ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bulant** Jméno: **Martin** Osobní číslo: **498932**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra teorie obvodů**

Studijní program: **Lékařská elektronika a bioinformatika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Kalkulátor baterie neuropsychologických vyšetření

Název bakalářské práce anglicky:

Calculator of neuropsychological test battery

Pokyny pro vypracování:

Neuropsychologická vyšetření jsou prováděna sadou testů, které zkoumají výkon ve specifických kognitivních doménách (paměť, inteligence, představivost, řečové dovednosti aj.). Vyhodnocení jednotlivých subtestů je prováděno psychologem a pomocí normativních tabulek převáděno a přepočteno do normovaných škál (IQ, percentily, Z-skóre, T-skóre, CELF-R).
1) Vytvořte aplikaci, kalkulátor, vyhodnocení baterie neuropsychologických vyšetření
2) Součástí software je databáze pacientů
3) Dbejte na zabezpečení citlivých informací

Seznam doporučené literatury:

[1] Kantowitz, B., Roediger III, H., & Elmes, D. (2014). Experimental psychology. Cengage Learning. (kapitola 3)
[2] FERJENČÍK, J. (2010). Úvod do metodologie psychologického výzkumu. Praha: Portál, 255 s.
[3] Mareš, P., Rabušic, L., & Soukup, P. (2015). Analýza sociálněvědních dat (nejen) v SPSS. Masarykova univerzita.
[4] Brandt, Mark J., et al. 'The replication recipe: What makes for a convincing replication?.' Journal of Experimental Social Psychology 50 (2014): 217-224.
[5] Simmons, Joseph P., Leif D. Nelson, and Uri Simonsohn. 'False-positive psychology: undisclosed flexibility in data collection and analysis allows presenting anything as significant.' (2016).
[6] Van't Veer, Anna Elisabeth, and Roger Giner-Sorolla. 'Pre-registration in social psychology—A discussion and suggested template.' Journal of experimental social psychology 67 (2016): 2-12.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Radek Janča, Ph.D. katedra teorie obvodů FEL

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

Datum zadání bakalářské práce: **31.01.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Radek Janča, Ph.D.	doc. Ing. Radoslav Bortel, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	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

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra teorie obvodů

Bakalářská práce

Kalkulátor baterie neuropsychologických vyšetření

Autor: Martin Bulant

Vedoucí práce: Ing. Radek Janča, Ph.D.

Studijní program: Lékařská elektronika a bioinformatika

Praha 2023

Prohlášení

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 ....

Podpis autora práce

Poděkování

Děkuji Ing. Radku Jančovi, Ph.D. za předání zkušeností a cenných rad při vypracování bakalářské práce. Neméně děkuji PhDr. Alici Maulisové, Ph.D a Mgr. Kateřině Bukačové za odborné konzultace v rámci řešení projektu s podporou Technologické agentury ČR (TL03000328) a Ministerstva zdravotnictví koncepčního rozvoje výzkumné organizace 0064203 FN Motol. Děkuji své rodině za psychickou podporu a trpělivost.

Abstrakt

Jedním z nejčastějších neurologických onemocnění u dětí je epilepsie, která může mít za následek změny ve výkonnosti kognitivních schopností. Měření kognitivní výkonnosti je v současnosti prováděno převážně za pomoci inteligenčních testů. Avšak v poslední době stoupá požadavek po přesnějším vyšetření jednotlivých funkcí širokého kognitivního profilu. Proto vzniká unikátní neuropsychologická testovací baterie pro děti ve věku 6–19 let, která si klade za cíl stát se standardním nástrojem pro posouzení kognitivní výkonnosti. Vyšetření je prováděno psychologem a výsledky jednotlivých subtestů je nutno manuálně porovnávat s normativními tabulkami metrických škál.

Ke zrychlení a přesnějšímu vyhodnocení testů bylo cílem vytvořit desktopovou aplikaci (kalkulátor) pro vznikající neurologickou testovací baterii, která umožní nejenom snadné vyhodnocení testů, ale i vedení záznamů o jednotlivých vyšetření pacientů.

Byla vytvořena aplikace obsahující databázový systém s grafickým uživatelským rozhraním. Databázový systém pracuje se dvěma typy databází, uživatelskou a pacientskou. Uživatelská databáze je určena pro správu a přístup psychologů k pacientské databázi. Uživatelské údaje a citlivé údaje o pacientech jsou zabezpečeny víceúrovňovým šifrováním za využití symetrického a asymetrického šifrování a kryptografického hashovaní. Grafické uživatelské rozhraní aplikace bylo vytvořeno podle současných trendů s důrazem na jednoduchost a uživatelskou přívětivost. Výsledky vyšetření pacientů lze vizualizovat v metrických škálách a reportovat ve formátu PDF.

Vzniklá aplikace splňuje funkční a bezpečnostní požadavky, byla otestována a je připravena pro budoucí implementaci skutečných normativních dat.

Klíčová slova: kalkulátor, aplikace, neuropsychologické testování, dětské normy

Abstract

One of the most common neurological diseases in children is epilepsy, which can cause changes in the cognitive performance. Currently, the measurement of the cognitive performance is mainly carried out using intelligence tests. However recently there has been a growing demand for a more accurate examination of the individual functions of the broad cognitive profile. Therefore, a unique neuropsychological test battery for children aged 6-19 years is created, which aims to become a standard tool for assessing cognitive performance. The examination is performed by a psychologist and the results of individual subtests must be manually compared with normative tables of metric scales.

The goal was to create a desktop application (calculator) for the emerging neurological test battery to accelerate and improve the accuracy of test evaluations, which will not only enable easy evaluation of tests, but also facilitate the management of individual patient examination records.

An application, containing a database system with a graphical user interface, has been created. The database system operates with two types of databases, user and patient ones. The user database is designed for psychologists to manage and access the patient database. User data and sensitive patient data are secured with multi-level encryption using symmetric and asymmetric encryption and cryptographic hashing. The graphical user interface has been created according to current trends, emphasizing simplicity and user-friendliness. Patient examination results can be visualized on metric scales and reported in PDF format.

The resulting application meets functional and security requirements, has been tested and is ready for future implementation with actual normative data.

Keywords: calculator, application, neuropsychological testing, children's norms

Seznam obrázků

Obsah

1. Úvod

Nejčastějšími onemocněními mozku objevujícím se v dětském věku je epilepsie a onkologické onemocnění, které mohou mít za následek negativní změny ve výkonosti kognitivních schopností [1]. Oblasti funkčních částí mozku vykazující deficit mohou být důležitým indikátorem přítomností epileptického ložiska nebo novotvaru. Měření kognitivního výkonu je tedy důležitým ukazatelem progrese onemocnění nebo naopak efektu léčby.

Měření výkonosti jednotlivých kognitivních schopností je v České republice prováděno z velké části pomocí inteligenčních testů [2]. Avšak v poslední době vzniká potřeba o vytvoření testovací baterie, která by umožnila přesnější vyšetření jednotlivých funkcí kognitivního profilu pacientů [2]. Samotné vyhodnocení testů je obecně časově náročné a vyžaduje porovnání výsledků s normativními tabulkami.

Motivací práce je vytvořit aplikaci (kalkulátor), který umožní automatické vyhodnocování testů a vedení pacientské databáze pro retrospektivní sledování změn výkonu kognice v jednotlivých neuropsychologických doménách.

1.1. Psychologické testování

Měření výkonosti jednotlivých kognitivních schopností je v České republice prováděno z velké části pomocí inteligenčních testů, např. Wechslerova inteligenční škála (WISC-III) [3] nebo pomocí inteligenční a vývojové škály pro děti IDS [4]. Avšak v poslední době vzniká potřeba o vytvoření testovací baterie, která by umožnila přesnější vyšetření jednotlivých funkcí kognitivního profilu [2]. Klinicky validní soubor zkoušek umožní přesně popsat profil kognitivních schopností jedince, stanovit jeho silné a slabé stránky, sestavit konkrétní kompenzačně-rehabilitační plán eventuálně dílčích obtíží a následně jeho efektivitu ověřit opakovaným testem [2].

Prozatím je možno provádět detailnější testování pouze některých z kognitivních funkcí, např. paměti TOMAL-2 (Testování paměti a učení) [5]. Kognitivní výkon jedince závisí např. na jeho věku a zralosti mozku, proto musí být výsledek testu vždy vztahován k průměrným výsledkům stejně staré zdravé populace, tzv. normativním datům. U mnoha dalších využívaných testů je věkové rozpětí normativních dat pro českou populaci úzké, a proto se v některých případech využívají testy ze zahraničí, které se ale neopírají o normy pro českou populaci a výsledky proto mohou být nepřesné či zavádějící [2].

Nově vznikající neuropsychologická testovací baterie si klade za cíl o vytvoření nového komplexního nástroje, který umožní detailnější testování jednotlivých funkcí kognitivního profilu

u dětí od 6 do 19 let. Tento soubor zkoušek se bude opírat o normativní data vztahující se k populaci ČR a umožní přesné popsání profilu kognitivních schopností jedince. Nová testovací baterie bude testovat kognitivní domény, které jsou popsány v kapitole 1.2.

1.2. Domény kognitivních testů

Nová testovací baterie testuje níže uveden kognitivní domény [2]:

1. **Paměť a učení**

   Paměť a pozornost jsou navzájem propojené a společně vytváří nástroj pro získání a zpracování informace z okolního světa. Učení představuje proces získávání nových informací, které jsou pamětí uloženy pro pozdější využití [6].

   Rozlišujeme několik druhů paměti [6]:

   - **Smyslová paměť** – Je to typ paměti umožňující reagovat na informace ze smyslů vyvolané stimuly. Smyslová paměť je nejrychlejší paměťový proces.
   - **Krátkodobá paměť**– Je to typ paměti, která ukládá informaci do tzv. krátkodobé paměti. To znamená, že daná informace je po určité časové době zapomenuta.
   - **Dlouhodobá paměť** – Umožňuje uložení informace na delší časovou periodu.

2. **Pozornost**

   Pozornost společně s pamětí je nezbytnou funkcí mozku [7]. Tato funkce umožňuje udržení chování na daný cíl, tedy umožňuje udržení koncentrace na daný podnět, jev či objekt [8]. Bez existence pozornosti by nebylo možné získávat nové informace [7].

   Pozornost je popisována z níže uvedených hledisek [9]:

   - **Kapacita** definuje množství jevů či informací, na které se může člověk soustředit. Toto množství je u každého jedince různé a může být i ovlivněna různými vlivy (např. únava, deprese anebo rušivými vlivy vnější prostředí) [10].
   - **Bdělost** (vigilita) definuje připravenost jedince na koncentraci na určitý jev, objekt nebo podnět.
   - **Udržení** (vigilance) je dlouhodobé udržení připravenosti jedince na koncentraci na určitou jev.
   - **Selektivita** či **výběrovost** je schopnost vybrat podnět nebo podněty z hlediska důležitosti, kterým bude dávána pozornost.
   - **Soustředění pozornosti** popisuje výběr zaměření pozornosti na daný jev či objekt z dalších.
   - **Distribuce** popisuje rozdělení pozornosti mezi více podnětů nebo jevů. Mechanismus, který umožňuje distribuci pozornosti, je označován jako tzv. **přepínání** pozornosti.

3. **Exekutivní funkce**

Vysvětluje výsledek cíleného a koordinovaného využití vícero kognitivních procesů (funkcí) pro efektivní a přizpůsobivou reakci na nečekanou situaci [11]. Exekutivní funkce jsou popisovány širokým komplexem vyšších psychologických funkcí, které zasahují do dalších kognitivních domén a do určité míry se s některými i překrývají (např. pracovní paměť, pozornost) [9]. Do tohoto komplexu spadá plánování, schopnost řešení problémů, vytváření hypotéz, kognitivní flexibilita, rozhodování, regulace, úsudek, schopnost využívat zpětnou vazbu a sebepercepce (sebepojetí) [9].

Exekutivní funkce tvoří řada podřízených kognitivních procesů, kde nejdůležitějším procesem je pracovní paměť [9].

4. **Řeč**

Řeč tvoří základ sociální interakce člověka a slouží pro dorozumívání a mezilidskou komunikaci [12]. Může být využita i jako nástroj pro zachycení, vyjádření či sdělení výsledků myšlení [13].

Rozlišujeme dva druhy řeči [14]:

- **Vnitřní řeč** (neboli tzv. řeč sama sobě) je forma řeči, kdy nejsou myšlenky projevovány do okolí [13]. Tento druh řeči je zaměřen pouze na sebe sama. Je často rychlý, útržkovitý a formován buď pomocí samotných slov nebo jednoduchých vět [14]. Díky tomu může docházet k problémům při přenosu myšlenek z vnitřní do vnější řeči [14].
- **Vnější řeč** je druh řeči, při které jsou myšlenky projevovány do okolí, a to v mluvené nebo písemné podobě [14]. Rozdíl mezi písemnou a mluvenou formou je, že při mluvené řeči jsou využívány doprovodné projevy (např. gestikulace, mimika atd.) [14]. U psané řeči tyto doprovodné projevy chybí, a proto tato forma řeči vyžaduje přesnějšího vyjadřování [13].

5. **Zraková percepce**

Zrakové vnímání (percepce) je u každého člověka individuální a je vyvíjeno už od narození [15]. Je to mimořádně komplexní proces, kdy při zpracování zrakových podnětů se zapojuje značná část mysli (kognitivní, exekutivní a emoční procesy) a značná část mozku [15]. Hlavní cílem zrakově-percepčních funkcí je zmapovat v co nejkratším čase sledovanou scénu, získat relevantní a smysluplné údaje [15]. Získané informace o předmětech a událostech v okolí mohou následně pomoci při účelném jednání [15]. Zrakově-percepční funkce jsou zodpovědné za orientaci a pohyb v prostoru, vnímání pohybu, kresbu nebo vytváření 2D a 3D modelů [15]. Do těchto funkcí spadá i vnímání barev anebo objektů, a to jak z pohledu vlastností (velikost, barva, poloha atd.), tak i tvarů [15].

6. **Motorická funkce**

   Motorické dovednosti jsou jedním ze základních kritérii, které jsou zkoumány a popisovány neuropsychologem [9]. Motorika je definována jako souhrn všech pohybových předpokladů, které vedou k úspěšné pohybové činnosti [16]. Pohybové schopnosti jsou u všech lidí odlišné a jsou vystihovány jako vrozené schopnosti, které ale můžou být ovlivněny (např. osobností, temperamentem, okolím atd.) [17].

   Rozlišujeme **jemnou** a **hrubou** motoriku [16].

   - **Hrubá motorika** je soubor všech pohybových aktivit člověka. Spadá sem i udržování stability a koordinace těla. Je zajišťována pomocí velkých svalových skupin [16].
   - **Jemná motorika** je označení pro pohyby ruky a prstů, které umožňují uchopení a manipulaci s malými předměty [17]. Je zajišťována pomocí malých svalových skupin, často ve spolupráci zrakovou kontrolou. Jemná motorika umožňuje tzv. **grafomotorický projev** (kresba, písmo) [17].

7. **Sociální kognice**

   Sociální poznání (kognice) je důležitou součásti sociální psychologie a zahrnuje způsoby, kterými si lidé vytváří úsudky, dojmy o sociálním světě (sebe sama, jiné osoby či skupiny) [18]. Do této domény patří i poznatky o vztazích v sociálním světě nebo hledání příčin chování lidí [18].

1.3. Normativní kontroly

Normativní kontrola je důležitým nástrojem při zjišťování výkonosti pacienta v jednotlivých kognitivních doménách. Je to nástroj sloužící ke srovnání a zjištění odlišnosti dosaženého výsledku testování pacienta vzhledem k vyšetřením provedených na široké škále vzorků z dané populace. Obvykle je referenční soubor respondentů tříděný podle různých kritérií (např. věk, pohlaví, národnost atd.) [19].

U výsledků testování (hrubého skóre) daného pacienta je kontrolována odlišnost od normy referenčního souboru, kde norma referenčního souboru je chápána jako průměrný výkon nebo reakce respondentů [19]. Typicky lze referenční výsledky reprezentovat pomocí tzv. Gaussovy křivky, kde je možno jednoduše vidět variabilita nebo pásmo normy a abnormy [20], [21].

Nově vznikající testová baterie bude konormována s intelektovým testem WASI-II [22]. Společně s baterii bude vytvořena i paralelní testová verze, která bude sloužit pro provádění retestů pacientů s krátkým časovým odstupem. Referenční soubor bude tvořen 1200 výsledků testů dětí v rozmezí věku 6–19 let. Klinickou skupinu vzorku referenčního souboru bude tvořit 30 dětí s onkologickým onemocněním a 30 dětí s epilepsií.

1.4. Metrické škály

1.4.1. Normální rozdělení pravděpodobnosti

Normální rozdělení pravděpodobnosti neboli Gaussovo rozdělení je jedno z nejdůležitějších pravděpodobnostních rozdělení, které je typické pro řadu biologických nebo psychických jevů [23]. Je to spojité rozdělení pravděpodobnosti, které je definováno dvěma parametry: **střední hodnota μ** a **rozptyl σ^2**.Oba parametry jsou reálné a rozptyl je navíc kladný ($\sigma > 0$). Střední hodnota μ je místo, kde je střed a maximální hodnota Gaussovy křivky (graf, který vizualizuje normální rozdělení) a rozptyl definuje, jak ostrý tvar bude mít Gaussova křivka. To znamená, že čím větší bude mít Gaussovo rozdělení rozptyl, tím bude mít Gaussova křivka plošší tvar (viz. Obrázek 1) [24].

Hustota pravděpodobnosti Gaussova rozdělení je definována následovně:

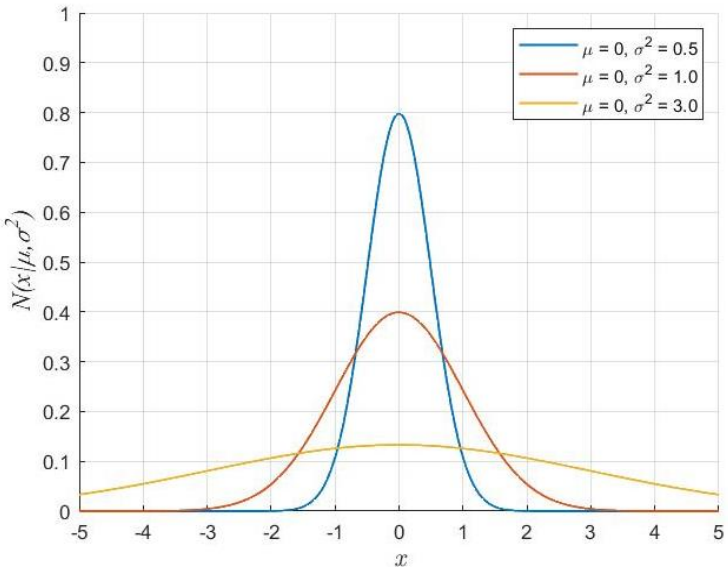
$$N(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \quad (1)$$

Speciální případem normálního rozdělení pravděpodobnosti tzv. **normované** neboli **standardizované normální rozdělení** $N(0,1)$, kde $\mu = 0$ a rozptyl $\sigma^2 = 1$ [25]. Hustota pravděpodobnosti standardizovaného normální rozdělení je pak definována následovně:

$$N(x|0,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, -\infty < x < \infty \quad (2)$$

Normální rozdělení má následující vlastnosti [24]:

- Gaussova křivka je symetrická a většina hodnot se soustřeďuje kolem střední hodnoty.
- Můžeme vždy určit procento hodnot spadající do určitého intervalu kolem střední hodnoty. V intervalu $\pm 1\sigma$ leží 68,26 % hodnot, v intervalu $\pm 2\sigma$ leží 95,34 % hodnot a v intervalu $\pm 3\sigma$ spadá 99,7 % hodnot [23].
- Sečteme-li více veličin, které pochází z normálních rozdělení. Výsledná veličina má opět náhodné rozdělení [25].
- Pokud k veličině, která pochází z normálních rozdělení, přičteme nebo odečteme libovolné číslo, nebo ji vynásobíme nebo vydělíme nenulovým číslem. Získáme znova veličinu pocházející z normálního rozdělení [25].



Obrázek 1: Vizualizace Gaussova rozdělení s různou hodnotou rozptylu

1.4.2. Standardizované skóre

Často jsou dosažené výsledky testů pacientů převáděny do standardizovaného skóre, které umožňuje lehčí interpretaci či srovnání s výsledky testů pocházející z odlišných distribucí (distribuce s odlišnou střední hodnotu nebo směrodatnou odchylkou) [23]. Typicky je využíváno **z-skóre**, který převádí hodnoty x z původní distribuce do standardizovaného normálního rozdělení $N(0,1)$ [23]. Hodnota z-skóre je vypočítána jako podíl rozdílu původní hodnoty x a střední hodnoty μ (původní distribuce) směrodatnou odchylkou σ (původní distribuce) [26].

$$z = \frac{(x-\mu)}{\sigma} \tag{3}$$

Hodnota z-skóre popisuje, o kolik směrodatných odchylek je původní hodnota x pod nebo nad průměrem původní distribuce [23].

Existují i další druhy standartních skórů, např.:

- T-skóre – střední hodnota μ'=50 a směrodatná odchylka $\sigma' = 10$
- Steny – střední hodnota $\mu' = 5{,}5$ a směrodatná odchylka $\sigma' = 2$
- Staniny – střední hodnota μ'= 5 a směrodatná odchylka $\sigma' = 2$
- IQ skóre – střední hodnota μ'= 100 a směrodatná odchylka $\sigma' = 15$

Tato standartní skóre je možno vypočítat za pomoci upravení vzorce pro výpočet z-skóre $z=\frac{(x-\mu)}{\sigma}$ (3). Vynásobením hodnoty z-skóre požadovanou směrodatnou odchylkou σ' a přičtením požadované střední hodnoty μ' je získáno požadované standartní skóre x'.

$$x' = \mu' + \sigma' z \tag{4}$$

1.4.3. Percentil

Jak bylo uvedeno v kapitole 1.4.2, jsou výsledky testů pacientů převáděny do standardizovaného skóre umožňující jednoduší interpretaci výsledků vyšetření pacientů. Další nástrojem pro ulehčení interpretace výsledků testů pacientů je percentil. Percentil je speciální kvantil, který rozděluje vzestupně setříděný statistický soubor obsahující vyšetření (např. paměti) velkého vzorku populace na dvě části a udává kolik procent populace má horší výsledek testu v testovaném kritériu (např. paměti) než vyšetřovaný pacient [27]. Dosáhne-li pacient v příslušném testovaném kritériu hodnoty 70 percentil, znamená to, že jeho výsledek testu je lepší než 70 % populace [28].

1.5. Motivace a cíl

Neuropsychologické vyšetření je prováděno sadou testů, které zkoumají výkonost v jednotlivých kognitivních doménách (např. řeč, paměť, pozornost atd.). Vyšetření je prováděno psychologem, kdy jedno vyšetření se pohybuje okolo 120 minut. Výsledky testování jsou manuálně přepočítávány do jednotlivých standartních skórů (percentil, z-skóre, t-skóre atd.), které jsou následně pomocí normativních tabulek vyhodnocovány a vizualizovány.

Cílem této bakalářské práce je vytvoření aplikace (kalkulátoru), která usnadní složité zpracování a vizualizaci výsledků testů pacientů, včetně možnosti vedení záznamů o pacientech a jejich vyšetření.

1.5.1. Požadavky na aplikaci

1. Databáze pro uložení údajů a testů pacientů
2. Zabezpečení citlivých informací pacientů
3. Grafické uživatelské rozhraní pro práci s kalkulátorem, vizualizaci údajů a výsledků testů pacientů.
4. Export v podobě PDF reportu

2. Návrh aplikace

2.1. Funkční požadavky

Aplikace vyvíjená v rámci této bakalářské práce bude obsahovat databázi pacientů a výsledků jejich vyšetření, na jejímž základě bude kalkulátor porovnávat výsledky s normami. Po konzultaci s budoucími uživateli a vedoucím bakalářské práce vyplynuly níže uvedené funkční požadavky na aplikaci.

Registrace nového uživatele

Při vytváření nového uživatele je požadováno zadat níže uvedené údaje uživatele:

- Jméno a příjmení
- Email
- Uživatelské jméno
- Heslo

Aplikace provede kontrolu, zda vložené uživatelské jméno nebo email již nejsou zaregistrovány. Pokud zadané údaje neexistují v uživatelské databázi, dojde k přidání nového uživatele do uživatelské databáze. Aplikací je následně vytvořena pacientská databáze pro nového uživatele, která slouží k ukládání záznamů o pacientech a jejich jednotlivých vyšetřeních. V opačném případě je zobrazena chybová zpráva.

Heslo je aplikací využito pro ověření totožnosti uživatele při přihlášení a k zabezpečení citlivých dat pacientů (podrobněji vysvětleno v kapitole 3.3.1). V případě zapomenutí hesla je email použit pro zaslání ověřovacího kódu.

Přihlášení uživatele

Při přihlášení do aplikace je vyžadováno po uživateli:

- Uživatelské jméno
- Heslo

Zadané údaje se zkontrolují s údaji uloženými v uživatelské databázi. Při shodě dojde k přihlášení uživatele, v opačném případě se zobrazí chybová zpráva. Po třetím neúspěšném pokusu o přihlášení, neumožní aplikace další přihlášení.

Přidání nového záznamu pacienta

Pro přidání nového záznamu pacienta je potřeba zadat níže uvedené údaje pacienta:

- Jméno a příjmení
- Datum narození
- Rodné číslo
- Pohlaví
- Dosažené vzdělání pacienta

Aplikace na základě rodného čísla provede kontrolu unikátnosti pacienta v pacientské databázi uživatele. V případě nalezení duplicity, je zobrazena chybová zpráva. V opačném případě je pro nového pacienta automaticky vygenerován univerzální identifikátor, který slouží k propojení údajů pacienta s výsledky jeho vyšetření.

Úprava údajů pacientů

Data pacientů je možno libovolně upravovat. Jedinou výjimkou je změna rodného čísla pacienta. V tomto případě aplikace provede znovu kontrolu unikátnosti rodného čísla v databázi uživatele a pokud není aplikací nalezena duplicita, umožní uložení změn.

Vymazání záznamu pacienta

Aplikace umožní odstranění záznamu jednoho nebo více vybraných pacientů. Vymazání je provedeno až po potvrzení uživatelem, a to společně se všemi údaji a výsledky testů, které se k pacientovi vztahují. Odstraněná data nelze obnovit.

Přidání výsledku testu pacienta

Ke každému pacientovi je možné zadat libovolný počet výsledků absolvovaných psychologických testů. Při přidání nového testu pacienta aplikace vygeneruje univerzální identifikátor (*test ID*) v rámci všech testů v pacientské databázi uživatele, který slouží k propojení pacienta a jeho testů. Pro přidání nového záznamu testu je třeba zadat datum, kdy byl test vykonán a jednotlivé výsledky baterie testování.

Úprava testu pacienta

Výsledky testu pacienta je možné libovolně upravit.

Vymazání testu pacienta

Aplikace umožní odstranění jednoho nebo více vybraných testů pacienta. Vymazání je provedeno až po potvrzení uživatelem.

Vizualizace testů pacienta a výstup aplikace

Aplikace vizualizuje na obrazovce zaznamenané výsledky testu pacienta v jednotlivých metrických škálách, typicky ve vztahu k distribuční funkci normální populace. Údaje o pacientovi a výsledky jeho testů může uživatel exportovat do dokumentu ve formátu PDF[1], u kterého se dá nastavit heslo pro zabezpečení a cesta pro uložení.

2.2. Struktura aplikace

Aplikace je rozdělena podle klasického schématu do dvou vrstev (bloků) a to na *frontend* a *backend*.

Frontend

Frontend je část aplikace, která se stará o interakci aplikace s uživatelem [29]. Do tohoto bloku spadá všechno, co uživatel vidí při práci s aplikací, např. styl a barva textu, ikony, obrázky, animace, ale i funkční prvky aplikace, jako např. tlačítka, zaškrtávací nebo textová pole, menu atd [29]. Do této části patří i vizualizace dat např. v podobě grafů různých typů nebo v podobě jednoduchých tabulek. V navržené aplikaci jsou pacientské údaje a výsledky testů uvedeny v souhrnných tabulkách společně s grafy spojnicové a bodového typu.

Backend

Backend je část aplikace, jejímž úkolem je zpracování, ukládání a poskytování dat [29]. Na rozdíl od frontendu není tato část viditelná, tedy nedochází k přímému kontaktu s uživatelem [29]. V aplikaci je pro práci a ukládání dat využita databáze.

2.3. Uživatelské rozhraní

Uživatelské rozhraní hraje zásadní roli při vytváření dojmu uživatele z aplikace. Proto jsou obrazovky aplikace sestaveny tak, aby jejich použití bylo pro uživatele co nejvíce intuitivní a aby ho logicky vedly při jeho práci. Při návrhu byl kladen důraz na to, aby funkční prvky (např. tlačítka, ikony, editovací pole atd.) měly dostatečnou čitelnost a jednoduchý design odpovídající současným trendům. Barevné kombinace jsou zvoleny tak aby neměly na běžnou práci uživatele

[1] Zkratka anglického názvu Portable Document Format (Přenosný formát dokumentů)

rušivý vliv. Pro upozornění uživatele na problém v aplikaci (chybové zprávy) jsou naopak využity výraznější barvy.

Jednotlivé obrazovky aplikace lze vidět v kapitole 4.Výsledná aplikace.

2.4. Volba programovacího jazyka

Pro implementaci bakalářské práce jsem si vybral programovací jazyk Python. Jedná se o vysokoúrovňový programovací jazyk, který vytvořil Guido Van Rossum v roce 1991 [30]. Python někdy bývá označován jako tzv. skriptovací jazyk, ale spadá do skupiny dynamických interpretovaných programovacích jazyků [30]. U této skupiny stačí pro spuštění programu jen jeho zdrojový kód a interpret.

Python je hybridní neboli multiparadigmatický programovací jazyk. To znamená, že při implementaci může být využito objektivně orientované paradigma, ale i procedurální anebo v určité míře funkcionální paradigma podle toho, co je v danou chvíli potřeba [31].

Python jsem si vybral z důvodu jeho jednoduchosti, univerzálnosti a širokého výběru knihoven (modulů), např. modul pro práci se soubory, modul pro tvorbu uživatelského rozhraní anebo moduly pro práci s databázemi (sqlite3, MySQL, PostgreSQL atd.). Navíc je možno využít řadu užitečných knihoven třetích stran, např. matematickou knihovnu *Numpy*, knihovnu *PyQt* pro tvorbu uživatelského rozhraní, knihovnu *Pandas* pro manipulaci a analýzu dat atd. Tyto knihovny ve spojení se standardními knihovnami vytvářejí z tohoto programovacího jazyka velmi silný implementační nástroj.

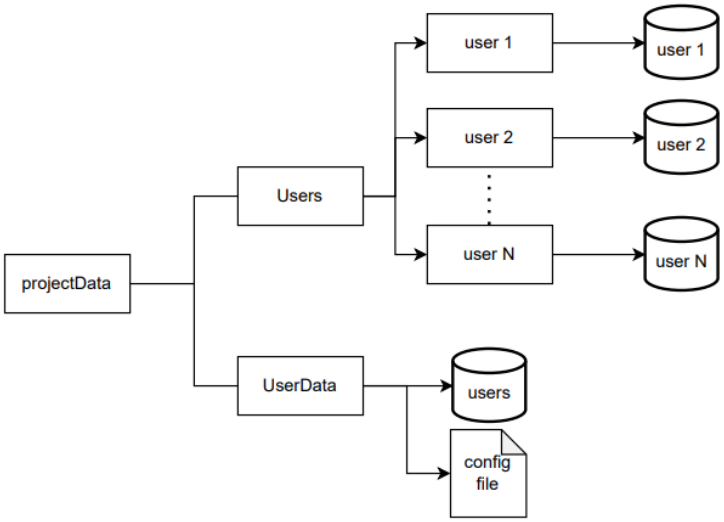
3. Implementace

3.1. Struktura adresáře aplikace

Po instalaci aplikace je vytvořena složka (adresář) v kořenovém adresáři počítače. Data, s kterými aplikace pracuje, jsou uložena v adresáři *projectData*. Tato složka je uložena v adresáři aplikace a je rozdělena na níže uvedené podadresáře (podsložky):

- Users – Adresář uživatelů obsahující složky a databáze uživatelů aplikace.
- UserData – Adresář uživatelských dat obsahující konfigurační soubor aplikace *config file* (nastavení aplikace) a uživatelskou databázi *users* (obsahuje uživatelské údaje podrobněji v kapitola 3.2.2).

Pro práci s adresáři je využit modul os.



Obrázek 2: Struktura adresáře aplikace

3.2. Databáze

3.2.1. Použité technologie

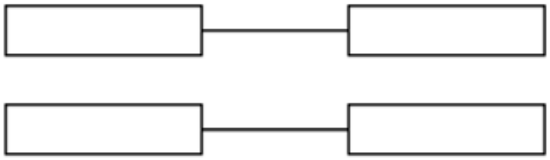
Pro databázi aplikace jsem zvolil databázový systém *SQLite*. Je to relační databázový systém, což znamená, že v databázi jsou data uchována ve dvourozměrných tabulkách (relacích) a mezi jednotlivými relacemi jsou vztahy (vazby). Databázové tabulky obsahují řádky (záznamy) a sloupce (atributy). Atributy jsou vždy určeny svým konkrétním datovým typem, který udává, jaké hodnoty lze vložit do daného sloupce. Navíc některé atributy plní funkci tzv. databázového klíče.

Rozlišujeme několik druhů databázových klíčů [32]:

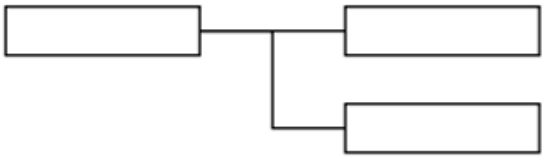
- **Primární klíč** (Primary key – Pk) označuje atribut nebo množinu atributů, které jednoznačně identifikují záznamy v databázové tabulce.
- **Kandidátní klíč** je atribut anebo množina atributů, které jednoznačně identifikují záznam v tabulce a mohou se stát primárním klíčem. Kandidátní klíč, který se nestal primárním klíče, je označován jako tzv. **alternativní klíč**
- **Cizí klíč** (Foreign key – Fk) neboli nevlastní klíč slouží pro vyjádření vztahu mezi záznamy dvou databázových tabulek (relacemi).

Vztahy mezi dvěma tabulkami databáze jsou trojího typu [33]:

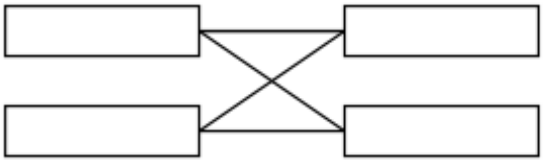
- **1:1** Jednomu záznamu z první tabulky odpovídá záznam z druhé tabulky a naopak.
- **1:N** Jednomu záznamu z první tabulky odpovídá více záznamů z druhé tabulky anebo žádné.
- **N:N** Každému záznamu z první tabulky může odpovídat více záznamů z druhé tabulky a naopak.



Obrázek 3: Vztah 1:1



Obrázek 4: Vztah 1:N



Obrázek 5: Vztah N:N

Databázový systém *SQLite* je charakterizován následujícími vlastnostmi [34]:

- Může běžet na libovolném operačním systému.
- Pro práci s touto databází není potřeba komunikace klient-server.
- Novou databázi není třeba instalovat ani konfigurovat.
- Každá databáze je umístěna ve vlastním souboru s příponou *.dbm* (Database manager).

Navíc existuje ve standartní knihovně jazyka Python modul *sqlite3*, který slouží pro komunikaci a práci s touto databází.

Díky těmto vlastnostem je *SQLite* databáze vhodná pro aplikaci řešenou v rámci této bakalářské práce.

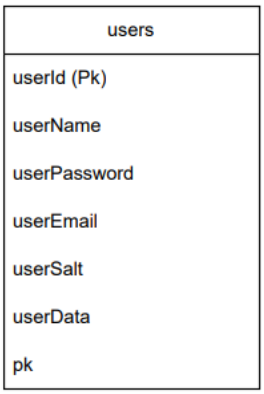
3.2.2. Databázová schémata aplikace

Aplikace pracuje s dvěma typy databází, **uživatelskou** a **pacientskou**. V rámci popisu těchto databází jsou použity pojmy *hash*, *kryptografická sůl*, *soukromý klíč* a *symetrické šifrování*, které jsou detailněji vysvětleny v kapitole 3.3.

Uživatelská databáze

Uživatelská databáze je využita v aplikaci pro ukládání a práci s uživatelskými informacemi. Obsahuje pouze jednu databázovou tabulku (relaci) s níže uvedenými atributy:

- **userId** – univerzální identifikátor (primární klíč)
- **userName** – hash uživatelského jména
- **userPassword** – hash uživatelského hesla
- **userEmail** – hash uživatelského emailu
- **userSalt** – kryptografická sůl uživatele
- **userData** – zašifrovaná uživatelská data (jméno a příjmení uživatele, klíč symetrického šifrování)
- **pk** – zašifrovaný soukromým klíčem uživatele



Obrázek 6: Relační model uživatelské databáze aplikace

Pacientská databáze

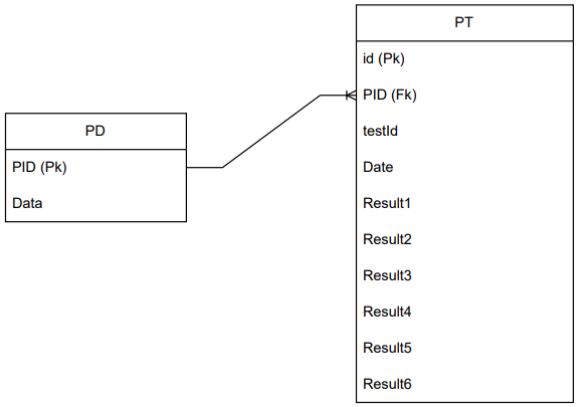
Jelikož aplikace pracuje pouze s údaji o pacientovi (jméno, příjmení, datum narození atd.) a s testy, které pacient podstoupil, má pacientská databáze jednoduché databázové schéma. Skládá se ze dvou relací, **pacientská data** (PD) a **pacientské testy** (PT), mezi kterými je vztah 1:N.

Tabulka pacientská data obsahuje následující atributy:

- **PID** – univerzální identifikátor pacienta (primární klíč)
- **Data** – šifrované údaje o pacientech (jméno, příjmení, datum narození, rodné číslo, dosažené vzdělání, pohlaví)

Tabulka s pacientskými testy obsahuje následující atributy:

- **Id** – číslo řádku v tabulce (primární klíč)
- **PID** – identifikátor pacienta, který je cizím klíčem, sloužícím jako reference na atribut PID v relaci pacientská data.
- **testId** – univerzální identifikátor testu
- **Date** – datum, kdy byl test proveden
- **Result1**, **Result2**... – výsledky testů pacienta



Obrázek 7: Relační model pacientské databáze

3.3. Šifrování

3.3.1. Použité technologie

V aplikaci je využito pro zabezpečení citlivých informací *symetrické šifrování*, *asymetrické šifrování* a *kryptografická hashovací funkce*.

Symetrické šifrování

Symetrické šifrování tvoří první vrstvu v zabezpečení pacientských údajů a je využito v aplikaci pro šifrování jednotlivých záznamů atributu *Data* v tabulce *PD* v pacientské databázi. Je to kryptografická metoda, která využívá k šifrování a dešifrování pouze jednoho klíče [35]. Výhodou tohoto způsobu šifrování je, že algoritmy pro symetrické šifrování mají nízkou výpočetní náročnost a díky tomu je rychlost šifrování a dešifrování vyšší než u algoritmů pro asymetrické šifrování [36]. Nevýhodou této metody je nutnost zabezpečení stejného klíče pro šifrování a dešifrování [36].

Ve standartní knihovně jazyka python neexistuje modul pro symetrické šifrování, a proto jsem využil knihovnu *Cryptography*. Z této knihovny jsem si vybral algoritmus *Advanced Encryption Standard* (AES). Je to rychlý a kryptograficky bezpečný algoritmus pracující s bloky dat [37]. To znamená, že vstupní zpráva je rozdělena do bloků o fixní délce, nejčastěji 128 bitů [36]. Pokud by určitý blok nebyl zcela zaplněný, vyplní se pomocí tzv. výplně (v angl. Padding), což mohou být náhodně vygenerované znaky [37].

AES algoritmus má i fixní délku klíče na 128, 192 nebo 256 bitů [37], ale pro účely této aplikace jsem využil klíč o délce 128 bitů, který je vždy uživateli vygenerován aplikací při

registraci a uložen společně se jménem a příjmením uživatele do atributu *userData* uživatelské databáze aplikace.

Asymetrické šifrování

Další vrstvou zabezpečení je asymetrické šifrování. Tato metoda je využita pro zabezpečení symetrického klíče a údajů o uživateli uložených v uživatelské databázi v atributu *userData*.

Asymetrické šifrování je kryptografický algoritmus, ve kterém se na rozdíl od algoritmů využívající symetrického šifrování, používá dvojice kryptografických klíčů, **soukromý klíč** (v angl. Private key) a **veřejný klíč** (v angl. Public key) [36]. Veřejný klíč je určený pro šifrování otevřeného textu a soukromý klíč je využit pro dešifrování zprávy, která byla zašifrována pomocí soukromého klíče. Soukromý a veřejný klíč jsou navzájem propojeny a vytváří tzv. **klíčový pár**, to znamená, že ze znalosti soukromého klíče můžeme odvodit jeho protějšek veřejný klíč, ale opačné odvození není možné [36]. Tyto vlastnosti jsou velkou výhodou, naopak nevýhodou je vyšší výpočetní náročnost [36].

Stejně jako v případě symetrického šifrování, neexistuje ve standartní knihovně jazyka Python modul pro asymetrické šifrování. Proto jsem využil znovu knihovnu *Cryptography* a z ní algoritmus RSA (iniciály autorů Rivest, Shamir, Adleman).

Každý uživatel, stejně jako v případě symetrického šifrování, má svůj soukromý klíč, který je generován při registraci. Knihovna *Cryptography* umožňuje RSA soukromý klíč kódovat do různých formátů např. PEM, DER, X962 atd. [38]. Tyto formáty mohou být i zašifrovány pomocí hesla. Této možnosti jsem využil ve své aplikaci, kdy každý uživatel má svůj soukromý klíč uložený v podobě PEM formátu v atributu *pk* v tabulce *users* uživatelské databáze. Tento klíč je zabezpečený heslem uživatele.

Kryptografická hashovací funkce

Hashování je poslední vrstvou zabezpečení aplikace. Tato metoda je využita pro autentizaci uživatele a pro zabezpečení soukromého klíče uživatele.

Kryptografická hashovací funkce je v kryptografii matematická funkce, která převádí vstupní text na posloupnost bytů fixní délky, která se označuje pod pojmem hash (někdy se taky označuje jako haš, výtah, otisk atd.) [39]. Tento výstup je pro zadaný vstup jednoznačný, to znamená, že z výstupu hashovací funkce není možné odvodit vstup [39]. Díky této vlastnosti se označuje hashovací funkce jako tzv. **jednocestná** nebo taky **jednosměrná** funkce [40].

Hashovací funkce splňuje další vlastnosti [39]:

- Při malé změně textu na vstupu, dojde k velké změně na výstupu. Toto se označuje jako tzv. lavinový efekt.
- Je velice nepravděpodobné najít dva různé vstupy, kterým bude odpovídat stejný hash, tedy nalezení kolizí.

V kryptografické hashovací funkci se navíc ještě využívá tzv. kryptografická sůl nebo jenom sůl. Je to řetězec náhodných bitů, které jsou doplněny do vstupního textu a poté je aplikována hashovací funkce [41]. Toto doplnění zajistí, aby měla hashovací funkce pro daný vstup více možných výstupních variant [41].

Pro vytváření a práci s hashem jsem použil knihovnu *bcrypt*, která je implementací hashovací funkce *bcrypt* založené na symetrické blokové šifře *Blowfish* [42].

3.4. Registrace uživatele

V této kapitolu je popsána implementace registrace nového uživatele v aplikaci.

1. Zadaní uživatelských údajů

Pro registraci nového uživatele je potřeba zadat následující údaje:

- Jméno a příjmení uživatele
- Uživatelské jméno
- Email uživatele
- Uživatelské heslo

2. Kontrola unikátnosti uživatelského jména a emailu

Aplikace nejprve provede kontrolu pomocí funkce *checkpw* knihovny *bcrypt*, zda v uživatelské databázi už neexistuje uživatel se stejným uživatelským jménem. Tuto samou kontrolu provede znova se zadaným emailem nového uživatele. Pokud je uživatelské jméno nebo email nalezeno v uživatelské databázi, je zobrazena chybová zpráva na registrační obrazovku aplikace. V opačném případě je z uživatelského jména, emailu a hesla vytvořen hash pomocí knihovny *bcrypt*.

3. Generování univerzálního identifikátoru uživatele

Po kontrole zadaného uživatelského jména a emailu je uživateli vytvořen univerzální identifikátor pomocí knihovny *shortuuid*. Tento identifikátor je využit jako primární klíč v uživatelské databázi aplikace.

4. Vytvoření pacientské databáze uživatele

Po vygenerování identifikátoru uživatele je vytvořena uživatelská složka a databázový soubor (pacientská databáze viz. kapitola 3.2.2), ve kterém jsou později uloženy záznamy pacientů a výsledky jejich testů. Identifikátor slouží jako pojmenování složky a databázového souboru uživatele.

5. Vytvoření šifrovacích klíčů a kryptografické soli

Po vytvoření pacientské databáze je následně vygenerována dvojice klíčů a kryptografická sůl:

- Klíč symetrického šifrování – vygenerovaný pomocí metody *generate_key* třídy *Fernet* knihovny *Cryptography*
- Soukromý klíč asymetrického šifrování – vygenerovaný pomocí metody *generate_private_key* třídy RSA knihovny *cryptography*
- Kryptografická sůl – vygenerována pomocí funkce *genSalt* knihovny *bcrypt*

Symetrický klíč je společně se jménem a příjmením uživatele zašifrováno pomocí veřejného klíče, který je vytvořený odvozením z vygenerovaného soukromého klíče uživatele.

Pomocí funkce *hashpw* knihovny *bcrypt* je vytvořen hash z uživatelského hesla a vygenerované kryptografické soli. Soukromý klíč je převeden do PEM formátu a zabezpečen pomocí tohoto vytvořeného hashe.

6. Vytvoření zálohy uživatelského hesla a jeho zabezpečení

Pro vytvoření zálohy uživatelského hesla (slouží pro obnovu uživatelského hesla viz. kapitola 3.6) je využita knihovna *keychain*, která umožňuje ukládat přihlašovací údaje do tzv. credential locker (operační systém Windows), Keychain (v operační macOs) anebo KWallet (Linux). Uživatelské heslo je zabezpečeno pomocí symetrického šifrování. Klíč symetrického šifrování je vytvořen pomocí třídy *Fernet* knihovny *Cryptography*, která umožňuje vygenerovat klíč symetrického šifrování z hesla za pomoci odvozovací funkce (v angl. Key derivation function – KDF). Odvozovací funkce je kryptografická funkce, která za pomoci hashovací funkce vytváří jeden nebo více šifrovacích klíčů z hesla, hlavního klíče atd. [43].

Heslo, ze kterého je vygenerován klíč symetrického šifrování, je vytvořeno z kombinace uživatelského jména a kryptografické soli uživatele.

7. Uložení do databáze

V závěru registrace uživatele jsou jednotlivé hashe (uživatelského jména, uživatelského emailu, uživatelského hesla), šifrované údaje (jméno a příjmení uživatele, klíč symetrického šifrování), soukromý klíč a kryptografická sůl uživatele uloženy jako záznam do uživatelské databáze aplikace.

3.5. Přihlášení uživatele

V této kapitolu je popsána implementace přihlášení uživatele v aplikaci.

1. Zadání přihlašovacích údajů a kontrola

Při přihlášení uživatel musí zadat:

- Uživatelské jméno
- Uživatelské heslo

Aplikace využije funkci *checkpw* knihovny *bcrypt* pro kontrolu správnosti zadaných údajů a existence uživatele v uživatelské databázi aplikace. Pokud jsou zadané přihlašovací údaje nesprávné je zobrazena chybová zpráva na přihlašovací obrazovku aplikace.

2. Dešifrování šifrovacích klíčů

Po úspěšné kontrole přihlašovacích údajů uživatele je načten záznam uživatele z uživatelské databáze. Z tohoto záznamu jsou vybrány atributy

- pk
- userData
- userSalt

Dále je pomocí funkce *hashpw* knihovny *bcrypt* vytvořen hash z uživatelského hesla a načtené kryptografické soli. Tento hash je využit pro dešifrování soukromého klíče uživatele. Získaný soukromý klíč je využit pro dešifrování uživatelských údajů (obsaženy v atributu *userData*). Po dešifrování uživatelských údajů je získáno jméno a příjmení uživatele a klíč symetrického šifrování pro dešifrování údajů pacientů v pacientské databázi uživatele.

3. Načtení pacientské databáze uživatele

V závěru přihlášení uživatele je načtena uživatelská složka a z ní pacientská databáze uživatele.

3.6. Obnova hesla

V této kapitolu je popsána implementace obnovy hesla uživatele v aplikaci.

1. Zadání uživatelských údajů a kontrola

Při obnově uživatelského hesla musí uživatel zadat:

- Uživatelské jméno
- Email

Aplikace pomocí funkce *checkpw* knihovny *bcrypt* zkontroluje, zda existuje uživatel s daným uživatelským jménem a emailem v uživatelské databázi aplikace. Pokud není uživatel nalezen v uživatelské databázi, je zobrazena chybová zpráva na obrazovku aplikace. V opačném případě aplikace zašle náhodně vygenerovaný ověřovací kód obsahující 12 číslic na zadaný email uživatele.

2. Zadání ověřovacího kódu a kontrola

Po uživateli je vyžadováno zadání prvního pětičíslí ze zaslaného ověřovacího kódu. Po zadání je aplikací provedena kontrola zadaných číslic a v případě chybného zadání je zobrazena chybová zpráva. V opačném případě je uživatel vyzván aplikací k zadání nového hesla.

3. Zpracování nového uživatelského hesla

Nejprve je pomocí knihovny *keychain* načteno šifrované staré heslo uživatele. Toto staré heslo je šifrováno pomocí symetrického šifrování. Klíč pro dešifrování je vytvořen pomocí třídy *Fernet* knihovny *Cryptography* z hashe uživatelského jména a kryptografické soli uživatele. Dešifrované staré heslo je využito pro dešifrování soukromého klíče, který je pak následně zašifrován pomocí nového hesla.

Dále aplikace provede záměnu starého uživatelského hesla v *credential locker* pomocí knihovny *keychain* za nové uživatelské heslo, které je zabezpečeno symetrického šifrování (stejně jako v kapitole 3.4 bod 6.).

Nakonec je z nového uživatelského hesla vytvořen hash pomocí funkce *hashpw* knihovny *bcrypt*, který je společně s nově zabezpečeným soukromým klíčem uživatele uložen do uživatelské databáze. Po uložení je změněna obrazovka aplikace na přihlašovací obrazovku a zobrazena zpráva o úspěšné změně hesla.

3.7. Zadání pacientských údajů

V této kapitole je popsáno implementace přidání nového záznamu pacienta do pacientské databáze uživatele.

1. Zadání pacientských údajů

Při přidávání nové pacienta do pacientské databáze je po uživateli požadováno zadat níže uvedené pacientské údaje:

- Jméno a příjmení
- Datum narození
- Rodné číslo
- Dosažené vzdělání
- Pohlaví

2. Kontrola platnost a unikátnost rodného čísla

Aplikace provede nejprve kontrolu platnosti zadaného rodného čísla. U rodného čísla je kontrolují následující vlastnosti:

- Musí obsahovat 9 nebo 10 číslic.
- Je dělitelné 11 beze zbytku.
- První šestičíslí obsahuje datum narození (ve formátu RRMMDD neboli rok-měsíc-den) a musí být shodné se zadaným datem narození pacienta. V některých případech je k měsíci narození přičteno číslo 20.
- U žen je k měsíci narození přičteno číslo 50.
- Zbytek po dělení 11 prvních devíti číslic desetimístného rodného čísla musí odpovídat poslední číslici rodného čísla, tzv. kontrolní číslici.

Pokud by zadané rodné číslo nesplňovalo alespoň jedno z výše uvedených vlastností, je zobrazena chybová zpráva na obrazovku aplikace.

Následně je kontrolováno aplikací unikátnost rodného čísla v pacientské databázi uživatele. V případě nalezení aplikací zadaného rodného čísla v databázi pacientů, je zobrazena chybová zpráva na obrazovku aplikace.

3. Generování univerzálního identifikátoru pacienta

Po úspěšných kontrolách je vygenerován unikátní identifikátor pomocí knihovny *uuid*, který slouží jako univerzální identifikátor pacienta v pacientské databázi a k propojení pacienta s výsledky jeho testů uloženy v tabulce *PT* (viz kapitola 3.2.2).

4. Uložení do pacientské databáze uživatele

Nakonec jsou údaje pacienta zašifrovány symetrickým šifrováním pomocí třídy *Fernet* knihovny *Cryptography* a společně s univerzálním identifikátorem pacienta uloženy do pacientské databáze uživatele.

3.8. Grafické uživatelské rozhraní

3.8.1. Použité technologie

Pro programování grafického rozhraní jsem využil knihovnu *PyQt5*. Je to knihovna třetí strany firmy Riverbank Computing [44]. Tato knihovna propojuje jazyk Python a multiplatformní knihovnu *Qt* napsanou v jazyku C++, která umožňuje vývoj grafického rozhraní pro mobilní a desktopové aplikace [44]. Pomocí této knihovny je možno vytvářet aplikace s komplexním uživatelským rozhraním. Obsahuje mnoho užitečných nástrojů pro vývoj např. nástroje pro práci s SQL databázemi, nástroje pro práci s multimédii, síťovou nebo Bluetooth komunikaci a další [44]. Navíc je možno si stáhnout *Qt* designer, pomocí kterého lze vytvořit jednoduše uživatelské rozhraní a následně vytvořené uživatelské rozhraní převést do kódu jazyka Python.

3.9. Kalkulátor

Kalkulátor aplikace je postavený na dvou knihovnách, na knihovně *Numpy* a *Scipy*. Jsou to open source knihovny umožňující vědecké a technické výpočty.

Pomocí knihovny *Numpy* a *Scipy* je aplikací po přihlášení uživatele dopředu přepočítány vektory hodnot pro jednotlivé metrické škály (Gaussovo rozdělení, percentil, z-skóre a t-skóre). Tyto vektory jsou následně využity při vytváření grafů umístěných na obrazovce aplikace.

3.10. Vizualizace výsledků

Pro vizualizaci výsledků testů daného pacienta je využita knihovna *Matplotlib*. Je to vykreslující knihovna pro jazyk Python, která umožňuje vykreslení základních typů grafů (např. 1D, 2D a 3D), ale i vizualizaci vědeckých výpočtů.

3.11. Výstup do PDF

Aplikace umožňuje vytvoření reportu o pacientovi. Pro vytváření PDF reportů jsem využil knihovnu *FPDF*, která navíc podporuje vytváření PDF souboru zabezpečeného pomocí hesla.

V tomto reportu je obsaženo:

- Jméno a příjmení pacienta
- Datum narození
- Rodné číslo
- Vzdělání
- Pohlaví
- Dvourozměrná tabulka obsahující výsledky jednotlivých testů daného pacienta

3.12. Distribuce aplikace

V této kapitole je popsáno postup, kterým je vytvořena distribuovatelná verze aplikace.

Vytvoření spustitelného souboru

Pro vytvoření spustitelného souboru aplikace jsem využil knihovnu *PyInstaller*. Tato knihovna umožňuje vytvořit spustitelný soubor, kdy nalezne veškeré moduly, knihovny a přídavné soubory (ikony atd.) využité v aplikaci a sloučí je dohromady.

Spustitelný soubor může být vytvořený pro libovolný operační systém (Windows, macOS, Linux atd.), ale musí být vytvořený na daném operačním systému. To znamená, že nemůže být vytvořený touto knihovnou spustitelný soubor na operačním systému Windows, který bude využíván na jiném operačním systému např. Linux.

Výsledný spustitelný soubor může být dvojího formátu:

- **OneFile** – Všechny knihovny, moduly a soubory aplikace jsou sloučeny v jediný spustitelný soubor. Po spuštění výsledného spustitelného souboru jsou všechny obsažené soubory zkopírovány do tzv. dočasné složky (v angl. Temporary folder), kde je následně aplikace spuštěna. Po ukončení aplikace je dočasná složka odstraněna.
- **OneDir** – Veškeré knihovny, moduly a soubory použité v aplikaci jsou sloučeny do jediné složky, která obsahuje i spustitelný soubor. Po spuštění spustitelného souboru nahraje soubory obsažené ve složce a spustí aplikaci. Díky tomu, že není potřeba vytvářet dočasné složky, je spuštění aplikace tímto způsobem rychlejší než v případě jediného souboru. Ale přináší to nevýhodu distribuce, kdy je potřeba distribuovat více souborů na jednou.

Pro aplikaci této bakalářské práce jsem zvolil formát OneDir, jelikož mi tento formát umožňuje snadnější debuggování a rychlejší spuštění aplikace. Operační systém, pro který je aplikace této bakalářské práce určena, je Windows.

Vytvoření instalačního programu

Jak bylo zmíněno výše, nevýhodou formátu OneDir je, že obsahuje všechny soubory (knihovny, moduly atd.) využité aplikací v jediné složce a díky tomu je tento formát nepraktický pro distribuci než v případě formátu OneFile. Proto jsem využil program *Inno setup*, který umožňuje vytvořit instalační program pro operační systém Windows, který automaticky umístí veškeré soubory aplikace (knihovny, moduly, ikony atd.) do hlavního adresáře počítače a vytvoří zástupce aplikace. Navíc v případě odinstalování aplikace, jsou automaticky odstraněny všechny soubory aplikace.

4. Výsledná aplikace

V následujících kapitolách je popsána a vysvětlena práce s aplikací na jednotlivých obrazovkách aplikace.

4.1. Registrace a přihlášení

4.1.1. Registrace nového uživatele

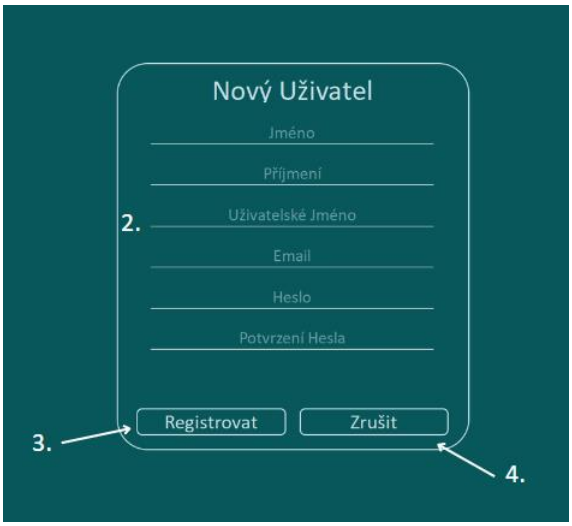
Po stisknutí tlačítka *Nový uživatel* (bod 1. Obrázek 8) je zobrazena registrační stránka aplikace (Obrázek 9). Zde je potřeba vyplnit následující údaje:

- Jméno a příjmení
- Uživatelské jméno
- Email – slouží pro odeslání ověřovacího kódu
- Heslo uživatele
- Potvrzení hesla

Následně je potřeba registraci potvrdit stisknutím tlačítka *Registrovat* (bod 3. Obrázek 9). Aplikací je provedena kontrola, zda zadané heslo a heslo pro potvrzení jsou totožné, v případě že by si neodpovídali, zobrazí se chybová zpráva. Následně je aplikací zkontrolováno, zda vložené uživatelské jméno nebo email už nejsou zaregistrované. Pokud by uživatelské jméno nebo email bylo už zaregistrované, zobrazí se chybová zpráva. V opačném případě je nový uživatel aplikací zaregistrován a změní se obrazovka aplikace na obrazovku pro přihlášení (Obrázek 10), kde je zobrazena zpráva o úspěšném přihlášení (Obrázek 10).

Pro návrat na přihlašovací obrazovku aplikace slouží tlačítko *Zrušit* (bod 4.Obrázek 9).

Obrázek 8: Přihlašovací obrazovka při registraci nové uživatele



Obrázek 9:Obrazovka pro registraci nového uživatele

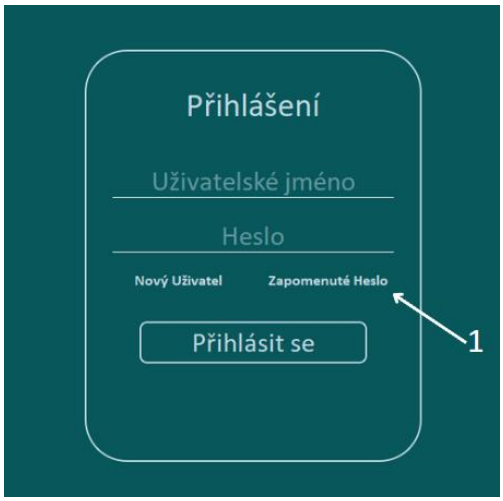
Obrázek 10: Přihlašovací obrazovka po úspěšném přihlášení uživatele

4.1.2. Přihlášení uživatele

Pro přihlášení je potřeba zadat níže uvedené údaje:

- Uživatelské jméno
- Heslo

Nakonec potvrdit přihlášení kliknutím na tlačítko *Přihlásit se* (bod 1. Obrázek 11). Aplikací je provedena kontrola zadaných přihlašovacích údajů. V případě nesprávných přihlašovacích údajů je zobrazena chybová zpráva (Obrázek 12).



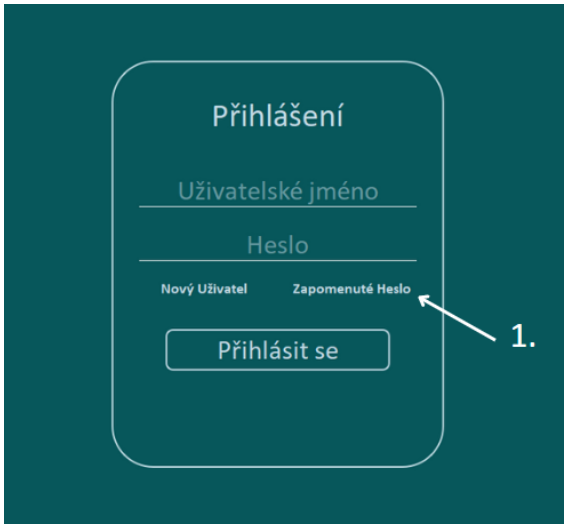
Obrázek 11: Obrazovka pro přihlášení

Obrázek 12: Přihlašovací obrazovka při zadání nesprávných přihlašovacích údajů

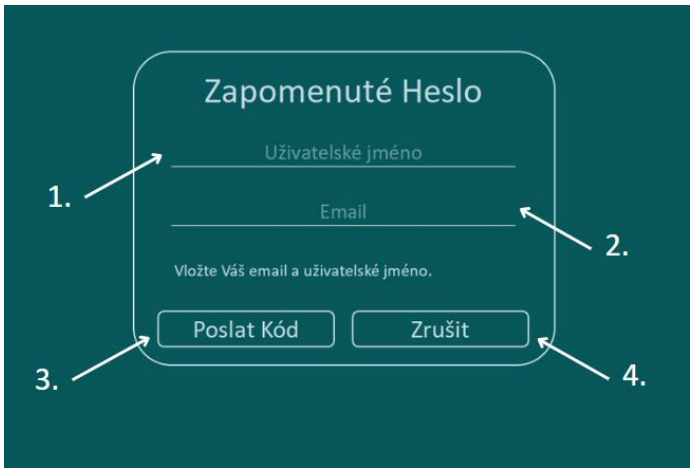
4.1.3. Obnova zapomenutého hesla

Pro obnovu hesla je zapotřebí stisknout tlačítko *Zapomenuté heslo* na přihlašovací obrazovce aplikace (bod 1. na Obrázek 13). Po stisknutí je zobrazena obrazovka pro ověření totožnosti (Obrázek 14). Zde je potřeba zadat uživatelské jméno (bod 2. Obrázek 14), email (bod 3. Obrázek 14) a potvrdit vložené údaje stisknutím tlačítka *Poslat kód* (bod 3. Obrázek 14). Aplikací je provedena kontrola zadaných uživatelských údajů. V případě nesprávnosti zadaných uživatelských údajů je zobrazena chybová zpráva. V opačném případě je aplikací zaslán ověřovací kód na zadaný email a obrazovka aplikace je změněna na obrazovku pro ověření kódu (Obrázek 15).

Tlačítko *Zrušit* na obrazovce pro zadání uživatelského jména a emailu (bod 4. Obrázek 14) slouží pro vrácení na přihlašovací stránku aplikace.



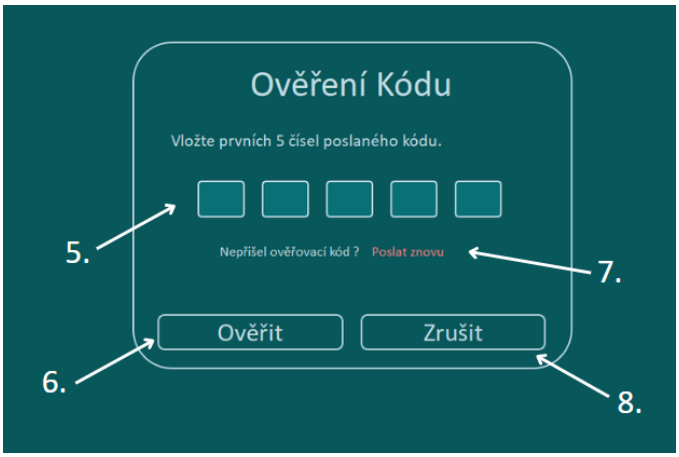
Obrázek 13: Přihlašovací obrazovka – obnova hesla



Obrázek 14: Obrazovka pro zaslání ověřovacího kódu

Ze zaslaného ověřovacího kódu je nutno vybrat první pětičíslí, které je potřeba vložit do oken (bod 5. Obrázek 15) připravených na stránce pro ověření kódu aplikace. V případě, že by ověřovací kód nebyl zaslán na zadaný email, je možno si nechal zaslat ověřovací kód znovu pomocí stisknutí tlačítka *Poslat znovu* (bod 7. Obrázek 15). Nakonec je potřeba zadaný ověřovací kód potvrdit kliknutím na tlačítko *Ověřit* (bod 6. Obrázek 15). Aplikací je následně provedena kontrola zadaného ověřovacího kódu. V případě chybného zadání ověřovacího kódu je zobrazena chybová zpráva na obrazovku aplikace. V opačném případě je změněna obrazovka aplikace na obrazovku pro zadání nového přihlašovacího hesla (Obrázek 16).

Tlačítko *Zrušit* (bod 4. Obrázek 16) slouží pro návrat na přihlašovací stránku aplikace.



Obrázek 15: Obrazovka pro ověření zaslaného ověřovacího kódu

Pro změnu přihlašovacího hesla je potřeba zadat nové heslo a nové heslo pro potvrzení a potvrdit kliknutím na tlačítko *Změnit heslo* (bod 9. Obrázek 16).

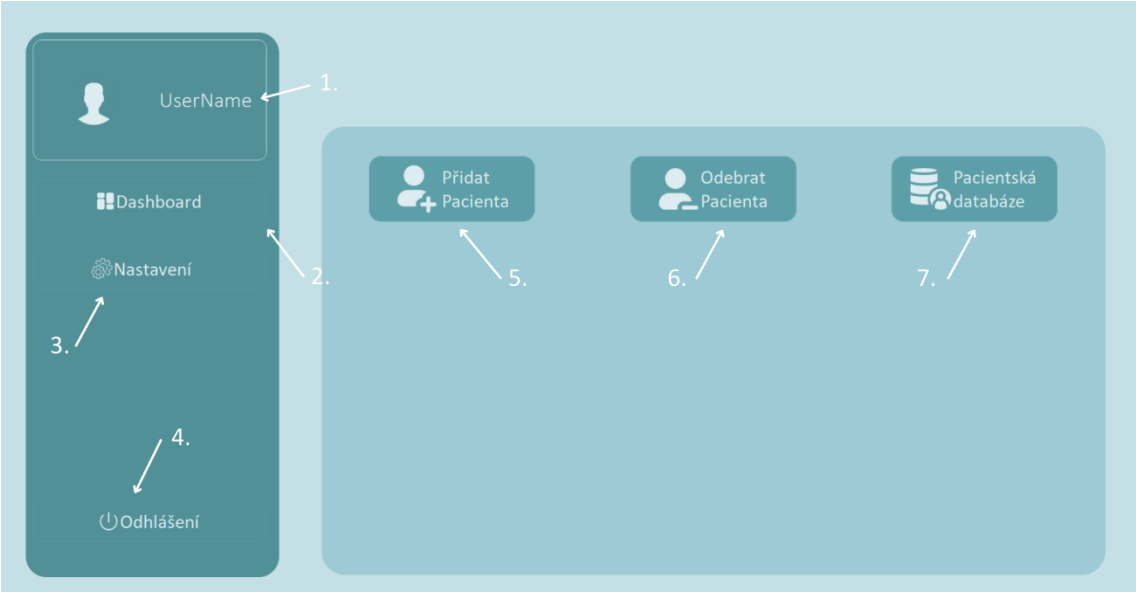
Tlačítko *Zrušit* (bod. 10 Obrázek 16) slouží pro návrat zpět na obrazovku pro zaslání ověřovacího kódu aplikace (Obrázek 14).

Obrázek 16: Obrazovka pro vytvoření nového hesla

4.2. Nástěnka (Dashboard)

Dashboard je část aplikace, která je zobrazena při přihlášení uživatele do aplikace. Tato stránka aplikace slouží jako hlavní rozcestník aplikace a obsahuje následující části:

1. UserName – Zde je zobrazeno uživatelské jméno uživatele.
2. Dashboard – Pro obnovení hlavní stránky aplikace.
3. Nastavení – Slouží pro přechod na stránku nastavení aplikace, kde je možno nastavit jazyk aplikace.
4. Odhlášení –Pro odhlášení uživatele z aplikace.
5. Přidat pacienta – Po kliknutí je zobrazena pacientská stránka a dialogové okno pro přidání nového pacienta.
6. Odebrat pacienta – Po kliknutí je zobrazena pacientská stránka pro odebrání pacienta nebo pacientů.
7. Pacientská databáze – Obrazovka aplikace je po klinutí změněna na pacientskou stránku.



Obrázek 17: Obrazovka dashboardu aplikace

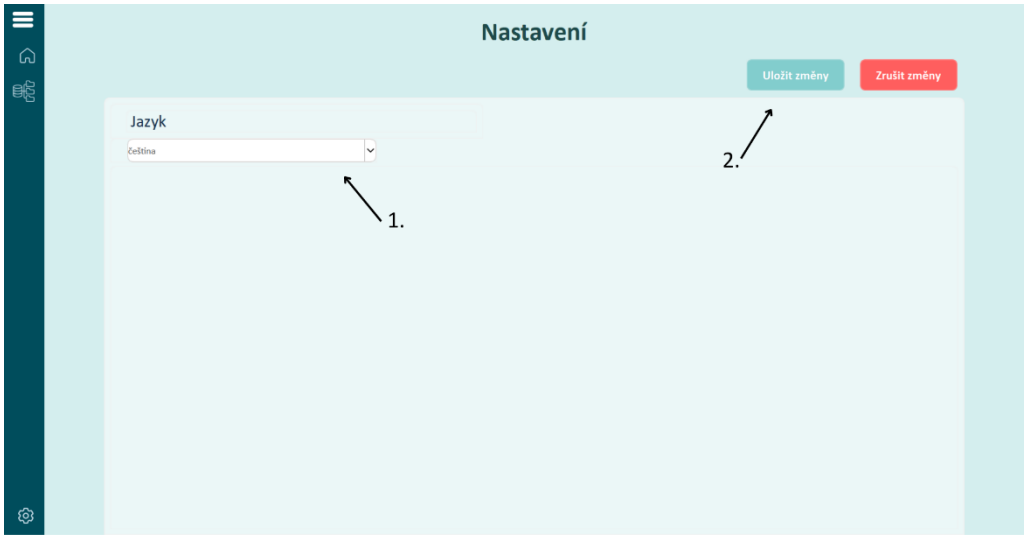
4.3. Nastavení

4.3.1. Nastavení jazyka

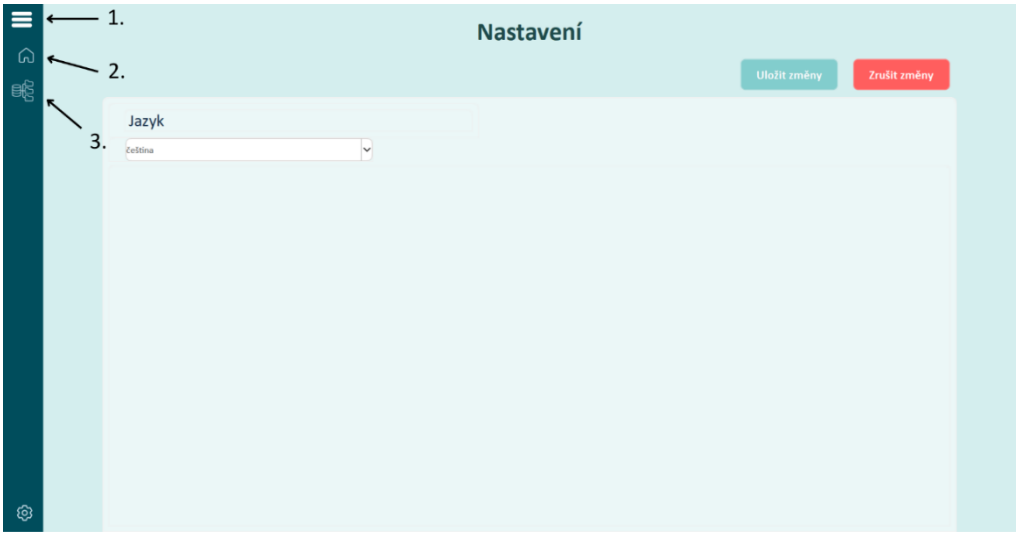
Pro nastavení jazyku aplikace je potřeba vybrat jazyk pomocí výběrového pole (bod 1.Obrázek 18). Následně uložit změny stisknutím tlačítka Uložit změny (bod 2. Obrázek 18).

Pro přechod zpět na hlavní stránku aplikace slouží tlačítko s ikonou domu (bod 2.Obrázek 19). a pro přechod na stránku s údaji pacientů slouží tlačítko s ikonou databáze (bod 3.Obrázek 19).

Postranní panel je možno rozšířit pomocí tlačítka s ikonou tří vodorovných čar (bod 1. Obrázek 19).



Obrázek 18: Nastavení jazyku aplikace



Obrázek 19: Návrat ze stránky nastavení

4.4. Pacientská stránka

Pacientská stránka zobrazuje pacienty uloženy v pacientské databázi.

4.4.1. Přidání nového pacienta

Pro přidání nového pacienta je potřeba kliknout na tlačítko *Přidat* na pacientské stránce aplikace (bod 1. Obrázek 20). Následně se zobrazí dialogové okno pro přidání nového pacienta (Obrázek 21). Zde je potřeba zadat následující údaje o pacientovi:

- Jméno
- Příjmení
- Datum narození – lze zadat i za pomoci kalendáře, který se zobrazí po kliknutí na ikonu kalendáře
- Rodné číslo
- Pohlaví pacienta
- Vzdělání – dosažené vzdělání pacienta

Po vyplnění je potřeba potvrdit kliknutím na tlačítko Přidat (bod 2. Obrázek 21). Dialogové okno pro přidání nového pacienta lze zavřít pomocí tlačítka *Zrušit* (bod 3. Obrázek 21)

Obrázek 20: Pacientská databáze – přidání pacienta

Obrázek 21: Dialogové okno pro přidání nového pacienta

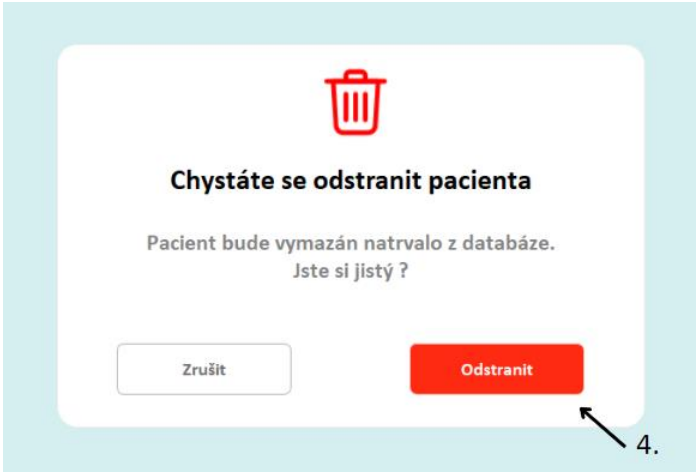
4.4.2. Odstranění záznamu pacienta

Odstranit záznamu pacienta z pacientské databáze uživatele je možno dvěma způsoby.

Prvním způsobem odstranění pacienta je pomoci tlačítka *Odstranit* (bod 1. Obrázek 22) na pacientské stránce aplikace. Po kliknutí je změna nástrojová lišta a na levé straně se zobrazí tlačítka pro výběr. Pro odstranění pacienta nebo pacientů je zapotřebí vybrat pacienta nebo pacienty (bod 2. Obrázek 23), kliknout na tlačítko *Odstranit* (bod 3. Obrázek 23) a potvrdit vymazání kliknutím na tlačítko *Odstranit* v dialogovém okně (bod 4. Obrázek 24).

Obrázek 22: Databázová stránka – odstranění pacientů pomocí tlačítka „Odstranit“

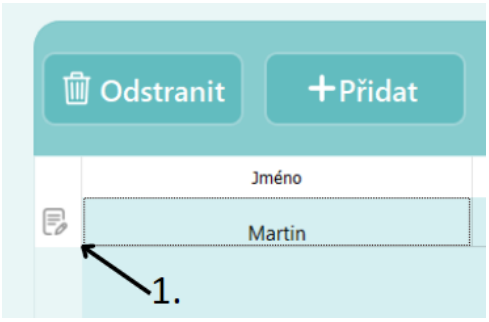
Obrázek 23: Databázová stránka pacientů – odstranění pacientů



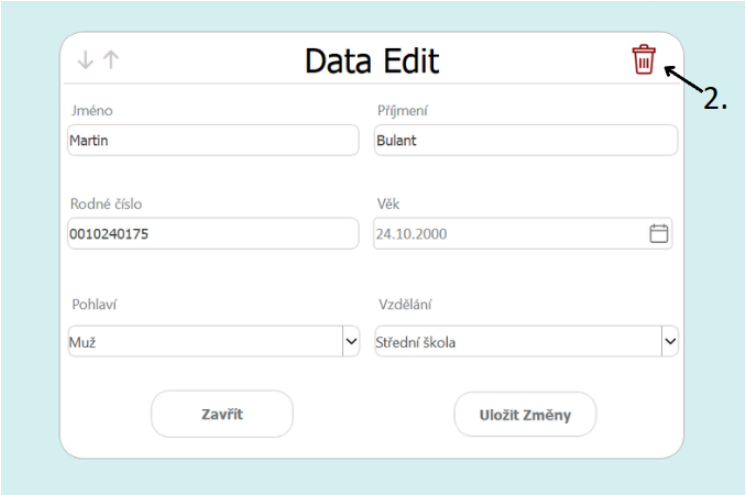
Obrázek 24: Dialogové okno při odstranění pacienta

Druhým způsobem vymazání záznamu pacienta z pacientské databáze je pomocí editovacího okna.

Pro odstranění je potřeba kliknout na ikonu editování (bod 1. Obrázek 25) nacházející se v levé postranní liště. Následně je zobrazeno dialogové okno pro úpravu údajů pacienta. Na tomto okně je potřeba klinout na tlačítko s ikonou odpadkového koše (bod 2. Obrázek 26) a potvrdit odstranění kliknutím na tlačítko Odstranit v dialogovém okně (bod 4. Obrázek 24).



Obrázek 25: Odstranění pacienta pomocí editovacího okna

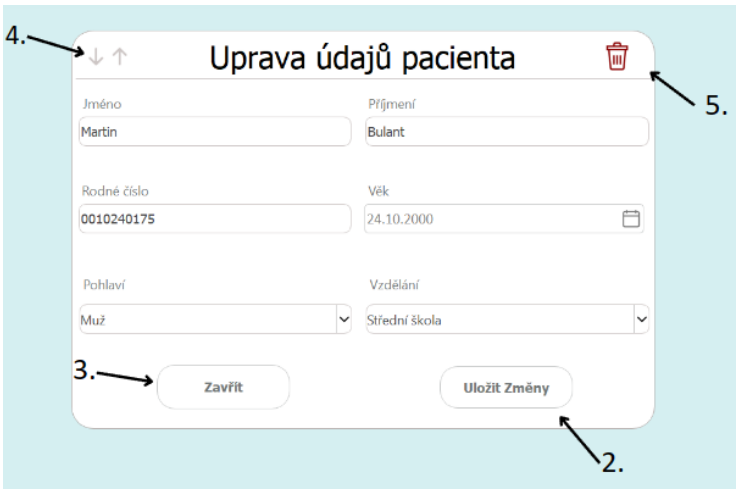


Obrázek 26: Editovací dialogový okno pacienta – vymazání pacienta

4.4.3. Úprava údaje pacienta

Po kliknutí na tlačítko s ikonou editování (bod 1. Obrázek 25) nacházející se v levé postranní liště, je zobrazeno dialogové okno pro úpravu údajů daného pacienta. Zde je možno upravit jednotlivé údaje libovolně až na rodné číslo, kde se aplikací kontroluje unikátnost. Po skončení úprav údajů pacienta je potřeba změny uložit kliknutím na tlačítko *Uložit změny* (bod 2. Obrázek 27).

Pomocí dialogového okna pro úpravu údajů pacienta je možno daného pacienta vymazat z pacientské databáze kliknutím na tlačítko s ikonou odpadkového koše (bod 5. Obrázek 27) umístěné v pravém horním rohu dialogového okna. Údaje pacientů lze procházet pomocí dvojice šipek umístěných v levým horním rohu dialogového okna (bod 4. Obrázek 27). Dialogové okno je možno zavřít pomocí tlačítka *Zavřít* (bod 3. Obrázek 27).



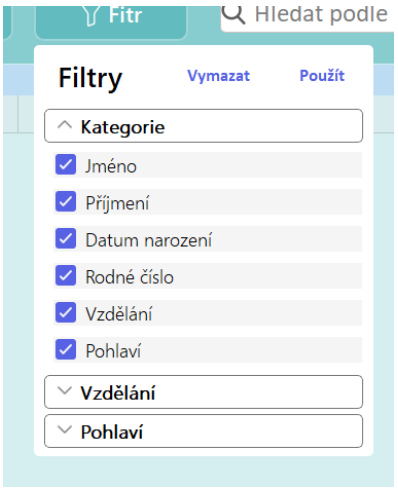
Obrázek 27: Dialogové okno pro úpravu údajů pacientů

4.4.4. Filtrace pacientů

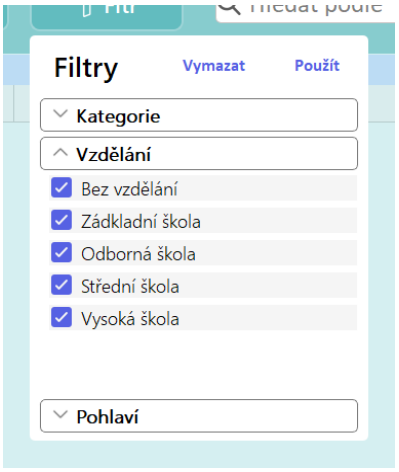
Pro zobrazení pacientského filtru je potřeba kliknout na tlačítko *Filtr* (bod 1. Obrázek 28). Po kliknutí je zobrazen pacientských filtr, pomocí kterého lze nastavit zobrazení pacientů na pacientské stránce aplikace. Pacienty lze filtrovat podle typu dosaženého vzdělání (Obrázek 30) a pohlaví (Obrázek 31). Pomocí pacientského filtru lze i nastavit, které údaje (jméno, příjmení, rodné číslo atd.) pacientů budou zobrazeny aplikací na pacientské stránce aplikace.

Pro aplikaci nastaveného filtru je potřeba kliknout na tlačítko *Použít* (bod 2. Obrázek 32) a pro odstranění filtru je potřeba kliknout na tlačítko *Vymazat* (bod 3. Obrázek 32).

Obrázek 28: Ukázka otevření pacientského filtru

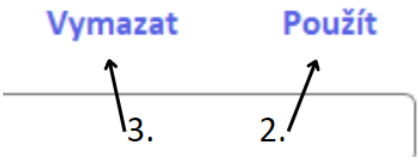


Obrázek 29: Pacientských filtr – výběr údajů pacientů, které budou zobrazeny.



Obrázek 30:Pacientský filtr – nastavení úrovně vzdělání

Obrázek 31: Pacientský filtr – nastavení pohlaví



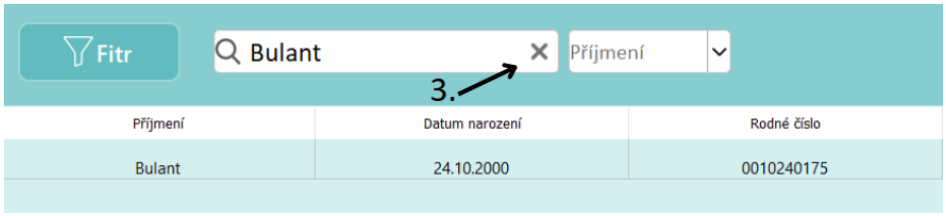
Obrázek 32: Tlačítko pro aplikaci a odstranění nastaveného pacientského filtru

4.4.5. Vyhledání záznamu pacienta

Pro vyhledání daného pacienta je potřeba vybrat kategorii vyhledávání (jméno, příjmení nebo datum narození) pomocí výběrového pole (bod 1 Obrázek 33). Následně zadat výraz pro vyhledání (jméno, příjmení nebo datum narození pacienta) do vstupního pole (bod 2. Obrázek 33) a potvrdit vyhledání stisknutím klávesy Enter na klávesnici.

Pro zrušení vyhledání je potřeba kliknout na tlačítko s ikonou X nacházející se v pravé části vyhledávacího pole (bod 3. Obrázek 34).

Obrázek 33: Výběr a zadání parametrů vyhledávání.



Obrázek 34: Zrušení vyhledávání

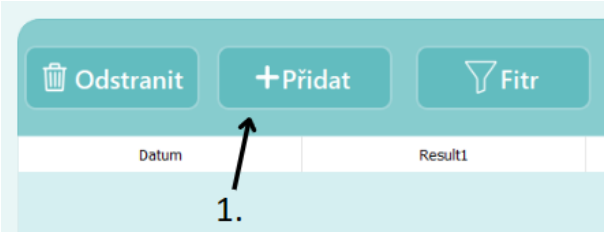
4.5. Stránka testů pacienta

Stránka testů pacienta zobrazuje testy daného pacienta.

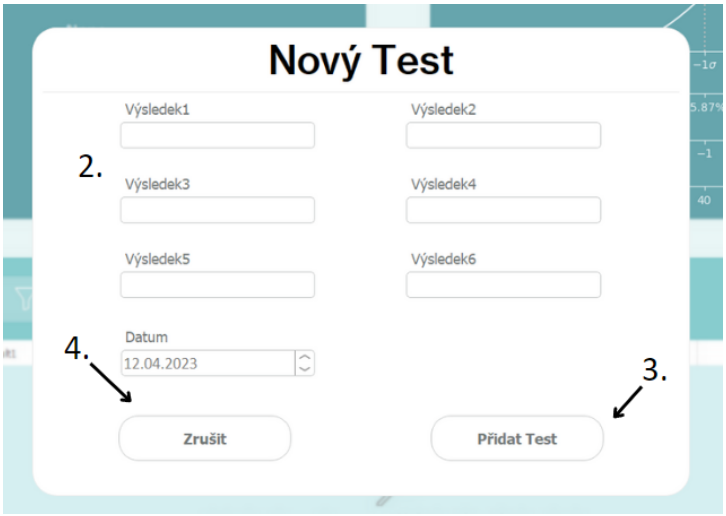
4.5.1. Přidání nové testu pacienta

Pro přidání nového testu pacienta je potřeba kliknout na tlačítko *Přidat* (bod 1. Obrázek 35) na pacientské stránce aplikace. Po kliknutí je zobrazeno dialogové okno pro přidání nového testu (Obrázek 36). Zde je potřeba zadat jednotlivé výsledky testu pacienta a vybrat datum, kdy byl test vykonán (bod 2. Obrázek 36). Nakonec pro uložení testu je potřeba kliknout na tlačítko *Přidat Test* (bod 3. Obrázek 36). Po kliknutí na tlačítko *Přidat Test* je dialogové okno uzavřeno, test pacienta je aplikací uložen a zobrazen na stránce pacientských testů.

Dialogové okno je možno zavřít kliknutím na tlačítko *Zrušit* (bod 4. Obrázek 36).



Obrázek 35:Zobrazení dialogového okna pro přidání nového testu pacienta.



Obrázek 36: Dialogové okno pro připadání nového testu pacienta.

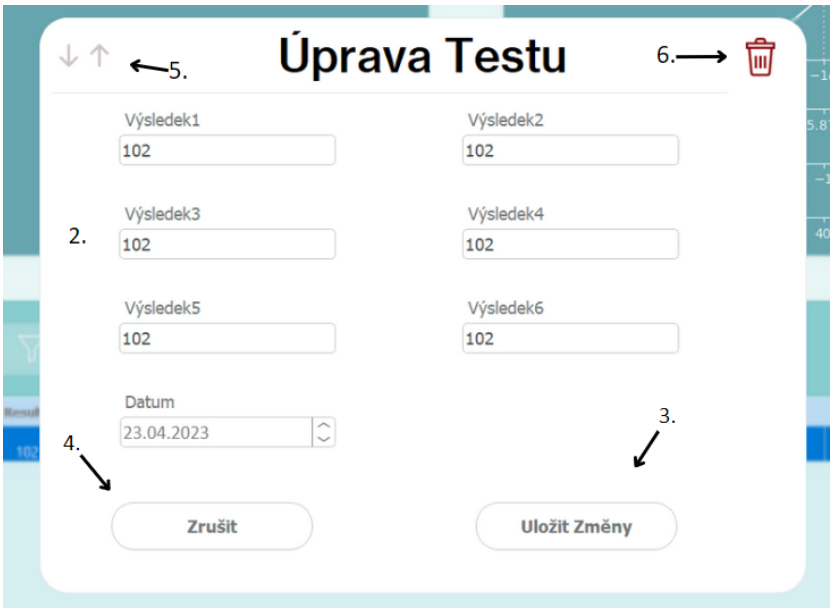
4.5.2. Úprava výsledků testu pacienta

Pro úpravu testu pacienta je potřeba kliknout na tlačítko s ikonou editování v postranní liště vedle daného testu pacienta (bod 1. Obrázek 37). Po kliknutí je zobrazeno dialogové okno, obsahující výsledky testu pacienta a datum, kdy byl test vykonán (bod 2. Obrázek 38). Zde je možno test libovolně upravit. Pro uložení změn je potřeba kliknout na tlačítko *Uložit Změny* (bod 3. Obrázek 38). Po kliknutí jsou provedené změny uloženy a dialogové okno je aplikací uzavřeno.

Dialogové okno lze zavřít pomocí tlačítka *Zrušit*, nacházející se nalevo od tlačítka *Uložit Změny* (bod 4. Obrázek 38).

Pokud byl pacient testován vícekrát, je možno mezi jeho jednotlivými výsledky testů přecházet pomocí dvojice šipek v levém horním rohu dialogového okna (bod 5. Obrázek 38). Jednotlivé testy lze odstranit pomocí tlačítka s ikonou odpadkového koše v pravém horním rohu dialogového okna (bod 6. Obrázek 38).

Obrázek 37: Zobrazení dialogového okna pro úpravu testu pacienta.



Obrázek 38: Dialogové okno pro úpravu testu pacienta

4.5.3. Odstranění výsledku testu pacienta

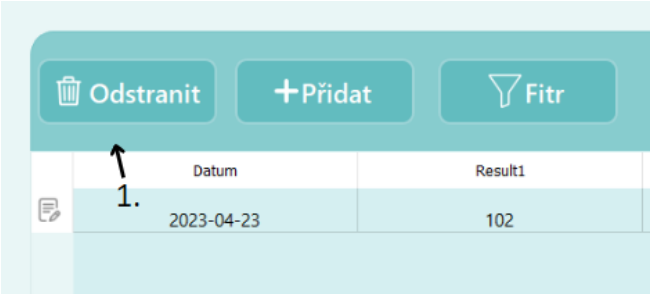
Test pacienta je možno odstranit pomocí tlačítka *Odstranit* na stránce testů pacienta nebo pomocí dialogového okna pro úpravu testu pacienta.

Tlačítko pro odstranění

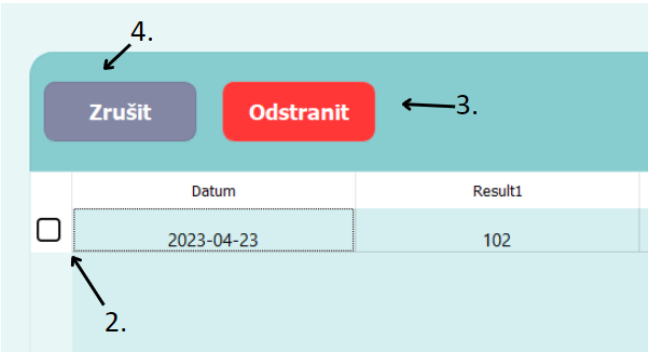
Pomocí této metody je možno odstranit jeden nebo více testů pacienta než v případě metody odstranění za pomoci dialogového okna pro úpravu testu.

Pro odstranění je potřeba kliknout na tlačítko *Odstranit* na stránce testů pacienta aplikace (bod 1. Obrázek 38). Po kliknutí na tlačítko *Odstranit* je změněna hlavní lišta a postranní lišta aplikace pro vymazání pacientských testů. V postranní liště je potřeba vybrat test nebo testy pacienta pro odstranění (bod 2. Obrázek 40), kliknout na tlačítko *Odstranit* (bod 3. Obrázek 40) a potvrdit odstranění v dialogovém okně (Obrázek 41), které je zobrazeno po kliknutí na tlačítko *Odstranit* na stránce testů pacienta aplikace.

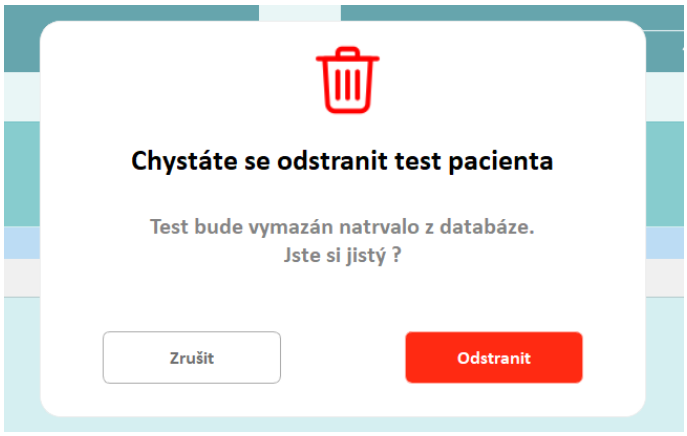
Odstranění testu je možno zrušit kliknutím na tlačítko *Zrušit* na stránce testů pacienta aplikace (bod 4. Obrázek 40).



Obrázek 39: Odstranění testu pacienta pomocí tlačítka odstranění



Obrázek 40: Výběr a odstranění testu pacienta



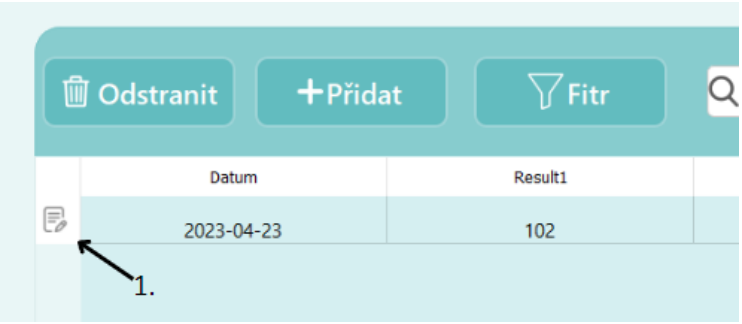
Obrázek 41: Dialogové okno pro potvrzení odstranění testu pacienta

Dialogové okno pro úpravu testu pacienta

Druhou možností pro odstranění testu pacienta je za pomoci dialogového okna pro úpravu testu pacienta.

Pro odstranění je potřeba kliknout v postranní liště vedle testu pacienta na ikonu editování pro zobrazení dialogového okna pro úpravu testu (bod 1. Obrázek 42) a následně kliknout na tlačítko s ikonou odpadkového koše (bod 2. Obrázek 43) nacházející se v pravém horním rohu dialogového okna. Vymazání testu pacienta je stejně jako v předchozí metodě potřeba potvrdit v dialogovém okně pro potvrzení odstranění (Obrázek 41), které je zobrazeno po kliknutí na tlačítko pro odstranění testu na dialogovém okně pro úpravu testu pacienta.

Pokud byl pacient testován pouze jednou, je dialogové okno automaticky uzavřeno. V opačném případě je v dialogovém okně pro úpravu testu zobrazen následující test pacienta.



Obrázek 42: Odstranění testu pacienta pomocí dialogového okna pro úpravu testu pacienta

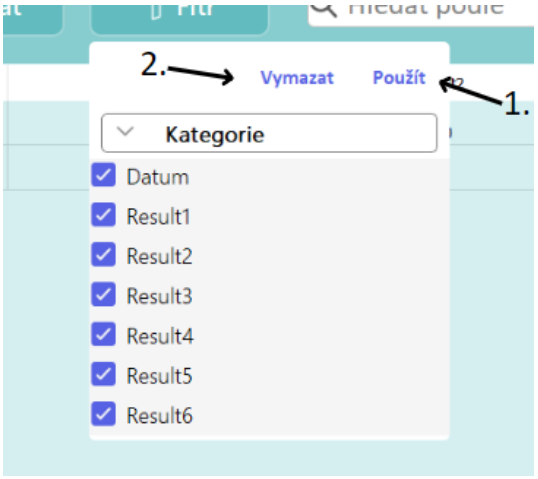
Obrázek 43: Dialogové okno pro úpravu testu pacienta – odstranění testu

4.5.4. Filtrace testů pacienta

Pomocí filtru lze nastavit, jak budou zobrazeny jednotlivé testy pacienta.

Pro zobrazení filtru je potřeba kliknout na tlačítko *Filtr* na stránce testů pacienta (bod 1. Obrázek 44). Po kliknutí je zobrazeno dialogové okno filtru, kde je možno vybrat, které parametry testu budou aplikací zobrazeny, např. datum, kdy byl test vykonán nebo jednotlivé výsledky testu pacienta (Result1, Result2 atd.). Pro aplikaci nastaveného filtru je potřeba kliknout na tlačítko *Použít* (bod 2. Obrázek 45). Nastavený filtr je možno zrušit kliknutím na tlačítko *Vymazat* v dialogovém okně filtru (bod 3. Obrázek 45).

Obrázek 44: Stránka testů pacienta – zobrazení filtru



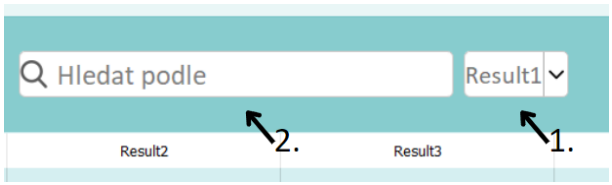
Obrázek 45: Nastavení filtru testů pacienta

4.5.5. Vyhledání testu pacienta

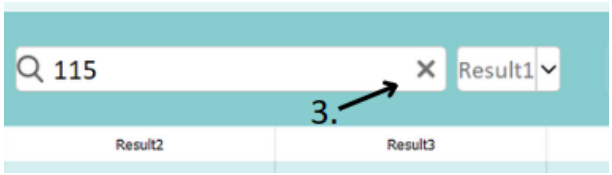
Vyhledávání testů pacienta je možno podle datumu, kdy byl test vykonán nebo podle dané hodnoty výsledku testů (Result1, Result2 atd..).

Pro vyhledání daného testu je potřeba vybrat kategorii, podle které bude test pacienta aplikací vyhledáván (bod 1. Obrázek 46) a zadat výraz pro hledání (bod 2. Obrázek 46). Nakonec stisknout klávesu Enter na klávesnici.

Pro Zrušení vyhledávání slouží tlačítko s ikonou X (bod 3. Obrázek 47).



Obrázek 46:Vyhledávání testu pacienta



Obrázek 47: Zrušení vyhledávání

4.5.6. Export do PDF

Údaje o daném pacientovy a jeho výsledky testů lze exportovat do souboru formátu PDF.

Pro export je potřeba kliknout na tlačítko *Export* (bod 1. Obrázek 48) na stránce testů pacienta. Po kliknutí je zobrazeno dialogové okno pro export do PDF. Zde je potřeba vybrat adresu, kam bude uložen výsledný report o pacientovy (bod 2. Obrázek 49). PDF report lze zabezpečit heslem vybráním možnosti nastavit heslo (bod 4. Obrázek 49). Po vybrání je zobrazeno vstupní pole pro zadání hesla, kterým bude zabezpečen výsledný report o pacientovy.

Nakonec je potřeba kliknout tlačítko *Export* (bod 3. Obrázek 49).

Obrázek 48: Stránka pacientských testů – export do PDF

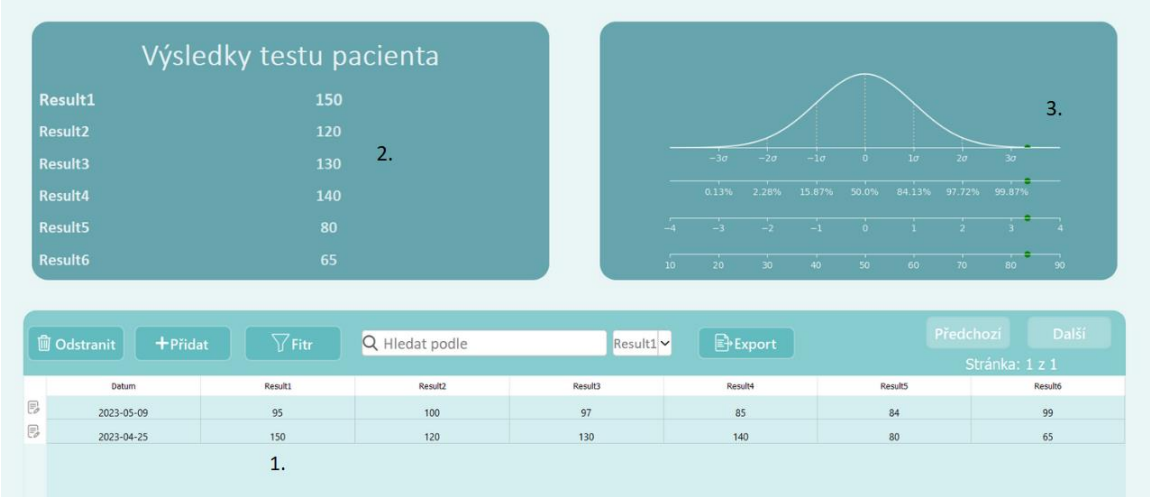
Obrázek 49: Ukázka exportu výsledků testů a údajů o pacientovy do formátu PDF

4.5.7. Vizualizace výsledků testu pacienta

Jednotlivé výsledky testu pacienta lze vizualizovat v jednotlivých metrických škálách (Gaussovo rozdělení, percentil, z-skóre a t-skóre) nacházející se v pravé horní okně stránky testů pacienta aplikace.

Pro vizualizaci je potřeba kliknout na zvolený test v tabulce testů pacienta (bod 1. Obrázek 50). Po klinutí na zvolený test, jsou výsledky testu pacienta zobrazeny v levém horním okně obrazovky aplikace (bod 2. Obrázek 50) a v metrických škálách v pravém horním okně obrazovky

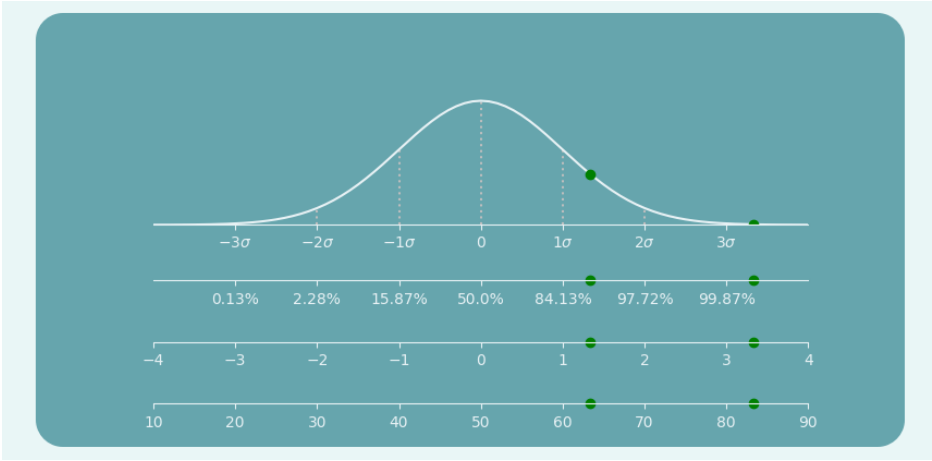
je vizualizován první výsledek ze zvoleného testu pacienta (bod 3. Obrázek 50). Pro zobrazení dalšího výsledku z testu pacienta v metrických škálách je potřeba kliknout na zvolený výsledek v levém horním okně obrazovky. Po kliknutí je zvolený výsledek zobrazen v jednotlivých metrických škálách v pravém horním okně aplikace.



Obrázek 50: Vizualizace výsledku testu pacienta

4.5.8. Porovnání dvou výsledků z testu pacienta

Jednotlivé výsledky z testu pacienta lze porovnávat v metrických škálách. Pro porovnání dvou výsledků z testu pacienta je potřeba nejprve vybrat, kliknout na hodnotu prvního výsledku pro porovnání v levém horním okně obrazovky aplikace a následně najet myší na druhý zvolený výsledek z testu pacienta. Po najetí myši na zvolený výsledek z testu pacienta, je hodnota výsledku zobrazena společně s první zvolenou hodnotou v metrických škálách (Obrázek 51).



Obrázek 51: Porovnání dvou zvolených výsledků z testu pacienta

5. Testování

Testování je důležitou fází ve vývoji aplikace, při které jsou zjišťovány jak chyby v implementaci, tak i nedostatky v designové části aplikace. Pro testování bylo vybráno několik různých uživatelů, kterým byl poslán odkaz na stáhnutí a instalaci aplikace. Následně byla domluveno setkání, při kterém byla aplikace dle bodů testovacího scénáře (uvedených v kapitole 5.1) testovacími uživateli procházena a jednotlivé problémy, nejasnosti a výtky na aplikaci byly zaznamenány. Výsledky testování aplikace a nalezené chyby byly uvedeny v kapitolách 5.2 a 5.3.

5.1. Testovací scénář

Níže jsou uvedeny body, podle kterých byla aplikace testovacími uživateli procházena.

1. Zaregistrujte se do aplikace.
2. Přihlaste se a otestujte odhlášení z aplikace.
3. Obnovte si staré heslo a pomocí nového hesla se přihlaste do aplikace.
4. Přidejte alespoň dva pacienty.
5. Zvolte jednoho pacienta, u kterého upravíte jeden z jeho údajů.
6. Otestuje vyhledání záznamu pacienta v pacientské stránce aplikace podle Vámi zvolené kategorie.
7. Vytvořte filtr pacientů.
8. K jednomu z Vámi přidaných pacientů přidejte alespoň 2 testy a následně otestujte vizualizaci jednotlivých výsledků testu pacienta.
9. Otestuje vyhledání testu pacienta na stránce testů pacienta aplikace podle Vámi zvolené kategorie.
10. Vytvořte filtr testů pacienta.
11. Otestujte odstranění testu pacienta pomocí tlačítka *Odstranit* nebo pomocí dialogového okna pro úpravu testů pacienta.
12. Vraťte se na pacientkou stránku aplikace a zde otestujte vymazání záznamu pacienta pomocí tlačítka *Odstranit* nebo pomocí dialogového okna pro úpravu údajů pacienta.
13. Změňte na stránce nastavení aplikace jazyk aplikace.

5.2. Výsledky testování

Průchod testovacích uživatelů aplikací byl bez větších problémů. Ohlasy na aplikaci převážně pozitivní, ale i tak byly zaznamenány následující výtky:

- „Aplikace je celkem hodně světlá, preferovala bych spíše tmavší verzi, která ale není aplikací podporována."
- „Kód, který je zaslaný na email při obnově hesla je dlouhý a hůře se čte. Chtělo by zaslaný kód zkrátit anebo rozdělit na více částí."
- „V pacientské stránce aplikace chybí informace o poloze v postranním panelu."
- „Při vyhledávání záznamu pacienta je potřeba zadat přesně daný údaj podle, kterého bude aplikace vyhledávat. Proto bych rozšířil aplikaci o částečné vyhledání, kdy bude stačit zadat jen část údaje."
- „Ze začátku mi nebylo jasné, jak přejít ze stránky pacientů na stránku testu daného pacienta."
- „Rozšířil bych filtry v aplikaci o možno vytvoření intervalů, např. u uložených pacientů by byla možnost si zvolit věkové rozmezí, které bude aplikací zobrazeno."

5.3. Nalezené chyby

Při testování aplikace testovacími uživateli byly objeveny následující chyby:

- Při registraci nového uživatele chybí kontrola, zda zadaná emailová adresa je validní.
- Při obnově hesla chybí kontrola, zda je počítač připojen k internetu.
- Při úpravě rodného čísla, datumu narození anebo pohlaví pacienta není zkontrolována validita rodného čísla pacienta.
- Datumu, kdy byl test vykonán, na stránce testů pacienta je zobrazován ve špatném formátu.

6. Závěr

V rámci této bakalářské práce jsem vytvořil desktopovou aplikaci, jejímž úkolem je vedení záznamů o pacientech a výsledcích jejich vyšetření, které jsou vyhodnocovány a vizualizovány v jednotlivých metrických škálách. Součástí implementace je i výstup údajů o pacientech a jejich vyšetřeních do souboru ve formátu PDF.

Aplikace vznikla pro potřeby zjednodušení psychometrické analýzy, a to na základě zadání psychologů Kliniky dětské neurologie, Fakultní nemocnice v Motole. V rámci realizace této aplikace bylo nutno nastudovat a využít mnoha různých modalit softwaru, a to od návrhu grafického rozhraní, přes vytváření databáze, zabezpečení a šifrování dat, až k finálnímu testování aplikace.

Grafické rozhraní aplikace bylo navrženo tak, aby bylo uživatelsky přehledné, s profesionálně působícím designem sledujícím současné trendy. Zároveň byl kladen důraz, aby jednotlivé obrazovky aplikace byly sestaveny tak, aby jejich použití bylo pro uživatele co nejvíce intuitivní a aby ho logicky vedly při jeho práci.

Pro vytvoření databáze jsem využil relační databázový systém *SQLite*. Tento systém má potenciál zpracovávat i velké objemy dat. V době dokončení práce nebyly k dispozici reálná data a výsledky normativní studie, proto je databáze připravena se základními atributy, které je možno v budoucnu podle potřeby dále upravovat a rozšiřovat.

Víceúrovňové zabezpečení dat v aplikaci bylo realizováno symetrickým a asymetrickým šifrováním a kryptografickým hashovaním s důrazem na ochranu uživatelských a pacientských osobních údajů. Implementované řešení by mělo v dostatečné míře chránit citlivá data pacientů a uživatelů a sleduje aktuální trendy v zabezpečení dat.

V rámci testování byly nalezeny nedostatky v implementovaném řešení aplikace, které nebylo možno z důvodu nedostatku času opravit. Zároveň byly doporučeny změny v uživatelském rozhraní aplikace (např. snížení jasu aplikace, zlepšení orientace v postranních panelech aplikace) a přidání funkcionalit rozšiřující vyhledávání a filtrování záznamů pacientů a jejich vyšetření.

Výsledná aplikace je určena pro operační systém Windows, pro který byl vytvořen instalační program. V budoucnu by bylo dobré vytvořit verzi aplikace podporující jiné operační systémy, např. Linux. V řešení aplikace nebyla realizována komunikace se serverem. Případné doplnění této funkcionality v budoucnosti umožní licencování aplikace a zálohování dat aplikace.

Vyvinutá aplikace je připravena na implementaci normativních dat, což je plánováno k realizaci v druhé polovině roku 2023. Aplikace je volně ke stažení v Github repozitáři

[45].Tímto byly splněny všechny body zadání bakalářské práce a představovaná aplikace nyní funguje jako technologický demonstrátor.

7. Použitá literatura

[1] G. L. Holmes, "Cognitive impairment in epilepsy: the role of network abnormalities," *Epileptic Disorders*, vol. 17, no. 2, pp. 101–116, Jun. 2015, doi: 10.1684/epd.2015.0739.

[2] K. Bukáčová, P. Lhotová, and A. Maulisová, "Neuropsychologická testová baterie pro děti," *E-psychologie*, vol. 15, no. 1, pp. 90–91, 2021, doi: https://doi.org/10.29364/epsy.394.

[3] Krejčířová Dana, Boschek Petr, and Dan Jiří, *WISC-III – Wechslerova inteligenční škála pro děti*, 1. Praha: Testcentrum, 2002.

[4] Grob Alexander, Meyer Christine S., and Hagmann-von Arx Priska, *IDS – Inteligenční a vývojová škála pro děti ve věku 5–10 let*, 1. Praha: Hogrefe–Testcentrum, 2013.

[5] Reynolds R.Cecil and Voress K. Judith, *TOMAL-2: Test of Memory and Learning – Second Edition*, 2. 2007.

[6] C. Zucchella, A. Federico, A. Martini, M. Tinazzi, M. Bartolo, and S. Tamburin, "Neuropsychological testing," *Pract Neurol*, vol. 18, no. 3, pp. 227–237, Jun. 2018, doi: 10.1136/practneurol-2017-001743.

[7] Dorazilová Aneta, *Skripta k předmětu Psychologie obecná a osobnosti: Kognitivní psychologie Mgr. Aneta Dorazilová*. Hradec Králové: Univerzita Hradec Králové , 2014. Accessed: Apr. 29, 2023. [Online]. Available: https://www.uhk.cz/file/edee/filozoficka-fakulta/studium/dorazilova_-_psychologie_obecna_a_osobnosti.pdf

[8] Z. Dvořáková, "Možnosti diagnostiky kognitivních funkcí u neurologických pacientů. Převod a validizace Neuropsychological Assessment Battery (NAB)," Univerzita Karlova, Filozofická fakulta, Katedra psychologie, Praha, 2021.

[9] P. Kulišťák, *Klinická neuropsychologie v praxi*, Vydání první. Praha: Univerzita Karlova, nakladatelství Karolinum, 2017.

[10] Z. Dvořáková, "Možnosti diagnostiky kognitivních funkcí u neurologických pacientů," Univerzita Karlova, Praha, 2021.

[11] C. Berryman, T. R. Stanton, K. J. Bowering, A. Tabor, A. McFarlane, and G. L. Moseley, "Do people with chronic pain have impaired executive function? A meta-analytical review," *Clin Psychol Rev*, vol. 34, no. 7, pp. 563–579, Nov. 2014, doi: 10.1016/J.CPR.2014.08.003.

[12] N. Hayes, *Základy sociální psychologie*, Vydání osmé. Praha: Portál, 2021.

[13] V. Plecerová and Y. Pužejová, *Psychologie*. České Budějovice: Střední zdravotnická škola a Vyšší odborná škola zdravotnická České Budějovice, 2016. [Online]. Available: https://publi.cz/books/339/14.html

[14] I. Plevová and A. Petrová, *Obecná psychologie*, 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 2012.

[15] R. Šikl, *Zrakové vnímání*. in Psyché (Grada). Grada, 2012. [Online]. Available: https://books.google.cz/books?id=baNKgp5u7oIC

[16] M. Valenta and kol., *Katalog posuzování míry speciálních vzdělávacích potřeb Část II. (diagnostické domény pro žáky s mentálním postižením)*, 1. Olomouc: Univerzita Palackého, 2012.

[17] V. Čadilová, K. Thorová, Z. Žampachová, and a kol., *Katalog posuzování míry speciálních vzdělávacích potřeb Část II. (diagnostické domény pro žáky s poruchami autistického spektra)*, 1. Olomouc: Univerzita Palackého, 2012.

[18] V. Jozef, S. Ivan, and S. Eva, *Sociální psychologie: Teorie, metody, aplikace*. in Psyché (Grada). Grada, 2019. [Online]. Available: https://books.google.cz/books?id=4TWfDwAAQBAJ

[19] M. Svoboda, P. Humpolíček, and V. Šnorek, *Psychologická diagnostika dospělých*. in Studium. Portál, 2010. [Online]. Available: https://books.google.cz/books?id=4pvJDwAAQBAJ

[20] P. Michaela and K. Jana, *Přehled poruch psychického vývoje*. Grada Publishing a.s., 2016. [Online]. Available: https://books.google.cz/books?id=WC-fDgAAQBAJ

[21] Baštecká Bohumila and Goldmann Petr, *Základy klinické psychologie*, 1. vyd. Praha: Portál, 2001.

[22] Wechsler David, *Wechsler Abbreviated Scale of Intelligence–Second Edition*, 2nd ed. Los Angeles: Pearson, 2011. doi: https://doi.org/10.1037/t15171-000.

[23] P. Soukup, P. Mareš, and L. Rabušic, *Statistická analýza sociálněvědních dat (prostřednictvím SPSS)*. Masarykova univerzita, 2019. [Online]. Available: https://books.google.cz/books?id=_hy1EAAAQBAJ

[24] Holčík Jiří and Komenda Martin, *Matematická biologie: e-learningová učebnice [online]*, 1st ed. Brno: Masarykova univerzita, 2015. Accessed: Apr. 29, 2023. [Online]. Available: https://portal.matematickabiologie.cz/

[25] D. Dostál, "Statistické metody v psychologii studijní opora pro rok 2022/23 k předmětům BSMP1, DSMP1, BSMP2 a DSMP2."

[26] J. Hendl, *Statistika v aplikacích*, Vyd. 1. Praha: Portál, 2014.

[27] M. Svoboda, P. Humpolíček, and V. Šnorek, *Psychologická diagnostika dospělých*. in Studium. Portál, 2010. [Online]. Available: https://books.google.cz/books?id=4pvJDwAAQBAJ

[28] E. Langford, "Quartiles in Elementary Statistics," *Journal of Statistics Education*, vol. 14, no. 3, p. null-null, Jan. 2006, doi: 10.1080/10691898.2006.11910589.

[29] Lemonaki Dionysia, "Frontend VS Backend – What's the Difference?" https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/#feintro (accessed May 07, 2023).

[30] "Python – Wikipedie." https://cs.wikipedia.org/wiki/Python (accessed May 06, 2023).

[31] M. Summerfield and L. Krejčí, *Python 3 Výukový kurz*, 1. Brno: Computer Press, 2013.

[32] Michael. Owens, *The definitive guide to SQLite*. Apress, 2006.

[33] Marek Laurenčík and Michal Bureš, *SQL – Podrobný průvodce uživatele*, 1st ed. Praha: Grada Publishing, a. s., 2018.

[34] "About SQLite." https://www.sqlite.org/about.html (accessed Apr. 28, 2023).

[35] Bezpalec Pavel, *Nové trendy v elektronických komunikacích Kryptografie*, 1st ed. Praha: Čvut, 2015. Accessed: May 07, 2023. [Online]. Available: https://publi.cz/book/232-kryptografie?b=0p6jq

[36] Durčák Pavel, "Symetrické a asymetrické šifrování | NaPočítači.cz." https://www.napocitaci.cz/33/symetricke-a-asymetricke-sifrovani-uniqueidgOkE4NvrWuNY54vrLeM677jX7sp3Lu-ZpLpGVMy1prA/ (accessed May 07, 2023).

[37] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Berlin Heidelberg, 2009. [Online]. Available: https://books.google.cz/books?id=f24wFELSzkoC

[38] “RSA — Cryptography 41.0.0.dev1 documentation.” https://cryptography.io/en/latest/hazmat/primitives/asymmetric/rsa/ (accessed May 07, 2023).

[39] Oštádal Radim, “Teoretický základ a přehled kryptografických hashovacích funkcí.” 2012.

[40] Janko David, “Lekce 3 - Hesla a biometrická ochrana.” https://www.itnetwork.cz/bezpecnost/hesla-a-biometricka-ochrana (accessed May 07, 2023).

[41] P. George Jose, S. Chatterjee, M. Patodia, S. Kabra, and A. Nath, “Hash and Salt based Steganographic Approach with Modified LSB Encoding Duplicate File Analyzer using N-layer Hash and Hash Table View project Advanced Image Encryption and Data Hiding Techniques View project Hash and Salt based Steganographic Approach with Modified LSB Encoding,” *Article in International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3297, no. 6, 2016, doi: 10.15680/IJIRCCE.2016.0406054.

[42] “bcrypt - Wikipedia.” https://en.wikipedia.org/wiki/Bcrypt#cite_note-provos-1 (accessed May 07, 2023).

[43] “Key derivation function - Wikipedia.” https://en.wikipedia.org/wiki/Key_derivation_function (accessed May 07, 2023).

[44] “Riverbank Computing | Introduction.” https://www.riverbankcomputing.com/software/pyqt/ (accessed May 07, 2023).

[45] Martin Bulant, “Github repozitář projektu.” https://github.com/EpiReC-ISARG/NBD-calculator/releases (accessed May 20, 2023).