných znaků. Přestože se PostgreSQL těší velké popularitě, tak je jeho dokumentace poněkud nekonzistentní a pro začátečníky tak může být někdy nepřehledná.[23]

- **MongoDB** – Jediným zástupcem tzv. NoSQL databázového systému je MongoDB. Zkratka NoSQL znamená Not only SQL. Od běžných relačních databázových systémů se NoSQL liší strukturou ukládání a získávání dat. Místo ukládání dat do tabulkových sloupců mohou takovéto systémy uchovávat data například ve stromových grafech, čímž se může změnit i jejich asymptotická efektivita při ukládání a vyhledávání. Často mohou fungovat na principu klíč–hodnota (key–value). Navíc jak název „Not only SQL“ napovídá, neznamená vždy, že se v takovéto databázi nikdy nedá vyhledávat pomocí SQL dotazů. Jde spíš jen o to zdůraznit, že takový systém není nutně omezený pouze na SQL dotazy, ale umožňuje i jiné způsoby získávání dat z databáze.

Systém MongoDB konkrétně ukládá data v dokumentech, které nápadně připomínají JSON objekty. To umožňuje rychlou a jednoduchou práci i s nestrukturovanými daty. MongoDB je open-source a dostupný zdarma. Existuje však i placená verze tohoto systému. Systém nevyžaduje pevně stanovené schéma dat, a ta tak mohou mít různou strukturu. Další předností tohoto systému je snadná horizontální škálovatelnost. MongoDB umožňuje vyhledávat data pomocí MongoDB Query Language (MQL), který se podobá SQL, ale je přizpůsobený pro práci s dokumenty.[23]

### 8.3 Frontend

- **React** – Jedním z populárních JavaScriptových frameworků je React. Jedná se o open-source nástroj pro tvorbu uživatelského rozhraní. Hlavním cílem tohoto frameworku je usnadnit vývojářům implementaci webových stránek a aplikací.

Struktura Reactu je založena na takzvaných komponentech. Tyto komponenty mohou představovat různé elementy nebo soubory elementů, které se mohou díky své implementaci používat opakovaně, což zaručuje snazší správu kódu.[24]

- **jQuery** – jQuery je JavaScriptová knihovna, která má za úkol usnadnit práci při vývoji klientského rozhraní webové stránky. Je známá pro svou snadnou správu událostí na stránce a asynchronní komunikaci s webovým serverem. Její popularita v poslední době upadá, je však stále užitečným nástrojem.[25]
- **Angular** – Angular nabízí robustní nástroje pro vysoce výkonné a škálovatelné aplikace. Je založen na jazyce TypeScript, což zvyšuje bezpečnost kódu a zjednodušuje správu kódu. Podobně jako framework React používá Angular k tvorbě aplikací spolupracující komponenty. Rovněž disponuje asynchronní komunikací se serverem. Velkou výhodou je i rozsáhlá komunita a kvalitní dokumentace.[26]

### 8.4 Výstup z rešerše technologií

#### Backend:

Framework Django se pro jeho vlastnosti jeví jako dobrá volba, avšak jak bylo zmíněno, jde o poměrně robustní a složité řešení pro menší aplikace. Navíc jsem si v rámci projektu chtěl vyzkoušet práci i se samostatným frontendovým frameworkem a Django je full-stack technologie. S frameworkem Flask mám již předchozí zkušenosti, avšak ne s jeho propojením s databází. Navíc absence objektivě relačního mapování dat se v mém projektu také zdá jako možná očekávaná komplikace.

Technologie Laravel i Spring Boot jsou obě snadno konfigurovatelné a vytvořené pro rychlý vývoj serverových částí webových aplikací. Laravel se zdá jako dobrá volba pro jeho jednoduché řešení rozličných problémů a jeho rozsáhlou dokumentaci, bohužel s tímto frameworkem, ani s frameworkem, od kterého je Laravel inspirován (Symfony), nemám žádné zkušenosti, a je tak pro mě v tomto projektu vhodnější volbou technologie **Spring Boot**, se kterou jsem byl na fakultě seznámen.

#### Databázový systém:

Data v mé webové aplikaci budou strukturovaná a mnoho funkcí bude záviset právě na vyhledávání v tabulkách, NoSQL databáze je tedy v tomto případě zbytečná. NoSQL databázové systémy jsou vhodné u projektů zpracovávající ohromné množství dat, například „Big Data“ projekty.

MySQL je populárnější databázový systém a bývá považován i za rychlejší než PostgreSQL. Samotné MySQL v základu nepodporuje ORM (objektově relační model), nicméně například ORM nástroj Hibernate umožňuje pracovat s MySQL databází jako by objektově relační databázi jednalo. Zatímco PostgreSQL má obecně silnou podporu objektově relačního mapování a obsahuje širší spektrum funkcí. Jako databázový systém jsem tedy zvolil **PostgreSQL**.



**Frontend:**

Pro frontendovou část aplikace jsem zvolil javascriptový framework **React JS**, neboť umožňuje vytvářet aplikace efektivně a s využitím méně kódu než aplikace psané pouze využitím čistého jazyka JavaScript. Další výhodou ve prospěch frameworku React je jeho popularita, kvůli které je vhodné práci s ním znát i pro budoucí využití. Potíží může být však pomalejší běh aplikace.

# Kapitola 9

## Implementace

Pro tvorbu webové aplikace jsem zvolil **vícevrstvou architekturu**, která se skládá se vzájemně spolupracujících vrstev. Konkrétně se jedná o vrstvu prezentační (uživatelské rozhraní), kterou představuje framework React, vrstvu pro logické zpracování dat (business logika) – Java Spring Boot a datovou vrstvu pro ukládání dat – PostgreSQL.

### 9.1 Backend

Pro serverovou část je použit jazyk **Java verze 8**. Pro automatický build projektu využívám nástroj **Maven**.

V souboru `src/main/resources/application.properties` jsou uvedeny konfigurace aplikace a hodnoty nutné k připojení backendu k databázi.

#### 9.1.1 Spring Boot

Pro umožnění použití funkcí jako webový server nebo REST nebo frameworku Spring Boot je v souboru `pom.xml` vložena následující závislost.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Objektově relační mapování (ORM) a práci s SQL dotazy nám umožňuje následující závislost.

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Soubor `src/main/java/cz/cvut/cta/security/SecurityConstants.java` obsahuje konstanty pro bezpečnostní obsluhu aplikace. Samozřejmě by tyto hodnoty musely být pro produkci v konfiguračních souborech a hesla v některém ze správce hesel (Key Manager).

### 9.2 Frontend

K implementaci klientské části byly použity jak třídní komponenty (`React.Component`) a funkční komponenty, což jsou běžné JavaScriptové funkce, které však vrací JSX (JavaScript Syntax Extension) produkující React elementy.

Tedy syntax takové funkce vypadá například takto:

```
function CodeCheckComponent() {
  // boolean
  const [isValid, setIsValid] = useState(false);

  ...

  // JSX syntax
  return (
    <div>
      ...
    </div>
  );
}

export default CodeCheckComponent;
```

## 9.3 Zabezpečení

### 9.3.1 Zabezpečení uživatelských hesel

Hesla k zabezpečení uživatelských účtů jsou hašována pomocí funkce **BCrypt**. Tento algoritmus je navržen tak, aby bylo obtížné odhalit původní heslo hrubou silou (tedy pomocí tzv. brute force attack). Hašování je jednosměrný proces, při kterém se vytvoří nový řetězec znaků, jenž je následně uložen v databázi. Ani osoby, které mají k databázi přístup, neznají heslo daného uživatele a při jeho ztrátě je potřeba jej změnit.

Funkce BCrypt navíc při hašování automaticky přidává náhodnou **kryptografickou sůl**, která dále zvyšuje bezpečnost, neboť zamezuje využití předem vypočítaných duhových tabulek (rainbow tables).

V mém případě využívám konkrétně implementaci třídy `PasswordEncoder`:

```
public BCryptPasswordEncoder(int strength)
```

Celočíselná hodnota „strength“ udává **počet iterací hašování**. Čím vyšší tato hodnota je, tím je exponenciálně delší výpočetní čas potřebný pro zakódování hesla. To také zároveň znamená vyšší odolnost proti útokům hrubou silou. Tato hodnota může být v rozmezí 4-31. Já jsem v mém případě zvolil hodnotu 11, neboť se zdá být dobrým kompromisem mezi rychlostí a bezpečností. Implicitní hodnota, pokud necháme funkci bez argumentu je 10, tudíž i tak by bylo zabezpečení na solidní úrovni.

```
@Bean
public PasswordEncoder passwordEncoder() {
  return new BCryptPasswordEncoder(11);
}
```

Bezpečnost zároveň závisí i na délce a složitosti hesla. Pro registraci uživatele tedy v aplikaci požaduji hesla alespoň o délce 5 znaků obsahující alespoň jedno číslo a malé písmeno.

Po přihlášení uživatele, se zadané heslo touto funkcí hašuje a výsledný hash se interně porovná s tím, který je uložen v databázi. Pokud jsou stejné, je zadané heslo považováno za platné.

```
public boolean matches(CharSequence rawPassword,
                        String encodedPassword) {
    return this.getPasswordEncoder().matches(rawPassword,
                                             encodedPassword);
}
```

### 9.3.2 Zabezpečení serverových požadavků

Pro správnou autorizaci uživatelů je v aplikaci použit **JSON Web Token (JWT)**. Tyto tokeny jsou vytvářeny na serverové straně aplikace, jak zobrazuje následující úryvek kódu. Můj token je digitálně podepsaný pomocí algoritmu SHA-256 a tajného klíče, který je znám pouze serveru.

```
public static String jwtGenerateAccessToken(@NotNull String username) {
    return JWT.create()
        .withIssuer(SecurityConstants.TOKEN_ISSUER)
        .withSubject(username)
        .withIssuedAt(new Date())
        .withExpiresAt(new Date(System.currentTimeMillis() +
                                SecurityConstants.JWT_TOKEN_EXPIRATION_TIME))
        .sign(SecurityConstants.JWT_TOKEN_ALGORITHM); // Signature
}
```

Po odeslání je token uložen v prohlížeči uživatele, kde jej následně klientská část aplikace připojí ke každému následujícímu požadavku pro prokázání identity.

Když server token obdrží zpět, je ověřen jeho digitální podpis a tvůrce. Pokud ověření proběhne úspěšně, je token dekodován a systém z něj získá uživatelský identifikátor.

```
// Create verifier
JWTVerifier verifier = JWT.require(SecurityConstants.JWT_TOKEN_ALGORITHM)
    .withIssuer(SecurityConstants.TOKEN_ISSUER)
    .build();

// Verify token
DecodedJWT decodedToken = verifier.verify(token);
```

Jedná se tedy o kompaktní zakódované řetězce znaků, které jsou zasílány mezi serverovou a klientskou stranou. JSON Web Token může přenášet různé druhy informací, které však nejsou implicitně zašifrované. Pokud je potřeba, aby byla data zašifrovaná, je potřeba tuto akci provést samostatně, nebo lze využít komunikačního protokolu SSL či TLS.

Jednou z informací, které může token uchovávat, je například čas expirace, tedy moment, po kterém je již token jako přihlašovací médium neplatný. Dále může samozřejmě nést data, pomocí kterých je odesílatel tokenu identifikován. V mém případě jde o přezdívku, které jsou v rámci systému mezi uživateli jedinečné. Samozřejmě je možné použít jakýkoliv jiný řetězec, pomocí kterého se dá uživatel jednoznačně identifikovat. Pokud by tato data byla někým změněna, byl by digitální podpis neplatný a server by tuto skutečnost detekoval. Na serverové straně je tento token integrovaný s funkcemi **Spring Security**, což je framework, který poskytuje metody autentizace a autorizace pro Spring aplikace.

K tomu, aby bylo možné JSON Web Tokeny v Java aplikaci použít, je třeba následující závislosti v souboru `pom.xml`.

```
<dependency>
  <groupId>com.auth0</groupId>
  <artifactId>java-jwt</artifactId>
  <version>4.3.0</version>
</dependency>
```

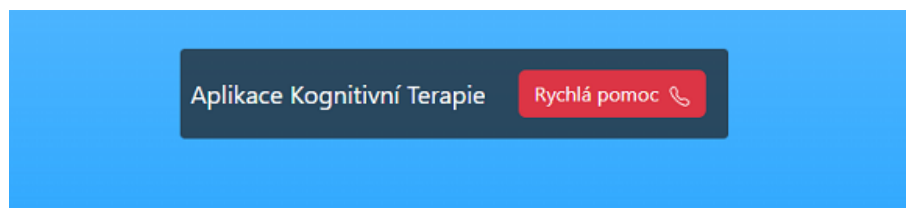
## 9.4 Grafické zpracování

Vzhled aplikace jsem nebral jako cíl bakalářské práce a zmínil jsem se o něm při volbě řešení. Vzhled by neměl narušovat cíl aplikace a v nejlepším případě cíl podporovat klidným prostředím. Výzkumy naznačují, že za uklidňující barvy mohou být považovány barvy modrá, růžová nebo zelená.[27] Modrá a zelená z toho důvodu, že nám připomínají přírodu, která je využívána v některých terapiích.[28]

Pro stylizaci prvků jsem zvolil nástroj **Bootstrap verze 5.2.3**, což je svobodný CSS framework umožňující stylizovat webové stránky. Bootstrap byl jako nástroj velmi užitečný a jsem rád, že jsem si práci s ním vyzkoušel. V některých případech mě limitovaly méně výstižné názvy tříd oproti verzi 5.0.0 a některé nekompatibilní React elementy (Například pro výběr roku narození – react-datepicker).

Další potřebné styly, kterými nástroj Bootstrap nedisponoval, jsem vytvořil v souboru `Cta-custom.css`, kde šlo hlavně o styly pro stránkování elementů.

Následují ukázky z programu po úpravě o připomínky z následného uživatelského testování.



*Vítejte na stránkách Kognitivní terapie*

### Přihlášení

[Vytvořit nový účet](#)

**Obrázek 9.1.** Uživatelské rozhraní pro přihlášení

Applikace Kognitivní Terapie
Rychlá pomoc
Odhlášt se

**Plánovač aktivit**

Můj plánovač aktivit – 9. 5. 2023

1. Máš něco v plánu, ale nechce se ti do toho? Odhadni, jak moc tě to bude/nebude bavit.

2. Až danou aktivitu provedeš, vrať se a porovnej, jak ses skutečně cítil/a.

Cíl: Třeba zjistíš, že máš z aktivit větší radost, než sis původně myslel/a.

Popis aktivity	Očekávaná radost (v %)	Reálná radost (v %)	Akce
<i>např.: Chěl bych jít do kina</i>			
Dojít si pro knihu	15 %	80 %	Smazat
Jít nakoupit	-40 %	50 %	Smazat
Co potřebuješ, nebo chceš udělat?	0 %	Vyplň po skončení aktivity %	

Obrázek 9.2. Uživatelské rozhraní Plánovače aktivit

Aplikace Kognitivní Terapie
Rychlá pomoc
Odhlásit se

**Deník myšlenek:**

Můj deník myšlenek - 9. 5. 2023

Trápí tě nějaká nedávná událost či myšlenka? Postupuj podle příkladu níže.

Cíl: Pokus se příště nepříjemné automatické myšlenky změnit co nejdříve, neboť uvidíš, že často nejsou pravdivé.

Situace	Automatické myšlenky	Pohled s odstupem
<p><b>Příklad:</b> Vedoucí na mě dnes kvůli mě chybě křičel</p> <p>Popiš, co se stalo.</p>	<p>Jsem k ničemu, vždy všechno zkazím, určitě mě vyrazí</p> <p>Co tě v dané situaci napadalo a jak ses cítil/a?</p>	<p>Už jsem spoustu věcí zvládl/a, takže není pravda, že bych zkazil/a všechno a byl/a k ničemu. Nemá důvod mě kvůli něčemu takovému vyhodit.</p> <p>Jsou tvé myšlenky z předchozího pole opodstatněné? Co bys člověku ve tvé situaci poradil/a?</p>

[< Zpět](#)

Obrázek 9.3. Uživatelské rozhraní Deníku myšlenek

Applikace Kognitivní Terapie

Rychlá pomoc

Odhlásit se

< Zpět

Najít uživatele podle ID

Najít uživatele podle jména

### Uživatelé:

Vygenerovat nový aktivací kód

ID	Username	Role	Aktivovaný	Zablokovaný	Akce
45	admin	ADMIN	Ano	Ne	Detail Zablokovat
46	user	BASIC_USER	Ano	Ano	Detail Odblokovat
58	lukasnovak7	BASIC_USER	Ano	Ne	Detail Zablokovat
77	Karel22	BASIC_USER	Ano	Ne	Detail Zablokovat

předchozí 1 2 3 4 5 další

Obrázek 9.4. Administrátorské rozhraní se seznamem uživatelů



## 9.5 Závěr implementace

V serverové části aplikace byly implementovány všechny navržené funkcionality.

V klientské části se nestihlo vzhledem k množství funkčních požadavků vytvořit uživatelské rozhraní pro možnost hlášení chyb, zobrazení profilu a vyplnění testu deprese.

Abych splnil kvalitativní požadavek pro šifrování citlivých údajů v databázi kromě šifrování databáze samotné, bylo potřeba využít technologie Transparent data encryption (TDE), neboli transparentní šifrování dat. Ve mnou zvoleném databázovém systému toto umožňuje například modul pgcrypto. Tato technologie vyžaduje pečlivou konfiguraci databáze a správu klíčů a certifikátů, kterou často provádí bezpečností odborníci.

Můj vlastní pokus o kódování textových dat pomocí algoritmu base64, aby data z databáze nebyla jednoduše čitelná, bohužel vzhledem k inverzi řízení (IoC) backendovým frameworkem Spring nebyl vhodný, neboť snižoval výkon aplikace.

Aby byl statický text klientské části lehce upravovatelný, bylo by potřeba využít systém pro správu obsahu jako například Strapi nebo Contentful, díky kterým by bylo například velmi jednoduché zobrazovat stránku různých jazycích.

# Kapitola 10

## Testování

### 10.1 Jednotkové testování

Unit testing, neboli jednotkové testování, je metoda, při které dochází k testování nejmenších částí, ze kterých se skládá aplikace. Zpravidla jde tedy o testování jednotlivých tříd, nebo funkcí (v kontextu OOP – metod). Důležité je, aby tyto části byly co nejvíce izolované od zbytku aplikace k otestování jejich samotných funkčností. Toho se často docílí simulování pomocných objektů s očekávanými vlastnostmi (mocking).

Jednotkové testy má většinou na starosti samotný vývojář, ten však u komplexnějších částí může z nezkušenosti nevědomě testovat pouze takové případy, pro které danou část aplikace vytvářel. V tento moment je vhodný tzv. black-box testing, při kterém daný tester nezná vnitřní strukturu testované části a zná pouze její účel.

### 10.2 Uživatelské testování

Uživatelské testy jsou druhem experimentu, při kterém se sleduje chování uživatelů určité aplikace, za účelem odhalení chyb a úskalí znemožňujících tuto aplikaci (efektivně) používat.

#### 10.2.1 Výběr testerů

Cílem bylo otestovat jak administrátorskou, tak uživatelskou část. Testeři byli vybíráni z řad studentů a rodinných příslušníků. Na základě zodpovězených otázek byli rozděleni do třech různých skupin A, B a C. Skupina C měla za úkol otestovat administrátorskou část aplikace. Skládala se z pěti osob, které disponovaly většími znalostmi z oblasti IT. Úkolem skupin A a B bylo otestovat část aplikace pro běžné uživatele. Skupinu A tvořily osoby aktivně obsluhující současné webové aplikace téměř každý den a skupinu B tvořily osoby, které neměly příliš zkušeností s každodenním obsluhováním webových aplikací. Obě skupiny A i B tvořilo též pět osob, z nichž část vykazovala příznaky deprese a druhá část žádné depresivní příznaky nevykazovala, ani nepocitovala.

#### 10.2.2 Prostředí

Jelikož jsem neměl možnost odstíněné místnosti, probíhaly testy individuálně v tichém prostředí u počítače za mé přítomnosti. Já jsem však do průběhu testů aktivně nijak nezasahoval a pouze předem instruoval testery ke sdílení pocitů a názorů při používání aplikace.

#### 10.2.3 Testovací scénáře

**Administrátorský test 1 – Zablokování uživatele:**

Postup:

1. Přihlaste se pomocí přidělených přihlašovacích údajů.

2. Vyberte sekci Uživatelé.
3. Vyhledejte pomocí pole „Najít uživatele podle jména“ uživatele „dummy“.
4. Zablokujte daného uživatele a uveďte jakýkoliv důvod blokace.
5. Jakkoli se přesvědčte, že je podle zobrazovaných dat daný uživatel zablokovaný.

**Výsledek:**

Skupina C: Testerů ze skupiny představující administrátory neměli s tímto úkolem téměř žádný problém a všichni splnili cíl. Dva z pěti testerů mi během úkolu sdělili, že způsob, kterým se aktualizuje seznam osob při vyhledávání uživatele není intuitivní, vzhledem k tomu, že je při testování instantní. Tento problém by mohlo vyřešit znázornění načítání.

Jako důkaz, že je uživatel zablokovaný, bralo zprvu většina testerů skutečnost, že se u daného uživatele tlačítko Zablokovat změnilo na Odblokovat. Následně se však dva z nich ještě přesvědčili v detailu uživatele, že se zde uložil i důvod blokace a příznak zablokování „Ano“.

**Administrátorský test 2 – Vygenerování nového aktivačního kódu:**

Postup:

1. Přihlaste se pomocí přidělených přihlašovacích údajů.
2. Vyberte sekci Uživatelé.
3. Vygenerujte nový aktivační kód.

**Výsledek:**

Skupina C: S přihlášením se k administrátorskému účtu a následným vygenerováním aktivačního kódu neměl nikdo z testerů potíží. Někteří však chybělo tlačítko pro automatické zkopírování kódu do schránky, na které jsou zvyklí z jiných aplikací, se kterými pracovali.

**Uživatelský test 1 – Registrace:**

Postup:

1. Vytvořte si nový uživatelský účet pomocí přiděleného aktivačního kódu a libovolně zvolte rok narození, nebo jej nechejte volný.
2. Po úspěšné registraci se odhlaste
3. Přihlaste pro ověření vytvoření účtu.

**Výsledek:**

Skupina A: Všichni testerů ze skupiny A úspěšně absolvovali celý scénář.

Skupina B: Tři z pěti testerů ze skupiny B mělo potíží dostat se z úvodní stránky s přihlášením na stránku registrace. To bylo pravděpodobně způsobeno tím, že nejsou na vložený odkaz „Vytvořit nový účet“ pod formulář s přihlášením zvyklí.

**Uživatelský test 2 – Použití Plánovače aktivit:**

Postup:

1. Přihlaste se pomocí přidělených přihlašovacích údajů.
2. Vyberte položku Plánovač aktivit.

3. Vytvořte nový plánovač aktivit a nezadávejte žádné jméno.
4. Změňte jméno nově vytvořeného plánovače.
5. Vyplňte dva řádky podle instrukcí na obrazovce.
6. Jeden z těchto řádků smažte.
7. Vytvořte další plánovač aktivit a následně jej smažte.

**Výsledek:**

Skupina A: Průchod tímto scénářem se povedl všem testerům z této skupiny podařil. Problém byl s pochopením funkce očekávané/reálné radosti. Někteří testéři navrhli místo procent používat k vyjádření emotikony, na které je dnes většina lidí z moderních aplikací zvyklá.

Některé testery také překvapilo, že implicitní jméno plánovače při vytvoření bez zadání jména, není stejné jako bylo uvedené v placeholderu pole.

Skupina B: Testéři ze skupiny B měli potíže z porozuměním poslednímu bodu úkolu, který může znít nesrozumitelně a navíc se v současném stavu dá plánovač aktivit smazat pouze v seznamu náhledů a ne když má uživatel plánovač otevřený.

Velká část testerů také chtěla zadat hodnotu procent reálné radost vyšší než 100%, na kterou se v testovací verzi hodnoty interně zaokrouhlovaly.

**Uživatelský test 3 – Použití Deníku myšlenek:**

Postup:

1. Přihlaste se pomocí přidělených přihlašovacích údajů.
2. Vyberte položku Deník myšlenek.
3. Vytvořte nový deník myšlenek a zadejte libovolné jméno.
4. Vyplňte dva řádky deníku podle instrukcí.

**Výsledek:**

Skupina A a B: Zde měly obě skupiny podobné potíže. Samotný proces se sice podařil všem, ale instrukce nebyly dostatečně zřejmé. Na otázku „Co tě o této situaci napadá, jak ses cítil/a?“ v druhém poli řada z testerů uvedla jednoduše: „blbě“ či použili podobný termín.

## ■ 10.2.4 Závěr uživatelského testování

Získané informace z uživatelského testování považuji za velmi přínosné, neboť mi mou aplikaci pomohly vidět z úhlu druhých. Tedy z úhlu potenciálních uživatelů. Mělo proběhnou testování prototypu, ten však nebyl dostatečně interaktivní a vypovídající.

Část testerů se podělila s návrhy pro vyšší intuitivnost některých detailů aplikace.

**Po testování jsem aplikoval následující změny:**

- Přidal ikony pro odhlášení a změně názvu místo klasického tlačítka.
- U deníku myšlenek jsem změnil druhé pole na „Co tě v dané situaci napadalo a jak ses cítil?“
- Název sloupce jsem změnil z „Původní myšlenka“ na „Automatické myšlenky“.
- Změnil jsem rozsah uvedené radosti z 0 až 100 na hodnoty -100 až 200.
- Přidal jsem v plánovači a deníku řádek s příkladem.

# Kapitola 11

## Závěr

Cílem práce bylo seznámit se s problematikou depresí a následně navrhnout a implementovat webovou aplikaci, která má za úkol lidem pomáhat zvládat prožívání deprese. Nejprve jsem se seznámil s odbornou literaturou zabývající se psychologíí a psychiatrií, ze kterých jsem se dozvěděl o diagnostice, příznacích, léčbě deprese a různých typech poruch souvisejících s depresí.

Následně jsem analyzoval existující aplikace zabývající se touto problematikou. Konkrétně se jednalo o aplikace Napanikař, VOS: Plán Mentálního Zdraví a MindDoc: Your Companion.

Na základě této analýzy jsem specifikoval systémové požadavky a případy užití. Dále byl proveden návrh včetně prototypu a řešerše technologií. Návrh jsem poté implementoval a pro serverovou část jsem použil framework Spring Boot, se kterým jsem již měl zkušenosti. Klientská část byla implementována v JavaScript frameworku React, se kterým jsem neměl zkušenosti téměř žádné.

Během implementace jsem aplikaci testoval a po dokončení serverové a klientské části bylo provedeno uživatelské testování.

Výsledkem je funkční prototyp aplikace.

### 11.1 Budoucí vylepšení

Dalším pokračováním práce by bylo dokončení zbývajících klientských částí aplikace a modernizace vzhledu, kromě toho ještě existuje řada rozšíření aplikace:

- Pro reálné produkční nasazení je třeba zajistit bezpečnost internetové komunikace pomocí protokolu HTTPS.
- Vytvořit off-line verzi aplikace, neboť by mohla přimět více lidí aplikaci používat.
- Pomocí systémového správce obsahu přidat možnost zobrazit aplikaci v jiných jazycích.
- Konzultovat aplikaci s odborníky a tím zvýšit její efektivitu při používání.

## Literatura

- [1] *Francis Bacon – Quotes, (On-line)*. [Přístup získán 10. ledna 2023] .  
<https://www.goodreads.com/quotes/63465-man-prefers-to-believe-what-he-prefers-to-be-true>.
- [2] *Jens Bjørneboe – Quotes, (On-line)*. [Přístup získán 10. ledna 2023] .  
<https://www.goodreads.com/quotes/3328047-he-who-hasn-t-experienced-a-full-depression-alone-and-over>.
- [3] *What is depression?, (On-line)*. [Přístup získán 20. prosince 2022] .  
<https://www.nimh.nih.gov/health/publications/depression>.
- [4] *World Health Organisation – Depression, (On-line)*.  
[Přístup získán 10. ledna 2023] .  
<https://www.who.int/news-room/fact-sheets/detail/depression>.
- [5] *Major Depression – Statistics, (On-line)*. [Přístup získán 20. prosince 2022] .  
<https://www.nimh.nih.gov/health/statistics/major-depression>.
- [6] Klára Látalová. *Suicidalita u psychických poruch*. první vyd. Grada Publishing, 2015. ISBN 978-80-247-4305-9.
- [7] Ján Praško. *Depresivní porucha a jak ji překonat*. první vyd. Galén, 2009. ISBN 978-80-7262-656-4.
- [8] *Duševní poruchy a poruchy chování*. druhé vyd. Psychiatrické centrum Praha, 2000. ISBN 80-85121-44-1.
- [9] *Smíšená úzkostně depresivní porucha, (On-line)*.  
[Přístup získán 22. prosince 2022] .  
<https://adicare.cz/deprese/smisena-uzkostne-depresivni-porucha/>.
- [10] Dagmar Seifertová. *Postupy v léčbě psychických poruch*. druhé vyd. Medical Tribune CZ, 2008. ISBN 978-80-8713-510-5.
- [11] *Aplikace Nepanikař, (On-line)*. [Přístup získán 9. ledna 2023] .  
<https://nepanikar.eu/aplikace-nepanikar/>.
- [12] *Co dnes víme o psychoterapii?, (On-line)*. [Přístup získán 11. ledna 2023] .  
<https://czap.cz/Blog-o-psychoterapii/13029207>.
- [13] *Aplikace VOS, (On-line)*. [Přístup získán 10. prosince 2022] .  
<https://vos.health/>.
- [14] *Aplikace MindDoc, (On-line)*. [Přístup získán 10. prosince 2022] .  
<https://minddoc.com/us/en>.
- [15] David D. Burns. *Cítit se dobře*. Eugenika, 2020. ISBN 978-80-8100-368-4.
- [16] Michael Neenan a WindyDryden. *Stručný přehled Kognitivní terapie (Z originálu: Cognitive Therapy in a Nutshell, 2006)*. portál, s.r.o., 2008. ISBN 978-80-7367-372-7.
- [17] *Core PHP vs. Laravel, (On-line)*. [Přístup získán 25. dubna 2023] .  
<https://positiwise.com/blog/core-php-vs-laravel-technology/>.

- [18] *Pros and Cons of Using Spring Boot, (On-line)*. [Přístup získán 28. dubna 2023] .  
<https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/>.
- [19] *Flask vs Django in 2023, (On-line)*. [Přístup získán 28. dubna 2023] .  
<https://www.netsolutions.com/insights/flask-vs-django/>.
- [20] *Flask Vs Django: Which Python Framework to Choose?, (On-line)*. [Přístup získán 28. dubna 2023] .  
<https://www.interviewbit.com/blog/flask-vs-django/>.
- [21] *Python Developers Survey 2018 Results, (On-line)*.  
[Přístup získán 29. dubna 2023] .  
<https://www.jetbrains.com/research/python-developers-survey-2018/>.
- [22] *What is MySQL?, (On-line)*. [Přístup získán 29. dubna 2023] .  
<https://www.talend.com/resources/what-is-mysql/>.
- [23] *Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others, (On-line)*. [Přístup získán 29. dubna 2023] .  
<https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>.
- [24] *Framework React, (On-line)*. [Přístup získán 16. května 2023] .  
<https://react.dev/>.
- [25] *Framework jQuery, (On-line)*. [Přístup získán 16. května 2023] .  
<https://jquery.com/>.
- [26] *Angular Framework features, (On-line)*. [Přístup získán 16. května 2023] .  
<https://angular.io/features>.
- [27] *The Role of Colors in Stress Reduction, (On-line)*.  
[Přístup získán 10. května 2023] .  
[https://www.researchgate.net/publication/314578015\\_The\\_Role\\_of\\_Colors\\_in\\_Stress\\_Reduction](https://www.researchgate.net/publication/314578015_The_Role_of_Colors_in_Stress_Reduction).
- [28] *The Art of Working With Nature in Nature-Based Therapies, (On-line)*. [Přístup získán 10. května 2023] .  
[https://www.researchgate.net/publication/342357858\\_The\\_Art\\_of\\_Working\\_With\\_Nature\\_in\\_Nature-Based\\_Therapies](https://www.researchgate.net/publication/342357858_The_Art_of_Working_With_Nature_in_Nature-Based_Therapies).





# Příloha A

## Zkratky

Následuje popis zkratk použitých v dokumentu.

API	Application Programming Interface
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DI	Dependency Injection
FR	Functional Requirement
HTA	Hierarchical Task Analysis
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IoC	Inversion of Control
JDBC	Java Database Connectivity
JS	JavaScript
JSON	JavaScript Object Notation
JSX	JavaScript Syntax Extension
JWT	JSON Web Token
KBT	Kognitivně Behaviorální Terapie
MVC	Model–View–Controller
NFR	Non-Functional Requirement
NoSQL	Not only SQL
OOP	Object-oriented programming
ORM	Object–Relational Mapping
OS	Operating system
REST	Representational State Transfer
SHA	Secure Hash Algorithm
SQL	Structured Query Language
SSL	Secure Sockets Layer
SSRI	Selective Serotonin Reuptake Inhibitor
TDE	Transparent Data Encryption
TLS	Transport Layer Security
UC	Use Case
XML	Extensible Markup Language
XSS	Cross-Site Scripting

### A.1 Soubory, které jsou součástí práce

CTA-Backend.zip	Zdrojový kód serverové části aplikace v Java Spring Boot
CTA-Frontend.zip	Zdrojový kód klientské části aplikace v ReactJS