



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**PLATFORMA PRO ZPRACOVÁNÍ FORMULÁŘŮ PRO
ZDRAVOTNICKÁ ZAŘÍZENÍ**

FORM PROCESSING PLATFORM FOR MEDICAL FACILITIES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PAVEL ŠUSTEK

VEDOUcí PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE, Ph.D.

BRNO 2019

Zadání diplomové práce



22190

Student: **Šustek Pavel, Bc.**

Program: Informační technologie Obor: Informační systémy

Název: **Platforma pro zpracování formulářů pro zdravotnická zařízení**

Form Processing Platform for Medical Facilities

Kategorie: Informační systémy

Zadání:

1. Seznamte se současným stavem zpracovávání formulářů ve zdravotnickém zařízení a zmapujte konkurenci v oblasti automatizované digitalizace papírových formulářů.
2. Formálně i neformálně popište navrhovanou platformu. Navrhněte konkrétní způsob využití, začlenění do procesů a nasazení platformy v zúčastněném zdravotnickém zařízení.
3. Implementujte nezbytnou část aplikace tak, aby mohla být nasazena a testována ve zdravotnickém zařízení.
4. Na základě uživatelských reakcí a zpětné vazby během testování ve zdravotnickém zařízení tuto platformu vylepšujte. Cílem je efektivní začlenění do procesů, přínos v oblasti snadné digitalizace dat a také jednoduchost a přehlednost ovládání pro méně technicky zdatné uživatele.
5. Reflektujte závěry z testování. Diskutujte dosažené cíle a navrhněte směry dalšího vývoje.
6. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Podle pokynů školitele

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část bodů 3 a 4 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Szóke Igor, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 22. května 2019

Datum schválení: 1. listopadu 2018

Abstrakt

Cílem této práce je vytvořit systém pro automatizaci a zjednodušení procesů spojených s papírovými formuláři ve spolupráci se zdravotnickým zařízením v Praze. Řešením je centralizovaný SaaS systém, který umožní jednotně tvořit papírové i digitální formuláře, s možností je spravovat, vyplnit online, či vytisknout na papír a automaticky načítat vyplněné hodnoty pomocí OMR z oskenovaných dokumentů. Poskytne také cloudové úložiště, export a bude připraven na integraci do rozsáhlé formulářové databáze Redcap. Pokryje tedy celý životní cyklus formuláře, což je největší výhodou. Je důležité systém navrhnout uživatelsky přívětivě a provést důkladné testování se zpětnou vazbou přímo ve zdravotnickém zařízení. Hlavní použité technologie jsou ASP.NET (MVC) a C#, JavaScript, HTML5, CSS3 a MSSQL. Výsledek práce by měl být výchozím bodem pro následný startup.

Abstract

The aim of this thesis is to develop a system for automation and simplification of paper forms processing in cooperation with gerontological facility. The solution is an SaaS system, that provides tools to design paper and digital forms, managing them, fill them online or print it on paper and then automatically process scanned documents via OCR algorithm. It also provides a simple cloud storage and the integration with complex form and survey database Redcap. It is very important to design a user friendly interface and to perform a set of thorough tests with feedback from medical facility. Fundamental used technologies are ASP.NET (MVC) and C#, Javascript, HTML5, CSS3 and MSSQL. The output of this work should lay the foundations to an ongoing startup.

Klíčová slova

Digitalizace dat, zdravotnictví, automatizace, formuláře, software jako služba, cloud

Keywords

Data digitization, healthcare, automation, forms, software as service, cloud

Citace

ŠUSTEK, Pavel. *Platforma pro zpracování formulářů pro zdravotnická zařízení*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szöke, Ph.D.

Platforma pro zpracování formulářů pro zdravotnická zařízení

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szökeho, Ph.D. Konzultace mi také poskytl Mgr. Radim Krupička, Ph.D. z ČVUT v Praze. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Šustek
19. května 2019

Poděkování

V první řadě bych rád poděkoval svému vedoucímu diplomové práce panu doktorovi Igoru Szökemu za ochotu při konzultacích, rady a čas, který mi věnoval. Dále bych chtěl vyjádřit poděkování panu doktorovi Radimu Krupičkovi, který zprostředkoval setkání se zdravotnickým zařízením a možnost reálného nasazení. V souvislosti s tím i zástupcům Gerontologického centra v Praze 8, za cenné poznatky během testování a kvalitní zpětnou vazbu.

Obsah

1	Úvod	3
2	Analýza současného stavu	5
2.1	Role formulářů v gerontologii	5
2.2	Aktuální stav práce s formuláři	8
2.3	Konkurence v oblasti digitalizace práce s formuláři	13
3	Návrh nového řešení	17
3.1	Požadavky na nové řešení	17
3.1.1	Požadavky ze strany zdravotnického zařízení	17
3.1.2	Další potřebné vlastnosti platformy	18
3.2	Návrh systému a integrace do stávajících procesů	19
3.3	Návrh s využitím UML	21
3.3.1	Diagram případů užití	21
3.3.2	Diagram komponent	22
4	Implementace	23
4.1	Metodika vývoje	23
4.2	Uživatelské rozhraní a popis aplikace	24
4.2.1	Přihlašovací stránka a domovská nástěnka	25
4.2.2	Navigační menu a akční panel	26
4.2.3	Způsob manipulace s dokumenty	26
4.2.4	Práce s šablonami a designer formulářů	27
4.2.5	Digitalizační nástroj	31
4.2.6	Zpracované formuláře a export	34
4.3	Technologie	36
4.4	Architektura	37
4.5	Struktura a soubory aplikace	39
4.6	Vybrané implementační detaily	42
4.6.1	Designer	42
4.6.2	OMR algoritmus a automatizace	44
4.6.3	Převod XML šablony do HTML	46
4.6.4	Správa souborů na disku - JSTree	46
4.6.5	REST API pro integraci	48
5	Testování	49
5.1	Testování kódu	49
5.2	Uživatelské testování	49

5.2.1	Sledované metriky a cíle	50
5.2.2	Průběh	50
5.2.3	Výsledky	51
5.3	Závěr z testování	54
6	Závěr	55
	Literatura	57

Kapitola 1

Úvod

Papírové, či elektronické formuláře jsou nedílnou součástí získávání cenných dat. V dnešní době je kladen velký důraz na digitalizaci informací získaných z těchto formulářů. Hlavním důvodem je potřeba tato data zpřístupnit k analýze, poskytnout je pro další výzkum nebo pro statistické vyhodnocování. Bohužel existují oblasti, kde není jednoduché podávat formuláře k vyplnění pouze elektronicky a při snaze o digitalizaci ručně vyplněných dokumentů jsou potřeba zbytečně složité procesy a kombinace různých systémů.

Cílem této práce je minimalizovat počet potřebných nástrojů během životního cyklu dokumentu a tedy vytvořit komplexní platformu, která umožní vytvářet jednotné papírové i digitální formuláře pomocí integrovaného designeru. Nabídne možnost je vyplnit elektronicky, či vytisknout na papír. Při zpracovávání tištěných dokumentů navíc bude usnadňovat práci automatizovaným rozpoznáváním dat z naskenovaného obrazu. Poskytne také jednoduché cloudové uložení a bude připravena na možnou integraci do rozsáhlé formulářové databáze REDCap. Celé řešení bude fungovat jako jednotná webová aplikace a bude umístěna na internetové adrese <https://formello.cz>.

Platforma by měla být v budoucnu univerzálním nástrojem pro práci s formuláři. Avšak aby byl vývoj postaven na reálných základech a výsledná aplikace byla co nejvíce použitelná v praxi, je tvorba platformy již od začátku stavěna na reálném problému, který řeší. Tím jsou geriatrická funkční vyšetření a práce s nimi. Tato problematika byla vybrána proto, že procesy v této oblasti a životní cyklus vyšetření, lze velmi dobře zobecnit na jiná využití formulářů a vyvíjené aplikace.

Pro vývoj a testování platformy bude tedy probíhat spolupráce s Gerontologickým centrem v Praze 8, které spolu s Fakultou humanitních studií Univerzity Karlovy a Fakultou biomedicínského inženýrství na ČVUT v Praze, pracuje na projektu s názvem „NV18-09-00587 - Dlouhodobá péče pro pacienty s demencí, analýza současné situace u nás a v Evropě, možnosti transformace institucionální péče s ohledem na zlepšení kvality života, kvalitu péče a využití zdrojů.“ Zařízení v současné době řeší problém s digitalizací funkčních vyšetření pacientů a s využitím finančních prostředků z projektu se snaží realizovat pokrok v dané oblasti. Tato diplomová práce je díky tomu úzce spjata s gerontologickou oblastí, aby vznikl reálně použitelný nástroj pro danou zdravotnickou kliniku. Je důležité systém navrhnout uživatelsky přívětivě a provést důkladné testování se zpětnou vazbou přímo v gerontologickém centru.

V rámci implementace bude navázáno na mou bakalářskou práci, ve které jsem navrhl a vytvořil prototypovou verzi algoritmu pro automatické rozpoznávání dat z papírových dotazníků pro využití právě v gerontologii. Implementoval jsem mechanismus, který dovede ve webovém prostředí načíst data z ručně vyplněných formulářů. Tento algoritmus bude tedy

převzat a začleněn jako jedna z vlastností celého systému. Konkrétně v části převodu papírových formulářů do elektronické podoby. Na výstup mé bakalářské práce navážu odladěním celého procesu digitalizace, pečlivým otestováním algoritmu a obalením dalšími potřebnými kroky pro reálné použití. Mimo převod do elektronické podoby bude především doplněna možnost vytvářet formuláře přímo v internetovém prohlížeči, skladovat je a procházet data přímo v platformě. Tím by měla být pokryta základní práce s tištěnými formuláři. Cenné znalosti procesů a stavu digitalizace v cílovém zdravotnickém zařízení, nabyté při bakalářské práci, budou velmi užitečné při analýze současné situace a návrhu řešení. Platforma bude také vycházet z výsledků dosavadní snahy o zlepšení digitalizace v cílové klinice, především vyvarováním se předešlých chyb, ale i navázáním na některé stávající a zavedené postupy a projekty. Více o aktuálním stavu je uvedeno v kapitole 2.

Diplomová práce je rozdělena do šesti kapitol. V kapitole 2 je analyzován současný stav řešení práce s formuláři v gerontologickém zařízení, objasněna role formulářů v této oblasti a provedeno porovnání konkurence v digitalizaci dotazníků. Kapitola 3 rozebírá požadavky na platformu z hlediska zdravotnického zařízení, ale i další potřebné vlastnosti, kterými musí řešení disponovat. Popisuje způsob, jakým bude aplikace začleněna do stávajících procesů a jak bude testována a nakonec rozebírá návrh platformy z technického hlediska a uživatelského rozhraní. Část 4 se zabývá implementací. Obsahuje tedy výčet použitých technologií a jejich zdůvodnění, postup při tvorbě aplikace a popis zvolené architektury. Dále jsou vybrány a rozebrány některé implementační detaily a zajímavá místa vývoje. Je zde také vysvětlena adresářová struktura aplikace. Kapitola 5 se věnuje testování ve zdravotnickém zařízení, jeho výsledkům a případným provedeným změnám. Poslední kapitola – 6, shrnuje výsledky práce a diskutuje dosažené cíle a možnosti dalšího rozvoje.

Kapitola 2

Analýza současného stavu

Jak již bylo zmíněno v úvodu, platforma by do budoucna měla být univerzální pro libovolné formuláře. Avšak pro lepší uchopení vývoje a testování byla zvolena specifická oblast, čelící problému digitalizace formulářů – gerontologie. Konkrétně se jedná o geriatrická funkční vyšetření. Před návrhem řešení a vývojem platformy je tedy potřeba tuto problematiku velmi dobře zanalyzovat. Zároveň se zamyslet, jak navrhnout nástroj tak, aby co nejvíce podpořil práci s funkčními vyšetřeními a urychlil procesy s tím spojené. Jelikož vývoj pro Gerontologické centrum v Praze 8 a testování nástroje přímo v tamních podmínkách je hlavní částí této diplomové práce, bude se vysvětlením problému a současné situace zabývat celá následující kapitola.

Pro lepší pochopení problematiky funkčních vyšetření je třeba stručně popsat gerontologii. Jedná se o vědní obor, který se zabývá souhrnem poznatků o stáří. Dělí se na tři typy:

- Klinická (Geriatric) – Řeší zdravotní stav a choroby ve stáří. Týká se všech klinických oborů dospělého věku. Zvláštní pozornost je věnována stařecké multimorbiditě, demenci, hypomobilitě, imobilitě, či kognitivnímu deficitu a dalším.
- Sociální – Analyzuje vztahy starého člověka a společnosti z pohledu demografie, psychologie, ekonomie či politologie.
- Experimentální – Zkoumá příčiny a mechanismy stárnutí.

Geriatrická funkční vyšetření se úzce vztahují především ke klinické a sociální gerontologii [2].

2.1 Role formulářů v gerontologii

Formuláře jsou nezbytnou součástí gerontologie. Jejich formou jsou vytvářeny testy, které díky soustavě vysoce specializovaných otázek, či úkolů, umožňují diagnostikovat zhoršení stavu nebo určit léčebný plán. Mohou také odhalit dosud nedignostikované poruchy. Velmi významnou roli hrají i při posuzování soběstačnosti pacienta, což může vést k rozhodnutí o poskytnutí typu a úrovně sociální péče. Provedení geriatrického funkčního vyšetření není pouze jednorázová činnost, vykonává se opakovaně, aby bylo možné sledovat informace o účinnosti léčby a průběhu onemocnění. Z pohledu gerontologie se tedy jedná o velké množství cenných dat o každém pacientovi. Získané informace jsou následně důležité pro

zlepšování celkové úrovně poskytované péče a pro vyhodnocování léčby pacienta. Neméně důležité jsou pro výzkum, analýzu a rozvoj v této oblasti.

Existují tři základní typy gerontologických vyšetření. Hodnocení stavu výživy (MNA), bolesti (analogové škály – obrázek 2.1) a posuzování kognitivních poruch (MOBID a PAI-NAD).

The form is titled "Hodnocení největší bolesti" (Pain Assessment). It includes fields for "Pacient číslo" (Patient number), "Oddělení" (Department), and "Datum" (Date). The main table has columns for "Datum" (Date), "Dni" (Days 0-30), and "Důvod a místo bolesti/bolí při čem?" (Reason and location of pain/pain during what?). The table is filled with data for each day from 1 to 30, with checkboxes for each day and handwritten notes in the "Důvod a místo..." column.

Datum	Dni	0	1	2	3	4	5	6	7	8	9	10	Důvod a místo bolesti/bolí při čem?
3. 12.	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	bolesti SRK
4. 12.	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
5. 12.	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
6. 12.	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
7. 12.	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
8. 12.	6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
9. 12.	7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
10. 12.	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
11. 12.	9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
12. 12.	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
13. 12.	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
14. 12.	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
15. 12.	13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
16. 12.	14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
17. 12.	15	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
18. 12.	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
19. 12.	17	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
20. 12.	18	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
21. 12.	19	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
22. 12.	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
23. 12.	21	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
24. 12.	22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
25. 12.	23	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
26. 12.	24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
27. 12.	25	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
28. 12.	26	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
29. 12.	27	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
30. 12.	28	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
31. 12.	29	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—
1. 1.	30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	—

Obrázek 2.1: Geriatrické funkční vyšetření – Hodnocení bolesti [7]

Mimo to se používají další biomedicínská vyšetření ke konkrétním chorobám, která získávají informace o funkčním stavu pacientů, fyzické zdatnosti, poruchách smyslů nebo výživě. Pro představu, například k vyhodnocení mobility pacienta slouží test, kdy má subjekt za úkol vstát, ujít 3 metry a vrátit se zpět na židli. Pomocí zaškrťovacích polí jsou poté bodovány úkony jako chůze, ale i postavení se, či otočení. Ukázka takového vyšetření je na obrázku 2.2. Jiný příklad, z oblasti psychiky a mentálního stavu pacienta, kde se sledují kognitivní poruchy, deprese nebo jiné psychiatrické vady, je vyšetření MMSE (Mini-Mental-State Examination). Zde se zkoumají jednotlivé kognitivní domény a opět se pomocí zaškrťovacích polí hodnotí na škále s maximem 30 bodů. 26 značí patologický nález. Velmi známý a hojně používaný je test MoCA – Montrealský test, který funguje na stejném principu, ale je citlivější. Ke zjištění deprese se používá opět vyšetření založené na checkboxech¹ – GDS (Geriatrická škála deprese podle Yesavage) o 15 otázkách s danými možnostmi odpovědí. Dalším druhem formulářů jsou dotazníky, které zkoumají sociální oblast, to znamená kvalitu bydlení, společenské aktivity, ekonomickou situaci. Zde se jedná o různorodé otázky, velmi často typu ANO/NE.

¹zaškrťovací pole

Rodné číslo: [redacted] Oddělení: 001 Datum vstupu: 04.12.2019

Test mobility

1. Get up and go test
 Pacienta posadíme na přiměřeně vysokou židli (výška židle je taková, aby při flexi cca 90 stupňů v kolennou spočívala jeho chodidla pevně na podložce). Požádáme je, aby vstal, šel přesně 2,44 metrů k vyznačenému mísku, otočil se, vrátil se a znovu se posadil na židli.

3 body	2 body	1 bod	0 bodů
Vstane <input type="checkbox"/> bez pomoci	<input checked="" type="checkbox"/> s pomocí rukou	<input type="checkbox"/> s dopomocí druhé osoby	<input type="checkbox"/> nesvede
Chůze <input type="checkbox"/> stabilní	<input checked="" type="checkbox"/> s pomůckou nebo vřavou	<input type="checkbox"/> pouze s dopomocí, výrazná nestabilita	<input type="checkbox"/> nesvede
Otočení <input type="checkbox"/> jistě	<input checked="" type="checkbox"/> nejistě, zaváhání	<input type="checkbox"/> velmi nejistě, potřeba dopomoci	<input type="checkbox"/> nesvede
Usednutí <input type="checkbox"/> bez pomoci	<input checked="" type="checkbox"/> s pomocí rukou nebo opření	<input type="checkbox"/> s dopomocí	<input type="checkbox"/> nesvede

Zde zaznamenejte potřebný čas ve vteřinách, od začátku vstávání do usednutí: 119

2. Obvyklý způsob pohybu
 chůze bez pomůcek s vycházkovou hůl s francouzskou hůl/holemi
 vozík či elektrický vozík trvale na lůžku s chodítkem

3. 4 metrová chůze - počet vteřin potřebný k ujetí přesně 4 metrové vzdálenosti obvyklým tempem: 119
 Není schopen chůze Odmítl/a test Začal/a test, avšak nedokončil/a celou vzdálenost

Datum: [redacted]

3. 4 metrová chůze - počet vteřin potřebný k ujetí přesně 4 metrové vzdálenosti obvyklým tempem: [redacted]
 Není schopen chůze Odmítl/a test Začal/a test, avšak nedokončil/a celou vzdálenost

Datum výstupu: 05.12.2019

1. Get up and go test

3 body	2 body	1 bod	0 bodů
Vstane <input type="checkbox"/> bez pomoci	<input checked="" type="checkbox"/> s pomocí rukou	<input type="checkbox"/> s dopomocí druhé osoby	<input type="checkbox"/> nesvede
Chůze <input type="checkbox"/> stabilní	<input checked="" type="checkbox"/> s pomůckou nebo vřavou	<input type="checkbox"/> pouze s dopomocí, výrazná nestabilita	<input type="checkbox"/> nesvede
Otočení <input type="checkbox"/> jistě	<input checked="" type="checkbox"/> nejistě, zaváhání	<input type="checkbox"/> velmi nejistě, potřeba dopomoci	<input type="checkbox"/> nesvede
Usednutí <input type="checkbox"/> bez pomoci	<input checked="" type="checkbox"/> s pomocí rukou nebo opření	<input type="checkbox"/> s dopomocí	<input type="checkbox"/> nesvede

Zde zaznamenejte potřebný čas ve vteřinách, od začátku vstávání do usednutí: 116

2. Obvyklý způsob pohybu
 chůze bez pomůcek s vycházkovou hůl s francouzskou hůl/holemi
 vozík či elektrický vozík trvale na lůžku s chodítkem

3. 4 metrová chůze - počet vteřin potřebný k ujetí přesně 4 metrové vzdálenosti obvyklým tempem: 115
 Není schopen chůze Odmítl/a test Začal/a test, avšak nedokončil/a celou vzdálenost

Obrázek 2.2: Geriatrické funkční vyšetření – Test mobility [7]

Některá vyšetření však také vyžadují obsluhu formuláře přímo pacientem. Například test hodin, který odhaluje velmi citlivě přítomnost kognitivních poruch. U něho je subjekt požádán, aby zakreslil ciferník hodin s ručičkami v určité poloze. Také u testu MMSE je zapotřebí interakce pacienta, jedná se o vyšetření paměti a mentálního stavu, kdy subjekt musí podle zapamatované předlohy se zavřenýma očima nakreslit různé obrazce. [5][7]

Z popisu a ukávek geriatrických funkčních vyšetření je tedy patrné, že jsou to specializované formuláře, od jednodušších po složitější, které ve velkém procentu využívají zaškrťovací pole k zaznamenávání výsledků a odpovědí pacientů. Každý formulář obsahuje rodné číslo pacienta, čímž ho jednoznačně identifikuje pro zpětné dohledání. V oboru gerontologie se, stejně jako v jiných oblastech zdravotnictví, pomalu začíná objevovat snaha o digitalizaci. V současné chvíli jsou veškeré formuláře podávány a vyplňovány papírově. Alespoň v cílovém zdravotnickém zařízení. To bude více objasněno v kapitole 2.2.

Pohled na funkční vyšetření z pohledu digitalizace

Protože jsou testy z velké části tvořeny právě zaškrťovacími políčky, dá se uvažovat o částečné automatizaci při digitalizaci. Tímto tématem se zabývala má bakalářská práce [7], ve které se věnuji návrhu a implementaci aplikace pro rozpoznávání a vyhodnocování dat z oskenovaných funkčních vyšetření. Bohužel však v této oblasti nebylo možné dosáhnout plné automatizace, právě kvůli některým vyšetřením, která jsou velmi unikátní a strojově velmi obtížně čitelná a vyžadují lidské oko pro vyhodnocení výsledku. Těmi jsou například

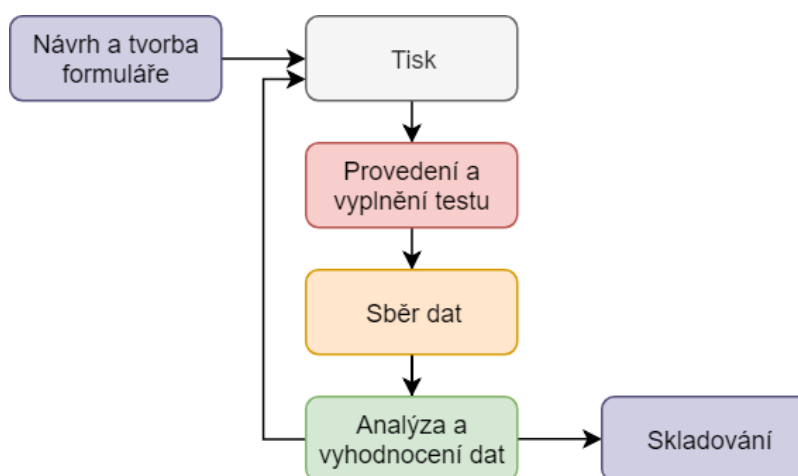
MMSE nebo test hodin, popsané výše. Další komplikací při digitalizaci jsou ručně psané texty, které je velmi náročné rozpoznávat a převádět, těch však není mnoho. Mezi ručně psané texty patří i číslice, jejichž strojové čtení by bylo velkým přínosem k automatizaci. Navzdory zmíněným překážkám se v reálné praxi při testování ukázalo, že ruční přepisování hodnot z checkboxů zabere při zpracovávání formulářů více než 75% veškerého času. I částečná automatizace výrazně pomáhá urychlit převod dat do digitální podoby. Nástroj na digitalizaci tedy v této oblasti má smysl. Po vytvoření nového řešení však nastal další problém, začlenění nástroje do celého procesu a životního cyklu formuláře, což je jedna z otázek, kterou se zabývá tato práce. Konkrétně v kapitole 3.2.

2.2 Aktuální stav práce s formuláři

Klinika, ve které bude platforma testována, Gerontologické centrum v Praze 8, úzce spolupracuje s Fakultou biomedicínského inženýrství ČVUT v Praze. Toto zdravotnické zařízení je velmi pokrokové, co se oblasti digitalizace týče a jeví již několik let snahy se v této problematice posunout kupředu. Výsledkem je projekt s názvem GDiag, který má za cíl celkově digitalizovat práci s funkčními vyšetřeními. Jedná se o soustavu různých nástrojů, které se snaží řešit jednotlivé části životního cyklu funkčního vyšetření. Bohužel však nejsou mezi sebou dostatečně sladěné. Navíc jsou poměrně složité na ovládání a používání těchto programů vyžaduje větší míru technických znalostí, což se ukázalo být hlavní komplikací pro zdravotnický personál. Další problém je, že byly vyvinuty jako desktopové aplikace a tím přichází komplikace s aktualizací systému a s jejich instalací na počítačích v gerontocentrech [7]. Využívání nástrojů s sebou tedy neslo řadu problémů, a tak vznikl požadavek na tvorbu nového komplexního nástroje, který by měl současné řešení nahradit, pokud možno i s využitím částí stávajících programů.

Je potřeba vše sladit a vytvořit unifikovanou platformu, s moderním a přívětivým uživatelským rozhraním, která bude všechny moduly a fáze funkčních vyšetření smysluplně propojovat. Zamyslet se nad vším jako nad celkem a ne pouze nad jednotlivými částmi a jejich úkoly. Aby bylo možné navrhnout nové a smysluplné řešení, musí být dobře zanalyzován aktuální stav práce s funkčními vyšetřeními v gerontologickém zařízení.

Životní cyklus funkčního vyšetření se dá rozdělit do několika fází. Ty jsou znázorněny na diagramu 2.3.



Obrázek 2.3: Fáze práce s funkčním vyšetřením

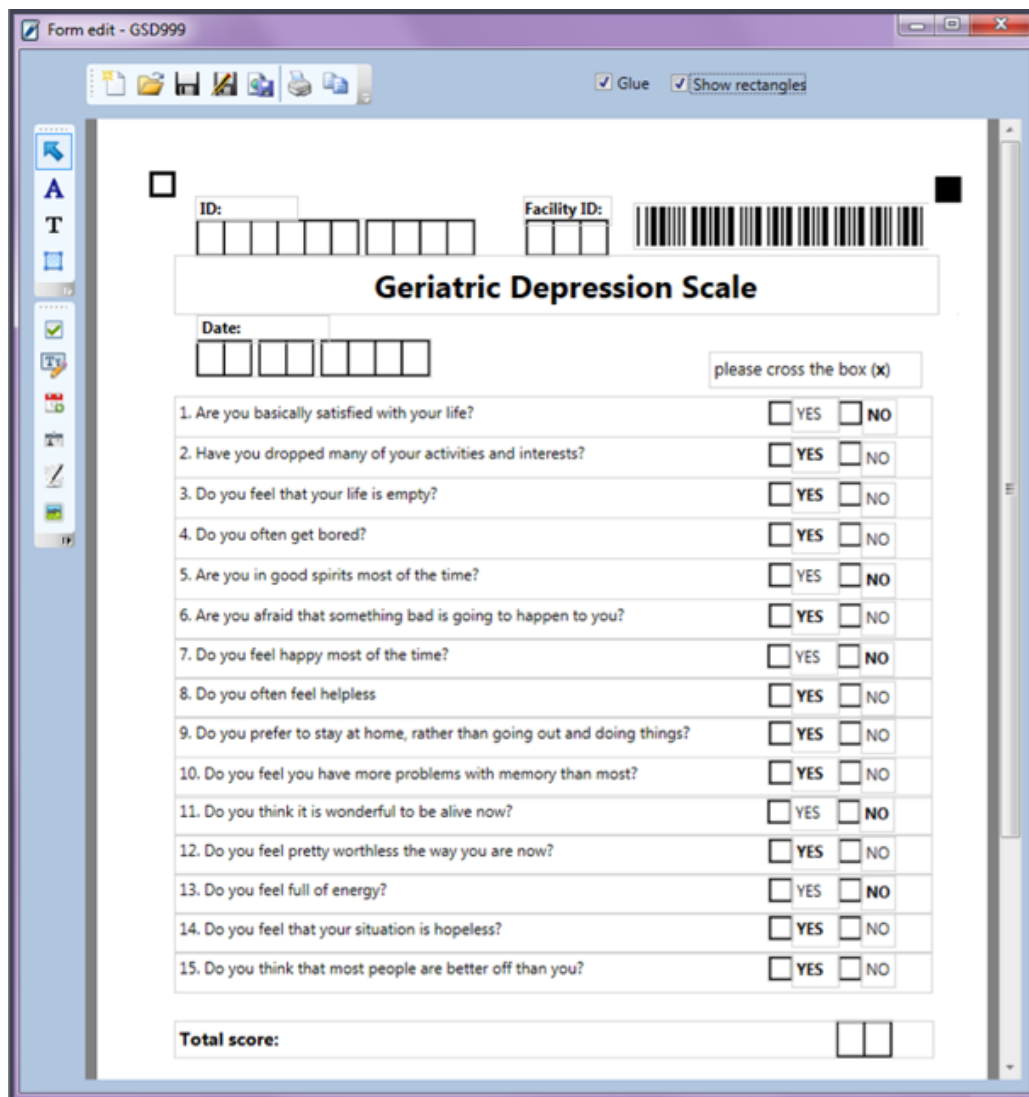
V rámci analýzy budou rozebrány jednotlivé etapy a jejich stav v současné chvíli. V závěru kapitoly bude provedeno celkové zhodnocení aktuální situace v gerontologickém centru.

Návrh a tvorba formuláře

Design a tvorba formuláře je první fází života funkčního vyšetření. Většina testů, které se používají, má jasně danou specifikaci toho, jak má vyšetření vypadat a co má obsahovat. To vše je stanoveno definicí konkrétního formuláře, který musí splňovat všechny náležitosti, aby fungoval spolehlivě a dalo se o toto vyšetření opřít při diagnóze, či vyhodnocování.

Způsobů, jak formulář vytvořit, existuje mnoho. Nejjednodušší je prostý zápis na papír, další možností je využít nástrojů pro malování nebo program Microsoft Excel. Existují také nástroje přímo pro tvorbu dotazníků a formulářů. Gerontologické centrum v Praze 8 se rozhodlo ve spolupráci s FBMI ČVUT v Praze vytvořit desktopový nástroj, specificky pro formuláře v gerontologii. Tento program se nazývá FormEdit. Obsahuje sadu prvků, kterými lze vytvořit test – pole pro datum, čárový kód, možnosti vložení textů, zaškrtačací pole, rámečky pro čísla, pro rodné číslo a řadu dalších. Prvky lze umisťovat do pracovní plochy dotazníku, která je vymezena čtyřmi čtverečky v rozích. Pro určení typu vyšetření lze na plochu také vložit čárový kód reprezentující název šablony. Tento nástroj je velmi komplexní, ale zatím nebyl nikdy používán přímo pracovníky zdravotnického zařízení, ale pouze proškolenými osobami z ČVUT. Pomocí tohoto nástroje byla vytvořena všechna současně používaná funkční vyšetření a vyexportována do PDF formátu a XML verze. Ta obsahuje seznam všech prvků a jejich vlastností, včetně přesného umístění na formuláři. Díky tomu lze dokument pouze pomocí tohoto seznamu znovu sestavit a upravovat.

V roce 2016 vznikla první snaha o přechod do webového prostředí a byl zahájen pokus o přepracování desktopové verze. Vznikla rozpracovaná část programu, kterou se nepodařilo dokončit. Zahrnovala ale tyto důležité vlastnosti – vykreslení XML šablony (pouze se zaškrtačovacími poli) v prohlížeči a možnost přidávat a přesouvat prvky na plátně. Zdrojový kód byl vytvořen pod licencí Unlicense, z důvodu navázání v dalším vývoji. Byl předán pro tuto diplomovou práci a je nutné implementovat funkce, které budou umožňovat vytvoření komplexního formuláře. Konkrétně dodat další prvky, jako vkládání textů, dat, čárových kódů, textových polí, umožnit generování PDF pro tisk, nabídnout podpůrné funkce k posouvání prvků při výstavbě dokumentu, nastavovat jejich vlastnosti a především vygenerovat výslednou šablonu, do které bude možné vkládat načtená data v další fázi práce v platformě. V kapitole 4 budou více popsány implementační a technické detaily. Při vytváření nových vyšetření se tedy stále používá původní verze. Aktuální nástroj pro tvorbu funkčních vyšetření je ukázán na obrázku č. 2.4.



Obrázek 2.4: Aktuálně používaná desktopová aplikace k návrhu a tvorbě papírových formulářů. Nástroj je naprogramovaný pro systém Windows.

Provedení funkčního vyšetření a vyplnění formuláře

Způsob provedení funkčního vyšetření je v režii ošetřujícího lékaře. Klíčové je, aby výstupem byl vždy vyplněný formulář. Od kvality vyplnění se přímo odvíjí chybovost při automatizovaném zpracování. Test může vyplnit doktor vlastní rukou během dotazování a úkolování pacienta. Případně delegovat tuto činnost na zdravotní sestru. Tento způsob produkuje poměrně kvalitně vyplněné výsledky. Ne všichni lékaři však u testů vyplňují data sami, mnohdy tuto práci přenechají samotnému pacientovi.

Kromě ručního vyplňování by šlo formuláře používat přímo na elektronickém zařízení. Například na tabletu nebo počítači. Tato varianta by ušetřila kroky skenování a zpracování dat z fotografie. Zatím tato metoda není lékaři preferována, ale zároveň dosud nebyla přímo dostupná. V cílové platformě by proto bylo vhodné tuto možnost zpřístupnit, seznámit s ní uživatele a během testování se zaměřit na její praktickou využitelnost. V době,

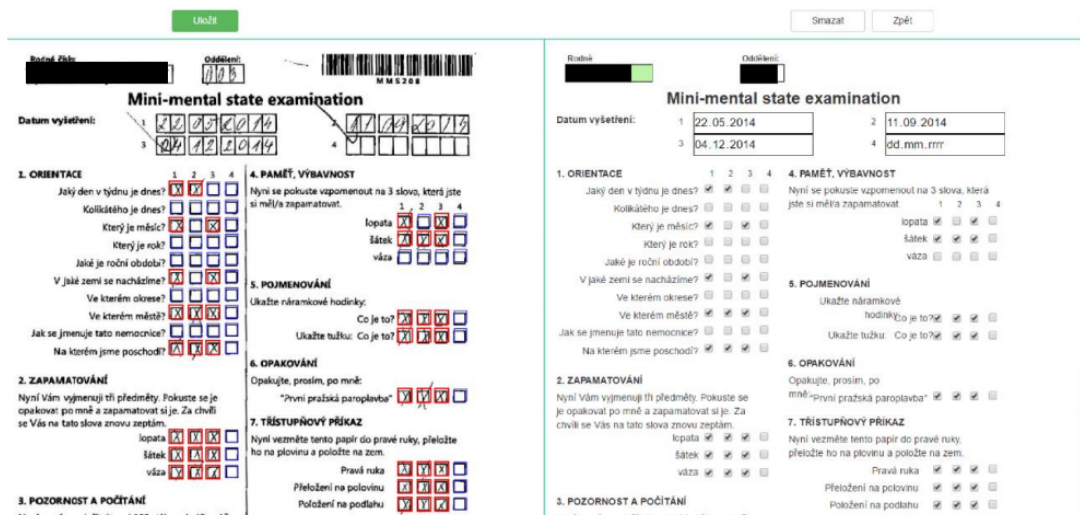
kdy se přenosná elektronická zařízení velmi rychle rozmáhají by tato funkcionalita neměla chybět.

Sběr dat

Sběr dat je z pohledu digitalizace nejnáročnější fází. Získat hodnoty z papírových dokumentů do elektronické podoby může být komplikované. Jak již bylo zmíněno výše, formuláře funkčních vyšetření jsou z velké části tvořeny zaškrťovacími poli. Jejich manuální přepisování trvalo přes 75% času při zpracovávání. Od tohoto způsobu již však Gerontologické centrum v Praze 8 upustilo. V současné chvíli jsou pracovníci ve zdravotnickém zařízení zvyklí skenovat vyplněné formuláře a automaticky je odesílat na FTP server. Zde k nim přistupuje nástroj pro digitalizaci oskenovaných formulářů.

Program pro zpracovávání formulářů se jmenuje FormScan. Je také součástí projektu GDiag. Tato desktopová aplikace velmi úzce spolupracuje s nástrojem pro tvorbu formulářů – FormEdit. XML výstup z editoru přesně specifikuje dané funkční vyšetření, FormScan je tak schopen určit polohu všech zaškrťovacích polí i ostatních prvků a zrekonstruovat formulář v editovatelné podobě – doplňovací šablonu. Při zpracovávání oskenovaného vyšetření nejprve dle čtverečků v rozích snímku ořízne a srovná pracovní plochu formuláře. Toto je zranitelné místo, jelikož se může stát, že pracovník provede skenování chybně a roh není nalezen. K tomu nástroj nabízí záchrannou možnost, označit rohy dokumentu ručně. Pokud ani to nepomůže, je třeba dokument ofotit znovu. V případě, že je plocha ořezána správně, je přečten čárový kód a k formuláři je přiřazena vhodná šablona. Následně jsou identifikovány checkboxy a jejich hodnoty přeneseny do doplňovací šablony. Všechna nalezená zaškrťovací pole jsou na původním snímku označena červeně a uživatel tak může zkontrolovat, zda byla nalezena všechna. Dále pracovník doplní číselné hodnoty a data. Rodné číslo je po vyplnění validováno vůči formátu, aby zde nevznikla zbytečná chyba při přepisu, jelikož je to klíčová hodnota. U dalších klíčových hodnot je kontrolována již pouze vyplněnost, pokud šablona nějaké obsahuje. Po potvrzení a uložení je vygenerován XML soubor s vyplněnými hodnotami a k tomu již ořezaný, upravený snímek.

Tímto způsobem tedy probíhá částečná automatizace při digitalizaci oskenovaných formulářů. Desktopovou aplikaci FormScan jsem v rámci své bakalářské práce optimalizoval a přeprogramoval do webové podoby. Avšak bude potřeba ji začlenit do platformy a rozšířit o integrace s dalšími částmi aplikace a předělat uživatelské rozhraní. Klíčové, pro správnou funkčnost celé aplikace, bude propojení s webovým designerem, které zatím chybí. Navíc modul není připraven pro export dat ve formátu vhodném pro databázi REDCap.



Obrázek 2.5: Webová verze nástroje FormScan pro digitalizaci papírových formulářů, vypracovaná v rámci mé bakalářské práce.

V současné chvíli je tedy stav digitalizace dat z papírových formulářů takový, že pracovníci zdravotnického zařízení jsou schopni sami pomocí nástroje FormScan digitalizovat oskenovaná funkční vyšetření a ukládat je na FTP server.

Analýza a vyhodnocování dat

Vyhodnocování dat je v oblasti gerontologie velice složité a vyžaduje vysokou expertízu. Často závisí na subjektivním názoru lékaře, tato fáze je tedy prováděna samotným ošetřujícím lékařem.

Ke generování statistických údajů by bylo potřeba pro každý typ formuláře zanalyzovat klíčové hodnoty a navrhnout možnost automatického zobrazování relevantních údajů. Zaškrtačací pole však nabízí potenciál pro různé statistiky a je to téma k probírání během schůzek a testování ve zdravotnickém zařízení.

Skladování

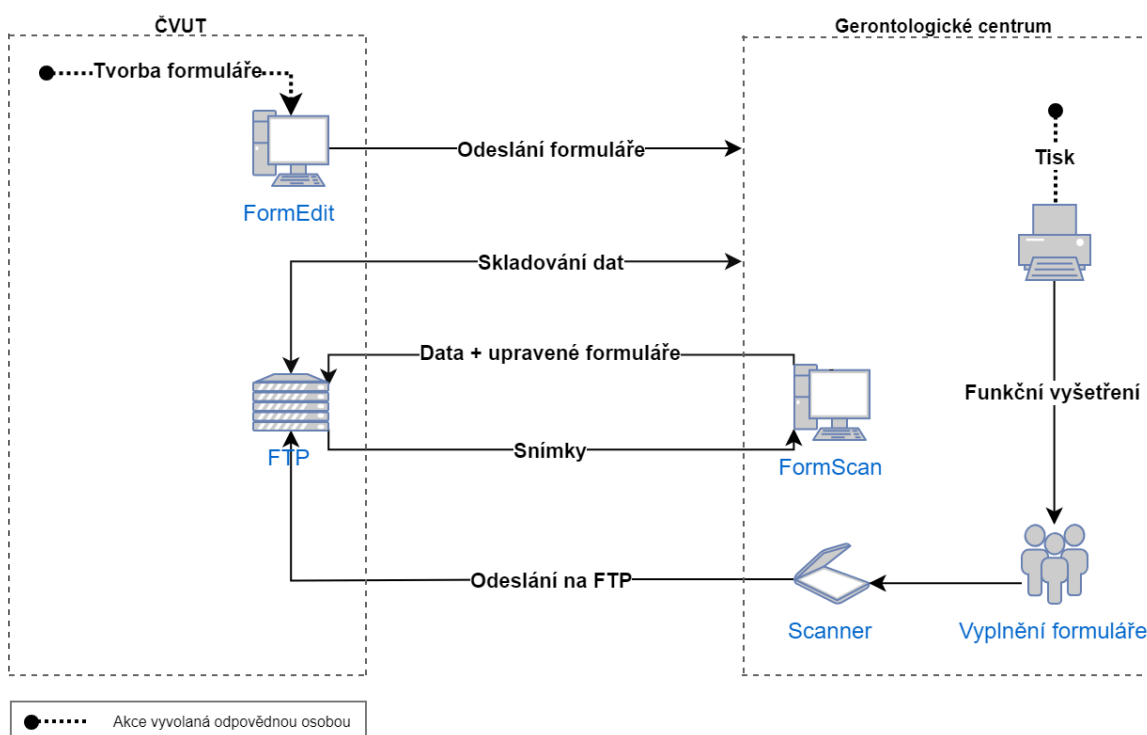
Není jednoduché skladovat velké množství formulářů. Některá gerontologická zařízení bohužel i v dnešní době skladují po vyhodnocení vyplněná vyšetření v archivech v papírové podobě. Což velmi znesnadňuje dostupnost testů pro případ potřeby do nich znovu nahlédnout a především sdílení dat s jinými pracovišti.

Gerontologické centrum v Praze 8 je díky částečné digitalizaci v této oblasti trochu pokročilejší. Funkční vyšetření rovnou skenuje a odesílá je na FTP server ČVUT, kde jsou dotazníky dále přebírány a zpracovávány nástrojem pro digitalizaci formulářů. Oskenované dokumenty jsou tedy vzdáleně přístupné. Na serveru se dá částečně orientovat dle názvu souboru a data vytvoření. Pro správu vyplněných vyšetření byl dokonce na FBMI Českého vysokého učení technického v Praze vyvinut i webový program. Měl za cíl poskytnout procházení digitalizovaných výsledků. Je to však další separátní aplikace, navíc je nutné do ní data ručně importovat a není to tedy optimální řešení. Program zůstal nepoužívaný a tak vznikl požadavek zakomponovat skladování oskenovaných funkčních vyšetření přímo do jednotné platformy, spolu s dalšími možnostmi filtrace a především s ostatními moduly

pro práci s vyšetřeními. Uživatel by tak mohl sledovat i historii daného funkčního vyšetření na stejném místě. Druhou možností je rozsáhlá formulářová databáze REDCap. Ideální by bylo poskytnout obě dvě možnosti.

Shrnutí současného stavu

Z analýzy jednotlivých fází práce s formulářem je patrné, že se nejedná o sehraný a unifikovaný proces. Etapy nejsou příliš sladěné a celý průběh vyžaduje nadbytečnou režii a asistenci univerzity. Největším problémem se zdá být fragmentace nástrojů a chybějící myšlenka spojující vše dohromady. Na druhou stranu je vidět snaha zdravotnického zařízení práci s formuláři celkově digitalizovat a také ochota přijmout nové řešení. Zároveň již učinilo řadu kroků k digitalizaci dat z papírové podoby. Aktuální stav práce s funkčními vyšetřeními v Gerontologickém centru v Praze 8 je znázorněn na obrázku č. 2.6.



Obrázek 2.6: Aktuální stav práce s formuláři v Gerontologickém centru v Praze 8

2.3 Konkurence v oblasti digitalizace práce s formuláři

Při mapování konkurence na trhu je potřeba vzít v úvahu, že digitalizace práce s formuláři není pouze o automatickém rozpoznávání zaškrtnutých polí. Tato technologie, OMR², případně doplněná o OCR³, vylepšuje fázi sběru dat, což je pouze jeden z mezikroků v celém procesu. Současně to je jedna z hlavních výhod. Musí ale být hodnoceny i další funkce a celkové zázemí pro práci s formuláři – od tvorby a zpracování, přes možnosti vyplňování, správu, návaznost fází, komplexitu a škálovatelnost celého řešení. Při mapování bude mimo jiné přihlíženo i k použitelnosti v cílovém zdravotnickém zařízení.

²Optical Mark Recognition

³Optical Character Recognition

Survio

Tato velmi propracovaná a především moderní platforma pro práci s dotazníky je asi největší potenciální konkurencí. Díky svému uživatelsky přívětivému rozhraní se stala velmi rychle populární, a hned v roce založení – 2012, získala první větší investici. Od té doby jeho uživatelská základna roste. Co Survio⁴ však dělá ještě větší konkurencí je fakt, že se jedná o firmu z České republiky. Může se tedy zdát, že cílí na stejné potřeby zákazníků a navíc působí na stejném trhu. Je zde však jeden zásadní rozdíl – aplikace umí pracovat pouze s webovými formuláři a koncepčně je podle toho navržena.

Tato služba nabízí při vytváření dotazníku řadu možností, zároveň poskytuje užitečné funkce při vyhodnocování dat a monitorování používanosti daného formuláře. Všechny nástroje a dotazníkové prvky jsou ale určeny pro vyplňování na webu a chybí tedy klíčové vlastnosti pro práci s papírovými formuláři, jako například čárový kód, či jiná identifikace dotazníku na papíře, nebo pole pro zakreslování libovolných obrazců. Nelze ani nahrát oskenovaný dokument, navíc při tvorbě a návrhu je uživatel velmi omezen možnostmi, jak prvky uspořádat. Řešení navrhované diplomové práci nabízí v tomto mnohem větší škálovatelnost. Survio navíc nedovoluje spravovat a rozdělovat dotazníky do složek a hierarchií, což je velmi podstatné pro organizaci práce.

Ačkoliv je tato aplikace velkou konkurencí v používání formulářů ve webovém prohlížeči a v této oblasti je důmyslně propracovaná, na poli práce s papírovými dotazníky a sbírání dat tímto způsobem, nepřináší žádné inovativní možnosti a je tedy nevhodná pro uživatele, kteří potřebují zvolit tuto cestu, tím pádem i pro Gerontologické centrum v Praze 8.

Survey Monkey

Jeden z nejpoužívanějších enterprise nástrojů pro práci s formuláři na světě – Survey Monkey⁵, je velmi podobná platforma jako Survio, s tím rozdílem, že cílí na velké podniky a instituce a poskytuje sady marketingových nástrojů. V této oblasti dominuje trhu a dostal se do podvědomí řady společností, v roce 2015 přesáhl počet uživatelů od založení 15 000 000. Tato platforma je ukázkou toho, že o takový software je zájem. V České republice existují dokonce státní instituce, které tento nástroj využívají.

I přesto, že cílí na kompletní přechod papírových dotazníků do digitální formy a má nemalou řadu nástrojů a integrací s jinými produkty, oproti platformě vytvářené v rámci této diplomové práce, má jednu velkou nevýhodu. Neumožňuje jinou možnost převádění papírových formulářů na digitální, než ruční přepisování. Na rozdíl od Survia však nabízí offline aplikaci, díky které je možné formuláře vyplňovat i bez připojení k internetu a později je hromadně nahrát. Nicméně pro potřeby zdravotnické kliniky a jiných zařízení, které potřebují využívat hromadné vyplňování papírových formulářů, je to nevhodný nástroj.

V oblasti digitálních formulářů a online nástrojů existuje ještě více úspěšných a propracovaných konkurenčních řešení. Mimo výše uvedené mezi největší z nich patří například Typeform, Qualtrics, Formstack, až po prostější řešení ve formě Google Forms, či Microsoft Forms. Z českých aplikací stojí za zmínku také Click4Survey nebo Netquest. Blíže byly sice rozebrány pouze dvě, avšak všechny zmíněné mají jedno společné, nenabízí automatizaci pro papírové formuláře a celkově tento způsob distribuce příliš nepodporují. To ponechává velkou výhodu a přináší inovaci na straně navrhovaného řešení.

⁴<https://survio.com>

⁵<https://surveymonkey.com>

Přestože práce s digitálními formuláři a cloudové řešení je jedním z cílů tohoto projektu, je potřeba analyzovat také konkurenci v oblasti digitalizace a práce s papírovými dotazníky a testy.

Remark Office OMR

Jak název napovídá, jedná se o OMR software, který převádí optické značky z papírových dokumentů do elektronické podoby k dalšímu zpracování⁶. Umožňuje také číst čárové kódy. Velkou výhodou je, že nabízí funkci rozpoznat strojově psaný text.

Tento nástroj bohužel existuje pouze v desktopové verzi. K tomu se vztahují i hardwarové požadavky na cílovém zařízení. Navíc je dostupný pouze pro Microsoft Windows. Způsob a nástroj použitý k vytváření dotazníků je na uživateli samotném, avšak vyhodnocovací mechanismus je i tak poměrně robustní a přízřivobitelný. Samotná data lze exportovat do dalšího produktu pro analýzu od tvůrčí společnosti a do dalších formátů, ale správu oskenovaných dokumentů a jejich ukládání online neposkytuje.

TestChecker

Tento software⁷ nesmí chybět v konkurenčních aplikacích, protože je momentálně nejpoužívanějším rozpoznávacím nástrojem v ČR. Stejně jako Remark, není konkrétně zaměřený a umožňuje čtení čárových kódů. Disponuje zajímavou funkcí, upozorněním na sporně vyplněné testy a vyzváním ke kontrole. Bohužel je program vytvořen pouze pro čtení polí a neumožňuje doplňovat jiné texty nebo hodnoty. Další nevýhodou je, že se rovněž jedná o desktopový program a licence se udělují na konkrétní počítač.[7]

Ani jeden z výše zmíněných nástrojů nenaplnuje požadavky zdravotnického zařízení. Jsou to programy určené výhradně pro rozpoznávání zaškrťávacích polí a neposkytují dostatečné zázemí pro práci s formuláři obecně. Podobný program byl již vyvinut v rámci projektu GDiag a ukázalo se, že k úplné digitalizaci nestačí.

Mimo český trh lze nalézt další OMR aplikace, například Verificare nebo FormScanner. Najít však reálně použitelný OMR software ve formě SaaS⁸ s cloudovým uložištěm se nepodařilo. V rámci analýzy se podařilo nalézt tuto online aplikaci.

- CloudOMR – Podle webových stránek nabízí možnost cloudového uložiště a vyhodnocování oskenovaných dokumentů v prohlížeči. Nepodařilo se však zaregistrovat a vyzkoušet produkt. Projekt nevypadá aktivní a dokumentace obsahuje pouze jednu kapitolu.

Z průzkumu aktuální situace na trhu vyplývá, že ačkoli produktů pro práci s formuláři je mnoho, žádný z nich však nespojuje dohromady cloudové zázemí pro práci s formuláři v prostředí prohlížeče spolu s OMR algoritmem pro vyhodnocování papírových dokumentů. Pro potřeby zdravotnického zařízení tedy nejsou vhodné.

⁶<https://acrea.cz/software/remark>

⁷<https://lightcomp.cz/tests-checker>

⁸Software as Service

Souhrnné srovnání hlavních konkurentů je v tabulce 2.1.

Název	Tvorba	Elektronické	Digitalizace (OMR)	Úložiště	SaaS
Survio	Ano	Ano	Ne	Ano	Ano
Survey Monkey	Ano	Ano	Ne	Ano	Ano
Remark Office OMR	Ne	Ne	Ano	Ne	Ne
TestChecker	Ano	Ne	Ano	Ne	Ne
Navrhované řešení	Ano	Ano	Ano	Ano	Ano

Tabulka 2.1: Srovnání konkurenčních nástrojů pro práci s formuláři.

Tato analýza přinesla důležité poznání, že největší inovací v navrhované platformě je právě OMR v prostředí internetu, ve formě SaaS. Při návrhu, vývoji i prezentování aplikace by na tuto vlastnost měl být kladen důraz.

Kapitola 3

Návrh nového řešení

Tato kapitola bude věnována návrhu aplikace. Soustředit se bude především na nové řešení pro zdravotnické zařízení. Platforma však má být do budoucna rozšiřitelná a zobecnitelná, je tedy potřeba vše, především návrh a proces práce s formulářem, navrhnout robustně a důsledně, aby byl zajištěn udržitelný rozvoj aplikace. Návrh musí vycházet také z analýzy požadavků, která bude provedena v úvodu této sekce.

3.1 Požadavky na nové řešení

Z analýzy současného stavu vyplynulo několik požadavků na aplikaci. Jejich výčet je uveden a rozebrán níže. Lze je rozdělit do dvou kategorií.

3.1.1 Požadavky ze strany zdravotnického zařízení

Tyto vlastnosti a funkce jsou spojeny přímo s potřebami zdravotnického zařízení a musí být splněny, aby bylo možné platformu začlenit do procesů v dané klinice.

Možnost navrhnout a vytvořit formulář v rámci aplikace

Uživatel musí mít možnost přímo v rámci platformy navrhnout a vytvořit formulář funkčního vyšetření, aby tato zodpovědnost mohla přejít na pracovníky zdravotnického zařízení. Nástroj by měl nabídnout možnost tisku, či exportu dotazníku do PDF, ale i digitální podobu ve formě online formuláře k editaci hodnot ve webovém prohlížeči.

Rozpoznávání zaškrťovacích polí

Pracovníci zařízení jsou zvyklí využívat nástroj pro podporu digitalizace, který rozpoznávání zaškrťovacích polí nabízí, nesmí tedy chybět ani v navrhovaném řešení. Musí splňovat stejné vlastnosti, jako již existující nástroj, kterými jsou především ořez a úprava oskenovaného dokumentu, OMR algoritmus, identifikace typu formuláře dle čárového kódu a podpora při přepisování ručně vyplněných údajů.

Ukládání v cloudu

Data a oskenované dokumenty musí být možné ukládat přímo v rámci platformy, protože rozdrobenost jednotlivých komponent procesu se ukázala být velmi důležitá. Jelikož je apli-

kace webová, nabízí se přímo cloudové řešení, kde budou všechny údaje dostupné online. Neměla by chybět základní možnost prohlížení digitalizovaných dokumentů.

Integrace s databází REDCap

Pro využití dat z funkčních vyšetření bude potřeba dodělat různé formy integrací s jinými systémy. V rámci této práce vznikl požadavek na přípravu na integraci s rozsáhlou formulářovou databází REDCap. Cílové zařízení disponuje programem pro import do tohoto systému. K datům z formulářů by tedy mělo být možné jednoduše přistoupit přímo z aplikace a přenést je do databáze REDCap.

Přihlašování a práce v rámci uživatelského účtu

Jelikož platforma bude webová a tedy přístupná přes internet, musí do ní být zabezpečen přístup autentizací a autorizací. V rámci svého účtu bude mít uživatel dle přidělené role možnost pracovat s formuláři a spravovat svůj prostor. Role uživatelů budou upřesněny během testování, pokud budou potřeba, v opačném případě až mimo tuto diplomovou práci, ale platforma pro to musí být připravena.

3.1.2 Další potřebné vlastnosti platformy

Aplikace musí splňovat i další požadavky, které přímo nesouvisí s potřebami zdravotnického zařízení, ale jsou to v podstatě nezbytné vlastnosti kvalitního programu, který má za cíl usnadnit a urychlit práci.

Uživatelská přívětivost

Pod tímto pojmem se neskrývá pouze příjemné a moderní uživatelské rozhraní (UI), ale také takzvané UX¹. Dobré UX znamená, že uživatel nebude muset kromě práce přemýšlet ještě nad tím, jak v systému něco udělat a kde to najít. Nebo alespoň co nejméně. Tento aspekt je v případě tohoto projektu velice důležitý, protože náročnost ovládání je jeden z hlavních důvodů, proč požadavek na nové řešení vznikl. Je tedy potřeba se nad vším důkladně zamyslet a navrhnout ovládací prvky intuitivně. I z tohoto důvodu budou probíhat uživatelská testování ve zdravotnickém zařízení, ale obsluha aplikace musí být již od začátku navržena pro co nejjednodušší užití. K tomu napomáhá i kvalitní UI, které s ovládáním úzce souvisí.

Vše v jednom řešení

Dalším požadavkem, který souvisí s UX, ale zároveň se týká celého procesu a míry využitelnosti platformy je, aby vzniklý software nebyl rozdrobený a obsahoval vše na jednom místě. Bez nutnosti aplikaci zbytečně opouštět a přesouvat práci a soustředění jinam. Toto je zároveň klíčový aspekt z pohledu zdravotnického zařízení, jedná se ale o celkově velmi důležitou vlastnost, proto je uvedena v této kategorii.

¹User experience

Moderní a standardní technologie

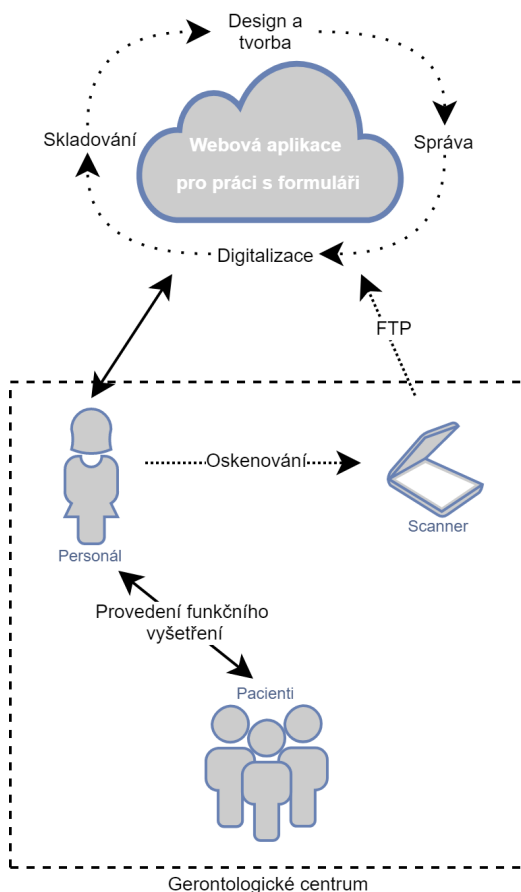
Technologie, které bude aplikace využívat musí být moderní, ale osvědčené a široce podporované, aby co nejdéle stačily rychle rozvíjející se oblasti webového vývoje.

3.2 Návrh systému a integrace do stávajících procesů

Analýze současného stavu procesů a práce s funkčními vyšetřeními v Gerontologickém centru v Praze 8 se podrobně věnuje kapitola 2. V rámci vysvětlení jednotlivých fází byly objasněny i problémy a slabá místa. Odstranění těchto problémů a způsob začlenění nového řešení do stávajících procesů je popsáno v této kapitole.

Jak již bylo řečeno, hlavním zjištěním z analýzy současného stavu je, že v současné době zdravotnické zařízení bojuje s rozdrobeností systémů a neunifikovaným řešením. Na obrázku 2.6 je graficky znázorněn proces práce s formuláři. Sjednocení do jedné platformy a centralizace všech nástrojů by měla přinést zlepšení. Začlenění do stávajících procesů by díky tomu znamenalo pouze odstranění nadbytečných kroků ze současné komplikované struktury. Adaptace na nový způsob se tím stává výrazně jednodušší.

Navrhovaný postup práce po jednotlivých fázích v novém řešení je znázorněn na obrázku 3.1 a popsán níže.



Obrázek 3.1: Proces práce s formuláři po začlenění navrhované platformy

1. Pracovník gerontologického centra se přihlásí pod uživatelským účtem do platformy na příslušné webové adrese. Ta je dostupná odkudkoliv a z libovolného zařízení, neohledně na operační systém. To je velká výhoda webového řešení. Po přihlášení do systému oprávněný uživatel vidí přehledovou nástěnku se základními údaji o formulářích. Spolu s navigací pro vstup do jiných částí aplikace tato stránka tvoří domovské zázemí pro uživatele a výchozí bod platformy.

V první fázi uživatel vstoupí do modulu pro tvorbu formulářů, kde pomocí webového nástroje vytvoří specializované geriatrické funkční vyšetření – takzvanou šablonu formuláře. Tu je možnost přímo v designeru vyexportovat do PDF a vytisknout nebo uložit v rámci platformy a později s ní pracovat online, či ji dále upravovat.

2. Po vytvoření a uložení šablony se uživatel vrátí zpět do správy šablon, kde má k dispozici vyhrazený prostor, ve kterém může hierarchicky uspořádat vytvořené šablony ve složkách a shlukovat si tak dotazníky dle typu, či projektu. I zde je možnost šablonu vytisknout pro vyplnění na papíře, případně ji kliknutím začít vyplňovat přímo v prohlížeči pomocí specializovaného nástroje.
3. Pokud uživatel zvolí vyplnění přímo v internetovém prohlížeči, načte se navržený formulář v digitální formě. Všechna pole jsou rozmístěna tak, jak byla vytvořena v designeru a odpovídají papírové podobě. Díky tomu může odpovědný pracovník zdravotnického zařízení s pacientem geriatrické funkční vyšetření provést i bez nutnosti zapisovat výsledky do papírového archu. Postačí k tomu počítač, či jiné zařízení s připojením k internetu. Tato funkcionality však z analýzy požadavků není prioritní, může být tedy dokončena, až budou plně funkční nezbytné vlastnosti.

Pokud je formulář vytištěn pro vyplnění na papíře, pokračuje pracovník nezměněným procesem, stejně jako dosud a během vyšetření zapisuje výsledky ručně pomocí tmavého pera či fixu do archu.

4. Po dokončení vyšetření je v případě papírového formuláře potřeba data přenést do digitální podoby. Pracovníci jsou zvyklí dokumenty skenovat a odesílat přímo na FTP server. Tato technika přenášení dat se osvědčila, nebude se tedy měnit metoda, ale místo serveru ČVUT se data budou přenášet přímo na server aplikace, kde se automaticky objeví v sekci dokumentů ke zpracování. Zde budou postupně odbavovány pracovníky zdravotnického zařízení pomocí funkcionality pro podporu digitalizace.

Uživatel tedy přímo v platformě vybere snímky, které chce digitalizovat a ty bude pomocí nástroje pro rozeznávání polí a úpravu snímku postupně zpracovávat. Data jsou přenášena do digitálního formuláře, stejně jako v případě online vyplňování. Uživatel už jen doplní ručně psané hodnoty a potvrdí správnost načtení zaškrtačkových polí.

5. Výstupem z geriatrického funkčního vyšetření je tedy elektronicky vyplněný formulář, k jehož datům lze snadno přistupovat a v případě, že byla data získána z tištěné podoby, tak navíc i upravený snímek papírového vyšetření.

Vše je k dispozici přímo v platformě, kde je možnost si výsledky organizovat po složkách a dále procházet. Mimo to lze data exportovat do databáze REDCap.

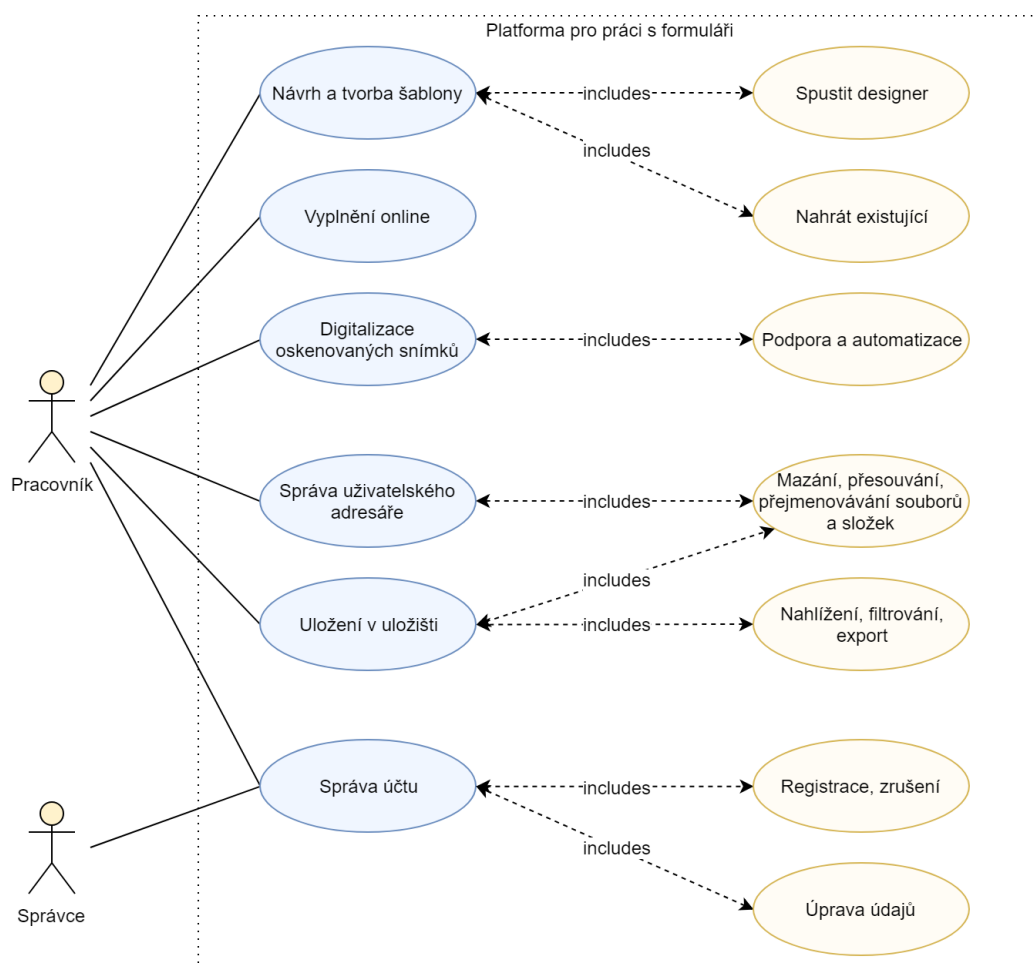
3.3 Návrh s využitím UML

Aplikace je navržena tak, aby splňovala všechny zmíněné požadavky a dala se snadno začlenit do výše popsaného procesu. Před samotnou implementací platformy, ať už uživatelského rozhraní, či architektury, je dobré se podívat na navrhovaný informační systém z více pohledů, zamyslet se nad nimi a některé hlavní z nich si vymodelovat. Tyto modely pomáhají si uvědomit, co vlastně bude programováno a mohou výrazně pomoci i ve fázi, kdy programátor přemýšlí o tom, jak provede implementaci.

Protože analýza při návrhu je velmi důležitá a potřebná, vznikl standardní modelovací jazyk – UML². Je nezávislý na programovacím jazyce a využívá objektů s pevně danou sémantikou, čímž jednoznačně a standardizovaně umožňuje popsat aplikaci. Díky tomu výrazně usnadňuje komunikaci a celkové porozumění dokumentace. [6] [9]

Níže budou pomocí UML diagramů popsány některé pohledy na navrhovaný informační systém, které mi přišly během návrhu nejdůležitější.

3.3.1 Diagram případů užití



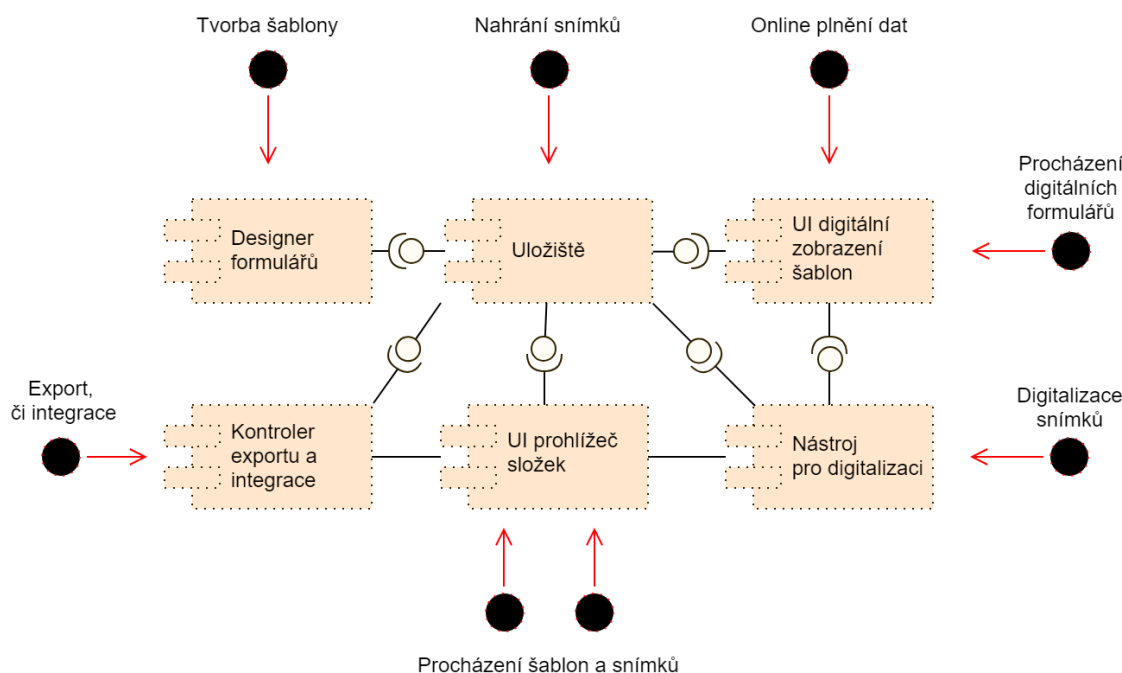
Obrázek 3.2: Diagram případů užití znázorňující interakci uživatele se systémem. Popisuje chování systému z vnějšího pohledu.

²Unified modeling language

Jedním z hlavních pohledů na informační systém při návrhu je, jak se v něm bude uživatel chovat a jaké procesy bude vyvolávat. Pro formální popis interakcí uživatele se systémem se využívá takzvaný „Use case diagram“. Ten se soustředí na chování systému z vnějšího pohledu. Popisuje funkce poskytované systémem, které přináší viditelný výsledek pro aktéra. Aktér může být libovolná entita interagující s programem, v tomto případě je to pracovník zdravotnického zařízení nebo administrátor. Diagram případů užití je zobrazen na obrázku číslo 3.2. [4]

3.3.2 Diagram komponent

Navrhovaný systém kombinuje více nástrojů, které jsou do jisté míry samostatné. Skládá se z pomyslných komponent. Ty spolu interagují, komunikují pomocí rozhraní a nadstaveb. Dohromady tvoří jeden funkční celek. Tato modulární struktura nabízí do budoucna snadné rozšíření, navíc rozhraní mezi nimi jsou škálovatelná. Strukturu a pohled na systém zevnitř bude nejlepší popsat nástrojem pro formální popis komponent softwaru – „UML component diagram“.



Obrázek 3.3: Diagram komponent [6], který popisuje interakci a propojení jednotlivých částí uvnitř systému.

Lze vidět, že jádrem platformy je uložisko. To interaguje se všemi ostatními moduly. Přímou do něho je možné nahrát oskenované formuláře, ostatní přístupy jsou pouze skrz rozhraní. Designer formulářů ukládá výslednou šablonu do uložiska ve formátu XML. K těmto souborům přistupují moduly pro digitální zobrazení šablon a nástroj pro digitalizaci. Přes rozhraní je vykreslují a zobrazují uživateli. Mimo ně k souborům přistupuje prohlížeč složek, který poskytuje nadstavbu pro tvorbu, správu a procházení hierarchické struktury uložiska. Posledním modulem je kontrolér pro export a integrace. Ten agreguje transformace dat k exportu do různých formátů.

Kapitola 4

Implementace

Při implementaci a výběru technologií použitých pro vývoj hrál největší roli fakt, že vyvíjený informační systém je webové řešení. Důležité bylo také přihlédnout k technologiím, jež byly použity v mé bakalářské práci [7], která tvoří jednu část celého systému, a protože cílem je vytvořit unifikovanou platformu, měl by i výsledný kód být jednotný.

Co se týká nasazení softwaru, bylo zvoleno SaaS¹ řešení, to znamená, že celá aplikace poběží na vlastním serveru a přístupná bude přes internet. Pro uživatele tím oproti klasickému on-premise řešení odpadá nutnost instalace na vlastní zařízení, celková závislost na vlastním HW a s tím spojené náklady. Pro správu a další vývoj systému je toto řešení také výhodné, díky své lepší udržitelnosti a jednotné verzi.

4.1 Metodika vývoje

Při implementaci byl zvolen agilní způsob vývoje. Vývoj probíhal ve dvoutýdenních iteracích. Na začátku každé iterace byly naplánovány cíle a úkoly pro dané dva týdny, na základě mé časové kapacity. Při plánování byl klíčový Backlog – prioritizovaný seznam úkolů k implementování. Do něho byly na začátku i v průběhu vývoje postupně přidávány nové úkoly, ale i nalezené chyby.

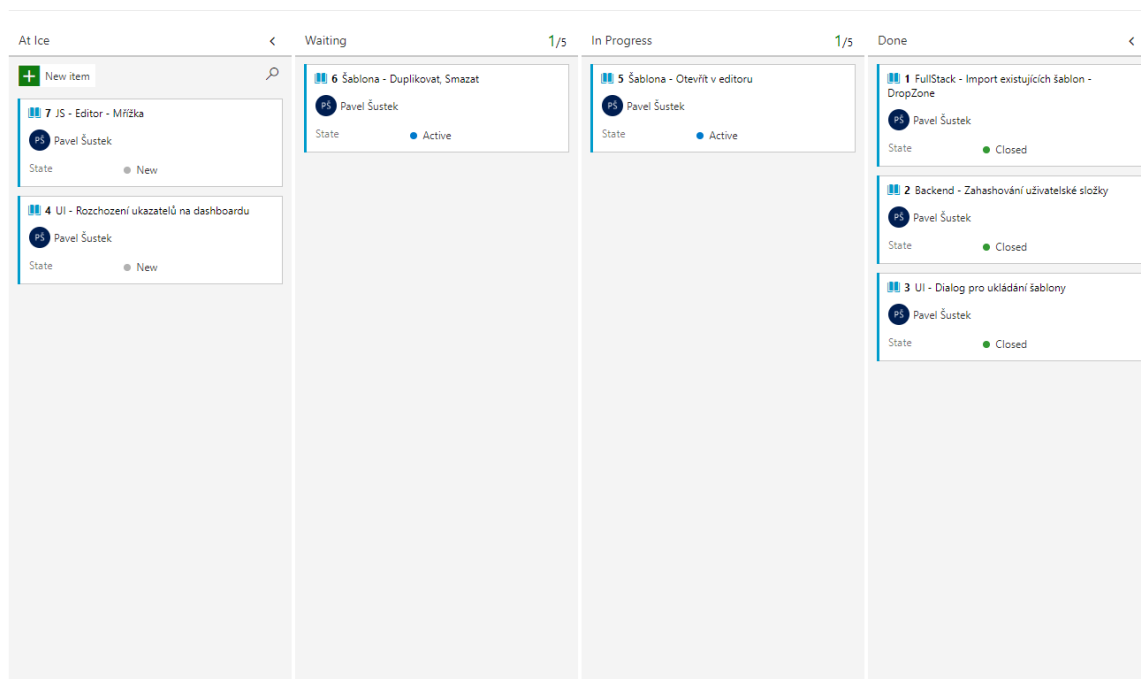
Vždy na začátku iterace byly z Backlogu vybrány úkoly, ideálně tak, aby byly odbaveny ty s největší prioritou. Vývoj probíhal však i do šířky, tedy zároveň byly implementovány úkoly z různých částí platformy.

Pro přehled v rámci daných dvou týdnů a organizaci práce sloužila takzvaná „Kanban tabule“. Ta se skládá z více částí – fází, které si definuje tým. V mém případě jsem si vydefinoval 4 části. První – At ice, na úkolu jsem ještě nezačal pracovat. Druhá Waiting, úkol už byl ve fázi In Progress, ale čeká na jinou událost, která nejde přímo vyřešit hned, je tedy v takovém režimu čekání, než se na něm bude pokračovat. Další fází je In Progress, na daném úkolu se pracuje. Poslední fáze, kterou jsem si definoval je Done. Úkol byl dokončen. V případě přesunutí do poslední fáze, jsem vždy vyplnil čas strávený na tomto úkolu a také propojení s příslušnými verzemi kódu. Tím je možné práci zpětně velmi dobře monitorovat a navracet se ke změnám a zjistit, v rámci jakého úkolu byly zpracovány. Díky evidenci času lze odhadnout i časovou náročnost dalších úkolů a projektu obecně. Na konci iterace byly vyhodnoceny výsledky a naplánovány další úkoly a cíle.

Díky agilnímu vývoji není potřeba tak podrobná technická specifikace, ještě před tvorbou samotného produktu. Naopak umožňuje rychleji reagovat na změny a více se adaptovat

¹Software as Service

měnícím se požadavkům, které se mohou objevit až v průběhu vývoje. Kanban tabule a prostředí Azure DevOps obecně byly zároveň velmi motivující. Na obrázku 4.1 je ukázána tabule uprostřed jedné z iterací. [1]



Obrázek 4.1: Kanban tabule v průběhu iterace

4.2 Uživatelské rozhraní a popis aplikace

Při návrhu UI² byly pro uživatelskou přívětivost dodrženy zavedené standardy u webových aplikací. Bylo zvoleno klasické postranní hierarchické menu, ze kterého se dá navigovat do jednotlivých částí aplikace. Od tohoto rozvržení stránky se odvíjel návrh všech ostatních částí platformy, který se soustředí nejen na přehledné prostředí, ale i efektivní využití prostoru.

Webová stránka proto vyplňuje celou šířku okna internetového prohlížeče, toto je důležité, jelikož v některých částech aplikace, jako například v designeru nebo digitalizačním nástroji, je potřeba efektivně využít maximální plochu obrazovky.

Na uživatelské rozhraní je kladen velký důraz během celé implementace a proto k němu již od začátku bylo přistupováno pečlivě. Cílem bylo vytvořit na pohled profesionální, seriózní, ale zároveň moderní, jednoduché a přívětivé prostředí. Celá aplikace má působit svižným a responzivním dojmem, obsahuje řadu animací postavených na CSS nebo JavaScript, které mají uživateli dodat pocit moderního UI. K jednodušším animacím je využit CSS framework `Animate.css`, díky kterému je o něco méně zatěžován webový prohlížeč.

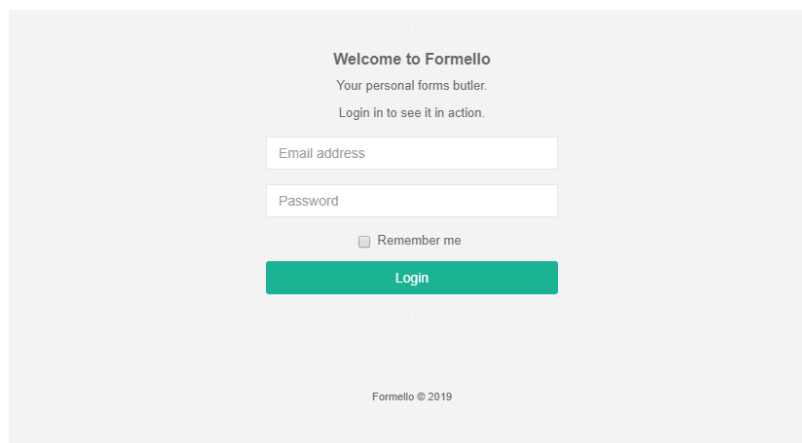
Při implementaci UI byla využita zakoupená šablona v jazyce HTML a CSS, která disponuje sadou předem připravených komponent, jež lze využít, či upravit. To usnadnilo mnoho práce a především snížilo případnou chybovost. Navíc komponenty jsou naprogramovány

²User Interface - Uživatelské rozhraní

tak, aby byly plně responzivní pro každé zařízení, což je dnes požadovaným standardem kvalitních webových aplikací.

4.2.1 Přihlašovací stránka a domovská nástěnka

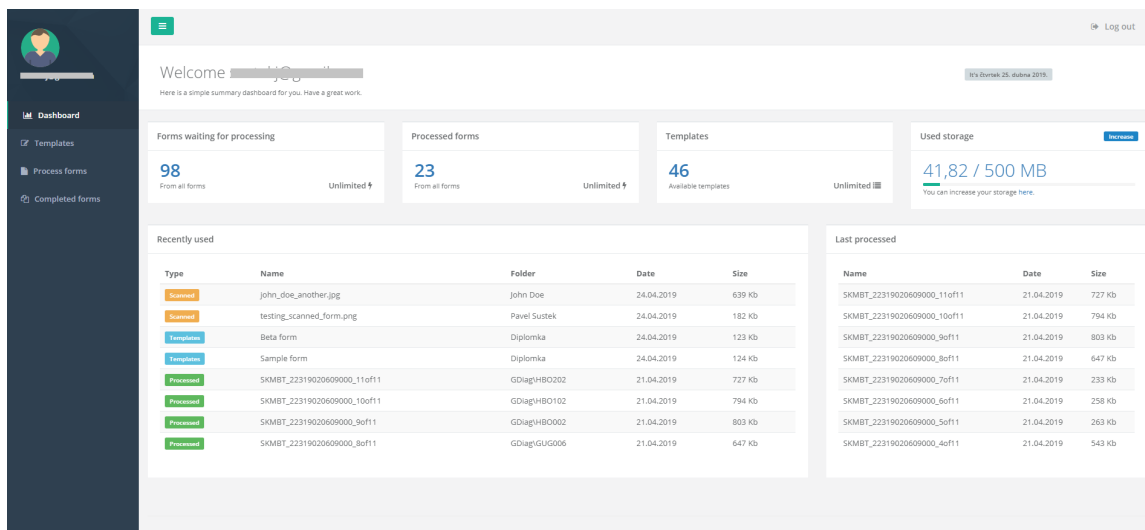
Vstup do platformy je možný pouze s uživatelským účtem. Přihlašovací stránka je velice jednoduchá a obsahuje pouze formulář pro zadání údajů k autentifikaci. Jako jediná ze všech podstránek nesdílí společné rozvržení v celé aplikaci a tedy se odlišuje od zbytku programu. Zároveň je to první pohled, který uživatel vidí při vstupu na internetovou adresu platformy.



Obrázek 4.2: Stránka pro přihlášení do aplikace

Po příchodu do aplikace je uživatel nasměrován na přehledovou nástěnku. Zde jsou rozmístěny různé užitečné widgety³, které dávají rychlý přehled o nejdůležitějších ukazatelích. Tato domovská stránka obsahuje 6 widgetů. První čtyři ukazují vždy jeden údaj, konkrétně počet formulářů čekajících na digitalizaci, počet zpracovaných formulářů a uložených v systému, počet předdefinovaných šablon a velikost využití dostupné paměti úložiště. Zbývající widgety jsou tabulky, jedna z nich ukazuje naposledy použité soubory s informací o názvu, data a typu souboru. Druhá tabulka slouží jako výčet naposledy digitalizovaných dokumentů. Do budoucna by tato nástěnka měla být plně konfigurovatelná. Výše popsanou uvítací obrazovku lze vidět na obrázku č. 4.3.

³Vizuální interaktivní prvek



Obrázek 4.3: Domovská nástěnka s navigačním menu po levé straně a praktickými widgety na jednom místě

4.2.2 Navigační menu a akční panel

Základem ovládání celé webové stránky je standartní postranní menu a na podstránkách poté akční panel s tlačítky, umístěný v horní části. Tento panel je v celé aplikaci umístěn jednotně ve stejném místě, aby uživatel přesně věděl, kde hledat akční tlačítka. Tyto dvě komponenty jsou, kromě přihlašovací stránky, přítomné ve všech částech platformy. Nechybí ani tlačítko pro odhlášení, které je umístěno vpravo nahoře, jak je zvykem, případně se lze odhlásit po kliknutí na uživatelské jméno.

Dalším prvkem pro rychlou orientaci v platformě je drobečková navigace. Ta napovídá, v jaké části aplikace se uživatel zrovna nachází.

Postranní menu se v případě přechodu do některých podstránek v platformě, přepne do pracovního režimu. Jedná se o minimalizovanou verzi menu skládající se pouze z ikon a možnosti expandování. Tato funkce je užitečná především u částí, kde je vyžadována velká pracovní plocha, například v designeru formulářů nebo v digitalizačním nástroji. Menu lze minimalizovat i ručně, kliknutím na tlačítko obsahující tři vodorovné příčky pod sebou. Pracovní režim je vidět na obrázku 4.7.

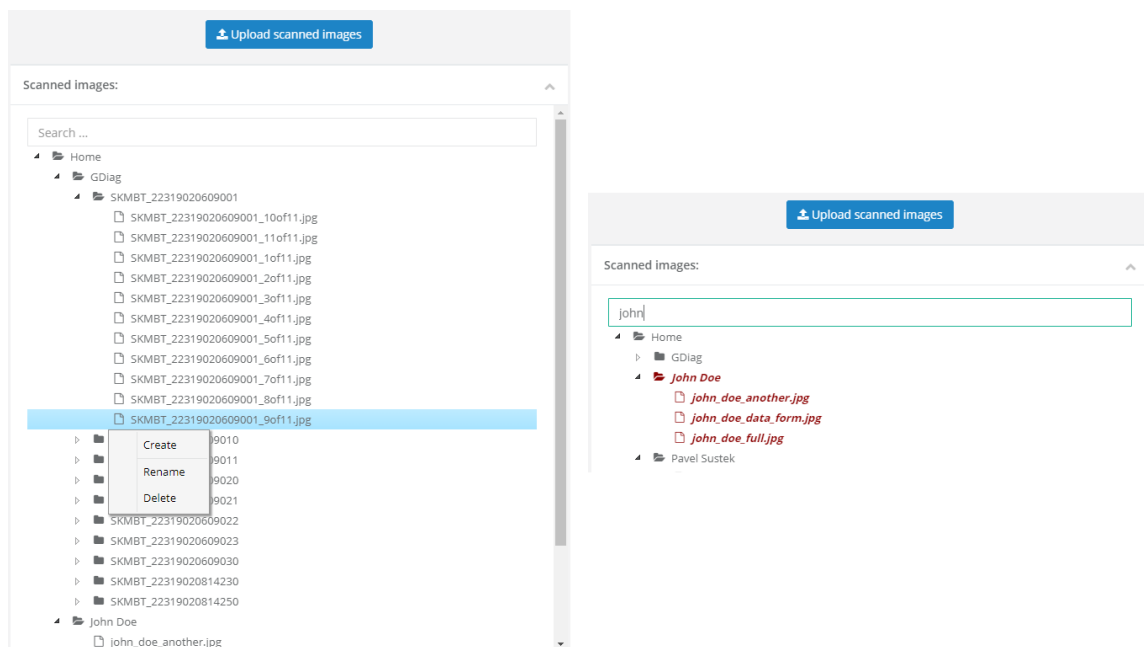
4.2.3 Způsob manipulace s dokumenty

Každý uživatel má k dispozici vlastní prostor, ve kterém může různorodě pracovat se svými dokumenty. Pro snadnou správu šablon i formulářů platforma nabízí interaktivní průzkumník, ve kterém je možné libovolně spravovat nahrané nebo vytvořené soubory. Tento integrovaný nástroj pro správu následuje funkcionalitami osvědčené standardní desktopové nástroje pro management složek a souborů s důrazem na snadnou práci.

Pomocí kontextového menu, které lze otevřít pravým tlačítkem myši, je možné vykonávat operace v domovském adresáři. Uživatel si zde může vytvořit libovolnou složku, přejmenovávat soubory a také jednoduše dokumenty mazat. Velmi užitečnou funkcionalitou je možnost přesouvání objektů mezi složkami pomocí přetažení myší. Lze také smazat celou složku najednou. Všechny změny se okamžitě promítnou v uživatelském prostoru. Detaily ohledně implementace tohoto nástroje jsou popsány v kapitole 4.6.

V adresářové struktuře je také možné vyhledávat. Průzkumník při hledání označuje nalezené soubory již během psaní, od třetího znaku a otevírá adresářový strom v příslušných místech.

Tento ovládací prvek je přítomný ve všech hlavních částech aplikace – Správa šablon, Dokumenty ke zpracování, Zpracované formuláře a je ukázán na obrázku 4.4.



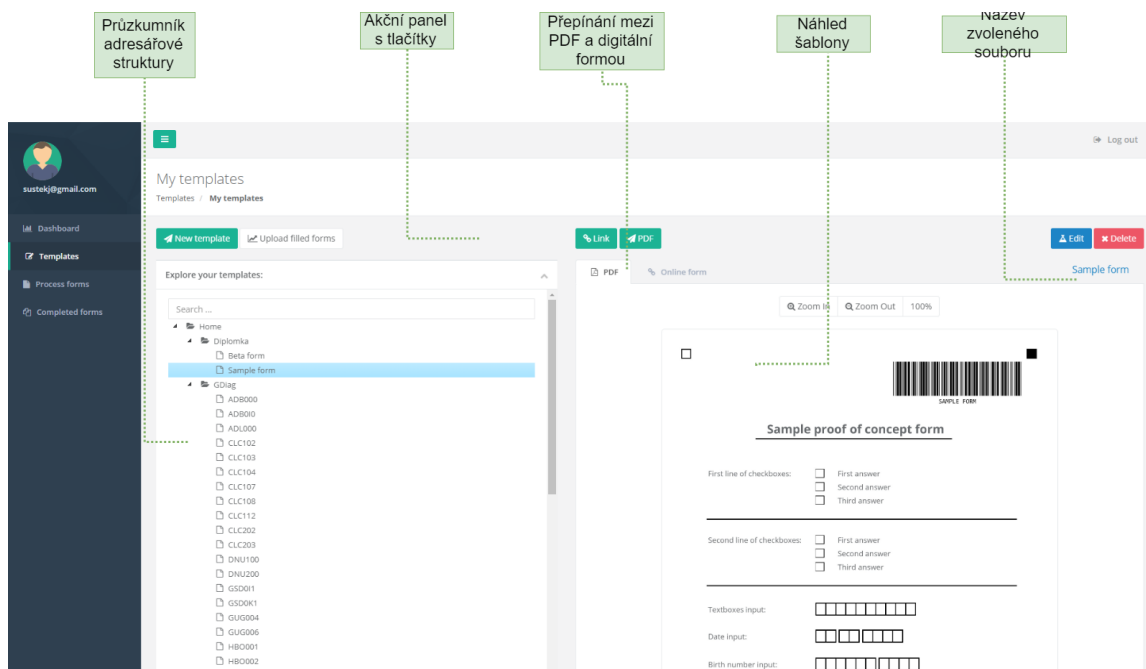
Obrázek 4.4: Nástroj pro manipulaci s dokumenty a složkami. Na obrázku vlevo je otevřeno kontextové menu. Vpravo lze vidět způsob vyhledávání v průzkumníku.

4.2.4 Práce s šablonami a designer formulářů

Práce s šablonami se skládá ze dvou částí, první je stránka pro správu, odkud lze vytvářet nové nebo upravovat existující. Tvoří ji tři hlavní sekce, průzkumník adresářové struktury, náhled vybrané šablony/formuláře a akční panel. První dvě části byly rozebrány výše – Průzkumník, ten slouží uživateli k manipulaci se složkami a zobrazování jejich obsahu. A náhled, který se zobrazí při výběru konkrétní šablony se na pravé straně.

Tlačítka pro vyvolání akcí jsou umístěna v akčním panelu. Odtud je možné přejít k vytváření nové šablony nebo do sekce k nahrávání již vyplněných šablon. Při výběru konkrétního objektu se objeví další tlačítka – pro export šablony do PDF k tisku, úpravě stávající nebo k jejímu smazání.

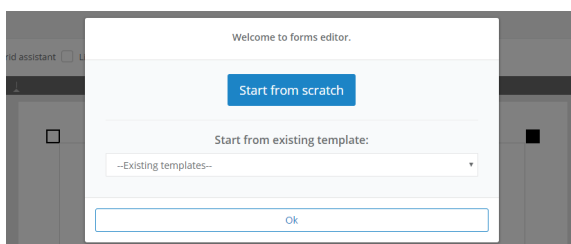
Stránka pro správu šablon je vidět na obrázku č. 4.5.



Obrázek 4.5: Stránka pro správu šablon

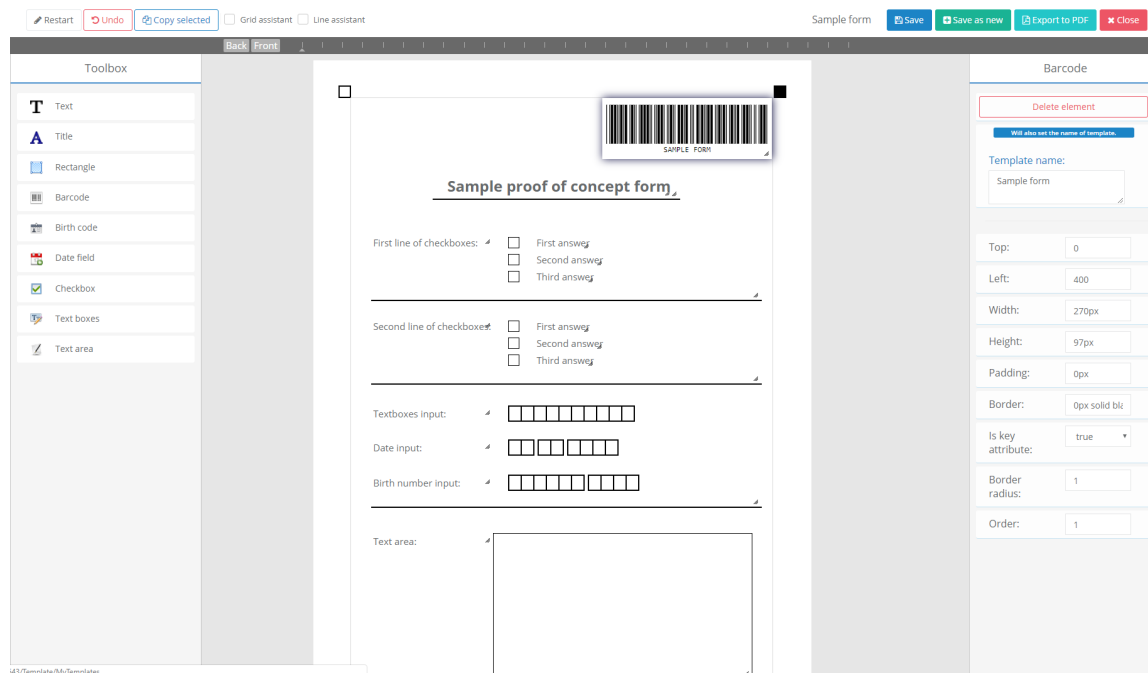
Vytvořit přehledné a intuitivní uživatelské rozhraní pro návrhář formulářů bylo velice důležité, jelikož je to nejkomplicovanější část na ovládání a zároveň stěžejní krok před dalšími fázemi. Od vytvořené šablony se vlastně odvíjí celá další práce a je proto potřeba, aby uživatel nebyl odrazen již v tomto kroku. Velkou roli zde tedy hrála UX analýza, aby uživatel měl s designerem příjemnou práci. Výsledkem je níže popsáný nástroj.

Při vstupu do návrháře je uživatel dotázán, zda chce navázat na některou z již vytvořených šablon, nebo začít na prázdné ploše. I vytvořený formulář lze tedy využít jako základ pro další.



Obrázek 4.6: Designer formulářů - Před zahájením práce lze vybrat, zda uživatel začne na prázdné ploše nebo využije existující šablonu

Hlavní oblastí designeru je plocha pro vkládání prvků. Ta se nachází uprostřed a je ohraničena čtyřmi čtverečky. Ty lemují efektivní část formuláře, která bude vyhodnocována při digitalizaci, mimo tuto oblast nelze vkládat objekty. Světlý čtverec určuje levý roh dokumentu. Podle těchto čtyř ukazatelů je později také vyrovnáván deformovaný obraz ze skeneru. Dalšími oblastmi jsou akční panel, inventář prvků, panel s vlastnostmi vybraného prvku a ukazatel vrstev formuláře. Designer je zobrazen na obrázku 4.7 a popsán níže.



Obrázek 4.7: Designer formulářů - Uprostřed se nachází plocha formuláře ohraničená čtverci, po levé straně je inventář formulářových komponent, vybraná je komponenta Barcode, napravo jsou vidět její vlastnosti. Nad plochou šablony je akční panel a pod ním ukazatel aktuální vrstvy, ve které se nachází zvolený prvek.

Akční panel obsahuje tlačítka, vykonávající speciální akce.

- Undo – Navrátí naposledy provedenou akci.
- Copy selected – V návrháři je možné vybrat jeden nebo více souborů a pomocí tohoto tlačítka je kopírovat, stejný prvek tedy není nutné definovat vícekrát. Označení prvku pro kopírování nebo hromadné zpracování se provádí stiskem klávesy CTRL a vybráním prvku nebo prvků. Vybrané prvky lze také hromadně přesouvat nebo upravovat jejich společné vlastnosti.
- Grid assistant – Při zaškrtnutí tohoto pole se uživateli při pohybu prvků po ploše začnou objevovat pomocné linky a objekt je přichycován a sdružován s jinými objekty tak, aby šlo vytvořit souměrné rozložení. Lze tak například poskládat prvky do přímký vedle sebe nebo pod sebe.
- Line assistant – Pokud k přesnému rozložení nestačí Grid assistant, je možné využít pomocné mřížky. Na tuto mřížku lze vkládat objekty podle předdefinovaných linií. Prvky se rovněž samy přichytávají a usnadňují tak výstavbu šablony. Nelze využívat obě metody zároveň.
- Save as new – Toto tlačítko uloží šablonu do systému, kde bude dále dostupná ve správě šablon. Před provedením akce je uživatel dotázán na název šablony. Výjimku tvoří případ, kdy je na formuláři přítomen čárový kód, potom je název předvyplněn právě jeho hodnotou.

- Save – Tato volba je dostupná pouze pokud je upravována existující šablona. Akce přepíše předchozí verzi a je zachována pouze novější.
- Export to PDF – Možnost exportovat šablonu do PDF přímo z návrháře. Lze tak učinit i ve správě šablon.
- Restart – Vyvolá zahození veškerých změn a spustí designer znovu.
- Close – Návrat zpět do správy šablon.

Inventář prvků je umístěn nalevo od plochy formuláře. K vložení objektu na pracovní plochu stačí vybrat libovolný prvek kliknutím levého tlačítka myši, přejet kurzorem kamkoliv na plochu šablony a znovu kliknout myši. Takto lze postupně vytvořit celou šablonu. K jednoduššímu vkládání poslouží funkcionality Grid assistant a Line assistant, popsané výše. Po vložení se prvek objeví na ploše a po levé straně od šablony jeho možné nastavení a vlastnosti. Lze se k němu kdykoliv vrátit a upravit, či přesunout jej.

Při pohybu objektů po ploše je nutné brát ohled na pořadí vložení. Prvek, který je vložen dříve leží v nižší vrstvě, než později přidaný a může se stát, že bude překryt. Každý prvek lze přesunout do vyšší, či nižší vrstvy pomocí tlačítek Front nebo Back, případně nastavení Z-index vlastnosti. Přesnou aktuální vrstvu je možno vidět na ukazateli vrstev formuláře, kde je znázorněn aktuálně vybraný prvek malou šipkou.

Designer v současné chvíli nabízí tyto formulářové prvky:

- Text – Tento prvek slouží k vložení textu do formuláře.
- Title – Představuje tučný text s větší velikostí písma, jedná se o předdefinovaný a doporučený nadpis, který však lze libovolně měnit.
- Rectangle – Nástroj pro tvorbu ohraničení, obdélníkových objektů, ale i například přímek, tedy podtržení, rozdělení plochy.
- Barcode – Velmi důležitou komponentou je čárový kód. Není přímo povinný na formuláři, avšak velice usnadňuje digitalizaci a urychluje celý proces, takže jeho použití je doporučeno. Jednoznačně totiž identifikuje šablonu a během zpracování dokumentu systém pozná, jakou šablonu k oskenovanému snímku přiřadit. Název šablony musí být unikátní, což je kontrolováno při ukládání, nelze tedy uložit dvě se stejným jménem. Pokud uživatel vloží tento prvek, v jeho vlastnostech je speciální pole – Template name. Psaním do něho se přímo generuje výsledný čárový kód. Tento prvek je také unikátní tím, že lze vložit na formulář pouze jeden.
- Birth code – Pole pro vepsání rodného čísla v českém formátu, speciálně pro účely zdravotnického zařízení Gerontologického centra v Praze 8. Ve webové podobě potom odpovídá textovému poli. Při digitalizaci je ověřována validnost vyplněné hodnoty.
- Date field – Políčka uspořádaná ve formátu pro ruční vepsání data. V elektronické formě je potom tento prvek reprezentován jako datumové pole s kalendářem.
- Checkbox – Velmi důležitým prvkem je zaškrtačací pole. Platforma pro tento prvek totiž poskytuje plnou automatizaci při digitalizaci. A to formou rozpoznávání hodnoty uvnitř pole a vyhodnocením výsledku. Při používání tohoto prvku je výhodné použít funkcionality kopírování komponent, například vytvořením skupiny čtyř polí a následným rozkopírováním, lze snadno a rychle vytvořit test se čtyřmi možnostmi odpovědi pro každou otázku.

- Text boxes – Tato komponenta slouží především k lepšímu zobrazení textového pole v PDF podobě. Pole pro písmena nebo čísla o zadaném počtu políček. Do budoucna je zde potenciál pro rozpoznávání písmen psaných hůlkovým písmem a číslic. V elektronické podobě se jedná o prostý textový vstup.
- Text area – Vytvoří velké textové pole, které je v elektronické podobě zobrazeno jako víceřádkové textové pole.

Při výběru libovolného prvku se zobrazí na pravé straně seznam vlastností prvku, kde je možné ručně upravovat vzhled dané komponenty. Zde je seznam vlastností, které lze upravit.

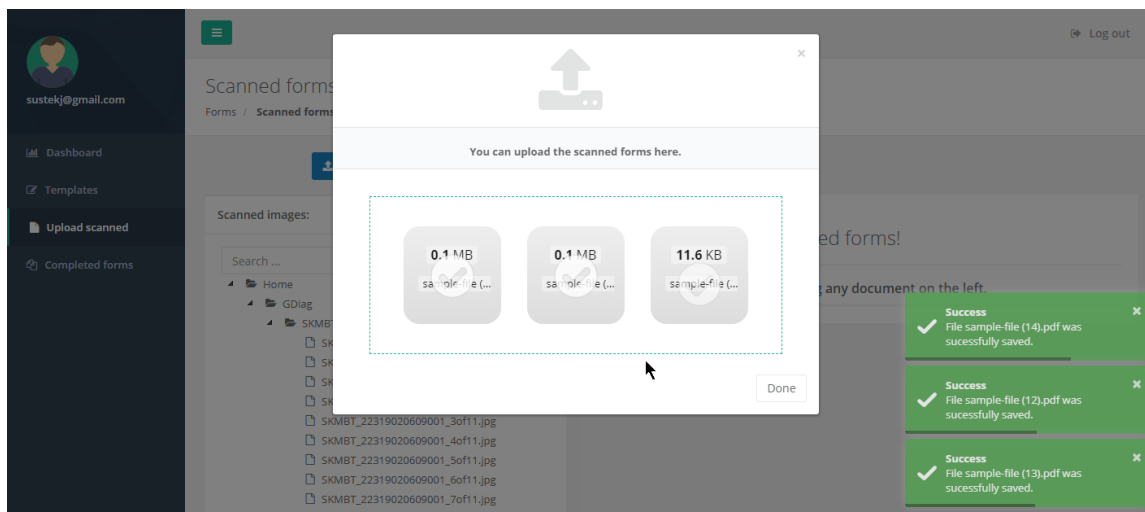
- Text – Pokud se jedná o prvek, který obsahuje text, tedy Title a Text, lze pomocí této vlastnosti tvořit jeho obsah. Zahrnuje také rozšířený textový WYSIWYG editor, díky kterému je možné velmi snadno tvořit formátovaný text.
- Top – Pozice prvku od horní hrany plochy v pixelech.
- Left – Umístění komponenty od levé hrany plochy v pixelech. Vlastnosti Top a Left se dynamicky mění s posouváním komponenty po ploše, je však možné je měnit přesněji pomocí přesného zadání.
- Width a Height – Těmito vlastnostmi lze určit rozměry daného prvku. U některých je tato možnost vypnuta, například u zaškrťovacího pole.
- Padding – Určuje odsazení uvnitř prvku.
- Border – Nastavení ohraničení. Lze šířku ohraničení komponenty, barvu a typ.
- Border radius – Úzce souvisí s ohraničením, udává zaoblení jeho rohů.
- Order – Číslo udávající vrstvu, ve které se nachází prvek. Číslo jedna znamená nejnižší vrstvu.

4.2.5 Digitalizační nástroj

Po vytvoření formuláře a následném vyplnění na papíře, je potřeba jej opět dostat do digitální podoby. K tomu v platformě slouží digitalizační proces, zahrnující částečnou automatizaci a pomocné funkcionality.

Stejně jako v případě šablon, i zde je výchozím bodem stránka pro správu, kde je možné nahrávat nové oskenované dokumenty nebo spravovat a prohlížet stávající. Vzhled pro jednoduchou orientaci zůstal stejný jako u šablon. To znamená, že vlevo se nachází průzkumník adresářové struktury a vpravo náhled vybraného nahraného dokumentu pro digitalizaci.

Do systému lze nahrávat soubory ve formátu JPEG, PDF, TIF a PNG. Je umožněno vložit i více souborů najednou. Soubory ve formátu TIF mohou obsahovat více snímků v jednom souboru. Do tohoto formátu exportuje řada skenerů a slouží pro urychlení hromadného skenování dokumentů. Takové soubory jsou rovněž platformou podporovány a po nahrání jsou rozděleny na jednotlivé snímky označené pořadovým číslem. Ty jsou potom vloženy do složky s názvem původního souboru.

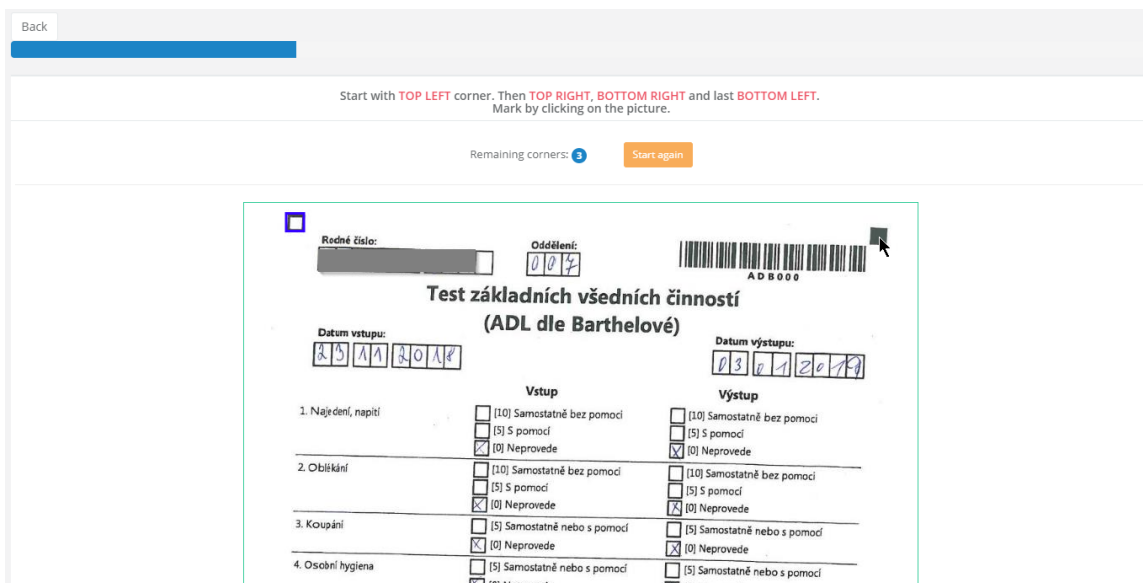


Obrázek 4.8: Hromadné nahrávání oskenovaných souborů. Zde je snímek aplikace ihned po nahrání tří PDF dokumentů.

Po stisku tlačítka pro zahájení digitalizace následuje několik kroků. Nejprve je oskenovaný snímek upraven, ořezán podle rohů ohraničujících plochu formuláře a vyrovnán. V prvním fázi dochází také k rozpoznávání čárového kódu a přiřazení příslušné elektronické šablony.

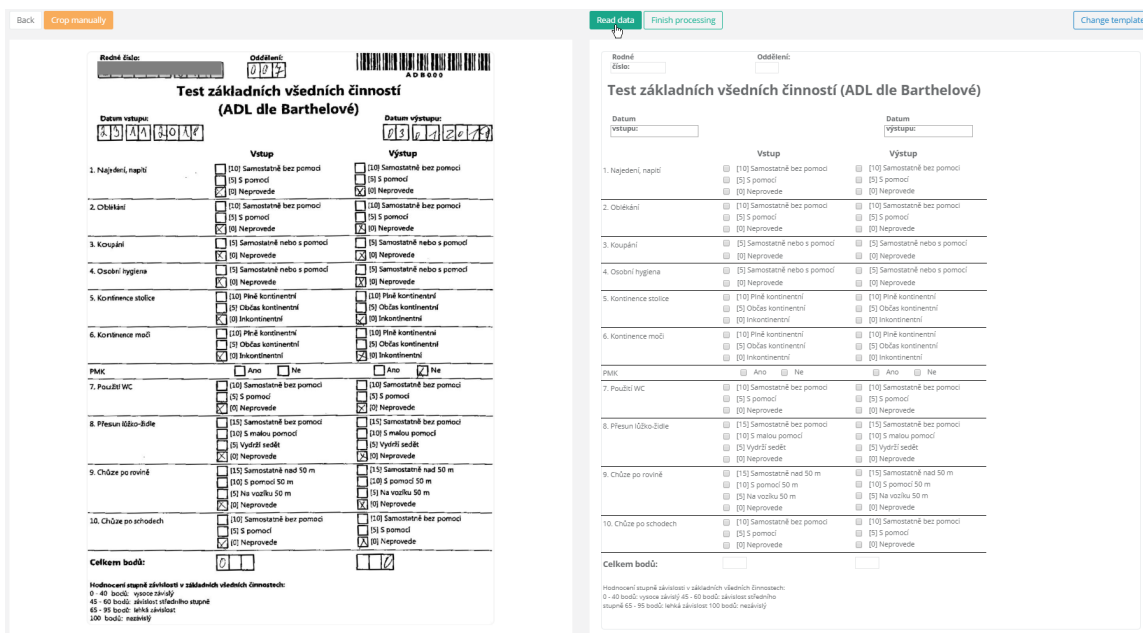
V případě, že je snímek nekvalitní nebo je ofocen tak, program nedokáže formulář správně ořezat, mohou být rohy označeny ručně. Pomocí nástroje pro manuální ořezání je potřeba postupně označit rohy od levého horního po směru hodinových ručiček. Při označování je důležité kurzorem myši mířit přímo na střed čtverečku, který značí okraj dokumentu.

Další problém, který může kvůli nízké kvalitě snímku nastat je, že nebude rozpoznán čárový kód. V tom případě se program chová stejně, jako kdyby tam žádný nebyl a uživatel je vyzván k ručnímu přiřazení elektronické šablony.



Obrázek 4.9: Manuální oříznutí dokumentu dle rohů v případě, že kvůli kvalitě snímku selhalo automatické zpracování.

Pokud vše proběhlo úspěšně, je v tuto chvíli originální snímek upraven a oříznut pouze na efektivní plochu formuláře a je k němu přiřazena správná šablona pro digitalizaci. Stav by měl tedy vypadat jako na obrázku č. 4.10.



Obrázek 4.10: Po úspěšném upravení a přiřazení snímku k šabloně, může uživatel vyplnit data ručně nebo spustit automatické čtení dat.

V této fázi lze data do digitální šablony vyplnit ručně a ukončit digitalizaci a nebo spustit automatické čtení dat a tím vyplnit zaškrťovací pole a případně i některé údaje

zapamatované z posledního zpracovaného formuláře ⁴. Po načtení dat má uživatel možnost zkontrolovat správnost rozpoznání polí. Všechna nalezená pole jsou označena – ta vyplněná jsou ohraničena červeně a prázdná modře. Ostatní údaje lze doplnit ručně a následně dokončit digitalizaci.

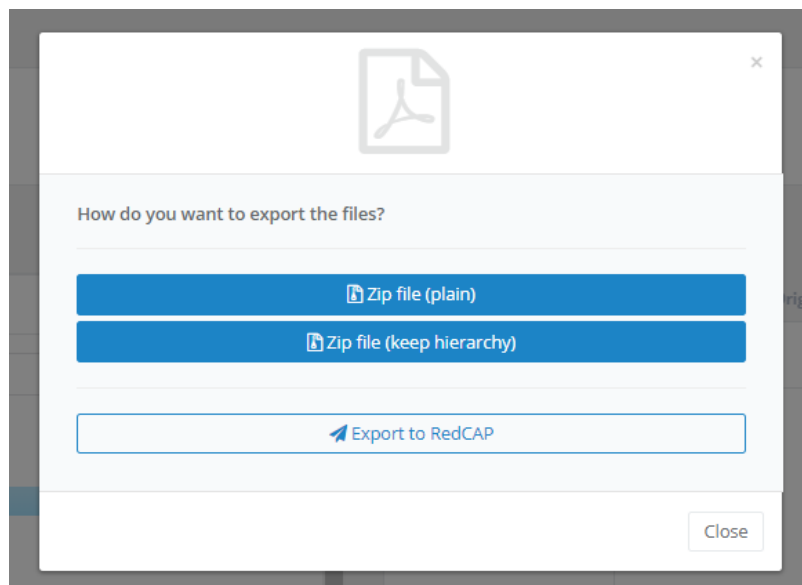
Obrázek 4.11: Po automatickém čtení dat jsou nalezená zaškrtnutá pole označena červeně a prázdná pole modře. Na snímku lze také vidět validaci rodného čísla, kdy je textové pole podbarveno zeleně u validního vstupu a červeně v opačném případě.

4.2.6 Zpracované formuláře a export

Po dokončení digitalizace je možné v systému své dokumenty procházet a prohlížet. Každý digitalizovaný formulář je uložen v elektronické podobě s vyplněnými daty, ale také je k němu přiložen původní oskenovaný dokument a ořezaný snímek, který byl použit pro úpravu. Stránka pro správu zpracovaných formulářů opět dodržuje zavedené uživatelské rozhraní a procházet dokumenty lze tedy obvyklým způsobem.

Kromě samotné správy souborů je zde také možnost exportu. Ten lze provést přímo nad jedním zvoleným souborem. Základní funkcionalitou je stažení a uložení do formátu ZIP. Výsledný archiv se skládá z 3 položek – XML kód šablony s vyplněnými hodnotami a dva snímky. Uložení do tohoto formátu je možné také hromadně, kdy jsou exportovány všechny zpracované soubory do jednoho archivu. Zde je možnost si vybrat, zda má být zachována hierarchie složek z platformy nebo ne. Vizte na obrázku č.4.12.

⁴Tato vlastnost je zatím určena pouze pro Gerontologické centrum v Praze 8 a není konfigurovatelná z uživatelského rozhraní.



Obrázek 4.12: Při hromadném exportu lze zvolit, zda má být zachována hierarchie složek z platformy, či ne.

Druhým způsobem je export do databáze RedCAP. Tato metoda je implementována speciálně pro cílovou kliniku Gerontologické centrum v Praze 8, která ve zmíněné databázi uchovává svá funkční vyšetření. Tento způsob využívá Formello API a pro účely této práce funguje v prototypové verzi. Pomocí této funkce by do budoucna mělo být snadné propojit a integrovat platformu s dalšími systémy, což je velmi důležité. Při zahájení exportu do RedCAP na konkrétním souboru, se dedikované tlačítko zbarví do oranžové barvy s načítáním, až dokud operace nedoběhne. U hromadného exportu se barva tlačítka nemění. Důležité je, že celá funkcionality pracuje na pozadí, uživatel tedy může mezitím využívat platformu dále.

V této kapitole byl popsán a rozebrán systém z pohledu UI. Nejdůležitější z výše zmíněných operací jsou však v práci rozebrány i z implementační stránky a to v kapitole č.4.6.

4.3 Technologie

Pro vývoj frontendové části aplikace, tedy toho, co vidí uživatel na straně svého prohlížeče, bylo využito standardních technologií HTML5 a CSS3. V aplikaci je, především kvůli uživatelské přívětivosti a reaktivitě, do prohlížeče přenesena i některá logika. Pro tyto účely byl využit programovací jazyk JavaScript a s ním knihovna obsahující velké množství užitečných funkcí JQuery. Pro asynchronní zpracovávání požadavků bez nutnosti načítání celé stránky byla využita technologie AJAX⁵.

Serverová část platformy – backend, je kompletně postavena nad Microsoft .NET frameworkem. Ten nabízí řadu technologií pro vývoj, ze kterých byl jako programovací jazyk zvolen C# a pro webovou část framework ASP.NET. K ukládání dat je využit relační databázový systém MSSQL. Aplikace běží na webovém serveru IIS⁶, určeném pro operační systém Windows.

Níže jsou explicitně přiblíženy další použité frameworky, které jsou pro implementaci velmi důležité.

ASP.NET MVC

ASP.NET je nástroj pro tvorbu webových aplikací a služeb. Nabízí více možností, jak k vývoji přistupovat a všechny mají nějaké výhody a nevýhody. Nejmodernějším způsobem a doporučenou volbou, je nadstavba MVC. Ta využívá architektury Model-View-Controller, popsané dále v podkapitole 4.4.

Framework nabízí širokou škálu usnadnění vývoje, mezi hlavní patří například technologie Razor. Ta umožňuje při programování na frontendu v HTML stránce přistupovat do backendové části kódu a tím provádět komplexnější úlohy nebo pracovat s databází. Zároveň obsahuje připravené metody – Razor helpers, jako validace a zabezpečení vstupů, či odesílání formulářů a podobné.

Dalším velkým pomocníkem v přehlednosti a znovupoužitelnosti kódu jsou Partial views, díky kterým lze stránku rozdělit do několika samostatných komponent, které lze vkládat i na jiné stránky. Každá komponenta může mít vlastní logiku s vlastním řadičem (controller). Navíc může být volána pomocí AJAX, a tak se vykreslí nebo obnoví bez nutnosti aktualizovat celou stránku. [7]

Využití tohoto frameworku výrazně pomohlo při vývoji a především vedlo k udržitelnějšímu a strukturovanějšímu kódu.

Git + Azure DevOps

Pro verzování vývoje byl použit nástroj Git. Kvůli zálohování a větší dostupnosti kódu je dobré při práci s Gitem využít nějakého systému, který nabízí online nadstavbu. Pro účely této diplomové práce jsem zvolil Azure DevOps. Jedná se o webově založenou službu, která umožňuje správu verzí v online repozitáři a je přímo integrovaná s Gitem. Mimo samotnou správu repozitáře nabízí i další funkce v balíčku Azure Devops Services, pro týmovou spolupráci a především plánování a monitorování postupu práce. Další velmi podstatnou funkcionalitou je automatický deploy (nasazení) aplikace včetně spouštění testů, sestavování řešení, přepisování testovacích konstant apod. Tyto nástroje bych rád do budoucna využíval. Funkce pro plánování a monitorování vývoje byly velkou součástí implementace. O tom je více řečeno v sekci 4.1.

⁵Asynchronous JavaScript and XML

⁶Internet Information Services

Bootstrap

K implementaci uživatelského rozhraní byl využit CSS framework Bootstrap 3.0. Ten obsahuje základní komponenty typické pro webové stránky, které jsou i dobře rozšiřitelné a především profesionálně nakódované. To umožňuje se více soustředit na samotnou logiku programu. Pro ještě lepší uživatelský dojem z UI byla využita šablona postavená právě na Bootstrapu.

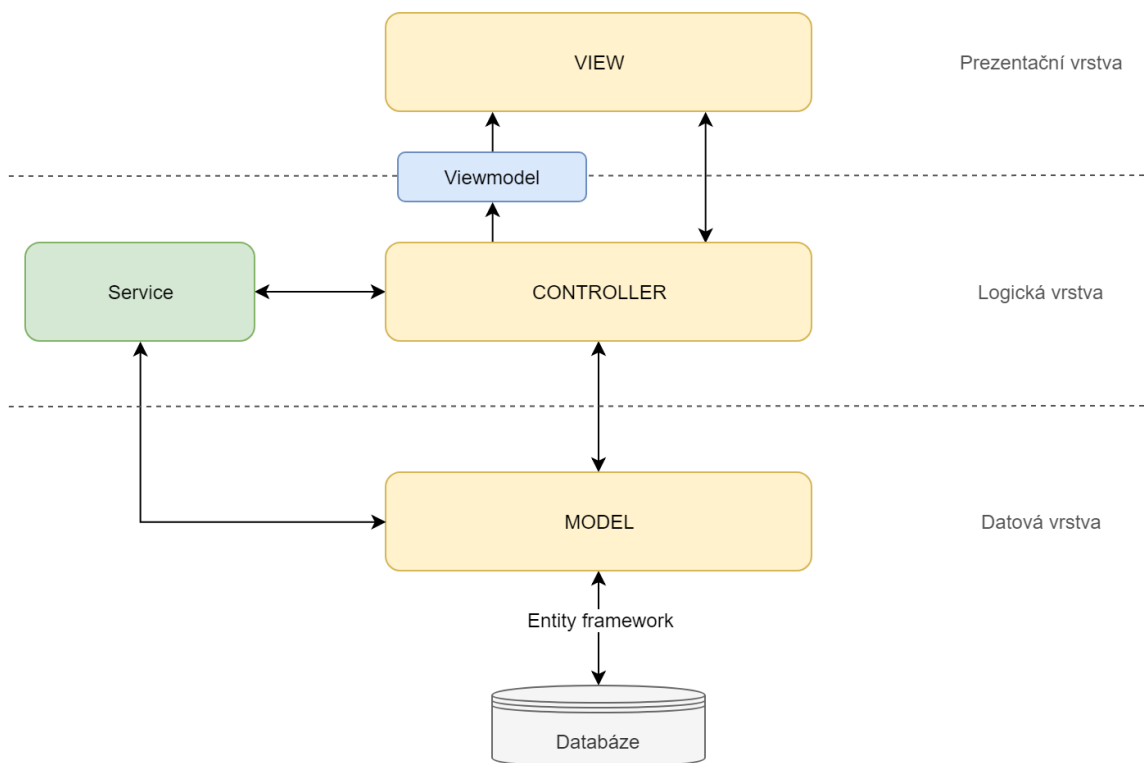
Mimo výše popsané byly využity i tyto další frameworky a knihovny: AForge, DotNet-Zip, JsTree, Dropzone, FlotCharts, Jasy Bootstrap, PdfJs, MetisMenu, SlimScroll, Toastr, PACE, Inspinia template, jQuery UI, iCheck, Ionic.Zip, Animate.css, jQuery DatePicker, Flurl, iTextSharp, Summernote, Html2Canvas.

4.4 Architektura

System je postaven na dnes velmi rozšířené architektuře MVC – Model View Controller.

Model–View–Controller

Architektura se nepoužívá pouze pro webové aplikace, přináší zde však velkou výhodu, protože nám nativně umožňuje rozdělit zobrazovací část, kterou vidí uživatel – View a serverovou – Model a Controller. Bez tohoto rozdělení by mohl vzniknout velmi nepřehledný kód, tímto naopak vznikne sice více tříd a souborů, avšak každý se svou jasně specifikovanou funkcí a nedochází k proplétání různých úrovní aplikace. Díky tomu se dá systém lépe rozšiřovat, testovat a stává se mnohem přehlednější, například pro nového programátora. Architektura rozděluje kód do tří hlavních vrstev, ty jsou zobrazeny na obrázku 4.13 a popsány níže.



Obrázek 4.13: MVC architektura použitá při vývoji

Model obsahuje datovou vrstvu aplikace a reprezentuje strukturu dat. Jak bude s daty zacházeno, již není jeho zodpovědnost. Často se v modelech nachází objekty reprezentující databázové tabulky a vztahy mezi nimi. Nad vlastnostmi objektů mohou být implementovány některé upravující metody, které záznam zeditují před posláním do vyšší vrstvy nebo uložením do databáze. Stejně tak zde může probíhat validace. V této práci je model nejméně využívaná vrstva, protože aplikace nepracuje s příliš rozsáhlou databází. Pro mapování databázových tabulek na objekty a zpět je používán Entity Framework.

Druhou vrstvou je Controller, česky Řadič. Jeho hlavním úkolem je propojení prezentační a datové vrstvy. Na této úrovni je vykonávána většina logiky aplikace. Pro ještě lepší čitelnost a rozdělení kódu je v programu využíváno takzvaných Services. To jsou třídy obsahující operace nad objekty a jiné algoritmické metody, které jsou volány z Řadiče. Tím v Controlleru zůstává kód přehlednější. Typický proces v této vrstvě probíhá tak, že uživatel z prezentační části – View, pošle požadavek s nějakými parametry do Řadiče. Ten na základě parametrů a podmínek rozhodne, jaká činnost se bude vykonávat a buď data pošle dále do datové vrstvy a nebo zavolá Service, kde se vykoná nějaká pokročilejší logika a výsledek je opět navrácen do Řadiče. Po dokonání logiky Controller odpoví zpět do View, kde už je výsledek podle jiných pravidel zobrazen uživateli. To však už není zodpovědnost této vrstvy.

Mezi logickou a prezentační vrstvou existuje jakási volitelná mezivrstva – Viewmodel. Ta obsahuje objekty, které Controller posílá do prezentační vrstvy jako výsledek, kde s nimi pracuje Razor engine.

Nejvyšší vrstvou je View – Pohled. Tato část přímo interaguje s uživatelem a zahrnuje i uživatelské rozhraní. Dala by se rozdělit na statickou, sem patří HTML a dynamickou podvrstvou – JavaScript. V této práci hraje dynamická část někdy velmi důležitou roli,

například v designeru formulářů. Avšak i přesto zůstává princip stejný. View zobrazuje data, která jsou mu zaslána z nižší vrstvy nebo ze strany uživatele v prohlížeči. [8][3]

Při implementaci byla dodržována doporučená metodika tvorby a pojmenování modelů, Řadičů i Views. Kdy se třída skládá z názvu entity, nad kterou pracuje a klíčového slova označující název vrstvy. Pro každý případ užití a dalo by se i říci, že položku menu z vyšší hierarchie, by měl existovat vlastní Controller. Ten může mít více pohledů a modelů, ale zaštiťuje především operace nad vybranou entitou.

Tolik k obecnému popisu architektury, z níž platforma vychází a na které je postavena. Další kapitoly se budou více věnovat konkrétnějšímu rozboru implementované aplikace.

4.5 Struktura a soubory aplikace

Zdrojový kód aplikace zachovává výše zmíněné principy MVC, dá se tedy rozdělit do tří vrstev. V této kapitole jsou popsány některé důležité soubory a třídy aplikace. Podrobnější implementační popis je poté vysvětlen v kapitole 4.6.

Frontend, pohledy a Javascript

Tato část je implementována převážně v technologiích HTML5 a CSS3. Framework ASP.NET MVC umožňuje vytvářet takzvané Masterpage, což jsou stránky, ve kterých jsou definovány stálé prvky, jednotné pro více podstránek. Například navigační menu, patička, záhlaví, drobečková navigace. V této práci je, s výjimkou přihlašovací stránky, vždy použita jedna společná Masterpage – „_Layout.cshtml“. Obsah uvnitř ní se generuje dle aktuálního Řadiče pomocí metody Razor technologie `RenderBody()`.

Při vývoji bylo dbáno na znovupoužitelnost již vytvořeného kódu, proto jsou i v prezentační části aplikace využívány jednou naprogramované dynamické komponenty, které lze umisťovat do více stránek – Partial Views. Tyto soubory začínají prefixem „_“ Příkladem některých klíčových částečných pohledů jsou:

- „_Navigation.cshtml“ – Navigační menu aplikace.
- „_FileTree.cshtml“ – Komponenta pro správu dokumentů.
- „_TemplatePreview.cshtml“ – Pohled zobrazující náhled šablony formuláře.
- „_Breadcrumbs.cshtml“ – Drobečková navigace.

Partial Views jsou umisťovány do mateřských Views. Ty představují typicky jednu ucelenou stránku, která je zobrazena při vstupu na její webovou adresu. Pohledy jsou do internetového prohlížeče k uživateli posílány z Řadičů. V aplikaci je dodržen standard architektury MVC, že jeden Řadič obhospodařuje Pohledy týkající se jedné entity, které se nachází ve složce, pojmenované právě po Controlleru. Například mějme entitu „Account“. Všechny Views, zobrazované Řadičem „AccountController“, se nachází ve složce „Account“. Entity lze v někdy chápat také jako fáze práce s formulářem. Typickým příkladem je „Process“, sdružující Pohledy a logiku týkající se zpracování oskenovaných dokumentů.

Aby aplikace působila co nejsvižnějším dojmem, velká část komponent se načítá dynamicky pomocí Javascript technologie AJAX. Uživatel tak nemusí přecházet na jinou stránku, aby vykonal HTTP požadavek, ale ten je prováděn na pozadí. Tímto způsobem je

vykonáváno například zobrazování náhledů šablon i formulářů, práce s adresářovou strukturou nebo nahrávání souborů. Cílem aplikace však zůstává rychlost, tato dynamická volání na pozadí jsou tedy vykonávána především pro méně náročné operace. Složitější logika, jako práce s obrazem je prováděna standardně.

Mimo AJAX metody je použit také Javascript a rozšiřující knihovna JQuery. Ty slouží rovněž k dodání jisté dynamičnosti a reaktivity platformy. Logika je v tomto případě vykonávána přímo v internetovém prohlížeči na straně uživatele, v některých částech se jedná o poměrně tlustého klienta – velká část výpočtů je přesměrována na cílové zařízení. Největšími takto postavenými komponentami jsou designer formulářů, prohlížeč adresářové struktury a manuální označování rohů snímku. Více o jejich implementaci v kapitole 4.6.

Důležitými složkami jsou také „Content“ a „Scripts“. První z nich obsahuje kaskádové styly a obrázky. Zároveň jsou zde také ukládány dočasné soubory během digitalizace formulářů. Druhá sdružuje zdrojové kódy v programovacím jazyce Javascript, které jsou odsud načítány do pohledů.

Backend a Řadiče

Jak již bylo několikrát zmíněno, v této vrstvě se vykonává většina logiky aplikace. Ta může být umístěna přímo v Controllerech nebo vyčleněna do samostatných pomocných tříd – Services, či dalších metod. Všechny metody řídicí logiku jsou však primárně volány z Řadičů. Zároveň jsou odsud posílány výsledky zpět do prezentační vrstvy a to včetně pohledů naplněných daty.

Zde jsou některé klíčové Řadiče:

- „HomeController.cs“ – Zobrazuje domovskou nástěnku a její data.
- „AccountController.cs“ – Zajišťující operace nad uživatelskými účty.
- „TreeViewController.cs“ – Obsluhující průzkumník adresářové struktury.
- „EditorController.cs“ – Vykonávající operace spojené s designerem formulářů.
- „BrowseController.cs“ – Slouží k procházení zpracovaných dokumentů.
- „Scan“ Controllery – Skupina Řadičů, které zajišťují digitalizaci.
 - „ScanController.cs“ – První v pořadí, zajišťuje správu a nahrávání oskenovaných dokumentů.
 - „ActionController.cs“ – Tento Řadič je vyvolán, pokud je potřeba manuálně označit rohy oskenovaného dokumentu.
 - „ProcessController.cs“ – Zajišťuje samotnou digitalizaci a volání přidružených metod. Více v kapitole 4.6.
- „DropzoneController.cs“ – Obstarává nahrávání různých typů souborů a oskenovaných dokumentů.

Do samostatné kategorie patří takzvané API Řadiče. Ty nevracejí žádný Pohled, ale typicky pouze čistá data ve formátu JSON, či XML. Jsou používány pro integraci platformy s dalšími aplikacemi.

Mimo Controllery a Services jsou součástí kódu také pomocné metody a třídy. Většina z nich obsahuje logiku volanou z více míst a sdružuje ji. Hlavní pomocné třídy jsou:

- „FileSystemHelpers.cs“ – Sdružují metody pro práci se souborovým systémem, kde jsou uchovávány formuláře a šablony.
- „Constants.cs“ – Třída obsahující důležité konstanty dostupné v celé aplikaci.
- „HTMLHelpersExtensions.cs“ – Metody rozšiřující standardní Razor komponenty.
- „ApiHelpers.cs“ – Třída pro konstrukci API požadavků a odpovědí.
- „Scan“ helpers – Skupina pomocných metod a tříd využívaných při digitalizaci. Mezi hlavní patří:
 - „ImgProcess.cs“ – Obsahující metody pro práci s obrazem pomocí knihovny AForge .NET.
 - „PaperProcess.cs“ – Logika zpracovávající digitalizaci a rozpoznávání zaškrtnutých polí je v této třídě.
 - „FormManip.cs“ – Zpracování výsledků digitalizace a jejich ukládání do XML šablon a označení na původním snímku.
 - „ScanXmlHelper.cs“ – Mapování XML šablony do HTML podoby.
 - „BarcodeImaging.cs“ – Metody pro rozpoznávání čárového kódu pomocí knihovny.

Do této vrstvy patří také externí knihovny napsané v programovacím jazyce C#.NET. Ty jsou ukládány ve složce Bin, pro dynamicky kompilované a verzované pomocí Nuget, a v adresáři Lib pro neměnné knihovny.

DB a modely

Databázová vrstva v platformě není rozsáhlá. Největší zastoupení má práce s uživatelskými účty. Data uživatelů jsou uchovávána v MSSQL databázi. Systém je připraven pro tvorbu rozsáhlejší vrstvy, nyní většinu informací, s kterými aplikace pracuje, tvoří soubory a dokumenty. Z důvodu zachování rychlosti databáze do budoucna a možného využití externích cloudových úložišť, jsou tato data skladována na souborovém systému serveru.

Avšak i přesto je v kódu celá řada Modelů. Jediný rozdíl oproti těm, které se týkají uživatelů je ten, že jsou plněny jiným způsobem, než z databáze. Rovněž ale slouží ke strukturovanému a objektovému předávání informací napříč vrstvami, kdy jsou v Řadiči plněny daty a posílány do Pohledů. Ty modely, které nejsou plněny z databáze jsou z důvodu rozlišení označeny suffixem ViewModel. Mezi významné patří například:

- „FileSystemViewModels.cs“ – Nesoucí informace o položkách ze souborového systému a jejich attributech.
- „TemplatePreviewViewModel.cs“ – Plněn daty z šablony v XML formátu a PDF.
- „DashboardViewModel.cs“ – Obsahuje data k zobrazení na domovské nástěnce.

Další důležité soubory řešení

Ve zdrojovém kódu hrají důležitou roli i různé konfigurační soubory, které je také důležité zmínit.

- „Web.config“ – Hlavní konfigurační soubor aplikace.

- „Formello.sln“ – Soubor sloužící pro otevření a načtení řešení v programu Microsoft Visual Studio.
- „BundleConfig.cs“ – Tato třída obsahuje seznam statických souborů, jako jsou zdrojové kódy v programovacím jazyce Javascript a kaskádové styly. Před spuštěním aplikace jsou tyto soubory zkomprimovány a na webovou stránku se nahrají ve zmenšené podobě.
- „Startup.Auth.cs“ – Konfigurace autorizace a způsobů autentizace do aplikace.
- „IdentityConfig.cs“ – Nastavení registrace nových uživatelů.
- „RouteConfig.cs“ – Definice způsobu navigování v aplikaci přes URL adresu.
- „Global.asax.cs“ – Třída, která je vyvolána jako první při spuštění aplikace a jsou odsud volány konfigurační metody.
- „FormSchema.xsd“ – Definice XML schématu, která je využívána u šablon formulářů.

4.6 Vybrané implementační detaily

V rámci této kapitoly jsou vybrány některé důležité a zajímavé implementační detaily a blíže popsány. Jsou zahrnuty i krátké ukázky zdrojových kódů. Veškerý zdrojový kód je k dispozici v příloze této práce.

4.6.1 Designer

Jednou z nejsložitějších komponent programu je designer, neboli návrhář formulářů. Prototyp této komponenty byl vytvořen již na Katedře biomedicínské informatiky FBMI ČVUT. Více ohledně autorství tohoto zdrojového kódu je uvedeno v kapitole 2.2.

Tento nástroj je vyvinut v programovacím jazyce Javascript, s využitím knihovny JQuery. Pracuje na principu dynamického generování HTML kódu. Obsahuje v současné podobě přibližně 2000 řádků kódu aplikační logiky. Před rozšířením v rámci této práce zahrnoval pouze zaškrtačací pole a jejich posouvání po pracovní ploše a vykreslení šablony do prohlížeče ze souboru XML. K těmto účelům byla vytvořena transformace XSLT, která definuje převod do formátu HTML. Projekt však skončil ve fázi prototypu a nebyl dokončen nebo nasazen.

Po převzetí projektu probíhalo rozšiřování funkcionalit iterativně, na základě nově vzniklých požadavků nebo nápadů, po dobu tří let. Protože byly postupně doplňovány nové metody bez předem promyšlené architektury, výsledkem je dlouhý, ne příliš čistě strukturovaný zdrojový kód v souboru „app.js“. Pro budoucí udržitelnost a rozšiřitelnost tohoto modulu bude potřeba nutná refaktorizace⁷. Aktuální stav funkcionalit a kompletní popis designu je uveden v kapitole 4.2.5.

Klíčovou roli hrají především dva objekty – „FormCreation“ a „pageSettings“. První zmíněný sdružuje metody obsluhující práci a manipulaci s prvky na pracovní ploše. Od jejich inicializace po smazání. Obsahuje 38 metod, přímo pracujících s komponentami formuláře. Mimo to zahrnuje také 22 akcí, čekajících na události na formuláři. Druhý zmíněný objekt je dedikován obecnému nastavení a definici vznikajícího formuláře, obsahuje mimo jiné i výčet povolených prvků a jejich vlastností. Klíčovými metodami pro vykreslení formuláře na pracovní ploše jsou „xmlParser“, pomocí níž dochází k namapování vlastností

⁷Zlepšení kvality zdrojového kódu.

prvků podle transformace XSLT a dostupných prvků do „pageSettings“ a „setPage“, která připraví stránku dle definovaných parametrů v „pageSettings“ a zavolá instanci objektu „FormCreation“.

```
$(".elements").append('<div class="elementsHeader">Toolbox</div>');
_.forEach(pageSettings.availableObjects, function (name, key) {
    var content = name;
    var friendlyName = elementsFriendlyNames[name];

    if (typeof pageSettings.availableObjectsImages[name] !== "undefined") {
        content = '';
    }

    $(".elements").append(
        '<div class="pdfObject btn btn-default" data-toggle="tooltip" data-placement="right" title="' + elementHint +
        '" data-type="' + name + '">' + content + '<span class="element-text-name">' + friendlyName + '</span>' +
        '</div><br/>');
});
```

Obrázek 4.14: Zdrojový kód v metodě „setPage“, který využívá objektu „pageSettings“ a dynamicky generuje HTML kód panelu s nástroji.

Popisovat celý zdrojový kód od základu by bylo velmi náročné, proto je vybráno pouze několik specifictějších, rovněž důležitých funkcionalit. Mezi hlavní patří například export do PDF, generování čárového kódu nebo možnost využití mřížky s přichytáváním při pohybu prvků.

K exportu do PDF byly využity dvě open-source knihovny – jsPdf⁸ a html2canvas⁹, které v kombinaci dokáží vytvořit z HTML kódu dokument ve formátu PDF. Proces probíhá tak, že je nejprve obsah pracovní plochy převeden do prvku jazyka HTML5 – „canvas“, pomocí druhé zmíněné knihovny. A JsPdf umožňuje samotný převod do PDF.

Další zajímavou funkcionalitou je generování čárového kódu, to probíhá přímo během zadávání jeho cílového obsahu do dedikovaného textového pole. Na něj je při stisku klávesy navázána akce přepočítávající čárový kód pomocí knihovny JsBarcode¹⁰. Generování probíhá ve standardním formátu CODE128. Není povolena diakritika.

V designeru byla využita ještě jedna externí knihovna – Summernote¹¹. Jedná se o open-source textový WYSIWYG editor, který si může uživatel zobrazit při vkládání textů do formuláře.

Mřížka, neboli grid, s přichytáváním objektů, je vytvořena pomocí jednoduchého principu, kdy pozadí tvoří čtverečky o velikosti 10 pixelů. Pokud je možnost mřížky zapnutá, toto pozadí se zobrazí a při přesouvání objektů není pozice objektu inkrementována o desetiny pixelu jako obvykle, ale souřadnice jsou přepisovány tak, že se mění vždy o celých deset pixelů. Přesouvání poté opticky působí dojmem, že se objekty přichytávají k mřížce, protože na jinou pozici, než právě hrany čtverečků objekt ani nelze přesunout. Přepisování souřadnic je ukázáno v ukázce kódu 4.16.

⁸<https://github.com/MrRio/jsPDF>

⁹<https://github.com/niklasvh/html2canvas/>

¹⁰<https://github.com/lindell/JsBarcode>

¹¹<https://github.com/summernote/summernote/>

```

positionToGridPosition: function (left, top) {
    if (this.gridToggleOn) {
        return {
            left: Math.round(left / this.gridStepLeft) * this.gridStepLeft,
            top: Math.round(top / this.gridStepTop) * this.gridStepTop
        };
    } else {
        return {
            left: left,
            top: top
        };
    }
},

```

Obrázek 4.15: Část zdrojového kódu, zajišťující přichytávání objektů k mřížce tím, že umožňuje posun prvků pouze ve skocích o jeden krok gridu.

4.6.2 OMR algoritmus a automatizace

Další rozsáhlou a složitou komponentou platformy je nástroj pro digitalizaci. Algoritmus optického rozpoznávání zaškrtnutých polí – OMR, byl převzat z mé vlastní bakalářské práce. [7] Je implementován v programovacím jazyce C# .NET, s využitím knihovny pro zpracování obrazu AForge¹². Detailní popis je k dispozici právě ve zmíněné bakalářské práci, případně v příloženém zdrojovém kódu.

Zahrnuje několik stěžejních kroků.

Upravení oskenovaného snímku

K úpravám snímku jsou využity třídy „ImgProcess“ a „PaperProcess“. Obraz je převeden na černobílý, poté dochází k ekvalizaci histogramu a následnému prahování – „Thresholdingu“, čímž dochází k binarizaci snímku a rozdělení všech pixelů pouze do černé nebo bílé barvy.

Poté jsou použity metody „FindCorners()“ a „CropByCorners()“ k ořezání dle rohů. První funkce postupně obraz invertuje, následně za využití třídy „BlobCounter“, nalezne objekty oddělené černým pozadím a o určitém rozsahu velikosti. Vyfiltruje ty, které jsou ve tvaru čtverce a seřadí je na základě jejich pozice ve snímku. Značky označující rohy dokumentu tvoří okraje. Jsou tedy vybrány nejkrajnější objekty, které odpovídají kritériím. Podle jejich souřadnic již metoda „CropByCorners()“ snímek ořízne.

Rovněž zde dochází k hledání čárového kódu, kdy je obraz padesátkrát iterativně procházen a jsou hledány určité charakteristiky odpovídající právě čárovému kódu, jež je případně dekodován.

¹²<http://www.aforgenet.com/>


```

//Upravení snímku před hledáním rohů
public static Bitmap PrepareImage(System.Drawing.Image images)
{
    var img = new Bitmap(images);
    //Upravení fotky
    img = ImgProcess.Grayscale(img);
    img = ImgProcess.Equalize(img);
    img = ImgProcess.Threshold(img);
    int treshVal = PaperProcess.GetCropThresholdValue(img);
    img = PaperProcess.Thresholding(img, treshVal);
    img = ImgProcess.IndexedBitmapToRgb(img);
    return img;
}
//Automatike hledání rohů
public static Bitmap AutomaticallyFindCorners(Bitmap img)
{
    List<Blob> corners = PaperProcess.FindCorners(img);
    img = PaperProcess.CropByCorners(img, corners);
    return img;
}
}

```

Obrázek 4.16: Předzpracování obrazu s využitím knihovny AForge. Metody „PrepareImage“ a „AutomaticallyFindCorners“, které postupně vylepší kvalitu obrazu a oříznou jej dle rohových značek.

Ruční označení rohů

Při špatné kvalitě oskenovaného dokumentu, nebo provedení samotného skenování, může dojít k chybnému výřezu dle rohů. V tomto scénáři je nutno rohy označit ručně v dedikovaném nástroji. Ten je založen na technologii HTML5 Canvas, díky níž je možné v internetovém prohlížeči kreslit na plátno pomocí jazyka Javascript. Na pozadí je tedy nastaven původní oskenovaný snímek formuláře. Při kliku na plátno je vyvolána metoda „canvas.strokeRect()“, která vykreslí čtverec okolo označeného místa a uloží si souřadnice kliknutí myši pomocí „getBoundingClientRect()“. Zároveň dekrementuje čítač označených objektů o jeden. Po dokončení jsou souřadnice odeslány do Řadiče „ProcessController“ a akce „CornerProcess“, která ořízne snímek za pomoci knihovny AForge.

Automatické čtení dat

Pro vyhodnocení zaškrťávacích polí je klíčové znát přesně jejich pozici. Ta je definována v XML šabloně, vytvořené pomocí designeru. Proto je potřeba snímek velmi přesně oříznout a vyrovnat ve fázi předzpracování, aby souřadnice odpovídaly. Na základě těchto souřadnic je poté každý prvek zvlášť vybrán a vyhodnocen. Výstupem je informace, zda je pole zaškrtnuté nebo ne.

Tuto úlohu řeší třída „ImgCheckBox“. Zaškrťávací pole má danou výšku, šířku a odsazení. Na základě těchto vlastností je vyříznuta část obrazu, kde se nachází a dále se pracuje pouze s ní. Pro vyhodnocení je nejprve odfiltrován okraj čtverce a hodnotí se pouze jeho vnitřek. V cyklu jsou procházeny jednotlivé pixely a vyhodnocovány, zda jsou černé nebo bílé. V případě, že zaškrtnutá část přesáhne určitou hodnotu, je pole považováno za vyplněné. Prahová hodnota je zvolena deterministicky na základě testování.

```

for (int i = innerRect.X; i < innerRect.Width + innerRect.X; i++)
{
    for (int j = innerRect.Y; j < innerRect.Height + innerRect.Y; j++)
    {
        if (img.GetPixel(i, j).B < 127){
            blackarea++;
        }
        area++;
    }
}

```

Obrázek 4.17: Část kódu, ve které jsou procházeny všechny pixely zaškrťovacího pole a dochází k vyhodnocování, zda jsou světlé nebo tmavé. Na základě výsledných údajů je rozhodnuto, zda je pole zaškrtnuté.

Pamatování naposledy vyplněných hodnot

Pro účely zařízení, kde je systém testován, jsou během zpracování předvyplněny i některé poslední hodnoty. Jsou jimi hlavní datum, oddělení a čárový kód. Je tomu tak proto, že velmi často zpracovávají vyšetření od jednoho pacienta za sebou, například po deseti. Tato funkcionality se může však hodit i pro jiná užití.

Hodnoty jsou ukládány do takzvaných Session proměnných jako klíč a hodnota, kde klíč je identifikátor prvku. Odtud jsou při dalším zpracování načteny a pokud se element opět nachází na formuláři, je vyplněn.

4.6.3 Převod XML šablony do HTML

Výstupem designera je šablona ve formátu XML. V ní jsou nejen ukládána data, ale i definuje vzhled. Při zobrazení v internetovém prohlížeči musí být však převedena do formátu HTML a tvořit odpovídající webový formulář. To řeší metoda „GXmlToHtmlString“. Převod probíhá ještě na serveru v jazyce C# .NET. Metoda postupně prochází XML šablonu a má pro každý prvek definovaný přepis do HTML verze. Designer funguje tak, aby měl každý formulářový prvek přesně definované vlastnosti, které jsou nezbytné pro správné vykreslení v prohlížeči jako HTML element. Obsahují tedy krom samotných názvů a hodnot další atributy, určující styl. Například velikost písma, šířku, výšku, přesnou pozici zleva a shora. Metoda tedy prochází šablonu a generuje HTML zdrojový kód. Takto je vytvořen strukturovaný formulář a odeslán do View k vykreslení. Zde je dekodován a funguje jako standardní formulář, který lze i s vyplněnými daty odeslat na server.

```

case "label":
    if (nameValueStyle.NameAttribute != String.Empty){
        htmlTextWriter.WriteLine(
            "<label name='" + nameValueStyle.NameAttribute + "' value='" +
            nameValueStyle.ValueAttribute + "' style='" +
            nameValueStyle.StyleAttributes + "'>" + xElement.Value + "</label>");
    }
    else{
        htmlTextWriter.WriteLine(
            "<label name='1' + id + "' value='" + nameValueStyle.ValueAttribute +
            "' style='" + nameValueStyle.StyleAttributes + "'>" + xElement.Value +
            "</label>");
        id++;
    }
    break;

```

Obrázek 4.18: Definice převodu prvku „label“ do jeho HTML verze.

4.6.4 Správa souborů na disku - JSTree

Zpracované formuláře, oskenované snímky i šablony jsou ukládány na souborovém systému, což umožňuje velmi rychlou práci s dokumenty a snadnou migraci dat. Na serveru má každý uživatel platformy svůj vlastní adresář, s přístupem pouze ke svým podsložkám. Myšlenkou

je, že tím má k dispozici vlastní pracovní prostor, kde si může libovolně vytvářet, přesouvat soubory a pracovat s adresářovou strukturou. Bylo ale potřeba vymyslet způsob, jak tuto možnost nejlépe zpřístupnit. Pro tyto účely byla vybrána open-source komponenta JSTree¹³, která nabízí uživatelské rozhraní simulující práci se složkami přímo v internetovém prohlížeči. Nad základními operacemi vyvolává události, které oznamují, že uživatel vykonal v komponentě nějakou akci, například vytvoření složky, přejmenování souboru nebo přesunutí. Tím však funkcionalita této knihovny končí a na tyto události bylo potřeba navázat implementací veškerých operací přímo nad souborovým systémem serveru v C#.NET. Níže je ukázka kódu v programovacím jazyce Javascript, který je vykonán na straně internetového prohlížeče, při přejmenování uzlu adresářové struktury.

```

$('#jstree').on('rename_node.jstree',
function (e, data) {
    var FileTreeObject = {
        CurrentPath: data.instance.get_path(data.node, '/'),
        CurrentName: data.node.text
    };
    $.ajax({
        type: "POST",
        url: '/TreeView/RenameNode',
        data: JSON.stringify(FileTreeObject),
        dataType: "json",
        contentType: 'application/json; charset=utf-8',
        success: function (response) {
            if (response.Success) {
                showSuccessMessage(response.Message, "Hotovo");
            } else {
                showErrorMessage(response.Message, "Chyba");
                $('#jstree').jstree(true).refresh(false, false);
            }
        },
        error: function () {
            showErrorMessage("Chyba během přejmenování.", "Chyba");
        }
    });
});

```

Obrázek 4.19: Kód, který je vyvolán při přejmenování uzlu adresářové struktury pomocí JSTree. Technologii AJAX je volána metoda na serveru, která obslouží danou akci a vrátí výsledek. Ten je opět zpracován a zobrazen uživateli.

Pro plynulost a přívětivost práce jsou veškeré operace prováděny na pozadí pomocí technologie AJAX a doprovázeny notifikacemi, které signalizují, zda a s jakým výsledkem akce došla. Proces tedy funguje tak, že uživatel například vytvoří složku, to vyvolá událost, která na pozadí, aniž by to omezilo uživatele v práci, zavolá serverovou metodu. Ta vykoná stejnou akci nad souborovým systémem a výsledek je vrácen opět do internetového prohlížeče, kde se vyvolá oznámení.

Pracovní prostor obsahuje v základu 4 adresáře. Templates, Scanned, Processed a Export. Každá reprezentuje jinou fázi práce s dotazníkem. Soubory nejsou na disku uloženy v čisté podobě. Například v Templates se pracuje s šablonami. Každá šablona se skládá ze dvou částí – XML verze a PDF. Tyto dva soubory jsou zabaleny do formátu ZIP a zašifrovány heslem, které je tajemstvím serveru. Jedná se spíše o základní prevenci před bezpečnostními hrozbami pro účely této diplomové práce, systém zabezpečení souborů bude v produkční aplikaci zcela jiný. Stejným způsobem jsou uchovávané i zpracované formuláře, kdy však balíček obsahuje vyplněnou šablonu.

Některé metody pracující se složkami obsahují také kontrolu duplikátů, aby nedocházelo k přepisování souborů. Tato potřeba vyplynula z analýzy požadavků. Pokud je tedy soubor ve složce již přítomný, změní se název novějším doplněním suffixu „dupl-“ a aktuální datum a čas.

¹³<https://www.jstree.com/>

4.6.5 REST API pro integraci

Kromě běžného exportu umožňuje platforma také přistupovat k datům skrz REST API. To je velice užitečné při integracích s jinými systémy. Tato funkcionality vznikla z důvodu požadavku na přímý export dat do systému RedCAP. Aplikace pro nahrávání dat do RedCAP již existuje, bylo tedy potřeba vytvořit přístupový bod, odkud si bude moct stahovat digitalizované formuláře. REST API je standardní způsob pro tento scénář.

Byly tedy implementovány dva koncové body – „endpoints“, na internetové adrese platformy se suffixem `/api/export` a `/api/export/done`, sloužící k exportu dat do systému RedCAP. API využívá k autentizaci standardu OAuth 2.0, kdy je k ověření zapotřebí takzvaný „Bearer token“. Ten je uživateli vygenerován na požádání. Endpointy přijímají HTTP GET požadavky a vrací data ve formátu JSON.

Při dotazu bez parametrů na první zmíněný koncový bod jsou vráceny všechny zpracované dokumenty, které byly označeny k exportu stiskem „Export all to RedCAP“, v sekci správy digitalizovaných formulářů. Po úspěšném stažení je pro další export potřeba soubory opět označit. Druhou variantou je synchronizace po jednom. K tomu slouží stejný endpoint, avšak je potřeba poslat parametr „ID“, určující, jaký dokument má být stažen.

V praxi momentálně integrace funguje tak, že při stisku tlačítka „Export to RedCAP“, platforma označí dokument a informuje aplikaci pro integraci o novém souboru k synchronizaci. Pošle zároveň ID tohoto dokumentu. Software třetí strany tedy zavolá požadavek na REST API i s identifikátorem a stáhne si daný dokument. Po dokončení integrace může oznámit, že formulář byl již zpracován, zasláním požadavku na adresu `/api/export/done` s identifikátorem dokumentu. Ten se vrátí do původního stavu a je zrušeno označení.

Ačkoliv je tato integrace implementována specificky, zůstává tento způsob velmi snadno rozšiřitelný pro případné další synchronizace s jinými systémy.

Kapitola 5

Testování

Součástí této diplomové práce je testování v reálném prostředí Gerontologického centra v Praze 8, ideálně přímo s cílovým personálem na reálných situacích, aby byl program co nejlépe začleněn do tamních procesů. V této kapitole bude popsán způsob, jakým testování probíhalo a jakých výsledků bylo dosaženo. Bude vyhodnocována zpětná vazba a analyzovány nové požadavky na systém.

Způsobů, jak otestovat software, je mnoho. V této diplomové práci se zaměřím především na chybovost systému, uživatelskou zkušenost a míru přínosu koncovým uživatelům.

5.1 Testování kódu

Chybovost programu je, kromě špatných dat, většinou způsobena chybným kódem nebo neošetřenou situací. Ten může být implementován již od počátku špatně, ale také se může stát, že chyba je zanesena až později, během úprav kódu nebo dodělávání dalších funkcionalit. Pro oba tyto případy je proto důležité psát jednotkové testy, které mají za cíl otestovat kód na úrovni jednotlivých metod – Unit testy. Ty slouží nejen k odhalení zanesené chyby, ale také k prevenci chyb tím, že si díky unit testu vývojář může uvědomit nějakou další chtěnou, či nechtěnou vlastnost metody. Jednotkové testy se většinou píší přímo během nebo po implementaci dané funkcionality, přesněji konkrétní metody. Existují metodiky, které navrhuje psát dokonce ještě před programováním samotné funkce, protože si programátor lépe uvědomí, jak se metoda má vlastně chovat a jaké výsledky má vracet. Psát kód spolu s unit testy se může někdy časově protáhnout a někdy jsou samotné testy náročnější pro přípravu a implementaci, než kód. Z těchto důvodů a zároveň vysokému potenciálu změn v této rané fázi vývoje platformy bylo od Unit testů upuštěno.

Pro zaručení kvality, snížení rizika chyb a nižší chybovosti obecně, jsem se rozhodl v rámci vývoje začlenit alespoň manuální integrační testování, kdy je při nasazení jakékoliv větší změny manuálně otestován hlavní uživatelský scénář systému. Při menších změnách pouze část, ve které se úprava vyskytla. Do budoucna však vzniká technický dluh a měly být dopsány jednotkové testy a pokryt většinu platformního kódu. [1]

5.2 Uživatelské testování

Zpětná vazba od cílových uživatelů a její vyhodnocení je velmi důležitou součástí agilního vývoje software, protože nejen pomáhá nalézt chyby v implementaci a odchylky od očekávaného chování, ale vede také ke zkvalitnění práce s programem.

Tento způsob testování nemůže provádět přímo programátor, protože již ví, jak se systém zachová a očekává nějaké konkrétní výsledky. Jeho chování v programu je tedy zkrácené a není přirozené. Aby test zůstal objektivní, musí být proveden nezávislým uživatelem.

5.2.1 Sledované metriky a cíle

Před provedením testování ve zdravotnické klinice bylo potřeba si připravit strukturu a promyslet průběh testování. Zároveň si nastavit očekávání, sledované metriky a cíle.

Jelikož se jedná o poměrně ranou fázi produktu, nebylo cílem pouze sledování výkonnosti, ale především zjistit, jak moc jsou naplněny požadavky, do jaké míry uživatelé opravdu pocítují usnadnění práce a získat případné návrhy na zlepšení. Tyto metriky jsou těžko měřitelné a mohou být subjektivní. Z výkonnostních pozorování lze měřit čas digitalizace s a bez využití automatizace a míru spolehlivosti, že automaticky načtená data jsou správná. Zde je tedy seznam sledovaných ukazatelů při testování.

- Orientace uživatele v systému – Míra dotazů, zmatenosti nebo potřeby asistence.
- Naplnění očekávání od nového řešení
 - Čeho si uživatelé váží a v čem vidí největší přínos.
 - Kde je prostor ke zlepšení, co by udělali jinak.
 - Zda bylo splněno vše, co od systému očekávají.
- Sesbírání návrhů pro zlepšení a vyhodnocení jejich důležitosti.
- Čas strávený digitalizací v závislosti na automatizaci.
- Míra spolehlivosti automatizace.

5.2.2 Průběh

Dne 2. 5. 2019 proběhlo setkání za účelem uživatelského testování v Řevnicích u Prahy. Přítomny byly tyto osoby.

- Bc. Pavel Šustek – Autor projektu.
- Mgr. Radim Krupička, Ph.D. – Tajemník katedry biomedicínské informatiky FBMI ČVUT v Praze, která úzce spolupracuje s cílovým zdravotnickým zařízením na projektu GDiag, který byl tamním dosavadním řešením.
- Mgr. Anna Beránková – Zástupkyně centra pro studium dlouhověkosti a dlouhodobé péče – CELLO a koordinátorka projektu zabývajícího se výzkumem dlouhodobé péče – NV18-09-00587, který využívá právě zmíněný systém GDiag.
- Iva Kolářová – Administrátorka projektu NV18-09-00587 a rovněž pracovnice CELLO.
- Ida Kubrichtová a Marie Mainerová – Pracovnice používající systém GDiag pro digitalizaci funkčních vyšetření.

Před zahájením testování byl popsán cíl projektu a důvod vzniku nového systému, poté došlo k představení jednotlivých zúčastněných osob, každý popsal svou roli v procesu zpracování funkčních vyšetření. Následně byly vysvětleny výhody i omezení webového řešení a

rozdíly proti stávajícímu systému. Z představení zúčastněných osob vyšlo najevo, že digitalizaci mají na starost dvě pracovnice. Tato část platformy tedy byla klíčová především pro ně. S designerem formulářů, ani exportem nepřijdou do styku. Vytváření šablon je přiděleno administrátorce projektu Ivě Kolářové a export a správa získaných dat koordinátorce magistře Anně Beránkové.

Nejprve jsem prezentoval všechny funkcionality platformy a postupně prošel fáze práce s formulářem. Během toho byly kladeny velmi užitečné dotazy nebo připomínky z praxe a oceněny některé funkce. Ty budou spolu s dalšími rozebrány ve výsledcích testování.

Po tomto školení jsem dal pracovnícím dedikovaným na digitalizaci společně za úkol zpracovat deset oskenovaných funkčních vyšetření od dvou pacientů. Během toho jsem spolu s ostatními sledoval jejich práci a odpovídal jsem na případné dotazy. Po dokončení jsem se optal na připomínky, návrhy ke zlepšení nebo zpětnou vazbu na systém a jak se jim pracovalo.

Poté se přistoupilo k detailnějšímu představení práce s designerem, ten se ukázal být silnou stránkou a vzbudil velmi pozitivní reakce. Následně jsme se bavili o metodách exportu, správě složek a práce s duplikáty. Nakonec jsme rozebírali jakým způsobem budou zřízeny uživatelské účty pro jejich organizaci a rozdělena oprávnění v systému. Na úplný závěr jsem ještě jednou vyvolal diskuzi nad možnými zlepšeními a nechal všechny účastníky ohodnotit platformu. Byly shrnuty výsledky, poznámky a zpětná vazba z testování a domluven další postup.

Výkonnostní metriky jsem testoval později a sám, jelikož mým cílem bylo sledovat rychlost zpracování v případě, že už je nástroj obsluhován zkušeným uživatelem.

5.2.3 Výsledky

Výsledky testování budou rozděleny do čtyř kategorií. Designer, digitalizace, výkonnostní metriky a celková práce s platformou. Pro každou z nich budou vyhodnoceny silné stránky, připomínky nebo dotazy a případně nalezené chyby.

Designer

Nástroj pro vytváření šablon zapůsobil velmi dobře. Obecně se všichni účastníci shodovali, že ušetří spoustu práce a je jednoduchý na používání. Dokonce rovnou vznikl požadavek o přístup do současné prototypové verze, aby již mohl začít být využíván.

- **Pozitivní body** – Kladně hodnocený byl vzhled a i přes velký počet nástrojů jednoduchost používání. Jednoznačně nejpozitivnější reakci však vzbudil podpůrný prvek pro umístování a zarovnávání na pracovní plochu – mřížka (grid). Co bylo rovněž přímo zmíněno jako velké usnadnění práce, je tlačítko pro navrácení poslední provedené akce. Tyto prvky v jejich současném řešení chybí. Oceněna byla také možnost upravit existující šablony a uložit je jako nové. Ohledně vytváření textů na formuláři byli uživatelé rádi za textový WYSIWYG editor.
- **Připomínky a náměty** – U designeru bylo také nejvíce připomínek a návrhů ke zlepšení. Například klávesové zkratky, kde jsme se dohodli na klávesách pro základní operace jako kopírování a vkládání prvků (Ctrl + C a Ctrl + V), označení všech objektů (Ctrl + A), navrácení akce (Ctrl + Z) a mazání pomocí tlačítka Delete. Dalším návrhem byla možnost sloučit více označených prvků na plátně do jednoho objektu, to je však poměrně větší zásah a zatím jsme se shodli, že nemá prioritu. Dalším návrhem byla možnost sloučit při exportu do PDF šablony do jednoho souboru.

- **Nedostatky** – Dvě poznámky jsem vyhodnotil jako mírné chyby a jejich opravě se v designeru budu věnovat jako první. Jedná se o příliš vysoký prvek čárového kódu, který u delších případů zabírá zbytečně moc prostoru, přičemž u něho výška není důležitá. Druhou záležitostí je nemožnost označit více prvků pomocí přetažení ukazatelem myši nad danou plochou, jako je tomu standardem v jiných systémech. Nutnost označit objekty po jednom za stisku klávesy Ctrl se ukázala být příliš nepraktická.

V rámci testování byli účastníci schůzky na závěr vyzváni k ohodnocení jednotlivých částí na stupnici 0-10, kde nula znamená nepoužitelné a deset značí - není co vytknout. Pro návrhář šablon byly známky takovéto.

Účastník	Hodnocení
R.K.	8
A.B.	9.5
Iva K.	9
M.M.	10
Ida K.	9.5
Průměr	9.2

Tabulka 5.1: Výsledné známky pro designer šablon od jednotlivých účastníků.

Digitalizace

V procesu digitalizace přináší platforma inovaci především do oblastí, kde není používán žádný nástroj s automatizací. Účastníci tohoto testování již zkušenost s jiným programem mají a je to tedy o to cennější zpětná vazba. Na druhou stranu jsou zvyklí na zavedené procesy a jejich reakce tedy může být ovlivněna.

- **Pozitivní body** – Na digitalizaci uživatelé hodnotili nejpozitivněji stručnost a jednoduchost ovládání. Vždy věděli ihned, kam mají kliknout. Nevyskytl se žádný dotaz typu „A co teď?“, což považuji za velký úspěch. Jelikož uživatelé měli zkušenost s podobným systémem, ihned se ptali na způsob korekce špatně ořezaného dokumentu, protože se to v praxi údajně stává poměrně často. Zde velmi ocenili možnost začít znovu, v případě špatně označeného rohu. Druhou věcí, kterou díky své zkušenosti ocenili, bylo tlačítko pro dokončení zpracování, bez zahájení automatického čtení dat. Mnohdy je údajně rychlejší jednoduché formuláře přepsat ručně, než dělat krok navíc. K pamatování původních hodnot měli neutrální postoj, protože tuto funkci obsahoval i původní nástroj. Za největší výhody systému byly však označeny dvě funkcionality, které jim v původním nástroji chyběly. Při nahrání souboru formátu TIF, který obsahuje více snímků, je vytvořena složka, zahrnující všechny snímky očíslované a lze je zpracovat postupně. Druhou ceněnou vlastností bylo, že formulář je po zpracování smazán ze složky k digitalizaci a přesunut na export a není třeba to provádět ručně.
- **Připomínky, náměty a nedostatky** – I zde vzniklo několik požadavků a námětů k úpravě. Prvním je, aby byl po vyexportování zpracovaný formulář označen nebo ideálně přesunut do složky s již exportovanými. Nejprioritnějším požadavkem, který by se dal považovat spíše za chybu, je chybějící možnost zpětně upravit data v elektronické podobě.

Ohledně algoritmu a míry spolehlivosti automatického čtení dat se účastníci vyjádřili poměrně pozitivně oproti jejich stávajícímu systému, avšak i přesto během práce několikrát museli u složitějších formulářů některá políčka opravit ručně. Hodnocení procesu digitalizace za jednotlivé účastníky je k nalezení v tabulce 5.2.

Účastník	Hodnocení
R.K.	8.5
A.B.	8
Iva K.	7.5
M.M.	8.5
Ida K.	9
Průměr	8.2

Tabulka 5.2: Výsledné známky pro digitalizační nástroj a celý proces s ním spojený.

Výkonnostní testování

Do testování digitalizace patří i test výkonnosti systému, ve kterém byla prověřena schopnost automatického čtení dat a její přínos a zároveň míra spolehlivosti.

Prvním testem bylo měření **času stráveného digitalizací** formulářů od 2 pacientů, dohromady se jednalo o 22 vyšetření v balíčcích po 11. Poprvé byla data zpracována za využití automatizace a podruhé bez. Za automatizaci je považováno čtení zaškrtačacích polí a pamatování naposledy uložených klíčových hodnot z předchozího formuláře. Výsledky jsou k nalezení v tabulce 5.3. Nutno dodat, že se jednalo o kvalitně naskenovanou dávku a nebylo třeba ani jednou ručně označovat rohy.

Test	Čas
S využitím automatizace	9 minut a 31 vteřin
Bez využití automatizace	17 minut a 1 vteřina
Ušetřený čas v sekundách	7 minut a 30 vteřin
Ušetřený čas v procentech	44.07%

Tabulka 5.3: Výkonnostní testování provedené na 22 formulářích od dvou pacientů.

Druhou sledovanou metrikou byla **spolehlivost rozpoznávání zaškrtnutých polí**. Měření bylo prováděno na stejném vzorku dat. Z výsledků v tabulce 5.4 lze vidět, že senzitivita, tedy procento rozpoznávaných zaškrtnutých polí je 88.31%. Kdežto podíl správně rozeznávaných prázdných polí je 99.89%. Z toho lze usoudit, že by mohl být zvolen nižší práh, kdy je pole považováno za vyplněné. Nicméně vzorová data byla vyplněna tenkou tužkou. V případě, kdy je do formulářů zapisováno tučnějším perem, mohla by senzitivita být přesnější a na specificitu by to nemělo vliv.

Sledovaný jev	Celkem	TP	FP	TN	FN	Senzitivita	Specificita
Rozpoznání zaškrtnutých polí	2260	272	2	1950	36	88.31%	99.89%

Tabulka 5.4: Testování přesnosti rozpoznávání provedené na 22 formulářích.

S těmito statistikami automatizace bylo ušetřeno během digitalizace přes 44% času, přínos je tedy poměrně značný.

Celkový dojem z platformy

Možná i proto, že účastníci doteď využívali poměrně zastaralý program pro systém Windows, byl jejich dojem z platformy velmi pozitivní. Aplikaci obecně ohodnotili jako moderní a propracovanou. Mimo uživatelské prostředí a přívětivost, se jim velmi zamlouvala volnost práce nad vlastním adresářem, což umožňuje vytvářet například složky s rozpracovanými formuláři a podobně. Kladně hodnotí i postupné přesouvání dokumentu během jeho zpracování skrz platformu. Ohledně systému jako takového byly pouze dvě připomínky. První je triviální, aby při mazání pro jistotu ještě uživatel musel tuto volbu potvrdit.

Největší připomínkou a zároveň velmi kvalitní zpětnou vazbou je, zda by šlo v platformě zavést týmy nebo organizace, pro které budou přiděleni uživatelé o různých oprávněních a budou pracovat nad jedním projektem. Například v tomto případě by byly dvě pracovnice pro digitalizaci, jedna manažerka exportu a jedna návrhářka šablon. V současné chvíli systém neumožňuje více uživatelů nad jedním pracovním prostorem a je to velmi dobrý podnět ke zpracování, který je téměř nutné zapracovat, pokud má být platforma využívána týmově, nebo v organizacích.

Hodnocení celkového dojmu z platformy ukazuje tabulka 5.5.

Účastník	Hodnocení
R.K.	9
A.B.	9.5
Iva K.	9
M.M.	10
Ida K.	9.5
Průměr	9.4

Tabulka 5.5: Hodnocení platformy jako celku dle jednotlivých účastníků.

5.3 Závěr z testování

Uživatelské testování bylo nesmírně přínosné. Bylo cenné a poučné vidět pohled na platformu od lidí, kteří s danou problematikou denně pracují. Systém vnímali jinou perspektivou, hledali způsoby, jak si jejich práci co nejvíce zjednodušit a snažili se odhalit případné problémy, se kterými se denně potýkají a jak jsou řešeny. Výsledkem je výborná zpětná vazba a řada velice kvalitních návrhů a nových požadavků, které jsou dost obecné a uplatnitelné i pro jiné zákazníky. Bohužel z časových důvodů nedojde k další iteraci testování v rámci této diplomové práce, ani implementaci změn. Velmi důležitým zjištěním z testování je, že cíloví uživatelé platformu vnímají velmi pozitivně a rádi ji budou používat. Z testování tedy vznikly další cíle a požadavky na systém, které je potřeba implementovat. Nejprve budou opraveny nedostatky a až následně se přistoupí k doděláním připomínek, či návrhů. Největší prioritu dle uživatelů má v nejbližší době funkce upravení digitalizovaných dat, označování více prvků v designeru přetažením kurzoru myši a práce v týmech pod společnou organizací.

Kapitola 6

Závěr

Cílem této diplomové práce bylo vytvořit webový informační systém, pro zjednodušení procesů spojených se získáváním dat z papírových formulářů. Měl by provázet uživatele všemi fázemi práce a poskytnout funkcionality ke snadnějšímu převodu do elektronické podoby. Řešení bylo cíleno pro konkrétní zdravotnickou organizaci, a proto by práce měla zahrnovat analýzu potřeb této kliniky a testování přímo s cílovými uživateli. Mimo to bylo mým cílem implementovat aplikaci dostatečně obecně na to, aby se dalo do budoucna navázat rozvinutím podnikatelského záměru s využitím i v dalších oblastech.

Stanovené cíle byly dosaženy a v současné době je aplikace dostupná v prototypové verzi na internetové adrese <https://formello.cz>. Výsledná platforma vychází z analýzy požadavků zdravotnického centra a také obecných předpokladů pro práci s tištěnými formuláři. Umožňuje navrhnout a vytisknout libovolný formulář s prvky pro sběr dat a čárovým kódem. Po vyplnění lze oskenované dokumenty hromadně nahrát do systému a převést do elektronické podoby s využitím automatizace ve formě rozpoznávání zaškrtačkových polí. Získaná data lze jednoduše procházet přímo v platformě, která zároveň poskytuje cloudové úložiště. Obsahuje také možnosti exportu dat a je připraveno rozhraní pro integraci s jinými systémy. K implementaci pomocné automatizace při digitalizaci, bylo využito mé bakalářské práce, ze které byl přejet a rozšířen zdrojový kód o funkce potřebné k začlenění do stávající platformy. Samotný algoritmus nebyl upraven.

Bylo provedeno testování přímo s cílovými uživateli, z něhož vzešly velmi cenné poznatky. Systém na uživatele působil intuitivně a přehledně, pozitivní reakce vyvolalo nejen uživatelské rozhraní, ale také praktičnost zahrnutých funkcionalit a míra usnadnění práce. Zpětná vazba byla vyhodnocena a vyplynuly z ní některé návrhy pro zlepšení. Kromě této formy prověření platformy, bylo provedeno i výkonnostní testování, kdy byl měřen ušetřený čas a spolehlivost poskytnutých automatizací. Při běžné práci s daty z cílové zdravotnické kliniky systém ušetří přes 44% stráveného času. To vnímám jako velký přínos.

Dle mého názoru se podařilo implementovat univerzální platformu, která nalezne uplatnění v celé řadě oborů. Jako první bude zavedena do provozu v testované zdravotnické klinice, mělo by k tomu dojít na začátku července 2019. V nejbližší době bych rád implementoval náměty získané během testování, například práci v týmech, nebo některá vylepšení uživatelského rozhraní. V druhé iteraci se chci věnovat samotnému algoritmu pro zpracování obrazu a především zahrnout metodu rozpoznávání ručně psaných čísel, což by přineslo další výrazné urychlení zpracování. Na závěr bych rád dodal, že praktické uplatnění platformy a spolupráce s cílovou klinikou, pro mě byla neocenitelná zkušenost, která mi kromě samotných nově nabytých vědomostí během implementace, dala i možnost získat pohled na problematiku od uživatelů, kteří se s řešeným problémem denně potýkají.

Literatura

- [1] Cohn, M.: *Succeeding with agile*. Addison-Wesley, 2009, ISBN 879-0-321-57936-2.
- [2] Flajšingerová, B.: *Rodina a její role v péči nesoběstačného seniora*. 2008, [Online; navštíveno 11.12.2018].
URL https://is.muni.cz/th/aa48o/bakalarska_prace.doc
- [3] Galloway, J.: *Professional ASP.NET MVC 5*. John Wiley Sons Inc, 2014, ISBN 9781118794753.
- [4] Huňka, F.: *Diagram případu užití*. [Online; navštíveno 4.1.2019].
URL http://www1.osu.cz/~hunka/vyuka/EkF/SoftEnginB/Prednasky/SWb_02x.pdf
- [5] Jurašková, B.: *Funkční geriatrické vyšetření a testy*. [Online; navštíveno 11.12.2018].
URL <https://slideplayer.cz/slide/2729993>
- [6] Pitman, N.: *UML 2.0 in a Nutshell*. O'Reilly Media, Inc., 2005, ISBN 0596007957.
- [7] Šustek, P.: *Webová aplikace pro digitalizaci papírových dotazníků*. 2016, [Online; navštíveno 11.12.2018].
URL <https://dspace.cvut.cz/handle/10467/67506>
- [8] Vaskaran, S.: *Design Patterns in C#: A Hands-On Guide with Real-World Examples*. Apress, 2018, ISBN 9781484236390.
- [9] Zendulka, J.: *Projektování programových systémů - Jazyk UML*. [Online; navštíveno 4.1.2019].
URL http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/7_umlch.pdf