

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Daniel Siažík

Bakalářská práce

Vedoucí práce: Ing. Marek Běhálek, Ph.D.

Ostrava, 2023

Zadání bakalářské práce

Student:

Daniel Siažík

Studijní program:

B0613A140014 Informatika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: AVE Soft s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Marek Běhálek, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního programu: doc. Mgr. Miloš Kudělka, Ph.D.

V IS EDISON zadáno: 07.11.2022 12:25:51

Abstrakt

Během odborné praxe v firmě AVE Soft jsem se zabýval implementací integrace s portálem ČÚZK. Cílem této integrace bylo získání údajů z katastru nemovitostí a následný import těchto údajů do informačního systému Evolio pomocí dálkového přístupu. Součástí mé práce byla také tvorba uživatelské rozhraní nové podatelny.

Klíčová slova

C#, XML, API, Angular

Abstract

During my professional practice at AVE Soft, I was involved in the implementation of integration with the CUZK portal. The goal of this integration was to retrieve data from the Land Registry and import it into the information system Evolio using remote access. I was also responsible for creating the user interface for the new mail client.

Keywords

C#, XML, API, Angular

Poděkování

Rád bych na tomto místě poděkoval především Ing. Martinu Prokešovi a celému vývojovému týmu ve společnosti AVE Soft za jejich podporu a ochotu pomoci mi během psaní této bakalářské práce. V neposlední řadě chci poděkovat také mému vedoucímu práce Ing. Marku Běhálkovi, Ph.D.

Obsah

| | |
|--|-----------|
| Seznam použitých symbolů a zkratk | 7 |
| Seznam obrázků | 8 |
| Seznam výpisů zdrojového kódu | 9 |
| 1 Úvod | 10 |
| 1.1 Firma AVE Soft s. r. o. | 10 |
| 1.2 Prostředí firmy | 10 |
| 2 Použité technologie | 12 |
| 2.1 Angular | 12 |
| 2.2 .NET | 12 |
| 2.3 C# | 13 |
| 2.4 SQL | 13 |
| 2.5 Microsoft Azure | 13 |
| 2.6 Postman | 14 |
| 2.7 Git | 14 |
| 3 Charakteristika úkolů | 15 |
| 3.1 Integrace ČÚZK | 15 |
| 3.2 Tvorba uživatelského rozhraní nové podatelny | 17 |
| 4 Řešení úkolů | 18 |
| 4.1 Integrace ČÚZK | 18 |
| 4.2 Tvorba uživatelského rozhraní nové podatelny | 31 |
| 5 Závěr | 41 |
| 5.1 Uplatněné vědomosti a znalosti | 41 |
| 5.2 Scházející vědomosti a znalosti | 41 |
| 5.3 Celkové zhodnocení praxe | 41 |

| | |
|--|-----------|
| Literatura | 43 |
| Přílohy | 43 |
| A Ukázkový výpis z katastru nemovitostí | 44 |

Seznam použitých zkratek a symbolů

| | |
|------|--|
| ČÚZK | – Český úřad zeměměřický a katastrální |
| DI | – Dependency injection |
| IS | – Informační systém |
| KN | – Katastr nemovistostí |
| LV | – List vlastnictví |
| WSDL | – Web Services Description Language |
| WSDP | – Webové služby dálkového přístupu |
| XML | – eXtensible Markup Language |
| XSD | – XML Schema Definition |

Seznam obrázků

| | | |
|-----|---|----|
| 3.1 | Ukázka modálního okna pro vytvoření listu vlastnictví | 16 |
| 3.2 | Ukázka uživatelského rozhraní staré podatelny | 17 |
| 4.1 | Sekvenční diagram komunikace WSDP klienta se službami | 19 |
| 4.2 | Aktivitní diagram pro stahování listu vlastnictví z KN | 22 |
| 4.3 | Aktivitní diagram pro importování vlastního listu vlastnictví | 23 |
| 4.4 | Modální okno pro placený výpis online | 29 |
| 4.5 | Zobrazení staženého LV v systému Evolio | 31 |
| 4.6 | Wireframe nové podatelny | 33 |
| 4.7 | Diagram komponent zachycující strukturu modulu | 38 |
| 4.8 | Uživatelské rozhraní nové podatelny | 39 |
| A.1 | Ukázka listu vlastnictví | 44 |

Seznam výpisů zdrojového kódu

| | | |
|-----|---|----|
| 4.1 | Pseudokód implementace třídy CUZKService | 20 |
| 4.2 | Pseudokód implementace addinu RequestLV | 24 |
| 4.3 | Pseudokód implementace addinu DownloadLV | 25 |
| 4.4 | Implementace tlačítka pro import vlastního XML | 28 |
| 4.5 | Metoda pro stažení placeného výpisu | 29 |
| 4.6 | Ukázka implementace HTML souboru Inbox komponenty | 34 |
| 4.7 | Ukázka implementace TypeScript souboru Inbox komponenty | 35 |
| 4.8 | Ukázka implementace InboxService komponenty | 36 |

Kapitola 1

Úvod

V úvodu své bakalářské práce bych rád popsal svou náplň a cíle, které jsem se snažil splnit. Byl jsem pověřen vývojem integrace mezi portálem ČÚZK a informačním systémem Evolio. Cílem této integrace bylo získávat údaje z KN a importovat je do IS Evolio s využitím dálkového přístupu. Dále mi byl přidělen úkol, jenž se týkal tvorby uživatelské rozhraní nové podatelny. V závěru mé bakalářské práce se zaměřím na shrnutí získaných poznatků a zkušeností. Popíši, které nové informace a dovednosti jsem získal během svého působení na praxi, a také se zaměřím na oblasti, se kterými jsem se již setkal v průběhu mého studia.

1.1 Firma AVE Soft s. r. o.

AVE Soft je inovativní společnost zabývající se vývojem softwaru. Jejich hlavním produktem je informační systém pro advokátní kanceláře a správu pohledávek nazývaný Evolio. Společnost je na trhu více než 15 let a může se pochlubit oceněními a certifikacemi, které získali díky svým výjimečným produktům. Firma je vedena dvojicí jednatelů Ing. Janem Tkáčem a Ing. Martinem Prokešem.

1.2 Prostředí firmy

Firma AVE Soft se vyznačuje příjemným a přátelským pracovním prostředím s menším počtem zaměstnanců, což umožňuje vzájemné poznání a vytváří příjemnou a uvolněnou atmosféru.

Pro správu úkolů se v této firmě používá kombinace dvou systémů. Prvním z nich je systém Evolio, který jsem používal k vykazování své práce a druhým je platforma GitLab, která se používá ke správě úkolů souvisejících s vývojem a vývojovými projekty. Přes tuto platformu mi byla zadávána většina mých úkolů. Tyto systémy mi pomáhaly efektivně spravovat mé úkoly a zajistit, abych měl lepší přehled o své práci.

Úkoly byly diskutovány s konzultanty nebo členy vývojového týmu, což mi dávalo možnost získat zpětnou vazbu a řešit jakékoli otázky, které vznikly během mé práce. Komunikace probíhala buď osobně v kanceláři, nebo přes online komunikační platformu Microsoft Teams.

Kapitola 2

Použité technologie

2.1 Angular

Angular je open-source framework pro tvorbu moderních webových aplikací. Tento framework je vyvíjen a udržován společností Google a používá se pro vytváření Single Page Applications (SPA) a dalších složitějších webových aplikací. Poskytuje mnoho funkcí pro usnadnění vývoje webových aplikací, jako je například two-way data binding, vkládání závislostí (dependency injection), direktivy (directives), pípy (pipes), a mnoho dalšího. [1]

2.1.1 TypeScript

TypeScript je nadstavba nad jazykem JavaScript, která umožňuje psát kód s typovou kontrolou. Umožňuje programátorům psát rozsáhlejší a složitější kód s menším rizikem chyb v průběhu vývoje a umožňuje lepší správu kódu v rozsáhlejších projektech. TypeScript také umožňuje použití moderních funkcí JavaScriptu, jako jsou třeba šablony řetězců, rozšířené objektové typy, například rozhraní a abstraktní třídy. [2]

2.2 .NET

.NET je softwarová platforma vyvinutá společností Microsoft, která poskytuje vývojářům nástroje pro tvorbu a provoz aplikací. Tato platforma obsahuje řadu knihoven, frameworků a technologií, které usnadňují vývoj aplikací v různých programovacích jazycích, jako je C#, Visual Basic nebo F#. Tyto jazyky nabízejí vysokou míru kompatibility a jednotný přístup k funkcím této platformy. .NET platforma je navržena tak, aby byla multiplatformní, což znamená, že umožňuje vývoj aplikací pro různé operační systémy, jako jsou Windows, Linux nebo macOS. [3]

2.3 C#

C# je objektově orientovaný programovací jazyk, který je určen pro vývoj aplikací na platformě Microsoft .NET. Jazyk C# byl poprvé vydán v roce 2000 a od té doby se stal jedním z nejpoužívanějších jazyků na světě. C# podporuje mnoho různých paradigmat programování, včetně objektového programování, funkcionálního programování a imperativního programování. Jazyk také obsahuje řadu funkcí, jako jsou například automatické typování, lambda výrazy a LINQ, které umožňují vývojářům vytvářet efektivní a elegantní kód. [4]

2.4 SQL

SQL (Structured Query Language) je standardizovaný strukturovaný dotazovací jazyk používaný k manipulaci s daty v relačních databázích. SQL nám umožňuje vytváření komplexních dotazů a operací nad daty, jako je výběr, vkládání, aktualizace a mazání záznamů. SQL dotazy se dělí na čtyři základní skupiny:

- DDL (Data Definition Language): Tyto příkazy se používají k definování a modifikaci struktury databáze (CREATE, ALTER, DROP).
- DML (Data Manipulation Language): Tyto příkazy se používají k manipulaci s daty (SELECT, INSERT, UPDATE, DELETE).
- DCL (Data Control Language): Tyto příkazy se používají k řízení přístupu k datům (GRANT, REVOKE).
- TCC (Transaction Control Commands): Tyto příkazy se používají k definování a řízení transakcí (COMMIT, ROLLBACK).

SQL se s postupem času vylepšilo a nabylo nových funkcí, včetně možnosti definovat a řídit nové objekty jako indexy, pohledy, triggerů a uložené procedury. Tyto funkce přibližují SQL charakteru programovacího jazyka. [5]

2.5 Microsoft Azure

Microsoft Azure je cloudová platforma od společnosti Microsoft, která poskytuje širokou škálu služeb pro vývoj, testování a nasazení aplikací a služeb v cloudu. Azure umožňuje uživatelům rychle vytvářet a provozovat aplikace a webové služby bez nutnosti spravovat vlastní fyzickou infrastrukturu a hardware. [6]

2.5.1 Azure Service Fabric

Azure Service Fabric je platforma pro vývoj a provoz rozsáhlých, škálovatelných a spolehlivých aplikací v cloudu. Umožňuje snadno vytvářet rozsáhlé a složité aplikace pomocí mikroslužeb a kontejnerů. Díky tomu lze aplikace snadno škálovat a aktualizovat bez nutnosti vypínání celé aplikace. [7]

2.6 Postman

Postman je nástroj sloužící k testování, dokumentování a vývoji API (Application Programming Interface). Umožňuje uživatelům vytvářet HTTP požadavky, odesílat je na různé endpointy a zobrazovat odpovědi od serveru. Postman také umožňuje ukládat a organizovat kolekce požadavků, sdílet je s jinými uživateli a generovat dokumentaci API. [8]

2.7 Git

Git je distribuovaný systém správy verzí, který slouží k ukládání, sledování a správě změn v kódu a souborech projektu. Umožňuje více vývojářům pracovat na stejném projektu současně a integrovat své změny bez konfliktů. Git také umožňuje snadnou zpětnou kontrolu vývoje projektu a umožňuje obnovit jakoukoli předchozí verzi kódu. [9]

2.7.1 Milestones

Milestones (Milníky) jsou významné body v historii vývoje projektu, které slouží k organizaci a sledování úkolů a cílů v rámci projektu. Tyto body jsou obvykle definovány specifickými cíli, například dokončením určité funkcionality, řešením konkrétního problému nebo splněním určitého termínu.

Milestones v Gitu umožňují vývojářům a týmům sledovat a spravovat vývoj projektu v časové ose. Přidávání a označování milníků v Gitu může pomoci při plánování a organizaci projektů, zlepšit komunikaci mezi týmy a poskytnout větší transparentnost a viditelnost projektu. [10]

Kapitola 3


Charakteristika úkolů

V této kapitole se zaměřím na popis úkolů, které mi byly během praxe zadány. Budou zde uvedeny jednotlivé úkoly spolu s jejich popisem a cíli.

3.1 Integrace ČÚZK

Cílem tohoto úkolu je implementovat integraci, jenž má zahrnout funkčnost pro stažení a aktualizaci listu vlastnictví z databáze KN a následné vytváření revizí těchto dat. Pro stažení dat bude využito webových služeb pro dálkový přístup. Uživateli bude umožněna plná kontrola nad aktualizacemi dat a bude mít možnost stahovat další placené výstupy dle vlastního uvážení. Dále bude implementována možnost importovat vlastní XML soubor, získaný z jiného zdroje.

Informační systém Evolio již disponuje uživatelským rozhraním, umožňující manuální zadání dat a vytvoření listu vlastnictví.



LV:

Pokud list vlastnictví importujete z Katastru nemovitostí, budou se automaticky vytvářet revize.

ZÁKLADNÍ ÚDAJENEMOVITOSTI

| | | |
|-------------------------|--|----------------------------------|
| Číslo listu vlastnictví | <input type="text"/> | NAČÍST VÝPIS Z KN ▾ |
| Obec | <input type="text" value="Obec... (min. 3 znaky)"/> | <input type="button" value="🔍"/> |
| Katastrální území | <input type="text" value="Katastrální území... (min. 3 znaky)"/> | <input type="button" value="🔍"/> |
| Pracoviště | <input type="text" value="Pracoviště"/> | |
| Stav k datu | <input type="text" value="26.03.2023"/> | |

Vlastníci, oprávněné osoby

[+ Přidat vlastníka](#)

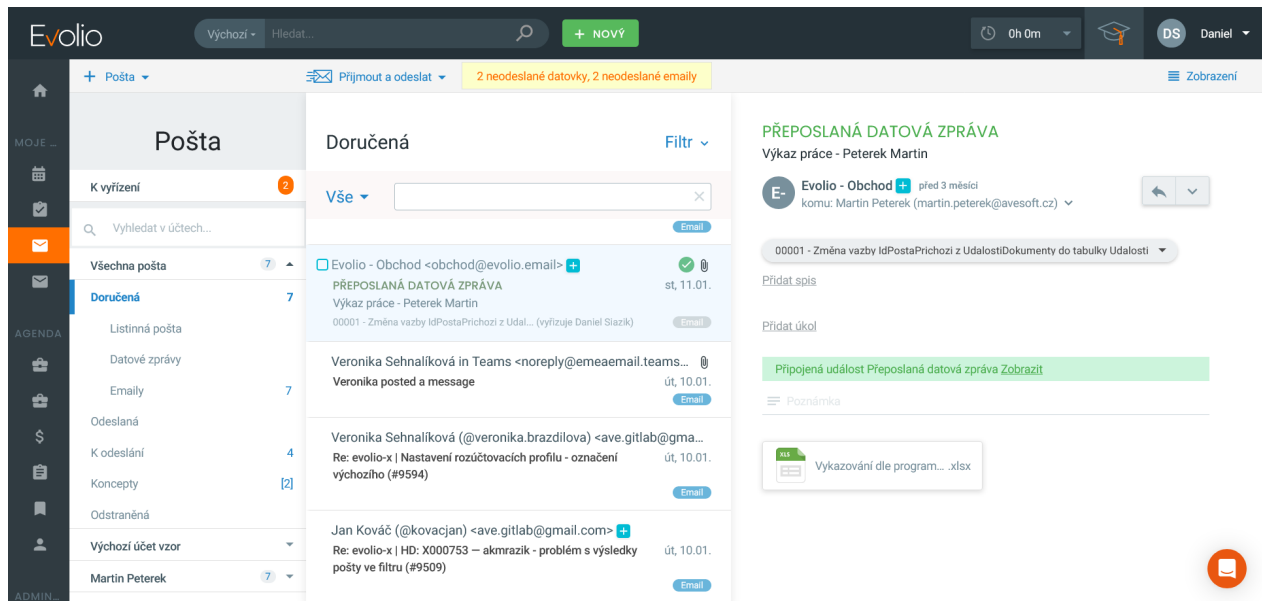
Obrázek 3.1: Ukázka modálního okna pro vytvoření listu vlastnictví

3.2 Tvorba uživatelského rozhraní nové podatelny

Cílem úkolu bylo navrhnout uživatelské rozhraní pro novou podatelnu, která bude sloužit k efektivnější správě datových zpráv, e-mailů a listinné pošty. Nová podatelna nahrazuje starou podatelnu, která se potýkala s problémy vysokého počtu requestů, což mělo negativní vliv na její výkon.

Pro získání veškerých potřebných dat bude nová podatelna využívat procedury. Tento přístup umožní rychlejší získávání dat a snížení počtu requestů.

Rád bych také zmínil, že na této části jsem pracoval společně s Ing. Zdeňkem Pozněrem, který mi mi byl nápomocen při implementaci, a také mi poskytl cenné rady, které mi pomohly lépe pochopit principy Angularu.



Obrázek 3.2: Ukázka uživatelského rozhraní staré podatelny

Kapitola 4

Řešení úkolů

4.1 Integrace ČÚZK

V této kapitole se zaměříme na analýzu, implementaci a testování integrace.

4.1.1 Analýza

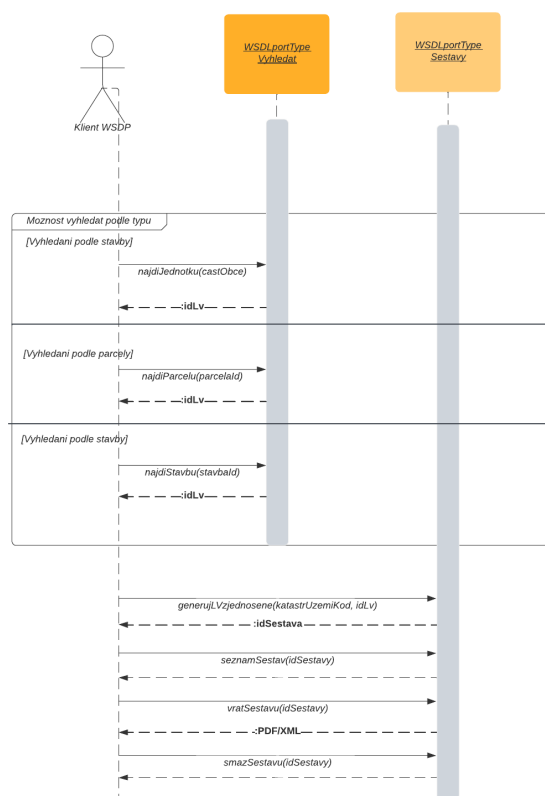
V rámci analýzy webových služeb bylo zjištěno, že rozhraní je definováno pomocí standardů Web Service Definition Language (WSDL) a XML Schema Definition Language (XSD). WSDL je XML formát, který se používá k popisu funkcionalit, které jsou poskytovány webovou službou. Obsahuje informace o tom, jaké operace jsou k dispozici, jakým způsobem se na službu přistupuje, jaké jsou datové typy vstupů a výstupů a jaké jsou požadavky na komunikaci s webovou službou. XSD je jazyk pro definování struktury datových souborů v XML formátu. Určuje, jaké elementy a atributy může XML dokument obsahovat a jaké jsou povoleny jejich hodnoty. Pro zabezpečení volání webových služeb je využit mechanismus Web Service Security (WSS), který zajišťuje autentizaci a autorizaci uživatelů při každém dotazu.

Sestavy jsou vráceny asynchronně, tj. služba po zařazení sestavy do fronty vrátí ID sestavy a uživatel si ji může po vytvoření pomocí služby přehledu sestav odebrat. Vstupem pro odebrání sestavy je ID sestavy. Pokud jsou splněny všechny předpoklady, je výstupem funkce vygenerovaný dokument ve formátu PDF nebo XML, který je součástí odpovědi. Pro přenos příloh je využita specifikace MTOM.

Výpis z katastru nemovitostí dokládá existenci shodného právního vztahu typu vlastnictví pro jednoho nebo více oprávněných subjektů k jedné nebo více nemovitostem. V částech označených A až F obsahuje:

- **Část A:** Označení právního vztahu a údaje o vlastníkovi nebo spoluvlastnících a jejich podílech nebo jiných oprávněných subjektech s uvedeným právním vztahem k nemovitostem uvedených v části B.

- **Část B:** Údaje o nemovitostech, tj. parcelách, parcelách evidovaných zjednodušeným způsobem, budovách, bytových jednotkách a nebytových prostorech. Stojí-li na pozemku cizí budova, je uvedeno i číslo LV, na kterém je budova zapsána jako vlastnictví. Obdobně je-li předmětem vlastnictví budova, nikoliv však pozemek, na kterém stojí, je uveden odkaz na číslo LV, na němž je zapsáno vlastnictví pozemku. Část B1 je určena pro zápisy jiných věcných práv, která mají subjekty uvedené v části A k cizím nemovitostem.
- **Část C:** Omezení vlastnických práv (břemena, zástavy apod.) k nemovitostem zapsaným v části B.
- **Část D:** Zápisy technické povahy.
- **Část E:** Výčet nabývacích titulů, tj. listin vztahujících se k vlastnictví nemovitostí v části B.
- **Část F:** Týká se pouze zemědělských pozemků a obsahuje jejich zařazení do bonitních půdně ekologických jednotek.



Obrázek 4.1: Sekvenční diagram komunikace WSDP klienta se službami

Pro vytvoření korespondující třídy výpisu z katastru nemovitostí bude použit nástroj *xsd2code*. Nástroj *xsd2code* slouží pro generování kódu ze schématu XSD. Konkrétně se jedná o generátor tříd pro jazyky C#, VB.NET, Java, TypeScript, Swift a další.

4.1.2 Implementace

V první fázi implementace bylo nezbytné vytvořit samostatnou třídu CUZKService, která slouží k propojení se službou a zajišťuje vazbu s mechanismem WSS pro zabezpečený přenos dat. Tato třída dále poskytuje metody pro tvorbu požadavků na službu na základě vstupních parametrů.

Dále tato třída obsahuje metodu pro vytvoření WSDP klienta, který přijímá vstupní parametry uživatelské údaje, koncový bod (endpoint) a vytvořený binding (vazbu). Tento klient slouží k zajištění komunikace s webovou službou a umožňuje využití metod této služby prostřednictvím aplikace.

Class CUZKService:

```
dataContext: EMapContext
```

```
Constructor(dataContext: EMapContext):
```

```
    this.dataContext = dataContext
```

```
Async Method GetCredentials() -> (string, string, string):
```

```
    credentials = await dataContext.credentials()
```

```
    if credentials?.UserName == null or credentials?.Password == null:
```

```
        throw new Exception("Nenalezeny uživatelské údaje")
```

```
    return (credentials.UserName, credentials.Password, endpoint)
```

```
Method InitializeBinding() -> CustomBinding:
```

```
    Security = CreateUserNameOverTransportBindingElement()
```

```
    Security.MessageSecurityVersion =
```

```
        WSSecurity10WSTrustFebruary2005WSSecureConversationFebruary2005WSSecurityPolicy11Bas
```

```
    Security.IncludeTimestamp = false
```

```
    TextMessageEncoding = TextMessageEncodingBindingElement()
```

```
    TextMessageEncoding.MessageVersion = Soap11
```

```
    HttpsTransport = HttpsTransportBindingElement()
```

```
    HttpsTransport.MaxReceivedMessageSize = 20000000
```

```
    HttpsTransport.MaxBufferSize = 20000000
```

```

binding = CustomBinding()
binding.Elements.Add(Security)
binding.Elements.Add(TextMessageEncoding)
binding.Elements.Add(HttpsTransport)

return binding

```

```

Async Method GetClientSestavy() -> sestavyClient:
    (username, password, endpoint) = await GetCredentials()

    client = sestavyClient(InitializeBinding(), EndpointAddress(endpoint))

    client.ClientCredentials.UserName.UserName = username
    client.ClientCredentials.UserName.Password = password

return client

```

Výpis 4.1: Pseudokód implementace třídy CUZKService

V pseudokódu výše je uvedená proměnná *dataContext*, která slouží jako spojovací uzel mezi aplikací a databází. Je implementován v rámci datové vrstvy aplikace a umožňuje nám pracovat s entitami, jako by se jednalo o běžné objekty. *DataContext* obsahuje definice entit, mapování mezi třídami entit a databázovými tabulkami, správu životního cyklu entit (např. vytváření, čtení, aktualizace, odstraňování) a transakční řízení. Umožňuje také provádět dotazy na databázi pomocí jazyka LINQ (Language-Integrated Query) a jiných mechanismů.

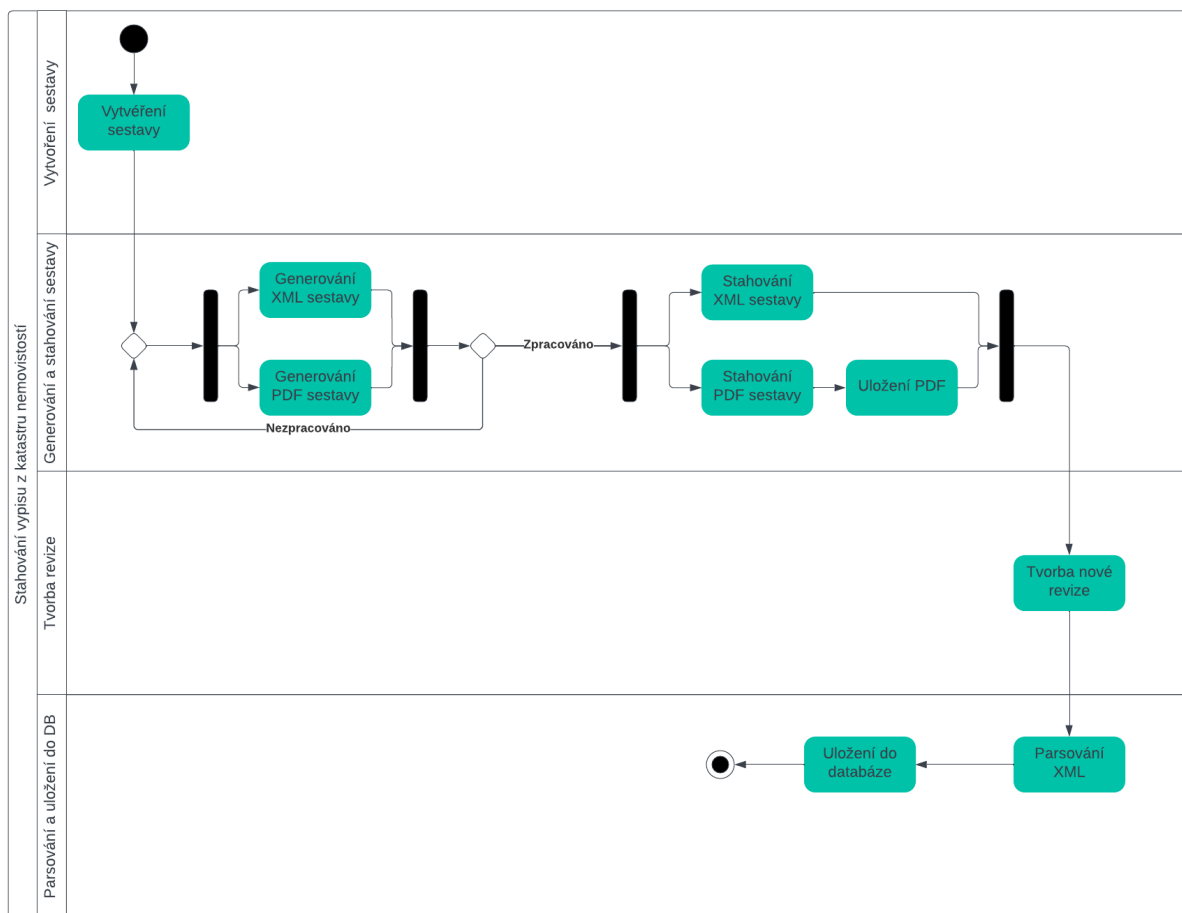
Zároveň zde využíváme návrhový vzor *Dependency injection*, který umožňuje oddělit vytváření a konfiguraci objektů od jejich použití. To znamená, že třída nevytváří a nekonfiguruje své závislosti sama, ale místo toho získává předpřipravené instance pomocí DI kontejneru, který tyto závislosti spravuje.

Pro implementaci stahování výpisu z katastru nemovitostí se nabízí rozdělit tento proces na čtyři části (mikroslužby), z nichž každá bude zodpovědná za jednu specifickou činnost:

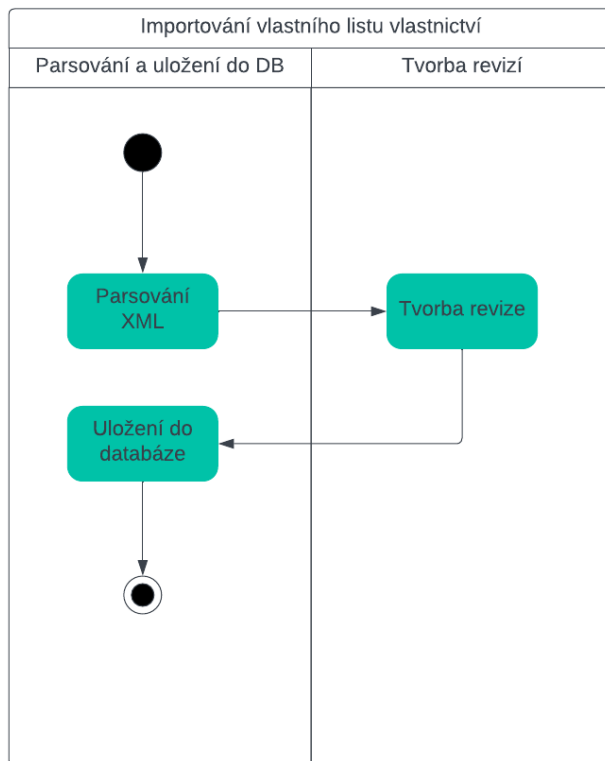
- **Vytvoření sestav** - Tato fáze zahrnuje vytvoření sestav obsahujících požadované informace pro výpis z katastru nemovitostí.
- **Generování sestav** - V této fázi se budou generovat sestavy ve formátech XML a PDF. PDF soubor se po stažení ihned uloží.
- **Tvorba revize** - Tato část se bude věnovat vytvoření revize staženého listu vlastnictví.

- **Parsování a uložení do DB** - V poslední fázi se bude parsovat dokument ve formátu XML a následně je potřeba tyto data uložit do databáze.

Tohle rozložení nám umožňuje rozdělit aplikaci na menší a nezávislé části, které mohou společně komunikovat.



Obrázek 4.2: Aktivitní diagram pro stahování listu vlastnictví z KN



Obrázek 4.3: Aktivitní diagram pro importování vlastního listu vlastnictví

Dále budeme implementovat tzv. Addin, což je modul navržený pro rozšíření funkcí a možností existujícího webového rozhraní API. Addin představuje samostatnou třídu, která bude zapouzdřovat funkcionalitu nových API endpointů.

```
class RequestLV: Integration
    cuzkService: CUZKService

    constructor(dataContext: EMapContext, cuzkService: CUZKService)
        super(dataContext)
        this.cuzkService = cuzkService

    async method ExecIntegration(lvInputParams: LVInputParams) -> Result
        client = await cuzkService.GetClientSestavy()
        try
            Log.Information("Stahování listu vlastnictví č. {0}", lvInputParams.lvID)

            task = await client.generujLVZjednoduseneAsync(
                cuzkService.GenerateLVRequestType(lvInputParams.lvID, lvInputParams.
                    uzemiKod, lvInputParams.datum, FormatSestavyType.pdf, true))
            response = task.GenerujSestavuResponse

            if response is null then
                return Error("Nepodařilo se stáhnout požadovaný list vlastnictví.")

            if response.vysledek.FirstOrDefault().kod != "0" then
                return Warning(response.vysledek.FirstOrDefault()?.Value)

            if response.reportList is null or response.reportList.Length == 0 then
                return Error("Vygenerované LV je prázdné.")

            xml = response.reportList.Where(x -> x.format == FormatSestavyType.xml).
                FirstOrDefault()
            pdf = response.reportList.Where(x -> x.format == FormatSestavyType.pdf).
                FirstOrDefault()

            idXML = xml.id
            idPDF = pdf.id
            Log.Information("Získáno ID xml({0}) a pdf({1}) souboru", idXML, idPDF)
```



```

if idXML is null or idPDF is null then
    return Error("Nepodařilo se najít ID sestavu")

downloadLVParams = new DownloadLVInputParams()
downloadLVParams.idPDF = idPDF
downloadLVParams.idXML = idXML
downloadLVParams.parameters = lVInputParams

return Forward<DownloadLV, downloadLVParams>(DownloadLVInputParams)

catch exception
    return Error(exception.Message)

finally
    client.CloseAsync()

```

Výpis 4.2: Pseudokód implementace addinu RequestLV

V prvním addinu vidíme, že dědí ze třídy *Integration*. To znamená, že před jeho samotným spuštěním se prvně zkontroluje, zda-li je integrace pro stahování listu vlastnictví aktivní. Poté se již spustí *ExecIntegration*. V této metodě se nejprve získá instance klienta z třídy *CUZKService* (viz 4.1) voláním metody *GetClientSestavy()*. Dále se snaží stáhnout list vlastnictví pomocí volání asynchronní metody na klientovi.

Následně se kontroluje, zda byl požadavek úspěšně zpracován (*response* není null) a zda výsledek neobsahuje chybu (vlastnost *kod* v prvním prvku pole *vysledek* je "0"). Pokud nastane nějaká chyba, vrátí se uživateli odpovídající chybová zpráva. V opačném případě se pokračuje sestavením vstupních parametrů pro následný addin *DownloadLV*.

```

class DownloadLV : Integration
    cuzkService: CUZKService

    DownloadLV(dataContext: EMapContext, cuzkService: CUZKService):
        base(dataContext)
        cuzkService = cuzkService

    async method ExecIntegration(downloadLVParams: DownloadLVInputParams) -> Result:
        client = await cuzkService.GetClientSestavy()

        iterations = 0

```

```

try:
    stateXML = ReportStavType.ceka
    statePDF = ReportStavType.ceka
    Log.Information("Stahuji XML a PDF soubory.")

while (stateXML != ReportStavType.zpracovan or stateXML != ReportStavType.
    zauctovan) and (statePDF != ReportStavType.zpracovan or statePDF !=
    ReportStavType.zauctovan) and iterations++ < 6:
    responsePDF = await cuzkService.GenerateResponseSestava(cuzkService.
        GenerateSestavuRequestType(downloadLVParams.idPDF))
    responseXML = await cuzkService.GenerateResponseSestava(cuzkService.
        GenerateSestavuRequestType(downloadLVParams.idXML))

    xml = responseXML.reportList.Where(x -> x.format == FormatSestavyType.xml)
        .FirstOrDefault()
    pdf = responsePDF.reportList.Where(x -> x.format == FormatSestavyType.pdf)
        .FirstOrDefault()

    stateXML = xml.stav
    statePDF = pdf.stav

if (stateXML == ReportStavType.zpracovan or stateXML == ReportStavType.
    zauctovan) and (statePDF == ReportStavType.zpracovan or statePDF ==
    ReportStavType.zauctovan):
    resultXML = await client.vratSestavuAsync(downloadLVParams.idXML)
    bytesXML = resultXML.GenerujSestavuResponse.reportList.FirstOrDefault().
        souborSestavy

    resultPDF = await client.vratSestavuAsync(downloadLVParams.idPDF)
    bytesPDF = resultPDF.GenerujSestavuResponse.reportList.FirstOrDefault().
        souborSestavy
    Log.Information("XML a PDF soubory jsou staženy.")

    price = resultPDF.GenerujSestavuResponse.reportList.FirstOrDefault()?.
        cena

    idPDFDoc = await this.DataContext.WriteDocument($"LV: {downloadLVParams.
        parameters.lvID} {downloadLVParams.parameters.datum.ToString("dd.MM.

```

```

        yyyy").pdf", bytesPDF, true)
Log.Information("PDF je uložen do DB.")

saveInpuPparameters = SaveInputParams(
    xml=bytesXML,
    parameters=downloadLVParams.parameters,
    idXML=downloadLVParams.idXML,
    idPDF=idPDFDoc, price=price
)

return Forward<SaveXML, SaveInputParams>(saveInpuPparameters)

if stateXML == ReportStavType.chybaprivytvareni or stateXML ==
    ReportStavType.chybaprizpracovani or stateXML == stateXML ==
    ReportStavType.zrusen
return Error("Došlo k chybě při vracení XML sestavy.");

if statePDF == ReportStavType.chybaprivytvareni or statePDF ==
    ReportStavType.chybaprizpracovani or statePDF == statePDF ==
    ReportStavType.zrusen
return Error("Došlo k chybě při vracení PDF sestavy.");

await Task.Delay(iterations * 2000);

return Error("Sestavu se nepodařilo vytvořit.");

catch exception
return Error(exception.Message)

finally
client.CloseAsync()

```

Výpis 4.3: Pseudokód implementace addinu DownloadLV

Addin *DownloadLV* se stará zejména o stahování obou sestav. Konkrétně se zde volají metody *GenerateResponseSestava* a *VratSestavuAsync*. Zde dochází ke zpracování stavu sestav. Dokud stav XML souboru nebo PDF souboru není buď *zpracován* či *zaúčtován*, metoda stále opakuje získávání souborů a kontrolování jejich stavu. Pokud se stav souborů změní na *zpracován* nebo *zaúčtován*, metoda uloží PDF do databáze a předá kontrolu další části, která se zabývá tvorbou revize.

Addin starající se o tvorbu revizí nejprve zkontroluje, zda již existuje záznam se stejným listem vlastnictví a katastrálním územím. Pokud ano, vytvoří se nová revize a XML data se uloží k této nové revizi. Pokud takový záznam neexistuje, je potřeba nejprve založit nový záznam a poté vytvořit první revizi k tomuto záznamu. Po vytvoření revize se XML soubor předává další části zpracování, která se zabývá parsováním dat. Jakmile jsou data zparsována, jsou uložena do databáze.

4.1.3 Úprava stavajícího uživatelského rozhraní

Nezbytnou součástí bylo napojení uživatelské rozhraní na nově vytvořenou integraci. Nejprve bylo nutné aktivovat dvě tlačítka, a to *Importovat vlastní XML* a *Placený výpis online*. Po stisknutí tlačítka *Importovat vlastní XML* se uživateli zobrazí možnost vybrat soubor.

```
importXML(event: any): void {
  let fileName = event.target.files[0].name;
  let parts = fileName.split('.');
  if(parts[parts.length - 1] !== 'xml') {
    this.notify.warning('Špatná přípona souboru.');
```

Špatná přípona souboru.

```
    this.xmlInput.nativeElement.value = "";
    return;
  }
  this.loading = true;
  let fileReader = new FileReader();
  fileReader.onloadend = (e) => {
    this.REService.importXML$(fileReader.result, this.idCase, null).subscribe(
      (res) => {
        this.emapData.loadCaseLVs(true);
        this.xmlInput.nativeElement.value = "";
        this.loading = false;
      },
      (err) => {
        this.xmlInput.nativeElement.value = "";
        this.loading = false;
      }
    );
  };
  let arraybuffer = fileReader.readAsArrayBuffer(event.target.files[0]);
}
```

Výpis 4.4: Implementace tlačítka pro import vlastního XML

Tento kód začíná kontrolou přípony souboru a následně pokračuje voláním addinu (*ParseXML*), který provádí deserializaci XML podle XSD souboru. V případě úspěšné deserializace se spouští addin, který vytváří nové revize a poté se kód opět vrátí do *ParseXML*, kde jsou data následně uložena do databáze. (viz 4.3)

Při stisknutí tlačítka *Placený výpis online* bylo nejprve potřeba vytvořit modální okno, aby uživatel mohl zadat vstupní data.



Importovat LV z online přístupu
UPOZORNĚNÍ! Za výpis z LV si ČÚZK si účtuje z účtu, kterým jste přihlášení.

Číslo listu vlastnictví: 208

Obec: Vacov (okres Prachatice)

Katastrální území: Benešova Hora

Pracoviště: Katastrální úřad pro Jihočeský kraj, Katastrální prac

Stav k datu: 15.04.2023

NAČÍST VÝPIS Z KN

Obrázek 4.4: Modální okno pro placený výpis online

```
requestLV(): void {
    if (!this.integration) {
        this.notify.warning('Integrace pro stáhování LV je vypnutá.');
```

```
        return;
    }

    if (!this.idCase || !this.LV || !this.selectedObce || !this.selectedUzemi || !
        this.selectedPracoviste) {
        this.notify.warning('Pro načtení z katastru je nutné vyplnit LV, obec,
            katastrální území a pracoviště');
```

```
        return;
    }

    if (!this.LV.cisloLV || !this.selectedObce.idKUObec || !this.selectedUzemi.
        idKUUzemi || !this.selectedPracoviste.idKUPracoviste) {
        this.notify.warning('Nesprávný formát vstupních polí.');
```

```
        return;
    }
}
```

```

}

this.modal.confirm2('Opravdu chcete načíst LV ${this.LV.cisloLV} z Katastru
nemovitostí ?').then(
(res) => {
  this.loading = true;
  this.REService.requestLV$(
    this.idCase,
    this.LV.cisloLV,
    this.selectedObce.idKUObec,
    this.selectedUzemi.idKUUzemi,
    this.selectedPracoviste.idKUPracoviste,
    this.stateToDate
  ).subscribe(
    (res) => {
      this.loading = false;
      this.dialog.close();
    },
    (err) => {
      this.loading = false;
    }
  );
},
(err) => {
  this.loading = false;
  return;
}
);
this.loading = false;
}

```

Výpis 4.5: Metoda pro stažení placeného výpisu

Modální okno obsahuje pole pro číslo LV, sloužící pro identifikaci daného LV, pole pro obec, katastrální území, pracoviště, a data, ke kterému má být sestaveno.

Po zadání vstupních dat se zavolá metoda, která nejprve zkontroluje zdali je integrace aktivní. Následně se provede kontrola, zda-li jsou všechny vstupní údaje vyplněné a zavolá addin *RequestLV* (viz 4.2).

4.1.4 Testování

Testování bylo prováděno pomocí nástroje Postman, který umožňuje testování webových služeb a API. V rámci testování a ladění klientských aplikací jsou zavedeny webové služby i v databázi na zkoušku. Data v této zkušební verzi jsou záměrně pozmeněna tak, aby neodpovídala platnému stavu KN. Celý proces byl podrobně otestován na testovacích doménách společně s uživatelským rozhraním, aby se zajistilo, že vše funguje správně, než bude vše nasazeno do produkčního prostředí.

The screenshot shows the Evolio system interface for a specific land charge (LV: 208). The interface includes a navigation bar with tabs like 'PŘEHLED', 'FINANCE', 'AKTIVITY 32', 'ÚKOLY 3', 'REALITY', 'NEMOVITOSTI', 'DOKUMENTY 3', 'VYKAZOVÁNÍ', and 'DEPOZITA +'. A green button 'PŘIDAT LV A NEMOVITOST' is visible in the top right. The main content area displays a table of land parcels (LV) under the charge LV: 208. The table has columns for 'TYP', 'EXEKUCE', 'VĚCNÉ BŘEMENO', 'ZÁSTAVY', and 'ZMĚNA'. The data rows are as follows:

| TYP | EXEKUCE | VĚCNÉ BŘEMENO | ZÁSTAVY | ZMĚNA |
|----------------------|---------|--|---------|------------|
| Pozemek 11/3 + Vacov | 0 | Věcné břemeno chůze: povinnost k: 11/1, oprávnění pro: 11/3 Věcné břemeno brání vody: povinnost k: 11/1, oprávnění pro: 11/3 | 0 Kč | 15.04.2023 |
| Pozemek 1 + Vacov | 0 | | 0 Kč | 15.04.2023 |
| Pozemek 675 + Vacov | 0 | | 0 Kč | 15.04.2023 |

Obrázek 4.5: Zobrazení staženého LV v systému Evolio

4.2 Tvorba uživatelského rozhraní nové podatelny

V následující části se zaměřím na tvorbu uživatelského rozhraní pro novou podatelnu.

4.2.1 Analýza

V této části se zaměřím na analýzu tvorby uživatelského rozhraní nové podatelny. Ta byla provedena až po vytvoření wireframu a po konzultacích s UI designerem. Z této analýzy jsme dospěli k závěru, že nejlepší způsob, jak rozdělit modul, je do čtyř komponent.

Nová podatelna bude sloužit jako správce pošty pro všechny typy zpráv, včetně e-mailů, datových zpráv a listinné pošty. Bude tak nahrazovat starou podatelnu, což přinese výhodu snížení počtu requestů na backend. Tento efekt bude dosažen hlavně díky tomu, že data budou primárně načítána z SQL procedur. Podatelna bude přinášet mnoho nových funkcí a vylepšení, které zlepší uživatelskou

zkušenost a efektivitu práce. Mezi nové prvky patří například možnost vytvoření vlastních složek, přesunutí do jiné složky. Další novinkou bude zobrazování předchozích aktivit s touto zprávou.

První komponentou je hlavní menu, které obsahuje konkrétní poštovní účty. Každý účet může obsahovat emaily, datové zprávy nebo listinnou poštu a má předem stanovené složky. Uživatel bude mít také možnost vytvořit si vlastní složku. Jako výchozí složka bude *Pošta k vyřízení*, která bude obsahovat poštu obsahující nevyřízené úkoly. Dále zde bude možnost vyhledávání mezi účty, přidat nebo upravit účet.

Druhou komponentou je seznam zpráv, který bude obsahovat zprávy v dané složce. Každá zpráva bude nabízet různé akce, jako je smazání zprávy nebo označení zprávy jako vyřízená. Uživatel bude mít také možnost označit více zpráv a provést hromadnou akci, jako je smazání, označení jako přečtené/nepřečtené nebo přesunutí do jiné složky. Přesunutí bude možné aplikovat pomocí drag and drop operace nebo si uživatel může ručně vybrat, kam má být pošta přesunuta.

Třetí komponentou je detail zprávy, která umožní uživateli zobrazit informace týkající se konkrétní zprávy. V této komponentě se také zobrazují úkoly, které jsou vázány na danou zprávu, přílohy a předchozí aktivity spojené s touto zprávou. Aktivity mohou zahrnovat přeposlání zprávy, odpověď na zprávu, přiřazení zprávy ke spisu, přidání a splnění úkolu. Bude zde také tlačítka pro přeposlání, odeslání či vytisknutí pošty.

Poslední komponentou je navigační panel, který umožní uživateli provádět hromadné akce nad zprávami a vyhledávat konkrétní zprávy v seznamu. Tento panel bude umístěn v horní části aplikace.

Navigation and utility icons: + NOVÁ ZPRÁVA, PRJMKOUT, Hledat ve zprávách a dokumentech..., Oznáči jako nepřečtené, Odstranit, Odpovědět, Přeposlat, ...

Doručená

Filter

| Avatar | Name | Status | Time |
|--------|---|--------|------|
| | Okresní soud Brčclav Rozhodnutí ve věci | ✓ | 9:30 |
| | Josef Novák Němá pratistany | ✓ | 9:30 |
| | AK Těpý, Velký & Vtřásek IKS KB mění název na Amundi OPPOK - ActiveCampaign - incoy Česká rep (vyřizuje Jan Tkač) | ✓ | 9:30 |
| | Ministerstvo spravedlnosti Rozhodnutí soudu: EPR 106001/2019-6; Usnesení | ✓ | 9:30 |

EPR 106001/2019-6; Usnesení

Bez přílohy

Náš sp. značka: EX 457/22
Náš č. jednací: EX 457/22-12
Do věstebních tabulek: Ne
Základní filtr: Ne

Vše sp. značka
Vše č. jednací
Základní filtr

Přidělit k vyřízení Poznámka

Plná moc Komerční banka v.2.2.docx (56 kB)
Zamyšlení o dalším strategickém postupu.docx (123 kB)

Stáhnout všechny přílohy

AKTIVITY K POŠTĚ

14.4.2022 12:31 Odchozí email Jiří Nováček • FW: EPR 106001/2019-6; Usnesení
14.4.2022 10:12 Předáno k vyřízení Jiří Nováček

Buttons: ODPOVĚDĚT, PŘEPOSLAT, VYTISKNOUIT, STÁHNOUT DORUČENKU

Obrázek 4.6: Wireframe nové podatelny

4.2.2 Implementace

V rámci implementační části se budeme věnovat tvorbě hlavních komponent aplikace. Každá komponenta se skládá ze tří hlavních souborů: TypeScript souboru (s příponou `.ts`), SCSS souboru (s příponou `.scss`) a HTML souboru (s příponou `.html`).

Pro lepší organizaci jsme vytvořili tzv. wrapper (obal), kterým bude komponenta *Inbox*. Tato komponenta bude obsahovat výše vyjmenované komponenty. Dále bude obsahovat tzv. Service. Service je jednou z klíčových komponent pro sdílení dat mezi různými částmi aplikace. Je to třída, která poskytuje funkcionalitu, jenž není přímo vázaná na žádnou konkrétní komponentu. Service může být použita pro ukládání dat, poskytování funkcí pro interakci s externími API nebo například cachování dat.

```
<div class="inbox" [class.open]="isOpen">
  <ev-inbox-navbar
    (onMenuForceOpen)="handleMenuForceOpen()"
  ></ev-inbox-navbar>
  <div class="wrapper drag_boundary">
    <ev-inbox-menu
      (onFolderDrag)="handleMailsFolderChange($event)"
      [accounts] = "accounts"
    ></ev-inbox-menu>
    <ev-inbox-list
      class="wrapper-item"
      [class.animate]="allowAnimations"
      (onDelete)="handleMailDelete($event)"
      (onDealWith)="handleMailDealWith($event)"
    ></ev-inbox-list>
    <ev-inbox-detail
      [class.fullscreen]="getFullscreen()"
      [opacity]="isMailOpen ? 1 : 0"
      [showLoading]="detailLoading"
      (onTaskClick)="handleTaskClick()"
      (onTaskOpen)="handleTaskOpen()"
    >
  </ev-inbox-detail>
</div>
</div>
```

Výpis 4.6: Ukázka implementace HTML souboru Inbox komponenty

Výše uvedený kód ilustruje, jak vypadá HTML soubor pro komponentu *inbox*, která obsahuje další komponenty. Mapování zajišťuje vzájemnou komunikaci mezi šablonou a komponentou. Property binding slouží k předání hodnoty do proměnné vnořené komponenty nebo direktivy a také k přiřazení hodnoty do vlastnosti elementu v DOM stromu, což umožňuje změnit chování oproti výchozí hodnotě atributu HTML elementu. V tomto případě například `[accounts] = "accounts"`. Kromě toho je zde také zobrazeno mapování metody na událost, tzv. event binding. To umožňuje naslouchat určité události a reagovat na ni pomocí event handleru, například `(onClose)="handleMailClose()"`.

V těchto komponentách se k asynchronnímu zpracování dat používá tzv. *Observable*, který je založen na návrhovém vzoru *Observer* a umožňuje streamování dat v čase, kdy se data objevují nebo se mění. Když se v aplikaci nějaká data změní, *Observable* zaznamená tyto změny a oznámí je všem odběratelům (subscriberům), kteří se na tato data zaregistrovali.

Dále budu popisovat jak funguje komunikace mezi *Inbox* komponentou a *Inbox Service*. Třída *Inbox* pomocí DI injektuje naši *Inbox Service*. Dále bude už jen na handlovat podle typu události volání metod z *Inbox Service*.

Následující popis se bude věnovat fungování komunikace mezi *Inbox* komponentou a *Inbox Service*. Třída *Inbox* využívá DI k injektování našeho *Inbox Service*. Poté bude již jen na této třídě, jak bude reagovat na dané typy událostí a jaké metody z *Inbox Service* bude volat.

```
export class InboxComponent implements OnDestroy, OnInit {
  private mail: IMailDetail;
  constructor(private ims :InboxService)

  handleTaskClick(task: IMailTask) : void {
    task.done = !task.done;
    this.ims.changeTaskState(task, this.mail);
  }

  handleTaskOpen(task: IMailTask) : void {
    this.ims.openTaskModal(task.idTask, this.mail);
  }
}
```

Výpis 4.7: Ukázka implementace TypeScript souboru *Inbox* komponenty

```

export class InboxService implements OnDestroy {
  public changeTaskState(task: IMailTask, mail: IMailDetail) {
    if (!task.idTask) {
      this.emap.patch('udalosti', { idUdalost: task.idEvent, vyrizeno: !task.done
        ? null : moment().format() }).subscribe(
        (res) => {
          this.updateDetailEventsAndTasks(mail, true);
        },
        (err) => {
          task.done = !task.done;
        }
      );
    } else {
      if (task.done) {
        this.emap.patch('ukoly', { idUkol: task.idTask, idUkolStav: 6 }).subscribe(
          (
            (res) => {
              this.updateDetailEventsAndTasks(mail);
            },
            (err) => {
              task.done = !task.done;
            }
          );
      } else {
        this.emap.patch('ukoly', { idUkol: task.idTask, idUkolStav: 1,
          idZamestnanecSplnil: null, datumSplneno: null }).subscribe(
          (res) => {
            this.updateDetailEventsAndTasks(mail);
          },
          (err) => { task.done = !task.done; }
        );
      }
    }
  }
}

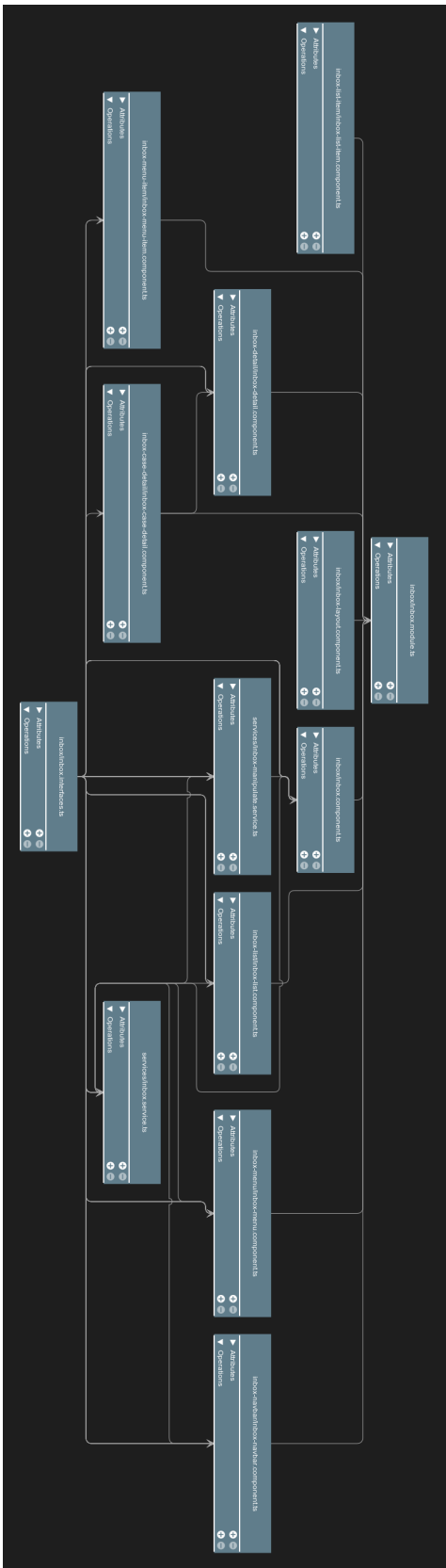
```

Výpis 4.8: Ukázka implementace InboxService komponenty

Výše jsem uvedl několik příkladů kódů, které ilustrují použití různých konceptů jako jsou property binding, event binding, DI injekce a další. Je třeba poznamenat, že další implementační kódy, které byly použity v projektu, jsou příliš rozsáhlé a detailní pro uvedení v této práci. Nicméně, výše uvedené ukázky by měly poskytnout základní přehled o tom, jak byl modul vyvíjen.

Během vývoje jsme provedli několik změn v designu od původního wireframu, aby byl modul co nejefektivnější a co nejvíce vyhovoval zákazníkům. Výsledný design modulu můžete vidět na níže uvedeném obrázku. Díky těmto změnám byl modul ještě uživatelsky přívětivější a efektivnější pro řešení požadovaných úkolů.

V následujícím obrázku je znázorněna struktura modulu, včetně všech jednotlivých komponent, které jsou zahrnuty v jeho designu.



Obrázek 4.7: Diagram komponent zachycující strukturu modulu

[+ NOVÁ ZPRAVA](#)
[PŘIJMOUT](#)
[Přetáhněte / Nepřetáhněte](#)
[Odstranit](#)
[Odpovědět](#)
[Přeposlat](#)

Hledat ve zprávách a dokumentech...

Emaily Filter

- Evolio - Obchod so, 11. 03.
- Výkaz práce - Šustr Stanislav**
1/2022 - Ukázkový advokátní spis (vyřizuje Daniel Slazák)
- Evolio - Obchod so, 04. 03.
- Rozhodnutí Výkaz práce - Šustr Stanislav**
1/2022 - Kopie rodinného domu - Nosičnice (vyřizuje Dan...
1/2023 - Spor o společenosti s Erollem (vyřizuje Daniel Sl...

Vyhledat v účtech...

- Pošta k vyřízení 3
- Datové zprávy 1
- Emaily** 2
- Listinná pošta
- Všechny datové zprávy 3
- Všechny emaily
- Všechna listinná pošta 1
- Výchozí účet vzor (email) 1
- SSustr (email) 2
- SSustr (email)
- Jan Kovář (email)
- Vojtěch Barbušák (019197m) 3
- SSustr (0646c7f) 3
- Přidat nebo upravit účet...

TEST2

Výkaz práce - Šustr Stanislav

obohod@evolio.email
Komu: stanislav.sustr@avesoft.cz
Doručeno: 11. 3. 2023 6:01

1/2022 - Ukázkový advokátní spis ...

Vyřídil poštu pro spisa 1/2022 Daniel Slazak

Přidat úkol Poznámka

Vykazování dle progra...xlsx (6 KB)

Výkaz práce - Šustr S...msg (32 KB)

Stáhnout všechny přílohy

AKTIVITY K POŠTĚ

17. 4. 2023 16:41 Prirazeno k vyřízení ke spisu uživatelem Daniel Slazak

Obrázek 4.8: Uživatelské rozhraní nové podatelny

4.2.3 Testování

V průběhu vývoje byly testovací aktivity prováděny testery, aby byla zajištěna nejvyšší kvalita produktu. Tyto testy byly prováděny průběžně během celého vývojového cyklu a byly důkladně dokumentovány. Díky tomu jsme byli schopni identifikovat a odstranit jakékoliv chyby a nedostatky v produktu v raných fázích vývoje, což nám umožnilo dodat výsledný produkt s minimálním počtem problémů. Tím jsme zajistili spokojenost zákazníků s kvalitou produktu a minimalizovali potřebu opravovat chyby po jeho vydání.

Kapitola 5

Závěr

5.1 Uplatněné vědomosti a znalosti

Během mé praxe jsem měl příležitost využít mnoho důležitých dovedností a znalostí, které jsem získal během svého bakalářského studia. Konkrétně jsem mohl uplatnit své znalosti z předmětů C# I a II. V oblasti vývoje internetových aplikací jsem využil znalosti z předmětu Vývoj aplikací v JavaScriptu a Vývoj internetových aplikací.

5.2 Scházející vědomosti a znalosti

Také jsem si uvědomil, že mi schází určité znalosti a dovednosti, které jsou důležité pro moderní vývoj softwaru. Konkrétně mi chyběly znalosti ohledně cloudových služeb, jako například Microsoft Azure nebo s verzovacím systémem Git. I když jsem měl předchozí zkušenosti s jazykem JavaScript, když jsem se potkal s frameworkem Angular a zejména s knihovnou RxJS, uvědomil jsem si, že v mých znalostech jsou mezery.

5.3 Celkové zhodnocení praxe

Celkově bych zhodnotil mou praxi jako velmi pozitivní a cennou zkušenost. Měl jsem příležitost uplatnit své znalosti a dovednosti získané během bakalářského studia a rozšířit je o nové oblasti, které mi dosud nebyly známé. Zároveň jsem si uvědomil, jak důležité je neustále se učit nové věci a zdokonalovat své znalosti a dovednosti, aby bylo možné uspět v oblasti IT.

Během praxe jsem měl příležitost pracovat v týmu a získat reálnou představu o komunikaci a práci v týmu, což bylo pro náplň mé práce nezbytné. Uvědomil jsem si také, že je důležité nejen umět pracovat s technologiemi, ale také se umět dobře komunikovat a spolupracovat s ostatními lidmi v týmu.

Velmi si vážím této příležitosti, která mi umožnila rozšířit své vědomosti a dovednosti a získat cennou zkušenost v profesionálním pracovním prostředí.

Literatura

1. *Angular* [online]. [cit. 2023-04-15]. Dostupné z: <https://angular.io>.
2. *TypeScript* [online]. [cit. 2023-04-15]. Dostupné z: <https://www.typescriptlang.org>.
3. *.NET* [online]. [cit. 2023-04-15]. Dostupné z: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>.
4. *Microsoft C#* [online]. [cit. 2023-04-15]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>.
5. *SQL* [online]. [cit. 2023-04-15]. Dostupné z: <https://aws.amazon.com/what-is/sql/>.
6. *Microsoft Azure* [online]. [cit. 2023-04-15]. Dostupné z: <https://azure.microsoft.com>.
7. *Azure Service Fabric* [online]. [cit. 2023-04-15]. Dostupné z: <https://azure.microsoft.com/cs-cz/products/service-fabric>.
8. *Postman* [online]. [cit. 2023-04-15]. Dostupné z: <https://www.postman.com/>.
9. *Git* [online]. [cit. 2023-04-15]. Dostupné z: <https://docs.gitlab.com/>.
10. *Milestones* [online]. [cit. 2023-04-15]. Dostupné z: <https://docs.github.com/en/issues/using-labels-and-milestones-to-track-work/about-milestones>.

Příloha A

Ukázkový výpis z katastru nemovitostí

| VÝPIS Z KATASTRU NEMOVITOSTÍ | | | | | |
|---|-----------------------|-------------------|-----------------------------------|-----------------------|------------------------------|
| prokazující stav evidovaný k datu 03.11.2021 22:09:57 | | | | | |
| Okres: CZ0315 Prachatice | Obec: 550621 Vacov | | | | |
| Kat.území: 602515 Benešova Hora | List vlastnictví: 208 | | | | |
| V kat. území jsou pozemky vedeny ve dvou číselných řadách (St. = stavební parcela) | | | | | |
| <i>A Vlastník, jiný oprávněný</i> | <i>Identifikátor</i> | | <i>Podíl</i> | | |
| <i>Vlastnické právo</i> | | | | | |
| Kühr Alois, Třešněná 8107, 11000 Sosnová | 140318/3903 | | | | |
| B Nemovitosti | | | | | |
| <i>Pozemky</i> | <i>Parcela</i> | <i>Výměra[m2]</i> | <i>Druh pozemku</i> | <i>Způsob využití</i> | <i>Způsob ochrany</i> |
| <i>St.</i> | <i>11/3</i> | <i>297</i> | <i>zastavěná plocha a nádvoří</i> | | |
| <i>Součástí je stavba: Benešova Hora, č.p. 21, bydlení</i> | | | | | |
| <i>Stavba stojí na pozemku p.č.: St. 11/3</i> | | | | | |
| | <i>1</i> | <i>362</i> | <i>zahrada</i> | | <i>zemědělský půdní fond</i> |
| | <i>675</i> | <i>184</i> | <i>zahrada</i> | | <i>zemědělský půdní fond</i> |
| B1 Věcná práva sloužící ve prospěch nemovitosti v části B | | | | | |
| Typ vztahu | | | | | |
| o Věcné břemeno chůze | | | | | |
| <i>Oprávnění pro</i> | | | | | |
| <i>Parcela: St. 11/3</i> | | | | | |
| <i>Povinnost k</i> | | | | | |
| <i>Parcela: St. 11/1</i> | | | | | |
| <i>Listina Usnesení soudu číslo deníku 476/1924 AAAAA:11111 AAAAA:111111 AAA49569532.</i> | | | | | |
| <i>AAAAA:1/1111 AAAAA:1/1111 Z-700004/1976-306</i> | | | | | |
| <i>Pořadí k AAAA AAAAA AAAAA AAAAA AAAAA A AAAAA A AAAAA AAAAA39566267</i> | | | | | |
| o Věcné břemeno bráni vody | | | | | |
| <i>Oprávnění pro</i> | | | | | |
| <i>Parcela: St. 11/3</i> | | | | | |
| <i>Povinnost k</i> | | | | | |
| <i>Parcela: St. 11/1</i> | | | | | |
| <i>Listina Usnesení soudu číslo deníku 476/1924 AAAAA:11111 AAAAA:111111 AAA49569532.</i> | | | | | |
| <i>AAAAA:1/1111 AAAAA:1/1111 Z-700004/1976-306</i> | | | | | |
| <i>Pořadí k AAAA AAAAA AAAAA AAAAA AAAAA A AAAAA A AAAAA AAAAA39566267</i> | | | | | |
| C Věcná práva zatěžující nemovitosti v části B včetně souvisejících údajů | | | | | |
| Typ vztahu | | | | | |
| o Věcné břemeno vedení | | | | | |
| <i>A AAAAA AAAAAAAAAA</i> | | | | | |
| <i>AAAAAAAA A AAAAAA AAAAAA AAAAAA AA AAA 11.1.1111.</i> | | | | | |
| <i>AAAAAAAA AAAAA AAAAA AA AAA 1142990522</i> | | | | | |
| <i>Oprávnění pro</i> | | | | | |
| <i>Parcela: 662/12</i> | | | | | |
| <i>Povinnost k</i> | | | | | |
| <i>Parcela: 1</i> | | | | | |
| <i>Listina Smlouva o věcném břemenu V3 769/1997 AAAAA:11111 AAAAA:111111 AAA86439707.</i> | | | | | |
| <i>AAAAA:1/1111 AAAAA:1/1111 Z-700017/1997-306</i> | | | | | |
| <i>Pořadí k AAAA AAAAA AAAAA AAAAA AAAAA A AAAAA AAAAA AAAAA39566267</i> | | | | | |
| <small>Nemovitosti jsou v územním obvodu, ve kterém vykonává státní správu katastru nemovitostí ČR Katastrální úřad pro Jihočeský kraj, Katastrální pracoviště Prachatice, kód: 306. strana 1</small> | | | | | |

Obrázek A.1: Ukázka listu vlastnictví