Dochádzkový systém

Bc. Jakub Vojtašák

Diplomová práca
2021

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav elektroniky a měření

Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Jakub Vojtašák**
Osobní číslo:	**A19505**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Forma studia:	**Kombinovaná**
Téma práce:	**Docházkový systém**
Téma práce anglicky:	**An Attendance System**

Zásady pro vypracování

1. Proveďte rešerši existujících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu.
3. Proveďte rozbor a analýzu požadavků na zvolené řešení.
4. Zpracujte model navržené aplikace.
5. Věnujte pozornost zabezpečení aplikace.

Forma zpracování diplomové práce: **Tištěná/elektronická**

Seznam doporučené literatury:

1. NEUSTADT, Ila; ARLOW, Jim. UML 2 a unifikovaný proces vývoje aplikací Computer Press, Albatros Media as, 2016.
2. JOHNSON, Glenn. Programming in HTML5 with JavaScript and CSS3: training guide. Redmond, Wash.: Microsoft, 2013. ISBN 978-0735674387.
3. UNHELKAR, Bhuvan. Software engineering with uml Auerbach Publications, 2017.
4. LETT, Jacob. Bootstrap 4 Quick Start: A Beginners Guide to Building Responsive Layouts with Bootstrap 4 Bootstrap Creative, 2018.
5. AKOBUS, Benjamin. Mastering Bootstrap 4: Master the latest version of Bootstrap 4 to build highly customized responsive web apps Packt Publishing Ltd, 2018.
6. BEN-GAN, Itzik; DAVIDSON, Louis; VARGA, Stacia. MCSA SQL Server 2016 Database Development Exam Ref 2-pack: Exam Refs 70-761 and 70-762 Microsoft Press, 2017.

Vedoucí diplomové práce: **doc. Ing. Petr Šilhavý, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce: **15. ledna 2021**
Termín odevzdání diplomové práce: **17. května 2021**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
DĚKAN
L.S.

Ing. Milan Navrátil, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2021

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky. Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 14. 5. 2021 Jakub Vojtašák v. r.

podpis studenta

ABSTRAKT

Táto diplomová práca sa zaoberá návrhom a implementáciou dochádzkového systému, ktorý je navrhnutý ako webová aplikácia a nasadený na platformu Azure. Teoretická časť sa venuje funkciám dochádzkového systému, vybraným existujúcim riešeniam, legislatívnym požiadavkám a požiadavkám zadávateľa. Pri návrhu a vývoji je kladený dôraz na bezpečnosť, preto sú opísané základné hrozby pri webových aplikáciách. Súčasťou tejto časti je aj popis technológií, ktoré boli pri implementácií použité. Praktická časť sa zaoberá návrhom a vývojom systému. Systém je implementovaný ako REST-API aplikácia, ktorá využíva na aplikačnej časti framework .NET Core 5 a na prezentačnej časti Javascriptovú knižnicu React. Súčasťou praktickej časti je kapitola, ktorá je zameraná na spôsoby zabezpečenia pred vybranými hrozbami webových aplikácií.

Klíčová slova: Dochádzkový systém, evidencia odpracovných hodín, webová aplikácia, .NET 5, C#, React.JS, Azure.

ABSTRACT

This diploma thesis deals with the design and implementation of an attendance system, which is designed as a web application and deployed on the Azure platform. The theoretical part deals with the functions of the attendance system, chosen existing solutions, legislative requirements and the needs of the client. During design and development emphasis is placed on security, therefore the basic threats of web applications are described. Part of this section is also a description of the technologies that were used in the implementation. The practical part deals with the design and development of the system. The system is implemented as a REST-API application, which uses the .NET Core 5 framework on the back-end and the React Javascript library on the front-end. Part of the practical part is a chapter that focuses on the ways in which the system is secured against selected web application threats.

Keywords: Attendence system, evidence of time-entries, web application, .NET 5, C#, React.JS, Azure.

Ďakujem svojmu vedúcemu práce doc. Ing. Petrovi Šilhavému, Ph.D. za odbornú pomoc a cenné pripomienky počas realizácie práce. Tiež by som chcel poďakovať svojej rodine a priateľom, ktorí ma počas štúdia a písania tejto práce podporovali.

OBSAH

ÚVOD

Rozvoj firiem a nárast počtu zamestnancov sa v dnešnej dobe zrýchľuje obrovským tempom. S nárastom počtu zamestnancov narastá aj administratívna náročnosť na správu, kontrolu ich dochádzky a odpracovaných hodín.

Zapisovanie a ukladanie informácií v písanej forme je už prežitok a firma nemôže byť konkurencieschopná, ak by sa nesnažila čo najviac údajov držať v digitálnej forme. Samozrejme, aj papierová forma ma oproti tej digitálnej určité výhody, avšak výhod digitálnej formy je omnoho viac. Medzi hlavné výhody patrí ich prístupnosť, rýchlosť vyhľadávania, možnosť predávania a v prípade využívania vhodných nástrojov, aj ich spravovanie a samotné vytváranie týchto informácií. Inak tomu nie je ani v prípade administratívnych informácií o zamestnancoch.

Každá spoločnosť, aj v prípade evidencie zamestnancov, musí dodržiavať legislatívu, ktorá prikazuje, ktoré všetky informácie je nutné o jednotlivých zamestnancoch viesť. Množstvo požadovaných informácií, ktoré musia zamestnávatelia viesť o svojich zamestnancoch sa neustále zvyšuje. Ukladanie týchto informácií vo forme papierových katalógov a záznamov je extrémne neefektívne, preto je nutné tieto informácie nejakým spôsobom digitalizovať. V prípade firiem s malým počtom zamestnancov, pripadá do úvahy ukladať tieto informácie vo forme digitálneho katalógu, akým je napríklad Microsoft Excel, avšak tieto katalógy bývajú pri väčšom počte zamestnancov neprehľadné. Preto začali vznikať rôzne informačné systémy, ktoré uľahčujú zamestnávateľom a ich administratívnym pracovníkom prácu s týmito údajmi.

Cieľom teoretickej časti diplomovej práce je analyzovať požiadavky na dochádzkový systém pre malú IT spoločnosť. Po zozbieraní týchto požiadaviek, budú porovnané a preskúmané vybrané dostupné riešenia, ktoré trh aktuálne ponúka. Taktiež budú vymenované a opísané najznámejšie typy kybernetických útokov, ktoré sú spojené s webovými aplikáciami. Praktická časť sa bude zaoberať návrhom a implementáciou dochádzkového systému na základe zozbieraných požiadaviek od zákazníka. Pri návrhu a implementácii je kladený dôraz aj na bezpečnosť aplikácie proti známym typom kybernetických útokov.

I. TEORETICKÁ ČASŤ

1 DOCHÁDZKOVÝ SYSTÉM

Teoretická časť diplomovej práce sa zaoberá analýzou a zhodnotením vybraných riešení dochádzkových systémov, ktoré trh aktuálne ponúka. Systém bude navrhovaný pre malú spoločnosť, ktorá má aktuálne 15 zamestnancov. Spoločnosť sa zaoberá vývojom softvéru. Väčšina zamestnancov počas obdobia pandémie COVID-19 bola nútená začať pracovať z domova, takže hlavnou požiadavkou pri výbere a následnom návrhu riešenia je dostupnosť dochádzkového systému. Systém by mal byť uložený v cloude, poprípade na internom serveri, aby si mohli zamestnanci svoje odpracované hodiny zaznamenať aj pri práci z domova.

1.1 Funkcie dochádzkového systému

Aj napriek tomu, že sa môže zdať, že jedinou funkciou dochádzkového systému by malo byť zaznamenávanie príchodu a odchodu zamestnancov, nie je tomu tak. Dochádzkové systémy v dnešnej dobe začínajú byť komplexné informačné systémy, ktoré majú množstvo ďalších funkcií. Medzi štandardné funkcionality, ktoré ponúka takmer každé dostupné cloudové riešenie patrí:

- zaznamenanie príchodu a odchodu z práce,
- zobrazenie aktuálnej dochádzky zamestnancovi,
- kontrolovanie nadčasov,
- ukladanie osobných dokumentov k zamestnancom,
- export dochádzky do mzdových programov,
- zaznamenanie zdravotného voľna,
- podanie/schválenie žiadosti o dovolenku,
- vytváranie rôznych exportov a veľa iných funkcií.

2 DOSTUPNÉ RIEŠENIA

Takmer každá firma potrebuje evidovať informácie o dochádzke svojich zamestnancoch. Dopyt po týchto riešeniach je veľký a aj trh ponúka obrovské množstvo dostupných riešení. Zo všetkých dostupných riešení bolo vybraných 5 riešení, ktoré sú podrobnejšie analyzované. Boli vybrané tieto dochádzkové systémy:

- Docházka GIRITON,
- Dochádzkový systém TULIP,
- Dochádzkový systém iTA,
- Systém Aktion,
- Dochádzkový systém Fingera.

Pri výbere bol dôraz kladený na prítomnosť webového rozhrania, cez ktoré je možné vykonávať všetky operácie, ktoré s dochádzkou súvisia. Základný výber riešení prebiehal tak, aby boli vybrané riešenia, ktoré najviac spĺňajú požiadavky zákazníka.

2.1 Docházka GIRITON

Systém „Docházka GIRITON“ je dochádzkový systém implementovaný spoločnosťou GIRITON Systems s. r. o.. Spoločnosť sa zaoberá vývojom cloudových podnikových aplikácií, ale aj ich prepojením s mobilnými zariadeniami. Spoločnosť ponúka možnosť firmám vyskúšať ich dochádzkový systém zadarmo na 30 dní. Na stránke spoločnosti sa tiež nachádza jednoduchá aplikácia, ktorá pri výbere počtu zamestnancov zobrazí mesačný poplatok za používanie systému. Súhrn poplatkov sa nachádza v tabuľke 2.1. [3]

Tab. 2.1 Poplatky za dochádzkový systém Girition

Počet zamestnancov	**Cena mesačne - bez DPH**
15 zamestnancov	605 CZK
50 zamestnancov	1480 CZK
100 zamestnancov	2 730 CZK

Okrem softvérového riešenia, spoločnosť ponúka aj hardvérové riešenie, tzv. „Píchací hodiny“. Toto zariadenie je dostupné v rôznych konfiguráciách. Základný rozdiel medzi jednotlivými konfiguráciami je spôsob, akým sa užívateľ môže autentifikovať do systému. Jednotlivé varianty podporujú tieto spôsoby autentifikácie: RFID karta, NFC čip, sken tváre, sken krvného riečišťa, odtlačok prstu. [3]

Zaujímavým doplnkom, ktorý spoločnosť ponúka je bezkontaktný teplomer, ktorý je možné pripojiť k prihlasovacím hodinám. Po nakonfigurovaní si musí zamestnanec pred prihlásením najprv odmerať teplotu. V prípade vysokej teploty sa zamestnancovi neuloží záznam o príchode a napr. neodomknú dvere na turnikete. [4]

Výhody:

- intuitívne webové rozhranie
- mobilná aplikácia
- 30-dňová bezplatná verzia
- mobilná aplikácia
- GPS informácia pri vložení záznamu

Nevýhody:

- dáta uložené v cloude poskytovateľa - strata kontroly nad údajmi
- absencia integrácie s internou databázou
- nemožnosť úpravy systému

2.2 TULIP

Spoločnosť TULIP Solutions CZ s. r. o. ponúka cloudový dochádzkový systém „TULIP“. Tento systém okrem sledovania dochádzky zamestnancov a plánovania pracovných zmien ponúka aj posielanie výplatných pások na zabezpečené účty. V systéme je tiež možnosť spravovať aj služobné cesty, viesť faktúry, ale aj nastaviť si ich schvaľovací proces. Spoločnosť sa prezentuje tým, že ich systém je plne v súlade s GDPR. Riešenie však nepodporuje prihlasovanie pomocou čipov alebo biometrických prvkov. V systéme taktiež nie je možné priradiť odpracované hodiny na určitý projekt. [5]

Spoločnosť na svojich stránkach ponúka 4 rôzne typy licencií. Licencie a následne ich ceny sú rozdelené podľa počtu zamestnancov. Cena za licenciu pre jedného bežného zamestnanca je 30 CZK a 780 CZK za personalistu. V prípade, že má spoločnosť málo používateľov dochádzkového systému, je nutné platiť minimálny mesačný poplatok uvedený v tabuľke 2.2. [5]

Výhody:

- automatická kontrola legislatívy a nadčasov
- plánovanie a schvaľovanie dovolenky online

Tab. 2.2 Poplatky za dochádzkový systém TULIP

Typ licencie	**Implementácia**	**Min. mesačný poplatok - bez DPH**
Do 50 zamestnancov	13 500 CZK	1 300 CZK
50 - 250 zamestnancov	od 13 500 CZK	3 800 CZK
Nad 250 zamestnancov	podľa analýzy	3 800 CZK
S plánovaním zmien	podľa analýzy	6 450 CZK

- plánovanie pracovných zmien zamestnancov
- možnosť napojenia na mzdové programy

Nevýhody:

- nemožnosť integrácie s terminálmi
- nemožnosť úpravy prostredia na mieru
- nemožnosť priradenia hodín na projekt

2.3 iTA

Dochádzkový systém „iTA“ bol vyvinutý spoločnosťou ELEKON s. r. o.. Táto spoločnosť sa zaoberá vývojom a výrobou časomerných zariadení a pôsobí v Českej republike už od roku 1991. Ich dochádzkový systém je navrhnutý pre malé a stredné firmy. Aj tento systém je zaradený medzi cloudové systémy, takže ponúka aj webové rozhranie, pomocou ktorého je možné sledovať pracovnú dobu zamestnancov a ich nadčasy. Taktiež je možné spravovať ich dovolenky, absencie, ale aj generovať rôzne exporty a reporty, ktoré s dochádzkou súvisia. Okrem webového rozhrania je dostupná aj mobilná aplikácia na Android a iOS. [6]

Spoločnosť ponúka aj ich vlastné terminály, ktoré slúžia na zaznamenávanie dochádzky pre zamestnancov. Na obrazovke terminálu sa zamestnancovi zobrazia aj informácie o jeho dochádzke. Terminály sa pohybujú od 18 000 CZK - 23 700 CZK. Pri výbere terminálu je možné zvoliť spôsob pripojenia terminálu a spôsob prihlasovania. Terminál je možné pripojiť buď pomocou LAN, WiFi alebo LTE. Pri výbere možností prihlasovania sú dostupné dve varianty:

- čip, karta, nálepka,
- čip, karta, nálepka a snímač odtlačkov prstov. [7]

Veľmi dôležitým faktorom pre mnohé firmy môže byť fakt, že si môžu dochádzkový systém vyskúšať na rok zadarmo. V prípade, že budú chcieť softvér iTa využívať aj naďalej, cena za jednu osobu, ktorá bude systém využívať je nastavená na 25 Kč za mesiac. Pre lepšie porovnanie ceny s ostatnými systémami bola vypočítaná cena pre 15, 50 a 100 osôb. Výsledky sú zobrazené v tabuľke 2.3.

Tab. 2.3 Poplatky za dochádzkový systém iTa

Počet zamestnancov	**Cena mesačne - bez DPH**
15 zamestnancov	375 CZK
50 zamestnancov	1 250 CZK
100 zamestnancov	2 500 CZK

Zaujímavé nízko-rozpočtové riešenie dochádzky bez využívania terminálov, ktoré spoločnosť ponúka ja využívanie mobilnej aplikácie a NFC tagov. NFC tag slúži ako "terminál". Po priložení mobilného telefónu k NFC tagu, aplikácia v telefóne rozozná na akom mieste sa aktuálne zamestnanec nachádza. Zamestnanec tak priamo v mobilnej aplikácií môže zadať svoj príchod/odchod. Toto riešenie môže byť vhodné najmä pre kontrolovanie pochôdzkovej trasy zamestnancov ostrahy. [8]

2.4 Fingera

Fingera je dochádzkový a prístupový systém, ktorý sa zakladá na princípe využívania odtlačkov prstov, rozpoznávaní tváre, bezkontaktných RFID kariet alebo mobilnej aplikácie. Podniky, ktoré tento systém využívajú sa radia do skupín malej a strednej veľkosti. Toto riešenie dodáva spoločnosť Innovatrics s. r. o., ktorá sa zaoberá biometrickými technológiami. Tento systém využíva viac ako 650 podnikov. Na webových stránkach spoločnosti je možné nájsť minimálne mesačné poplatky za tento systém, ktoré sú rozdelené do 5 kategórií podľa počtu zamestnancov. Pre jednoduchšie porovnanie boli znovu vybrané ceny pre 15, 50 a 100 zamestnancov. Tabuľka 2.4 zobrazuje vypočítané výsledky. [9]

Tab. 2.4 Poplatky za dochádzkový systém Fingera

Počet zamestnancov	**Cena mesačne - bez DPH**
15 zamestnancov	420 CZK
50 zamestnancov	1 180 CZK
100 zamestnancov	na vyžiadanie

Výhody:

- prehľad prítomných osôb na pracovisku
- podpora s ekonomickými systémami
- grafické reporty
- prostredie pre správu zadaných hodín

Nevýhody:

- v dochádzkovom module je absencia schvaľovacieho procesu absencií, je nutné dokúpiť ďalší systém
- nemožnosť priraďovať odpracované hodiny k určitému projektu

Spoločnosť ako najnovší produkt ponúka „tvárový termín“. Terminál má okrem čítačky kariet a prstov zabudovaný aj teplomer a infračervenú kameru. Pomocou teplomeru je možné zmerať teplotu zamestnancov už pri vstupe do práce. Infračervená kamera slúži na overenie živosti danej osoby. Okrem spomenutého terminálu spoločnosť ponúka aj ďalšie terminály. Cenu jednotlivých terminálov však spoločnosť na svojich stránkach neuvádza. [9]

2.5 Aktion

Za dochádzkovým systémom, ktorý nesie obchodný názov Aktion stojí česká spoločnosť EFG CZ. Táto spoločnosť sa už 30 rokov zaoberá slaboprúdovými systémami a zabezpečením. Okrem toho sa spoločnosť venuje vývoju softvéru a hardvéru aj v oblasti identifikačných systémov. Ich systém Aktion využíva viac ako 3 000 spoločností na Slovensku a v Českej republike. Na stránke produktu sa nachádza aj interaktívna mapa, na ktorej sú zobrazené všetky spoločnosti, ktoré Aktion používajú. [10]

Systém Aktion je cloudovým dochádzkovým systémom. V prípade veľkých spoločností, je možné využiť ako úložisko ich vlastné servery. Pre malé spoločnosti ponúka riešenie „Online dochádzkový systém Action Cloud“. Na webovej stránke dochazkaonline.cz je možné si dochádzkový systém nakonfigurovať a vytvoriť objednávku. Keďže spoločnosť ponúka aj mobilnú verziu ich aplikácie, prípadne verziu pre tablety, ktoré môžu slúžiť ako terminál, je možné pri konfigurácií zvoliť verziu s dochádzkovým terminálom alebo bez dochádzkového terminálu. Na výber sú 2 verzie terminálov. Cena za terminál s bezkontaktným snímačom je 17 490 CZK. Drahšia verzia okrem bezkontaktného snímača obsahuje aj snímač odtlačkov prstov. Jeho cena je 20 790 CZK. Pri

výbere je nutné zvoliť počet osôb, ktoré budú dochádzkový systém využívať. Na základe tohto čísla sa odvíja mesačný/ročný poplatok. V tabuľke 2.5 sú zobrazené ceny pri vybraných počtoch zamestnancov. [10]

Tab. 2.5 Poplatky za dochádzkový systém Aktion

Počet zamestnancov	**Cena mesačne - bez DPH**
15 zamestnancov	462 CZK
50 zamestnancov	738 CZK
100 zamestnancov	1 038 CZK

Keďže sa spoločnosť zaoberá hardvérom aj softvérom, ponúkajú aj ich vlastné snímače eSmartReader, ktoré sú pripojené do siete pomocou LAN. Snímače je možné pripojiť na cloud, ale aj na vlastný server a slúžia ako dochádzkové terminály. Snímač podporuje identifikáciu pomocou karty, odtlačku prsta alebo vstupného kódu. Po prihlásení do terminálu sa na dotykovej obrazovke zobrazia prehľadné tlačidlá. Zamestnanec si pomocou nich môže zaznamenať vybranú činnosť. [10]

Výhody:

- automatická kontrola legislatívy a nadčasov
- plánovanie a schvaľovanie dovolenky online
- možnosť samo-inštalácie - žiadne vstupné poplatky
- 30-dňová skúšobná bezplatná verzia

Nevýhody:

- terminály je možné pripojiť iba pomocou LAN
- nemožnosť priraďovať odpracované hodiny k určitému projektu

2.6 Vyhodnotenie

Po preskúmaní jednotlivých vybraných systémov, zhodnotení výhod a nevýhod bolo dosiahnuté nasledujúce stanovisko. Každé vybrané riešenie má určité výhody a nevýhody vzhľadom na potreby zadávateľa. Väčšina základných požiadaviek ako je zaznamenávanie dochádzky zamestnancov a prístup z webového rozhrania zamestnancov spĺňajú všetky vybrané riešenia. Z pohľadu ceny by najlacnejšou voľbou bol systém iTa od spoločnosti ELEKON. Jednou z požiadaviek do budúcna a to umiestnenie systému na interný server zadávateľa spĺňa z vybraných riešení iba systém Aktion. Tento systém

však neponúka medzi základnými funkcionalitami možnosť priradenia hodín k určitému projektu. Väčšina spomínaných systémov ponúka množstvo funkcionalít, ktoré by boli zadávateľovi na jeho požiadavky nepotrebné a systém by mohol byť pre zamestnancov neprehľadný.

Každý z vybraných systémov uvádza, že je možné systém v určitých prípadoch upraviť na požiadavky klienta a tým pádom, by sa mohli naplniť všetky požiadavky zadávateľa. Doba, za akú by spoločnosti vedeli svoje systémy upraviť na požiadavky klienta, by mohla trvať aj niekoľko mesiacov a samozrejme, by sa radikálne zvýšili náklady na udržiavanie a rozširovanie systému. Keďže zadávateľom je spoločnosť, ktorá sa sama venuje vývoju informačných systémov, vedenie spoločnosti aj po odprezentovaní vybraných systémov rozhodlo, že najlepšou možnosťou by bolo vytvorenie vlastného systému, ktorý by spĺňal presne ich požiadavky. Je nutné dodať, že spoločnosť plánuje v najbližších dvoch rokoch rozšíriť počet zamestnancov na dvojnásobok, z čoho určite vyplynú aj nové požiadavky na systém. Keďže zamestnanci spoločnosti disponujú znalosťou technológií, v ktorom bude systém vyvíjaný, bude jednoduché systém rozšíriť o ďalšie potrebné funkcionality.

3 POŽIADAVKY ZADÁVATEĽA

Pred návrhom a implementáciou dochádzkového systému je nutné zaznamenať všetky funkcionality, ktoré musí systém spĺňať. Pri tejto fáze najdôležitejšiu úlohu zohráva vedenie, ale aj pripomienky samotných zamestnancov, ktorí budú systém využívať. Po niekoľkých stretnutiach s vedením spoločnosti boli zozbierané požiadavky, ktoré sú spísané v nasledujúcich dvoch podkapitolách. Požiadavky sú na základe unifikovaného procesu vývoja aplikácií rozdelené na funkčné a nefunkčné požiadavky. [2]

3.1 Funkčné požiadavky

Aplikácia, ktorá bude vyvinutá sa bude volať „Evidence System“. Používatelia budú mať rozdielne roly. Používateľovi bude priradená jedna z nasledovných rolí: zamestnanec, manažér, personálny pracovník alebo admin. Na základe jednotlivých používateľských rolí sú rozdelené aj funkčné požiadavky.

ZAMESTNANEC

- Zamestnanec sa prihlási pomocou emailu a hesla, ktoré si sám zvolil po otvorení aktivačného odkazu, ktorý mu prišiel na email.
- Zamestnanec má možnosť si skontrolovať svoju dochádzku za aktuálny mesiac, aj stav schválenia jednotlivých záznamov.
- Zamestnanec má možnosť zadať do aplikácie svoj odpracovaný čas.
- Pri zadávaní musí vybrať projekt a počet hodín koľko odpracoval. Taktiež zadá poznámku, na čom presne pracoval.
- Zamestnanec môže za jeden deň pracovať na niekoľkých projektoch.
- Zamestnanec môže v systéme požiadať o dovolenku, deň voľna alebo zadať sviatok.
- Zamestnanec si môže skontrolovať stav jeho žiadostí o dovolenku.

MANAŽÉR

- Manažér môže vykonávať všetky operácie, ktoré môže aj zamestnanec.
- Ak nemá manažér na svojom profile zvoleného nadriadeného, zadané odpracované záznamy a dovolenka sa mu automaticky schvália.

- Manažér si môže zobraziť a schváliť dochádzku svojich podriadených zamestnancov.
- Manažér môže vytvoriť projekty. Projekt môže byť interný alebo externý. Projekt môže mať časový rozpočet. Pri vytváraní projektu môže zadať kontakt na externú osobu a pridať ďalšie informácie do poznámky.
- Manažér si môže zobraziť odpracované časové záznamy na základe vybratého projektu.

PERSONÁLNY PRACOVNÍK

- Rola môže vytvárať účty zamestnancom.
- Pri vytváraní zamestnanca sa vyberie jeho rola, druh úväzku a jeho nadriadený. Nadriadeného je nutné zvoliť iba pri roli „Zamestnanec“.
- Pracovník môže generovať export dochádzky do súboru na základe vopred vybratých nastavení.

3.2 Nefunkčné požiadavky

- Dochádzkový systém bude implementovaný ako webová aplikácia s využitím frameworku .NET Core, ktorý bude počas prevádzky dostupný na internom serveri spoločnosti.
- Aplikácia bude využívať relačný databázový systém Microsoft SQL Server.
- Databázový systém bude vytvorený na cloudovej platforme Azure od spoločnosti Microsoft.
- Žiadne dáta aplikácia z databázy nevymazáva.
- Doba načítania stránky nesmie presiahnuť 3 sekundy.
- Webová aplikácia bude Single-Page aplikácia, ktorá bude komunikovať s dátovou vrstvou pomocou API požiadaviek.
- Webová aplikácia bude optimalizovaná pre Full-HD rozlíšenie.
- Celá aplikácia bude v anglickom jazyku.

4 LEGISLATÍVNE POŽIADAVKY

Zavádzať dochádzkový systém a evidovať odpracovanú dobu zamestnancov nie je iba na rozhodnutí zamestnávateľa. Každý zamestnávateľ v Českej republike sa musí riadiť zákonom 262/2006 Sb. tzv. zákonníkom práce. Tento zákonník ukladá povinnosti a práva nielen zamestnancom, ale hlavne zamestnávateľom. Evidenciou dochádzky sa zaoberá zákon 262/2006 Sb. §96. Jeho znenie je nasledovné:

1. „Zamestnávateľ je povinný viesť pri jednotlivých zamestnancoch evidenciu s vyznačením začiatku a konca
   (a) odpracovanej
      i. zmeny [§ 78 odst. 1 písm. c],
      ii. práce nadčas [§ 78 odst. 1 písm. i) a § 93],
      iii. doby v dobe pracovnej pohotovosti (§ 95 odst. 2),
   (b) pracovnej pohotovosti, ktorú zamestnanec držal [§ 78 odst. 1 písm. h) a § 95].
2. Na žiadosť zamestnanca je zamestnávateľ povinný umožniť zamestnancovi nahliadnuť do jeho účtu pracovnej doby alebo evidencie pracovnej doby, a do jeho účtu mzdy a vytvárať si z nich výpisy, prípadne rovnopis na náklady zamestnávateľa.“ [1]

Na základe tohto zákona, zamestnávateľ musí evidovať pri každom zamestnancovi odpracovanú dobu, nadčasy a dobu v pracovnej pohotovosti. Spoločnosť, pre ktorú bude dochádzkový systém navrhnutý, nefunguje na princípe pracovných zmien, ale zamestnancom umožňuje voľnú pracovnú dobu s podmienkou, že zamestnanec by mal mať na konci mesiaca odpracovaných 8 hodín v priemere na každý pracovný deň. Na základe tejto informácie, budú hodiny, ktoré odpracuje zamestnanec navyše, zvýraznené pri exporte na konci mesiaca.

5 HROZBY WEBOVÝCH APLIKÁCIÍ

Množstvo hrozieb a typov kybernetických útokov rastie v dnešnej dobe enormnou rýchlosťou. Keďže aj v evidenčných systémoch sú ukladané osobné údaje zamestnancov, je nutné sa pozrieť aj na možné útoky na webové aplikácie. Kedže druhov útokov na webové aplikácie je obrovské množstvo, nie je možné obsiahnuť princíp všetkých. Preto boli vybrané iba niektoré z najznámejších útokov. Princíp fungovania je opísaný v nasledujúcich podkapitolách.

5.1 Cross-Site scripting útoky - XSS

Cross-Site scripting je typom útoku, ktorý funguje na princípe spustenia škodlivého Javascript kódu. Pri útoku je možné využiť napríklad neošetrené formulárové polia, do ktorých je možné vložiť HTML kód. Tento kód sa napríklad uloží do databázy a následne sa na základe HTML kódu môže vykresliť webové stránka. Tento spôsob ukladania HTML kódu býva často využívaný v redakčných webových systémoch, kde prispievatelia môžu vkladať vlastný HTML kód. [11]

5.1.1 Spôsob vykonávania útoku

Spôsobov ako vykonať XSS útok je niekoľko. Najprimitívnejší spôsob je vloženie škodlivého javascript kódu do neošetreného formulára, ktorý je následne renderovaný cieľovým používateľom, napr.:

```
Nezabezpečený <script>alert("XSS útok") web.</script>
```

Týmto spôsobom môžeme u používateľov aplikácie spustiť škodlivý kód aj bez vedomia administrátora alebo programátora danej aplikácie. Samozrejme, názorný príklad by neznamenal pre používateľa žiadne veľké nebezpečenstvo, ale ukazuje princíp ako XSS útoky fungujú. Skripty je možné vytvoriť oveľa sofistikovanejšie a môžu spôsobiť oveľa väčšiu škodu. Pomocou vloženého Javascriptu môže útočník ukradnúť používateľovu aktuálnu reláciu a tým pádom môže vykonávať v aplikácii všetky akcie ako obeť a dostať sa k citlivým údajom. V prípade, že je útok dobre prepracovaný, obeť si vôbec nemusí všimnúť, že bola napadnutá. [11]

5.2 SQL Injection útoky

SQL injection je typ webového útoku, ktorý funguje na princípe podstrčenia, resp. vloženia škodlivého kódu do aplikácie, ktorého cieľom je napadnúť databázovú vrstvu aplikácie. Ak sa to útočníkovi podarí, je schopný meniť pomocou tohto kódu logiku SQL príkazov, ktoré sa spúšťajú voči databáze. Môže zmazať, upraviť záznamy alebo tabuľky v databáze, prípadne získať citlivé dáta, ku ktorým by nemal mať prístup. [13]

5.2.1 Spôsob vykonávania útoku

Štandardným vstupným bodom, ktoré SQL Injection útoky využívajú sú formuláre na webových stránkach. V prípade, že obsah vložený do formulárového poľa nie je kontrolovaný na zakázané znaky a z obsahu sa priamo vytvára SQL príkaz, môže dôjsť k tomu, že útočník vloží svoju časť SQL príkazu, ktorá sa spustí nad cieľovou databázou. [13]

Príklad nesprávneho vytvárania SQL príkazov v kóde aplikácie:

```
prikaz = "SELECT * FROM uzivatelia WHERE email = '" + zadanyEmail + "';"
```

V prípade, že používateľ zadá do poľa email napr.

`"email';DROP TABLE uzivatelia;"` alebo `"email' or 'b'='b"`

vznikne z pôvodného príkazu na vybratie hodnôt vybraného používateľa upravený výraz, ktorý v 1. prípade vymaže tabuľku užívatelia a v druhom prípade vyhodnotí vo všetkých prípadoch podmienku v príkaze ako pravdivú, takže útočník získa všetky záznamy z tabuľky `uzivatelia`.

```
"SELECT * FROM uzivatelia WHERE email = 'email' or 'b'='b';"
```

```
"SELECT * FROM uzivatelia WHERE email = 'email';
          DROP TABLE uzivatelia; --';"
```

Samozrejme, podobných spôsobov je oveľa viac. Útočník môže využiť aj príkazy JOIN alebo UNION, ktoré mu dovolia operovať a spúšťať príkazy aj nad ostatnými tabuľkami v databáze.

5.3 Cross-Site Request Forgery útoky - XSRF / CSRF

CSRF je spôsob útoku na webové stránky, ktorej základ je prinútenie používateľa otvoriť stránku napadnutej aplikácie, ktorá vykonáva akciu, o ktorej používateľ nevie. Cieľom Cross-Site Request Forgery útokov je donútenie používateľa vykonať nejaké akcie bez toho, aby o tom vedel. Pri tomto útoku je nutné, aby útočník napadnutú stránku dobre poznal a aby používateľ otvoril infikovanú stránku, alebo klikol na napadnutý odkaz. [12]

5.3.1 Spôsob vykonávania útoku

Je situácia, že používateľ je prihlásený v aplikácií, ktorá je nezabezpečená a na zmenu osobných údajov u daného používateľa využíva GET požiadavky. Útočníkovi sa podarí donútiť obeť kliknúť na nasledovnú stránku:

```
https://nezabezpeceny-web.cz/email/zmen=?utocnikov@email.cz
```

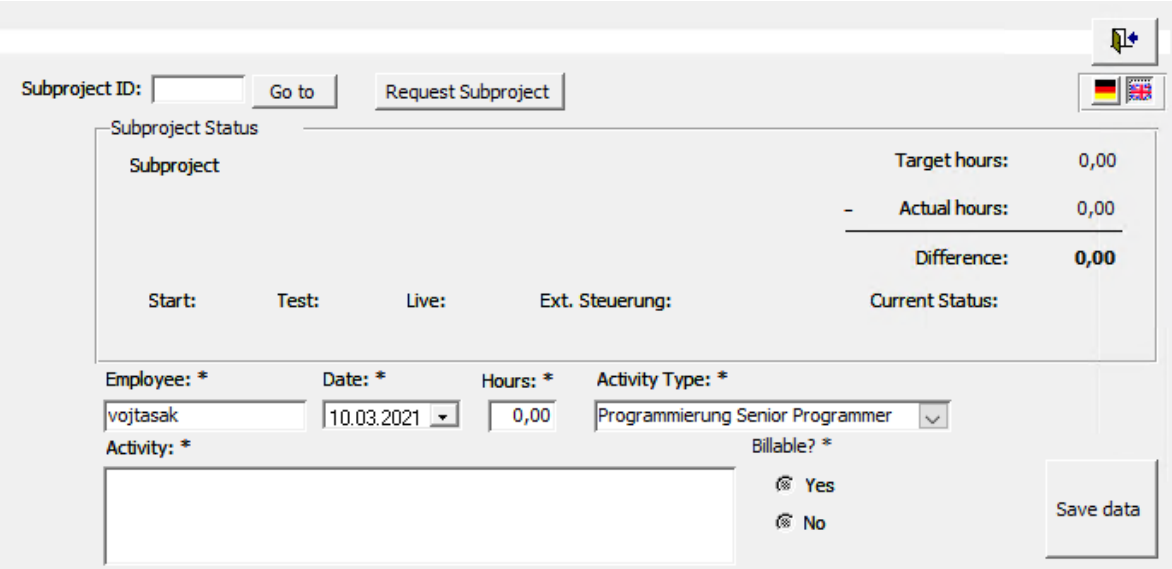
V prípade, že používateľ je autentizovaný napríklad pomocou cookies, požiadavka zaslaná do aplikácie vyzerá ako validná a útočníkovi sa podarilo zmeniť email na svoj.

6 AKTUÁLNY STAV

Pred návrhom dochádzkového systému je nutné zistiť ako spoločnosť aktuálne eviduje odpracovaný čas a na základe toho navrhnúť systém, ktorý tento proces zjednoduší a urobí prehľadnejším.

Spoločnosť má aktuálne 15 zamestnancov a sídli v Brne. Z toho je 10 vývojárov, 2 administrátori, 1 personálna zamestnankyňa, projektový manažér a vedúci pobočky. Spoločnosť je dcérskou firmou nemeckej firmy.

Spoločnosť pracuje na niekoľkých projektoch. Vývojári aktuálne zaznamenávajú odpracovaný čas do jednoduchej aplikácie, ktorá je vytvorená v MS Access (Obr. 6.1). Aplikácia je uložená na externom serveri materskej spoločnosti. V aplikácií si zamestnanci vyberú projekt na ktorom pracovali, dátum, počet hodín ktoré odpracovali a v poznámke bližšie špecifikujú na čom pracovali. Tento zaznamenaný čas musí ručne kontrolovať projektový manažér. Aplikácia bola primárne určená nemeckým zamestnancom, takže je primárne v nemčine a len niektoré texty sú preložené.



Obr. 6.1 Aktuálna dochádzková aplikácia

V prípade, že chce zamestnanec požiadať o dovolenku alebo o zdravotné voľno, musí prísť za manažérom, prípadne mu napísať cez aplikáciu Skype. Ten mu ju následne schváli alebo neschváli. Pretože k aplikácií je prístup iba cez pripojenie na vzdialený počítač a zapínanie aplikácie je pomerne časovo zdĺhavé, manažér si zaznamenáva dovolenky zamestnancov do excelovej tabuľky. Zamestnanec v dochádzkovej aplikácií nemá právo zapisovať dovolenku do aplikácie. Na konci mesiaca následne manažér zapisuje dovolenky a využité voľná do spomínanej aplikácie všetkých zamestnancov na základe excelovej tabuľky, ktorú si vytvoril. Tabuľku následne musí ručne upraviť a poslať mzdovej účtovníčke, ktorá na základe toho spracuje mzdy pre zamestnancov.

Spoločnosť sa začína rozrastať o ďalších zamestnancov a vedeniu spoločnosti prestáva používaný systém a proces vyhovovať. Aktuálny systém má niekoľko nevýhod a problémov, ktoré sa veľmi často vyskytujú. Medzi hlavné problémy patria:

- aplikácia je primárne určená pre nemeckú spoločnosť, takže česká pobočka má malý vplyv na možnosť úpravy aplikácie, ale aj používateľských práv pre zamestnancov,
- manažér si dovolenky zapisuje ručne a vyskytli sa prípady, že zamestnancovi dovolenku schválil, ale zabudol si ju zapísať do jeho tabuľky,
- aplikácia je pomalá a samotný prístup k nej je veľmi zdĺhavý,
- veľká časť používateľského prostredia a hlášok je v nemeckom jazyku,
- obmedzená možnosť exportu údajov.

7 POUŽITÉ TECHNOLÓGIE

Pri výbere technológií bolo nutné do diskusie zapojiť aj spoločnosť, ktorá bude dochádzkový systém v budúcnosti využívať. Keďže hlavnými dôvodmi vytvorenia vlastného dochádzkového systému bola aj kontrola nad kódom a možnosť úprav, prípadne ďalších rozšírení systému na mieru, je nutné zvoliť také technológie, ktorými spoločnosť a jej zamestnanci disponujú a ovládajú. Spoločnosť svoje produkty vyvíja v jazyku C# na platforme .NET Core, preto je podľa skúseností vedenia, ale aj vývojárov vhodná technológia aj na vývoj dochádzkového systému.

Cieľom práce je vytvorenie jednostránkovej aplikácie, ktorá bude komunikovať s dátovou vrstvou pomocou REST API požiadaviek. Preto je nutné vybrať aj technológiu, ktorá bude použitá na používateľské rozhranie. Výber tejto technológie bol na autorovi práce. Autor ako technológiu na prezentačnú vrstvu zvolil knižnicu React. Táto Javascriptová knižnica bola vybraná z dôvodu, že autor práce s ňou nemal žiadne skúsenosti a videl to ako spôsob ako sa ju naučiť a tým rozšíriť svoje znalosti. Výber frameworku bol opäť konzultovaný a odsúhlasený zadávateľom.

7.1 ASP.NET Core

ASP.NET Core je multiplatformový, výkonný, open-source framework od spoločnosti Microsoft, ktorý slúži na vývoj moderných aplikácií. Podporuje vývoj cloudových aplikácií, ktoré sú pripojené cez internet. Na rozdiel od staršej verzie .NET, už nie je priamo spojený s operačným systémom Windows, ale je možné tento framework využívať na Linuxe a macOS. Framework podporuje:

- vývoj webových aplikáciií a webových služieb,
- vývoj IoT aplikácií,
- vývoj aplikačnej vrstvy mobilných aplikácií. [14]

.NET je založený na objektovo orientovanom programovaní (OOP). OOP je vývojový model, ktorý slúži na rozdelenie softvéru a kódu na menšie kúsky, ktoré je jednoduchšie spravovať a kombinovať. [15]

Výhodou je aj množstvo integrovaných nástrojov a knižníc, ktoré pomáhajú vývojárom zabezpečiť aplikácie. Ponúka zabudované nástroje na správu autentifikácie, autorizácie, ochrany dát, vynútenie HTTPS, aplikačných tajomstiev, prevenciu pred XSRF/CSRF a CORS manažment. Tieto nástroje pomáhajú vývojárom pri vývoji robustných a bezpečných ASP.NET Core aplikácií. [16]

7.2 React

React je Javascriptová knižnica, ktorá slúži na vytváranie používateľských rozhraní. S použitím Reactu je možné vytvoriť komplexné jednostránkové aplikácie. Základnou myšlienkou je vytváranie jednoduchých komponentov, ktoré majú svoj stav. V prípade, že sa stav zmení, tak sa obnoví iba daná časť a nie celá stránka. [17]

7.3 Swagger

Swagger je voľne dostupný framework určený na návrh, tvorbu, dokumentáciu REST API. Obsahuje nástroje, pomocou ktorých je veľmi jednoduché vytvoriť dokumentáciu k implementovaným API a je možné jednotlivé API testovať aj bez implementovaného používateľského rozhrania. [18]

7.4 Material-UI

Material-UI je knižnica, ktorá obsahuje znovupoužiteľné React komponenty, ktoré je možné jednoducho použiť a prípadne upraviť. Táto knižnica pomáha k rýchlemu a jednoduchšiemu vývoju webových aplikácií. [19]

7.5 Azure

Azure patrí medzi produkty Microsoft. Je to cloudová platfroma, ktorá ponúka viac ako 200 rôznych produktov a služieb. Medzi služby, ktoré boli využité aj v tejto diplomovej práci patria napríklad SQL servery, webové úložiská a ich správa. Azure však ponúka množstvo iných služieb, ktoré sú postavené na báze cloudu. [20] [21]

II. PRAKTICKÁ ČASŤ

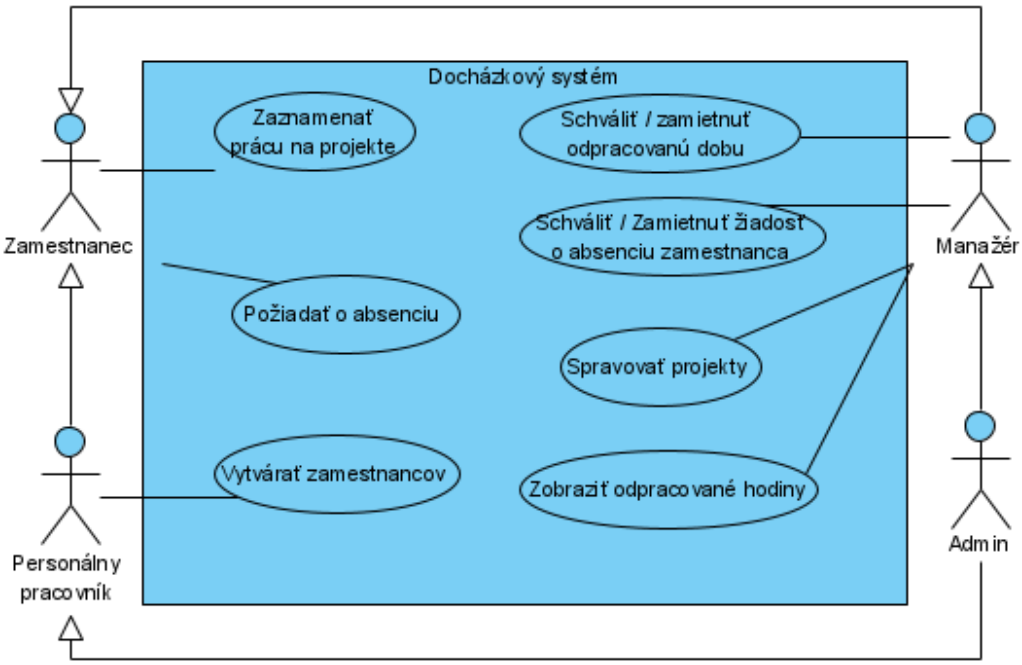
8 NÁVRH SYSTÉMU

Systém je navrhnutý na základe funkčných a nefunkčných požiadaviek zadávateľa. Riadi sa unifikovaným vývojom aplikácií.

Na základe funkčných požiadaviek bude vytvorený diagram prípadov použitia ku jednotlivým rolám, ktoré budú systém využívať. Následne budú spísané jednotlivé úspešné a alternatívne scenáre, ktoré v systéme môžu nastať. Aplikácia bude využívať model Code First, je nutné preto navrhnúť jednotlivé triedy, ktoré budú predstavovať a uchovávať informácie o jednotlivých častiach systému. Za pomoci týchto tried sa vygeneruje databáza, ktorá bude jednotlivé informácie ukladať.

8.1 Prípady použitia a aktéri

Diagram prípadov použitia opisuje, ako vidia systém používatelia a čo môžu v systéme vykonávať. V diagrame sú zobrazené možnosti použitia, ktoré môže používateľ v systéme vykonať. Diagram nerieši ako jednotlivé súčasti budú naimplementované. Jeho hlavnou úlohou je opísať ako má systém fungovať a znázorniť jeho funkcionalitu. Tento diagram býva jedným z prvých diagramov, ktorý pri vývoji softvéru býva vytvorený. Na základe diagramu sa architekti systému a zadávateľ dokážu zhodnúť či naozaj navrhovaný systém má robiť to, čo sa očakáva. Nasledujúci obrázok 8.1 zobrazuje základné funkcionality priradené k jednotlivým aktérom. Kompletný diagram je v prílohe.



Obr. 8.1 Zjednodušný diagram prípadov použitia

8.1.1 Aktéri

Na základe diagramu prípadov použitia (obr. 8.1) je vidieť, že so systémom budú pracovať 4 rôzne typy používateľov. Každý z týchto používateľov bude mať aj iné práva a povolené akcie v navrhovanom systéme. Systém bude rozlišovať týchto aktérov:

- Admin,
- Manažér,
- Personálny pracovník,
- Zamestnanec.

8.2 Scenáre

Základné prípady použitia, ktoré boli navrhnuté v predchádzajúcej kapitole budú v tejto kapitole rozpracované na úspešné a alternatívne scenáre.

Prípad použitia: Vytvorenie prístupu do systému

Aktéri: Manažér, Personálny pracovník, Admin

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér klikne na položku Zamestnanci.	
2.		Systém zobrazí obrazovku s listom zamestnancov a tlačidlom na pridanie nového zamestnanca.
3.	Aktér zvolí možnosť Pridať nového zamestnanca.	
4.		Systém zobrazí formulár, v ktorom je nutné vyplniť informácie o zamestnancovi.
5.	Aktér vyplní informácie o zamestnancovi a potvrdí formulár.	
6.		Systém pošle správu s odkazom na aktivovanie účtu na zadaný email.
7.		Systém zobrazí notifikáciu o úspešnom pridaní zamestnanca a pridá zamestnanca do zoznamu.

Tab. 8.1 Vytvorenie zamestnaneckého prístupu do systému

Prípad použitia: Aktivovanie účtu

Aktéri: Zamestnanec

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér otvorí doručený odkaz na emailovú adresu do 24 hodín od doručenia.	
2.		Systém zobrazí formulár na zadanie emailu a vytvorenie nového hesla.
3.	Aktér vyplní svoj email, zvolí si heslo, ktoré bude spĺňať všetky podmienky bezpečného hesla a heslo ešte raz potvrdí.	
4.		Systém zobrazí aktérovi, že došlo k úspešnému resetovaniu hesla.
5.	Aktér zvolí možnosť prihlásiť sa.	
6.		Systém zobrazí prihlasovací formulár.
7.	Aktér zadá svoj email a zvolené heslo.	
8.		Systém aktéra prihlási do systému a zobrazí sa mu úvodná stránka.

Tab. 8.2 Aktivovanie účtu zamestnancom

Alternatívny scenár:

	Aktér	**Systém**
1.1.	Aktér otvorí odkaz po vypršaní platnosti.	
2.1.		Systém zobrazí formulár na zadanie emailu a vytvorenie nového hesla.
3.3.	Aktér vyplní svoj email, zvolí si heslo a heslo ešte raz potvrdí.	
4.1.		Systém zobrazí aktérovi, že resetovanie hesla nebolo úspešné.

Tab. 8.3 Alternatívny scenár č.1 Aktivovanie účtu

Prípad použitia: Zaznamenať prácu na projekte

Aktéri: Zamestnanec

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér v menu vyberie položku Časové záznamy	
2.		Systém aktérovi zobrazí jeho časové záznamy a možnosť pridať nový časový záznam.
3.	Aktér zvolí možnosť pridanie nového časového záznamu.	
4.		Systém zobrazí aktérovi formulár, kde aktér musí zvoliť dátum, projekt na ktorom pracoval a musí pridať poznámku, kde bližšie špecifikuje na čom pracoval.
5.	Aktér vyplní všetky potrebné údaje a formulár potvrdí.	
6.		Systém záznam uloží so statusom záznamu čakajúceho na schválenie a aktorovi zobrazí notifikáciu o úspešnom vytvorení záznamu.

Tab. 8.4 Zaznamenanie práce na projekte

Alternatívny scenár:

	Aktér	**Systém**
5.1.	Aktér nevyplní niektorý z povinných údajov.	
6.1.		Systém aktéra upozorní červeným textom, ktorý údaj nevyplnil.
7.1.	Aktér má možnosť daný údaj doplniť a scenár pokračuje 6. bodom úspešného scenára.	

Tab. 8.5 Alternatívny scenár č.1 Zaznamenanie práce na projekte

Prípad použitia: Vytvorenie požiadavky na neprítomnosť

Aktéri: Zamestnanec

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér v menu vyberie položku Neprítomnosti	
2.		Systém aktérovi zobrazí jeho záznamy o neprítomnosti a možnosť pridať novú žiadosť.
3.	Aktér zvolí možnosť vytvorenie novej žiadosti.	
4.		Systém zobrazí aktérovi formulár, kde aktér musí zvoliť typ neprítomnosti, dátum, počet hodín a môže pridať poznámku
5.	Aktér vyplní všetky potrebné údaje a formulár potvrdí.	
6.		Systém záznam uloží so statusom žiadosti čakajúcej na schválenie a aktérovi zobrazí notifikáciu o úspešnom vytvorení žiadosti.
7.	Aktér má možnosť si skontrolovať stav žiadosti.	

Tab. 8.6 Vytvorenie žiadosti na neprítomnosť

Alternatívny scenár:

	Aktér	**Systém**
5.1.	Aktér nevyplní niektorý z povinných údajov.	
6.1.		Systém aktéra upozorní červeným textom, ktorý údaj nevyplnil.
7.1.	Aktér má možnosť daný údaj doplniť a scenár pokračuje 6. bodom úspešného scenáru.	

Tab. 8.7 Alternatívny scenár Vytvorenie žiadosti na neprítomnosť

Prípad použitia: Schváliť / zamietnuť odpracovanú dobu

Aktéri: Manažér, Admin

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér zvolí možnosť Schváliť odpracované hodiny.	
2.		Systém zobrazí aktérovi všetky záznamy, ktoré čakajú na schválenie.
3.	Aktér označí záznamy, ktoré chce schváliť a vyberie možnosť schváliť záznamy.	
4.		Systém zobrazí aktérovi potvrdzovacie okno, či chce naozaj záznamy schváliť.
5.	Aktér potvrdí svoj príkaz.	
6.		Systém schváli dané záznamy a schválené záznamy sa zmažú z listu záznamov čakajúcich na schválenie.
7.	Aktér označí záznamy, ktoré chce zamietnuť a vyberie možnosť zamietnuť dané záznamy.	
8.		Systém zobrazí aktérovi potvrdzovacie okno, či chce naozaj záznamy zamietnuť.
9.	Aktér potvrdí svoj príkaz.	
10.		Systém zamietne dané záznamy a zmaže záznamy z listu záznamov čakajúcich na schválenie.

Tab. 8.8 Schválenie / zamietnutie časových záznamov

Alternatívny scenár:

	Aktér	**Systém**
9.1.	Aktér nepotvrdí svoj príkaz.	
10.1.		Systém vráti aktéra na obrazovku s nepotvrdenými záznamami a scenár pokračuje podľa bodu 7.

Tab. 8.9 Alternatívny scenár Zamietnutie časového záznamu

Prípad použitia: Schváliť / zamietnuť žiadosť o neprítomnosť

Aktéri: Manažér, Admin

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér zvolí možnosť Schváliť odpracované hodiny.	
2.		Systém zobrazí aktérovi všetky záznamy, ktoré čakajú na schválenie.
3.	Aktér označí záznamy, ktoré chce schváliť a vyberie možnosť schváliť záznamy.	
4.		Systém zobrazí aktérovi potvrdzovacie okno, či chce naozaj záznamy schváliť.
5.	Aktér potvrdí svoj príkaz.	
6.		Systém schváli dané záznamy a schválené záznamy sa zmažú z listu záznamov čakajúcich na schválenie.
7.	Aktér označí záznamy, ktoré chce zamietnuť a vyberie možnosť zamietnuť dané záznamy.	
8.		Systém zobrazí aktérovi potvrdzovacie okno, či chce naozaj záznamy zamietnuť.
9.	Aktér potvrdí svoj príkaz.	
10.		Systém zamietne dané záznamy a zmaže záznamy z listu záznamov čakajúcich na schválenie.

Tab. 8.10 Schválenie / zamietnutie žiadosti o neprítomnosť

Alternatívny scenár:

	Aktér	**Systém**
3.1.	Aktér neoznačí žiadne záznamy a vyberie možnosť schváliť záznamy.	
4.1.		Žiadne záznamy nebudú schválené a scenár pokračuje podľa bodu 3.

Tab. 8.11 Alternatívny scenár Schválenie žiadosti o neprítomnosť.

Prípad použitia: Zobraziť a exportovať odpracované hodiny

Aktéri: Manažér, Admin

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér vyberie možnosť Schválené odpracované hodiny.	
2.		Systém zobrazí schválené odpracované záznamy všetkých zamestnancov zoskupené na základe projektu.
3.	Aktér zvolí filtračné nastavenia pomocou dostupných nástrojov.	
4.		Systém zobrazí záznamy na základe nastavených filtrov.
5.	Aktér zvolí možnosť Export.	
6.		Systém vytvorí CSV súbor.
7.	Aktér si otvorí stiahnutý súbor.	

Tab. 8.12 Zobrazenie a exportovanie odpracovaných hodín

Prípad použitia: Vytvoriť projekt

Aktéri: Manažér, Admin

Úspešný scenár:

	Aktér	**Systém**
1.	Aktér zvolí možnosť Projekty.	
2.		Systém zobrazí vytvorené projekty a možnosť vytvoriť nový projekt.
3.	Aktér zvolí možnosť Vytvoriť nový projekt.	
4.		Systém zobrazí formulár na vytvorenie nového projektu.
5.	Aktér vyplní názov projektu, pridá poznámku, označí či je projekt interný a môže vyplniť plánovaný rozpočet a formulár potvrdí.	
6.		Systém údaje zvaliduje, vytvorí nový projekt a aktérovi zobrazí hlášku o úspešnom vytvorení.

Tab. 8.13 Vytvorenie nového projektu

8.3 Návrh tried a databázy

Pri vývoji bude na aplikačnej vrstve využitý ORM framework - Entity Framework a postup Code First - Najskôr Kód. Tento framework automaticky mapuje tabuľky v databáze na objekty a tým umožňuje pracovať s tabuľkami ako s objektmi.

Pri postupe Code First vývojár začne najskôr vytvárať triedy v C#, s ktorými bude v aplikácií pracovať a budú predstavovať objekty. Po vytvorení jednotlivých tried pomocou Entity Frameworku sú z tried vygenerované tabuľky v SQL databáze. Generovanie SQL tabuliek a ich štruktúru je možné upraviť pomocou:

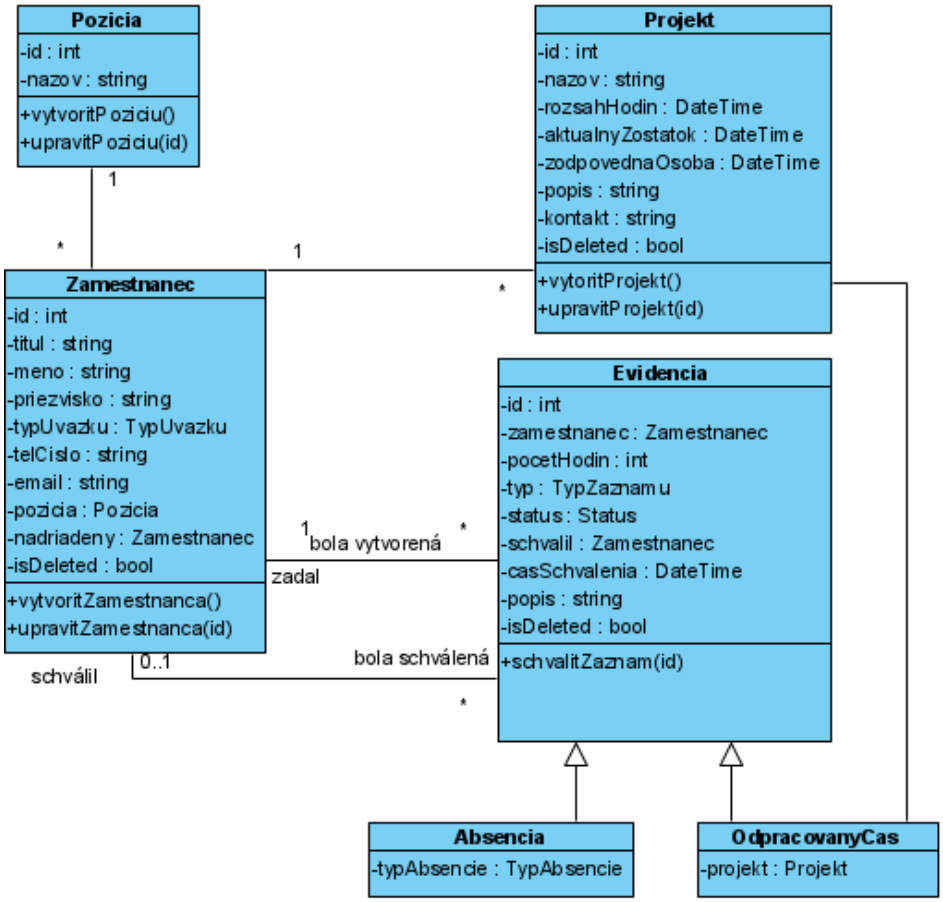
- Dátovej anotácie atribútov,
- Fluent API. [22]

Tieto spôsoby konfigurácie budú využité aj pri implementácií.

Na základe funkčných požiadaviek boli navrhnuté nasledovné objekty, ktoré budú predstavovať triedy, s ktorými bude pracovať aplikačná vrstva systému:

- **Pozícia** - trieda predstavuje informácie o pracovnej pozícii, na ktorej zamestnanec pracuje,
- **TypUvazku** - trieda predstavuje informácie o pracovnom úväzku zamestnanca,
- **ZamestnaneckyStatus** - trieda predstavuje informácie o stave zamestnanca,
- **Zamestnanec** - trieda uchováva všetky informácie o zamestnancovi,
- **Evidencia** - táto trieda predstavuje nadradenú triedu pre záznamy, ktoré zamestnanci môžu do systému zadať,
- **Absencia** - zdedená trieda z triedy „Evidencia“, predstavuje absenčné záznamy,
- **OdpracovanyCas** - zdedená trieda z triedy „Evidencia“, predstavuje záznamy, ktoré zamestnanci odpracovali,
- **EvidenciaStatus** - predstavuje schvaľovací status, v akom sa aktuálne nachádzajú záznamy,
- **Projekt** - trieda predstavuje informácie o projekte,
- **ProjektStatus** - trieda predstavuje status projektu.

Zjednodušený diagram tried, vzťahy a násobnosti medzi jednotlivými triedami je možné vidieť na obrázku 8.2. Na obrázku je možné vidieť aj atribúty jednotlivých tried.



Obr. 8.2 Diagram tried

Z vygenerovanej databázy bol vytvorený aj ERD diagram, ktorý je možné nájsť v prílohách. ERD diagram znázorňuje tabuľky v SQL databáze, primárne kľúče, cudzie kľúče a vzťahy medzi vytvorenými tabuľkami. Okrem nami navrhnutých tried boli vytvorené aj ďalšie tabuľky, ktoré sú súčasťou nástroja ASP.NET Core Identity.

9 IMPLEMENTÁCIA PROTOTYPU

Implementácia prototypu evidenčného systému prebiehala vo vývojových prostrediach od spoločnosti Microsoft. Systém bol implementovaný ako Single-Page aplikácia, využívajúca REST API. Jednotlivé vrstvy aplikácie boli preto vyvíjané oddelene. Aplikačná časť bola vyvíjaná vo vývojovom prostredí Microsoft Visual Studio Professional 2019. Prezentačná časť systému bola vyvíjaná v editore Visual Studio Code. Vývoj prezentačnej aj aplikačnej časti prebiehal súčasne.

9.1 Aplikačná a dátová vrstva

Ako bolo už spomenuté v kapitole Použité technológie, systém bol vyvíjaný v jazyku C# na platforme .NET 5.

Projekt bol vytvorený v prostredí Visual Studio. Pri vytváraní projektu je možnosť si vybrať z niekoľkých šablón, ktoré vývojárovi ušetria množstvo času so základným nastavením. V prípade tohto systému bola vybratá šablóna „ASP.NET Core with React.js“. Po zvolení ostatných nastavení vývojové prostredie vygeneruje projekt so základnou štruktúrou súborov a vzorovými stránkami. Tieto vzorové stránky boli vymazané.

9.1.1 Nainštalované balíčky

Počas implementácie bolo nainštalovaných a pridaných niekoľko knižníc, ktoré aplikácia využíva. Knižnice boli nainštalované pomocou nástroju NuGet. Tento nástroj slúži na správu všetkých dostupných a nainštalovaných knižníc. Boli nainštalované tieto doplnkové knižnice:

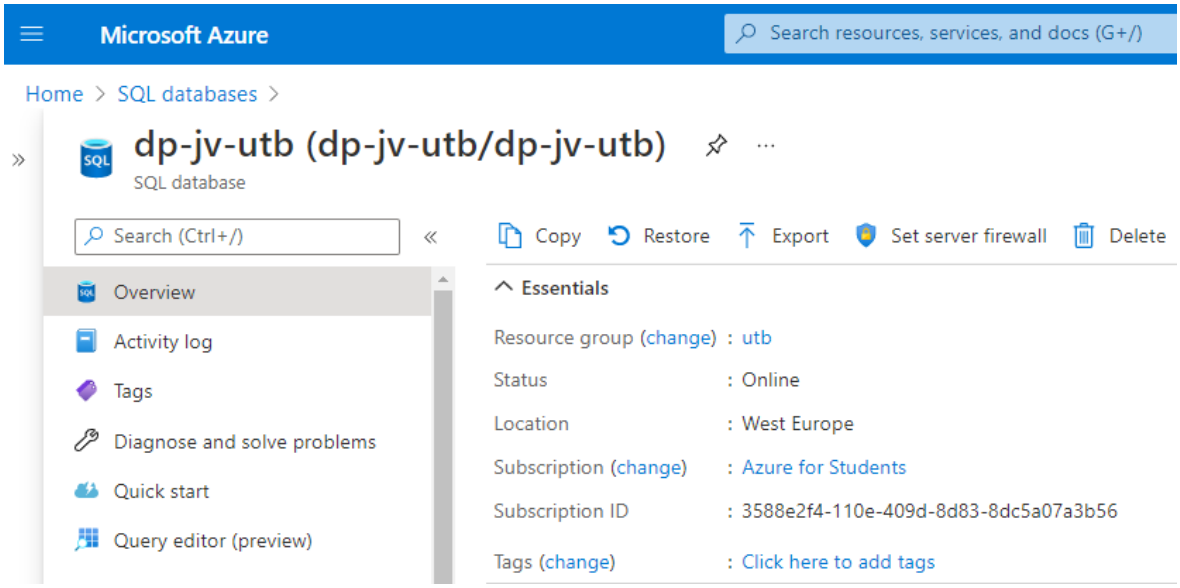
- **Azure.Extensions.AspNetCore.Configuration.Secrets** - integrácia s Azure Key Vault,
- **Microsoft.ApplicationInsights.AspNetCore** - integrácia s Application Insight, ktoré slúži na monitorovanie aplikácie po nasadení na server,
- **Microsoft.AspNetCore.ApiAuthorization.IdentityServer** - balíček je využitý pri implementácií autorizácie a autentifikácie do systému,
- **EntityFrameworkCore** - obsahuje niekoľko knižníc, ktoré zabezpečujú manipuláciu s dátami a mapovanie tried na tabuľky v SQL databáze,
- **SendGrid** - knižnica, ktorá pomáha s integráciou so systémom SendGrid, ktorý zabezpečuje odosielanie emailov,

- **Swashbuckle.AspNetCore** - nástroje na jednoduché vytvorenie API dokumentácie.

9.1.2 Vytvorenie databázového serveru

Aplikácia využíva Entity Framework Core, ktorý dokáže mapovať tabuľky na triedy v C#. Tieto tabuľky dokáže tento framework aj automaticky vygenerovať, takže nie je nutné ich vytvárať ručne.

Pre správu a ukladanie dát bol vytvorený a nakonfigurovaný SQL server na platforme Azure (Obr. 9.1). Pre vývoj prototypu bola využitá študentská licencia. Po nakonfigurovaní bol v nastaveniach aplikácie zmenený pripájací reťazec do vytvoreného SQL serveru.



Obr. 9.1 SQL Server v Azure

9.1.3 Vytvorenie tried

Na základe diagramu tried, ktorý bol vytvorený v časti Návrh systému boli v projekte naimplementované triedy. Pomocou príkazov `Add-Migration` a `Update-Database` spustených v konzole Package Manger, boli na základe tried vygenerované SQL tabuľky vo vytvorenom SQL serveri.

9.1.4 Autentifikácia a autorizácia

Pri implementácií autentifikácie a autorizačnej logiky bol použitý middleware IdentityServer. Každému požívateľovi pri prihlasovaní je pomocou implementovanej logiky v metóde `GetProfileDataAsync`, ktorá sa nachádza v triede `ProfileService.cs` pridelená informácia o jeho roli, priradením listu tried `Claim`. Trieda implementuje roz-

hranie `IProfileService`. Informácie o používateľovi a jeho roli sú tak súčasťou aj `HttpContext`, ale aj JWT tokenu, ktorý je po prihlásení uložený v lokálnej pamäti prehliadača.

Pri poslaní požiadavky na API server, je kontrolované či používateľ, ktorý požiadavku poslal je autorizovaný na vykonanie danej požiadavky. Každá API metóda, ktorú je možné zavolať, má pomocou atribútu `Authorize` určené, ktoré systémové roly môžu danú požiadavku zavolať (Obr. 9.2). V prípade, že identita, ktorá zavolala API metódu, na ktorú nemá povolenie, server odošle stavový kód 403, čo znamená, že odosielateľovi je zakázaný prístup k danej API metóde.

```
[HttpPost]
[Authorize(Roles = Roles.Admin + "," + Roles.Manager)]
0 references | Jakub Vojtasak, 5 days ago | 1 author, 3 changes | 0 requests | 0 exceptions
public async Task<ActionResult<Project>> PostProject(ProjectCreateUpdateApiModel projectModel)
{
    var project = new Project();

    project.Name = projectModel.Name;
    project.Description = projectModel.Description;
    project.CreatedAt = projectModel.CreatedAt;
    project.UpdatedAt = projectModel.UpdatedAt;
    project.Budget = projectModel.Budget;
    project.Status = await _context.ProjectStatuses.FindAsync(projectModel.StatusId);

    _context.Projects.Add(project);
    await _context.SaveChangesAsync();

    return CreatedAtAction("GetProject", new { id = project.Id }, project);
}
```

Obr. 9.2 API Metóda - Vytvor projekt

9.1.5 Swagger - Dokumentácia API

Na dokumentáciu API požiadaviek a jednoduchší prístup k jej parametrom a návratovým typom, bol použitý a implementovaný framework Swagger. Pri implementácii bolo nutné zaregistrovať middleware v triede `Startup.cs` (Obr. 9.4). V aplikačných nastaveniach `appsettings.json` (Obr. 9.3), sú uložené základné nastavenia, ktoré sú použité pri registrácií Swagger middlewarov.

```
"SwaggerOptions": {
  "JsonRoute": "swagger/{documentName}/swagger.json",
  "Description": "Evidence API",
  "UIEndpoint": "v1/swagger.json"
},
```

Obr. 9.3 AppSettings - Swagger nastavenia

```
var swaggerOptions = new SwaggerOptions();
Configuration.GetSection(nameof(SwaggerOptions)).Bind(swaggerOptions);

app.UseSwagger(option =>
{
    option.RouteTemplate = swaggerOptions.JsonRoute;
});

app.UseSwaggerUI(option =>
{
    option.SwaggerEndpoint(swaggerOptions.UIEndpoint, swaggerOptions.Description);
});
```

Obr. 9.4 Startup.cs - Registrovanie Swagger middleware

9.1.6 SendGrid - Posielanie emailov

Systém nepoužíva klasický systém registrácie, ale používateľský účet musí byť vytvorený nadriadenými rolami, ktoré už majú prístup do systému. Účet je vytvorený bez hesla. Používateľovi je odoslaný email s odkazom, pomocou ktorého si môže zvoliť heslo a následne sa do systému prihlásiť.

Aby bolo možné emaily reálne odosielať, je nutné zvoliť emailovú službu. V rámci vývoja bola zvolená platforma SendGrid, ktorá bola integrovaná a implementovaná vo vyvíjanom prototype. Do tejto platformy sa bolo nutné zaregistrovať a vytvoriť si API prihlasovacie údaje. Logika odosielania emailov sa nachádza v triede `EmailSender.cs`. Táto trieda implementuje rozhranie `IEmailSender`.

9.2 Prezentačná vrstva

Vrstva, ktorú vidí používateľ systému a volá zvolené požiadavky bola implementovaná pomocou jazyka Javascript s využitím frameworku React.js. Vývoj prebiehal v editore Visual Code, ktorý zastrešuje spoločnosť Microsoft. Je nutné upozorniť, že používateľské rozhranie v prototype bolo optimalizované na Full HD rozlíšenie, takže pri otvorení na inom zariadení nemusí aplikácia vyzerať správne.

9.2.1 Nainštalované balíčky

Pri inštalovaní Javascript balíčkov bol využívaný správca Javascriptových balíčkov - nástroj NPM. Pomocou tohto nástroja vieme zabezpečiť, že pri presune zdrojového kódu, nie je nutné presúvať zdrojové kódy použitých knižníc. Pri spustení konzolového príkazu `npm install`, nástroj automaticky nainštaluje všetky potrebné knižnice, ktoré sú špecifikované v súbore `package.json`.

Pri implementácii bolo okrem knižníc súvisiacich priamo s React.js, použitých aj niekoľko iných knižníc:

- **Material UI** - sada knižníc, ktoré pomohli k rýchlejšiemu vývoju používateľského rozhrania,

- **Moment.js** - využitie pri práci s dátumami a časmi,
- **Axios** - klient, ktorý bol využitý pri vytváraní API požiadaviek,
- **Bootstrap** - sada nástrojov, ktorá pomáha pri tvorbe užívateľského rozhrania webových aplikácií,
- **ESLint** - využitý pri implementácií na analýzu Javascriptového kódu a nájdenie možných problémov v kóde.

9.2.2 Navigácia a prístupové práva

Vyvinutý systém má niekoľko stránok a položiek menu, medzi ktorými môžu používatelia prechádzať. Každý používateľ môže mať priradenú inú rolu, s ktorou súvisia aj iné dostupné akcie, ktoré môže vykonávať. Aplikačná vrstva má pri každej prijatej požiadavke naimplementovanú logiku, ktorá kontroluje, či má daný používateľ právo na danú akciu. V prípade že nemá, daná akcia sa nevykoná a server odošle používateľovi zamietavý stavový kód.

Skutočnosť, že každá používateľská rola má iné práva a nemala by mať dostupné všetky stránky, bola zohľadnená aj pri vývoji prezentačnej vrstvy. V súbore `routes.js` sa nachádzajú všetky informácie o dostupných stránkach v aplikácii. Sú uložené v tvare polí objektov, kde každý objekt predstavuje jednu stránku. Pri každej stránke sú uvedené tieto informácie:

- názov v používateľskom menu,
- cesta v štruktúre aplikácie,
- ikona, ktorá sa zobrazí pri názve v menu,
- React komponenta - funkcia v Javascripte, ktorá bude zaregistrovaná k danej ceste a zavolaná po otvorení danej cesty vo webovom prehliadači,
- pole so zoznamom rolí, ktoré majú právo otvoriť danú stránku (Obr 9.5).

Pri spustení aplikácie sú všetky polia prechádzané a na základe roly prihláseného používateľa je v súbore `App.js` vytvorená pre každý záznam komponenta s názvom `<AuthorizeRoute />`. V tejto triede je kontrolované, či má rola prihláseného používateľa prístup ku danému záznamu. V prípade kladného výsledku, je daná cesta a Javascriptová funkcia zaregistrovaná a používateľovi bude dostupná na danej adrese.

```
export const projectManagement = [
  {
    path: "/projetManagement/employeeProjectEntries",
    name: "Approved entries",
    icon: AccessAlarmIcon,
    component: EmployeeProjectHours,
    roles: ['ADMIN', 'MANAGER']
  }
];
```

Obr. 9.5 Routes.js - Projektové záznamy

Ďalšia kontrola používateľskej roly prebieha pri vykresľovaní menu v súbore nazvanom `SideMenu.js`, čo zabezpečí že používateľ uvidí v grafickom rozhraní iba tie položky menu, ktoré sú mu povolené.

10 ZABEZPEČENIE

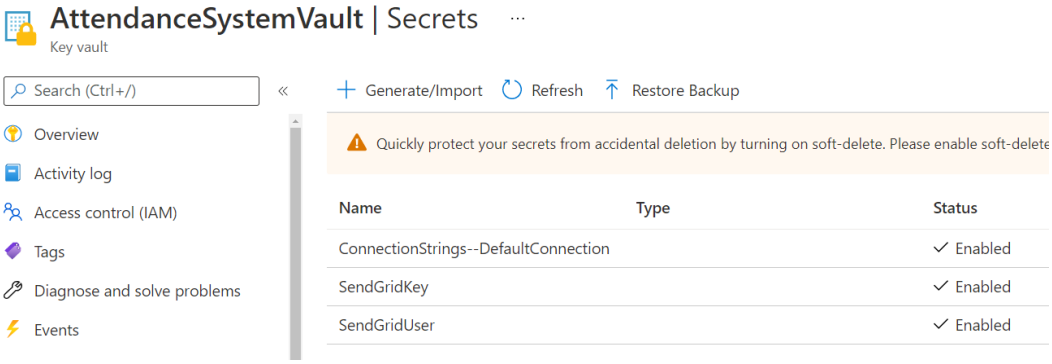
Aj napriek tomu, že aplikácia bude počas jej reálneho používania umiestnená iba na internej sieti zadávateľa a k sieti, majú prístup iba zamestnanci danej siete je nutné dbať na jej zabezpečenie. V tejto kapitole budú popísané bezpečnostné prvky a metódy, ktoré boli pri implementácii použité.

10.1 Ochrana citlivých nastavení a údajov

Implementovaná aplikácia používa niekoľko externých služieb, do ktorých sa musí autentifikovať. V našom prípade to je pripojenie do databázy a využitie mailového klienta na odosielanie emailov. Tieto údaje je nutné niekde uložiť, aby aplikácia mala k nim počas behu prístup. Štandardne aplikácia načítava tieto údaje z konfiguračného súboru, ktorý bol súčasťou verzovaného súboru a zmeny v tomto súbore boli trackované pomocou systému git. Tieto údaje nie sú zašifrované, preto bol v rámci bezpečnosti zvolený iný spôsob.

Pri lokálnom vývoji boli tieto údaje ukladané v tzv. User Secrets. Je to JSON konfiguračný súbor, ktorý má rovnakú štruktúru ako konfiguračný súbor aplikácie, avšak je uchovávaný iba lokálne na danom počítači a nenachádza sa vo verzovanej projektovej štruktúre. [23]

Pri nasadení na web bola využitá služba Key vault (Obr. 10.1), ktorá je súčasťou balíka služieb Azure. Túto služba má priamu integráciu s Aplikačnou službou. Tieto citlivé údaje sú v nej zašifrované a sú prístupné iba nami vytvorenej aplikačnej službe.



Obr. 10.1 Key vault

V prípade, ak by sa aj útočníkovi podarilo získať prihlasovacie údaje do databázy, nemohol by sa k nej len tak jednoducho pripojiť. Prístup do databázy je chránený firewallom, ktorý zabezpečuje, že databáza nie je prístupná verejnosti, ale komunikuje iba so zvolenými IP adresami.

10.2 Zabezpečenie požiadaviek a komunikácie so serverom

Systém je implementovaný ako REST API aplikácia, ktorej používateľská časť komunikuje so serverovou časťou pomocou GET, POST, PUT, DELETE požiadaviek. Tieto požiadavky je nutné zabezpečiť, aby ich server vedel autentifikovať a autorizovať. Aby server mohol overiť, že požiadavku posiela overená identita, je do hlavičky každej požiadavky pridaný JWT (JSON Web token). Tento token sa skladá z troch častí:

- hlavičky,
- dát,
- a podpisu. [24]

Pomocou tohto tokenu dokáže server overiť identitu odosielateľa a danú REST požiadavku vykonať. Avšak token nie je zašifrovaný, preto aplikácia používa iba HTTPS protokol, ktorý zabezpečí, že požiadavka nebude počas prenosu odchytená a upravená inou nebezpečnou osobou. Tento JWT token tiež obsahuje v tele dát aj informácie o role používateľa. Na základe priradenej roly sú na strane používateľa vytvorené tzv. autorizované cesty. Tým je zabezpečené, že sa používateľ nedostane na stránku, ktorú nemá povolenú. To, či má používateľ prístup k nejakej akcii je overované aj na strane serveru na základe priradených povolení.

10.3 Zabezpečenie autentizácie a relácie

Do aplikácie sa používatelia prihlasujú pomocou mena a hesla. Tieto údaje je nutné zabezpečiť, aby neboli odchytené a zneužité treťou stranou. Aplikácia preto na odosielanie týchto informácií používa POST požiadavku a HTTPS protokol, ktorý zabezpečí ochranu týchto údajov.

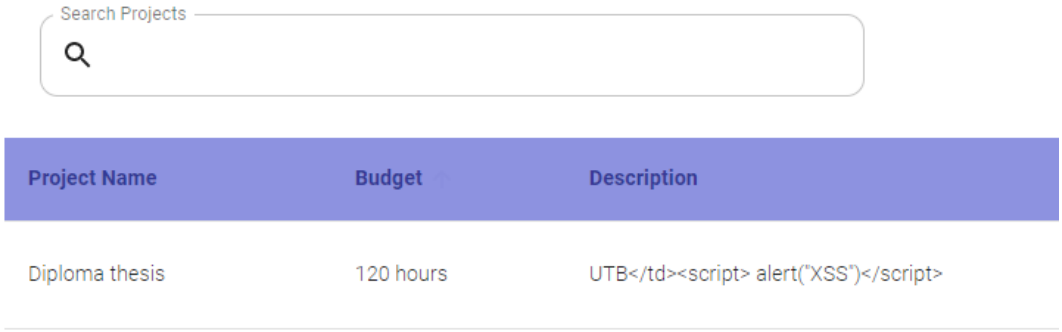
Pri prvom prihlásení je používateľovi poslaný email s webovým odkazom, ktorý má iba obmedzenú platnosť. Pomocou tohto linku si zvolí svoje prvé heslo. Týmto je zabezpečené, že heslo nebude posielané ako čitateľný text cez zraniteľné kanály.

10.4 Zabezpečenie pred XSS útokmi

Aby bola aplikácia zabezpečená pred Cross-Site Scriptingom, je nutné ošetriť, aby text, ktorý zadal používateľ a je uložený v databáze, nespustil škodlivý kód pri vykresľovaní v prehliadači.

O toto zabezpečenie sa z veľkej častí stará samotný framework React. Každý text ktorý je vykresľovaný v prehliadači je automaticky ošetrený a nebezpečné znaky sú kódované. Takže v prípade, že sa používateľ pokúsi o vloženie škodlivého kódu, nepodarí

sa mu to. React zabezpečí, že sa zadaný kód zobrazí ako reťazec znakov a nespustí sa, viď obrázok 10.2.



Obr. 10.2 Prevencia vočí XSS útok

10.5 Zabezpečenie pred SQL Injection

Na prácu s dátami a databázou je využívaný Entity Framework Core. Ako už bolo spomínané v implementačnej časti, tento ORM framework slúži na prácu s dátami a databázou. Na prístup k dátam nie sú používané klasické SQL dotazy ale tzv. LINQ. Sú to príkazy písané v jazyku C# , ktoré sú následne automaticky pomocou frameworku transformované do SQL príkazu.

Týmto je systém zabezpečený aj proti potencionálnym vložením nebezpečných SQL príkazov.

10.6 Zabezpečenie pred CSRF útkomi

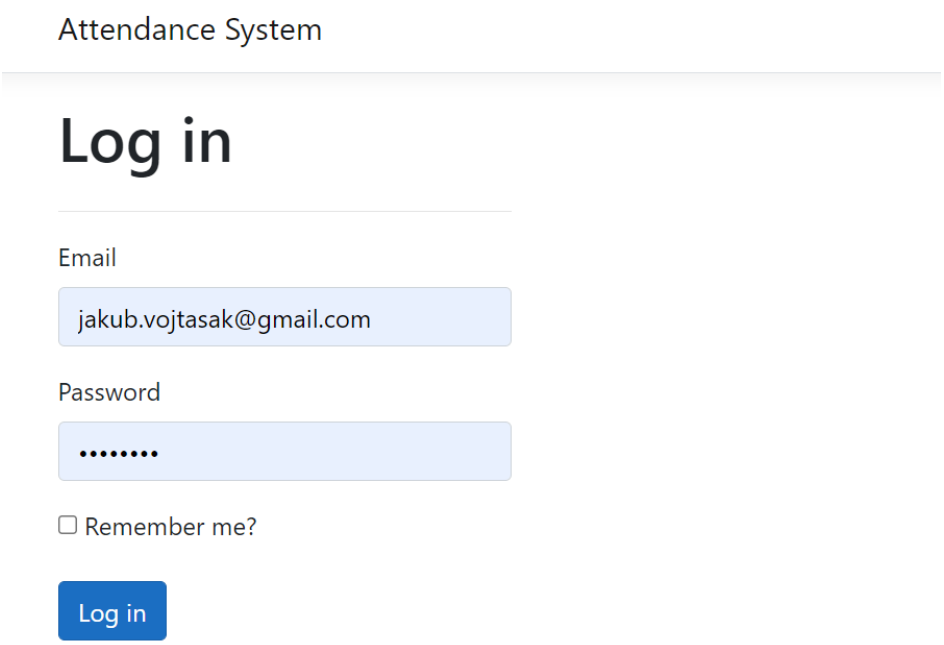
Systém nevyužíva iba Cookies ako spôsob autentifikácie požiadaviek, ktoré môžu byť zneužité pri Cross-site request forgery útokoch. Do hlavičky každej požiadavky je vložený autorizačný JWT token, ktorý je uložený v lokálnej pamäti prehliadača. Keďže CSRF útoky fungujú na princípe zneužitia Cookies, použitím lokálnej pamäte a JWT je systém pred týmto útokom zabezpečený.

11 SPRIEVODCA APLIKÁCIOU

Táto kapitola sa venuje používateľskému rozhraniu vytvorenej aplikácie. Postupne budú opísané všetky funkcionality vytvorenej aplikácie. Sprievodca môže slúžiť ako príručka pre zamestnancov, ktorí budú systém využívať. Je v pláne túto príručku v budúcnosti vždy aktualizovať aj po pridaní nových funkcionalít. Ako bolo spomenuté v predchádzajúcich kapitolách, spoločnosť, ktorá bude vyvinutý systém používať sa plánuje rozrásť, preto by mali noví zamestnanci po prečítaní tejto príručky so systémom vedieť samostatne pracovať.

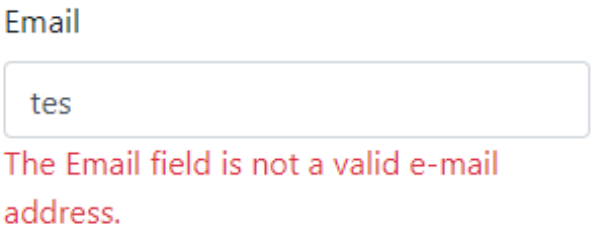
11.1 Prihlásenie používateľa

Po otvorení aplikácie sa používateľovi zobrazí úvodný formulár (Obr. 11.1). Pre vstup do aplikácie musí zadať email a heslo. Heslo si užívateľ zvolí sám po otvorení aktivačného linku, ktorý mu príde na firemný email. Na úvodnej stránke je aj možnosť obnovenia hesla.



Obr. 11.1 Prihlasovacia stránka používateľa

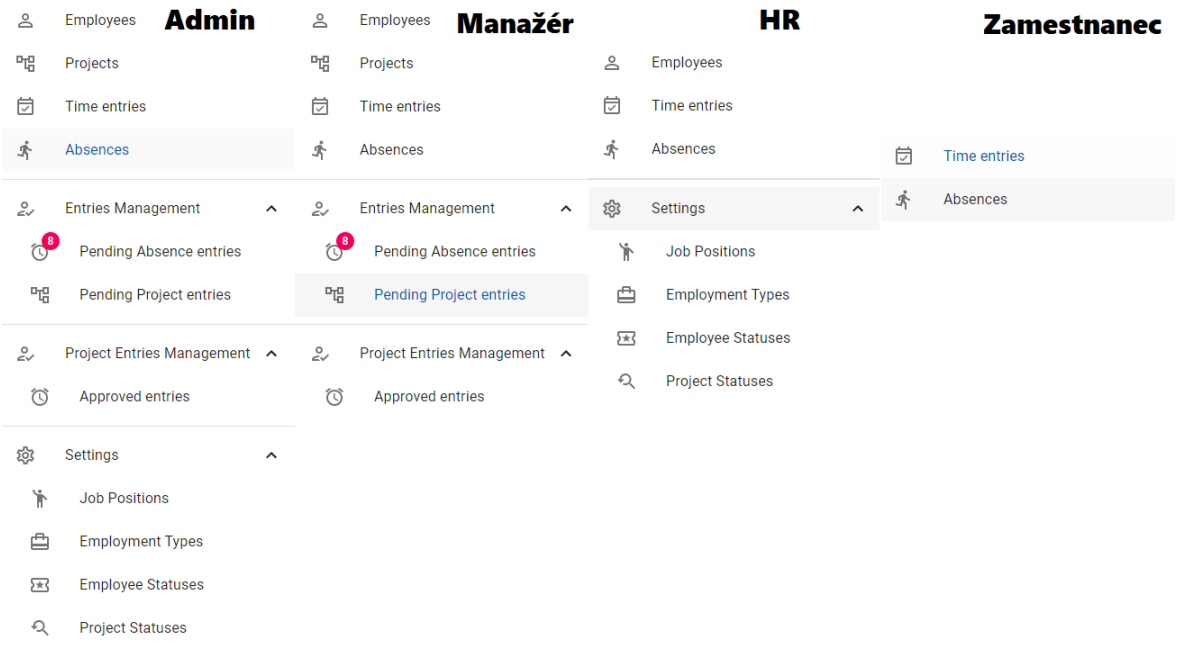
V prípade, že užívateľ zadá nesprávny email alebo heslo, na obrazovke sa zobrazí text červeným písmom, že išlo o neúspešný pokus o prihlásenie. V prípade, že do poľa email, používateľ zadá email v zlom tvare, je o tom upozornený ešte pred odoslaním formulára (Obr 11.2).



Obr. 11.2 Chybný email

11.2 Menu pre jednotlivé roly

Pri vytváraní zamestnancov je nutné každému zamestnancovi priradiť rolu. Na základe priradenej roly, sa prihlásenému používateľovi zobrazí rozdielne menu na ľavej strane obrazovky. Menu je rozdelené do štyroch častí. Na obrázku 11.3 sú zobrazené položky menu pre jednotlivé roly. Podpoložky, ktoré sa nachádzajú v skupinách je možné kliknutím na hlavnú položku skryť alebo zobraziť. Sú zobrazené iba tie položky, ku ktorým ma daná rola právo. V nasledujúcich podkapitolách budú postupne vysvetlené funkcionality jednotlivých položiek menu.



Obr. 11.3 Menu na základe roly používateľa

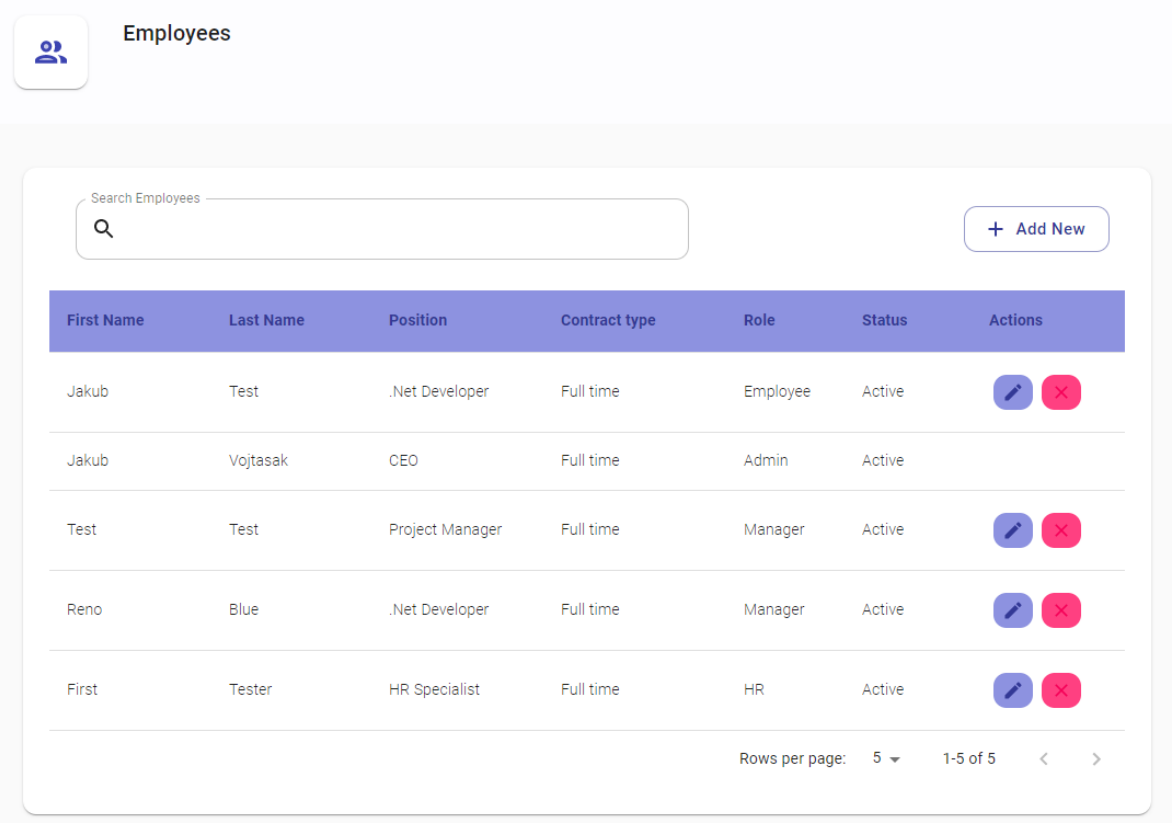
11.3 Zamestnanci

Prvá položka v menu je určená správe zamestnancov. Po kliknutí na túto položku sa zobrazí zoznam zamestnancov, ktorí sú v systéme vytvorení (Obr. 11.4). V spodnej časti si môže požívateľ zvoliť počet záznamov na jednej stránke a šípkami sa medzi

týmito stránkami pohybovať.

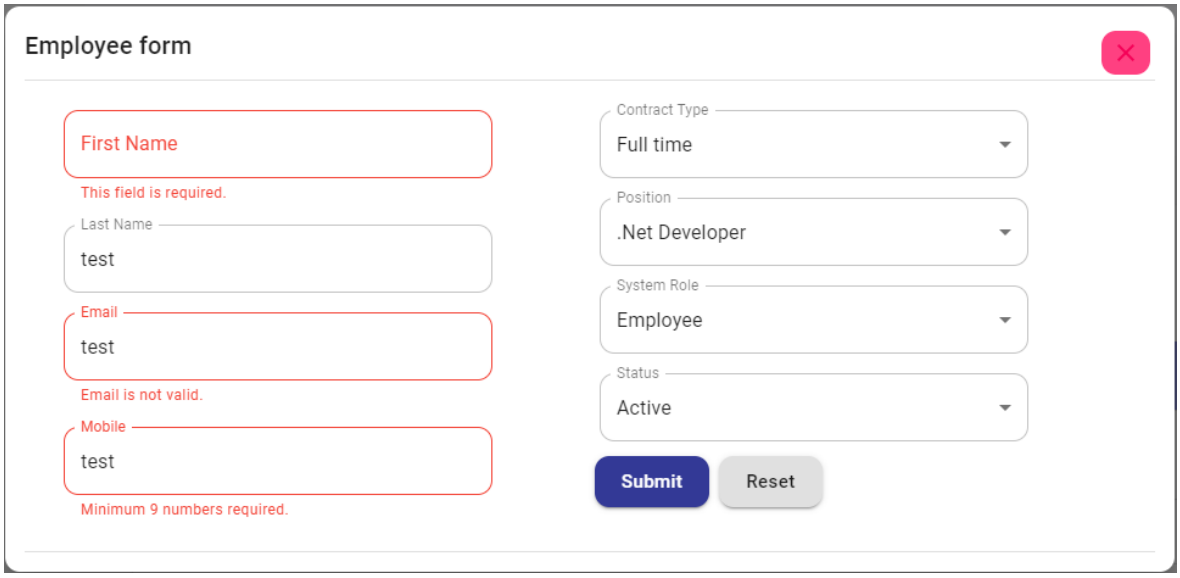
Pri každom zamestnancovi je možné vidieť jeho pozíciu vo firme, typ kontraktu, jeho rolu v systéme a status. V poslednom stĺpci sú pri každom zázname zobrazené 2 tlačidlá, pomocou ktorých môže používateľ upraviť alebo zmazať daný záznam. Užívateľov, ktorí majú rolu Admin môže upravovať iba používateľ s rovnakou rolou.

Nad hlavičkou tabuľky sa na ľavej strane nachádza pole, pomocou ktorého môže používateľ jednotlivé záznamy filtrovať na základe mena a priezviska. Záznamy sú filtrované dynamicky bez nutného obnovenia stránky. Na pravej strane sa nachádza tlačidlo na pridanie nového zamestnanca.



Obr. 11.4 Zoznam zamestnancov

Po kliknutí na tlačidlo pridať zamestnanca sa zobrazí vyskakovacie okno s formulárom, ktorý je nutné vyplniť. V prípade, že používateľ niektoré z povinných polí nevyplní, formulár sa neodošle, ale upozorní používateľa, že niektoré údaje nevyplnil. Taktiež je kontrolovaná validita pri emaile a telefónom čísle (Obr. 11.5). V pravom dolnom rohu je okrem tlačidla odoslať aj tlačidlo „Reset“, ktoré po kliknutí vyprázdni daný formulár.

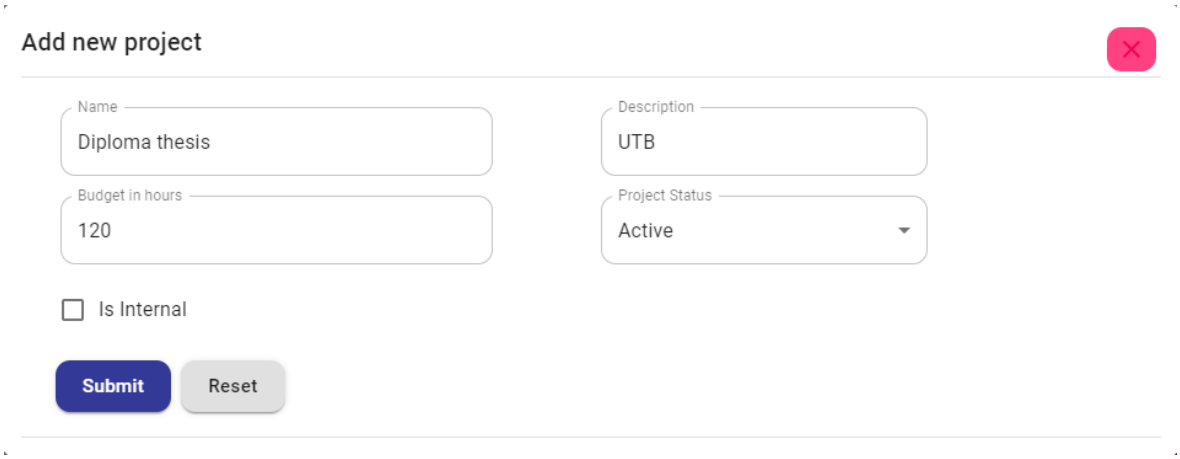


Obr. 11.5 Okno pridať zamestnanca

Po odoslaní formulára je zamestnancovi na uvedený email odoslaný uvítací email s webovým odkazom na aktivovanie účtu v systéme (Príloha 3.1). Po kliknutí na daný odkaz sa mu zobrazí formulár (Príloha 4.1), v ktorom vyplní svoj pridelený email a zvolí si heslo, ktoré bude používať na autentifikáciu do systému.

11.4 Správa projektov

Druhá položka v menu je určená správe projektov. V tejto časti je možné vytvárať projekty, ktoré si následne zamestnanec môže vybrať pri zadávaní odpracovaných hodín. Pri projekte je možné zadať názov projektu, rozpočet hodín na projekt, popis k projektu a status projektu (Obr 11.6). Taktiež je možné označiť projekt ako interný. Počet hodín, ktoré používateľ zadá do poľa rozpočet bude slúžiť na kontrolu pri schvaľovaní odpracovaných hodín zamestnancov.

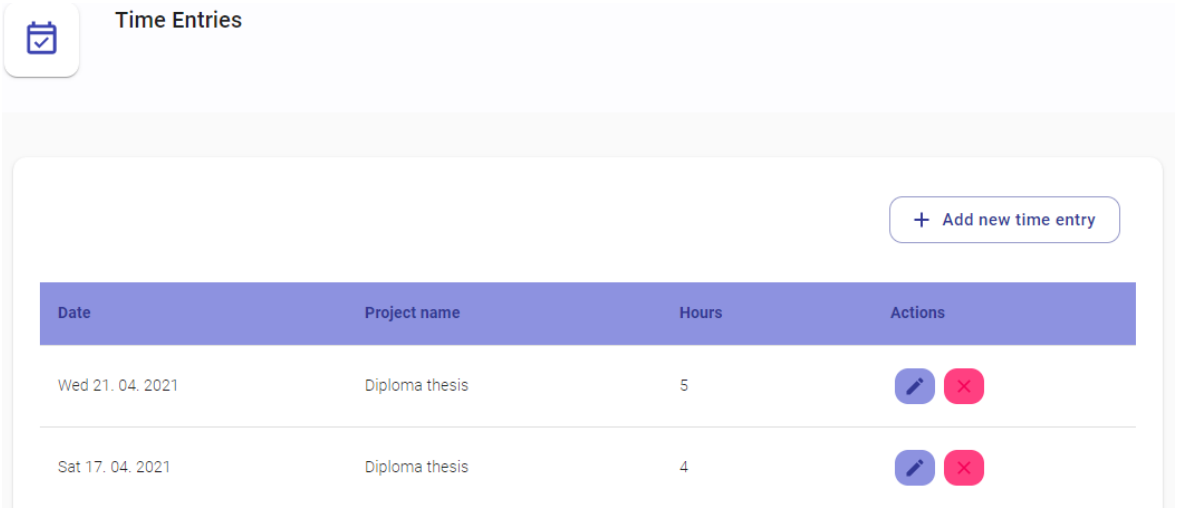


Obr. 11.6 Vytvoriť nový projekt

11.5 Časové záznamy

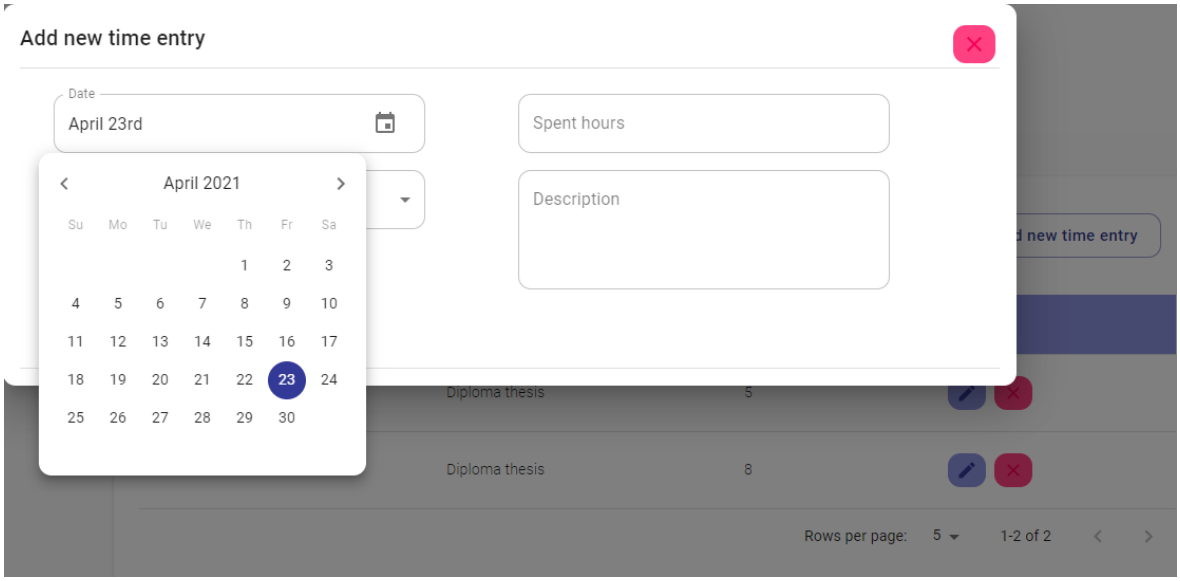
Po kliknutí na položku „Časové záznamy“ sa používateľovi zobrazia časové záznamy, ktoré do systému zadal (Obr. 11.7). Záznamy sa zobrazujú od najnovšieho po najstarší. Používateľovi sa zobrazujú záznamy, ktoré zadal za daný mesiac a boli schválené alebo čakajú na schválenie. Taktiež sa mu zobrazia aj všetky záznamy, ktoré boli vrátené alebo neboli stále schválené. Na každom riadku sa v poslednom stĺpci nachádzajú dve tlačidlá, pomocou ktorých môže používateľ záznam upraviť alebo zmazať. V prípade, že zamestnanec upraví záznam, ktorý bol už schválený, vráti sa naspäť do neschváleného stavu a bude znovu odoslaný na schválenie nadriadeným.

Pre prehľadnosť si v spodnej časti tabuľky môže používateľ zvoliť počet záznamov na jednej karte a šípkami medzi týmito kartami prechádzať.



Obr. 11.7 Zoznam časových záznamov

Po kliknutí na tlačidlo „Pridať nový časový záznam“ sa používateľovi zobrazí vyskakovacie okno s formulárom (Obr. 11.8). Používateľ musí vyplniť dátum, vybrať z listu projekt, vyplniť počet odpracovaných hodín a vyplniť poznámku na čom pracoval. Pre jednoduchší výber dátumu sa používateľovi zobrazí po kliknutí na ikonu kalendára okno, pomocou ktorého si môže jednoducho vybrať ľubovoľný dátum. Po odoslaní formulára sa časový záznam pošle na schválenie manažérom.

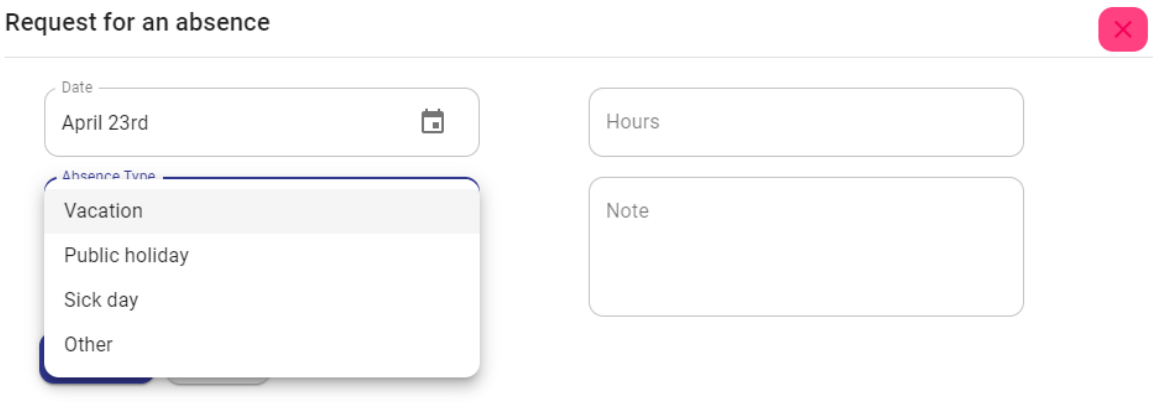


Obr. 11.8 Pridať časový záznam

11.6 Absencie

V časti „Absencie“ sa zobrazia používateľovi záznamy o jeho absenciách v práci. Pri každom zázname je zobrazený dátum, typ absencie, počet hodín a stav schvaľovania (Obr. 11.9). Ak žiadosť o absenciu ešte nebola schválená, je možné daný záznam vymazať alebo editovať.

Pri vytváraní záznamu je nutné vybrať dátum, typ absencie, počet hodín a poznámku k žiadosti. Ako pri vytváraní časových záznamov je možné použiť nástroj na zvolenie dátumu.

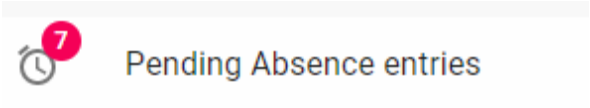


Obr. 11.9 Vytvoriť žiadosť o neprítomnosť

11.7 Schválenie absencií

V sekcii „Management časových záznamov“ sa nachádzajú dve položky menu, pomocou ktorých je možné záznamy vytvorené v sekciách „Absencie“ a „Časové záznamy“ schvaľovať. Prvou položkou je „Schválenie neprítomností“.

V tejto časti sa používateľovi zobrazia všetky záznamy, ktoré čakajú na schválenie. V prípade, že nejaké záznamy čakajú na schválenie, je o tom používateľ upozornený červeným odznakom s počtom čakajúcich žiadostí. Tento odznak sa zobrazí pri ikone danej položky v menu (Obr. 11.10).



Obr. 11.10 Počet čakajúcich žiadostí

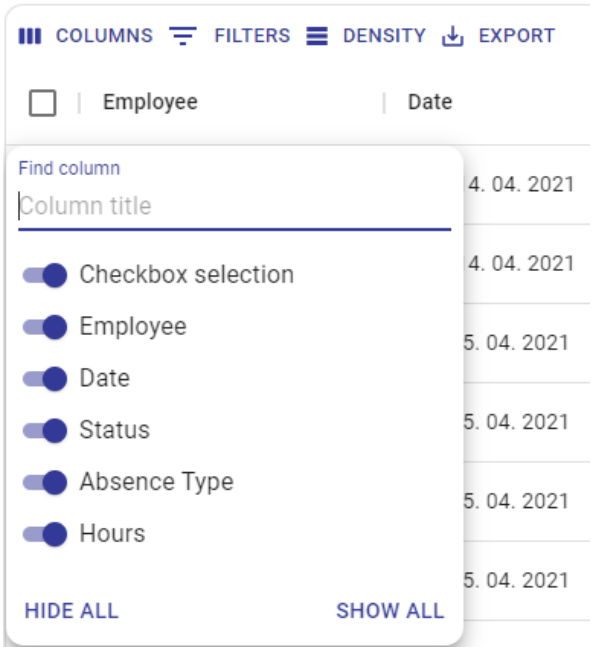
V tabuľke záznamov (Obr. 11.11) sa nachádzajú informácie o mene zamestnanca, dátume, kedy zamestnanec žiada o voľno, statuse schvaľovania, typu absencie a počte hodín. Vedľa každého záznamu sa v prvom stĺpci nachádza políčko, pomocou ktorého môže používateľ daný záznam označiť. Označené záznamy môže následne pomocou dvoch tlačidiel nad tabuľkou následne schváliť alebo zamietnuť. Používateľ má tiež možnosť označiť všetky záznamy, pomocou označovacieho políčka v hlavičke tabuľky.

Obr. 11.11 Vytvoriť žiadosť o neprítomnosť

Pracovník má možnosť v používateľskom prostredí nad záznamami vykonávať niekoľko operácií, ktoré mu môžu pomôcť v prehľadnosti a rýchlejšiemu schvaľovaniu. Nad hlavičkou tabuľky sa nachádzajú 3 tlačidlá, ktoré slúžia na grafickú úpravu a filtrovanie záznamov a tlačidlo na export záznamov.

11.7.1 Tlačidlo Stĺpce

Tlačidlo „Stĺpce“ slúži na filtrovanie jednotlivých stĺpcov, ktoré používateľ vidí. Pomocou prepínačov si môže používateľ vybrané stĺpce skryť alebo zobraziť. V prípade, že sa v budúcnosti táto tabuľka rozšíri o veľké množstvo stĺpcov, je možné stĺpce vyhľadávať a filtrovať aj na základe zadaného textu. V spodnej časti sa nachádzajú dve tlačidlá, pomocou ktorých je možnosť všetky stĺpce naraz skryť alebo zobraziť (Obr. 11.12).



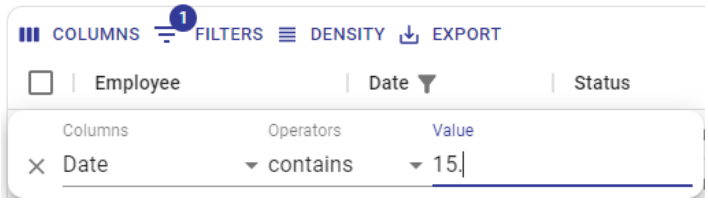
Obr. 11.12 Tlačidlo Stĺpce

11.7.2 Tlačidlo Filtre

Pomocou tlačidla „Filtre“ si používateľ môže filtrovať jednotlivé záznamy (Obr. 11.13). Je nutné vybrať stĺpec, operátor vyhľadávania a text, ktorý používateľ hľadá. Je možné vybrať z nasledovných operátorov:

- obsahuje - zadaný reťazec sa nachádza v hľadanej hodnote,
- rovná sa - zadaný reťazec sa musí rovnať hľadanej hodnote,
- začína s - hľadaná hodnota začína so zadaným reťazcom,
- končí s - hľadaná hodnota končí so zadaným reťazcom.

V prípade, že má používateľ filter zapnutý, pri tlačidle Filtre sa zobrazí informatívny odznak s počtom aktivovaných filtrov.



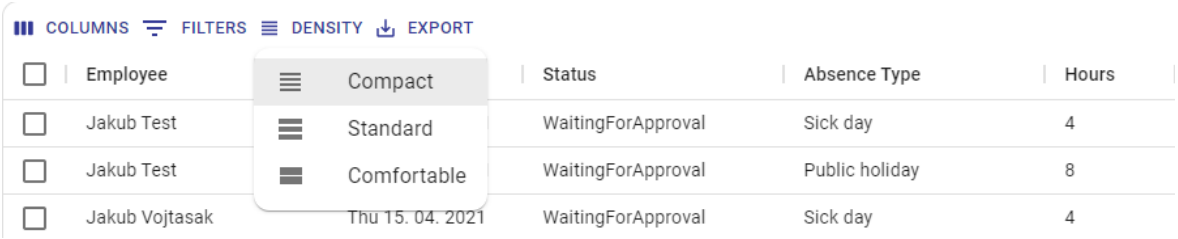
Obr. 11.13 Tlačidlo Filtre

11.7.3 Tlačidlo Hustota

Tlačidlo „Hustota“ slúži pre komfortnejšie zobrazenie záznamov. Používateľ si môže po kliknutí na tlačidlo zvoliť šírku jednotlivých záznamov. Je možné si vybrať z troch druhov zobrazení:

- kompaktné,
- štandardné,
- komfortné.

Na obrázku 11.14 môžme vidieť aktivované kompaktné zobrazenie.



Obr. 11.14 Tlačidlo Hustota

11.7.4 Tlačidlo Export

Posledné tlačidlo „Export“ slúži na export záznamov do CSV súboru. Po kliknutí na tlačidlo sa exportujú záznamy, ktoré používateľ vidí na obrazovke (Obr. 11.15). To znamená, že filtre, ktoré používateľ nastavil pomocou nástrojov Stĺpce a Filtre sa použijú aj pri exporte. V prípade, že sú niektoré záznamy označené, sú exportované iba dané záznamy.

	A	B	C	D	E
1	Employee	Date	Status	Absence Type	Hours
2	Jakub Test	2021-04-14	WaitingForApproval	Sick day	4
3	Jakub Test	2021-04-14	WaitingForApproval	Public holiday	8
4	Jakub Vojtasak	2021-04-15	WaitingForApproval	Sick day	4

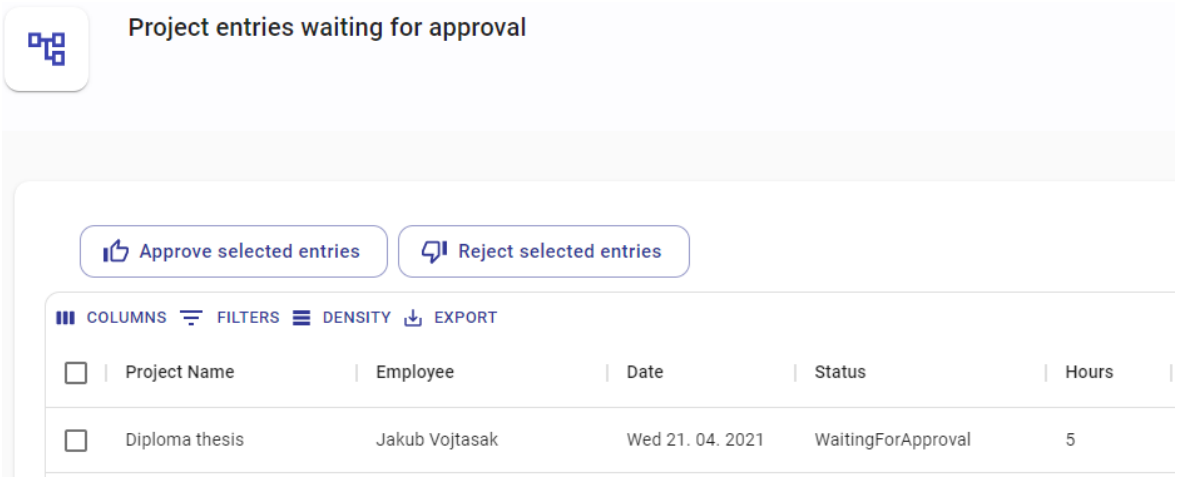
Obr. 11.15 Exportovaný súbor

11.8 Schválenie časových záznamov

Druhou položkou v sekcii „Management časových záznamov“ je „Schválenie projektových záznamov“. V tejto obrazovke sa používateľovi zobrazia všetky časové záznamy, ktoré čakajú na schválenie.

Grafické rozhranie, funkcie a nástroje, ktoré táto stránka ponúka je veľmi podobné s položkou „Schválenie neprítomnosti“. Používateľ má k dispozícií nástroje na upravovanie, filtrovanie a export zobrazených dát. Ako pri schvaľovaní záznamov o neprítomnosti, má možnosť záznamy označiť a následne pomocou dvoch tlačidiel vo vrchnej časti schváliť alebo zamietnuť.

Hlavným rozdielom medzi týmito dvomi obrazovkami je stĺpec Projekt, ktorý obsahuje názov projektu, ku ktorému sa viaže daný časový záznam (Obr. 11.16).



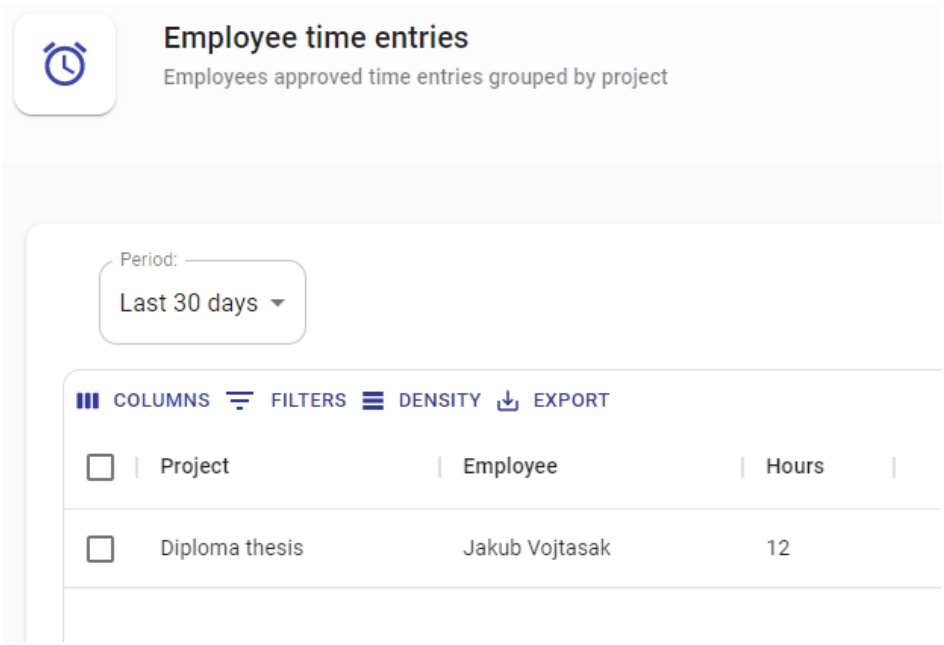
Obr. 11.16 Schválenie projektových záznamov

11.9 Schválené záznamy

Tretia sekcia v menu je určená schváleným hodinám zamestnanca. Po kliknutí na položku „Schválené záznamy“ sa používateľovi zobrazia záznamy zamestnancov zoskupené na základe projektov, na ktorom pracovali za dané časové obdobie, ktoré je možné zvoliť pomocou rozbaľovacej ponuky, ktorá sa nachádza nad tabuľkou. Štandardne je nastavené obdobie „Tento mesiac“, ale používateľ má na výber ešte tieto dve možnosti:

- posledných 30 dní,
- všetky časové záznamy,

Po vybratí ktorejkoľvek možnosti sa záznamy automaticky zobrazia na základe vybraného časového obdobia. Počet hodín uvedený pri zázname je súčet všetkých hodín za časové obdobie a za daný projekt (Obr. 11.16).



Obr. 11.17 Schválené časové záznamy

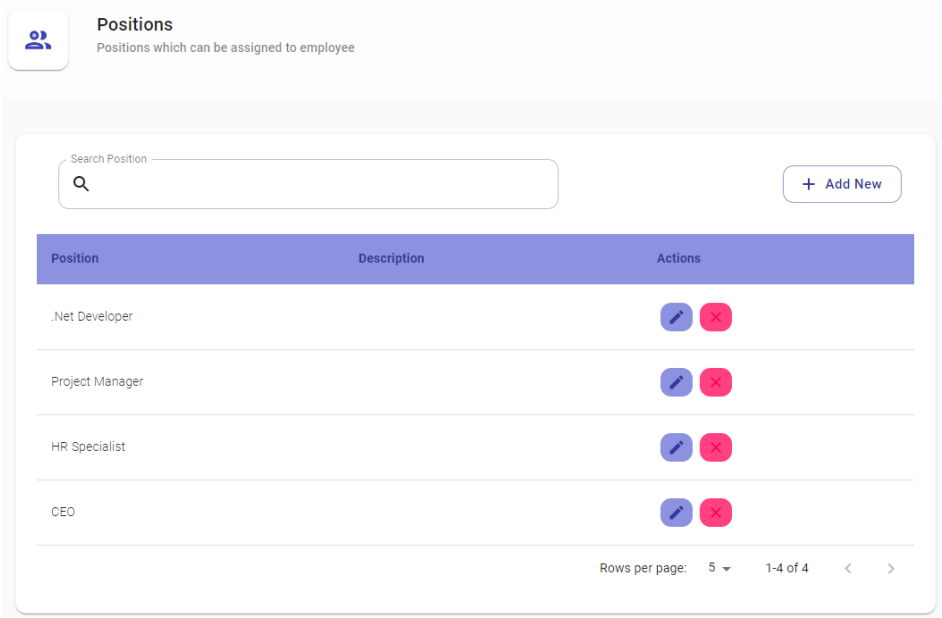
11.10 Nastavenia

Posledná sekcia je určená na vytvorenie a nastavenie rozbaľovacích polí, ktoré si používateľ v systéme vyberá najmä pri vytváraní zamestnanca, alebo pri vytváraní nového projektu. Táto sekcia je primárne určená roli Admin a niektoré časti sú prístupné aj Personálnemu pracovníkovi. V tejto sekcii sa nachádzajú tieto položky menu:

- Pracovné pozície,
- Typy zamestnania,
- Statusy zamestnanca,
- Projektové statusy.

Z používateľského rozhrania vyzerajú všetky stránky spomenutých nastavení takmer identicky, ako je vidieť na obrázku 11.18. Líšia sa názvom stránky, prípadne ikonou.

Pri vytváraní možností pre jednotlivé nastavenia, je nutné zadať názov a prípadne je možné zadať poznámku. Nastavenie je možné mazať a upravovať pomocou tlačidiel, ktoré sa nachádzajú pri každom nastavení. Nad tabuľkou sa nachádza pole, pomocou ktorého je možné vyhľadávať záznamy a filtrovať na základe hľadaného názvu.



Obr. 11.18 Schválené časové záznamy

12 NASADENIE, TESTOVANIE A PLÁNOVANÉ ROZŠÍRENIA

12.1 Nasadenie aplikácie

Po dokončení prototypu a lokálneho vývoja, bola aplikácia nasadená na web. Pri nasadení bola využitá služba Azure a jej testovacie servery, ktoré sú po zaregistrovaní a overení študentského emailu dostupné na 12 mesiacov bezplatne.

Pre aplikáciu bolo nutné vytvoriť na platforme Azure službu `App Service`. Následne bolo nutné nakonfigurovať ďalšie nastavenia, ktoré sú potrebné pre správne spustenie projektu na serveri. Po prvom pokuse nasadenia aplikácie sa vyskytli ďalšie problémy, ktoré bolo nutné vyriešiť. Po nasadení na server je zložité hľadanie chýb v aplikácii, pretože štandardne vývojár nedostane chybovú hlášku, ale iba kód chyby. Pre účel jednoduchšieho hľadania chýb, bol v aplikácii naimplementovaný a integrovaný nástroj `Application Insights`. Tento nástroj je určený vývojárom pre monitorovanie behu aplikácie, sledovanie vytvorených požiadaviek a detekovanie prípadných anomálií a chýb. Okrem spomenutých informácií nástroj ponúka množstvo ďalších užitočných informácií spojených s telemetriou. [25]

Po implementovaní nástroja v aplikácii, bolo nutné túto službu pridať aj v prostredí Azure a spojiť ju s vytvorenou službou `App service`, ktorá zabezpečuje beh aplikácie na serveri. Následne po spustení služby `Application Insights` v Azure, pomocou nástroja `Live Metrics` bolo možné vidieť hlášky chýb v nasadenej aplikácii. Všetky chyby boli odstránené a aplikáciu bolo možné spustiť.

12.2 Testovanie aplikácie

Po úspešnom nasadení aplikácie na testovací server služby Azure, bola aplikácia nasadená aj na server zadávateľa. Bolo nutné vytvoriť aj nový databázový server a všetky nutné nastavenia znovu nakonfigurovať. Aplikáciu sa podarilo úspešne nasadiť a spustiť aj na serveroch zadávateľa.

Po úspešnom nasadení, bolo priamo u zadávateľa zorganizované krátke stretnutie ohľadom vyvinutého prototypu evidenčného systému. Školenia sa zúčastnil autor práce, ktorý toto školenie viedol, vedúci pobočky, manažér a jeden vývojár. Pred stretnutím bol každému zúčastnenému poslaný aj sprievodca aplikáciou, ktorý je súčasťou tejto práce. Na tomto stretnutí autor práce predstavil používateľské rozhranie, funkcie aplikácie a plánované rozšírenia. Po predstavení bol každému zúčastnenému vytvorený používateľský účet v aplikácii s priradenou rolou odpovedajúcej pracovnej pozícii. V rámci stretnutia mali zúčastnení možnosť so systémom samostatne pracovať. Počas testovania padlo niekoľko otázok, ktoré autor práce zodpovedal. Prototyp splnil očakávania vedenia spoločnosti a bolo jednohlasne odsúhlasené ďalšie rozšírenie systému.

12.3 Návrhy na vylepšenie

V rámci stretnutia bola autorovi práce poskytnutá kladná spätná väzba k prototypu. Všetky námety na vylepšenia a ďalšie rozšírenia boli spísané a sú uvedené v nasledujúcej časti:

- rola Admin by mala mať možnosť resetovať heslo používateľovi,
- rozpočet dovolenky u zamestnanca - automatická kontrola počtu využitých dní,
- kontrola počtu odpracovaných hodín,
- pridanie dočasne zastupujúceho zamestnanca,
- pridanie úvodnej stránky s informáciami o prihlásenom zamestnancovi - počet odpracovaných hodín v danom mesiaci, dostupná / použitá dovolenka v rámci roka, rola v systéme
- pri správe projektov zobraziť počet hodín, ktoré boli už schválené na daný projekt,
- pri schvaľovaní odpracovaných hodín, kontrolovať či nie je prekročený rozpočet - možnosť zobraziť zostávajúci počet hodín,
- pri liste zamestnancov, možnosť skryť už neaktívnych zamestnancov,
- pridať kalendár s vizualizáciou ľudí, ktorí majú v dané dni dovolenku,
- prispôsobiť používateľské rozhranie aj na mobilné rozlíšenie.

Uvedený zoznam návrhov určite nie je konečný a bude sa v budúcnosti ďalej rozširovať. Po ich implementácií je naplánovaná aj skúšobná prevádzka, ktorej by sa mali zúčastniť všetci zamestnanci. V rámci tejto mesačnej skúšobnej prevádzky budú využívať obidva spôsoby evidencie súčasne. Po ukončení skúšobnej prevádzky budú porovnané výsledné reporty z aktuálne používaného systému s reportami z nového systému. Ak sa budú výsledky zhodovať a nenájdu sa závažné chyby, bude aktuálny evidenčný systém nahradený novým vyvinutým systémom.

ZÁVER

Cieľom tejto diplomovej práce bolo navrhnutie a implementácia prototypu dochádzkového systému so zameraním sa na bezpečnosť webových aplikácií. Spoločnosť už podobný systém aktuálne využíva, ale nespĺňa všetky ich požiadavky.

Trh ponúka množstvo riešení a každé z riešení má svoje výhody a nevýhody. Riešenie bolo navrhované pre malú softvérovú spoločnosť, ktorá sa bude v blízkej budúcnosti značne rozširovať, čo určite spôsobí aj nárast a zmenu požiadaviek na systém. Síce niektoré z riešení by bolo určite možné upraviť požiadavkám zadávateľa, ale náklady na údržbu a ďalšiu úpravu na zákazku by systém značne predražili. Keďže spoločnosť disponuje technologickými znalosťami, ktoré sú potrebné pri vývoji a úpravách webových systémov, bolo rozhodnuté, že najlepšou voľbou bude si navrhnúť a naimplementovať vlastný systém.

Aby autor práce úspešne splnil cieľ práce, bolo dôležité vybrať si správnu metodiku a stanoviť si postup práce. Pri implementácii sa vyskytlo niekoľko problémov a otázok, avšak vďaka výbornej komunikácii so zadávateľom všetky problémy boli rýchlo vyriešené. Prototyp so základnými funkcionalitami sa podarilo implementovať v stanovenom termíne.

Po úspešnom nasadení na interný server zadávateľa, autor odprezentoval výsledky práce pred vedením spoločnosti. Autor predstavil grafické rozhranie aplikácie, základné funkcionality systému a zatiaľ plánované rozšírenia. Vedúci pracovníci mali tiež prístupný manuál a možnosť samostatne pracovať so systémom a otestovať implementované funkcie. Počas testovania padlo niekoľko návrhov na vylepšenia, ktoré boli zapísané.

Po otestovaní prototypu vedením a zamestnancom spoločnosti, dostal autor práce kladné hodnotenie od zadávateľa a súhlas na pokračovanie v implementácii ďalších rozšírení. Na základe úspešného odprezentovania výsledkov a následného kladného hodnotenia od zadávateľa považujem túto prácu za úspešnú. Táto práca je rozdelená do do dvanástich kapitol.

Úvod práce sa zaoberá výhodami informačných systémov a nevýhodami ukladania informácií v papierovej forme. Je tu stručne opísaný obsah práce a plánovaný postup pri dosahovaní stanoveného cieľa.

V prvej kapitole je popísané fungovanie spoločnosti, pre ktorú je systém určený. Sú tu tiež špecifikované základné funkcie existujúcich dochádzkových systémov.

V druhej kapitole sú analyzované vybrané dostupné riešenia. Pri každom vybranom systéme sú analyzované výhody a nevýhody, orientačná obstarávacia cena riešenia pri 15, 50 a 100 zamestnancoch. Prípadne sú spomenuté zaujímavé funkcionality systémov. V závere kapitoly sú riešenia zhodnotené a sú uvedené dôvody, prečo bolo rozhodnuté

implementovať vlastný systém.

Tretia kapitola sa zaoberá požiadavkami zadávateľa. Požiadavky sú na základe unifikovaného vývoja aplikácií rozdelené na funkčné a nefunkčné. Funkčné požiadavky predstavujú interakcie používateľov, ktoré musí systém spĺňať. V nefunkčných požiadavkách sú spomenuté nároky na aplikáciu z technologického a implementačného hľadiska.

Štvrtá kapitola sa zameriava na legislatívne požiadavky, konkrétne na zákonník práce. Je uvedené znenie zákona a v závere je spomenuté, akým spôsobom funguje zadávateľ.

Piata kapitola opisuje a rozoberá princípy a hrozby možných webových útokov, pred ktorými je nutné aplikáciu zabezpečiť. Pri každej vybranej hrozbe je opísaný princíp fungovania a jednoduchý príklad vykonávania útoku.

Šiesta kapitola je určená aktuálnemu stavu v spoločnosti u zadávateľa. Je opísaná jej štruktúra a spôsob, ako aktuálne zamestnanci zaznamenávajú odpracované hodiny a akým zdĺhavým a nevyhovujúcim spôsobom funguje schvaľovanie dovolenky. Záver kapitoly je venovaný problémom, s ktorými sa spoločnosť stretáva pri používaní aktuálneho riešenia.

V siedmej kapitole sú opísané hlavné technológie a ich použitie, ktoré boli použité pri vývoji evidenčného systému. Technológie boli vybrané na základe konzultácie so zadávateľom. Autor si musel väčšinu technológií pred použitím naštudovať.

Ôsma kapitola popisuje návrh implementovaného systému. Boli určené prípady použitia a stanovení aktéri systému. Títo aktéri predstavujú systémové roly. Ku každému z hlavných prípadov použitia bol rozpísaný úspešný a alternatívny scenár. Súčasťou tejto kapitoly je aj návrh tried a ich atribútov a ich vzájomné vzťahy.

V deviatej kapitole sú prakticky opísané použité technológie a spôsob vyriešenie daných problémov. Táto kapitola je rozdelená na dve časti, na implementáciu aplikačnej časti a na implementáciu prezentačnej časti.

Desiata kapitola sa zaoberá spôsobmi praktického riešenia problémov so zabezpečením systému. Každá podkapitola opisuje riešenie konkrétneho problému a použitých technológií.

V jedenástej kapitole sa nachádza používateľská príručka. Príručka slúži ako sprievodca aplikáciou pre zamestnancov, ktorí budú s implementovaným systémom pracovať. V príručke sú opísané jednotlivé stránky aplikácie a ich funkcionality.

Dvanásta kapitola opisuje spôsob, akým bola prototypová aplikácia nasadená na web a ako boli riešené problémy pri nasadení. Po úspešnom nasadení bol prototyp otestovaný zadávateľom, ktorý poskytol návrhy na vylepšenie. Súčasťou kapitoly sú návrhy na vylepšenia a ďalšie plánované rozšírenia.

ZOZNAM POUŽITEJ LITERATÚRY

[1] *§ 96 zákona č. 262/2006 Sb., zákoník práce.* In: Zákony pro lidi.cz [online]. © AION CS 2010-2021 [cit. 20. 2. 2021]. Dostupné z: https://www.zakonyprolidi.cz/cs/2006-262#p96

[2] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky.* 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

[3] *Docházka GIRITON* [online]. GIRITON Systems s.r.o. [cit. 25. 2. 2021]. Dostupné z: https://giriton.com/cs

[4] *Bezkontaktní teploměr* [online]. GIRITON Systems s.r.o. [cit. 25. 2. 2021]. Dostupné z: https://giriton.com/cs/attendanceClock

[5] *Docházkový systém* [online]. TULIP Solutions CZ s.r.o. [cit. 24. 2. 2021]. Dostupné z: https://tulipize.cz/dochazkovy-system/

[6] *Docházkový systém iTA* [online]. ELEKON, s.r.o. [cit. 24. 2. 2021]. Dostupné z: https://www.firemnidochazka.cz/co-to-umi/

[7] *Docházkové terminály* [online]. ELEKON, s.r.o. [cit. 24. 2. 2021]. Dostupné z: https://www.firemnidochazka.cz/cenik

[8] *Docházkové terminály* [online]. ELEKON, s.r.o. [cit. 24. 2. 2021]. Dostupné z: https://www.firemnidochazka.cz/cenik

[9] *Docházkový systém Fingera* [online]. Innovatrics s.r.o [cit. 24. 2. 2021]. Dostupné z: https://www.fingera.com/cs/dochazkovy-system-fingera/

[10] *Aktion - mějte firmu pod palcem* [online]. EFG CZ spol. s.r.o. [cit. 24.

[11] BARNETT, Ryan. *Bezpečnostní chyby e-shopů: Chyby ve validaci vstupních dat – Cross Site Scripting* [online]. ESET software spol. s r.o. [cit. 2021-04-20]. Dostupné z: https://www.eset.com/cz/blog/prevence/bezpecnostni-chyby-e-shopu-chyby-ve-validaci-vstupnich-dat-cross-site-scripting/

[12] *Cross Site Request Forgery* [online]. ESET software spol. s.r.o. [cit. 2021-04-20]. Dostupné z: https://www.soom.cz/clanky/484--Cross-Site-Request-Forgery

[13] BARNETT, Ryan. *SQL Injection* [online]. 2011 [cit. 2021-04-20]. Dostupné z: http://projects.webappsec.org/w/page/13246963/SQL20Injection

[14] ROTH, Daniel, Rick ANDERSON a Shaun LUTTIN. *Introduction to ASP.NET Core* [online]. Microsoft, 2020-04-17 [cit. 2021-02-25]. Dostupné z: https://docs.microsoft.com/cs-cz/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0

[15] TROELSEN, Andrew a Phil JAPIKSE. *Pro C# 8 with .NET Core 3* SPRINGER, 2020. ISBN 978-1-4842-5755-5.

[16] *Overview of ASP.NET Core Security* [online]. Microsoft, 2018-10-24 [cit. 2021-02-25]. Dostupné z: https://docs.microsoft.com/en-us/aspnet/core/security/?view=aspnetcore-5.0

[17] *React* [online]. Microsoft, [cit. 2021-04-20]. Dostupné z: https://reactjs.org/

[18] *API Development for Everyone* [online]. Microsoft, [cit. 2021-04-21]. Dostupné z: https://swagger.io/

[19] *MATERIAL-UI* [online]. Microsoft, [cit. 2021-05-04]. Dostupné z: https://material-ui.com/

[20] *What is Azure?* [online]. Microsoft, [cit. 2021-05-04]. Dostupné z: https://azure.microsoft.com/en-us/overview/what-is-azure/

[21] WALI, Mohamed. *Learn Microsoft Azure: Build, manage, and scale cloud applications using the Azure ecosystem.* Packt Publishing, 2018. ISBN 978-1789617580.

[22] *Configurations in Entity Framework Core* [online]. Microsoft, [cit. 2021-04-25]. Dostupné z: https://www.entityframeworktutorial.net/efcore/configuration-in-entity-framework-core.aspx

[23] PORTER, Layla. *User Secrets in a .NET Core Web App* [online]. [cit. 2021-04-24]. Dostupné z: https://www.twilio.com/blog/user-secrets-in-a-net-core-web-app-html

[24] *Introduction to JSON Web Tokens* [online]. [cit. 2021-04-24]. Dostupné z: https://jwt.io/introduction

[25] *Application Insights* [online]. [cit. 2021-05-02]. Dostupné z: https://docs.microsoft.com/cs-cz/azure/azure-monitor/app/app-insights-overview

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

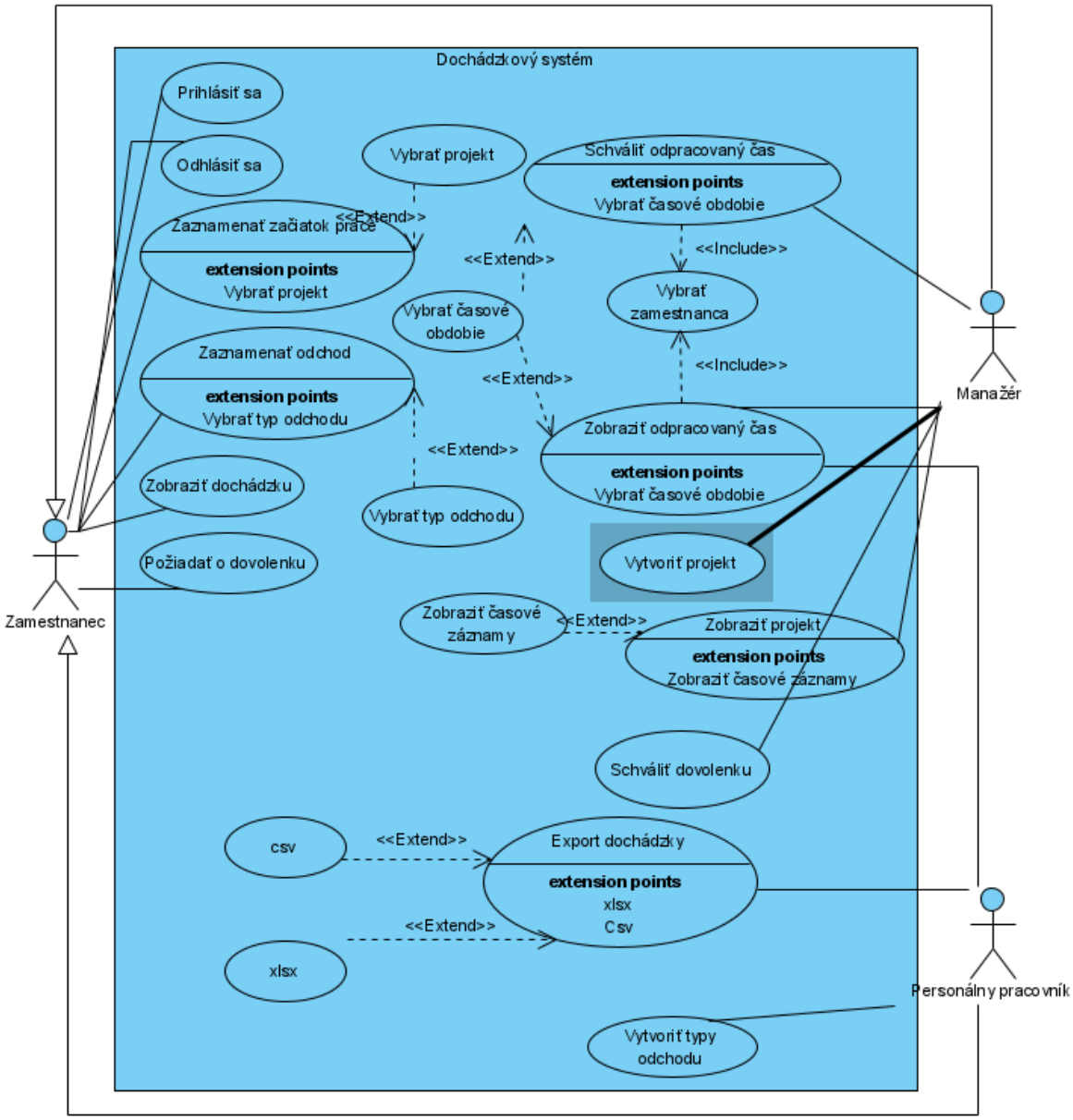
RFID	Radio Frequency Identification
NFC	Near Field Communication
GPS	Global Positioning System
GDPR	General Data Protection Regulation
LAN	Local Area Network
WiFi	Wireless Fidelity
LTE	Long Term Evolution
SQL	Structured Query Language
API	Application Programming Interface
Full-HD	Full High Definition
HTML	Hyper Text Markup Language
XSS	Cross-site scripting
XSRF/CSRF	Cross-site request forgery
REST	Representational State Transfer
IoT	Internet of Things
OOP	Object-Oriented Programming
HTTPS	Hypertext Transfer Protocol Secure
CORS	Cross-Origin Resource Sharing
ORM	Object–Relational Mapping
ERD	Entity Relationship Diagram
JWT	JSON Web Tokens
LINQ	Language Integrated Query

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

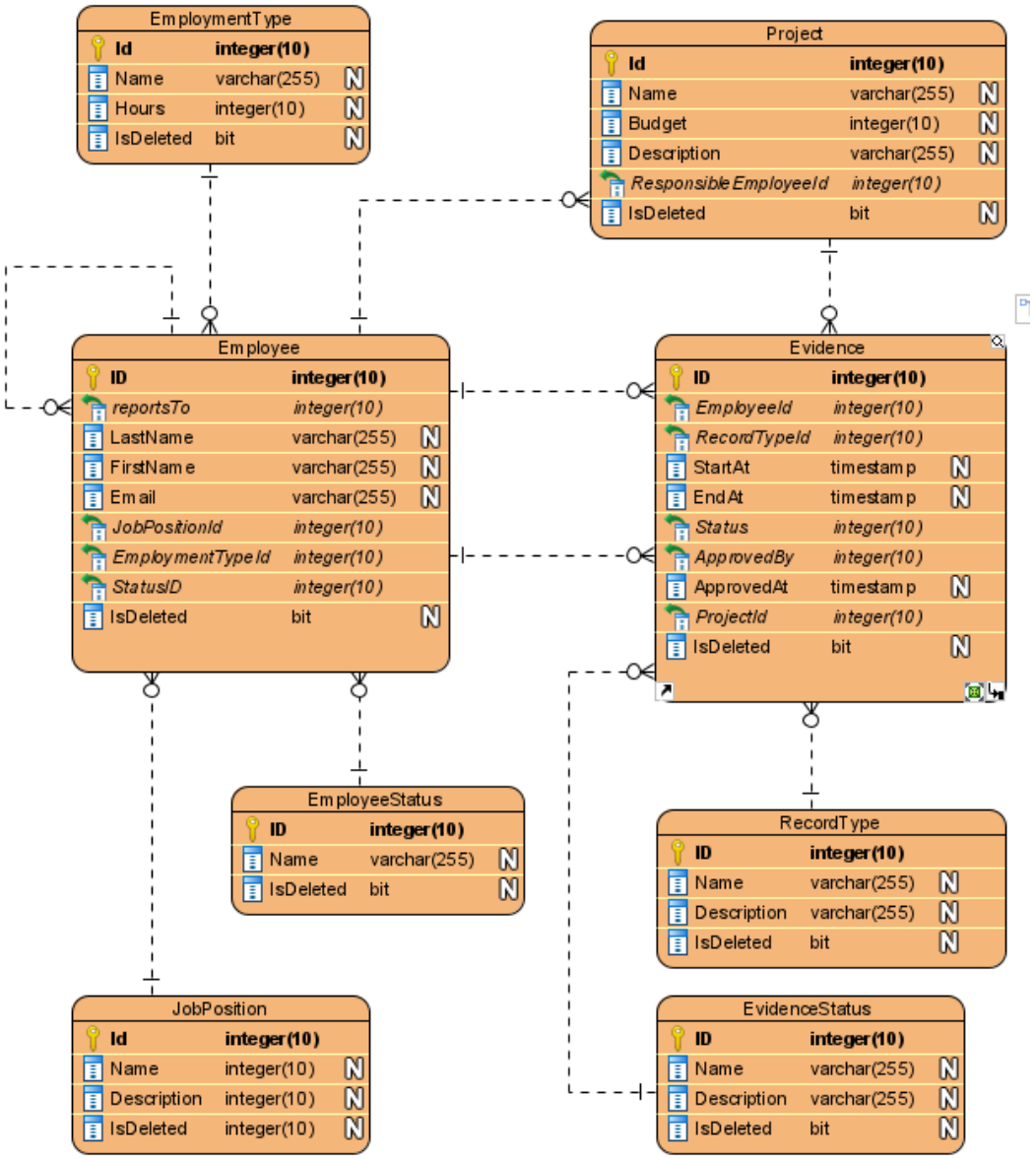
ZOZNAM PRÍLOH

PRÍLOHA P I. DIAGRAM PRÍPADOV POUŽITIA



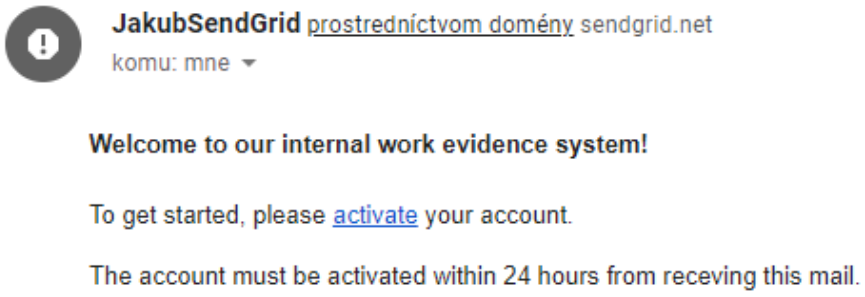
Obr. 1.1 Diagram prípadov použitia

PRÍLOHA P II. ERD DIAGRAM



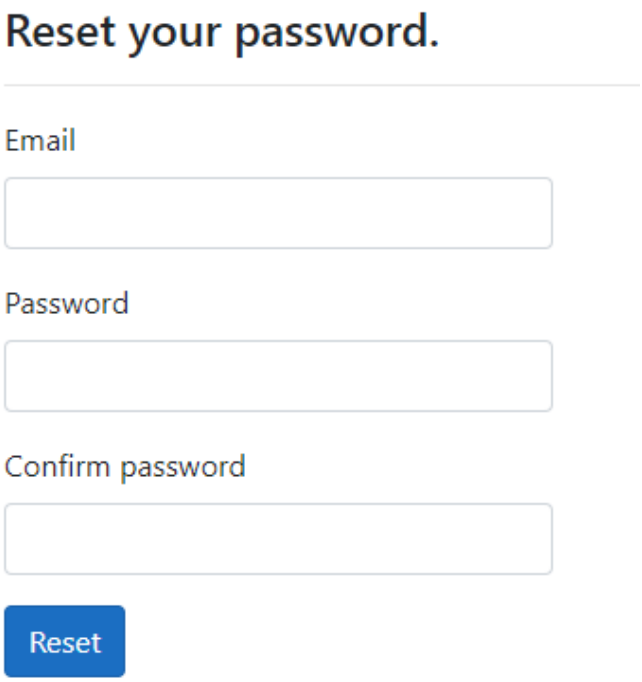
Obr. 2.1 ERD diagram

PRÍLOHA P III. UVÍTACÍ EMAIL



Obr. 3.1 Diagram prípadov použitia

PRÍLOHA P IV. RESET A NASTAVENIE PRVÉHO HESLA



Obr. 4.1 Formulár na resetovanie a nastavenie prvého hesla