MASARYKOVA
UNIVERZITA

FAKULTA INFORMATIKY

Návrh a implementácia systému pre digitálnu správu majetku

Diplomová práca

DOMINIK BORBULIAK

Brno, jar 2025

MASARYKOVA
UNIVERZITA

FAKULTA INFORMATIKY

Návrh a implementácia systému pre digitálnu správu majetku

Diplomová práca

DOMINIK BORBULIAK

Vedúci práce: RNDr. Jaroslav Pelikán, Ph.D.

Katedra počítačových systémov a komunikácií

Brno, jar 2025

FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

Vyhlásenie

Vyhlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Dominik Borbuliak

Vedúci práce: RNDr. Jaroslav Pelikán, Ph.D.

Poďakovanie

Ďakujem vedúcemu práce RNDr. Jaroslavovi Pelikánovi, Ph.D. za odborné vedenie práce, vecné rady a pripomienky. Taktiež ďakujem rodičom a priateľke za podporu počas štúdia. Na záver chcem poďakovať kolegom z firmy intecs data, s.r.o. za diskusie počas návrhu systému.

Zhrnutie

Cieľom tejto diplomovej práce je implementovať systém pre správu majetku pre firmu intecs data, s.r.o., ktorý umožní minimalizovať nedostatky existujúcej správy majetku, ktorá prebieha pomocou súboru v programe MS Excel. Samotnej implementácii predchádza návrh a spracovanie požiadaviek na tento systém. Výsledný systém je vo firme pravidelne používaný a významne prispel k efektivite správy firemného majetku.

Kľúčové slová

C#, information systems, .NET, React, REST API, SQL, Web Application

Obsah

Úvod 1

1 Úvod do problematiky 2
- 1.1 O firme intecs data s.r.o. . . . 2
- 1.2 Existujúca správa majetku . . . 2
- 1.3 Nedostatky existujúcej správy majetku . . . 2
  - 1.3.1 Nekonzistencia dát . . . 3
  - 1.3.2 Časovo náročná inventúra . . . 3
  - 1.3.3 Limitované možnosti automatizácie . . . 4
- 1.4 Zhodnotenie problematiky . . . 4

2 Návrh 5
- 2.1 Požiadavky na systém . . . 5
  - 2.1.1 Funkčné požiadavky . . . 5
  - 2.1.2 Nefunkčné požiadavky . . . 11
- 2.2 Návrh obrazoviek . . . 12
- 2.3 Dátový model . . . 16
  - 2.3.1 Dodaný návrh dátového modelu . . . 17
  - 2.3.2 Úprava dátového modelu . . . 17
- 2.4 Atchitektúra . . . 18
  - 2.4.1 Klienstka časť systému . . . 19
  - 2.4.2 Serverová časť systému . . . 19

3 Implementácia 20
- 3.1 Azure Dev Ops . . . 21
  - 3.1.1 Nastavenie repozitárov . . . 21
  - 3.1.2 Kontinuálna integrácia a nasadenie . . . 22
- 3.2 Cloudové služby Azure . . . 22
- 3.3 Klientska časť systému . . . 23
  - 3.3.1 Štruktúra projektu . . . 24
  - 3.3.2 Autentifikácia a autorizácia užívateľov . . . 26
  - 3.3.3 Komunikácia so serverovou časťou systému . . . 26
- 3.4 Serverová časť systému . . . 27
  - 3.4.1 Štruktúra serverovej časti . . . 27
  - 3.4.2 Autentifikácia a autorizácia užívateľov . . . 28

3.4.3 Vrstva infraštruktúry 29
3.4.4 Doménová vrstva 30
3.4.5 Aplikačná vrstva 31

4 Testovanie a spätná väzba **32**
4.1 Metóda testovania 32
4.2 Spätná väzba a jej zapracovanie 32

Záver **34**

Bibliografia **36**

A Zoznam príloh **38**

Zoznam obrázkov

1.1 Nekonzistencia dát v pôvodnej správe majetku 3
2.1 Diagram pripadov použitia 6
2.2 Vygenerovaný QR kód majetku 8
2.3 Návrh úvodnej obrazovky na výber modulu 12
2.4 Návrh obrazoviek s tabuľkou 13
2.5 Návrh obrazovky s formulárom pre úpravu detailu majetku 14
2.6 Návrh obrazoviek so skenerom 15
2.7 Návrh mobilenej obrazovky 16
2.8 Návrh architektúry systému 18
3.1 Úlohy v nástroji Targetprocess 20
3.2 Vetvy vzdialených repozitárov 21
3.3 Štruktúra serverovej časti 28

Úvod

Rast firiem so sebou typicky prináša aj navyšujúce sa množstvo majetku vlastneného firmou. Firmy v rôznych sektoroch, potrebujú na vykonávanie svojej činnosti rôzny majetok. V sektore informačných technológii sa primárne jedná o monitory, počítače, stoly a mnoho ďalšieho majetku, ktorý zamestnancom umožňuje vykonávať ich prácu. Každý zamestnanec zvyčajne potrebuje svoje vlastné zariadenie na prácu, takže množstvo majetku rastie s počtom zamestnancov.

Firmám so sídlom v Českej republike vyplýva zo zákona povinnosť inventarizovať takýto majetok [1]. Udržať si prehľad o menšom počte majetku je možné pomocou obyčajného papieru prípadne tabuliek v programe MS Excel. Rýchlo rastúce firmy v sektore informačných technológii, môžu veľmi rýchlo naraziť na limity takejto inventarizácie. Medzi časté problémy patria napríklad preklepy, vymazané riadky, nekonzistencia dát a iné. Tieto problémy, je takmer vždy možné odstrániť, no časom to pre firmu prestane byť ekonomicky výhodne.

Tomuto problému čelí aj firma intecs data s.r.o.[1], ktorá inventarizáciu rieši primárne pomocou programu MS Excel. Pribúdajúce množstvo zamestnancov a majetku vedie k nekonzistencii a chýbajúcim dátam, ktorých oprava zaberá veľké množstvo času. Tieto problémy, taktiež komplikujú pravidelnú inventúru, ktorá prebieha raz ročne. Pri tejto inventúre je potrebné overiť či firma vlastní majetok, ktorý je zaznamenaný vo firemnom účtovníctve a v prípade chýb je potrebné toto účtovníctvo vysporiadať.

Cieľom tejto diplomovej práce, je navrhnúť a implementovať digitálny systém pre správu majetku, ktorý nahradí súčasnú správu majetku vo firme intecs data s.r.o. Medzi kľúčové vlastnosti systému patrí minimalizácia počtu nekonzistencií a chýb v dátach, a taktiež aj funkcionalita, ktorá zrýchli pravidelnú ročnú inventúru.

1. `https://www.intecs.cz/`

1 Úvod do problematiky

1.1 O firme intecs data s.r.o.

Primárne zameranie firmy intecs je analýza firemných dát iných firiem a tvorba reportov, ktoré im pomáhajú robiť obchodné rozhodnutia. Reporty sú tvorené pomocou nástroja Power BI[1], ale v prípadoch, kedy je funkcionalita príliš komplikovaná ich nahradzujú webové aplikácie, čo patrí do vedľajšieho zamerania firmy.

1.2 Existujúca správa majetku

Správa majetku prebieha vo firme pomocou súboru v programe MS Excel, ku ktorému má prístup len obmedzený počet zamestnancov. Medzi zamestnancov s prístupom k tomuto súboru patrí oddelenie zamerané na ľudské zdroje, správu kancelárii, infraštruktúru firmy a účtovníčka. Súbor si zdieľajú cez internet aby nedošlo k vzájomnému prepisovaniu dát v prípade, že ho upravujú dvaja zamestnanci súčasne. Hlavným obsahom súboru, je tabuľka s majetkom, ktorý vlastní firma. Každý záznam v tejto tabuľke reprezentuje jeden majetok a informácie o ňom. Medzi tieto informácie patrí napríklad názov, kancelária, v ktorej sa majetok nachádza, osoba v užívaní, inventárne číslo a mnoho ďalšieho. Majetok, ktorý zo zákona musí byť súčasťou firemného účtovníctva je pomocou inventárneho čísla spojený so záznamom v aplikácii, ktorú využíva účtovníčka na správu účtovníctva.

1.3 Nedostatky existujúcej správy majetku

Proces správy majetku popísaný v sekcii 1.2 má veľa nedostatkov. Nekonzistencia dát, limitované možnosti automatizácie a časovo náročná inventúra, ktorá musí prebiehať raz ročne patria k najviac limitujúcim nedostatkom.

---

1. `https://www.microsoft.com/cs-cz/power-platform/products/power-bi`

1.3.1 Nekonzistencia dát

Najčastejším problémom sú chýbajúce údaje, prípadne stĺpce obsahujúce údaje, ktoré patria do iného stĺpca. Program MS Excel poskytuje len obmedzené možnosti na zabránenie takýmto problémom, takže nekonzistencia dát sa zvyšuje s pribúdajúcim počtom majetku. Pomocou pravidelnej kontroly a úpravy dát je možné tento problém dočasne odstrániť, ale samotná kontrola je časovo náročná.

J	K	L
Interní číslo faktury	osoba v užívání	Poznámka
	vyřazeno	
	vyřazeno	
	při inventuře -máme v kanceláři?	
	při inventuře -máme v kanceláři?	

Obr. 1.1: Nekonzistencia dát v pôvodnej správe majetku

Obrázok 1.1 znázorňuje nekonzistenciu dát, kde v niektorých prípadoch je osoba v užívaní neznáma, prípadne obsahuje údaje, ktoré by sa mali nachádzať v stĺpci poznámka.

1.3.2 Časovo náročná inventúra

Pre majetok, ktorý je evidovaný v účtovníctve, je potrebné raz ročne spraviť inventúru a overiť, že sa daný majetok nestratil a je stále vo vlastníctve firmy. Inventúru zvyčajne vykonáva pár zamestnancov, pričom postupne prechádzajú jednotlivé kancelárie a snažia sa v súbore na správu majetku dohľadať takýto majetok a následne k nemu poznačia, že bol nájdený. Samotné dohľadanie majetku v súbore je zdĺhavý proces, pretože dáta nemusia byť konzistentné, prípadne môžu kompletne chýbať. Po prejdení všetkých kancelárii si vyfiltrujú majetok, ktorý nebol nájdený, ešte raz skontrolujú či sa nejedná o chybu v dátach a po tejto kontrole predajú súbor účtovníčke. Účtovníčka následne tieto zmeny reflektuje do svojho programu a vysporiada firemné účtovníctvo.

1.3.3 Limitované možnosti automatizácie

Rastúci počet zamestnancov vo firme vedieť k nákupu nových zariadení, ktoré potrebujú na prácu a zvyčajne si tieto zariadenia berú domov. Typicky sa jedná o notebooky, pričom pri prevzatí takéhoto zariadenia musí každý zamestnanec podpísať predávaci protokol. Podobne aj pri likvidácii majetku, je potrebné podpísať protokol o likvidácii majetku, ktorý následne využije účtovníčka na korektné vysporiadanie účtovníctva. Tieto dokumenty sa vytvárajú ručne, čo pri väčšom množstve zariadení a zamestnancov zaberá veľa času. Automatizácia procesu generovania týchto protokolov na základe dát v súbore na správu majetku, by celý proces výrazne zrýchlila, no program MS Excel takúto možnosť neumožňuje.

1.4 Zhodnotenie problematiky

Nedostatky popísané v sekcii 1.3 je možné vyriešiť prechodom na iný program alebo webovú aplikáciu pre správu majetku. Na trhu existujú rôzne riešenia, ktoré by umožnili eliminovať väčšinu nedostatkov, no nákupom takéhoto riešenia bude firma limitovaná na funkcionalitu poskytovanú daným riešením. Požiadavka na inú funkcionalitu by mala za následok prechod na iné riešenie, čo by v prípade nekompatibility dát medzi týmito riešeniami mohlo byť komplikované. Minimalizovať pravdepodobnosť prechodu na iné riešenie, by bolo možne na základe rozsiahlej analýzy a výberu správneho riešenia. Takáto analýza spolu s cenou samotného riešenia by firmu stála približne rovnako ako implementácia vlastného riešenia. Implementácia vlastného riešenia má veľkú výhodu v tom, že je možné do aplikácie pridať takmer akúkoľvek funkcionalitu a taktiež ho môže implementovať vývojársky tím, ktorý je súčasťou firmy. Na základe tohto prišla požiadavka od vedenia firmy na implementáciu vlastného systému pre správu majetku.

Cieľom tejto diplomovej práce je navrhnúť a implementovať systém pre správu majetku, tak aby sa pomocou tohto systému podarilo eliminovať nedostatky existujúcej správy majetku.

2 Návrh

Systém je potrebné navrhnúť tak, aby umožňoval všetko čo umožňuje existujúca správa majetku, ale zároveň obsahoval doplňujúcu funkcionalitu, ktorá užívateľom spríjemní používanie, prípadne pridá možnosti, ktoré nie sú možné v existujúcej správe majetku.

2.1 Požiadavky na systém

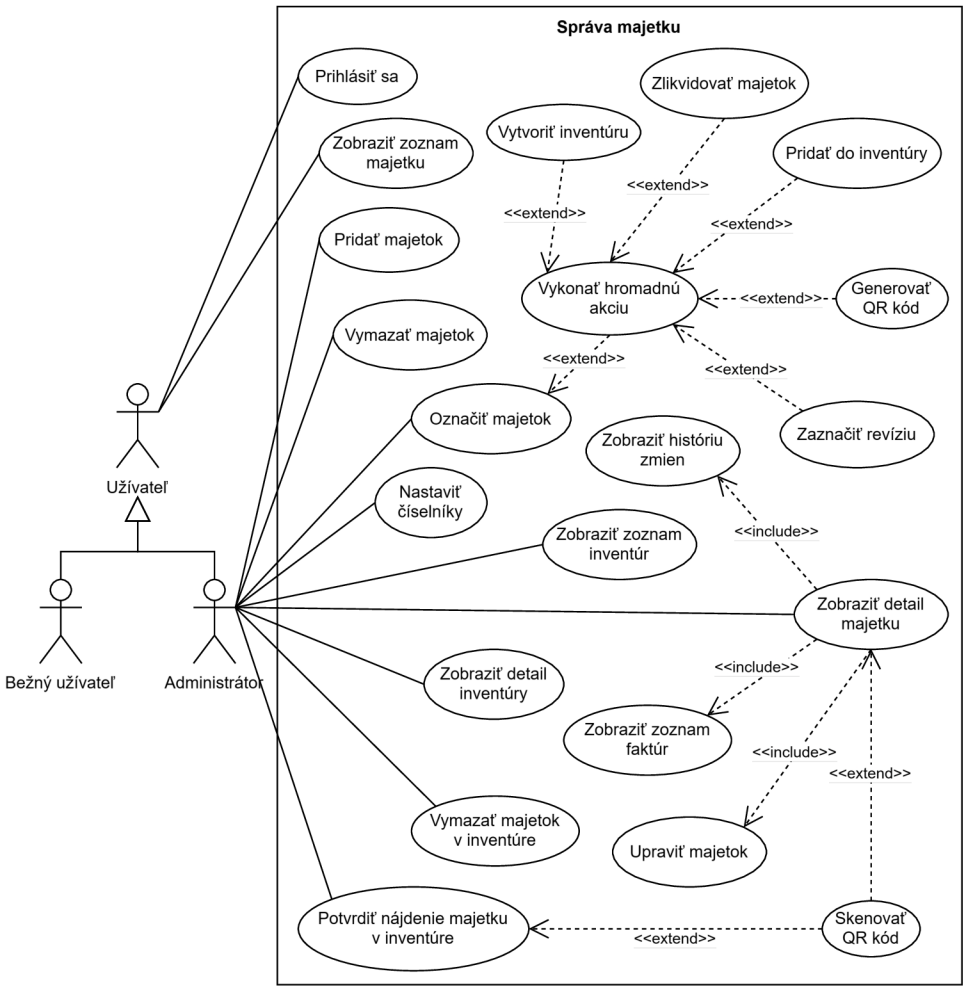
Požiadavky na systém je možné rozdeliť na funkčné a nefunkčné. Funkčné požiadavky popisujú akú funkcionalitu má systém poskytovať a nefunkčné požiadavky popisujú rôzne obmedzenia na implementáciu funkčných požiadaviek [2].

2.1.1 Funkčné požiadavky

Jednotlivé funkčné požiadavky na systém sú reprezentované pomocou diagramu prípadov použitia[1], kde sú zobrazené jednotlivé typy užívateľov a konkrétna funkcionalita, ku ktorej majú prístup.

Užívateľov systému je možné rozdeliť na dve skupiny. Prvá skupina užívateľov reprezentuje radových zamestnancov, ktorý majú prístup len k veľmi obmedzenej funkcionalite. Táto skupina užívateľov je ďalej označovaná ako *bežní užívatelia*. Skupina užívateľov s plným prístupom je tvorená z oddelenia zameraného na ľudské zdroje, správu kancelárii, infraštruktúru firmy a účtovníčky. Táto skupina užívateľov je ďalej označovaná ako *administrátori*. Obe tieto skupiny užívateľov sú reprezentované jednou entitou, ale máju odlišnú rolu na základe, ktorej sa určujú ich oprávnenia.

1. Z anglického use case



Obr. 2.1: Diagram pripadov použitia

Autentifikácia a autorizácia

Každý zamestnanec firmy ma firemný e-mail, ktorý používa na prihlásenie do väčšiny aplikácii. Tieto účty sú vytvorené pomocou služby Microsoft Entra Id, ktorá umožňuje jednotné prihlásenie[2]. Jednotné prihlásenie je spôsob akým sa užívateľ vie prihlásiť do viacerých nezávislých systémov pomocou tých istých prihlasovacích údajov. Pri

2. Z anglického single sign on

použití jednotného prihlásenia sa taktiež minimalizuje počet potrebných prihlásení užívateľa.[3]

Firma využíva veľké množstvo aplikácii od firmy Microsoft, do ktorých je možné obmedzovať prístup na základe skupín, ktoré sú priradené jednotlivým užívateľom. Pri nástupe nového zamestnanca je vytvorený firemný účet a zároveň pridelené skupiny, ktoré mu určujú oprávnenia do takmer všetkých aplikácii, ktoré bude používať. Výnimkou by nemal byť ani systém na správu majetku, takže systém musí poskytovať autentifikáciu na základe firemného účtu a autorizáciu na základe skupín, priradených k tomuto účtu.

Zoznam majetku

Najpodstatnejšou funkcionalitou systému je prehľad majetku, ktorý užívateľom poskytuje zoznam majetku formou tabuľky, v ktorej môžu filtrovať a zoraďovať jednotlivé záznamy. Ďalej majú užívatelia možnosť skrývať jednotlivé stĺpce tabuľky a možnosť exportovať vyfiltrované záznamy. Táto funkcionalita imituje existujúcu správu majetku, takže jej používanie je intuitívne.

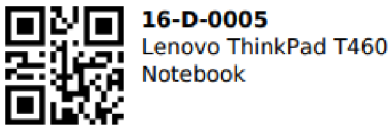
Prístup k tejto funkcionalite majú bežní užívatelia aj administrátori. Bežný užívateľ v tomto zozname vidí len majetok, ktorý je priradený k jeho osobe a jedná sa o jedinú funkcionalitu, ktorú má prístupnú. Naopak administrátori, tu vidia prehľad majetku firmy bez ohľadu na držiteľa.

Zoznam majetku je pred filtrovaný tak aby zobrazoval len aktívny majetok, ale pomocou filtrov umožňuje užívateľom zobraziť aj vyradený majetok. Toto vylepšenie užívateľom spríjemňuje používanie systému pretože typicky, ich zaujíma len aktívny majetok a na vyradený majetok sa pozerajú len občasne.

Hromadné akcie nad zoznamom majetku

Administrátori majú možnosť označiť jednotlivé záznamy a následne pre označené záznamy vykonať akciu. Medzi tieto akcie patrí generovanie QR kódov, zaznačenie revízie, likvidácia majetku, vytvorenie inventúry a pridanie do existujúcej inventúry. Každá akcia otvorí modálne okno s formulárom pre vyplnenie potrebných informácii, pričom formulár poskytuje validáciu údajov.

QR kódy vygenerované pre majetok obsahujú zakódovaný identifikátor majetku. Existujú rôzne tlačiarne, ktoré sa špecializujú na tlačenie QR kódov, ale firma vlastní iba obyčajnú atramentovú tlačiareň, ktorá zvládne tlačiť na rôzne veľkosti a druhy papiera. Tlačenie QR kódov prebieha za pomoci existujúcej tlačiarne a použitia lepiaceho papiera, ktorý obsahuje sedemnásť riadkov so štyrmi nálepkami na každom riadku. QR kódy sa generujú aj v menšom počte ako je počet nálepiek. Pomocou formulára na generovanie QR kódov, je možné zadať počet využitých nálepiek, čo zabezpečí, že sa lepiaci papier môže použiť viac krát. Okrem QR kódu obsahuje nálepka aj evidenčné číslo, kategóriu a názov majetku, ktoré sa nachádzajú vedľa QR kódu. V prípade potreby QR kódu na malý nábytok je možné tieto informácie odstrihnúť a nalepiť len malý QR kód. Nálepky sa nachádzajú aj na notebookoch, ktoré zamestnanci často vkladajú a vyberajú z tašky, takže materiál lepiaceho papiera je matný, čo zabezpečuje dlhšiu životnosť samotnej nálepky.



Obr. 2.2: Vygenerovaný QR kód majetku

Pridanie nového majetku

Pri nákupe nového majetku vo firme, je potrebné takýto majetok pridať do správy majetku. Pridanie majetku umožňuje formulár, do ktorého užívateľ zapíše všetky potrebné údaje o majetku. Medzi tieto údaje patrí napríklad meno, sériové číslo, cena na faktúre a mnoho ďalšieho. Formulár poskytuje validáciu údajov, aby nebolo možné vložiť chybné vstupy. Po vložení korektných údajov a potvrdení sa majetok pridá do správy majetku. Takémuto majetku sa automatický priradí typ a evidenčné číslo na základe ceny majetku.

Majetok od 5 000 do 80 000 českých korún sa nazýva drobný hmotný majetok a majetok nad 80 000 českých korún sa nazýva dlho-

dobý hmotný majetok. Oba tieto typy majetku majú priradenú skratku, ktorá sa následne používa v evidenčnom čísle.

Formát evidenčného čísla je *YY-T-XXXX*. *YY* reprezentuje posledné dvojčíslie roku, v ktorom bol majetok kúpený, *T* reprezentuje skratku typu majetku, ktorý sa priradil automaticky na základe jeho ceny a *XXXX* reprezentuje poradie majetku pre daný rok a typ.

Mazanie majetku

Po pridaní majetku do správy majetku si ho účtovníčka zaeviduje do programu na účtovníctvo. V tomto programe je majetok evidovaný podľa faktúr a každej priradí interné číslo faktúry. Toto číslo následne účtovníčka prenesie do systému na správu majetku.

Ak sa vymaže majetok v systéme na správu majetku, ale zabudne vymazať v programe, ktorý používa účtovníčka, tak vznikne nekonzistencia, ktorá spôsobí problémy pri inventúre. Z tohoto dôvodu je možné mazať len majetok, ktorý nemá priradené interné číslo faktúry.

Funkcionalita mazania majetku sa bude typicky používať len vo veľmi výnimočných prípadoch a to hlavne ak užívateľ pri pridávaní majetku spraví veľa chýb a vymazanie majetku bude efektívnejšie ako jeho úprava.

Prehľad majetku a úprava majetku

Funkcionalita popísaná v sekcii 2.1.1 zobrazuje užívateľom len základné informácie. Užívatelia majú prístup k detailnejším informáciám o danom majetku formou pre kliku z vyššie spomenutého zoznamu.

Úprava majetku prebieha pomocou pred vyplneného formulára, ktorý je možne využiť na zobrazovanie detailnejších informácii o majetku jednoduchým pridaním needitovateľných vstupov. Cenu majetku je možné upraviť len pridaním, upravením alebo zmazaním faktúry, pričom táto zmena neovplyvňuje už priradený typ majetku, pretože by bolo potrebné meniť aj evidenčné číslo. V prehľade majetku je dostupný zoznám faktúr a možnosť ich pridania, úpravy a mazania pričom nie je možné zmazať poslednú faktúru.

Každá úprava majetku je historizovaná, pričom prehľad majetku obsahuje tabuľku s prehľadom jednotlivých zmien na majetku a každý záznam obsahuje údaj o tom kto a kedy zmenu uskutočnil.

Skenovanie QR kódu

Systém poskytuje užívateľovi skener QR kódov, ktorý užívateľa automaticky presmeruje na prehľad majetku. Užívateľ má možnosť zvoliť kameru, ktorou sa QR kódy budú skenovať a táto voľba je uložená v úložisku prehliadača aby užívateľ nemusel túto voľbu opakovať pri novom otvorení skeneru.

Zoznam a prehľad inventúr

Systém zobrazuje zoznam inventúr, ktorý obsahuje základné informácie o inventúrach ako typ, dátum inventúry a tlačidlo na pre klik do prehľadu konkrétnej inventúry. Užívateľov typicky zaujíma len najnovšia inventúra, takže zoznam je zoradený podľa dátumu.

Prehľad inventúry obsahuje informácie o majetku, ktorý podlieha danej inventúre pomocou podobnej tabuľky ako v sekcii 2.1.1. Jednotlivé záznamy v tejto tabuľke obsahujú naviac informáciu o tom či bol majetok nájdený, metódu ktorou bol nájdený a možnosť vymazať daný majetok z inventúry.

Potvrdiť nájdenie majetku v inventúre je možné pomocou akcie nad označenými záznamami, ktorá otvorí modálne okno a umožní užívateľovi napísať komentár, ktorý sa následne pridá k označeným záznamom. Druhou možnosťou ako potvrdiť nájdenie majetku v inventúre je skenovanie QR kódu, pomocou skeneru, na ktorý sa užívateľ dostane pomocou tlačidla v prehľade inventúry. Funkcionalita skeneru je totožná ako v sekcii 2.1.1, s tým rozdielom, že naskenovanie užívateľa nepresmeruje na prehľad majetku, ale automaticky označí naskenovaný majetok v danej inventúre ako nájdený.

Nastavenia

Systém poskytuje užívateľovi možnosť spravovať číselníky potrebné pre jeho prácu. Medzi tieto číselníky patria lokácie, dôvody likvidácie, kategórie majetku, typy inventúr a typy majetku. Správa jednotlivých číselníkov prebieha pomocou tabuľky kde jednotlivé záznamy reprezentujú hodnoty číselníkov.

Pomocou formulára, ktorý sa zobrazí po kliknutí na tlačidlo je možné pridať novú hodnotu do číselníka. Tento formulár poskytuje validáciu vstupov, aby nebolo možné vložiť neplatné dáta. Väčšina

číselníkov obsahuje len názov hodnoty. Kategória majetku obsahuje okrem názvu aj nastavenie východzej hodnoty o tom, či daná kategória vyžaduje revíziu. Typ majetku obsahuje aj rozsah ceny majetku, ktorý do nej spadá a skratku daného typu, ktorá sa používa na generovanie evidenčného čísla.

Hodnoty v číselníkoch je možné vymazať pomocou tlačidla pri jednotlivých záznamoch, pričom samotné mazanie musí užívateľ potvrdiť pomocou modálneho okna. Mazanie pre hodnoty číselníkov, ktoré sú naviazané na majetok, je možné len v prípade, že nie sú použité v aktuálnom zázname majetku.

2.1.2 Nefunkčné požiadavky

Väčšinu jednoduchších interných aplikácii tvorí firma intecs pomocou Power App[3]. Tieto aplikácie zdieľajú podobný dizajn a sú dostupné na jednom mieste priamo z aplikácie MS Teams[4].

Správa majetku je druhý systém, ktorý je implementovaný ako webová aplikácia, pričom ten existujúci je implementovaný pomocou ASP.NET MVC. Model, pohľad a ovládače[5] patria medzi tri hlavné časti MVC architektúry[4]. Firma uvažuje o tom, že by tieto systémy a všetky ďalšie, spojila do jednej webovej aplikácie, ale zároveň sa chce vyhnúť architektúre MVC. Rozhodnutie o tom, či tieto aplikácie firma spojí závisí na počte ďalších systémov, ktoré bude potrebovať. Preto je správa majetku navrhnutá tak, aby do nej bolo v budúcnosti možné pridať ďalší systém bez komplikovaných úprav a taktiež má podobný dizajn ako už existujúci systém.

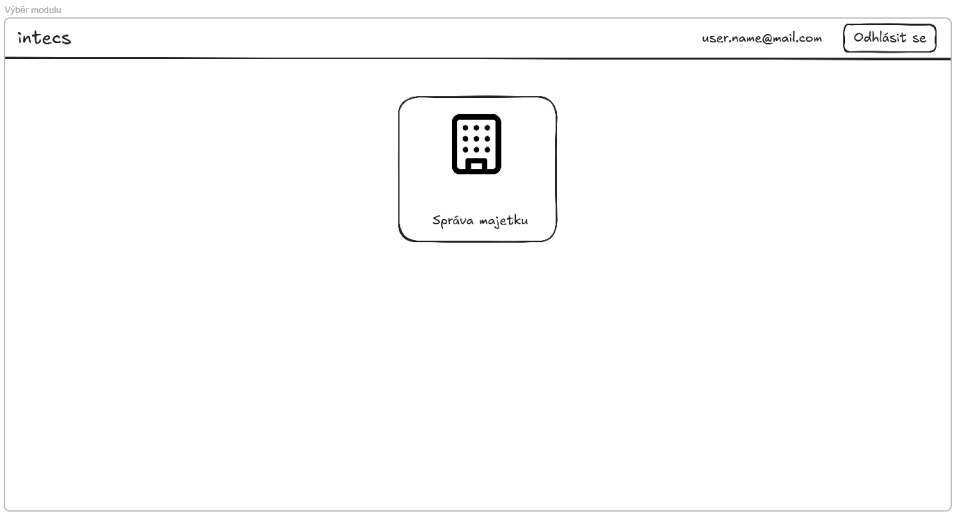
Systém poskytuje skenovanie QR kódov, ktoré sú primárne skenované pomocou mobilného zariadenia. Preto, je systém navrhnutý tak aby bol responzívny a užívateľom sa používal príjemne na mobilných zariadeniach.

---

3. `https://www.microsoft.com/en-us/power-platform/products/power-apps`
4. `https://www.microsoft.com/cs-cz/microsoft-teams/group-chat-software`
5. Z anglickeho model, view a controller

2.2 Návrh obrazoviek

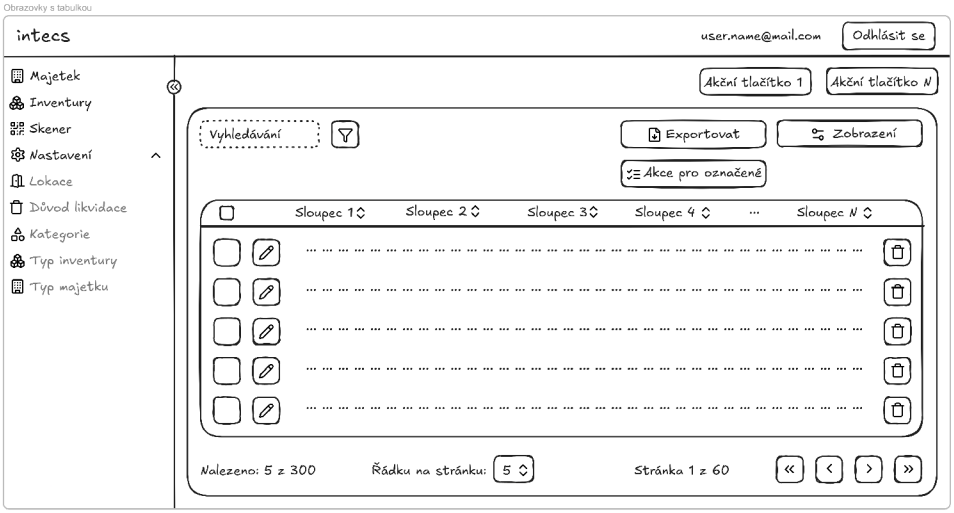
Pri vstupe do systému musí užívateľ zvoliť modul, ktorý ho zaujíma. Pod pojmom modul rozumieme konkrétny systém, ktorý chce užívateľ používať. Momentálne je dostupný len modul pre správu majetku, ktorý reprezentuje systém na správu majetku. V budúcnosti je možné jednoducho pridať ďalší systém pomocou nového modulu, ktorý ho reprezentuje.



Obr. 2.3: Návrh úvodnej obrazovky na výber modulu

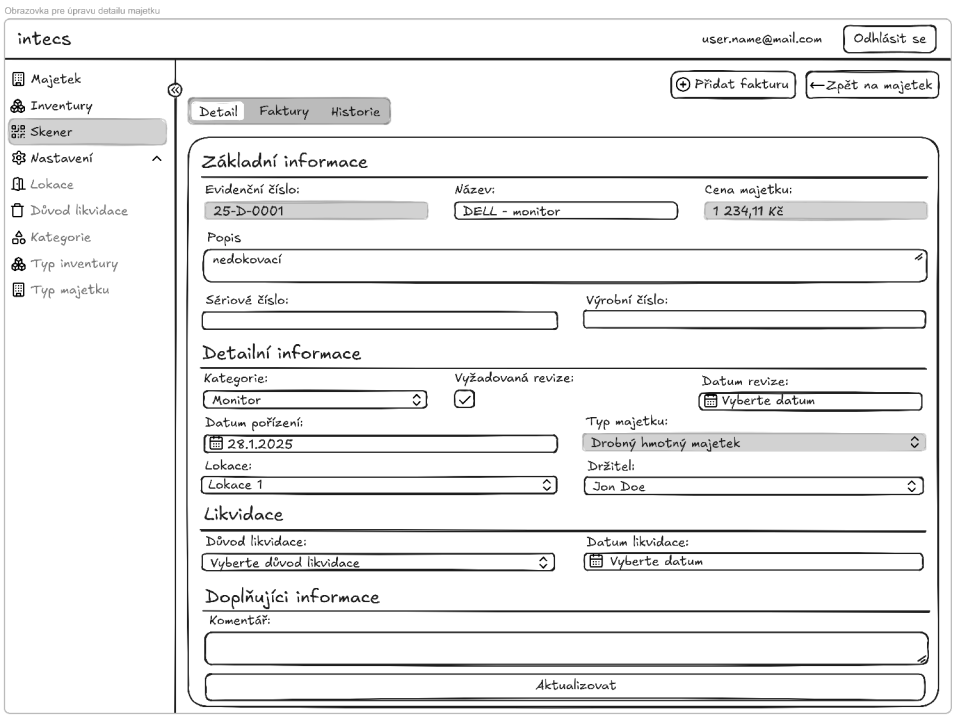
Systém na správu majetku nahradzuje súbor v programe MS Excel, takže je prirodzené, že väčšina obrazoviek tohto systému bude obsahovať tabuľku. Tieto tabuľky obsahujú globálne vyhľadávanie naprieč všetkými dátami, detailnejšie filtrovanie po stĺpcoch, ovládacie prvky na radenie záznamov a skrývanie stĺpcov. Niektoré tabuľky obsahujú možnosť označenia záznamov a následne vykonanie hromadnej akcie nad označenými záznamami. Záznamy, ktoré bude možné editovať alebo mazať obsahujú ikonu, ktorá po kliknutí otvorí modálne okno pre editáciu záznamu prípadne jeho mazanie. V prípade komplexnejších entít ikona na editáciu záznamov namiesto otvorenia modálneho okna presmeruje na separátnu obrazovku. Obrazovky s tabuľkami obsahujú taktiež akcie, ktoré užívateľovi umožnia prechod na inú

obrazovku, prípadne otvoria modálne okno, pomocou, ktorého vie užívateľ vytvoriť nový záznam.



Obr. 2.4: Návrh obrazoviek s tabuľkou

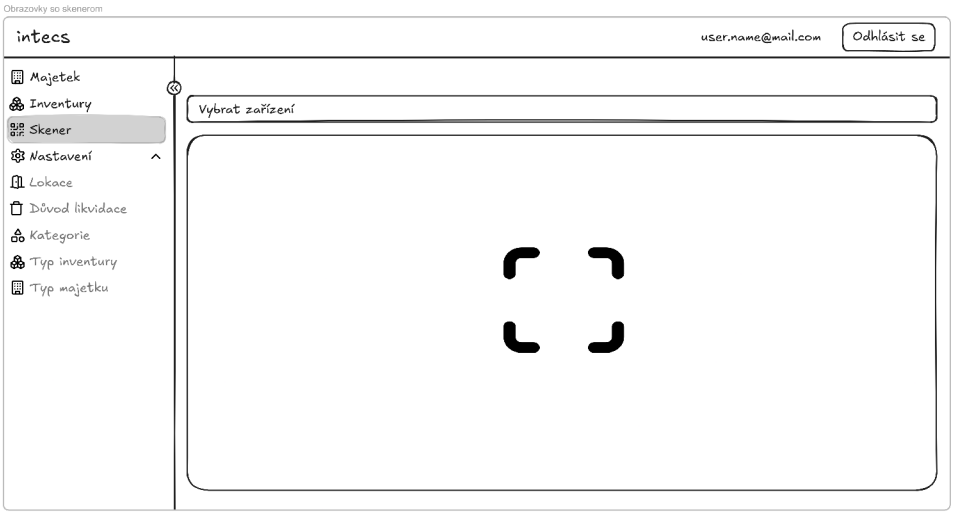
Obrazovka s detailom majetku obsahuje tri karty, medzi ktorými môže užívateľ prepínať. Prvá karta umožňuje užívateľovi upraviť detail majetku pomocou formuláru a ďalšie dve karty obsahujú tabuľky so zoznamom faktúr priradených k majetku a zoznamom zmien na majetku. Tieto karty vyzerajú skoro totožne ako na obrázku 2.4, keďže obsahujú tabuľky. Jediný rozdiel je ovládací prvok na prepínanie kariet, ktorý je zobrazený nad tabuľkou. Bez ohľadu na zobrazenú kartu, môže užívateľ prejsť naspäť na zoznam majetku, prípadne pridať novú faktúru pomocou akčných tlačidiel.



Obr. 2.5: Návrh obrazovky s formulárom pre úpravu detailu majetku

Vytvorenie nového majetku prebieha pomocou formuláru, ktorý je príliš veľký na to, aby bol zobrazený v modálnom okne. Z tohto dôvodu, má vytvorenie nového majetku separátnu obrazovku, ktorá obsahuje podobný formulár ako na obrázku 2.5. Tieto formuláre sa líšia len dátami, ktoré vyžadujú od užívateľa. Napríklad, pri vytváraní majetku je potrebné vyplniť údaje o faktúre, namiesto údajov o likvidácii, prípadne nie je zobrazené evidenčné číslo a dátum revízie.

Súčasťou modulu na správu majetku je aj skener, ktorý ma separátnu obrazovku. Obrazovka so skenerom umožňuje užívateľovi vybrať kameru, pomocou, ktorej sa skenujú QR kódy. Súčasťou skeneru sú aj ovládacie prvky na priblíženie prípadne oddialenie kamery.



Obr. 2.6: Návrh obrazoviek so skenerom

Systém na správu majetku, sa primárne používa na zariadeniach s väčším displejom, no inventúry prebiehajú primárne za pomoci mobilného zariadenia, ktoré má menšiu zobrazovaciu plochu. Jednotlivé obrazovky sa preto, dynamicky prispôsobujú zobrazovacej šírke zariadenia. Medzi hlavné zmeny, ktoré sa automaticky prispôsobia patrí

- pre usporiadanie jednotlivých údajov vo formulároch do jedného stĺpca,
- skrytie bočného menu, ktoré je nahradené za rolovacie menu v hlavičke stránky a
- rozšírenie modálnych okien na celú šírku zariadenia.

Obr. 2.7: Návrh mobilenej obrazovky

2.3 Dátový model

Ako dátové úložisko na väčšine projektov používa firma SQL Server[6] a výnimkou nie je ani správa majetku. Pri vývoji systémov a aplikácii firma používa prístup, kde sa najprv vytvorí databáza[7]. Jedná sa o prístup kde za pomoci nástrojov na reverzné inžinierstvo sa z databázy vytvoria modely, ktoré sa následne používajú v systéme[5]. Tento prístup má tú výhodu, že tvorba a správa databázy sa môže prenechať zamestnancovi, ktorý nie je členom aplikačného tímu, takže

6. `https://www.microsoft.com/en-us/sql-server`
7. Z anglického Database First

vývojár sa môže sústrediť iba na tvorbu aplikácie. Naopak, to komplikuje nasadenie aplikácie, pretože je potrebné súčasne nasadiť novú verziu aplikácie a aplikovať zmeny v databáze. Tento proces je možné automatizovať, ale je to komplikovanejšie keďže to vyžaduje synchronizáciu procesov v dvoch úplne odlišných tímoch. Riešením je používať prístup kde sa najprv tvorí kód[8]. Tento prístup umožní vývojárovi vytvoriť kód, ktorý popíše modely a následne pomocou knižnice vygeneruje databázu[6]. Výhodou tohto prístupu je, že proces nasadenia je veľmi jednoduché automatizovať tak, aby zmeny vykonané v databáze a programovom kóde sa nasadili súčasne. Firma tento prístup zatiaľ nepoužila, ale rozhodla sa ho otestovať v rámci systému na správu majetku.

2.3.1 Dodaný návrh dátového modelu

Pri počiatočnom návrhu systému sa počítalo s prístupom kde sa najprv vytvorí databáza, takže firma dodala entito relačný diagram popisujúci existujúcu databázu pre správu majetku. Tento diagram je k nahliadnutí ako príloha A k diplomovej práci formou súboru **erd-original.svg**. Tabuľka *asignee* je súčasťou systému, ale jej správa prebieha pomocou automatického procesu, ktorý zabezpečuje jej aktualizáciu na základe dát v Microsoft Entra ID. Tento automatický proces nie je súčasťou diplomovej práce, pretože ho firma už dodala pred návrhom systému.

2.3.2 Úprava dátového modelu

Pri návrhu dátového modelu sa vychádzalo z dodaného entito relačného diagramu. Diagram je potrebné upraviť, pretože po jeho vytvorení prišli požiadavky na novú funkcionalitu a nebol navrhnutý na prácu s prístupom kedy sa databáza generuje na základe kódu.

Pôvodný dátový model používa e-mailovú adresu ako identifikátor užívateľa, ktorý vytvoril alebo upravil entitu. Tento prístup nie je ideálny, pretože je potrebné vyhľadať užívateľa na zistenie dodatočných informácii. Namiesto e-mailovej adresy užívateľa je vo výslednom dátovom modely použitá referencia na konkrétneho užívateľa, čo tento problém odstráni.

8. Z anglického Code First

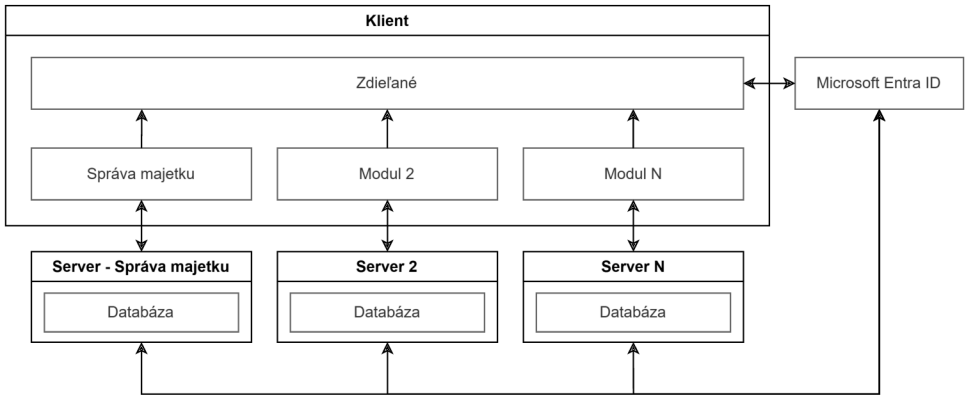
Do niektorých entít je potrebné pridať atribút *DeletedAt*, ktorý je možné použiť na simuláciu mazania entity. Táto zmena je nevyhnutná, pretože údaje o majetku sú historizované a bez tohto atribútu nie je možné mazať entity, ktoré sú priradené k majetku. Za pomoci tohto atribútu je možné simulovať mazanie entity tak, že entita bude stále existovať, takže jej hodnoty bude možné zobrazovať v historických dátach, no užívateľ ju už nebude môcť použiť.

Vďaka prístupu kedy sa generuje databáza podľa kódu, je možné odstrániť tabuľku *stockTakingMethod*. Táto tabuľka je nahradená za enumeráciu, ktorá sa do databázy automaticky ukladá ako textová hodnota, ale v kóde sa s ňou pracuje ako s enumeráciou, ktorá ma obmedzené hodnoty.

Mimo týchto väčších zmien je potrebné pridať a odstrániť atribúty, ktoré udržiavajú dodatočné informácie, ale neprinášajú novú funkcionalitu. Názvy tabuliek a atribútov, sú prispôsobené tak aby sedeli s názvami modelov a obmedzenia na jednotlivé hodnoty boli zrevidované. Výsledný entito relačný diagram je k nahliadnutí ako príloha A formou súboru **erd-final.svg**.

2.4 Atchitektúra

Architektúra systému je rozdelená na klientsku a serverovú časť, pričom obe komunikujú s Microsoft Entra ID.



Obr. 2.8: Návrh architektúry systému

2.4.1 Klienstka časť systému

Komunikácia medzi klientskou časťou a Microsoft Entra ID sa využíva na autentifikáciu užívateľa, zistenie role prihláseného užívateľa a generovanie tokenu, ktorý sa posiela spolu s každou požiadavkou na server.

Klientska časť momentálne obsahuje len modul pre správu majetku, ale v budúcnosti bude možné tieto moduly pridávať. Vopred nie je jasné koľko týchto modulov bude, takže zatiaľ sú separované pomocou priečinkov v rámci jedného projektu. Jednotlivé komponenty a funkcionalita modulov, ktoré budú použiteľné aj pre iné moduly sú mimo priečinku pre modul. Takáto separácia zjednodušuje proces extrakcie jednotlivých modulov do separátnych projektov v prípade, že by ich bolo už veľa a projekt by sa stal neprehľadným.

2.4.2 Serverová časť systému

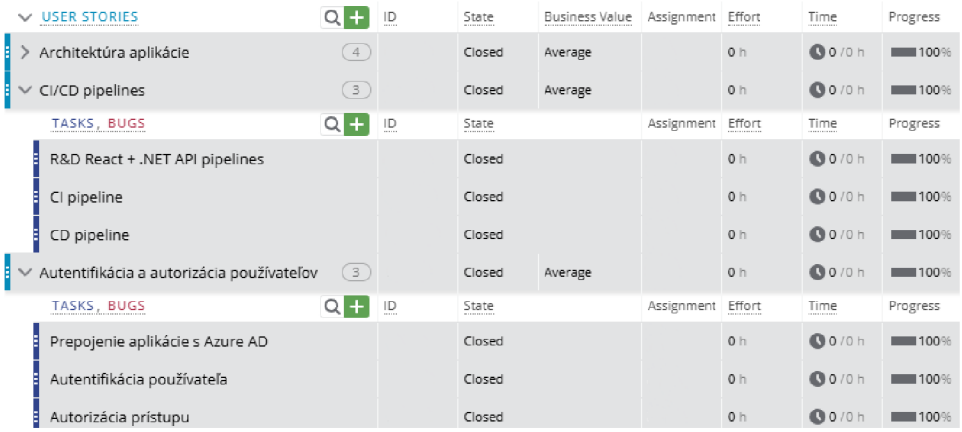
Serverová časť využíva Microsoft Entra ID na validáciu tokenu, ktorý je súčasťou požiadavku. Token obsahuje aj rolu užívateľa pomocou, ktorej je možné kontrolovať oprávnenia na vykonanie požiadavku.

Každý modul v klientskej časti má separátnu serverovú časť a k nej priradenú databázu. Pri pridávaní modulov budú vznikať ďalšie na sebe nezávislé projekty. Funkcionalitu, ktorú je možné použiť pre ďalšie moduly je bez znalosti ďalších modulov ťažké určiť. Zdieľaný projekt, ktorý by obsahoval takúto funkcionalitu bude jednoduché vytvoriť hneď, ako sa takáto funkcionalita objaví, takže zdieľaný projekt zatiaľ nie je súčasťou architektúry. Firma ma k dispozícii internú knižnicu obsahujúcu metódy na prácu s Azure cloudovými službami a rôznymi generickými metódami, ktorú je možné využiť pri implementácii. Táto knižnica nie je braná ako zdieľaná funkcionalita serverovej časti systému, pretože sa používa vo viacerých nezávislých systémoch, ktoré firma vyvíja.

Komunikácia medzi serverovou a klientskou časťou prebieha za pomoci protokolu HTTP, pričom pri pribúdajúcom počte modulov bude nevyhnutná komunikácia aj naprieč rôznymi serverovými časťami. V prípade, že komunikácia pomocou HTTP protokolu bude neprehľadná ju bude možné nahradiť za zbernicu správ.

3 Implementácia

Implementácia klientskej a serverovej časti, prebieha paralelne, pričom sa postupuje po jednotlivých funkcionalitách. Tieto funkcionality sú rozdelené na menšie úlohy a na ich zaznamenávanie je použitý nástroj Targetprocess.



USER STORIES	ID	State	Business Value	Assignment	Effort	Time	Progress
Architektúra aplikácie 4		Closed	Average		0 h	0 / 0 h	100%
CI/CD pipelines 3		Closed	Average		0 h	0 / 0 h	100%
TASKS, BUGS	ID	State		Assignment	Effort	Time	Progress
R&D React + .NET API pipelines		Closed			0 h	0 / 0 h	100%
CI pipeline		Closed			0 h	0 / 0 h	100%
CD pipeline		Closed			0 h	0 / 0 h	100%
Autentifikácia a autorizácia používateľov 3		Closed	Average		0 h	0 / 0 h	100%
TASKS, BUGS	ID	State		Assignment	Effort	Time	Progress
Prepojenie aplikácie s Azure AD		Closed			0 h	0 / 0 h	100%
Autentifikácia používateľa		Closed			0 h	0 / 0 h	100%
Autorizácia prístupu		Closed			0 h	0 / 0 h	100%

Obr. 3.1: Úlohy v nástroji Targetprocess

Systém na správu majetku má tri separátne prostredia, pričom každé z týchto prostredí slúži na niečo iné. Medzi tieto prostredia patria:

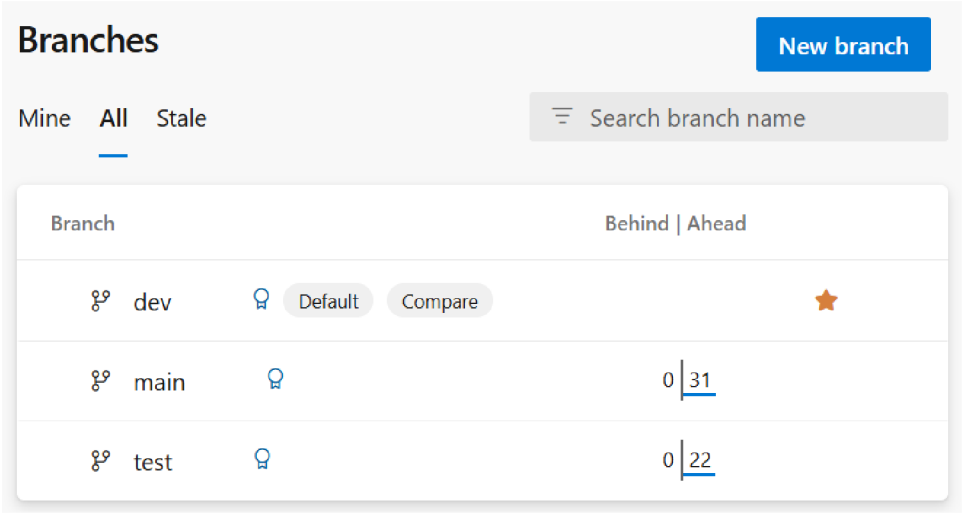
- *Vývojové prostredie* obsahuje zdrojový kód, ktorý sa spúšťa výhradne lokálne a je určené pre vývojárov systému. Zdrojový kód z tohto prostredia nie je nasadený na webovú stránku.
- *Testovacie prostredie* obsahuje zdrojový kód s novou funkcionalitou, ktorý je nasadený na separátnu webovú stránku pomocou, ktorej užívatelia môžu otestovať novú funkcionalitu.
- *Produkčné prostredie* obsahuje zdrojový kód, ktorý bol otestovaný a následne nasadený na webovú stránku, ktorá slúži pre všetkých užívateľov systému.

3.1 Azure Dev Ops

Služba Azure Dev Ops[1] slúži na ukladanie, zdieľanie, verzovanie a nasadenie zdrojového kódu pomocou vzdialených repozitárov a nástrojov na kontinuálnu integráciu a nasadenie.

3.1.1 Nastavenie repozitárov

Zdrojový kód systému je rozdelený na serverovú a klientsku časť pričom pre každú existuje separátny vzdialený repozitár. Tieto repozitáre sú ďalej rozdelené na tri vetvy, pričom každá obsahuje zdrojový kód pre príslušné prostredia.



Obr. 3.2: Vetvy vzdialených repozitárov

Tieto vetvy sú nakonfigurované tak, aby do nich vývojár nemohol pridávať nový kód priamo. Jedinou cestou na pridanie nového kódu, je vytvorenie separátnej vetvy pre nový kód a následná žiadosť o spojenie týchto vetiev[2]. Každú žiadosť kontroluje aspoň jeden vývojár, ktorý sa nepodieľal na implementácii danej funkcionality, čo pomáha odchytiť chyby pred nasadením na testovacie prostredie.

1. `https://azure.microsoft.com/en-us/products/devops`
2. Z anglického Pull request

3.1.2 Kontinuálna integrácia a nasadenie

Nástroje na kontinuálnu integráciu pri každej žiadosti o spojenie vetiev automaticky spúšťajú príkazy, ktoré zostavia kód a skontrolujú dodržiavanie štandardov pre kód. Neúspešné zostavenie kódu alebo nedodržiavanie štandardov vedie k zablokovaniu možnosti spojiť vetvy, pričom oprava týchto problémov vedie k opätovnému povoleniu tejto možnosti.

Zdrojový kód, ktorý sa nachádza vo vetve určenej pre prostredie s webovou stránkou je potrebné vždy nasadiť. Kód môže byť nasadený na webovú stránku ručne, no je to opakujúca sa činnosť, ktorú je možne automatizovať pomocou nástrojov na kontinuálne nasadenie. Pomocou týchto nástrojov sa vždy po tom, čo sa do vetvy určenej pre prostredie s webovou stránkou dostane nový kód, tento kód nasadí na definovanú webovú stránku.

3.2 Cloudové služby Azure

Systém na správu majetku používa tieto cloudové služby Azure:

- *App Service* - Služba založená na HTTP protokole, ktorá slúži na hosťovanie webových aplikácii, aplikačných rozhraní a ďalších, ktoré je možné jednoducho škálovať[7]. Systém na správu majetku túto službu využíva na hosťovanie serverovej časti. Produkčné a testovacie prostredie majú separátny App Service, pričom jediný rozdiel medzi nimi je nastavenie premenných prostredia tak aby používali správnu konfiguráciu.

- *Key vault* - Služba zabezpečujúca bezpečné uloženie a správu prístupov k rôznym tokenom, heslám, certifikátom a ďalším[8]. Systém na správu majetku využíva túto službu na ukladanie pripojovacích reťazcov[3] k databáze, pričom každé prostredie má svoj vlastný Key vault.

- *Storage account* - Služba umožňujúca ukladanie súborov, rad, tabuliek a ďalších, pričom k nim umožňuje pristupovať odkiaľkoľvek pomocou HTTP a HTTPS protokolov[9]. Systém na

3. Z anglického connection string

správu majetku využíva túto službu na hosťovanie klientskej časti. Klientska časť je napísaná v JavaScript-e takže zhotovením aplikácie vzniknú súbory so statickým obsahom, ktoré je možné nahrať do tejto služby. Pre klientsku časť sa používa táto služba, pretože je lacnejšia ako App Service.

- *Microsoft Entra ID* - Služba poskytujúca manažment prístupov a identít, ktorú môžu zamestnanci použiť na prístup k rôznym externým zdrojom[10]. Systém na správu majetku túto službu využíva na umožnenie prihlásenia sa zamestnancov do systému a kontrolu oprávnení a prístupu k jednotlivým častiam. Pre systém na správu majetku existujú v Microsoft Entra ID dve role reprezentujúce administrátorov a bežných zamestnancov. Rola pre administrátorov je pridelená k užívateľským skupinám, ktoré majú mať administrátorský prístup. Tieto skupiny, sú priradené k účtu užívateľa pri nástupe do firmy, takže užívatelia automaticky získajú prístup aj do správy majetku. Rola pre bežných užívateľov nie je priradená k účtom ani skupinám a je určená výhradne pre systém, ktorý ju používa ako východziu rolu pre prihláseného užívateľa.

Medzi jednotlivými službami Azure dochádza k vzájomnej komunikácii, pričom pomocou správy prístupov je zabezpečené aby bola táto komunikácia povolená len v rámci jedného prostredia. To znamená, že služba z testovacieho prostredia nemá oprávnenie komunikovať so službou z produkčného prostredia. Jednotlivé služby je možne konfigurovať pomocou webového portálu od firmy Azure, takže okrem oprávnení komunikácie medzi službami je obdobným spôsobom zabezpečené, že tieto služby budú môcť konfigurovať len oprávnení zamestnanci.

3.3 Klientska časť systému

Klientska časť systému je napísaná pomocou JavaScript-ovej knižnice React ako jednostránková aplikácia[4]. Jednostránková aplikácia je webová aplikácia, ktorá na začiatku načíta celý obsah webového dokumentu a jeho časti následne dynamicky upravuje, bez toho aby

4. Z anglického single page application (SPA)

ho načítala celý znova. Takéto webové stránky sa užívateľom používajú príjemnejšie, ale medzi ich hlavnú nevýhodu patrí obtiažná optimalizácia pre vyhľadávače.[11] Správa majetku je interný systém, ku ktorému zamestnanci pristupujú cez záložku v prehliadači, takže ju nie je potrebné optimalizovať pre vyhľadávače.

Na zjednodušenie nastavenia štýlov jednotlivých častí systému je použitá knižnica *Tailwind CSS*[5], ktorá umožňuje nastaviť štýl jednotlivých komponentov pomocou pridávania menných tried[6]. Vďaka tejto knižnici tak nie je potrebné vytvárať ďalšie súbory na nastavenia štýlov jednotlivých komponentov čo výrazne zrýchľuje samotnú implementáciu. Najväčšou nevýhodou tejto knižnice je, že odstráni všetky východzie štýly z už existujúcich HTML elementov. Tento problém je možné vyriešiť vytvorením vlastných komponentov, ktoré reprezentujú jednotlivé elementy a majú definované štýly. Samotný proces vytvárania komponentov je zdĺhavý, takže pre správu majetku sú použité už existujúce komponenty dostupné z *shadcn*[7]. Shadcn je súbor verejne dostupných komponentov, ktoré je možne skopírovať do projektu manuálne alebo pomocou príkazu. Tieto komponenty sú následne súčasťou projektu, takže ich je možné prispôsobiť potrebám konkrétneho systému. Vďaka tomuto prístupu je potrebné vytvoriť len pár komponentov manuálne a iba pár tých dostupných je potrebné upraviť. Kombinácia shadcn a Tailwind CSS naviac umožňuje veľmi jednoducho meniť farebnú schému systému, pričom na generovanie vstupného súboru pre túto schému je možné využiť webovú stránku, ktorá tento proces výrazne zjednodušuje.

3.3.1 Štruktúra projektu

Štruktúra projektu je rozdelená na dva logické celky, pričom jeden obsahuje komponenty a funkcionalitu, ktorú bude možné využiť pre budúce moduly a druhý obsahuje komponenty a funkcionalitu, ktorá je využiteľná striktne pre správu majetku. Tieto logické celky sú rozdelené pomocou priečinkov, čo výrazne urýchľuje samotnú implementáciu a zároveň umožňuje pridávanie nových modulov.

5. `https://tailwindcss.com/`
6. Z anglického class name
7. `https://ui.shadcn.com/`

Štruktúra prvého celku, ktorý obsahuje zdieľanú funkcionalitu a komponenty vyzerá nasledovne:

```
src
├── assets
├── auth
├── components
│   ├── ui
│   └── ...
├── hooks
├── layout
├── models
├── modules
│   └── property
├── state
├── utils
├── App.tsx
├── main.tsx
└── ...
```

Priečinok *assets* obsahuje obrázky a nastavenie farebnej schémy, ktoré sa používajú naprieč celým systémom, čo zabezpečuje jednotný dizajn bez ohľadu na modul. Nastavenie autentifikácie a autorizácie je v priečinku *auth*. Všetky komponenty, ktoré sú zdieľané pre viacero modulov sa nachádzajú v priečinku *components*, pričom tie z shadcn sú oddelené od tých manuálne vytvorených pomocou priečinku *ui*. Rozloženie komponentov, ktoré zdieľajú všetky stránky sú uložené v priečinku *layout*. Niektoré komponenty potrebujú na správne vykreslenie komplexnejšie dáta, ktoré sú reprezentované modelmi v priečinku *models*. Jednotlivé komponenty si udržujú svoj stav, ktorý zdieľajú s inými komponentami. V prípadoch, kedy je zdieľanie týchto stavov medzi komponentami komplikované z dôvodu veľkého počtu zanorení sa používajú globálne stavy a tie sú uložené v priečinku *state*. Funkcionalita, ktorú zdieľajú viaceré komponenty je uložená v priečinku *hooks*, prípadne v *utils*, ak ju zdieľajú iné súbory. Priečinok *modules* zabezpečuje oddelenie jednotlivých modulov, pričom jediný modul v ňom je modul pre správu majetku. Jednoduchým pridaním nového priečinku je možné priať ďalší modul reprezentujúci nový systém.

Štruktúra logického celku pre správu majetku je podobná štruktúre zdieľaného celku. Priečinky obsahujúce konfiguráciu a nastavenie nie sú súčasťou, pretože nastavujú systém ako celok a nie jednotlivé moduly. Stránky, ktoré vykresľujú komponenty sú uložené v priečinku

pages, pričom niektoré obsahujú formulár. Tieto formuláre obsahujú validáciu vstupných dát, ktorá je definovaná pomocou schém, a tie sú uložené v priečinku *schemas*. Modul pre správu majetku komunikuje so serverovou časťou pomocou súborov uložených v *_generatedClient* a *services*.

3.3.2 Autentifikácia a autorizácia užívateľov

Knižnica od firmy Microsoft zabezpečuje autentifikáciu a autorizáciu pomocou účtov v Microsoft Entra ID. Proces prihlásenia prebieha na stránkach firmy Microsoft, na ktorú sú užívatelia automaticky presmerovaní pri vstupe do systému a po úspešnom prihlásení sú presmerovaní naspäť. Pomocou jednotného prihlásenia je zabezpečené, že užívateľ je prihlásený do systému automaticky, ak bol s rovnakým účtom prihlásený do inej webovej stránky. Po úspešnom prihlásení knižnica umožňuje získavanie informácii ako meno, e-mailová adresa a rola prihláseného užívateľa.

Systém môže mať v budúcnosti viacero modulov, pričom pre každý modul budú existovať separátne role, ktoré bude mať užívateľ priradené. Systém obsahuje funkciu zdieľanú naprieč modulmi, ktorá dostane na vstupe zoznam povolených rolí a východziu rolu. Zoznam prechádza zľava doprava a vráti prvú rolu, ktorú má užívateľ priradenú, inak vráti východziu hodnotu. Modul na správu majetku obsahuje separátnu funkciu, ktorá volá zdieľanú funkciu na zistenie role, pričom do zoznamu uvedie role určené pre správu majetku od role s najvyššími oprávneniami, po rolu s najnižšími oprávneniami, a ako východziu hodnotu používa rolu s najnižšími oprávneniami. Takéto nastavenie zabezpečí, že ak daný užívateľ bude mať viac ako jednu rolu, tak vždy vráti tú s najvyššími oprávneniami.

3.3.3 Komunikácia so serverovou časťou systému

Komunikácia medzi klientskou a serverovou časťou prebieha pomocou HTTP protokolu. Serverová časť poskytuje definičný súbor vo formáte JSON, ktorý popisuje adresy, vstupné modely a výstupné modely jednotlivých koncových bodov[8]. Na vygenerovanie tried, funkcií a typov, ktoré sa následne používajú v celom module na správu majetku

8. Z anglického endpoint

sa používa knižnica. V klientskej časti sú definované dva skripty, ktoré stiahnu definičný súbor a generujú nové funkcie a modely. Pre celý modul je definovaná jedna inštancia vygenerovanej triedy, do ktorej sa posiela funkcia, ktorá pred každým volaním vygeneruje nový token a pripojí ho ku požiadavku pre serverovú časť. Tento token obsahuje informácie o prihlásenom užívateľovi a je nevyhnutný pre úspešnú autentifikáciu a autorizáciu požiadavku.

Vygenerované súbory sa ukladajú do priečinku *_generatedClient*, ktorý je samostatný pre každý modul. Vďaka tejto separácii bude pri pridaní ďalšieho modulu možné definovať nové skripty na stiahnutie definičného súboru a generovanie separátnej triedy, funkcií a typov bez väčších problémov.

3.4 Serverová časť systému

Serverová časť je napísaná pomocou programovacieho jazyku C#, ako aplikačné rozhranie. Na rozdiel od klientskej časti sú jednotlivé moduly implementované ako kompletne separátne aplikačné rozhrania. Zdieľaná logika medzi modulmi nie je extrahovaná do separátneho projektu, pretože nebolo možné identifikovať časti vhodné na zdieľanie. V prípade nového modulu je možné takúto logiku extrahovať do separátneho projektu alebo *NuGet* balíčku, ktorý budú používať moduly.

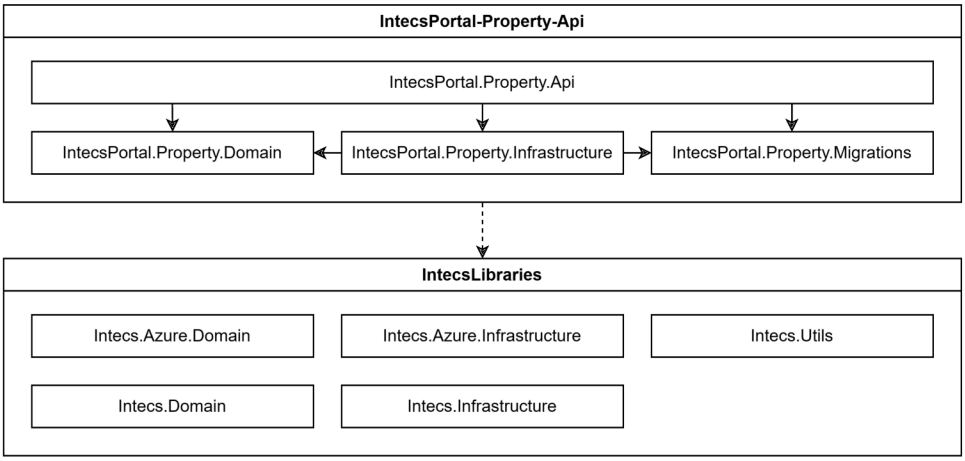
3.4.1 Štruktúra serverovej časti

Štruktúra modulu na správu majetku pozostáva zo štyroch projektov, ktoré reprezentujú špecifické vrstvy, ktoré medzi sebou komunikujú a preto sa navzájom referencujú aby mohli využívať funkcionalitu iných projektov. Jedná sa o tieto projekty:

```
IntecsPortal-Property-Api
├─ IntecsPortal.Property.Api
├─ IntecsPortal.Property.Domain
├─ IntecsPortal.Property.Infrastructure
└─ IntecsPortal.Property.Migrations
```

Okrem vyššie spomenutých projektov sa využívajú aj projekty z internej knižnice, ktoré poskytujú metódy na prácu so službou Key Vault, databázou a atribútmi. Projekty serverovej časti referencujú

projekty internej knižnice, ktorých funkcionalitu využívajú, ale na obrázku 3.3 sú tieto referencie vynechané pre prehľadnosť.



Obr. 3.3: Štruktúra serverovej časti

3.4.2 Autentifikácia a autorizácia užívateľov

Autentifikácia a autorizácia je zabezpečená pomocou knižnice od firmy Microsoft za pomoci účtov v Microsoft Entra ID. Každá požiadavka na serverovú časť musí obsahovať token reprezentujúci užívateľa, inak serverová časť odpovedá chybou. Token sa následne kontroluje voči službe Microsoft Entra ID a overuje sa, že bol vytvorený práve touto službou. Server odpovedá chybou v prípade, že token nebol vytvorený touto službou.

Po úspešnej autentifikácii sa kontroluje to či daný užívateľ má oprávnenia na vykonanie daného požiadavku. Každý koncový bod[9] serverovej časti má pomocou atribútu definované, aké role majú oprávnenie na prístup. Server odpovedá chybou, v prípadoch, kedy token reprezentuje užívateľa, ktorý nemá požadovanú rolu.

Informácie o užívateľovi z Microsoft Entra ID sú dostupné v tvrdeniach[10], ktoré sú po úspešnej autentifikácii a autorizácii doplnené o informácie z databázy. Databáza obsahuje doplňujúce informácie o

9. Z anglckého endpoint
10. Z anglického claims

užívateľovi, ktoré sú synchronizované s Microsoft Entra ID, takže ich je možné spárovať s údajmi v tvrdeniach na základe jednoznačného identifikátora. Toto rozšírenie tvrdení umožňuje pristúpiť k akýmkoľvek údajom o užívateľovi pri vykonávaní požiadavku.

3.4.3 Vrstva infraštruktúry

Projekt *IntecsPortal.Property.Infrastructure* reprezentuje vrstvu infraštruktúry, ktorej primárny účel je práca s databázou. Hlavnou súčasťou projektu sú definície entít a ich konfigurácia, ktoré reprezentujú jednotlivé tabuľky databázy. Tieto entity sú použité v databázovom kontexte na základe, ktorého sa generujú migračné súbory, ktoré majú v názve časovú známku, takže ich je možné usporiadať v čase. Každý súbor obsahuje dve metódy, pričom jedna upraví databázu na novší stav a druhá na stav predchádzajúcej migrácie. Tieto migrácie je nutné spustiť pomocou príkazu do konzoly alebo kódu pričom príkaz automaticky aplikuje migrácie podľa toho v akom stave sa aktuálne databáza nachádza. Na ukladanie migračných súborov sa používa separátny projekt *IntecsPortal.Prperty.Migrations*.

Mimo entít reprezentujúcich tabuľky sú definované entity obsahujúce vlastnosti[11], ktoré zdieľajú viaceré entity. Tieto entity sú použité formou dedičnosti pre entity, ktoré vyžadujú tieto vlastnosti. Prvá z nich obsahuje vlastnosť *Id*, ktorú využívajú všetky entity. Druhá z nich reprezentuje sledovateľnú entitu, pri ktorej je potrebné sledovať kto a kedy daný záznam vytvoril prípadne upravil. Databázový kontext slúži na prácu s jednotlivými tabuľkami a obsahuje metódy, ktoré tieto zmeny uložia. Okrem východzích metód na ukladanie zmien sú v ňom definované rozširujúce, ktoré na vstupe dostanú Id užívateľa. Vďaka týmto metódam nie je potrebné pri každej sledovateľnej entite definovať, kto a kedy ju upravil prípadne vytvoril, pretože to je pri ukladaní automaticky doplnené.

V systéme sa namiesto databázového kontextu používajú repozitáre, ktoré sú definované v tejto vrstve. Každá entita má svoj repozitár, ktorý obsahuje metódy na prácu s danou entitou a na ich vstupe a výstupe sa používajú doménové modely. Tieto modely sa pomocou knižnice mapujú na entity a naopak.

11. Z anglického properties

Produkčné a testovacie prostredie je napojené na existujúcu databázu, ktorá obsahuje dáta. Na zabezpečenie súbežnej práce nad databázou vo vývojovom prostredí je použitá služba *Docker*, ktorá umožňuje spustenie lokálnej inštancie databázy bez toho aby bola dostupná na internete. Každý vývojár tak pracuje nad svojou kópiou databázy a tým pádom si ju navzájom neprepisujú.

3.4.4 Doménová vrstva

Doménovú vrstvu reprezentuje projekt *IntecsPortal.Property.Domain* a jej primárny účel je poskytovanie biznisovej logiky. Táto logika je implementovaná ako metódy v službách[12], ktoré sú rozdelené podľa entít, s ktorými pracujú.

Okrem služieb tu sú definované objekty na dátový transfer a doménové modely. Doménové modely sa posielajú do vrstvy infraštruktúry a vznikajú z objektov na dátový transfer, ktoré prichádzajú z aplikačnej vrstvy. Na mapovanie medzi týmito modelmi sa používa rovnaká knižnica ako vo vrstve infraštruktúry. Pri mapovaní z niektorých objektov na dátový transfer do doménových modelov sú potrebné dodatočné údaje získané z biznisovej logiky, a preto, je nutné definovať objekty na dátový transfer v doménovej vrstve a nie aplikačnej vrstve. Ako vstup a výstup v doménovej vrstve sú použité objekty na dátový transfer.

Mimo vyššie spomenuté sú súčasťou doménovej vrstvy aj definície výnimiek, ktoré sa používajú na signalizovanie chyby, ktorú je možné v aplikačnej vrstve korektne spracovať a užívateľovi vrátiť korektnú chybovú hlášku. Taktiež tu sú definované kontrakty pre služby a repozitáre z vrstvy infraštruktúry a konštanty, ktoré sa používajú naprieč všetkými vrstvami.

Doménová vrstva je taktiež zodpovedná za generovanie QR kódov. Generovanie QR kódov prebieha pomocou dvoch knižníc, ktoré umožňujú vygenerovanie QR kódu a prácu so súborom vo formáte PDF. Vygenerované QR kódy sa vkladajú na správne miesto v PDF súbore a vedľa nich sa generuje evidenčné číslo, kategória a názov majetku. Základnou jednotkou PDF súborov sú body[13], takže dimenzie papiera, ktoré sú v centimetroch sa prepočítavajú na body. Pri generovaní kódov sa najprv vynechá potrebný počet nálepiek, ktoré

12. Z anglického services
13. Z anglického points

už boli na danom papieri použité a následne sa postupuje po jednom QR kóde. Tento postup zabezpečuje čo najväčšiu precíznosť, takže vygenerovaný QR kód a text sedí presne na pozíciu nálepky na lepiacom papieri, bez ohľadu na to či sa nachádza na jeho konci alebo začiatku. Výsledné PDF obsahuje viac stránok s QR kódmi v prípade, že ich je viac ako počet nálepiek na jednom papieri.

3.4.5 Aplikačná vrstva

Projekt *IntecsPortal.Property.Api* reprezentuje aplikačnú vrstvu a slúži primárne na definovanie koncových bodov aplikačného rozhrania, ktoré poskytujú funkcionalitu klientskej časti systému.

Jednoduchšie dáta, ako číselné identifikátory sa posielajú do koncových bodov pomocou parametrov v adrese koncového bodu. Tie komplexnejšie pomocou formátu JSON ako telo požiadavky na koncový bod. Tieto dáta sú následne automaticky skonvertované na objekt na dátový transfer, ktorý sa posiela do doménovej vrstvy na spracovanie požiadavku.

Každá požiadavka, ktorú spracuje doménová vrstva môže skončiť s chybou, ktorá je pomocou *middleware-u* odchytená a skonvertovaná na HTTP odpoveď, ktorá odpovedá danej chybe. Vďaka tomuto nie je potrebné odchytávať jednotlivé výnimky v každom koncovom bode a odchytávanie je definované na jednom mieste. V prípade neočakávanej výnimky sa automaticky skonvertuje na generickú chybu, ktorú je vhodné zobraziť užívateľovi.

Súčasťou tohto projektu je aj aplikačná konfigurácia, ktorá je definovaná pre každé prostredie, v ktorom môže aplikácia fungovať. V tejto konfigurácii sú definované informácie potrebné na pripojenie sa k službám Key vault a Microsoft Entra ID. Okrem týchto údajov, tu sú uložené dimenzie a rozmery papieru, na ktorý sa tlačia QR kódy vygenerované systémom. Nie sú definované v databáze, pretože zatiaľ nie je potrebné ich často meniť, ale v budúcnosti je možné tieto údaje presunúť do databázy a umožniť ich modifikovať priamo užívateľom modulu na správu majetku.

4 Testovanie a spätná väzba

Počas celej implementácie a po jej ukončení prebiehalo tetovanie, pričom výsledkom testovania bola spätná väzba, ktorú bolo potrebné zapracovať.

4.1 Metóda testovania

Systém bol testovaný výhradne manuálne prostredníctvom zamestnanca, ktorý používal najčastejšie pôvodnú správu majetku. V priebehu implementácie boli testované jednotlivé funkcionality, ktoré sa vždy po implementácii nasadili na testovacie prostredie. V prípade nájdenia chyby, boli tieto chyby opravené a následne znova otestované na testovacom prostredí. V prípade, že sa nejednalo o chybu voči zadaniu, boli vytvorené nové úlohy, ktoré boli zapracované samostatne.

Po dokončení implementácie sa začal systém aktívne používať, pričom paralelne pri jeho používaní dochádzalo k testovaniu a návrhom na vylepšenie. Každá chyba bola podľa závažnosti opravená a návrhy na vylepšenie boli prediskutované v rámci firmy. Návrhy, ktoré nemali dostatočný prínos boli zahodené a neboli implementované. Prvotné používanie predstavovalo kontrolu dát, opravu chýb v dátach, ktoré vznikli pri plnení databázy zo súboru pôvodnej správy majetku a prvú inventúru.

4.2 Spätná väzba a jej zapracovanie

Testovanie odhalilo len pár chýb, ktoré boli opravené obratom po ich nahlásení. Jednalo sa o doladenie podmienok kedy je možné mazať niektoré entity a chybnú validáciu vstupných dát v jednom formulári. Hlavnou príčinou týchto chýb bol nedostatočný popis v úlohe po tom, čo sa dohodli tieto funkcionality na video hovore.

Hlavným výsledkom testovania boli požiadavky na doplňujúcu funkcionalitu, ktorá spríjemní užívateľom používanie systému. Tieto požiadavky pozostávali prevažne z drobností ako zvýraznenie testovacieho prostredia pomocou použitia červenej farby pre pozadie navigačného menu namiesto modrej, pridanie vyhľadávania pre po-

ložky vo formulároch, ktoré sa vyberajú zo zoznamu hodnôt, úprava východzieho počtu riadkov v tabuľkách a podobné.

Jednou z požiadaviek bolo aj dočasné ukladanie filtrov v tabuľkách a radenie stĺpcov. Pre užívateľov to bolo dôležité najmä v zozname všetkého majetku kde užívateľ po vyfiltrovaní a zoradení majetku šiel na detail majetku. Po návrate na zoznam majetku musel záznamy znova filtrovať a radiť, čo bolo zdĺhavé a nepríjemné. Tradičný spôsob pre ukladanie filtrovania a radenia tabuliek je uložiť ich do adresy stránky. Výhodou tohto riešenie je možnosť zdieľania odkazu na stránku, pričom užívateľ, ktorý takýto odkaz použije uvidí rovnaké filtrovanie a radenie ako užívateľ, ktorý ho poslal. Toto riešenie nebolo pre správu majetku vhodné, pretože modul obsahuje viacero pre klikov na zoznam majetku, pričom nie vždy sa jedná o krok späť v histórii. Implementované riešenie ukladá filtre a radenie pre všetky tabuľky do pamäte prehliadača, čo zabezpečí dostatočne dlhé uloženie a ich načítanie pri vstupe akýmkoľvek spôsobom na danú tabuľku.

Druhým veľkým problémom, ktorý mali užívatelia bola nekonzistencia v názvoch majetku. Názvy majetku boli často krát veľmi podobné, pretože firma nakúpila viacero produktov rovnakej značky a líšili sa len modelom. Typický príklad sú notebooky kde sa názvy líšia len číslom a písmenom modelu, pričom niekto píše do názvu majetku najprv značku a potom model a niekto iný najprv model a potom značku. Nejedná sa o veľkú nekonzistenciu, ale takto rozhádzané názvy potom komplikujú radenie majetku podľa názvu. Jedným z riešení bolo oddeliť názov a model majetku, pričom bolo možné vytvoriť doplňujúci atribút, prípadne model ukladať do popisu majetku. Obe tieto varianty neboli ideálne, pretože užívatelia chceli vidieť tento názov aj na QR kóde, takže by bolo potrebné upraviť aj generovanie kódu. Implementované riešenie je generovanie zoznamu názvov majetku na základe hodnôt, ktoré v systéme už sú, pričom užívateľ môže zadať úplne novú hodnotu, prípadne vybrať zo zoznamu. Zoznam sa automaticky filtruje pri písaní, takže užívateľovi ponúka len relevantné možnosti.

Záver

Cieľom tejto diplomovej práce bolo navrhnúť a implementovať digitálny systém pre správu majetku pre firmu intecs data, s.r.o.. Tento systém mal nahradiť existujúcu správu majetku, ktorá mala nedostatky ako nekonzistencia dát, chýbajúce dáta a dlhotrvajúca pravidelná ročná inventúra. Pôvodná správa majetku prebiehala za pomoci súboru v programe MS Excel.

Proces tvorby tohto systému začal spracovaním požiadaviek na systém, ktoré poskytla firma. Tieto požiadavky bolo potrebné zanalyzovať a prípadne nejasnosti vykomunikovať so zamestnancom, ktorý používal pôvodnú správu majetku najčastejšie. Väčšina požiadaviek popisovala funkcionalitu, ktorú poskytoval súbor v programe MS Excel, ale zároveň existovali také, ktoré popisovali novú funkcionalitu. Medzi novú funkcionalitu patrilo napríklad generovanie QR kódov, ktoré zrýchlilo pravidelnú ročnú inventúru a sprístupnenie systému všetkým zamestnancom, pričom bežní užívatelia majú prístup len k majetku, ktorý je priradený k ich osobe.

Po spracovaní požiadaviek bolo potrebné upraviť dátový model, ktorý dodala firma tak, aby sa v systéme dobre používal. Následne bolo potrebné navrhnúť architektúru systému tak aby do neho bolo možné v budúcnosti pridávať ďalšie systémy, ktoré firma bude potrebovať. Samotný návrh architektúry nemal byť príliš zložitý, pretože nebolo isté, či budú vznikať ďalšie systémy. Architektúra klientskej časti bola navrhnutá tak, aby všetky systémy boli v rámci jedného projektu ako separátne moduly oddelené pomocou priečinkov. Architektúra serverovej časti bola naopak navrhnutá tak aby, jednotlivé moduly boli ako separátne projekty.

Takto navrhnutý systém bol následne implementovaný, pričom počas jeho implementácie dochádzalo k testovaniu jednotlivých funkcionalít. Chyby odhalené testovaním boli opravené a návrhy na vylepšenia, ktoré schválila firma boli implementované. Po dokončení implementácie prebehla finálna kontrola počas, ktorej zamestnanec firmy upravil nezrovnalosti v dátach, ktoré neboli odstránené pri migrácii dát a uskutočnil prvú inventúru.

V súčasnosti sa tento systém pravidelne používa a pridáva sa do neho nový majetok, ktorý firma nakúpila. Do budúcna sa plánuje

pridanie funkcionality, ktorá umožní generovanie predávacích a likvidačných protokolov. Taktiež sa zvažuje pridanie nového modulu, pre už existujúci systém na správu faktúr, ktorý je postavený nad staršou technológiou a migrácia do nového systému by umožnila pridanie nových funkcionalít.

Bibliografia

1. ČESKO. *§ 29 odst. 1 úZ č. 56/2002 Sb., zákona č. 563/1991 Sb., o účetnictví, jak vyplývá z pozdějších změn - znění od 22. 2. 2002*. 2002. Dostupné tiež z: https://www.zakonyprolidi.cz/cs/2002-56#p29-1. In: Zákony pro lidi.cz [online]. © AION CS 2010–2025 [cit. 24. 2. 2025].
2. SOMMERVILLE, Ian; SAWYER, Pete. *Requirements Engineering: A Good Practice Guide*. 1st. USA: John Wiley & Sons, Inc., 1997. ISBN 0471974447.
3. *What is single sign-on in Microsoft Entra ID?* [online]. omondiatieno a kol., 2024-09-30 [cit. 2025-03-19]. Dostupné z : https://learn.microsoft.com/en-us/entra/identity/enterprise-apps/what-is-single-sign-on.
4. *ASP.NET MVC Overview* [online]. Rick-Anderson a kol., 2023-06-16 [cit. 2025-03-20]. Dostupné z : https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview.
5. *Database First* [online]. SamMonoRT a kol., 2020-10-14 [cit. 2025-03-23]. Dostupné z : https://learn.microsoft.com/en-us/ef/ef6/modeling/designer/workflows/database-first.
6. *Creating a Model* [online]. SamMonoRT a kol., 2020-10-14 [cit. 2025-03-23]. Dostupné z : https://learn.microsoft.com/en-us/ef/ef6/modeling.
7. *App Service overview* [online]. msangapu-msft a kol., 2025-03-18 [cit. 2025-04-01]. Dostupné z : https://learn.microsoft.com/en-us/azure/app-service/overview.
8. *About Azure Key Vault* [online]. msmbaldwin a kol., 2024-09-11 [cit. 2025-04-01]. Dostupné z : https://learn.microsoft.com/en-us/azure/key-vault/general/overview.
9. *Storage account overview* [online]. akashdubey-ms a kol., 2025-03-04 [cit. 2025-04-01]. Dostupné z : https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview.

10. *What is Microsoft Entra ID?* [online]. barclayn a kol., 2025-03-05 [cit. 2025-04-01]. Dostupné z : `http://learn.microsoft.com/en-us/entra/fundamentals/whatis`.

11. *SPA (Single-page application)* [online]. mfuji09 a kol., 2024-12-19 [cit. 2025-04-02]. Dostupné z : `https://developer.mozilla.org/en-US/docs/Glossary/SPA`.

A Zoznam príloh

Súčasťou tejto práce sú prílohy, ktoré sa nachádzajú v informačnom systéme Masarykovej univerzity[1]. Jedná sa o tieto prílohy:

- Súbor **erd-original.svg** zobrazuje entito relačný diagram, ktorý firma dodala pred návrhom systému. Autor tejto diplomovej práce nie je autorom tohto diagramu.
- Súbor **erd-final.svg** zobrazuje finálny entito relačný diagram, ktorý vytvoril autor tejto diplomovej práce, na základe súboru **erd-original.svg**.
- Priečinok **IntecsPropertyApp.zip** obsahuje zdrojové kódy systému pre správu majetku a súbor **README.md**, ktorý popisuje potrebné kroky pre spustenie systému.
- Súbor **vyjadrenie.pdf** obsahuje vyjadrenie vedúcej prevádzky[2], ktorá používa systém pre správu majetku najčastejšie.

1. `https://is.muni.cz/th/yif3e/`
2. Z anglického Head of Operations