TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: Informační technologie

Návrh informačního systému pro bezpečnostní agenturu

Design of Information System for Security Agency

Diplomová práce

Autor: Bc. **Martin Vitouš**

Vedoucí práce: doc. Ing. Jiřina Královcová, Ph.D.

Konzultant: Ing. Miloš Hernych

V Liberci 29. 5. 2009

Zadání

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum 29.5.2009

Podpis

Poděkování

Rád bych poděkoval vedoucí mé diplomové práce doc. Jiřině Královcové za vedení a za konzultace.

Abstrakt

Tato zpráva popisuje návrh informačního systému pro bezpečnostní agenturu. Primárním cílem práce bylo navrhnout protokol pro komunikaci mezi centrálou bezpečnostní agentury a bezpečnostními ústřednami nainstalovanými v jednotlivých hlídaných objektech. Sekundárním cílem bylo správně navrhnout informační systém podle metodiky popsané René Steinem, využívající UML diagramy a textové popisy k těmto diagramům. Součástí práce je i popis realizace celého informačního systému včetně komunikace s bezpečnostními ústřednami. Vlastní realizace již ovšem není součástí práce.

Celá zpráva je rozdělena do pěti kapitol. V první kapitole je popsána architektura celého informačního systému, který je pro větší přehlednost rozdělen na vnější a vnitřní část, každá z nich je zde popsána zvlášť. Ve druhé kapitole jsou popsány technologie použité pro návrh a realizaci informačního systému. Jsou to převážně technologie od společnosti Microsoft, neboť autor práce je certifikovaný pro vývoj aplikací za použití těchto technologií a má s nimi již několikaletou zkušenost. Ve třetí kapitole je obsažen návrh protokolu pro komunikaci mezi centrálou bezpečnostní agentury a bezpečnostními ústřednami. Jsou zde uvedeny požadavky na tuto komunikaci, popis komunikace, typy zpráv a schéma šifrování.

Ve čtvrté a zároveň nejdelší kapitole je popsán návrh informačního systému zpracovávajícího zprávy přijaté z bezpečnostních ústředen a poskytující další služby zákazníkům a pracovníkům bezpečnostní agentury. Nejprve jsou zde popsány jednotlivé role v systému, následně je zde diagram popisující jednotlivé případy užití v systému a podrobný popis těchto případů. Následují návrhy jednotlivých vrstev informačního systému, které podrobně popisují třídy použité pro zpracování dat a tabulky pro jejich uložení v databázi.

V poslední páté kapitole je popsána realizace jednotlivých části informačního systému. Dále je zde nastíněno použití informačního systému pro jednotlivé role uživatelů a jeho srovnání s návrhem.

Klíčová slova: informační systém, bezpečnostní agentura, business analýza, softwarový design

Abstract

This report describes a design of information system for security agency. The primary objective of this work was to design a protocol for communication between the central of the security agency and security board installed in the guarded premises. Secondary objective was to correctly design an information system according to the methodology described by René Stein, using UML diagrams and text descriptions of these diagrams. This work includes a description of the implementation of the entire information system, including communications with the security boards. The implementation is not part of the work.

Full report is divided into five chapters. In the first chapter there is a description of the architecture of the entire information system, which is for greater clarity divided into inner and outer part, each of which is described here separately. The second chapter describes techniques used for designing and implementing an information system. They are predominantly technologies from Microsoft, because the author of the work is certified for the development of applications using these technologies and has many years of experience with them. The third chapter contains the design of protocol for communication between the central of the security agency and security boards. There are listed the requirements for this communication, a description of communication, message types and encryption scheme.

The fourth and longest chapter describes the design of an information system processing messages received from the security boards and providing services for customers and employees of the security agency. At first, there is description of roles in the system and then there is a diagram describing the use cases in the system and a detailed description of these use cases. It is followed by design of various layers of information system, a detail description of the data-processing and data-storage in the database.

The last fifth chapter describes the implementation of each part of the information system. There are also outlined the uses of the information system for individual user roles compared with the design.

Keywords: information system, security agency, business analysis, software design

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Celá tato práce se zabývá problematikou návrhu informačního systému pro bezpečnostní agenturu. V tomto oboru je důležité znát v každém okamžiku stav hlídaných objektů. Ke zjišťování tohoto stavu musí bezpečnostní ústředny v jednotlivých hlídaných objektech komunikovat s centrálou.

Tato práce má dva hlavní cíle jedním z nich je navrhnout a realizovat komunikaci mezi centrálou a bezpečnostními ústřednami a druhým je podrobně navrhnout informační systém zpracovávající data z bezpečnostních ústředen a umožňující provádět další operace, potřebné k chodu bezpečnostní agentury.

První cíl byl stanoven na základě požadavků firmy A2D z Liberce. Ta požaduje navrhnout protokol pro komunikaci pomocí protokolu IP mezi centrálním počítačem s operačním systémem Windows a bezpečnostními ústřednami Siemens Sintony 60 – IC 60. Tato firma zajišťuje hardwarové napojení ústředen na Internet a implementaci navrženého protokolu do ústředen. Jednotlivé požadavky na komunikaci, stejně jako návrh samotné komunikace je uveden v kapitole 3.

Druhým a zároveň hlavním cílem této práce je navrhnout informační systém zpracovávající informace získané z bezpečnostních ústředen a jsou k němu přidány další požadavky, které by podle autora této práce mohla mít bezpečnostní agentura na svůj informační systém. Autor zde dává důraz hlavně na velmi podrobný návrh systému. Chce zde ukázat, jak by se správně měly navrhovat informační systémy. Postupuje přitom podle metodologie navržené René Steinem (Stein, 2009). Zatímco v praxi by se některé části tohoto návrhu konzultovali se zákazníkem, tato první verze systému zatím žádného zákazníka nemá, ale je možné, že ho bude mít v budoucnu a systém se bude dále vyvíjet podle jeho konkrétních požadavků.

Celý informační systém je v rámci této práce realizován, ale je zde počítáno s tím, že pokud by měl být nasazen v praxi, je potřeba provést ještě další úpravy jak v návrhu, tak i v realizaci. Tyto úpravy by již byly prováděny podle požadavků zákazníka.

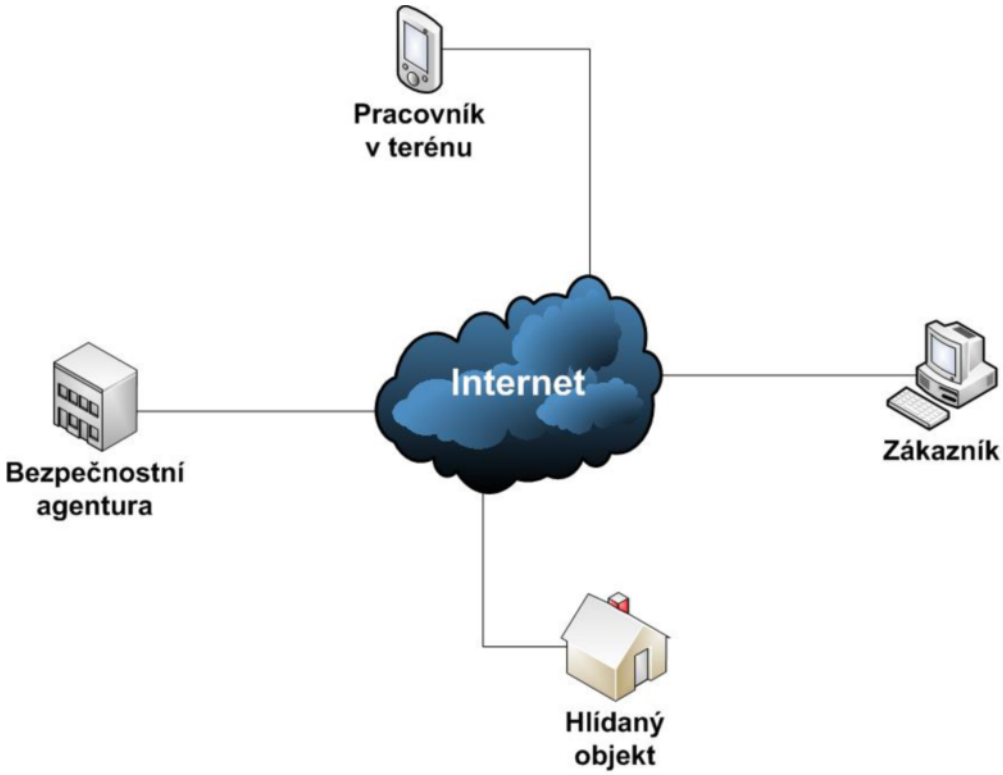
1. Architektura informačního systému

Architekturou informačního systému budeme chápat seznam a rozložení jednotlivých prvků systému a vazeb mezi nimi. Obvykle by sem patřily ještě hardwarové a softwarové požadavky na počítače v systému, ale to v našem případě zanedbáme.

Celý systém rozložíme na dvě části. Vnější částí budeme rozumět prvky mimo bezpečnostní agenturu a vnitřní částí prvky uvnitř bezpečnostní agentury. Zatímco ve vnitřní části jsou prvky propojeny pomocí vnitřní sítě (nebo i v rámci jednoho počítače), ve vnější části je spojovacím prvkem Internet, to se týká i připojení k vnitřní části.

1.1 Vnější část informačního systému

Vnější část informačního systému (viz Obrázek 1) obsahuje pracovníky v terénu, zákazníky a hlídané objekty. Na schématu je uvedena i bezpečnostní agentura, která reprezentuje vnitřní část informačního systému.



Obrázek 1 - Schéma vnější části informačního systému

Pracovník v terénu

Pod tímto pojmem myslíme techniky a ostrahu, kteří vyjíždějí do hlídaných objektů provádět činnosti jako instalaci a deinstalaci bezpečnostních ústředen, opravy poruch a zásahy při narušení bezpečnosti. Tito přistupují do informačního systému pomocí webového rozhraní na svých přenosných počítačích.

Zákazník

Pod tímto pojmem myslíme zákazníky, kteří se již registrovali do systému a také potenciální zákazníky, kteří se možná v budoucnu zaregistrují a budou využívat služeb bezpečnostní agentury. Obě tyto skupiny přistupují do systému přes webové rozhraní.

Hlídaný objekt

Tímto pojmem rozumíme budovu nebo část budovy, která je hlídána bezpečnostní agenturou, je zde nainstalována bezpečnostní ústředna a ta komunikuje s bezpečnostní agenturou pomocí pro tento účel navrženého protokolu (viz Komunikace mezi centrálou a bezpečnostními ústřednami).

1.2 Vnitřní část informačního systému

Vnitřní část informačního systému (viz Obrázek 2) obsahuje webovou aplikaci, databázi, Windows službu a pracovníky v bezpečnostní agentuře.

Webová aplikace

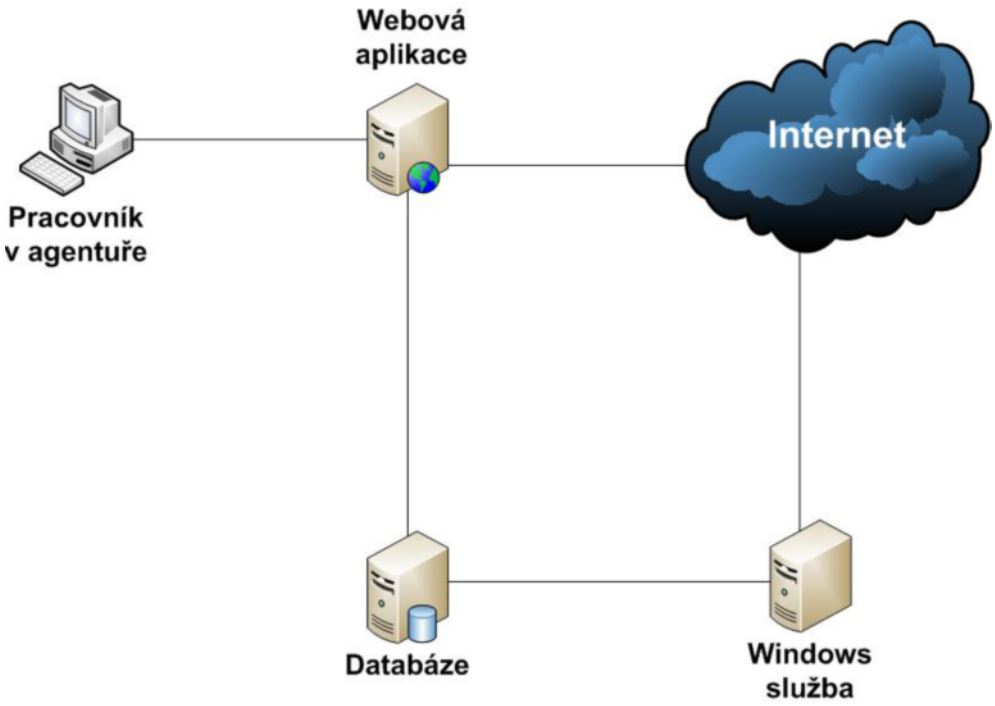
Komunikuje s jednotlivými uživateli informačního systému, informuje je na základě dat uložených v databázi a tato data podle jejich příkazů mění.

Databáze

Slouží jako úložiště dat pro informační systém, komunikuje s webovou aplikací a Windows službou

Windows služba

Komunikuje s bezpečnostními ústřednami a na základě této komunikace ukládá do databáze aktuální stav hlídaných objektů.



Obrázek 2 - Schéma vnitřní části informačního systému

Pracovník v agentuře

Pod tímto pojmem myslíme dispečery, kteří sledují stav hlídaných objektů a v případě potřeby přidělují objekty technikům nebo ostraze. Přistupují do informačního systému pomocí webového rozhraní.

2. Technologie pro návrh a realizaci informačního systému

Pokud chceme navrhovat a realizovat informační systém, musíme si uvědomit, jaké technologie k tomu budeme využívat, ať už se jedná o platformu operačního systému, běhové prostředí, webový a databázový server, vývojové prostředí, programovací jazyk a další používané technologie.

2.1 Platforma operačního systému – Microsoft Windows

Jádro informačního systému (webová aplikace, databáze a Windows služba) poběží na některém z řady operačních systémů Windows. Pravděpodobně to bude některá ze serverových verzí, např. Windows Server 2008. Může to ovšem být i některá klientská verze v případě, že bezpečnostní agentura nebude počítat s velkým množstvím klientů a nebude chtít velké počáteční investice. Uživatelé systému (pracovník v agentuře, pracovník v terénu a zákazník), přistupující do systému přes webové rozhraní mohou pracovat i na jiném operačním systému.

2.2 Běhové prostředí – Microsoft .NET Framework

V dnešní době je běžné, že aplikace potřebují ke svému spuštění nejen operační systém, ale také takzvané běhové prostředí, které aplikacím nabízí spouštěcí rozhraní a potřebné knihovny. V našem případě to bude Microsoft .NET Framework verze 3.5, které bude potřeba ke spuštění jak webové aplikace tak i Windows služby.

2.3 Webový server – Microsoft IIS (Internet Information Services)

Každá webová aplikace potřebuje ke svému běhu takzvaný webový server. Jedná se o aplikaci, která přijme požadavek od klienta pomocí http protokolu a na základě něj vybere příslušnou část webové aplikace (webovou stránku), zpracuje ji a výsledek opět pomocí http protokolu odesílá zpět klientovi. V našem případě použijeme jako webový server aplikaci Microsoft IIS verze 7, která je součástí Windows.

2.4 Databázový server – Microsoft SQL Server

Jelikož informační systém je aplikace většího rozsahu a může ji používat najednou více uživatelů, je nutné pro uchovávání dat místo prostého souboru použít databázi na databázovém serveru, který bude řídit přístup uživatelů k datům v databázi. V našem případě

použijeme Microsoft SQL Server 2008. Pro menší bezpečnostní agenturu postačí verze Express, která je zdarma, pro větší je možno použít například verzi Standard či Enterprise.

2.5 Vývojové prostředí – Microsoft Visual Studio

Abychom mohli naprogramovat informační systém, je nejlepší použít integrované vývojové prostředí obsahující textový editor s množstvím funkcí jako doplňování a zvýrazňování syntaxe, kompilátor, debugger, správu projektů a další užitečné věci. Jelikož jsme se rozhodli pro tvorbu informačního systému postaveného na technologiích od společnosti Microsoft, bude nejvhodnější sáhnout i po vývojovém prostředí od stejné společnosti. Proto použijeme Microsoft Visual Studio 2008. I zde nejspíše sáhneme po verzi Express, která je zdarma, ale máme na výběr i z dalších verzí jako Standard, Professional nebo Team System.

2.6 Programovací jazyk – C#

Abychom mohli realizovat informační systém, musíme vybrat jeden nebo více programovacích jazyků, pomocí nichž tuto realizaci provedeme. Protože náš informační systém využívá .NET Framework, musíme zvolit jeden z jazyků, který je tímto prostředím podporován. Nejčastěji používané jsou C# a Visual Basic. Vybereme si první z nich ve verzi 3.0, která je přítomna ve vybrané verzi běhového prostředí.

2.7 Jazyk pro psaní webových stránek – ASP.NET

Protože náš informační systém má mít webové rozhraní, je nutné najít jazyk, ve kterém by bylo možno psát webové stránky. Nestačí nám klasické HTML, protože tyto stránky by měly být dynamické a vypisovat data podle požadavku uživatelů. Přímo se nám nabízí jazyk ASP.NET, který je součástí .NET Framework.

2.8 Struktura webové stránky – ASP.NET MVC (Model-View-Controler)

Máme již zvolen jazyk ASP.NET pro tvorbu webových stránek, ale nyní nám vyvstává dilema, jakou strukturu stránek použít, neboť tento jazyk nabízí dvě varianty. Starší WebForms připomínají tvorbu Windows aplikací, můžeme zde používat podobné ovládací prvky jako ve Windows. Nevýhodou této varianty je nutnost uchovávání stavu (kontextu). Proto zvolíme druhou variantu MVC neboli Model-View-Controler. Tento návrhový vzor je již starší, ale společnost Microsoft ho přidala do ASP.NET teprve v nedávné době.

Princip této struktury spočívá v tom, že existuje určité množství řadičů (controler), které mají určité metody. Uživatel namísto klasické webové adresy obsahující jméno souboru s webovou stránkou, zadá adresu obsahující název řadiče, název metody a případně parametry a systém předá řízení řadiči a ten provede požadovanou metodu, a pokud tato má nějaký výstup, zobrazí pohled (view). Pro práci s daty se zde používá takzvaný model, což může být skupina tříd obsahujících data.

2.9 Objektově-relační mapper – LINQ to SQL

Abychom v informačním systému nemuseli pracovat přímo s relačními daty, která jsou uložena v databázi, pomocí SQL příkazů, použijeme takzvaný objektově-relační mapper, který nám tabulky v databázi namapuje na datové objekty, a s těmi již můžeme normálně pracovat. V našem případě použijeme LINQ to SQL, protože je přímo součástí prostředí .NET Framework a je jednoduchý na použití. K získávání datových objektů z databáze používáme dotazovací jazyk LINQ, který je přímo integrovaný do jednotlivých jazyků v prostředí .NET Framework.

3. Komunikace mezi centrálou a bezpečnostními ústřednami

Důležitým prvkem celého informačního systému je komunikace mezi centrálou a bezpečnostními ústřednami. Pomocí této komunikace se pracovníci bezpečnostní agentury dozvídají aktuální stav bezpečnostních ústředen a celých hlídaných objektů.

3.1 Požadavky na komunikaci

Komunikace mezi centrálou a bezpečnostními ústřednami musí splňovat následující požadavky:

1) Musí být uskutečněna pomocí protokolu IP (bude vedena přes Internet).
2) Musí být při ní možné jednoznačně identifikovat odesilatele zpráv (aby se předešlo zahlcení centrály falešnými zprávami a jiným falešným vysíláním).
3) Musí využívat pouze jednoduché zprávy a šifrování (viz dále).

3.2 Návrh komunikace

Komunikace bude probíhat pomocí protokolu TCP, což je transportní protokol nad síťovým protokolem IP. Tento protokol používáme proto, že je nutné navázat spojení což pomocí protokolu UDP nelze. Komunikace bude vždy začínat na straně bezpečnostní ústředny. Ta bude znát veřejnou adresu centrály, ale sama mít veřejnou adresu nemusí. Celá komunikace probíhá ve 3 krocích:

1) Ústředna odešle svoji identifikaci centrále zašifrovanou společným klíčem.
2) Centrála určí totožnost ústředny a odešle ji jednorázový klíč pro zašifrování zprávy zašifrovaný specifickým klíčem ústředny.
3) Ústředna vytvoří zprávu a pro kontrolu k ní ještě přidá svou identifikaci, to celé zašifruje obdrženým jednorázovým klíčem a odešle centrále.

Přitom velikost jednotlivých klíčů je sedm bytů a velikost identifikace bezpečnostních ústředen je deset bytů.

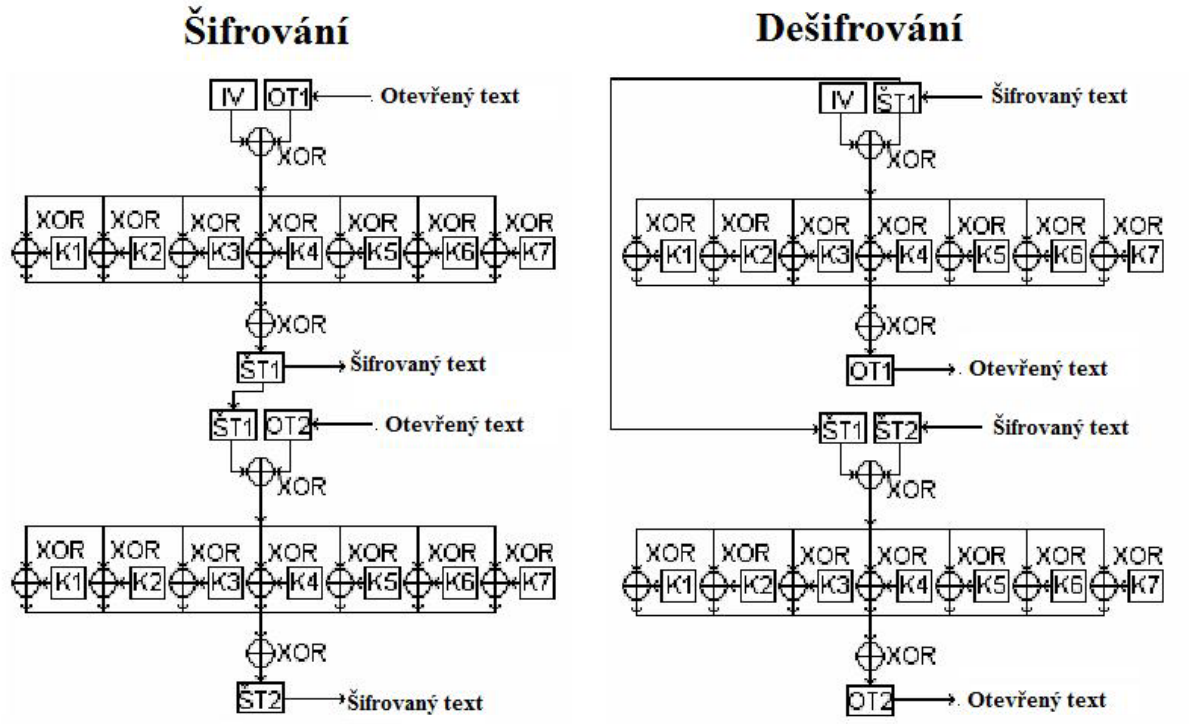
3.3 Typy zpráv

Bezpečnostní ústředna může poslat následující typy zpráv:

1) Vše v pořádku – bezpečnostní ústředna pracuje správně a nehlásí žádné narušení
2) Poplach – bezpečnostní ústředna hlásí narušení bezpečnosti
3) Porucha – bezpečnostní ústředna nepracuje správně

3.4 Použité šifrování

Pro šifrování bylo použito blokové schéma dle požadavků zadávající firmy (viz Obrázek 3) (Papouch s.r.o, 2004).



Obrázek 3 - Schéma šifrování a dešifrování

Jedná se o symetrickou šifru, která se řadí do skupiny blokových šifer. Jako předloha pro tvorbu šifrovacího algoritmu je použito blokové šifrování v módu CFB. Šifrování zprávy se provádí tak, že na začátku použijeme inicializační vektor (pevně zvolené číslo) k zašifrování (XOR) prvního bloku dat otevřeného textu. Výsledek tohoto šifrování se dále šifruje (XOR) s jednotlivými znaky klíče. Tím vznikne sedm různých mezivýsledků, které se šifrují (XOR) v jeden jediný. Výsledek se použije jako šifrovaná data a také k šifrování dalšího otevřeného textu. IV je inicializační vektor, OT1 a OT2 jsou první a druhý blok otevřeného textu, ŠT1 a ŠT2 jsou dva bloky šifrovaného textu; K1 – K7 jsou jednotlivé znaky klíče.

4. Návrh informačního systému

Pokud navrhujeme informační systém, je velmi důležité si podrobně rozmyslet, jaké všechny funkce má systém splňovat, jaké v něm budou role a co bude moci každá z rolí v systému dělat. Nejprve si navrhneme a popíšeme jednotlivé role v našem informačním systému. Dále si navrhneme diagram případů užití (neboli Use Case diagram), ve kterém jsou vidět role a co může která role v systému dělat. Následovat bude business analýza neboli podrobný návrh případů užití. Při návrhu reálného systému je tato analýza vypracovávána spolu s klientem a klient ji musí závazně odsouhlasit před dalším návrhem systému. V našem případě je klient pouze imaginární.

Pokud je celá business analýza vypracována, je možné přistoupit k návrhu vlastního informačního systému. Informační systém bude vícevrstvý a jeho jádrem je business vrstva, která se bude navrhovat jako první. Návrh business vrstvy neboli systémový design se skládá z diagramu tříd, stavového diagramu a podrobného popisu jednotlivých tříd.

Mezi vrstvy, které jsou pod business vrstvou a jejichž služby business vrstva využívá, jsou vrstva přístupu k datům a databázová vrstva. Databázovou vrstvu navrhneme pomocí ER (Entity – Relationship) diagramu neboli diagramu tabulek a vztahů mezi nimi. Ten vytvoříme na základě diagramu tříd. Jako vrstvu přístupu k datům použijeme LINQ to SQL, které převezme strukturu z ER diagramu.

Naopak vrstvy, které jsou nad business vrstvou a které používají služby business vrstvy, jsou vrstva aplikační a vrstva uživatelského rozhraní. Tyto vrstvy budeme navrhovat na základě Use Case diagramu a business analýzy. Jelikož náš systém bude vycházet z architektury MVC budeme jako modely brát třídy z business vrstvy, aplikační vrstva pak budou řadiče a vrstvou uživatelského rozhraní budeme rozumět jednotlivé pohledy.

4.1 Uživatelské role v systému

Potenciální zákazník

Tuto roli má zákazník bezpečnostní agentury, který se ještě nezaregistroval do systému. Po spuštění aplikace se jako jediný nepřihlašuje do systému. Jediné akce, které může v systému provádět, je získávání informací o bezpečnostní agentuře a registrace do systému.

Zákazník

Tuto roli má zákazník bezpečnostní agentury, který se již zaregistroval do systému. Po spuštění aplikace se musí přihlásit. V systému může získávat informace, editovat své osobní údaje, zobrazit si seznam vlastních hlídaných objektů, přidávat hlídané objekty, editovat hlídané objekty, rušit hlídané objekty.

Dispečer

Tato role představuje zaměstnance bezpečnostní agentury, který řídí konkrétní pobočku a má pod sebou techniky a ostrahu. Na každé pobočce je více dispečerů, ale ti se střídají ve směném provozu a vždy pouze jeden je přihlášený do systému. Jeho úkolem je reagovat na podněty přicházející buď od zákazníků, nebo od bezpečnostních ústředen a přidělovat objekty technikům a ostraze.

Technik

Tato role představuje zaměstnance bezpečnostní agentury, který instaluje a opravuje bezpečnostní ústředny a další zabezpečovací systémy v hlídaných objektech. Pokud je mu přidělen objekt k instalaci nebo opravě provede práci a v systému potvrdí splnění úkolu.

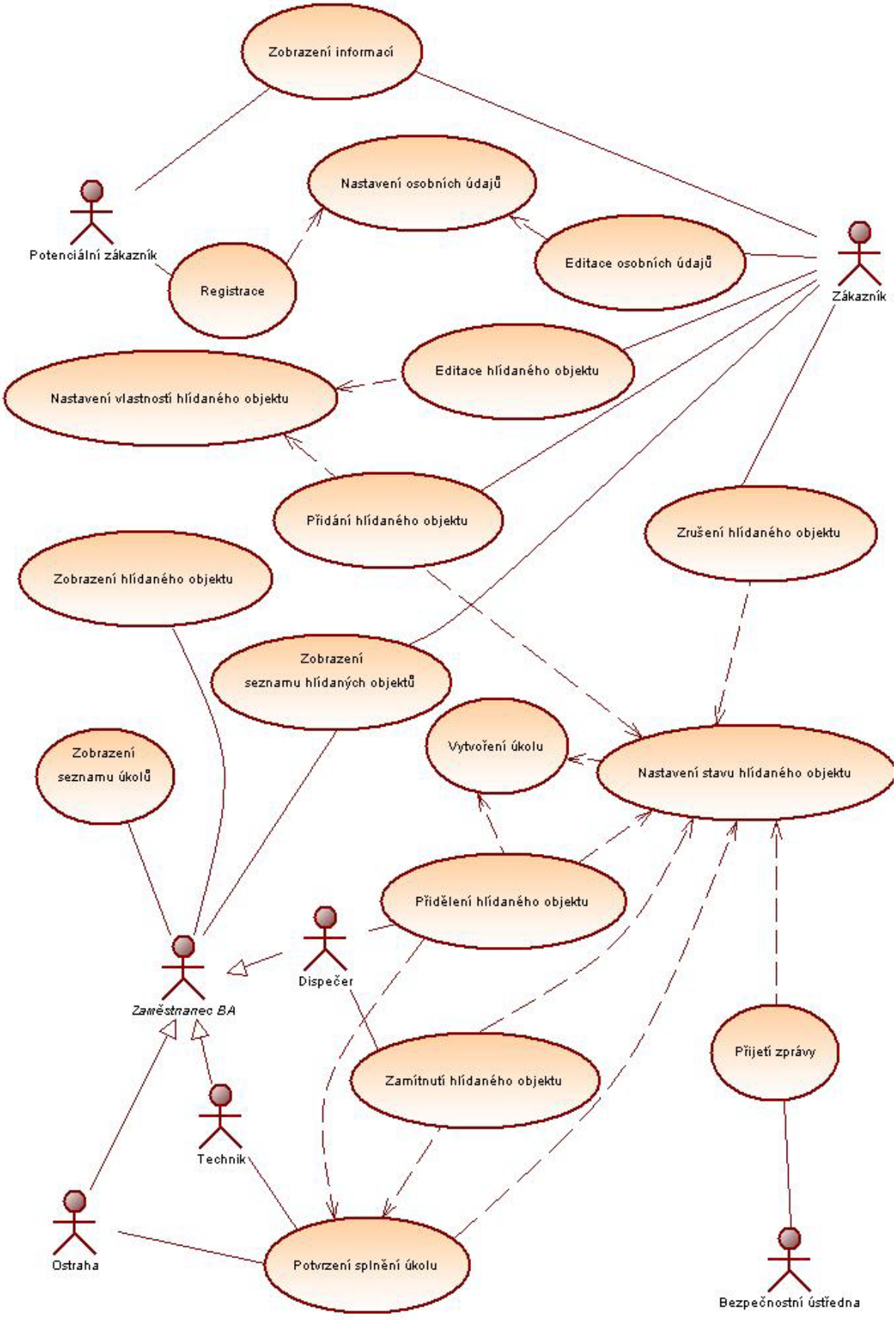
Ostraha

Tato role představuje zaměstnance bezpečnostní agentury, který provádí zásahy v hlídaných objektech, kde došlo k poplachu. Pokud je mu přidělen objekt, provede zásah a v systému potvrdí splnění úkolu.

4.2 Use Case diagram

Use Case diagram neboli diagram případů užití slouží k počátečnímu logickému utřídění množství požadavků zákazníka, které často bývají mlhavé a chaoticky vyslovované, na funkce, které by měl informační systém obsahovat. Tento diagram stejně jako následná business analýza jsou psány z pohledu zákazníka a nezabývají se technologickými aspekty řešení. Neřeší se zde technologická platforma ani použitá databáze. Pouze se zde zjišťuje, jaké procesy budou v systému probíhat a jací uživatelé ho budou používat.

Celý diagram se sestává z aktorů, případů užití a vazeb mezi nimi. Pod pojmem aktor nerozumíme konkrétního uživatele systému, ale roli v systému, kterou může hrát jedna nebo více osob případně i jiný systém. Případem užití potom myslíme jednu funkci systému. Tato funkce bývá obvykle přiřazena relací jednomu nebo více aktorům. Může se ovšem stát, že



Obrázek 4 - Use Case diagram neboli diagram případů užití

několik případů užití sdílí stejnou funkčnost, jinak řečeno zjistíme, že část scénáře se opakuje ve více případech užití. Společnou část můžeme vyjmout do samostatného případu užití a ostatním případům užití ho přidělíme relací, kterou značíme přerušovanou čarou.

Další věc, která se může v diagramu vyskytnout je generalizace neboli zobecnění a to jak aktorů tak případů užití. Tuto věc si můžeme představit jako dědičnost v objektovém programování. Předek má přiděleny nějaké případy užití a jeho potomci je mají automaticky také, aniž by bylo potřeba zapisovat tyto relace do diagramu.

V našem diagramu případů užití (Obrázek 4) vidíme Potencionálního zákazníka, který si může v systému zobrazovat informace nebo se registrovat. Dále je zde Zákazník, který může editovat osobní údaje, zobrazovat si seznam hlídaných objektů a také přidávat, editovat a rušit tyto objekty. Vidíme zde, že případy užití „Registrace“ a „Editace osobních údajů“ využívají společnou část „Nastavení osobních údajů“. Stejně tak „Přidání hlídaného objektu“ a „Editace hlídaného objektu“ využívají „Nastavení vlastností hlídaného objektu“.

Jelikož zaměstnanci bezpečnostní agentury v rolích Dispečer, Technik a Ostraha mají několik společných případů užití je v diagramu založen aktor „Zaměstnanec BA“, který může zobrazovat seznam hlídaných objektů, seznam úkolů a detaily hlídaného objektu. Aktor Dispečer pak dále může přidělovat a zamítat hlídaný objekt, zatímco aktoři Technik a Ostraha mohou potvrdit splnění svého úkolu. Je zde vidět, že při přidělení hlídaného objektu aktorem Dispečer se zároveň vytvoří úkol, potvrdí splnění úkolu a nastaví stav hlídaného objektu. Stav hlídaného objektu se nastavuje i při přidání hlídaného objektu, jeho zrušení, zamítnutí, potvrzení splnění úkolu a při přijetí zprávy.

4.3 Business analýza

Use Case 1 – Zobrazení informací

V případě požadavku na informace je potřeba tyto zobrazit ve snadno srozumitelné a dostupné formě. Potenciální zákazník musí mít možnost se dozvědět základní informace o firmě, o poskytovaných produktech včetně alespoň přibližné ceny, kontaktní informace, případně reference na významné zákazníky.

Tabulka 1 - Aktoři pro Use Case 1 – Zobrazení informací

Název aktora	Popis aktora
Potenciální zákazník	Aktor, který získává informace o firmě a na jejich základě se rozhoduje, zdali se zaregistruje a stane se zákazníkem firmy.
Zákazník	Aktor, který je již zaregistrovaný a pouze si rozšiřuje své informace o firmě.

Frekvence užití

Jednou za minutu

Základní tok událostí

1. Uživatel spustí aplikaci nebo dá příkaz k přechodu na úvodní obrazovku.
2. Systém zobrazí základní informace včetně názvu firmy, loga, stručného popisu firmy, novinek.
3. Uživatel zadá příkaz k zobrazení podrobnějších informací o firmě.
4. Systém zobrazí podrobnější informace včetně zaměření firmy, stručné historie firmy a dalších užitečných informací.
5. Uživatel zadá příkaz k zobrazení informací o produktech firmy.
6. Systém zobrazí informace o jednotlivých produktech včetně podrobného popisu produktu, ceny, dostupnosti produktu, akčních slev.
7. Uživatel zadá příkaz k zobrazení referencí.
8. Systém zobrazí reference na předchozí významné zákazníky včetně kontaktních údajů na příslušné zástupce zákazníků. Všechny informace zde budou zobrazeny až po odsouhlasení příslušnými zákazníky.
9. Uživatel zadá příkaz k zobrazení kontaktních informací.
10. Systém zobrazí kontaktní informace firmy včetně adresy provozovny, telefonů na jednotlivá pracoviště, faxů, emailových adres, mapky zobrazující polohu provozovny.

Poznámka: Uživatel může zadávat příkazy pro zobrazení informací v různém pořadí a systém mu zobrazuje odpovídající informace.

Use Case 2 – Nastavení osobních údajů

V případě že je v aplikaci potřeba nastavit osobní údaje zákazníka, je nutné získání potřebných informací od uživatele. V případě registrace (viz Use Case 3) je nutno získat všechny povinné údaje a dále ty, které jsou nepovinné, ale uživatel je chce uvést. V případě editace údajů (viz Use Case 4) je nutno získat pouze údaje, které chce uživatel pozměnit

oproti již uloženým. Všechny údaje vložené uživatelem je nutné zkontrolovat a případně uživatele informovat o špatně zadaných.

Tabulka 2 - Aktoři pro Use Case 2 – Nastavení osobních údajů

Název aktora	Popis aktora
Potenciální zákazník	Aktor, který zadává informace o sobě při registraci do systému.
Zákazník	Aktor, který pozměňuje již dříve zadané informace.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Systém zobrazí uživateli formulář pro zadání osobních údajů a předvyplní údaje již obsažené v systému (pouze v případě editace údajů).
2. Uživatel zadá své jméno a příjmení (povinné pole).
3. Uživatel zadá své přihlašovací jméno do systému (povinné pole).
4. Uživatel zadá a potvrdí své heslo do systému (povinné pole).
5. Uživatel zadá svou emailovou adresu (povinné pole).
6. Uživatel zadá svoje telefonní číslo (povinné pole).
7. Uživatel zadá svoje faxové číslo.
8. Uživatel zadá svou ulici a číslo popisné (povinné pole).
9. Uživatel zadá své město (povinné pole).
10. Uživatel zadá své poštovní směrovací číslo (povinné pole).
11. Uživatel vybere svůj kraj (povinné pole pouze pro české zákazníky). Zdrojem dat bude číselník krajů.
12. Uživatel vybere svůj stát (povinné pole). Zdrojem dat bude číselník států.
13. Uživatel zadá příkaz pro uložení údajů do systému
14. Systém údaje zkontroluje
15. Systém údaje uloží (v případě registrace založí nového zákazníka).

Rozšíření toku událostí

Zadávání hesla

Uživatel při zadávání hesla neuvidí znaky, které napíše. Celé heslo musí proto zadat ještě jednou a systém zkontroluje shodu. Při editaci se původní heslo nepředvyplňuje, ale pokud uživatel nezadá nové heslo, bude původní zachováno.

Kontrola údajů

Pokud systém najde ve vyplněných údajích chybu, upozorní na ni uživatele a vyzve ho, aby údaje opravil.

Use Case 3 – Registrace

Pokud se potenciální zákazník rozhodne stát zákazníkem firmy, je nutné, aby se zaregistroval do systému a vyplnil o sobě všechny potřebné informace (viz Use Case 2). Potom se již může přihlašovat do systému v roli zákazníka a může provádět všechny operace jako zákazník.

Tabulka 3 - Aktoři pro Use Case 3 – Registrace

Název aktora	Popis aktora
Potenciální zákazník	Aktor, který se registruje do systému.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Uživatel zadá příkaz pro zaregistrování do systému.
2. Uživatel zadá osobní údaje (viz Use Case 2).
3. Systém oznámí uživateli, že byl úspěšně zaregistrován
4. Uživatel je přihlášen do systému jako zákazník.

Use Case 4 – Editace osobních údajů

V případě, že se změní některé informace o zákazníkovi, měl by co nejdříve opravit údaje uložené v systému, aby byla zachována správnost uložených údajů. Uživatel vyplní jednu nebo více informací o sobě, které se změnily (viz Use Case 2) a systém změny uloží.

Tabulka 4 – Aktoři pro Use Case 4 – Editace osobních údajů

Název aktora	Popis aktora
Zákazník	Aktor, jehož osobní údaje se změnily a on tuto změnu potřebuje uložit do systému.

Frekvence užití

Jednou za deset minut

Základní tok událostí

1. Uživatel zadá příkaz pro změnu osobních údajů.
2. Uživatel zadá osobní údaje (viz Use Case 2).

3. Systém oznámí uživateli, že údaje byly úspěšně změněny.

Use Case 5 – Zobrazení seznamu hlídaných objektů

Každý uživatel si může zobrazit seznam hlídaných objektů, pro které má oprávnění. Zobrazované objekty mohou být různě filtrovány. Uživatelé zde mohou zadávat různé příkazy pro práci s hlídanými objekty. Možnosti použití filtrů a operací s objekty jsou vázány na role v systému.

Tabulka 5 - Aktoři pro Use Case 5 – Zobrazení seznamu hlídaných objektů

Název aktora	Popis aktora
Zákazník	Aktor, kterému se zobrazí jím zadané hlídané objekty.
Dispečer	Aktor, kterému se zobrazí všechny objekty z jeho pobočky.
Technik	Aktor, kterému se zobrazí jemu přidělené objekty.
Ostraha	Aktor, kterému se zobrazí jemu přidělené objekty.

Frekvence užití

Každou vteřinu

Základní tok událostí

1. Uživatel zadá příkaz pro zobrazení seznamu hlídaných objektů.
2. Uživatel vybere pobočku (Zákazník).
3. Uživatel vybere stav hlídaného objektu (Zákazník, Dispečer, Technik, Ostraha).
4. Uživatel vybere, zda se mají zobrazit i zamítnuté objekty (Zákazník, Dispečer).
5. Uživatel zadá příkaz k filtrování.
6. Systém vypíše data podle určeného filtru.

Rozšíření toku událostí

Výběr pobočky

Systém nabídne uživateli pobočky uložené v systému a variantu „všechny“.

Výběr stavu hlídaného objektu

Systém nabídne uživateli možné stavy hlídaného objektu (viz Use Case 11).

Use Case 6 – Nastavení vlastností hlídaného objektu

V případě, že je v aplikaci potřeba nastavit vlastnosti hlídaného objektu, je nutné získání potřebných informací od uživatele. V případě přidání hlídaného objektu (viz Use Case 7) je nutno získat všechny povinné údaje a dále ty, které jsou nepovinné, ale uživatel je chce uvést.

V případě editace hlídaného objektu (viz Use Case 8) je nutno získat pouze údaje, které chce uživatel pozměnit oproti již uloženým. Všechny údaje vložené uživatelem je nutné zkontrolovat a případně uživatele informovat o chybném zadání.

Tabulka 6 - Aktoři pro Use Case 6 – Nastavení vlastností hlídaného objektu

Název aktora	Popis aktora
Zákazník	Aktor, který vyplňuje vlastnosti vlastního hlídaného objektu.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Systém zobrazí uživateli formulář pro zadání vlastností hlídaného objektu a předvyplní údaje již obsažené v systému (pouze v případě editace hlídaného objektu).
2. Uživatel zadá název objektu (povinné pole).
3. Uživatel vybere pobočku bezpečnostní agentury (povinné pole). Zdrojem dat bude seznam poboček načtený z databáze.
4. Uživatel zadá ulici a číslo popisné objektu (povinné pole).
5. Uživatel zadá město, kde se objekt nachází (povinné pole).
6. Uživatel zadá poštovní směrovací číslo objektu (povinné pole).
7. Uživatel vybere kraj, kde se objekt nachází (povinné pole pouze pro objekty v České republice). Zdrojem dat bude číselník krajů.
8. Uživatel vybere stát, kde se objekt nachází (povinné pole).
9. Uživatel zadá telefonní číslo objektu.
10. Uživatel zadá faxové číslo objektu.
11. Uživatel zadá emailovou adresu objektu.
12. Uživatel zadá příkaz pro uložení údajů do systému
13. Systém údaje zkontroluje
14. Systém údaje uloží (v případě přidání objektu vytvoří v systému nový objekt).

Use Case 7 – Přidání hlídaného objektu

Zákazník může do systému přidávat objekty, které požaduje střežit bezpečnostní agenturou. Pokud zákazník přidává nový objekt, musí o něm vyplnit všechny potřebné informace (viz Use Case 6).

Tabulka 7 - Aktoři pro Use Case 7 – Přidání hlídaného objektu

Název aktora	Popis aktora
Zákazník	Aktor, který přidává vlastní hlídaný objekt.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Uživatel zadá příkaz pro přidání hlídaného objektu.
2. Uživatel nastaví vlastnosti hlídaného objektu (viz Use Case 6).
3. Systém nastaví stav objektu na „Nový“ (viz Use Case 11)
4. Systém oznámí uživateli, že objekt byl úspěšně přidán.

Use Case 8 – Editace hlídaného objektu

V případě, že se změní některé informace o hlídaném objektu, měl by zákazník co nejdříve opravit vlastnosti uložené v systému, aby byla zachována správnost uložených vlastností. Zákazník vyplní jednu nebo více informací o hlídaném objektu, které se změnily (viz Use Case 6) a systém změny uloží. Pokud byl hlídaný objekt zamítnut dispečerem může ho zákazník editovat a tím ho znovu převést do stavu „Nový“.

Tabulka 8 - Aktoři pro Use Case 8 – Editace hlídaného objektu

Název aktora	Popis aktora
Zákazník	Aktor, jež vlastní hlídaný objekt, jehož vlastnosti se změnily a on tuto změnu potřebuje uložit do systému.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Uživatel zadá příkaz pro změnu vlastností hlídaného objektu.
2. Uživatel zadá vlastnosti hlídaného objektu (viz Use Case 6).
3. Systém nastaví stav objektu na „Nový“ (pokud byl ve stavu „Zamítnutý“) (viz Use Case 11)
4. Systém oznámí uživateli, že údaje byly úspěšně změněny.

Use Case 9 – Zrušení hlídaného objektu

Zákazník, který si objednal hlídání svého objektu, může toto hlídání zrušit. Objekt již dále nebude hlídán bezpečnostní agenturou a postupně přejde do neaktivního stavu.

Tabulka 9 - Aktoři pro Use Case 9 – Zrušení hlídaného objektu

Název aktora	Popis aktora
Zákazník	Aktor, který ruší vlastní hlídaný objekt.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Uživatel zadá příkaz pro zrušení hlídaného objektu.
2. Systém nastaví stav objektu na „Zrušený“ (viz Use Case 11)
3. Systém informuje uživatele o úspěšném zrušení objektu.

Use Case 10 – Zobrazení hlídaného objektu

Uživatel si může zobrazit informace o hlídaném objektu. Je zde zobrazen název objektu, adresa, telefon, email, fax, stav objektu. Pro techniky je zde zobrazen i kód hlídaného objektu a bezpečnostní klíč, tyto údaje jsou automaticky generovány systémem při přidání objektu a technik je musí nastavit na bezpečnostní ústředně při její instalaci v objektu.

Tabulka 10 - Aktoři pro Use Case 10 – Zobrazení hlídaného objektu

Název aktora	Popis aktora
Zákazník	Aktor, kterému se zobrazí jím zadaný hlídaný objekt.
Dispečer	Aktor, kterému se zobrazí objekt z jeho pobočky.
Technik	Aktor, kterému se zobrazí jemu přidělený objekt.
Ostraha	Aktor, kterému se zobrazí jemu přidělený objekt.

Frekvence užití

Každou vteřinu

Základní tok událostí

1. Uživatel zadá příkaz pro zobrazení hlídaného objektu.
2. Systém zobrazí informace o objektu.

Use Case 11 – Nastavení stavu hlídaného objektu

V systému má každý hlídaný objekt přidělen svůj stav a tento se v průběhu času mění. Mění ho systém jako reakci na prováděné akce (Přidání hlídaného objektu – viz Use Case 7, Zrušení hlídaného objektu – viz Use Case 9, Přidělení hlídaného objektu – viz Use Case 12, Zamítnutí hlídaného objektu – viz Use Case 13, Potvrzení splnění úkolu – viz Use Case 16, Přijetí zprávy – viz Use Case 17).

V tomto Use Case nejsou žádní aktoři, neboť je prováděn systémem automaticky.

Frekvence užití

Každou vteřinu

Základní tok událostí

1. Uživatel (Systém) zadá příkaz pro nastavení stavu hlídaného objektu včetně hodnoty nového stavu.
2. Systém změní stav hlídaného objektu.

Rozšířený tok událostí

Možné stavy hlídaného objektu

- Nový – hlídaný objekt byl zadán zákazníkem a nebyl doposud dále zpracován
- Instalace – některý z techniků byl určen pro instalaci zabezpečení hlídaného objektu
- Zabezpečený – zabezpečení objektu bylo nainstalováno a je v provozu
- Poplach – v objektu byl vyvolán poplach a ještě nebyl naplánován zásah
- Zásah – některý člen ostrahy byl určen k provedení zásahu v hlídaném objektu
- Porucha – zabezpečovací zařízení se neozývá nebo signalizuje poruchu a ještě nebyla naplánována oprava
- Oprava – některý z techniků byl určen pro opravu zabezpečení hlídaného objektu
- Zrušený – hlídaný objekt byl zrušený zákazníkem, ale ještě nebyla naplánována deinstalace zabezpečovacího zařízení
- Deinstalace – byla naplánována deinstalace zabezpečovacího zařízení v hlídaném objektu
- Neaktivní – objekt není již nadále hlídaný
- Zamítnutý – objekt zamítnutý dispečerem

Vytvoření úkolu

Pokud se objekt dostane do stavů „Nový“, „Poplach“, „Porucha“ nebo „Zrušený“ je vytvořen úkol pro dispečera (viz Use Case 15), který musí dále přidělit objekt technikovi nebo ostraze (viz Use Case 12).

Use Case 12 – Přidělení hlídaného objektu

Pokud je k hlídanému objektu vytvořen úkol pro dispečera, musí dispečer přidělit hlídaný objekt technikovi nebo ostraze. Tímto se vytvoří úkol (viz Use Case 15), který pověřený aktor poté musí splnit a potvrdit splnění(viz Use Case 16).

Tabulka 11 - Aktoři pro Use Case 12 – Přidělení hlídaného objektu

Název aktora	Popis aktora
Dispečer	Aktor, který přidělí hlídaný objekt technikovi nebo ostraze.

Frekvence užití

Jednou za minutu

Základní tok událostí

1. Uživatel zadá příkaz pro přidělení hlídaného objektu.
2. Uživatel vybere technika (ostrahu), kterému má být hlídaný objekt přidělen.
3. Uživatel potvrdí přidělení hlídaného objektu.
4. Systém vytvoří úkol vybraného technika (ostrahy) (viz Use Case 15).
5. Systém nastaví stav hlídaného objektu (viz Use Case 11).
6. Systém potvrdí splnění úkolu dispečera (viz Use Case 16).

Rozšířený tok událostí

Výběr technika (ostrahy)

Systém nabídne uživateli možné techniky (ostrahy) z jeho pobočky.

Use Case 13 – Zamítnutí hlídaného objektu

Pokud je nějaký problém s nově přidaným objektem dispečer ho může zamítnout a objekt pak není hlídaný bezpečnostní agenturou.

Tabulka 12 - Aktoři pro Use Case 13 – Zamítnutí hlídaného objektu

Název aktora	Popis aktora
Dispečer	Aktor, který zamítne hlídaný objekt.

Frekvence užití

Jednou za hodinu

Základní tok událostí

1. Uživatel zadá příkaz pro zamítnutí hlídaného objektu.
2. Systém nastaví stav hlídaného objektu na „Zamítnutý“ (viz Use Case 11).
3. Systém potvrdí splnění úkolu dispečera (viz Use Case 16).

Use Case 14 – Seznam úkolů

Zaměstnanci bezpečnostní agentury (Dispečer, Technik a Ostraha) si mohou nechat zobrazit seznam svých úkolů a to buď aktuálních anebo již splněných. Dispečer si zároveň může nechat zobrazit i úkoly ostatních zaměstnanců na pobočce.

Tabulka 13 - Aktoři pro Use Case 14 – Seznam úkolů

Název aktora	**Popis aktora**
Dispečer	Aktor, kterému se zobrazí všechny úkoly pro dispečery z jeho pobočky a také úkoly ostatních zaměstnanců na pobočce.
Technik	Aktor, kterému se zobrazí jemu přidělené úkoly.
Ostraha	Aktor, kterému se zobrazí jemu přidělené úkoly.

Frekvence užití

Každou vteřinu

Základní tok událostí

1. Uživatel vybere zaměstnance (Dispečer).
2. Uživatel vybere, zda chce zobrazit i splněné úkoly.
3. Uživatel zadá příkaz pro zobrazení seznamu úkolů.
4. Systém vypíše seznam úkolů.

Rozšíření toku událostí

Výběr zaměstnance

Systém nabídne uživateli zaměstnance pobočky uložené v systému a variantu „dispečeři“, která vypíše všechny úkoly pro dispečery, a variantu „všichni“, která reprezentuje úkoly pro všechny zaměstnance pobočky.

Use Case 15 – Vytvoření úkolu

Úkoly se v systému vytvářejí automaticky jako reakce na některé akce. Po změně stavu objektu (do stavů „Nový“, „Poplach“, „Porucha“, „Zrušený“ – viz Use Case 11) se vytvoří úkol pro Dispečery. Po přidělení hlídaného objektu (viz Use Case) se vytvoří úkol příslušnému Technikovi nebo Ostraze.

V tomto Use Case nejsou žádní aktoři, neboť je prováděn systémem automaticky.

Frekvence užití

Každou minutu

Základní tok událostí

1. Systém vybere zaměstnance.
2. Systém vybere pobočku.
3. Systém vybere objekt.
4. Systém vybere prioritu úkolu.
5. Systém vytvoří a uloží úkol.

Rozšíření toku událostí

Výběr zaměstnance

Systém buď vybere jednoho z techniků nebo z ostrahy nebo vybere „dispečery“ podle toho, na jakou akci reaguje.

Výběr objektu

Systém vybere objekt podle akce, na kterou reaguje.

Výběr priority

Pokud jde o reakci na změnu stavu na „Poplach“ nebo „Porucha“, případně nápravu těchto stavů („Zásah“, „Oprava“), dojde k výběru priority „Vysoká“, v ostatních případech dojde k výběru priority „Normální“.

Use Case 16 – Potvrzení splnění úkolu

Tento Use Case může být proveden aktérem nebo automaticky. K prvnímu případu dochází v situaci, kdy Technik nebo Ostraha splnili svůj přidělený úkol a chtějí tuto skutečnost zachytit v systému. Ke druhému případu dochází ve chvíli, kdy dispečer přidělí hlídaný objekt Technikovi nebo Ostraze a tím splní svůj úkol, což systém automaticky potvrdí.

Tabulka 14 - Aktoři pro Use Case 16 – Potvrzení splnění úkolu

Název aktora	Popis aktora
Technik	Aktor, který potvrzuje splnění jemu přidělených úkolů.
Ostraha	Aktor, který potvrzuje splnění jemu přidělených úkolů.

Frekvence užití

Každou minutu

Základní tok událostí

1. Uživatel zadá příkaz k potvrzení splnění úkolu.
2. Uživatel zapíše komentář ke splnění úkolu.
3. Uživatel potvrdí splnění úkolu.
4. Systém nastaví stav hlídaného objektu (viz Use Case 11)
5. Systém nastaví úkol jako splněný.

Use Case 17 – Přijetí zprávy

Pokud přijde zpráva z bezpečnostní ústředny a je ověřena její pravost je tato zpráva uložena do systému a podle jejího obsahu je nastaven stav příslušného hlídaného objektu (viz Use Case 11). Některé zprávy ovšem pouze potvrzují činnost ústředny a nemění stav příslušného hlídaného objektu.

Tabulka 15 - Aktoři pro Use Case 17 – Přijetí zprávy

Název aktora	Popis aktora
Bezpečnostní ústředna	Aktor, který vkládá zprávy do systému.

Frekvence užití

Každou vteřinu

Základní tok událostí

1. Uživatel zadá příkaz pro vložení zprávy do systému.
2. Systém uloží zprávu.
3. Systém nastaví podle potřeby stav hlídaného objektu (viz Use Case 11).

Rozšířený tok událostí

Možné typy zpráv

- Vše v pořádku – pravidelně odesílaná zpráva signalizující, že ústředna je v pořádku
- Poplach – zpráva odeslaná z ústředny signalizující poplach na ústředně

- Porucha – zpráva odeslaná z ústředny signalizující poruchu na ústředně
- Timeout – ústředna se neozývá, bráno jako porucha na ústředně

4.4 Systémový design – Diagram tříd

Diagram tříd (viz Obrázek 5) je použit k návrhu business vrstvy informačního systému a znázorňuje třídy a číselníky, ze kterých bude tato vrstva složena. Obvykle se do diagramu tříd kromě vlastních tříd a vazeb mezi nimi uvádějí i atributy a metody jednotlivých tříd. V našem případě jsou tyto z důvodu lepší čitelnosti diagramu vynechány a budou popsány až v další části.

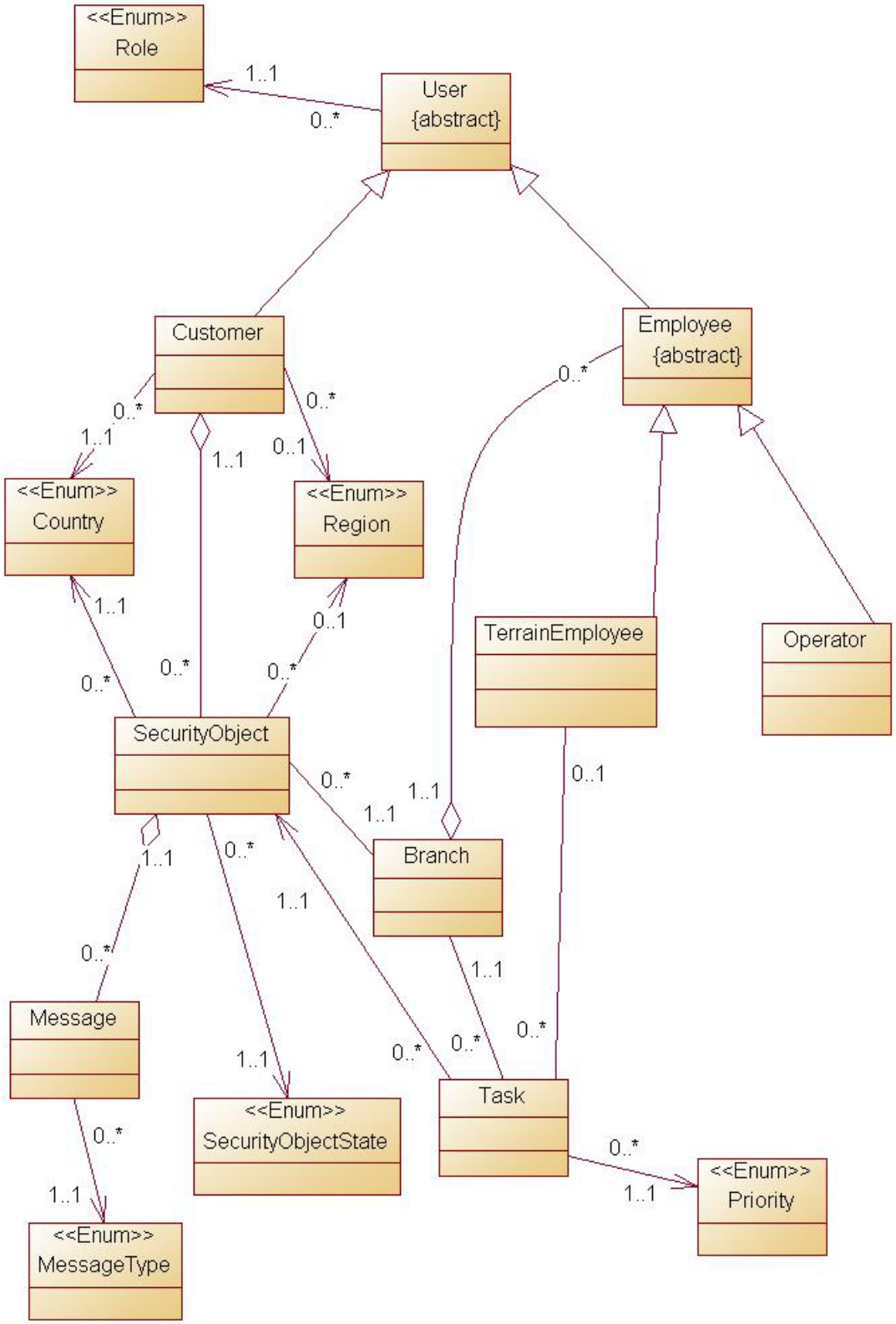
V diagramu tříd existuje několik typů vazeb, které jsou graficky rozlišené. První z nich je takzvaná generalizace neboli zobecnění, je znázorněna uzavřenou šipkou a představuje dědičnost mezi třídami. Dalším typem vazby je asociace, značená plnou čarou, která může a nemusí být ukončena otevřenou šipkou. Tato vazba značí obecný vztah mezi dvěma třídami a šipka značí, že vztah je pouze jednosměrný. Speciální formou asociace je agregace, značená kosočtvercem, která určuje celek a jeho části, mezi nimiž je ovšem volná vazba.

Uživatel informačního systému je reprezentován třídou **User**. Od této třídy dále dědí třídy **Customer**, představující Zákazníka, a **Employee**, představující Zaměstnance bezpečnostní agentury. Od třídy **Employee** dále dědí třídy **Operator**, představující Dispečera, a **TerrainEmployee**, představující Technika a Ostrahu. Zákazník má vazbu na hlídané objekty, které zadal do systému (třída **SecurityObject**). Zaměstnanec bezpečnostní agentury má vazbu na pobočku, ve které pracuje (třída **Branch**). Terénní zaměstnanci (Technik a Ostraha) mají vazbu na své úkoly (třída **Task**), které mají provést nebo které již provedli. Všechny úkoly mají vazbu na hlídaný objekt a pobočku, ale některé nemusí mít vazbu na terénní zaměstnance, toto jsou úkoly pro Dispečera dané pobočky. Hlídaný objekt má vazbu na zprávy přicházející z jeho bezpečnostní ústředny do systému (třída **Message**).

Třídy **User** a **Employee** jsou abstraktní a slouží pouze jako předci pro další třídy, tato skutečnost je v diagramu vyjádřena klíčovým slovem **abstrakt**.

Kromě tříd jsou v diagramu uvedeny i číselníky (označené klíčovým slovem **Enum**). Každý uživatel má přiřazenu svoji roli v systému (číselník **Role**). Každý zákazník a každý hlídaný objekt mají přiřazenou zemi (číselník **Country**). Na rozdíl od země, kraj (číselník **Region**) mají přiřazen pouze někteří zákazníci a některé hlídané objekty. Jsou to ti, kteří mají jako zemi přiřazenou Českou republiku. Každý hlídaný objekt má dále přiřazený svůj stav (číselník

SecurityObjectState). Každý úkol má přiřazenu svou prioritu (číselník **Priority**). Každá přijatá zpráva má přiřazen typ zprávy (číselník **MessageType**).



Obrázek 5 - Diagram tříd

4.5 Systémový design – Stavový diagram

Stavový diagram obsahuje stavový stroj (stavový automat), který vyjadřuje stavy určitého objektu a přechody mezi těmito stavy. Stav je situace, kdy modelovaný objekt splňuje nějakou podmínku, provádí nějakou operaci nebo čeká na událost. Přechod je spojení dvou stavů a znázorňuje přechod z jednoho stavu do druhého při splnění nějaké podmínky. Každý stavový stroj začíná počátečním stavem (značí se vyplněným kruhem) a končí jedním nebo více konečnými stavy (značí se dvojitým kruhem).

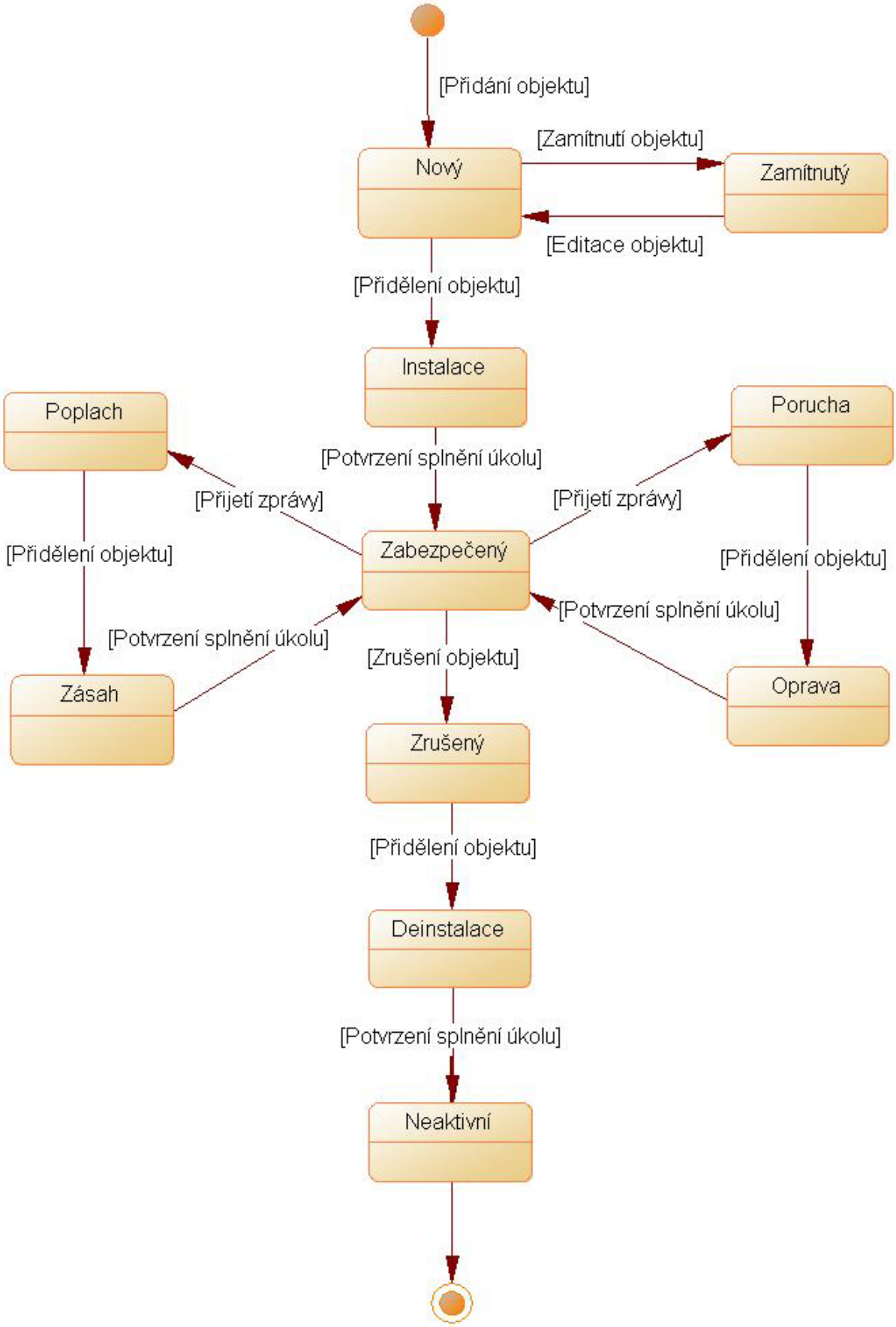
V našem případě se jedná o stavový diagram pro stavy hlídaného objektu (Obrázek 6). Popisy jednotlivých stavů jsou uvedeny u Use Case 11. Přechody mezi těmito stavy nastávají v době provádění funkcí systému (jejich názvy jsou uvedeny u jednotlivých přechodů), které volají funkci pro změnu stavu objektu.

Popíšeme si život jednoho konkrétního hlídaného objektu (do závorek budeme uvádět aktuální stav objektu). Nejprve zákazník přidá hlídaný objekt do systému („Nový“). Dispečer objekt zamítne („Zamítnutý“) a zákazník znovu edituje jeho obsah („Nový“). Dispečer již nemá námitky proti hlídanému objektu a přidělí ho technikovi („Instalace“). Technik, který dostal objekt přidělen, zde instaluje bezpečnostní ústřednu a poté potvrdí splnění přiděleného úkolu („Zabezpečený“). Nyní je objekt ve stavu, ve kterém se objekty obvykle nachází po většinu svého života.

V objektu dojde k narušení bezpečnosti a bezpečnostní ústředna odešle do systému zprávu „Poplach“, která je přijata („Poplach“). Dispečer zareaguje přidělením objektu ostraze („Zásah“). Ostraha zde provede zásah, zjistí příčinu narušení bezpečnosti a příslušně zareaguje. Následně potvrdí splnění přiděleného úkolu („Zabezpečený“).

V bezpečnostní ústředně dojde k poruše a ta odešle do systému zprávu „Porucha“ (případně přestane odesílat zprávu „Vše v pořádku“ a v systému se vygeneruje zpráva „Timeout“). Dispečer zareaguje přidělením objektu technikovi („Porucha“). Technik zde provede opravu a potvrdí splnění přiděleného úkolu („Zabezpečený“).

Zákazník se rozhodne, že již nechce svůj objekt nechat střežit bezpečnostní agenturou a zruší ho („Zrušený“). Dispečer zareaguje přidělením objektu technikovi („Deinstalace“). Technik zde provede deinstalaci bezpečnostní ústředny a potvrdí splnění přiděleného úkolu („Neaktivní“). Objekt je dále uchováván v systému pouze z důvodu archivace a žádný z uživatelů k němu nemá přístup.



Obrázek 6 - Stavový diagram

4.6 Systémový design – Textový popis tříd

Tato část systémového designu obsahuje podrobný popis tříd obsažených v diagramu tříd, včetně popisu veřejných atributů a metod tříd. Všechny třídy a číselníky business vrstvy jsou v namespace (jmenném prostoru) **BIS.Business**.

Abstraktní třída User

Tabulka 16 - Atributy třídy User

Název	Typ	Popis
Id	int?	Identifikační číslo uživatele v databázi
FullName	string	Jméno a příjmení
Username	string	Přihlašovací jméno
Password	string	Heslo
Email	string	Emailová adresa
Phone	string	Telefonní číslo
Role	Role	Role v systému

Tabulka 17 - Statické metody třídy User

Název	Popis
User GetUser(string username)	Vyhledá a vrátí uživatele daného uživatelského jména.
bool ValidateUser(string username, string password)	Ověří, zdali existuje uživatel s daným jménem a heslem.

Tabulka 18 - Abstraktní metody třídy User

Název	Popis
List<SecurityObject> GetSecurityObjectList()	Implementace této metody vrací seznam hlídaných objektů dostupných danému uživateli.

Tabulka 19 - Metody třídy User

Název	Popis
bool CheckUsername(string username)	Ověří, jestli uživatel může mít dané uživatelské jméno (při vytváření nebo změně).

Třída Customer (potomek třídy User)

Tabulka 20 - Atributy třídy Customer

Název	Typ	Popis
CustomerId	int?	Identifikační číslo zákazníka v databázi
Fax	string	Faxové číslo
Street	string	Ulice a číslo popisné
City	string	Město
Zip	string	Poštovní směrovací číslo
Region	Region?	Kraj
Country	Country	Stát

Tabulka 21 - Konstruktor třídy Customer

Název	Popis
Customer()	Vytvoří nového zákazníka.

Tabulka 22 - Implementace abstraktních metod třídy Customer

Název	Popis
List<SecurityObject> GetSecurityObjectList()	Vrací seznam hlídaných objektů vytvořených a patřících zákazníkovi.

Tabulka 23 - Metody třídy Customer

Název	Popis
void Save()	Uloží zákazníka do databáze (po vytvoření nebo editaci).

Abstraktní třída Employee (potomek třídy User)

Tabulka 24 - Atributy třídy Employee

Název	Typ	Popis
EmployeeId	int?	Identifikační číslo zaměstnance v databázi
Branch	Branch	Pobočka, kde zaměstnanec pracuje

Tabulka 25 - Abstraktní metody třídy Employee

Název	Popis
List<Task> GetTaskList()	Implementace této metody vrací seznam úkolů zaměstnance.

Třída Operator (potomek třídy Employee)

Tabulka 26 - Implementace abstraktních metod třídy Operator

Název	Popis
List<SecurityObject> GetSecurityObjectList()	Vrací seznam hlídaných objektů v dispečerově pobočce.
List<Task> GetTaskList()	Vrací seznam úkolů dispečera.

Třída TerrainEmployee (potomek třídy Employee)

Tabulka 27 - Implementace abstraktních metod třídy TerrainEmployee

Název	Popis
List<SecurityObject> GetSecurityObjectList()	Vrací seznam hlídaných objektů přidělených terénnímu pracovníkovi.
List<Task> GetTaskList()	Vrací seznam úkolů terénního pracovníka.

Třída SecurityObject

Tabulka 28 - Atributy třídy SecurityObject

Název	Typ	Popis
Id	int?	Identifikační číslo hlídaného objektu v databázi
Name	string	Název
Branch	Branch	Pobočka
Street	string	Ulice a číslo popisné
City	string	Město
Zip	string	Poštovní směrovací číslo
Region	Region?	Kraj
Country	Country	Stát
Phone	string	Telefonní číslo
Fax	string	Faxové číslo
Email	string	Emailová adresa
Code	string	Kód
SecureKey	string	Bezpečnostní klíč
State	SecurityObjectState?	Stav
CustomerId	Int	Identifikační číslo zákazníka v databázi

Tabulka 29 - Konstruktor třídy SecurityObject

Název	Popis
SecurityObject()	Vytvoří nový hlídaný objekt.

Tabulka 30 - Statické metody třídy SecurityObject

Název	Popis
SecurityObject GetSecurityObject(int id)	Vyhledá a vrátí hlídaný objekt s daným id.

Tabulka 31 - Metody třídy SecurityObject

Název	Popis
void Save()	Uloží hlídaný objekt do databáze.
void Cancel()	Zruší hlídaný objekt.
void Delegate(TerrainEmployee otherEmployee)	Přidělí hlídaný objekt terénnímu zaměstnanci.
void Refuse()	Zamítne hlídaný objekt.

Třída Branch

Tabulka 32 - Atributy třídy Branch

Název	Typ	Popis
Id	int	Identifikační číslo pobočky v databázi
Name	string	Název

Tabulka 33 - Statické metody třídy Branch

Název	Popis
List<Branch> GetBranches()	Vrátí seznam všech poboček v systému.

Tabulka 34 - Metody třídy Branch

Název	Popis
List<TerrainEmployee> GetTerrainEmployees()	Vrátí seznam všech terénních zaměstnanců na pobočce.
List<TerrainEmployee> GetTechnicians()	Vrátí seznam všech techniků na pobočce.
List<TerrainEmployee> GetSecurityGuards()	Vrátí seznam všech ostrah na pobočce.
List<Task> GetTaskList()	Vrátí seznam všech úkolů na pobočce.

Třída Task

Tabulka 35 - Atributy třídy Task

Název	Typ	Popis
Id	int	Identifikační číslo úkolu v databázi
SecurityObject	SecurityObject	Hlídaný objekt
Employee	Employee	Zaměstnanec
Comment	string	Komentář
IsActive	bool	Je úkol aktivní?
CreateTime	DateTime	Čas vytvoření
ConfirmTime	DateTime?	Čas potvrzení provedení
Priority	Priority	Priorita

Tabulka 36 - Statické metody třídy Task

Název	Popis
Task GetTask(int taskId)	Vyhledá a vrátí úkol s daným id.

Tabulka 37 - Metody třídy Task

Název	Popis
void Confirm(string comment)	Potvrdí splnění úkolu.

Třída Message

Tabulka 38 - Atributy třídy Message

Název	Typ	Popis
SecurityObject	SecurityObject	Hlídaný objekt, ze kterého přišla zpráva
Text	string	Text
Type	MessageType?	Typ
EncodedRandomKey	byte[]	Náhodný klíč zašifrovaný klíčem hlídaného objektu
IsActive	bool	Zpráva ještě nebyla zpracována a uložena
TimeInterval	int	Čas (v ms) mezi voláními časovače

Tabulka 39 - Konstruktor třídy Message

Název	Popis
Message(byte[] message)	Vytvoří nový objekt pro přijetí zprávy a zpracuje data přijatá z ústředny v 1. kroku komunikace.

Tabulka 40 - Statické metody třídy Message

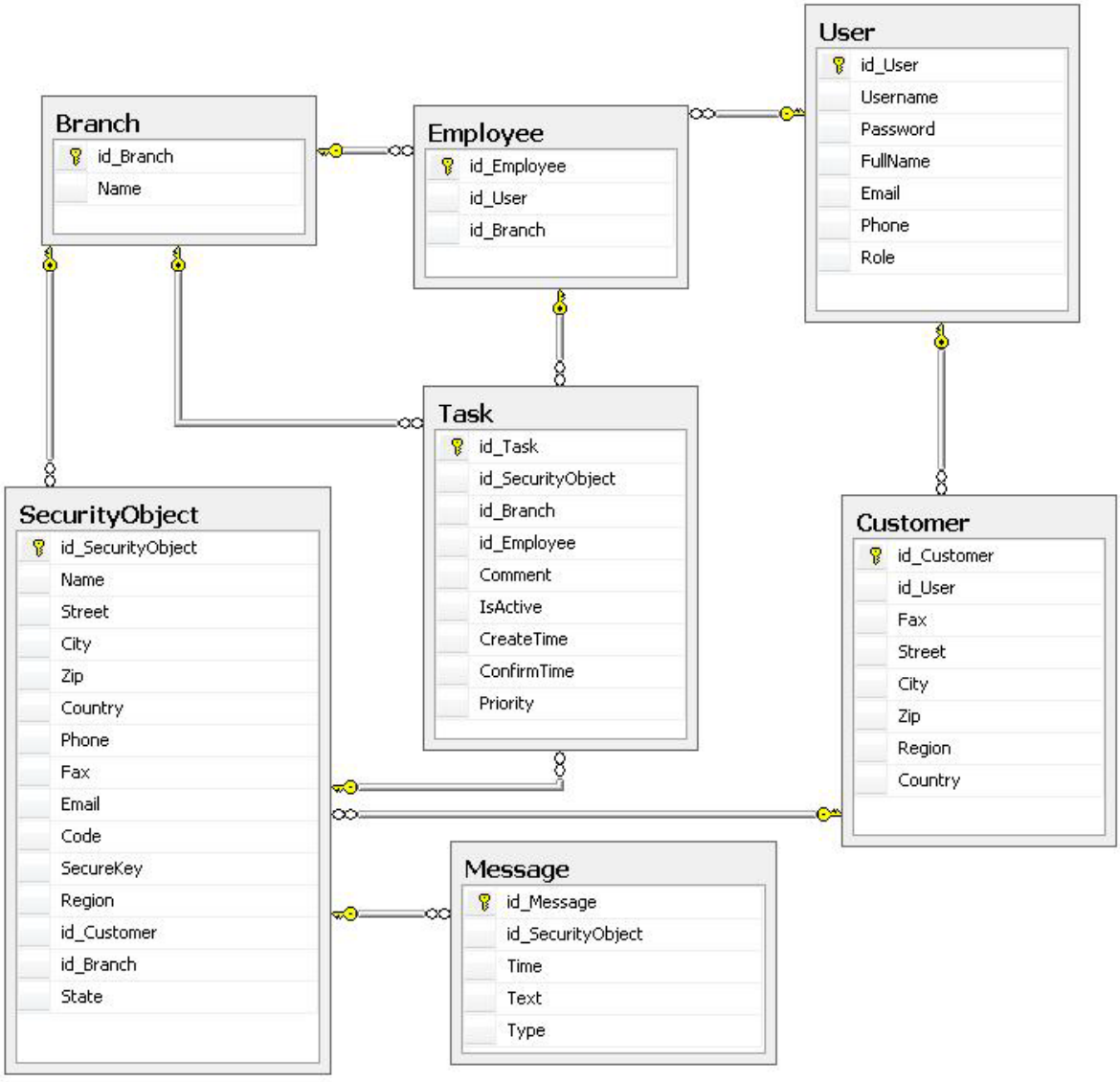
Název	Popis
void OnTimer(object o, ElapsedEventArgs ea)	Vyhledává ústředny, které se dlouho neozvaly.

Tabulka 41 - Metody třídy Message

Název	Popis
void Receive(byte[] message)	Zpracuje data přijatá z ústředny ve 3. kroku.

4.7 Návrh databázové vrstvy – ER diagram

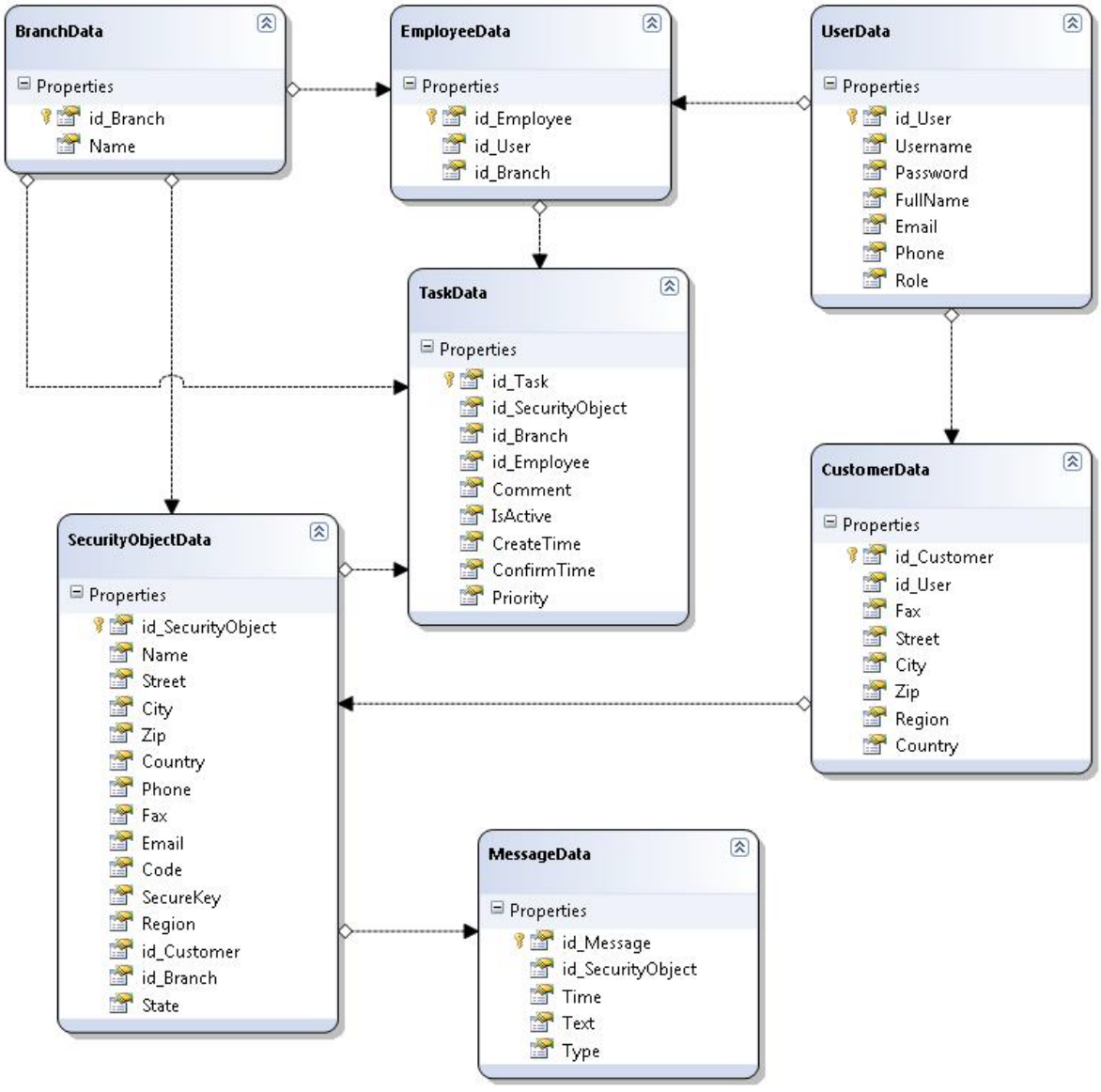
ER diagram (Entity-relationship, viz Obrázek 7) znázorňuje tabulky v databázi a vztahy mezi nimi. Jsou zde zobrazeny primární klíče jednotlivých tabulek. Jak je vidět tabulky v databázi odpovídají třídám v business vrstvě, neboť slouží k ukládání dat z business objektů. Tento diagram byl vytvořen v programu Microsoft SQL Server Management Studio a při jeho vytváření se přímo vytvořily i tabulky v databázi na Microsoft SQL Serveru.



Obrázek 7 - ER diagram

4.8 Návrh vrstvy pro přístup k datům

Jak již bylo řečeno, pro vrstvu pro přístup k datům použije náš informační systém LINQ to SQL, které mapuje tabulky v databázi na datové třídy. Diagram na obrázku (viz Obrázek 8) byl vytvořen v programu Microsoft Visual Studio 2008 a k jeho vytvoření postačí vybrání tříd z databáze a designer sám vytvoří příslušné třídy. Tyto třídy budou dále použity při implementaci business vrstvy pro načítání dat z databáze pomocí dotazů LINQ.



Obrázek 8 - Návrh LINQ to SQL

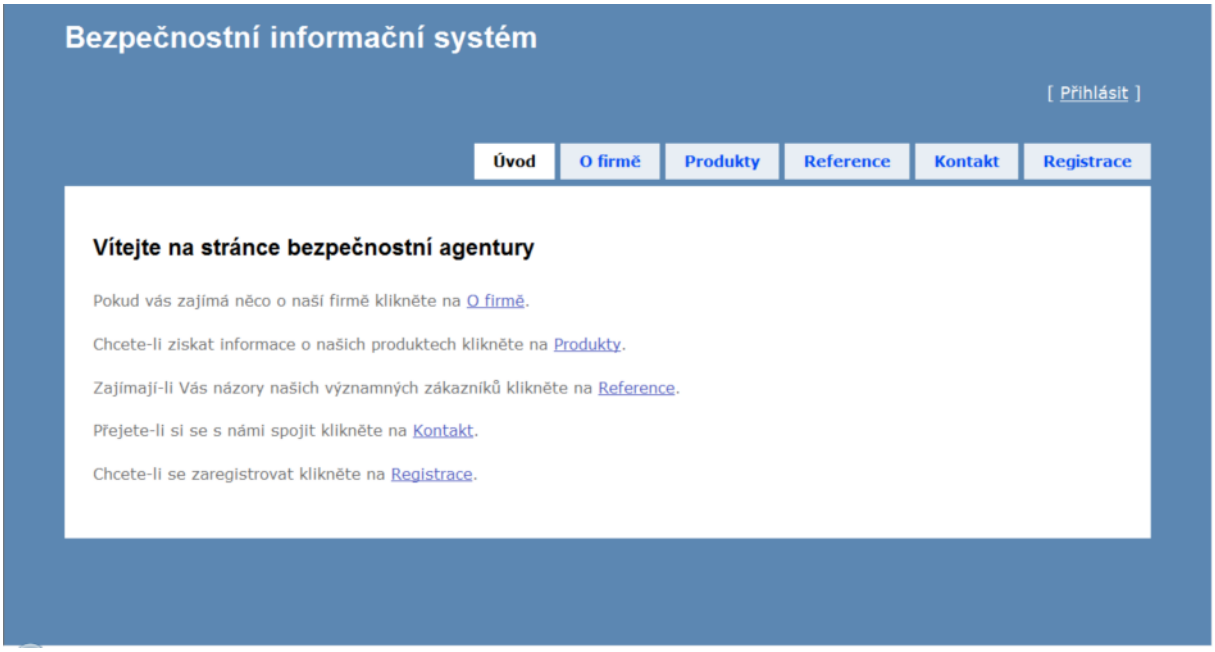
5. Realizace informačního systému

Informační systém byl realizován podle návrhů v předchozích kapitolách. Podle návrhu databázové vrstvy byla vytvořena databáze na databázovém serveru. Podle návrhu vrstvy pro přístup k datům vznikla dynamická knihovna. Ta je využívána další dynamickou knihovnou představující business vrstvu. Windows služba obsahuje TCP Server pro komunikaci s bezpečnostními ústřednami a využívá business vrstvu k šifrování a dešifrování zpráv a k ukládání dat do databáze.

Jedinou částí systému, která má uživatelské rozhraní je webová aplikace pro komunikaci informačního systému se zákazníky a pracovníky bezpečnostní agentury. Ta využívá business vrstvu pro práci s daty. Ukážeme si na ní, jak byly splněny požadavky na systém uvedené v business analýze.

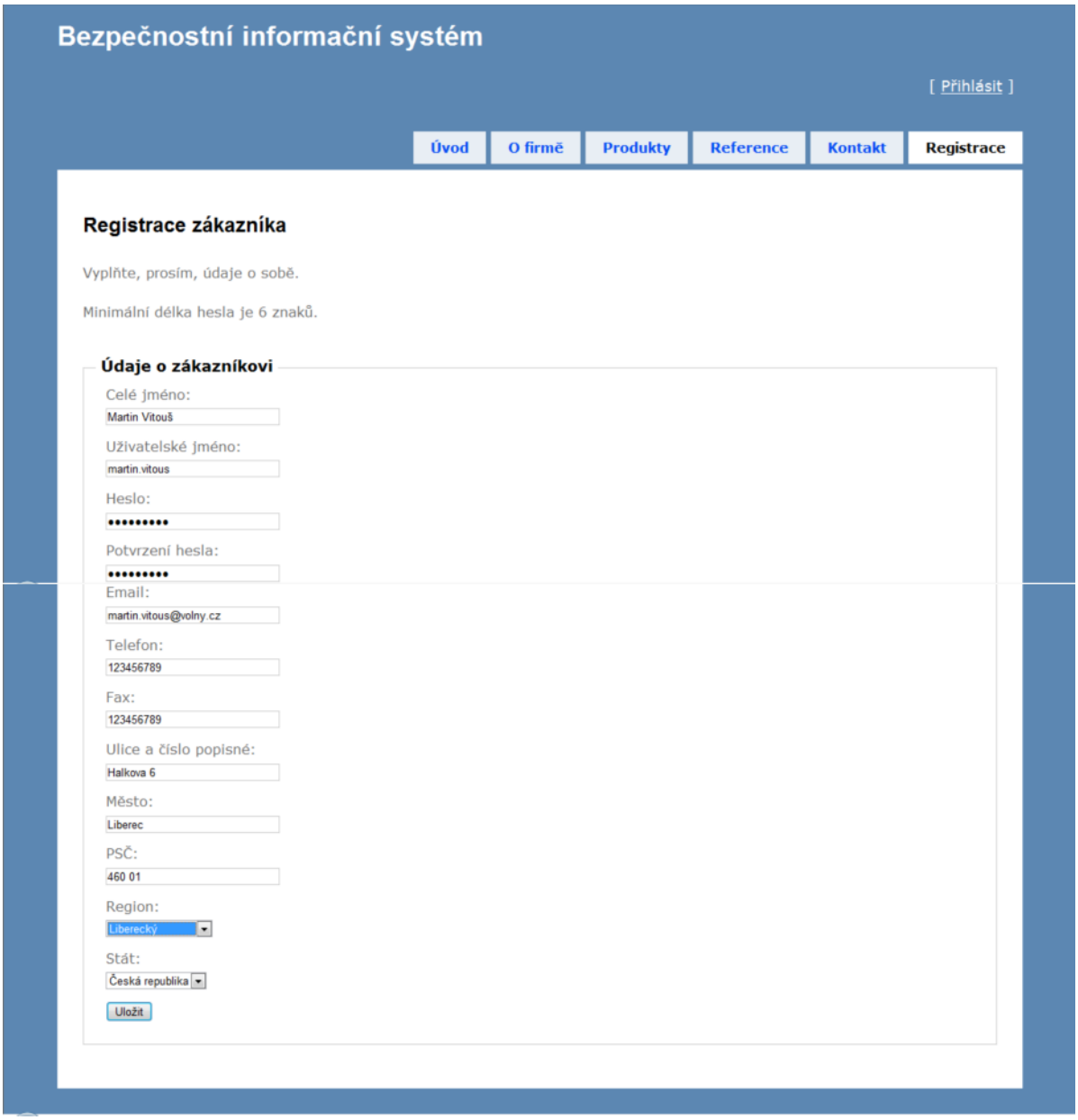
5.1 Použití systému pro roli Zákazník

Potenciální zákazník bezpečnostní agentury spustí webovou aplikaci. Zobrazí se mu úvodní stránka (Obrázek 9). Potenciální zákazník si zde může prohlédnout informace o bezpečnostní agentuře (Use Case 1).



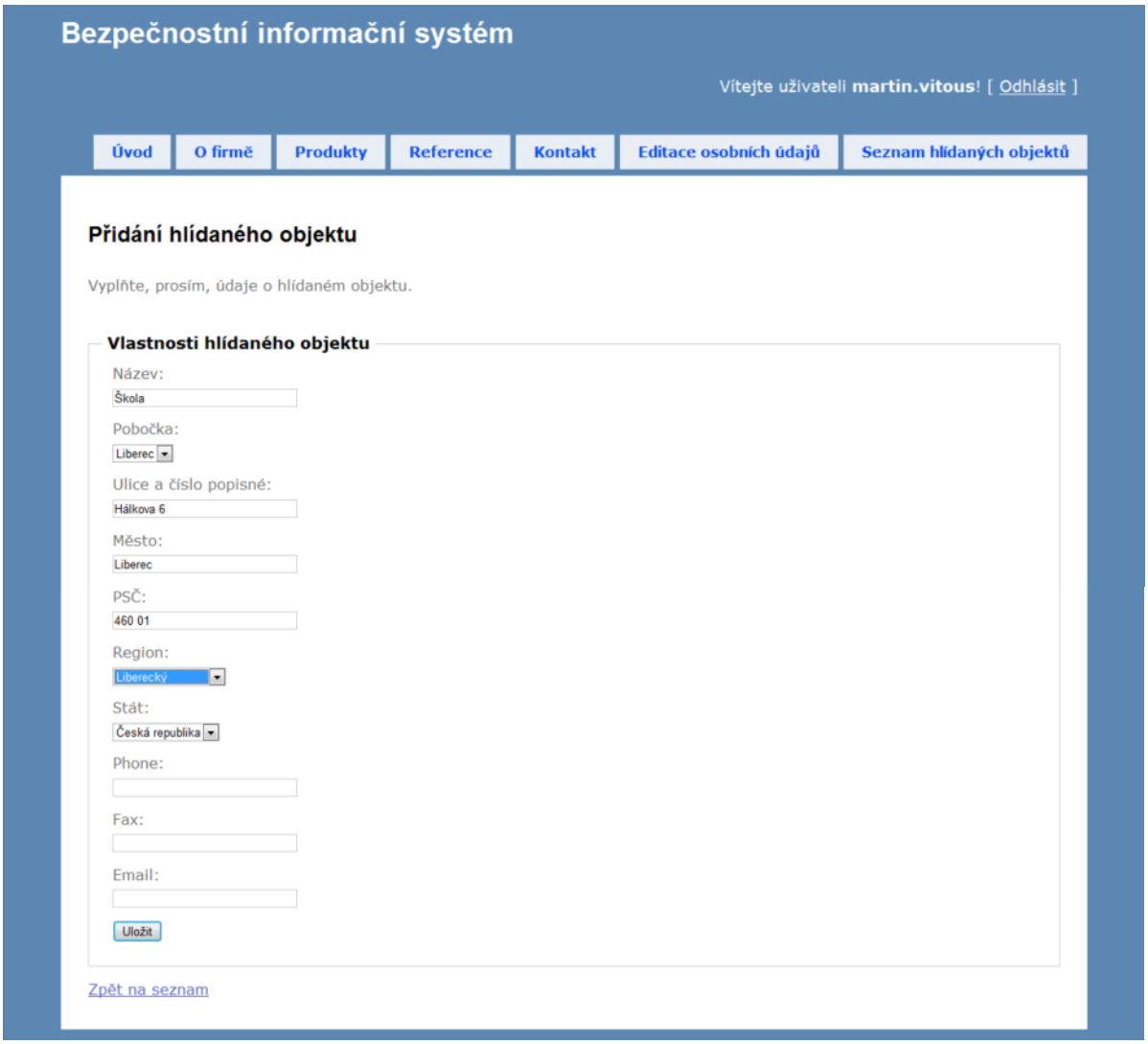
Obrázek 9 - Titulní stránka aplikace

Uživatel zadá pokyn k registraci do systému. Zobrazí se mu formulář pro registraci zákazníka (Obrázek 10), do kterého vyplní údaje o sobě.



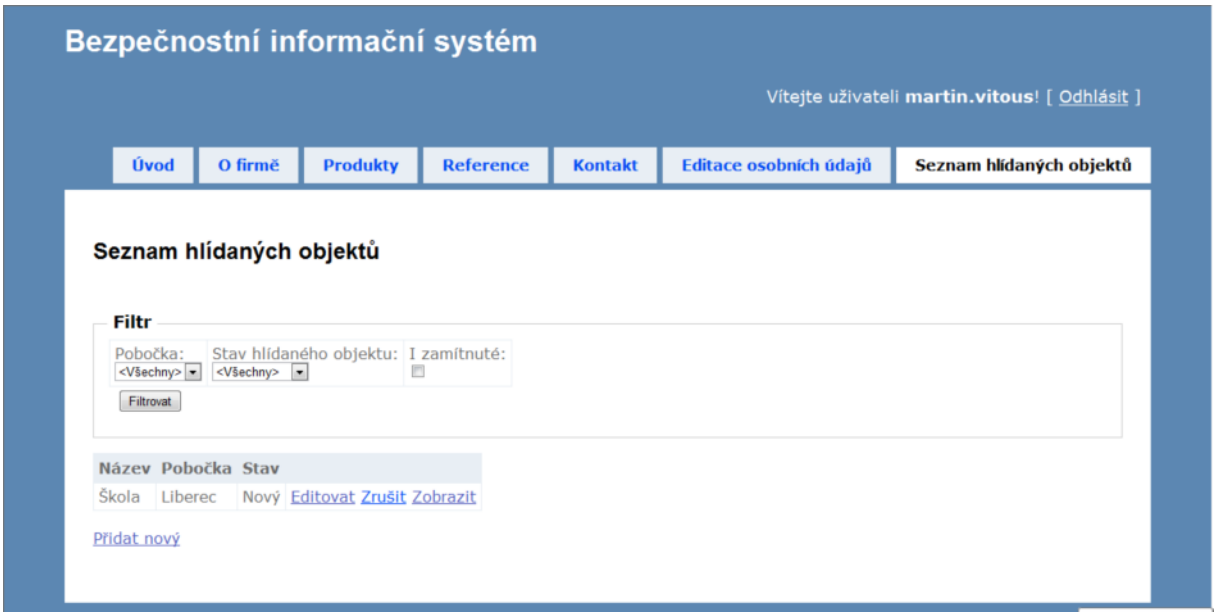
Obrázek 10 - Registrace zákazníka

Uživatel stiskne tlačítko „Uložit“. Systém zkontroluje údaje zadané uživatelem a vytvoří nový účet v systému (Use Case 2 a 3). Zároveň při tom uživatele přihlásí do systému. Po přihlášení do systému je uživatel již v roli Zákazník a přibudou mu v menu tlačítka „Editace osobních údajů“ a „Seznam hlídaných objektů“. Po stisknutí prvního tlačítka se zobrazí dialog pro editaci osobních údajů, který je obdobný jako dialog pro registraci zákazníka (Use Case 4), po stisku druhého tlačítka obdrží uživatel seznam hlídaných objektů, který je na začátku prázdný (Use Case 5). Zde stiskne tlačítko pro přidání nového hlídaného objektu. Zobrazí se formulář pro přidání hlídaného objektu (Obrázek 11) a uživatel vyplní jeho vlastnosti.



Obrázek 11 - Přidání hlídaného objektu

Uživatel stiskne tlačítko uložit. Systém zkontroluje zadané údaje a uloží hlídaný objekt do systému (Use Case 6 a 7). Potom zobrazí seznam hlídaných objektů i s nově přidaným (Obrázek 12). Uživatel si zde může hlídaný objekt editovat (Use Case 8), zrušit (Use Case 9) nebo zobrazit (Use Case 10). Editace hlídaného objektu vypadá stejně jako jeho přidání, zrušení hlídaného objektu se provede po stisku tlačítka a nezobrazuje se při tom žádný formulář a pro zobrazení hlídaného objektu je použit dialog (Obrázek 13).



Obrázek 12 - Seznam hlídaných objektů

Obrázek 13 - Zobrazení hlídaného objektu

5.2 Použití systému pro roli Dispečer

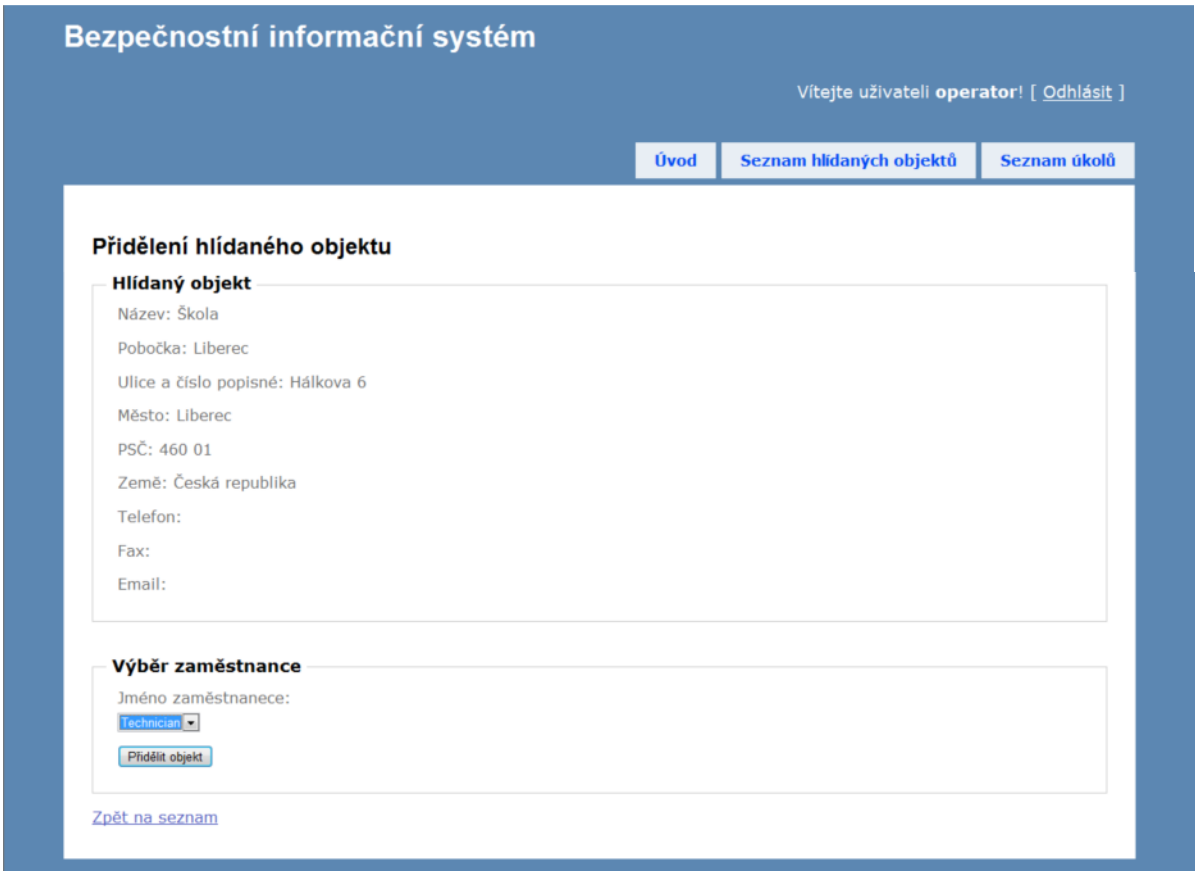
Zákazník přidal do systému nový hlídaný objekt, nyní přichází na scénu dispečer. Spustí si webovou aplikaci a stiskne tlačítko přihlásit, zobrazí se mu přihlašovací dialog, který vyplní (Obrázek 14). Po přihlášení do systému se mu zobrazí seznam úkolů, kde vidí nový hlídaný objekt (Obrázek 15).

Obrázek 14 - Přihlášení do systému

Obrázek 15 - Seznam úkolů

Nyní se může rozhodnout, jestli ho přidělí technikovi k instalaci bezpečnostní ústředny (Use Case 12) nebo ho zamítne (Use Case 13). Aby se mohl lépe rozhodnout, může si zobrazit seznam hlídaných objektů a v něm si najít příslušný objekt a zobrazit si údaje o něm. Nakonec

se rozhodne, že hlídaný objekt přidělí technikovi a stiskne příslušné tlačítko, zobrazí se mu formulář, kde vybere technika a stiskne tlačítko přidělit objekt (Obrázek 16).

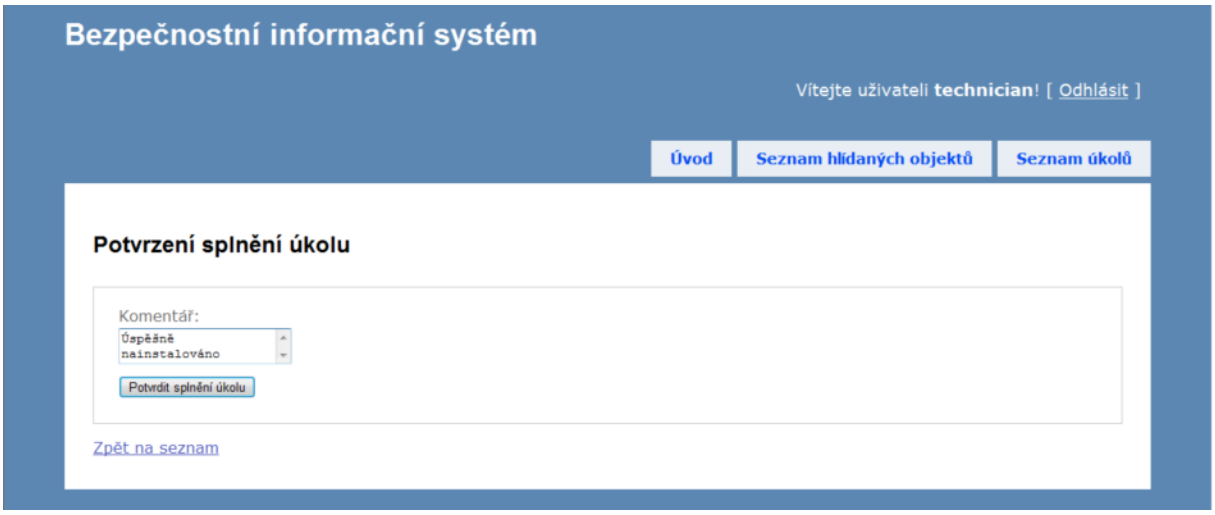


Obrázek 16 - Přidělení hlídaného objektu

5.3 Použití systému pro roli Technik (Ostraha)

Následující popis se bude týkat role Technik, ale role Ostraha může v systému používat stejné funkce.

Jak již víme, Zákazník vytvořil v systému nový objekt a Dispečer ho přidělil Technikovi. Technik se přihlásí do systému stejně jako předtím Dispečer a stejně jako on uvidí seznam úkolů, které na něj čekají. Vidí na něm, že musí provést instalaci bezpečnostní ústředny v nově přidaném objektu. Ústřednu namontuje a poté si znovu zobrazí seznam úkolů a klikne na tlačítko potvrdit splnění a na příslušném formuláři vyplní komentář (Obrázek 17)



Obrázek 17 - Potvrzení splnění úkolu

Závěr

Předložená práce se zabývá návrhem informačního systému pro bezpečnostní agenturu. Jeho hlavním cílem má být monitorování stavu hlídaných objektů. Celý tento text je členěn do několika částí. V první části, kterou představují první dvě kapitoly, je popsána architektura navrhovaného informačního systému a vypsány technologie použité při realizaci tohoto systému. V dalších dvou kapitolách je podrobně popsáno řešení hlavních cílů práce (body 3 a 4 zadání) a nakonec v poslední kapitole je nastíněna realizace systému.

Prvním cílem, který se podařilo splnit, byl návrh protokolu pro komunikaci mezi centrálou bezpečnostní agentury a bezpečnostními ústřednami nainstalovanými v hlídaných objektech. Tento protokol byl navržen přesně podle požadavků firmy A2D a bylo v něm použito dodané schéma pro šifrování (Papouch s.r.o, 2004). Následovala jeho implementace do Windows služby představující server a otestování proti testovacímu klientu.

Druhý významný cíl, který se podařilo v této práci splnit, byl podrobný návrh informačního systému. Hlavní důraz zde byl dán na business vrstvu aplikace, která je základem celého systému. Pro návrh této vrstvy byla použita metodika navržená René Steinem (Stein, 2009), jehož postupy byly použity i při vytváření jednotlivých UML diagramů (Stein, 2003-2005). Protože v této části nebylo nijak přesně specifikované zadání, celý informační systém byl navržen autorem práce tak, aby obsahoval funkce využitelné v bezpečnostní agentuře a zároveň rozsah jeho návrhu odpovídal předepsanému rozsahu této práce.

Zde se dostáváme ke sporným bodům této práce, hlavním sporným bodem je nedostatečná specifikace požadavků pro druhou a zároveň hlavní část práce ze strany potenciálního uživatele. Pokud bude v budoucnu informační systém nasazován do provozu, bude muset být tato část pečlivě konzultována s případným klientem a po konzultaci bude pravděpodobně nutné upravit jak návrh, tak i realizaci celého systému podle požadavků klienta.

Bibliografie

Microsoft. (nedatováno). *MSDN - Microsoft Developer Network*.

Papouch s.r.o. (2004). Ethernetový převodník EC485 - Katalogový list.

Stein, R. (2009). Materiály ze školení "Objektovými principy a návrhovými vzory řízený design a vývoj kvalitních aplikací 1".

Stein, R. (2003-2005). *Návrh aplikací v jazyce UML*. Načteno z interval.cz: http://interval.cz/serialy/navrh-aplikaci-v-jazyce-uml/