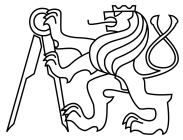
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Analýza a návrh změn pokladního systému na Koupališti Flošna
Student:	Lenka Stejskalová
Vedoucí:	Ing. Petra Pavlíčková, Ph.D.
Studijní program:	Informatika
Studijní obor:	Informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Zpracujte analýzu stávajícího systému, který se používá pro zákazníky na koupališti, zaměřte se hlavně na funkcionality s ohledem na bezpečnost.
Vypracujte optimalizovaný nový systém, jehož základním modulem bude automat pro dobíjení čipových hodinek a detailní návrh čtečky čipu na automatu, který obslouží velkou část zákazníků koupaliště bez nutnosti zásahu personálu.
Implementujte algoritmus, který načte informace o čipových hodinkách a o jejich majiteli a dále dobije kredit na čipové hodinky.
Proveďte otestování nových funkcionalit.
Proveďte zhodnocení bezpečnosti při dobíjení.
Proveďte srovnání a zhodnocení stávajícího a navrhovaného řešení systému.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 28. ledna 2016

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra počítačových systémů

Bakalářská práce

Analýza a návrh změn pokladního systému na Koupališti Flošna

Lenka Stejskalová

Vedoucí práce: Ing. Petra Pavlíčková, Ph.D.

16. května 2016

Poděkování

Ráda bych poděkovala své vedoucí bakalářské práce Ing. Petře Pavlíčkové, Ph.D., za odborné vedení, za pomoc a rady při zpracování této práce. Mé poděkování patří též Janu Konvalinkovi, MBA, za spolupráci při získávání údajů pro analytickou část práce. Velké poděkování náleží mé rodině a přátelům za podporu, trpělivost a povzbuzování po dobu mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2016 .....

Odkaz na tuto práci

Stejskalová, Lenka. *Analýza a návrh změn pokladního systému na Koupališti Flošna*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Práce se zaměřuje na analýzu a návrh změn pokladního systému na Koupališti Flošna v Hradci Králové. Cílem práce je zanalyzovat nynější stav systému, navrhnout jeho vylepšení a toto řešení srovnat se stávajícím řešením. První část práce obsahuje analýzu současného stavu systému. Druhá část práce se zaměřuje na návrh optimalizovaného řešení systému a návrh dobíjecího automatu. Práce se též zaměřuje na bezpečnost z hlediska systému a dobíjení. Práce analyzuje bezpečnost čipových karet a použitý typ šifrování dat. Bylo navrhnuto řešení, které zefektivnilo chod systému, eliminovalo problémy stávajícího řešení a zjednodušilo práci personálu koupaliště. Byl implementován a otestován dobíjecí automat. Návrh byl předán majiteli koupaliště.

Klíčová slova pokladní systém, analýza systému, UML model, use case, automat, bezpečnost systému, bezkontaktní karty, čtečka čipů

Abstract

The thesis focuses on the analysis and the proposal of changes of cash system at Koupaliště Flošna in Hradec Králové. An objective of work is to analyze present state of the system, to propose its improvements and this solution compare with the contemporary solution. The first part of this thesis contains

an analysis of present state of system. The second part of thesis concentrates on the proposal of optimised solution of the system and the design of rechargeable automat. The thesis also focuses on the security regarding to the system and recharging. The thesis analyzes the security of smard cards and used type of data encryption. The proposed solution improves the system operation, eliminates some issues of the contemporary solution and simplifies a work of staff of the swimming pool. A rechargeable automat was implemented and tested. The proposal was referred to the owner of the swimming pool.

Keywords cash system, system analysis, UML model, use case, automat, system security, contactless cards, chip reader

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Jako téma své bakalářské práce jsem si vybrala Analýzu a návrh změn pokladního systému na Koupališti Flošna v Hradci Králové. Důvodem k výběru tohoto téma je především to, že jsem s tímto systémem již pracovala, podrobně znám jeho výhody i nevýhody, nedostatky a úskalí systému a jsem schopna tyto nedostatky eliminovat. K výběru též pomohly skvělé vztahy s vedením koupaliště, zejména s panem Janem Konvalinkou, MBA, který mi ochotně pomáhal během analyzování celého systému.

Koupaliště Flošna používá systém navržený firmou IVAR, a.s., která se zabývá navrhováním a implementací systémů pro rekreační, zábavní a kulturní centra. Samotný systém na koupališti je pouze modulem komplexního systému, který se skládá z modulu Wellness Studia Flošna, z modulu Restaurace Flošna, z modulu Městských lázní Hradec Králové a z modulu Aquacentra Hradec Králové. Všechny tyto moduly dohromady tvoří ucelený systém. Systém umožňuje zákazníkovi se zaregistrovat, získá tzv. čipové hodinky, náramek s čipem, který v sobě nese informaci o kontě zákazníka. Na toto konto si zákazník může vkládat peníze, kterými pak platí jednotlivé vstupy do areálů a služby. Pro takto registrovaného zákazníka platí též jiný, zvýhodněný ceník vstupů a služeb.

Koupaliště Flošna se nachází v centru Hradce Králové, blízko historického jádra města a oblíbeného obchodního centra. Úspěšnost koupaliště ovšem nespočívá pouze ve strategickém umístění a monopolu v okruhu 30 km, ale i v příjemném osvěžení a v široké nabídce služeb areálu. V horkých letních dnech, kdy je město rozpálené, se koupaliště potýká s tisíci lidmi vstupujícími do areálu. U pokladen se tvoří fronty čím dál více nespokojených zákazníků, rychlá obsluha je tedy hlavním požadavkem. Cokoliv, co napomůže urychlit proces vstupu do areálu, je vítáno.

Výsledkem práce bude zoptimalizovaný systém pro vydávání vstupenek, který bude moci používat i nekvalifikovaný personál. Novinkou v tomto systému bude implementace dobíjecího automatu. Ten zákazníkovi umožní zjistit stav svého konta na čipových hodinkách a zároveň ho může zákazník využít

k dobití peněz na konto. Tento automat bude mít za následek především zrychlení obsluhy zákazníků na koupališti.

Práce je rozdělena na dvě části, teoretickou a praktickou. V praktické části je vytvořena analýza systému, navržena jeho optimalizace, implementován a otestován dobíjecí automat a následně je srovnáno stávající a navrhované řešení.

Kapitola 1

Cíl práce

Cílem práce je především optimalizovat stávající řešení a zefektivnit jeho používání. V práci je provedena analýza, na jejímž základu je navržen, implementován a otestován nový modul systému - dobíjecí automat. Smyslem dobíjecího automatu je ulehčit práci personálu koupaliště a urychlit obsluhu zákazníků. Dalším cílem práce je zhodnotit, zda a na jaké úrovni je bezpečnost stávajícího řešení pokladního systému a jak lze případně zvýšit bezpečnost. Dobíjecí automat je pro zákazníky samoobslužný, jedním ze základních požadavků na tento modul je zajištění nejvyšší možné bezpečnosti při dobíjení. V neposlední řadě si práce klade za cíl vytvořit komplexní analýzu pokladního systému koupaliště se zaměřením na funkcionality s ohledem na bezpečnost.

Kapitola 2

Literární rešerše

2.1 Pokladní systémy

2.1.1 Současný stav řešení problematiky

Na trhu vstupenkových a pokladních systémů je nyní mnoho společností zabývajících se tímto odvětvím. K tomu přispívá i dlouhodobý přesun podnikání do elektronické formy. Velká nabídka systémů tak naplní poptávku po rozmanitosti systémů nejrůznějších zaměření.

2.1.2 Možnosti řešení

Volba řešení pokladního systému se odvíjí od požadavků majitele a potřeb jeho podnikání. Podnikatel se v současné době nemusí omezovat, buď si vybere z široké škály nejoptimálnější produkt, nebo si nechá produkt sestavit na míru. Systémy se liší svou komplexností. Nejjednodušší pokladní systémy se skládají i z pouhé jedné komponenty. Složitější systémy tvoří více propojených komponent. Nejsložitější systémy se dělí na databázové a operační části.

2.2 Analýza

2.2.1 Metodiky vývoje softwaru

Metodika vývoje softwaru je souhrn postupů pro návrh a vývoj softwarového systému. Postup vývoje se dělí do 4 základních fází:

- plánování
- analýza (specifikace)
- návrh
- implementace

Po implementaci typicky následuje testování, instalace a údržba produktu. Fáze analýzy a návrhu by správně měla trvat většinu času životního cyklu (až 80 %). Pokud se nevěnuje dostatečná pozornost na počátku procesu, mohou být na pozdější fáze kladeny vyšší nároky.

Existují 2 základní metodiky vývoje softwaru - klasické a agilní. Klasické metodiky jsou složitější a přesnější, řeší více problémů, kladou důraz na dokumentaci. Mezi nejpoužívanější klasické metodiky se řadí Unified Process (UP) a Modelem řízený vývoj (MDA). Oproti tomu agilní metodiky se zaměřují více na samotnou tvorbu produktu, jeho úpravy provádějí na základě zpětné vazby, minimalizují dokumentaci. Nejčastěji uplatňovanými agilními metodikami jsou metodiky Extrémní programování (XP) a SCRUM. [1]

2.2.2 UML jazyk

UML (Unified Modeling Language) je standardní jazyk pro specifikování, vizualizaci, konstrukci a dokumentaci struktur softwarových systémů [2]. Lze ho definovat úslovím „jeden obraz vydá za tisíc slov“. Diagramy vytvořené pomocí UML jazyka pomáhají k popisování a vizualizaci návrhu a struktury softwarového systému. UML jako nástroj k definování struktury systému je velmi užitečný způsob, zejména pokud se se systémem seznamuje nový uživatel. [3]

UML bylo zveřejněno roku 1997 skupinou jménem Object Management Group. Jazyk byl zamýšlen tak, že měl poskytovat profesionální vývojářské komunitě jednotný a stabilní jazyk pro vytváření počítačových aplikací. Důvodem, proč se stal standardem, byla jeho nezávislost. Navíc UML je jazyk a ne metodologie, snadno se začlení mezi hotové struktury systému bez nutnosti změny. [4]

Základními kameny UML jsou předměty, relace a diagramy. Relace ukazují, jak spolu předměty souvisí a popisují tak funkcionalitu. Předměty a relace dohromady tvoří diagram, nejdůležitější část celého procesu.

2.2.3 Strukturální diagramy

Strukturální diagramy reprezentují statickou část systému, části, které tvoří hlavní strukturu. Dělí se na 4 druhy diagramů. Class diagramy (diagram tříd) jsou diagramy reprezentující objektově orientovaný pohled na statický systém. Je určen obecně pro vývojářské účely. Object diagram (diagram objektů) popisuje instanci class diagramu. Tyto diagramy jsou nejblíže reálným scénářům při implementaci. Využívají se k tvorbě prototypů systému. Component diagram (diagram komponent) se skládá z komponent a jejich vztahů mezi nimi, komponenty jsou třídy, rozhraní a interakce. Reprezentují pohled na implementaci systému. Deployment diagram (diagram nasazení) zobrazuje fyzické umístění zařízení a jejich spojení. [5]

2.2.4 Diagramy chování

Diagramy chování popisují dynamickou stránku systému, zaznamenává pohyb jednotlivých částí systému. UML má 5 typů diagramů chování. Use Case diagram (diagram případů užití) je množina případů užití, účastníků a vztahů mezi nimi. Popisuje jednotlivé funkcionality systému, vztahy mezi systémem a činností aktérů. Sequence diagram (sekvenční diagram) je diagram interakcí. Diagram zobrazuje sekvence volání z objektu do objektu, které vykonávají určité funkcionality. Collaboration diagram (diagram komunikace) je jiný druh diagramu interakcí. Reprezentuje strukturu systému a přijatá či odeslaná volání systému. Je podobný sekvenčnímu diagramu, ale účelem diagramu komunikací je ukázat objekty a jejich interakce. Statechart diagram (stavový diagram) zobrazuje „životní cyklus“ systému. Systém se bude vyskytovat v různých stavech, do kterých se dostane reakcemi na různé vnitřní či vnější podněty. Posloupnost těchto stavů je popsána stavovým diagramem. Activity diagram (diagram aktivit) znázorňuje tok řízení systému, zaznamenává funkce systému a to, jak bude systém přesně pracovat. [5]

2.2.4.1 Use case diagram

Use case zaznamenává vztahy mezi účastníky systému. Ukazuje chování systému pod různými podmínkami, jak systém odpovídá na požadavky účastníků. Diagram shromažďuje různé scénáře a sekvence chování systému. [6]

Při návrhu systému je velmi důležité zaznamenat dynamičnost, chování systému za běhu. Aby se chování mohlo měnit, musí systém obsahovat nějaké vnější či vnitřní faktory tvořící interakce. Tyto faktory jsou účastníci a jimi vytvářené interakce se nazývají relacemi. Jeden use case popisuje část funkcionality systému, pro popis celé dynamické stránky systému je třeba více use cases. [7]

Účelem use case je shromáždit požadavky na systém včetně vnitřních a vnějších vlivů. Ukazují vzájemné působení mezi požadavky a účastníky. [7]

Use case lze zaznamenávat různými způsoby, od textu přes sekvenční tabulky až po programovací jazyky. Nejčastěji se kreslí do diagramu, kde se znázorňují účastníci, funkcionality v podobě use case a vztahy mezi nimi.

2.2.4.2 Diagram aktivit

Daigram aktivit zobrazuje tok aktivit. Aktivitou je myšlena funkce provedená systémem. Tok aktivit se může větvit pomocí elementů `fork` nebo `join`. [8]

Diagram zobrazuje dynamické chování systému. Slouží také ke konstrukci systému. Diagram se skládá z aktivit, asociací a podmínek. Diagram jako jediný také může popisovat tok aktivit mezi systémy jako například mezi databází nebo jinými systémy. [8]

Diagram se používá pro modelování aktivit z hlediska businness požadavků. Má tedy větší dopad na pochopení práce systému než na implementační detaily. [8]

2.3 Bezpečnost a spolehlivost systémů

„Nasazení informačních systémů a informačních technologií se v dnešní době stalo nutnou podmínkou úspěšnosti firem ve všech oblastech hospodářské činnosti. Příčinou tohoto faktu je, že se IS/IT staly jedním z rozhodujících faktorů rozvoje a konkurenceschopnosti hospodářských organizací.“ [9, s. 9]. Informační systémy zaujímají místo i v oblastech státní správy a bankovnictví.

U každého systému je velmi důležité zabývat se i jeho spolehlivostí, bezpečností a životností. S rozvojem informačních technologií totiž vzrůstá i možnost jejich zneužití. Informace, údaje a majetek, se kterými se v systémech pracuje, se staly velkým terčem kyberzločinců.

2.3.1 Pojmy

Pro správné pochopení problematiky je nutné vysvětlit alespoň následující pojmy. Bezpečnost systému budeme chápat jako míru odolnosti vůči rušivým zásahům, *„míru pravděpodobnosti, že ani činností těchto soustav, ani selháním jejich funkcí nedojde ke škodám a úhoně lidské společnosti a jejího životního prostředí, resp. určité skupiny lidí.“* [10, s. 11].

Spolehlivost *„je míra pravděpodobnosti, že po jistou dobu či v jistém rozpětí jiných, na systém působících nezávisle, proměnných se jejich systémové funkce nebudou odchylovat od požadovaných hodnot více než o dovolené odchylky.“* [10, s. 11]

2.3.2 Informační bezpečnost

Pojem informační bezpečnost dle [11, slide 6] chápeme jako *„souhrn prostředků a postupů na zabezpečení důvěrnosti, integrity a dostupnosti informací, na zabezpečení autentizace uživatelů a zdrojů, účtovatelnosti operací, jakož i zabezpečení ochrany proti neautorizované manipulaci, modifikaci nebo zničení, resp. poškození informací v informačním systému.“* S doplněním podle [9] se jedná o ochranu informací během celého jejího životního cyklu, během vzniku, zpracování, ukládání, manipulace a zničení.

Zde je namístě přesněji si definovat pojmy výše uvedené definice.

- *„Důvěrnost (confidentiality) je vlastnost, která zaručuje, že informace nebude dostupná neautorizovanému subjektu.“*
- *„Integrita (integrity) je vlastnost, která zaručuje úplnost a přesnost zpracované, resp. přenášené informace.“*

- *„Dostupnost (availability) je vlastnost, která zaručuje, že informace bude dostupná autorizovanému subjektu.“* [11, slide 6]

Pro dosažení maximálního účinku ovšem nestačí zajistit zabezpečení pomocí hesel. Systém dosahuje nejvyšší možné bezpečnosti pouze spoluprací různých typů zabezpečení jako je antivirová ochrana, zálohování a další. Hlavním požadavkem je komplexnost a provázanost jednotlivých opatření vztahujících se na určitý celek - informační systém.

[11] dělí informační bezpečnost na 2 části:

- počítačová bezpečnost - prostředky zabezpečující počítače a ochranu jeho dat,
- síťová bezpečnost - prostředky zabezpečující data během přenosu (komunikace s jiným počítačem).

2.3.3 Hrozby a útoky

2.3.3.1 Hrozby

Hrozby jsou události či osoby nějakým způsobem umožňující narušení bezpečnosti. Objevují se v místech, která jsou slabinou v systému, kde by mohlo dojít ke škodám či ztrátám dat - zranitelná místa. [12] dělí hrozby takto:

- objektivní
  - přírodní (přírodní katastrofy)
  - fyzikální (záření)
  - technické (porucha či krádež komponent)
- subjektivní
  - neúmyslné (nesprávná obsluha systému, neproškolené osoby)
  - úmyslné (útok)

Objektivním přírodním hrozbám jako jsou požár, povodeň a výpadky proudu, se jen těžko vyhýbá, mohou udeřit kdekoliv. U těchto hrozeb je vhodné spíše než na prevenci se zaměřit na havarijní plán a minimalizaci dopadů ohrožení.

Subjektivní ohrožení tvoří drtivou většinu ohrožení systému. Roli zde hraje lidský faktor. Může se jednat jak o vlastní zaměstnance - běžné uživatele systému, tak o externí pracovníky či hackery. *„Vlastní pracovníci představují nejrizikovější faktor ohrožení informací. Podle odhadů způsobují zhruba 80 % případů porušení ochrany informací. Musíme přitom vzít do úvahy takové vlivy, jako je neodborné zacházení a chyby operátorů, lidské selhání nebo omyl, nespokojenost, zloba a pomstychtivost.“* [9, s. 19]

Subjektivním neúmyslným hrozbám se běžně předchází uplatněním různých oprávnění uživatelům systému.

Největší hrozbou z hlediska potenciální škody je subjektivní úmyslná hrozba - útok. Charakteristikou takové hrozby je její zdroj (uvnitř či vně systému), motivace k útoku, frekvence a kritičnost uplatnění hrozby. Nejčastěji jde o modifikaci údajů neautorizovaným subjektem, kopírování a odposlech informací či integrace virů do systému.

2.3.3.2 Útoky

Útok (incident) je *„úmyslné využití zranitelného místa, tj. využití zranitelného místa ke způsobení škod/ztrát na aktivech IS, nebo neúmyslné uskutečnění akce, jejímž výsledkem je škoda na aktivech“* [12, s. 15]. Jedná se o jakoukoli nečekanou akci způsobující nezvladatelnost či změny systému, neautorizované použití či zneužití systému.

Útoky lze dělit různě, dle stupně škody na systému, dle úmyslnosti, podle toho, na co je útok veden (hardware, software, data). Základní dělení útoků je:

- aktivní
- pasivní

Pasivní útoky jsou především útoky odposlechem, kde nedochází ke změně systému. Odposlechem je míněno např. neautorizované kopírování aktiv.

Aktivní útoky lze dále dělit na:

- útok přerušením - znepřístupnění, porucha, vymazání programu či aktiva
- útok změnou - zásah do aktiva, změna dat
- útok přidáním hodnoty - vytvoření a přidání falešných dat

Důležitým aspektem je ochrana před útoky. Před pasivními útoky se brání především prevencí jejich vzniku. Ochrana před aktivními útoky je detekce útoků, poučení a adekvátní reakce na případný již proběhlý útok.

2.3.4 Zvýšení bezpečnosti

Bezpečnost informačního systému lze definovat pomocí tří výše představených pojmů - důvěrnost, integrita a dostupnost. Jejich poměr pak formuluje daný systém a požadavky na něj.

Na zvýšení bezpečnosti systémů obecně se lze dívat ze čtyř směrů:

- *„z hlediska návrhu a konstrukce uvažovaného systému tak, aby kromě svých základních požadovaných funkcí vykazoval též co největší provozní spolehlivost a životnost;“*
- *„z hlediska analýzy spolehlivosti jistého již existujícího systému;“*

- *„z hlediska spolehlivosti interakce mezi umělými, člověkem vytvořenými systémy a lidskými operátory (řidiči, piloty, dispečery apod.), resp. lidskými uživateli;“*
- *„z hlediska doporučení a norem pro zajištění a zaručení (garantování) spolehlivé funkce systému.“* [10, s. 17]

Tato hlediska platí pro systémy obecně, nejvíce však pro technické systémy. Z obecného hlediska jsou ale nejdůležitější následující opatření:

- použití nejkvalitnějších komponent
- zálohování
- předvídání ohrožení a s tím související návrh protiopatření

Bezpečnost informačních systémů se dělí specifičtěji. Dle [9] se realizuje kombinací následujících opatření:

- organizační
- fyzická
- technická
- programová
- šifrovací
- zálohovací
- antivirová

2.3.4.1 Organizační opatření

Organizační opatření jsou různá pravidla, nařízení a směrnice vydané v dané společnosti. Jedná se o dokumenty určující zodpovědnosti uživatelů systémů. Nařízení musí být jasně vymezená, vydaná písemně a každý uživatel s nimi musí být předem seznámen (před manipulací se systémem). Aby organizační opatření plnila svou maximální funkci, je třeba tyto dokumenty pravidelně aktualizovat.

2.3.4.2 Fyzická opatření

Fyzická opatření chrání systém z jeho fyzické stránky - zajišťují fyzickou ochranu. Úkolem těchto opatření je chránit objekt, ve kterém je systém umístěn, před přírodními katastrofami i před neoprávněným vniknutím osob. Součástí je i zajištění nepřetržitého zdroje elektřiny.

2.3.4.3 Technická opatření

Technická opatření mají za úkol zajistit stabilitu systémů díky kvalitnímu hardwaru a jeho autorizovanému servisu. Z hlediska financí patří tato opatření k těm nákladnějším.

2.3.4.4 Programová opatření

Programová opatření chrání systém přímo v počítačích, zabraňují neoprávněným přístupům k datům a monitorují a logují (zejména podezřelé) aktivity. Obecně nejdůležitějším způsobem ochrany jsou hesla a přístupová práva.

Heslo Ochrana systému pomocí hesla je nejběžněji používanou metodou zabezpečení. Heslo plní funkci autentizace uživatele, určuje, zda přihlašovaný uživatel opravdu je tím, za koho se vydává. Útoky na systém jsou nejčastěji prováděny zjištěným heslem, ať už prozrazeným, uhodnutým či dešifrovaným. Proto by v systémech měla platit alespoň základní pravidla pro tvoření hesel. Absolutní základ pro vytvoření nového hesla je jeho minimální délka. Dalším neméně důležitým aspektem je pravidelná změna a neopakování hesel po určitou dobu, nevytváření hesel s obsahem vlastního či uživatelského jména a správná kombinace číslic, symbolů a znaků a jejich velikostí.

Oprávnění uživatelů Oprávnění uživatelů je mechanismus rozlišování různých stupňů autorit v informačním systému. Práva určují uživatelům či skupinám uživatelů, co smí a co nesmí v systému dělat. Každý uživatel by měl mít povoleno provádět pouze takové operace, které potřebuje ke své práci, jinak by mohlo zbytečnou manipulací dojít k ohrožení systému v podstatě neoprávněnou osobou.

2.3.4.5 Šifrování

Šifrování informací je prováděno, aby nebylo možné běžnými prostředky data přečíst (bez znalosti šifrovacího klíče). Zašifrovaná data jsou tak bezpečná i při přenosu dat po veřejné komunikaci. Kryptografii lze využít při šifrování dat, disků, komunikací a podpisů dat a dokumentů, tzv. digitální podpis.

2.3.4.6 Zálohování

Zálohování je proces vytváření kopie dat na jiné úložiště dat, použití nadbytečných prostředků ke zvýšení bezpečnosti. Zálohování dat je prováděno jako prevence před případným selháním informačního systému. Vytváření záloh systému se dělí na:

- provozní zálohování - pro případnou obnovu systému
- archivace - archivování hodnot [9]

Zálohy se dle stupně využití v čase dělí na statické a dynamické. Statické zálohy jsou k systému připojeny trvale a v případě poruchy maskují chybu a nedovolí tak jejímu proniknutí dál do systému. Oproti tomu dynamické zálohy se připojí k systému až po spuštění přepínače záloh, který detekuje poruchu. Využívají se tedy, až když je to nezbytně nutné.

2.3.4.7 Antivirová ochrana

Antivirová ochrana je bezpečnostní mechanismus, který brání systém před napadením prostřednictvím počítačových virů. Počítačový vir je program infiltrující se do programu systému, ve kterém zanechá svou kopii s možností dalšího šíření se. Před takovým napadením lze ochránit systém především využitím antivirových nástrojů, které hledají určité skupiny virů, používáním pouze legálních programů a používáním kontrolních programů na ochranu datových úložišť.

2.4 Čipové hodinky

2.4.1 Čipové karty

Základem čipové karty je čip nesoucí data. Karty nejčastěji slouží k identifikaci majitelů. *„Pro využití v rámci bezpečnostní infrastruktury jsou nejzajímavější mikroprocesorové (smart) karty, které obsahují podobné komponenty jako celý počítač – procesor, specializované kryptografické koprocesory, různé typy pamětí a vstupně/výstupní kanály integrované na jediném čipu. Moderní čipy mají implementovánu řadu bezpečnostních mechanismů, které ztěžují různé typy útoků na bezpečnost a jsou odolné proti útokům invazivním (použití chemikálií a mechanických nebo fyzikálních interakcí) i neinvazivním (například použití diferenciální analýzy spotřeby DPA). Neméně důležitou částí čipové karty je zabudovaný software – operační systém, který je zpravidla umístěn v paměti ROM čipu. Právě kombinace možností čipu a funkcí operačního systému je podstatou konkrétní čipové karty. Čipová karta je specializovaný miniaturní kryptografický počítač, který komunikuje s PC nebo terminálem prostřednictvím kontaktního nebo radiového přenosu a bezpečně realizuje kryptografické a datové operace.“* [13, odst. 1]

Čipové karty se dělí na kontaktní a bezkontaktní.

2.4.2 Kontaktní čipové karty

Kontaktní karta má zabudovanou kontaktní plošku s osmi kontakty. Rozhraní je standardizováno normou ISO 7816. Kontakty slouží k napájení, kominukaci a převedení signálu a programovacího napětí. [13]

Novinkou u kontaktních čipových karet je standard ISO 7816-12. Tato norma umožňuje vyrábět karty integrující USB rozhraní přímo na čipu, jinak známé USB-ICC. [13]

2.4.3 Bezkontaktní čipové karty

Bezkontaktní čipové karty komunikují s terminálem na krátkou vzdálenost (do 10 cm) bez nutnosti přímého fyzického kontaktu. Existuje mnoho různých druhů bezkontaktních rozhraní:

- nízkofrekvenční - technologie EM4102
- vysokofrekvenční - standard ISO 14443, ISO 15693
- ultra krátké vlny [14]

Bezkontaktní čipové RFID karty kompatibilní s technologií EM4102 využívají nízkofrekvenční rozsah kmitočtu 125 kHz. Karty se standardy ISO 14443 nebo ISO 15693 komunikují na frekvenci 13,56 MHz.[14] Typ karet Gen 2 UHF Card pracuje mezi 868 a 928 MHz.

„V současné době lze, díky pokračující miniaturizaci a snižující se spotřebě, realizovat pohodlně i komplexní kryptografické operace založené na algoritmech RSA nebo ECC s využitím radiového přenosu, je však nutno řešit nová bezpečnostní rizika spojená s radiovým přenosem (neoprávněné čtení, odposlouchávání, přesměrování).“[13, odst. 3]

2.4.4 Čipové karty MIFARE

Koupaliště Flošna používá pro identifikaci zaregistrovaných zákazníků bezkontaktní čipové karty značky MIFARE společnosti NXP Semiconductors N.V. s rozhraním ISO 14443. Jde o bezkontaktní kartu MF1ICS50 z rodiny karet MIFARE Classic, jedna z nejpoužívanějších bezkontaktních karet.

Tabulka 2.1 ukazuje vlastnosti karet MIFARE Classic EV1.

Karta MF1ICS50 komunikuje dle standardu ISO 14443 typu A. Data i potřebné napájení je transportováno bezkontaktně, data se přenášejí rychlostí 106 kbit/s. Karta komunikuje na vzdálenost až 100 mm v závislosti na anténě karty. Karta obsahuje paměť EEPROM, paměť s obsahem 1 Kbyte. Paměť je organizována v 16 sektorech se 4 bloky o 16 bytech. Poslední blok každého sektoru obsahuje 2 tajné klíče. [15]

2.4.5 Bezpečnost čipové karty MIFARE

Karta MF1ICS50 využívá technologii antikolize. Antikolizní funkce dovoluje pracovat s více kartami v dosahu terminálu najednou. Algoritmus vybere každou kartu individuálně a provede transakce s jejími daty, následně kontroluje správnost provedení (zda byla provedena správná operace s daty správné karty). [15]

	MIFARE Classic EV1
RF Interface	Contactless only, ISO/IEC 14443-2
Protocol	ISO/IEC 14443-3
UID	7-byte UID, RID, 4-byte NUID
Comm, Speed	106 kbps
EEPROM	1KB, 4KB
Memory Model	Compact, Sectors & 16-byte block
Crypto	Crypto-1
Key Length	48-bit
Authentication	3-pass mutual
Comm, Security	Encrypted
MIsmartApp	N/A
Transaction MAC	N/A
Multi Key Sets	N/A
Proximity Check	No
Virtual Card Select	No
CC Certification	No
ISO 7816-4 APDU	No

Tabulka 2.1: MIFARE Classic EV1 - vlastnosti

[2.1] MIFARE Classic EV1. In: NXP [tabulka]. NXP, 2016. [cit. 2016-05-13]. `https://www.mifare.net/wp-content/uploads/2015/03/MIFARE-Classic-EV1.pdf`.

Autentikace karty je prováděna tříchodovým autentikačním protokolem standardu ISO/IEC 9798-2-4 (Three-Pass Mutual Authentication). Protokol zajišťuje bezpečnou komunikaci mezi dvěma stranami bez nutnosti výměny šifrovacích klíčů. Autentikace probíhá ve třech krocích:

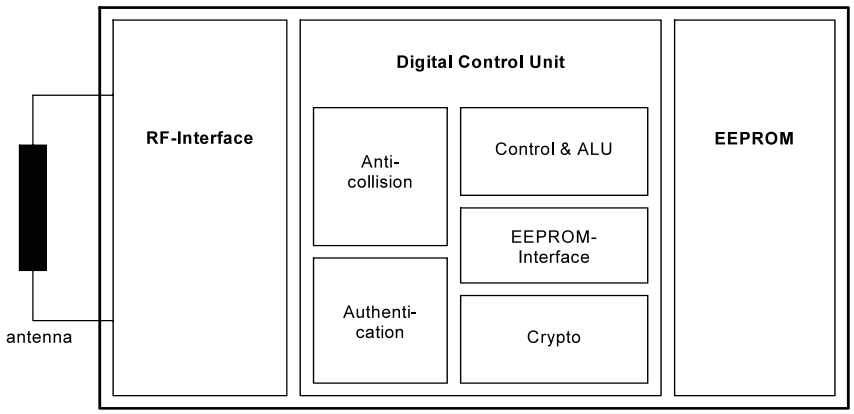
1. Strana A si zvolí šifrovací `s` a k němu dešifrovací `t` klíč. Strana A pošle zprávu `m` zašifrovanou šifrovacím klíčem `s` straně B.

2. Strana B si zvolí šifrovací `r` a k němu dešifrovací `q` klíč. Strana B obdrží zprávu `E(s, m)`, zašifruje ji svým šifrovacím klíčem `r` a odešle straně A.

3. Strana A obdrží zprávu `E(r, E(s, m))`, dešifruje ji dešifrovacím klíčem `t`, čímž získá zprávu `E(r, m)`, kterou odešle straně B.

   ```
   D(t, E(r, E(s, m))) = E(r, m)
   ```

   Strana B vlastní dešifrovací klíč ke klíči `r`.



Obrázek 2.1: Blokové schéma MF1ICS50

[2.1] Block diagram. In: MF1ICS50, Functional specification [obrázek]. NXP, 2008. [cit. 2016-05-13]. `http://www.nxp.com/documents/data_sheet/M001053_MF1ICS50_rev5_3.pdf`.

Karta používá šifrovací algoritmus Crypto1 vytvořený společností NXP Semiconductors. Crypto1 je proudová šifra s klíčem délky 48 bitů. Skládá se z LFSR (lineární zpětná vazba posuvných registrů), lineární funkce zpracovávající LFSR a kombinační filtrovací funkce sloužící k vygenerování výstupního klíče. Algoritmus určí, které bity budou použity pro vytvoření nového bitu, který bude posunut do registru. Vstupní bit je poté xorován s nově vytvořenými bity z předchozího kroku a vytvoří tím nový bit. Kombinční filtrovací funkce zaručuje bezpečnost šifrování. [16] [17]

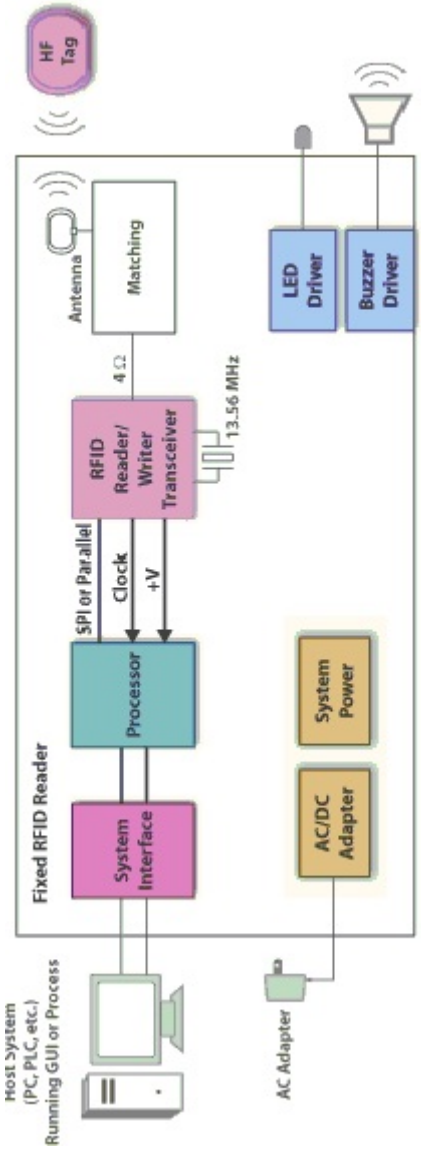
Samozřejmostí je unikátní sériové číslo každé karty.

2.4.6 Čtečky čipových karet

Koupaliště Flošna využívá čtečky čipů typu Promag PCR340 - Dual Frequency RFID and MIFARE Reader a Promag PCR300M - Desktop MIFARE Reader. Obě čtečky komunikují s kartami MIFARE Classic, s počítačem jsou spojeny USB či RS232 rozhraním.

Návrh RFID (Radio Frequency Identification) čtečky obsahuje relativně málo komponent, zejména části přijímací/vysílací čip, mikrokontroler, anténu

a rozhraní systému, které zpracuje informace přijaté čtečkou. Čtečky takového návrhu vyrábí mnoho společností, mezi nimi právě i NXP Semiconductors.



Obrázek 2.2: Schéma RFID čtečky

[2.2] RFID reader chips simplify the design of tag reading systems but specific implementations require choosing from a variety of interfaces and MCU options. In: Design Considerations for HF-RFID Reader Systems [obrázek]. Tomáš Jecha. [cit. 2016-05-11]. `http://www.digikey.com/en/articles/techzone/2011/aug/design-considerations-for-hf-rfid-reader-systems`.

Kapitola 3

Praktická část

3.1 Analýza stávajícího řešení systému

3.1.1 Stávající řešení

V současné době areál využívá produkt společnosti IVAR, a.s. Společnost se zabývá návrhem, implementací a sestavením systémů pro automatizaci, identifikaci a odbavení. Součástí systému je návrh hardwarových komponent a využití identifikačních karet a čipů. Koupaliště Flošna i celý hradecký rekreační areál využívá modulární vstupenkový a pokladní systém VAPS určený zejména pro sportovní a kulturní zařízení a stadiony. Systém se může pyšnit stavebnicovým charakterem, umožňuje tak sestavit systém bez ohledu na velikost podnikání zákazníka, systém lze zároveň bez větších komplikací rozšířit o další komponenty firmy [18]. Společnost provedla instalaci systému již při vzniku koupaliště a stále dodává pravidelné aktualizace systému. Stávající řešení je tedy funkční a testované.

3.1.2 Čipové hodinky

Základním prvkem systému je možnost zákazníka vytvořit si na koupališti vlastní konto s kreditem - tzv. čipové hodinky. Jde o nylonový náramek s čipem, který nese informaci o kontě zákazníka. Systém ukládá následující informace o kontě:

- jméno a příjmení zákazníka,
- identifikační číslo čipu,
- identifikační čísla konta,
- aktuální kredit,
- dospělý/dítě

- datum posledního dobíjení kreditu

Volitelně lze nastavit následující informace:

- telefon
- email
- blokované služby
- firma

Konto tedy vlastní 3 druhy identifikačních čísel. Nejde tu ale o duplicity. Identifikační číslo čipu je číslo daného čipu, který zákazník vlastní. V případě, že by zákazník náramek ztratil, rozbil ho či mu byl odcizen, zaměstnanec koupaliště uloží ke kontu zákazníka identifikační číslo jiného čipu, který zákazník dostane. Odpadá tedy nutnost vytvoření zcela nového konta a ztráta nabitých peněz.

Identifikační čísla konta jsou dvě rozdílná čísla konta, podle kterých systém vyhledává dané konto. Pomocí jednoho čísla vyhledávají konto moduly na Flošně, pomocí druhého moduly v Městských lázních. Zde by se do jisté míry mohlo jednat o duplicity, pravděpodobně jde o problém rozdílných verzí programu na Flošně a v Městských lázních. Jelikož je rekreační centrum na Flošně výrazně mladší než Městské lázně, mají jisté rozdíly v programech (nastavení automatické slevy 10 % při platbě čipovými hodinkami apod.).

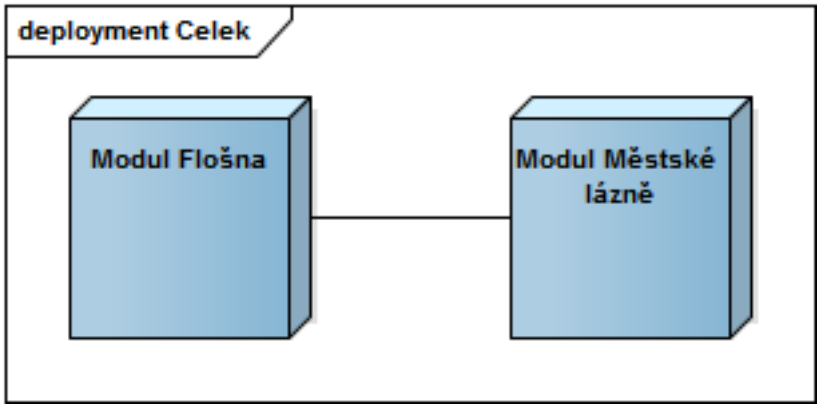
3.1.3 Pohled na celý systém

Celý pokladní systém VAPS pracuje se dvěma servery umístěnými na vzdálených místech v Hradci Králové. Tím se systém rozděluje na dvě hlavní místa fungování - Městské lázně a Flošna. Všechny moduly systému jsou připojeny k jednomu z těchto serverů podle místa provozu. Tyto servery jsou duplicitní a synchronizují svá data každých 15 minut.

Systém ukládá všechny pohyby na kontech všech zákazníků od začátku provozu po současnost. Loguje se každé dobití peněz na konto včetně způsobu dobití, každá účtovaná služba, projití turniketem, blokace, odblokování, změna údajů konta, atd. Loguje se i změna v nastavení systému. U každého logu se zaznamenává čas a kdo danou změnu konta či systému provedl - přihlášený uživatel.

3.1.4 Moduly systému

Celý systém se díky dvěma duplicitním serverům rozdělil na pomyslné dva díly. Jsou rozděleny spíš z geografického než z funkčního hlediska, je praktičtější využívat server blízko programu. Tyto oddíly jsou dále rozděleny na moduly. Každý takový modul je napojen na server. Moduly nejsou pevně ohraničeny,



Obrázek 3.1: Moduly systému

jedná se spíše o účty, které dovolují poskytovat pouze předem povolené a definované služby. Takové moduly se dají jednoduše přidávat, odebírat a měnit.

3.1.5 Moduly v areálu Flošna

Systém v areálu Flošna je tvořen jedním serverem a několika moduly. Tyto moduly jsou rozděleny do 3 kategorií:

- vnější pokladna
- vnitřní pokladna
- kreditní terminály

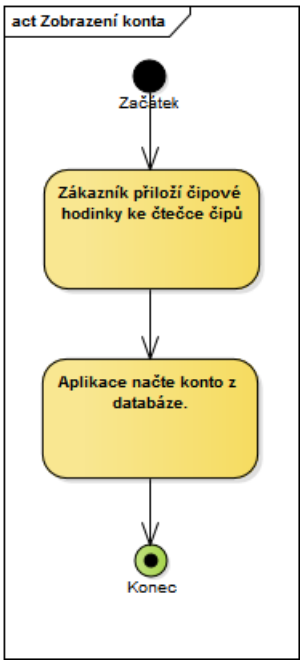
Vnější pokladna je tvořena modulem Letní koupaliště. Vnitřní pokladna se skládá z modulů Wellness studio a Squash. Kreditní terminály jsou terminály, kde lze pouze zaplatit danou částku čipovými hodinkami, ne dobíjet je. Tyto terminály jsou součástí modulu Restaurace a Fitness.

Tyto moduly nejsou vzájemně propojeny, nijak spolu neinteragují. Manažerský účet ovšem umožňuje zobrazit libovolné informace o všech modulech zvlášť, tzv. sestavy. Tyto sestavy též plní funkci při sestavování měsíčních a ročních výkazů.

3.1.6 Model případů užití

V modelu hrají svou roli 2 účastníci - zákazník a obsluha. Zákazník je majitel konta. Obsluha je uživatel systému, který má přiděleny práva Manažer nebo Pokladník.

Na obrázku v Příloze C.1 je zobrazen diagram případů užití nejdůležitějších funkcionalit systému. Nejčastěji se bude jednat o následující případy užití:



Obrázek 3.2: Diagram aktivit: Zobrazení konta

- UC1 Zobrazení konta
- UC2 Dobití konta
- UC3 Přihlášení do systému
- UC4 Účtování služby
- UC5 Tisk vyúčtování

3.1.6.1 UC1 Zobrazení konta

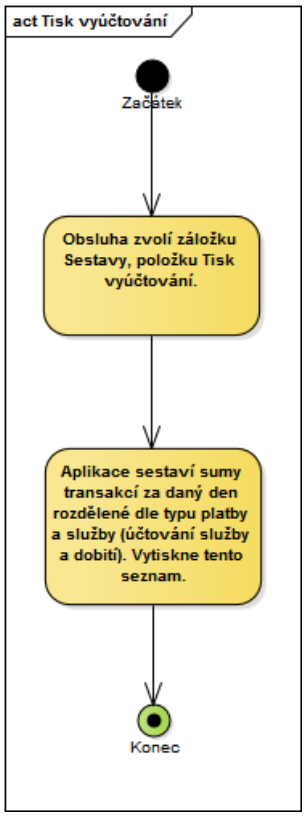
Případ užití popisuje postup zobrazení informací o stavu konta zákazníka (obrázek diagramu aktivit 3.2).

3.1.6.2 UC2 Dobití konta

Případ užití popisuje způsoby a možnosti dobití částky na konto zákazníka (obrázek diagramu aktivit v Příloze C.2).

3.1.6.3 UC3 Přihlášení do systému

Případ užití popisuje postup přihlášení uživatele do aplikace včetně kontroly přihlašovacích údajů (obrázek diagramu aktivit v Příloze C.3).



Obrázek 3.3: Diagram aktivit: Tisk vyúčtování

3.1.6.4 UC4 Účtování služby

Případ užití popisuje sekvenci úkonů prováděných při účtování služeb zákazníkovi (obrázek diagramu aktivit v Příloze C.4).

3.1.6.5 UC5 Tisk vyúčtování

Případ užití popisuje postup vytvoření dokladu o kompletním vyúčtování za daný den (obrázek diagramu aktivit 3.3).

3.1.7 Diagram nasazení

Jak už bylo výše uvedeno, celý systém je rozdělen na dvě hlavní části - Flošna a Městské lázně. Každá tato část má v blízkosti vlastní server a dále se dělí do modulů.

Hlavní modul Flošna je složen z několika menších modulů. Základem je samozřejmě server. Operačním systémem serveru je MS Windows Server 2008. Na něm běží MS SQL Server s nástrojem SQL Nexus a aplikací VAPS. Tento server komunikuje s dalšími částmi systému pomocí http protokolu (LAN

kabel). Server je přímo spojen s pokladnou Letního koupaliště, pokladnou Wellnessu a kanceláří vedení koupaliště. Všechny tyto části jsou tvořeny počítači s operačním systémem Microsoft Windows XP. Každý počítač spouští aplikaci VAPS s přihlášením pomocí čipů, k počítačům jsou tedy připojeny pomocí USB i čtečky čipů. Pokladny mají navíc připojeny i tiskárny lístků, na lístku je vytisknut čárový kód pro průchod turnikety do areálu. Dále je k počítačům připojena čtečka Městských karet, unikátních karet pro Dopravní podnik města Hradec Králové umožňujících platbu lístků na MHD. Kreditem na těchto kartách lze platit i vstupenky do areálu koupaliště. Do budoucna je plánováno zrušení tohoto způsobu platby, neboť čtečka způsobuje častá selhání aplikace VAPS. V místnostech pokladen jsou dále ovladače turniketů (například pro průjezd kolařů, kočárků či hendikepovaných), ovladače jsou ale připojeny pouze k serveru, server kontroluje průchody turnikety včetně toho, kdo (uživatel či zákazník) a kdy prošel.

Dalšími menšími moduly jsou modul Fitness a modul Restaurace. Tyto moduly jsou vybaveny pouze kreditním terminálem, terminálem, kam uživatel zadá částku a přiložením čipových hodinek se částka strhne z kreditu zákazníka. Terminály jsou připojeny k pokladně Wellness, kde je zvlášť zobrazena suma z jejich tržeb.

V serverovně je k serveru dále připojen hardware starající se o chod bezpečnostních kamer, LAN sítě, WiFi sítě a telefonického spojení uvnitř i vně systému.

Diagram nasazení je zobrazen na obrázku v Příloze C.5.

3.1.8 Oprávnění systému

V systému VAPS se lze pohybovat jako jeden ze 4 druhů uživatelů. Každý z těchto uživatelů používá stejný typ čipů, který je nabízen zákazníkům. Uživatelé se dělí do následujících skupin:

- manažer
- pokladník
- pracovník s kreditním terminálem
- uživatel čipu na projití turniketem

Poslední možností přihlášení je servisní pracovník společnosti IVAR, ten ovšem do systému obvykle nepřistupuje (pouze ve výjimečných případech, poruchách), proto není mezi skupinami uživatelů definován.

Pokud se bude postupovat od uživatelů s nejnižším oprávněním, nejprve je definován „uživatel čipu na projití turniketem“. Takový uživatel nemá přístup do systému, jeho čip funguje pouze jako klíč k turniketům. V praxi mají taková uživatelská práva využití např. pro funkci strojníka. Ten má pak přístup k celému areálu a zároveň nehrozí neoprávněný přístup k programům.

O stupeň vyšším oprávněním je „pracovník s kreditním terminálem“. Má přístup kamkoli v areálu a do systému zasahuje pouze ve velmi omezené míře. Uživatel pracuje pouze s kreditním terminálem, který je vzdáleně připojen k pokladně Wellness studia. Pracovník pouze zvolí částku, zákazník přiloží čip a částka se stržena z konta zákazníka.

„Pokladník“ je uživatel bezprostředně pracující s pokladním systémem. Jeho běžnou náplní práce je prodej vstupenek, manipulace s kontem zákazníků a prodej čipových hodinek. Každá změna na kontě zákazníka či vytvoření nového konta se loguje do záznamů včetně jména přihlášeného uživatele.

„Manažer“ je nejvyšší oprávnění v systému, může dělat prakticky cokoli. Oproti pokladníkovi smí vytvářet a definovat nové služby, smí zablokovat či odblokovat konto zákazníka, smí měnit nastavení systému, v případě potřeby smí převést kredit z konta na konto, smí provádět složitější storna, apod. Toto oprávnění mají z bezpečnostních důvodů pouze dva uživatelé systému. Manažer může navíc zobrazit tzv. sestavy ze všech modulů, kde se dozví sumy za jednotlivé služby s ohledem na způsob platby. Tyto sestavy hrají velkou roli ve vyúčtování za dané období.

3.2 Návrh optimalizovaného systému

3.2.1 Možnosti řešení optimalizace

Koupaliště Flošna používá již funkční, otestovaný a instalovaný pokladní systém a za předpokladu, že firma IVAR, a.s. nepřestane dodávat aktualizace a nepřestane provozovat servis systému, není důvod ke změně systému. Ke stejnému závěru dojdeme, i pokud bereme v úvahu to, že když bude chtít Koupaliště Flošna změnit vstupenkový systém, nebude nadále součástí modulů systému pro okolní hradecké rekreační areály. Výměna modulů systémů pro všechny areály by se pak mohla časově výrazně prodloužit o testování.

Na druhou stranu je zde důvod, proč uskutečnit změnu v systému - samotný systém VAPS je implementován pro jakýkoliv obecný sportovní areál. Systém není vytvářen na míru společnosti, i když se snaží přizpůsobit. Jsou zde minimálně využívané až nevyužívané funkce, které v GUI systému působí pro uživatele pokladny zmateně, a zároveň systém umožňuje dopustit se chyb při přihlašování více uživatelů.

S ohledem na všechna hlediska je nejvýhodnějším řešením optimalizovat stávající systém. Samotný reenginering je prováděn mimo fungování stávajícího systému, není třeba systém přerušovat. Testování optimalizovaných částí nezabere mnoho času oproti testování nového systému. Reenginering zajistí, že ve vylepšeném řešení systému bude to nejlepší ze stávajícího řešení a optimalizované části nedostatečně kvalitně řešených částí systému.

3.2.2 Návrh optimalizací

V případě, že by se Koupaliště Flošna rozhodlo pro optimalizaci stávající podoby systému, je třeba v nynějším řešení najít stěžejní nedostatky a navrhnout vylepšené řešení daného problému.

Největším problémem je nejednotná verze aplikací v jednotlivých modulech systému. Tento problém přetrvává již od počátku nasazení systému do části Flošna. Modul Městské lázně byl nasazen o mnoho let dříve než modul Flošna. Flošna tedy byla zprovozněna s novější verzí programu, ale modul Městských lázní aktualizován nebyl. Tento nesoulad ovšem zapříčinil špatné identifikování konta zákazníka. Systém v Městských lázních vyhledává konto v databází dle evidenčního čísla, systém na Flošně vyhledává konto podle identifikačního čísla. Tato čísla jsou od sebe vždy různá a zároveň jedinečná. Konto navíc obsahuje další jedinečné identifikační číslo - číslo čipu. Toto číslo slouží k identifikaci čipu - hardwaru, pokud by majitel konta ztratil čipové hodinky, lze jednoduše kontu změnit číslo čipu a zákazníkovi tak podat nové čipové hodinky ke stejnému kontu. Ve výsledku tak každé konto v databázi obsahuje zbytečné množství informací.

S rozdílnou verzí systémů přichází i další, ač nepříliš výrazné rozdíly v nastavení. Tyto rozdíly nehrají roli ve funkcionalitě systému, spíše se jedná o problém například při zastupování personálu zaměstnancem využívajícím standardně jinou verzi systému.

Dalším problémem je přihlašování uživatelů systému k pokladnám. Každá pokladna daného modulu má spuštěnou svou instanci aplikace, tyto instance jsou pak jednotlivě viditelné z účtu manažera. Manažer má tak přehled o jednotlivých pokladnách, resp. uživatelích u pokladny. Problém se objeví, pokud se k aplikaci přihlásí jeden uživatel na více pokladnách. Systém tento případ nijak nekontroluje, manažerský účet vidí pouze sumu těchto pokladen - vyhledává tržbu podle přihlášených uživatelů. Mohou se pak objevit potíže v nepřesnostech v tržbě a následným zjišťováním viníka nepřesnosti.

3.2.3 Návrh nového modulu systému

Součástí optimalizace systému je požadavek na vybudování dobíjecího automatu vznesený majitelem koupaliště. Ideou dobíjecího automatu je stroj umožňující registrovaným zákazníkům dobít kredit na čipové hodinky. Automat si budou zákazníci obsluhovat sami. Hlavním důvodem k požadavku na vytvoření takového automatu je snaha přesunout jednoduché úkony mimo pokladny obsluhované personálem koupaliště. Dobíjení hotovosti na čipové hodinky patří k nejčastějším nabízeným službám, je to snadná a stále se opakující činnost, která pouze zdržuje pokladníky a prodlužuje frontu u vstupu do koupaliště. Odsun alespoň části zákazníků mimo standardní pokladny ulehčí práci personálu a zákazníci stráví méně času ve frontě u vstupu do areálu.

3.3 Návrh dobíjecího automatu

3.3.1 Analýza požadavků

Dle [19] je cílem analýzy požadavků nalézt hranice systému, upřesnit zadání a odhad pracnosti a zachytit omezení kladená na systém. Tyto nároky se obecně dělí dle kategorizace FURPS:

- Functionality - funkcionalita, splnění plánu, bezpečnost
- Usability - použitelnost, složitost využívání systému, lidský faktor
- Reliability - spolehlivost, stabilita systému
- Performance - výkon, doba odezvy, složitost běhu
- Supportability - podpora, rozšiřitelnost

Požadavky se dělí na funkční a nefunkční.

3.3.1.1 Funkční požadavky

Funkční požadavky definují, co má navrhovaný systém umět, jaké služby bude uživatelům nabízet, jaké je jeho chování. Funkční požadavky mají podobu cílů, kterých chce zadavatel prostřednictvím systému dosáhnout.

Hlavním funkčním požadavkem na nový modul systému je usnadnění práce personálu koupaliště u pokladen. Více než třetinu poskytnutých služeb tvoří dobíjení čipových hodinek nebo jen prostě zjištění stavu konta na čipových hodinkách a jejich platnost. Jedná se o snadnou činnost, kde stačí přiložit čip ke čtečce a v programu zvolit částku k dobití. Úkon je to snadný, ovšem zákazníků s tímto požadavkem je mnoho. Vytvořením automatu, který tyto služby zařídí, se až o třetinu sníží pracovní nároky na personál.

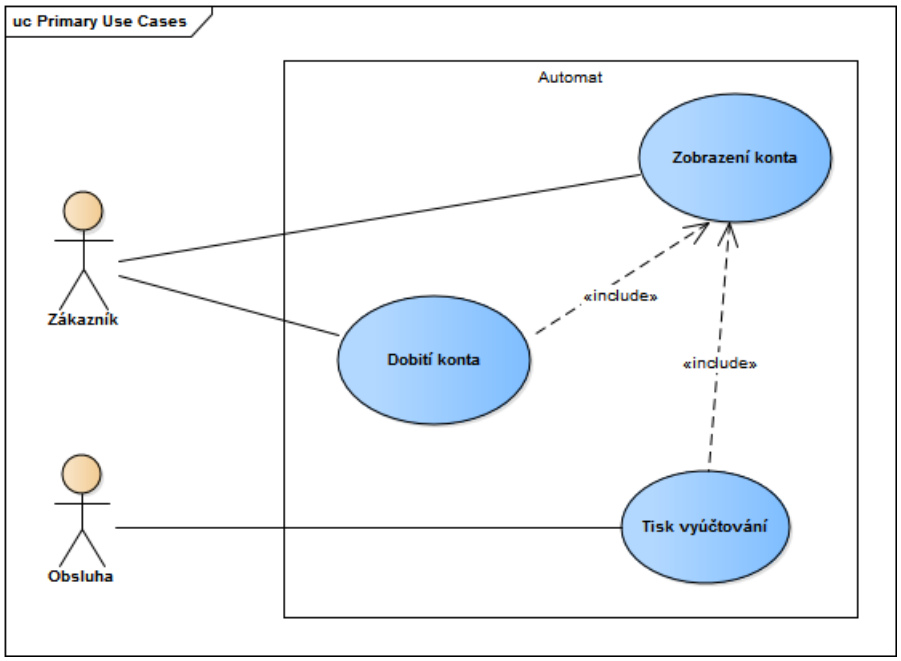
S tím souvisí i další funkční požadavek - urychlení obsluhy zákazníků. Tím, že se až třetina zákazníků přesune k dobíjecímu automatu, ostatní zákazníci budou rychleji obslouženi a vpuštěni do areálu koupaliště.

Třetím funkčním požadavkem je poskytování informací o kontě zákazníka. Automat bude kromě funkce dobíjení ještě navíc zobrazovat informace o stavu, platnosti a majiteli konta. Automat nebude umožňovat měnit záznamy o majiteli konta.

3.3.1.2 Nefunkční požadavky

Nefunkční požadavky kladou omezení na systém, požadavky na architekturu, které musí systém splňovat. Definují nároky na použité technologie a výkon.

Nefunkčním požadavkem na dobíjecí automat je dostupnost 99 % času běhu automatu. Je třeba, aby se na automat dalo spolehnout, zejména v horkých letních dnech, kdy by se začala tvořit fronta i u tohoto automatu.



Obrázek 3.4: Diagram případů užití modulu Automat

3.3.2 Návrh hardwaru automatu

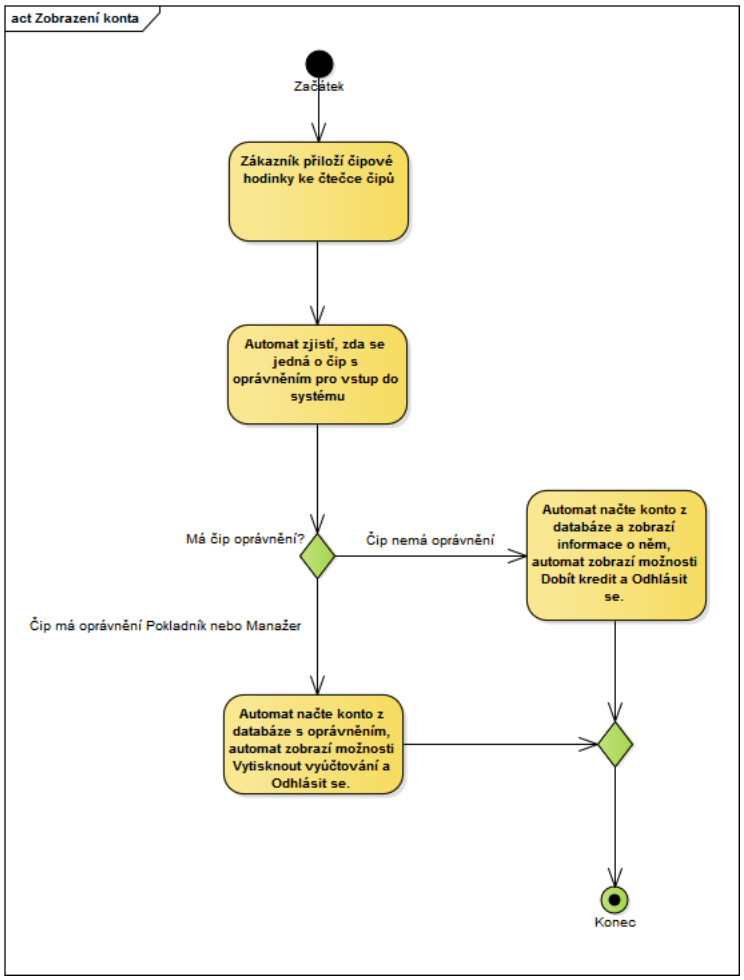
Majitel koupaliště požaduje klasický automat se čtečkou čipových hodinek - čtečka čipu MIFARE MF1ICS50 (ISO/IEC 14443 Type A), umožňující dobíjení elektronického konta prostřednictvím hotovosti a kreditní kartou. To vyžaduje hardware pro rozpoznání mincí a bankovek. Automat musí mít i čtečku bezkontaktních karet a klávesnici pro zadání PINu karty. Pro zákazníky s kontaktní platbou kreditní kartou bude automat nabízet klasický terminál pro kreditní karty. Samozřejmostí je dotykový display zobrazující informace o kontě a možnosti dobíjení. Automat bude disponovat též tiskárnou, po každém úspěšném dobití vytiskne zákazníkovi účtenku stejného formátu, jaký tiskne tiskárna u klasické pokladny.

3.3.3 Model případů užití

V modelu hrají svou roli 2 účastníci - zákazník a obsluha. Zákazník je majitel konta. Obsluha je uživatel systému, který má přidělena práva „Manažer“ nebo „Pokladník“.

Na obrázku 3.4 je zobrazen diagram případů užití nově implementovaného automatu. Nejčastěji se bude jednat o následující případy užití:

- UC1 Zobrazení konta
- UC2 Dobití konta (include Zobrazení konta)



Obrázek 3.5: Diagram aktivit: Zobrazení konta

- UC3 Tisk vyúčtování (include Zobrazení konta)

3.3.3.1 UC1 Zobrazení konta

Případ užití popisuje postup zobrazení informací o stavu konta zákazníka (obrázek diagramu aktivit 3.5).

Účastníci:

- zákazník
- systém

Předpoklady:

- Konto zákazníka není blokováno.

Minimální plnění: Systém získá informace o kontu zákazníka.
Úspěšné plnění: Čtečka z čipu přečte ID konta zákazníka a systém v databázi nalezne následující informace o kontu zákazníka:

- ID konta
- jméno
- příjmení
- typ konta
- kredit
- datum posledního dobití kreditu

Scénář úspěšného plnění:

1. Zákazník přiloží čipové hodinky ke čtečce čipů.
2. Modul systému MifareBlock načte data z čipu - přečte všechny ID karty.
3. Systém vybere správné ID (ID pro daný modul) a dle tohoto čísla v databázi nalezne unikátní záznam.
4. Systém načte záznam a zkontroluje informace (poslední dobití bylo uskutečněno před méně než rokem, konto není blokováno).
5. A. Systém zjistí, že se jedná o čip bez oprávnění.
   a) Systém uloží informace do aplikace.
   b) Aplikace zobrazí informace o kontu a zobrazí možnost dobít kredit nebo se odhlásit.
5. B. Systém zjistí, že se jedná o čip s oprávněním.
   a) Systém uloží informace do aplikace.
   b) Aplikace zobrazí informace o kontu a zobrazí možnost vytisknout denní tržbu nebo se odhlásit.

Výjimky:

- 4. Systém zjistí, že konto bylo naposledy dobito před více než rokem.
  - Systém uloží do aplikace informace o pozastaveném kontě.
  - Aplikace zobrazí hlášku o pozastaveném kontě a nutnosti dobít konto na standardní pokladně.

- 4. Systém zjistí, že je konto blokováno.
  - Systém uloží do aplikace informace o blokování.
  - Aplikace zobrazí hlášku o blokování konta.

Frekvence použití: bez omezení

3.3.3.2 UC2 Dobití konta

Případ užití popisuje způsoby a možnosti dobití částky na konto zákazníka (obrázek diagramu aktivit v Příloze C.6).

Účastníci:

- zákazník
- systém

Předpoklady:

- Konto zákazníka není blokováno.
- Systém má načtené informace o kontě zákazníka.

Minimální plnění: Systém dobije kredit na konto zákazníka.

Úspěšné plnění: Zákazník vloží finanční obnos a systém tuto částku převede na konto zákazníka.

Scénář úspěšného plnění:

1. Zákazník v Aplikaci zvolí možnost Dobít kredit.
2. Aplikace zobrazí zprávu, že automat čeká na vložení hotovosti, vložení kreditní karty či přiložení bezkontaktní kreditní karty.
3. A. Zákazník zvolí platbu hotovostí a vloží hotovost.
   a) Automat zjistí, jaká hotovost byla vložena.
   b) Aplikace zobrazí sumu vložené hotovosti.
   c) Pokud zákazník se sumou souhlasí, zvolí možnost Dobít a scénář pokračuje. Pokud zákazník vloží další hotovost, scénář se vrací na bod a).
   d) Automat uloží v systému sumu dobití kreditu.
   e) Systém uloží v databázi sumu ke kontu s ID, které bylo načteno (viz Předpoklady).
   f) Systém vrátí aplikaci informaci o vyřízení transakce.
   g) Aplikace zobrazí zprávu o vyřízení transakce.
   h) Systém vymaže uložené informace o načteném kontě.

i) Aplikace automaticky odhlásí konto, zobrazí zprávu o čekání na načtení dalšího konta.

3. B. Zákazník zvolí platbu kontaktní kreditní kartou.

a) Aplikace zobrazí obrazovku pro zadání částky pro dobití.

b) Zákazník zadá částku a poté zvolí Potvrdit sumu.

c) Aplikace zobrazí zvolenou částku a možnosti Změnit částku a Zaplatit.

d) Pokud zákazník zvolí možnost Změnit částku, scénář se vrátí na bod a). Pokud zákazník zvolí možnost Zaplatit, scénář pokračuje.

e) Aplikace zobrazí zprávu o čekání na vložení kreditní karty.

f) Zákazník vloží kreditní kartu do terminálu.

g) Terminál kreditních karet se spojí s kreditní kartou a pokusí se o provedení transakce.

h) Terminál kreditních karet zobrazí zprávu o požadování zadání PIN kódu.

i) Zákazník zadá PIN. Pokud je zadán správný PIN, scénář pokračuje. Pokud je zadán špatný PIN, terminál zobrazí zprávu o špatném PINu a tento bod se opakuje. Pokud je potřetí zadán špatný PIN, terminál neprovede transakci, zobrazí informaci o neprovedení transakce, ukončí spojení, Aplikace zobrazí zprávu o neprovedení transakce a scénář pokračuje bodem o).

j) Terminál kreditních karet provede transakci a pošle zprávu systému o úspěšném provedení.

k) Automat uloží v systému částku dobití kreditu.

l) Systém uloží v databázi sumu ke kontu s ID, které bylo načteno (viz Předpoklady).

m) Systém vrátí aplikaci informaci o vyřízení transakce.

n) Aplikace zobrazí zprávu o vyřízení transakce a odebrání kreditní karty z terminálu.

o) Systém vymaže uložené informace o načteném kontě.

p) Aplikace automaticky odhlásí konto, zobrazí zprávu o čekání na načtení dalšího konta.

3. C. Zákazník zvolí platbu bezkontaktní kreditní kartou.

a) Aplikace zobrazí obrazovku pro zadání částky pro dobití.

b) Zákazník zadá částku a poté zvolí Potvrdit sumu.

c) Aplikace zobrazí zvolenou částku a možnosti Změnit částku a Zaplatit.

d) Pokud zákazník zvolí možnost Změnit částku, scénář se vrátí na bod a). Pokud zákazník zvolí možnost Zaplatit, scénář pokračuje.

e) Aplikace zobrazí zprávu o čekání na přiložení kreditní karty.

f) Zákazník přiloží kreditní kartu do terminálu.

g) Terminál kreditních karet se spojí s kreditní kartou a pokusí se o provedení transakce.

h) A. Terminál kreditních karet zobrazí zprávu o požadování autentizace -zadání PIN kódu.

   i. Zákazník zadá PIN. Pokud je zadán správný PIN, scénář pokračuje. Pokud je zadán špatný PIN, terminál zobrazí zprávu o špatném PINu a tento bod se opakuje. Pokud je potřetí zadán špatný PIN, terminál neprovede transakci, zobrazí informaci o neprovedení transakce, ukončí spojení, Aplikace zobrazí zprávu o neprovedení transakce a scénář pokračuje bodem o).

h) B. Terminál nepotřebuje autentizovat kreditní kartu.

i) Terminál kreditních karet provede transakci a pošle zprávu systému o úspěšném provedení.

j) Automat uloží v systému částku dobití kreditu.

k) Systém uloží v databázi sumu ke kontu s ID, které bylo načteno (viz Předpoklady).

l) Systém vrátí aplikaci informaci o vyřízení transakce.

m) Aplikace zobrazí zprávu o vyřízení transakce.

n) Systém vymaže uložené informace o načteném kontě.

o) Aplikace automaticky odhlásí konto, zobrazí zprávu o čekání na načtení dalšího konta.

Výjimky:

- 3. B. j) + 3. C. i) Provedení transakce terminálu kreditních karet skončí chybou (selhání spojení apod.).
  - Terminál kreditních karet zobrazí zprávu o neprovedení transakce.
  - Aplikace zobrazí zprávu o nevyřízení transakce z důvodu selhání spojení.

– Systém vymaže uložené informace o načteném kontě.
– Aplikace automaticky odhlásí konto, zobrazí zprávu o čekání na načtení dalšího konta.

Frekvence použití: bez omezení

3.3.3.3 UC3 Tisk vyúčtování

Případ užití popisuje postup vytvoření dokladu o kompletním vyúčtování za daný den (obrázek diagramu aktivit v Příloze C.7).

Účastníci:

- obsluha
- systém

Předpoklady:

- Konto obsluhy není blokováno.

Minimální plnění: Systém vytiskne sumu tržby za daná den.

Úspěšné plnění: Systém z databáze vybere všechny transakce provedené daný den a vytiskne jejich sumu v daném formátu.

Scénář úspěšného plnění:

1. Obsluha zvolí možnost Tisk vyúčtování.
2. Aplikace dá systému požadovaný příkaz.
3. Systém v databázi vyhledá všechny transakce s datem provedení rovným danému datu a obsluhou, která to provedla, s názvem Automat.
4. Systém načte jednotlivé záznamy a dočasně si je uloží.
5. Po načtení všech záznamům systém sečte všechny transakce jednotlivých typů.
6. Systém pošle jednotlivé sumy aplikaci.
7. Aplikace vytiskne sumy v předem zadaném formátu (formát uložený v nastavení systému).
8. Aplikace zobrazí zprávu o vyřízení tisku.
9. Systém vymaže uložené informace o načteném kontě.
10. Aplikace automaticky odhlásí konto, zobrazí zprávu o čekání na načtení dalšího konta.

Výjimky: žádné

Frekvence použití: bez omezení

3.3.4 Stavový diagram

Zpočátku se automat nachází ve stavu, kdy čeká na přiložení čipových hodinek ke čtečce. Po přiložení načte a zobrazí informace o kontě zákazníka a nabídne zákazníkovi dobít kredit či odhlásit se. Pokud zákazník zvolí možnost dobití konta, automat bude standardně očekávat vhození mincí či bankovek. Dále má zákazník možnost dobíjet prostřednictvím kreditní karty, tuto možnost zvolí na displayi společně s částkou dobíjení. Karta může být přiložena či vložena, tedy může se jednat o bezkontaktní nebo kontaktní platbu, na dobíjení se nic nemění. Dle nastavení banky může být vyžadována autentizace karty pomocí PINu. Pokud tomu tak je, zákazník zadá na klávesnici PIN. V případě, že bude třikrát zadán špatný PIN, dobíjení bude neúspěšné a automat se dostane do stavu zamítnutí operace. V případě úspěšné autentizace bude částka dobita a konto bude automaticky odhlášeno. Z bezpečnostního hlediska k automatickému odhlášení dojde v jakémkoli stavu, pokud uplyne 1 minuta od poslední akce. Pokud je k automatu přiložen čip zaměstnance koupaliště s oprávněním „Pokladní“ či „Manažer“, bude automat umožňovat vytisknout vyúčtování - kompletní soupis tržeb automatu. Stavový diagram je zobrazen na obrázku v Příloze C.8.

3.4 Implementace dobíjecího automatu

3.4.1 SW podpora systému

„Celá HW sestava, která tvoří pružnou stavebnici schopnou se přizpůsobit konkrétním provozním a instalačním podmínkám, je podporována rozsáhlým SW vybavením. Moderní SW platforma pracující v režimu klient server je bezpečným prostředím pro veškeré nadstavbové funkce které jsou od systému očekávány. I SW část se skládá z několika spolupracujících částí a mají specializované zaměření.“[20, s. 6] Skládá se z:

- VapsEng
- PoklEng
- MifareBlock
- TestBox
- speciální moduly

VapsEng je *„real time jádro systému obsluhující v reálném čase veškeré HW části systému. Program kontroluje stav turniketů, terminálů apod., provádí sběr dat, jejich vyhodnocení a odbavuje požadavky, které přicházejí od systému.“* [20, s. 6] PoklEng je SW část systému pro uživatele, především personál pokladen. MifareBlock je modul rozšiřující VapsEng a PoklEng o práci s Mifare bezkontaktními kartami. Program TestBox spravuje šatní skříňky.

Obrázek 3.6: Instance SQL Server

[3.6] Instance SQL Server. In: Seznámení a instalace microsoft sql serveru. DOTNETPORTAL [obrázek]. Tomáš Jecha. [cit. 2016-05-11]. `http://www.dotnetportal.cz/clanek/140/Seznameni-a-instalace-Microsoft-SQL-Serveru`.

3.4.2 Databázový server

Řídící jádro systému je tvořeno počítačem s operačním systémem Microsoft Windows Server 2008. Na počítač byl instalován Microsoft SQL Server. Instalace databázového systému Microsoft SQL Server (zvaného též zkráceně SQL Server) tvoří tzv. instanci. SQL Server umožňuje na počítači vytvořit i několik takových instancí různých či stejných verzí systému. Instance se vzájemně neovlivňují a podporují update. [21]

SQL Server první verze byla vytvořena roku 1989 na systému Sybase. Přelom přišel s verzí 8.0 (rok 2000), bylo přepsáno jádro a přidána možnost zpětné kompatibility. [21]

3.4.3 Implementace SW systému

Programovacím jazykem SW nadstavby je jazyk Delphi. Jedná se o jazyk a IDE, grafické integrované vývojové prostředí, pro tvorbu různých aplikací. Kompilátory IDE jsou psané v jazyce Object Pascal. Programování v Delphi je založeno na používání malý programů vykonávající dané činnosti. Delphi umožňuje vizuální návrh GUI, zdrojový kód je vytvořen automaticky.

3.4.4 Implementace automatu

Automat bude stejně jako aplikace u pokladen využívat MifareBlock pro práci s Mirafe kartami. Bude napojen na real time jádro systému VapsEng i PoklEng. Automat bude v zásadě se systémem komunikovat stejně jako počítače - pokladny. Rozdíl bude v aplikaci a povolených službách.

Aplikace pro automat musí být dostatečně intuitivní ve všechny zákazníky koupaliště. Musí být snadno čitelná a jednoduchá.

Jako ukázku práce s SQL Serverem provedu implementaci funkcí pro automat v jazyku C++. Jedná se o funkce:

- najít konto v databázi,
- přidat na konto částku,
- souhrn všech operací za daný den.

Pro jazyk C++ existují knihovny starající se o práci s MS SQL databázemi. Mezi takové patří například knihovna SQLAPI++. Knihovna vytváří chybějící můstek mezi jazykem C++ a databází MS SQL.

Základem je podpůrná funkce spravující připojení se k databázi.

```
SAConnection con;
con.Connect(
            "automat",     // database name
            "test",        // user name
            "test",        // password
            SA_ODBC_Client);
```

Těmito dvěma příkazy dojde k připojení se k databázi typu ODBC se jménem 'automat'.

Dále automat potřebuje, aby z přiloženého čipu v databázi nalezl majitele tohoto čipu. Tuto funkci provedeme následujícími příkazy.

```
int id_client;
SACommand cmd(&con, "SELECT * FROM zakaznik WHERE id = :1");
cmd.Param(1).setAsLong() = id_client;
while(cmd.FetchNext())
{
    printf("Row: jmeno = '%s', prijmeni = '%s', kredit = %ld\n",
     (const char*)cmd.Field("jmeno").asString(),
     (const char*)cmd.Field("prijmeni").asString(),
     cmd.Field("kredit").asLong());
}
con.Commit();
```

Příkaz `while` vypíše všechny řádky, které mají `id` rovno ID čipu zákazníka. Toto ID bude získáno modulem MifareBlock.

Zákazník by mohl požadovat dobití částky na své konto. To provede následující sekvence příkazů.

```
int id_client;
int money;
SACommand cmd(&con, "UPDATE zakaznik SET kredit = :1
WHERE id = :2");
cmd.Param(1).setAsLong() = money;
cmd.Param(2).setAsLong() = id_client;
cmd.Execute();
con.Commit();
```

Poslední funkce vyhledávající všechny transakce za daný den je implementována stejným principem jako funkce vyhledávající konto zákazníka. Funkce bude vyhledávat transakce SQL příkazem

```
SELECT * FROM zakaznik WHERE date > current_0
AND date < current_24"
```

kde `current_0` a `current_24` jsou časová razítka označující daný den v 0:00 a 23:59.

3.4.5 Další knihovny

Pro práci s MS SQL Serverem existují i další knihovny. Pro jazyk C++ je takovou další knihovnou například knihovna Qt. Knihovna byla navržena pro vývoj grafických aplikací, je multiplatformní a existují i verze pro další nejpoužívanější programovací jazyky. Qt je rozdělena do mnoha modulů, který zpracovávají každý jiný problém. Knihovna disponuje moduly pro práci s SQL, Bluetooth či síťovými prvky, nabízí možnost skriptování pomocí JavaScript, možnost přístupu k portů a mnoho dalších funkcionalit.

3.5 Testování implementace

Implementace byla testována lokálně. Pro otestování jsem použila databázový systém MS SQL Server 2008 R2 a jeho nástroj Management Studio, ve kterém lze pohodlně zadávat SQL příkazy. Pro testování byla vytvořena databáze 'automat' s tabulkou 'zakaznik' s omezeným počtem sloupců. Pro účely testování jsem vytvořila proměnné ID, jméno, příjmení zákazníka, identifikátor druhu čipových hodinek (dospělý nebo dítě) a kredit.

Při testování byla implementace nejprve použita v programu NetBeans IDE. Tento program umí spolupracovat s knihovnami Qt. Qt ovšem na operačním systému MS Windows s NetBeans pracuje pouze ve verzích 4.6.2 a 4.8 a pouze s verzemi NetBeans 6.9 až 7.4. Dalším problémem je nutnost instalace kompilátoru MinGW a to pouze verze 4.4.0 z roku 2009. Při použití na operačním systému Windows by tak mohly často nastávat problémy s kompatibilitou.

Druhým pokusem bylo testování implementace v programu MS Visual Studio 2013. V tomto případě bylo vhodné využít knihovnu SQLAPI++, která se specializuje na použití v několika IDE. SQLAPI++ též komunikuje s více druhy databází, optimisticky vzato dokonce se všemi často používanými databázemi. V případě změny databáze by pak nenastal problém kompatibility.

Testovány byly všechny požadované funkce. Pro testování u zákazníka je třeba ošetřit propojení již implementovaných modulů s modulem nově navrhovaného automatu. Zejména propojení modulu spravujího komunikaci mezi čipy a softwarem vyžaduje zvláštní pozornost při testování.

3.6 Zhodnocení bezpečnosti

3.6.1 Bezpečnost systému

3.6.1.1 Organizační opatření

Koupaliště Flošna vydalo své vlastní směrnice a poučení o bezpečnosti při práci se systémem. Každý uživatel je s těmito pravidly seznámen každý rok.

3.6.1.2 Fyzická opatření

Proti výpadku elektřiny je server chráněn záložním zdrojem. Fyzicky je celý areál (včetně hardwaru systému) chráněn před neoprávněným vstupem ocelovou svinovací roletou v případě zavřeného koupaliště a turnikety v případě otevřeného koupaliště. Celý areál je monitorován bezpečnostními kamerami. Kamery jsou po celý rok 24 hodin denně sledovány personálem koupaliště. Serverovna je stále zamčená bezpečnostní cylindrickou vložkou FAB. Ostatní místnosti systému jsou zamčené nebo pod dohledem školeného personálu.

3.6.1.3 Technická opatření

Systém je v případě potřeby pod dohledem servisu společnosti IVAR, a.s.

3.6.1.4 Programová opatření

Systém loguje jakoukoli jeho změnu či přístup do aplikace, logují se i všechny tržby a jiné změny na kontech zákazníků. Přístup do aplikace je umožněn pouze s přístupovým jménem a heslem (přes čipy). Tyto čipy s přístupem k aplikaci (čipy, jejichž uživatel má umožněno vykonat jakoukoli změnu na systému či v databázi, změnu, která se bude logovat) vlastní pouze proškolený personál, případně jim jsou takové čipy půjčeny na dobu pracovní směny.

Jednotlivé čipy se rozlišují dle uživatelských práv. V systému jsou vytvořeny výše zmíněné 4 úrovně oprávnění. Uživatelé díky tomu mohou provést jen změny nezbytné k práci daného zaměstnance, čímž je automaticky vyvarují vážnějších chyb.

3.6.1.5 Šifrování

Šifrování je využíváno zejména při používání čipových hodinek. Při komunikaci mezi čtečkou a čipovými hodinkami se využívá šifrovacího algoritmu Crypto1. Crypto1 je proudová šifra využívající LFSR. Samotné použití LFSR nezaručuje dostatečnou ochranu dat. Pro zvýšení bezpečnosti se využívají různé metody, které způsobí zrušení lineárnosti a tím i předvídatelnosti algoritmu. Algoritmus Crypto1 využívá navíc nelineární kombinační funkci, proto je bezpečnost zajištěna.

3.6.1.6 Zálohování

Na bezpečnost ze strany zálohování je tu kladen největší důraz. Celý systém rekreačních areálu Hradce Králové disponuje dvěma servery, které se každých 15 minut vzájemně kontrolují kvůli změnám a případně se zrcadlově zduplikují. V případě výpadku jednoho ze serverů je druhý schopen převzít řízení celého systému.

3.6.1.7 Antivirová ochrana

Koupaliště využívá standardních komerčních produktů společnosti ESET.

3.6.2 Bezpečnost automatu

Hlavním bezpečnostním prvkem automatu je automatické odhlašování konta zákazníka. Ačkoliv by při navrhované implementaci automatu nemohlo dojít k odečtu jakékoli částky z konta, stále by se jednalo o narušení soukromí zákazníka. Navržený automat by automaticky odhlašoval přihlášeného zákazníka po 1 minutě nečinnosti.

Pokud by ale vedení koupaliště do budoucna zvažovalo více operací prováděných automatem (například placení vstupenek na koupaliště atd.), jednalo by se i o změny kreditu konta o záporné sumy a byla by tak potřeba potvrdit takové platby opětovným přiložením čipových hodinek. Pokud by se použitý čip u prvního i druhého přiložení shodoval, pak by byla operace provedena. Teoreticky by se mohlo stát, že se první zákazník neodhlásí, během jedné minuty se k automatu dostane další zákazník a provede platbu z konta prvního zákazníka. Při nutnosti potvrzení platby by taková situace nenastala.

Dalším bezpečnostním opatřením je možnost instalace senzoru pohybu a jeho nastavení na sledování správné vzdálenosti. Senzor pohybu by kontroloval, jak dlouho zákazník nestojí u automatu a po určité době by případně přihlášené konto zákazníka automaticky odhlásil.

Automat bude dále chráněn monitorováním bezpečnostní kamerou.

Další bezpečnostní prvky vyplývají z používání čipů a jejich bezpečnosti.

3.7 Srovnání stávajícího a navrhovaného řešení

Stávající řešení je plnohodnotný funkční systém. Je testovaný nejen na daném koupališti, ale i v jiných areálech po celé republice. Funkční systém byl nainstalován včetně hardwaru (server, turnikety, čtečky čipů, čipy). V části Wellnessu systém kontroluje ovládání šatních skříněk, je zde umístěna samostatná čtečka čipů pro zobrazení čísla skříňky a její odemčení. Čipové hodinky tedy umožňují i dočasné držení informace o přidělené skříňce, jejíž číslo zadá personál při vstupu. Jedny čipové hodinky - jedno konto - jeden kredit je tedy platný pro vstupy do několika zařízení po celém Hradci Králové, včetně Restaurace Flošna. Nemenší výhodou stávajícího řešení je to, že personál si již na práci se systémem zvykl a je zbytečné školit ho na nový.

Naopak nevýhodami současného systému jsou menší nedostatky v řešení přihlašování uživatelů a neshodné verze hlavních modulů.

Navrhované řešení se liší v optimalizaci stávajícího a implementaci nového modulu. Optimalizace současného řešení zahrnuje úpravu kontrolování přihlašování uživatelů.

Návrh automatu je požadavkem vedoucího koupaliště. Hlavním účelem automatu je zjednodušení a zrychlení obsluhy zákazníků. Automat musí primárně umět dobíjet částku na konto zákazníka a to jak hotovostí, tak kreditní kartou. Samozřejmostí je čtečka čipových hodinek. Automat se tak stane součástí systému, dalším modulem, který hromadí tržbu. Z toho vyplývá, že automat musí umět vytisknout sumu tržby za daný den pouze po přihlášení zaměstnaneckého čipu.

Srovnávat stávající a navrhované řešení není ideální, neboť navrhovaným řešením je pouhá optimalizace stávajícího. Optimalizace by pak byla jistě lepším řešením. Návrh automatu je též pouze vylepšením stávajícího řešení.

Lepším srovnáním je komparace stávajícího řešení, tedy systému vytvářeného pro libovolné zařízení sportovně-kulturního účelu, a řešení vytvořeného na míru požadavkům koupaliště. Výše v práci jsou zmíněny výhody a nevýhody obou řešení. Po provedení návrhu optimalizací je znát, že stávající řešení plně vyhovuje požadavkům koupaliště. Navíc byl systém instalován společně s potřebným hardwarem, což je jistě velká výhoda z hlediska toho, že není nutné jednotlivé komponenty obstarávat a zabývat se jejich vzájemnou kompatibilitou. Nedostatky jsou jen menšími, tak říkajíc kosmetickými vadami na jinak více než dostačujícím systému. Výsledkem srovnání je závěr, že stávající řešení s menší optimalizací a rozšířením v podobě automatu je nejvhodnější.

Závěr

Cílem mé bakalářské práce bylo optimalizovat stávající řešení a zefektivnit jeho používání. Byla provedena analýza současného systému, který implementovala společnost IVAR, a.s. Řešení společnosti bylo navrženo pro sportovně-kulturní zařízení obecně. Jedná se o produkt VAPS, vstupenkový a pokladní odbavovací systém pro plavecké bazény. Produkt nabízí kromě softwaru i hardware v podobě serverů, turniketů či zařízení pro identifikaci osob.

Vedení koupaliště je s tímto řešením spokojeno, společnost IVAR pravidelně dodává aktualizace a i nový personál se se systémem snadno naučí pracovat. Výhodou tohoto systému je jeho stavebnicové řešení, tedy možnost k současnému stavu systému připojit další modul bez větších komplikací. Této vlastnosti jsem využila při návrhu dalšího modulu systému - dobíjecího automatu.

Dobíjecí automat byl implementován na žádost vedoucího koupaliště za účelem zrychlení obsluhy zákazníků. Automat byl navržen pro dobíjení čipových hodinek zákazníků, je plánováno umístění čtečky čipů, terminálu kreditních karet a tiskárny účtenek. Automat bude jako všechny ostatní samostatné moduly připojen k serveru. Funkce automatu byly lokálně otestovány.

Dalším cílem práce bylo zhodnotit bezpečnost stávajícího systému a navrhovaného automatu. Zjistila jsem, že hlavními bezpečnostními prvky systému jsou zálohování mezi dvěma servery a používání bezpečných šifrovacích metod při manipulaci s údaji zákazníků. Tato opatření využívá i navrhovaný automat. Dobíjecí automat je pro zákazníky samoobslužný, zajištění bezpečnosti tohoto modulu bylo základním požadavkem na tento modul.

Řešení s optimalizací v podobě přidání modulu dobíjecího automatu lze snadno uplatnit v praxi. Modul komunikuje se serverem, využívá stejný typ čtečky čipů jako ostatní moduly a pracuje s klasickým terminálem pro kreditní karty.

Práci jsem nabídla vedení koupaliště k projednání o rozšíření systému. Do budoucna by bylo možné rozšířit mé řešení automatu o další funkce jako placení vstupenek z konta zákazníka apod. Toto rozšíření by bylo třeba navíc

ošetřit z hlediska bezpečnosti, protože by se jednalo o manipulaci s konty o záporné částky. Bylo by potřeba kontrolovat, zda operaci provádí majitel hodinek. Bylo by též vhodné v budoucnu implementovat menší optimalizace v již funkčních modulech. Jedná se o optimalizace v části Návrh optimalizací. Za nejdůležitější bych považovala implementaci kontroly přihlašování stejných uživatelů ve více instancích aplikace.

Práce mi poskytla pohled na bezpečnost systému z širšího a praktičtějšího hlediska. Přinesla mi nové poznatky z oboru čipů a zabezpečení pomocí čipových karet. Do budoucna bych ráda v práci pokračovala rozšířením automatu o další funkcionality a zvýšením bezpečnosti celého systému.

Literatura

[1] Richta, K.: Metodiky vývoje software, MDA. duben LS 2010/2011, ČVUT, Praha, Přednáška BI-SI1.

[2] TutorialsPoint: UML - Tutorial. `http://www.tutorialspoint.com/uml/index.htm`, cit. 2016-05-02.

[3] Systems, S.: The Unified Modeling Language (UML). `http://www.sparxsystems.com.au/platforms/uml.html`, cit. 2016-05-02.

[4] developerWorks, I.: UML basics: An introduction to the Unified Modeling Language. `http://www.ibm.com/developerworks/rational/library/769.html`, cit. 2016-05-02.

[5] TutorialsPoint: UML - Modeling Types. `http://www.tutorialspoint.com/uml/uml_modeling_types.htm`, cit. 2016-05-02.

[6] Cockburn, A.: *Writing effective use cases*. Boston: Addison-Wesley, 2001, iSBN 0-201-70225-8.

[7] TutorialsPoint: UML - Use Case Diagrams. `http://www.tutorialspoint.com/uml/uml_use_case_diagram.htm`, cit. 2016-05-02.

[8] TutorialsPoint: UML - Activity Diagrams. `http://www.tutorialspoint.com/uml/uml_activity_diagram.htm`, cit. 2016-05-15.

[9] Dobda, L.: *Ochrana dat v informačních systémech*. Praha: Grada Publishing, vydání 1. vydání, 1998, iSBN 80-7169-479-7.

[10] Mirko Novák, V. Š., Zdeněk Votruba: *Bezpečnost a spolehlivost systémů*. Praha: Vydavatelství ČVUT, vyd. 2. přeprac. vydání, 2003, iSBN 80-010-2807-0.

[11] Lórencz, R.: Informační bezpečnost. květen LS 2014/2015, ČVUT, Praha, Přednáška BI-BEZ.

[12] Petr Hanáček, J. S.: *Bezpečnost informačních systémů*. Praha: Úřad pro státní informační systém, první vydání, 2000, iSBN 80-238-5400-3.

[13] Ing. Ivo Rosol, C.: Čipové karty. http://www.systemonline.cz/it-security/cipove-karty.htm, cit. 2016-05-12.

[14] Buček, J.: 11. Úvod do zabezpečení pomocí čipových karet. květen LS 2015/2016, ČVUT, Praha, Přednáška BI-BEZ.

[15] B.V., N.: MF1ICS50. http://www.nxp.com/documents/data_sheet/M001053_MF1ICS50_rev5_3.pdf, 2008-01-29, cit. 2016-05-13.

[16] Nohl, K.: Cryptanalysis of Crypto-1. https://www.cs.virginia.edu/~kn5f/Mifare.Cryptanalysis.htm, 2008-01-29, cit. 2016-05-13.

[17] Sanfelix, E.: Crypto Series: Mifare Crypto1. http://www.limited-entropy.com/crypto-series-mifare-crypto1/, 2009-09-11, cit. 2016-05-13.

[18] IVAR: VAPS - Modulární vstupenkový a pokladní systém. http://info.ivar.cz/rs/aplikace-pro-sportovni-arealy/odbavovaci-system-pro-stadiony-a-sportovni-haly/, cit. 2015-12-01.

[19] Mlejnek, J.: Analýza a sběr požadavků. říjen ZS 2015/2016, ČVUT, Praha, Přednáška BI-SI1.

[20] Škopec, A.: Předmět zprávy: RE: VAPS system - technicke detaily. E-mailová zpráva, 2016-05-05, cit. 2016-05-11.

[21] Jecha, T.: Seznámení a instalace microsoft sql serveru. DOT-NETPORTAL. http://www.dotnetportal.cz/clanek/140/Seznameni-a-instalace-Microsoft-SQL-Serveru, cit. 2016-05-11.

Příloha A

Seznam použitých zkratek

UML Unified Modeling Language

FURPS kategorizace požadavků

IT informační technologie

IS informační systém

GUI Graphical User Interface

UP Unified Process

MSA Modern Structured Analysis

UC Use Case

SW software

HW hardware

MS Microsoft

IDE Integrated Development Environment

EEPROM Electrically Erasable Programmable Read-Only Memory

LFSR Linear Feedback Shift Register

RFID Radio Frequency Identification

Příloha B

Seznam použitých softwarů

Texmaker

Enterprise Architect

Microsoft SQL Server 2008 R2 Management Studio Express

NetBeans IDE 8.0.1

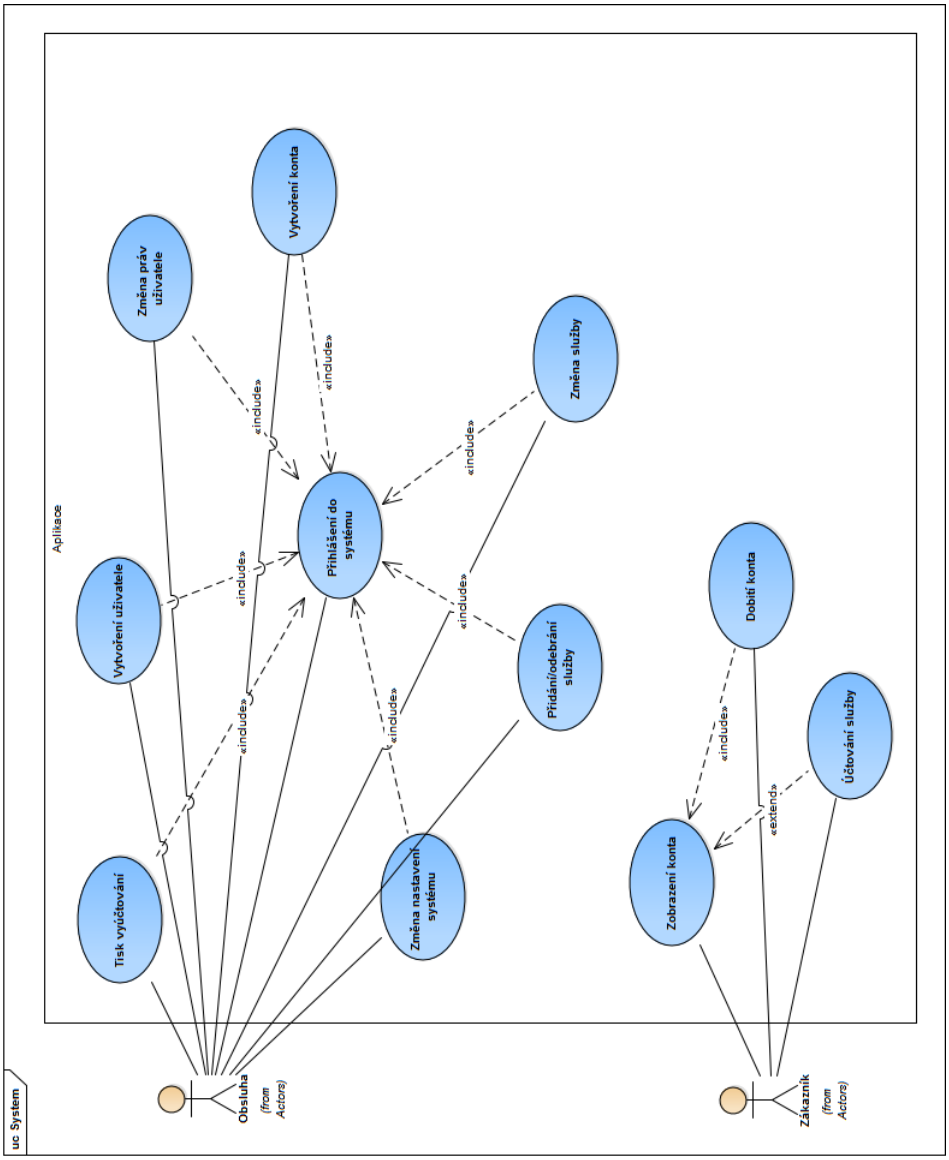
Microsoft Visual Studio 2013

SQLAPI++

Qt 4.6.4

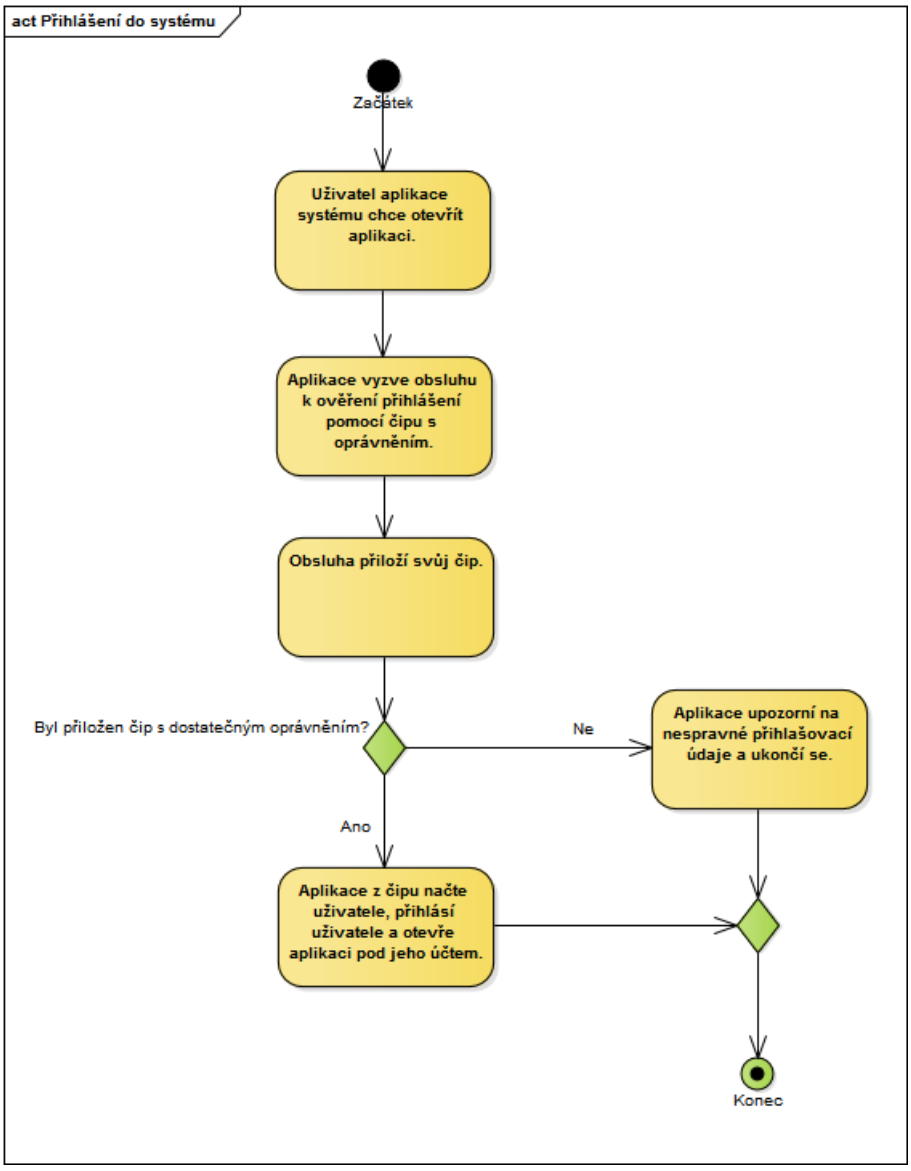
Příloha C

Přílohy

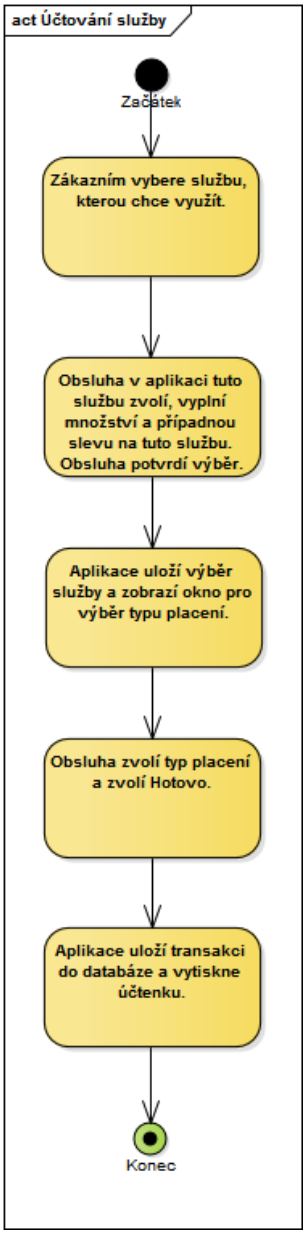


Obrázek C.1: Diagram případů užití modulu Flošna

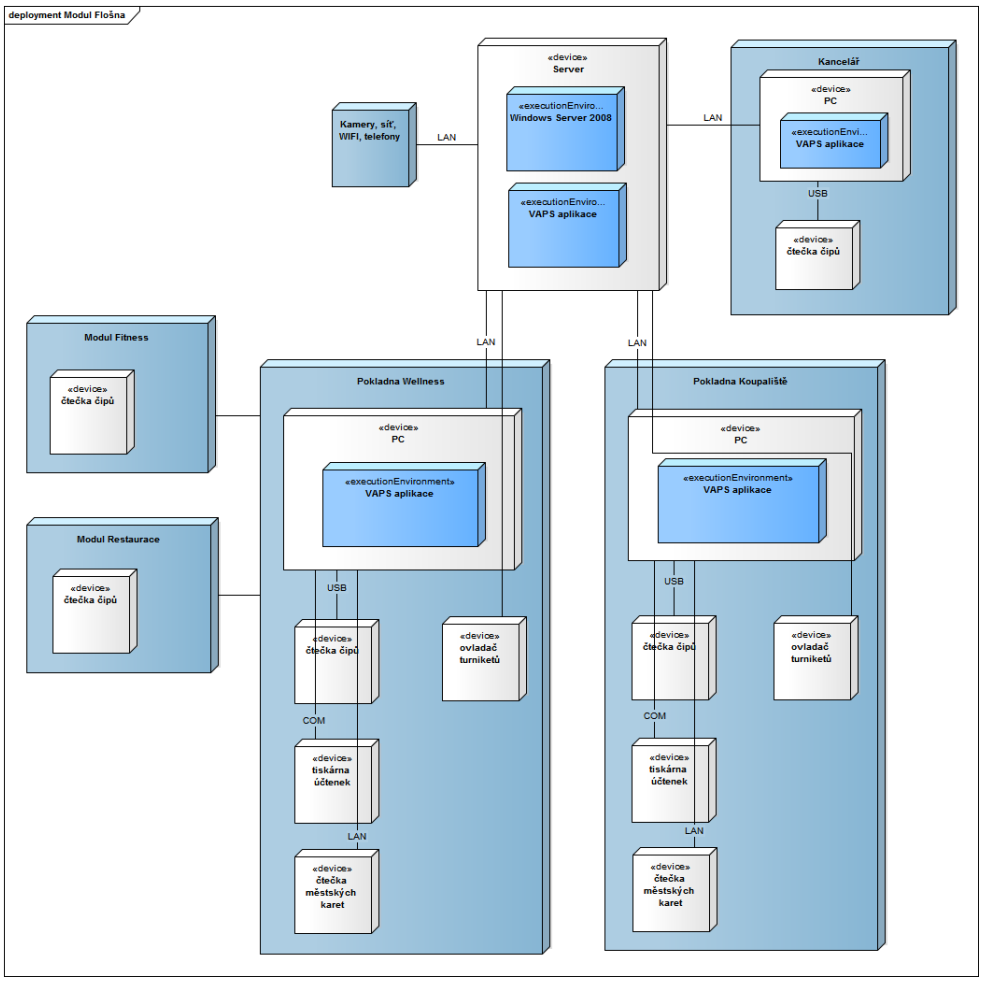
Obrázek C.2: Diagram aktivit: Dobití konta



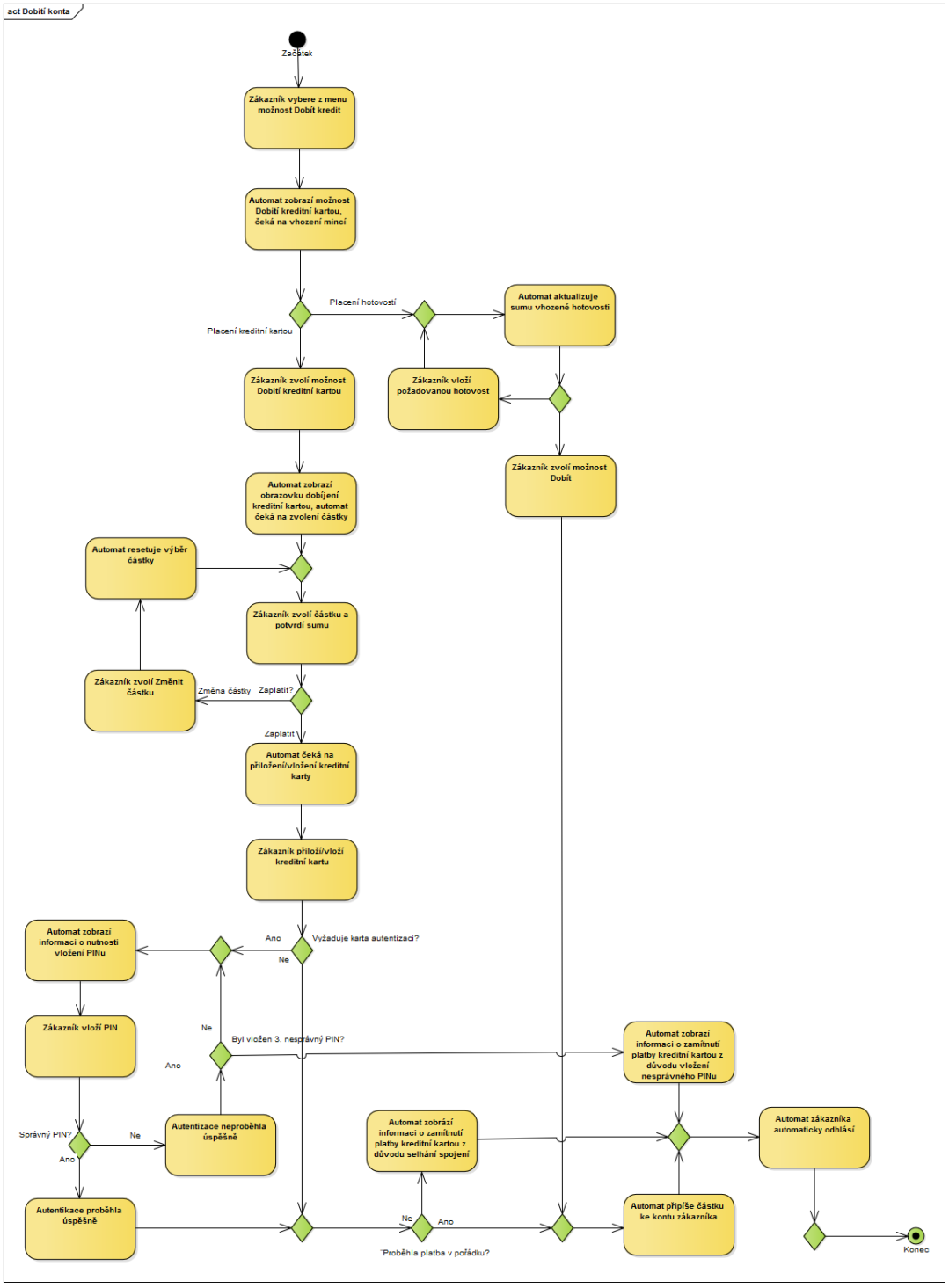
Obrázek C.3: Diagram aktivit: Přihlášení do systému



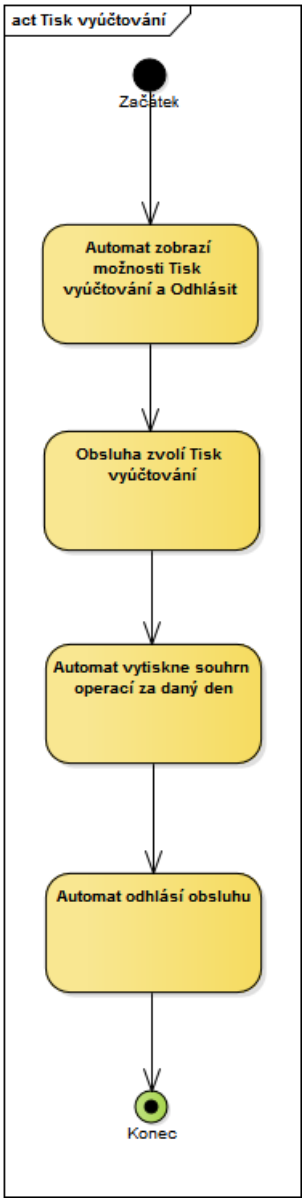
Obrázek C.4: Diagram aktivit: Účtování služby



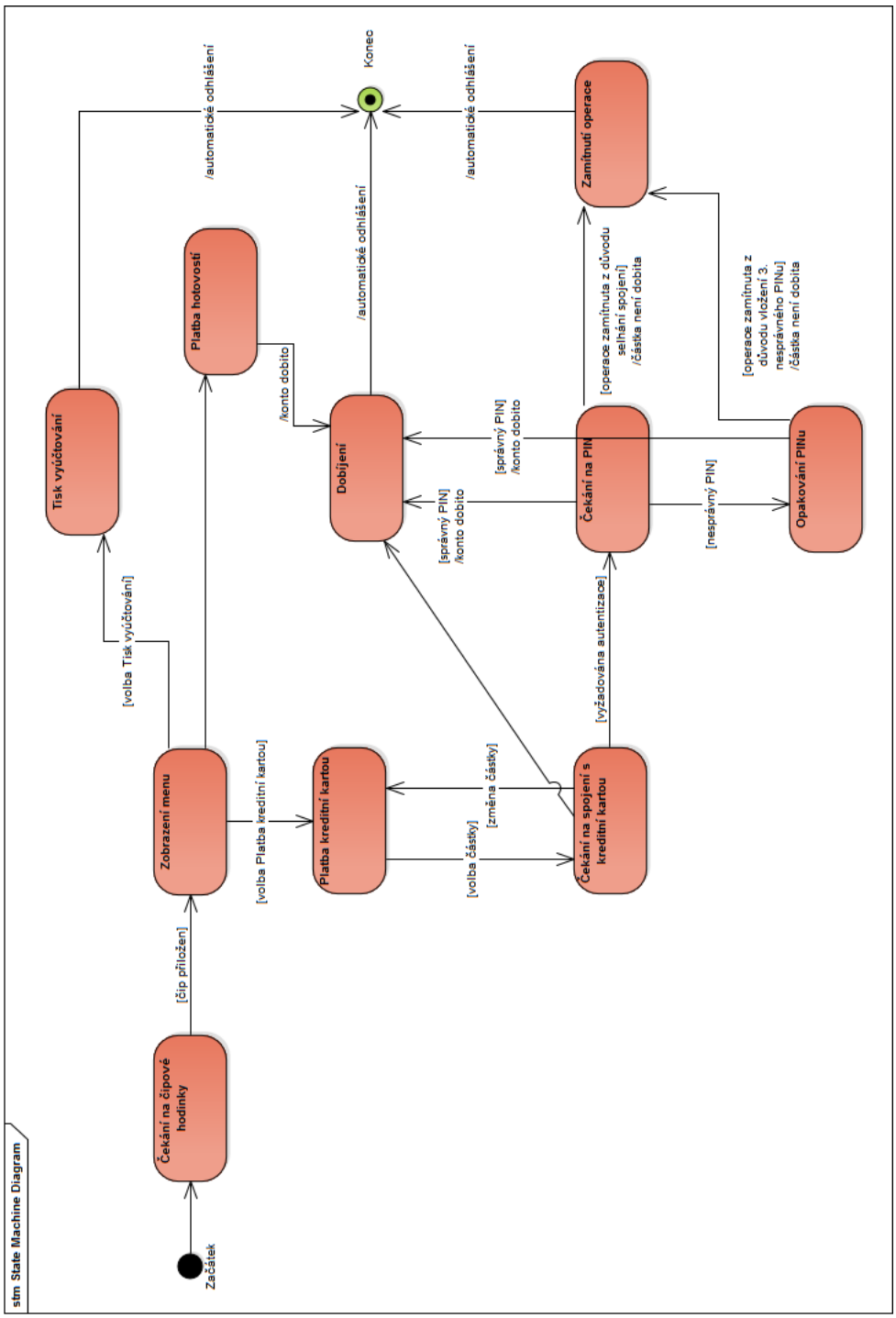
Obrázek C.5: Diagram nasazení modulu Flošna



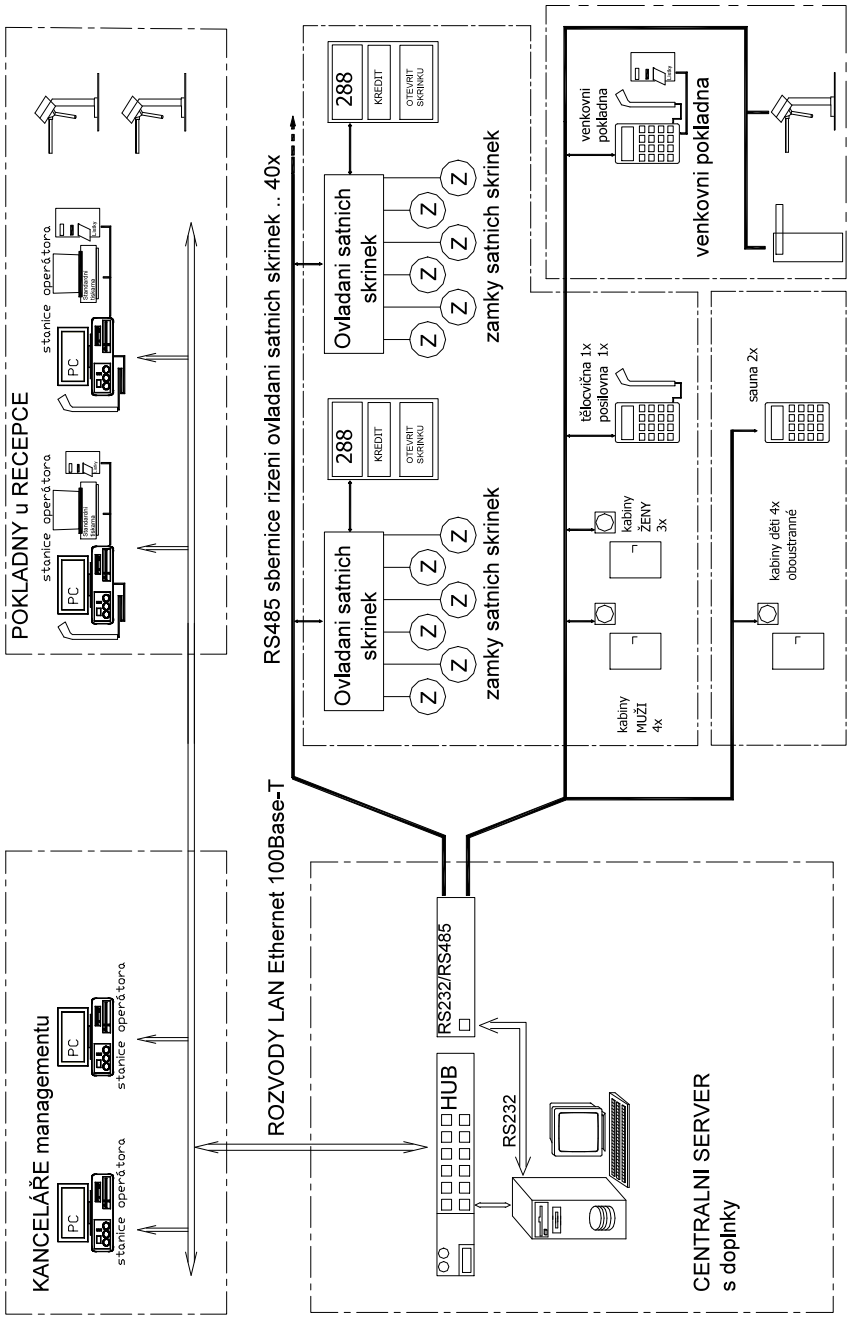
Obrázek C.6: Diagram aktivit: Dobití konta



Obrázek C.7: Diagram aktivit: Tisk vyúčtování



Obrázek C.8: Stavový diagram automatu



Obrázek C.9: Blokové modelové schéma systému IVAR

[C.9] Blokové modelové schéma systému. In: Předmět zprávy: RE: VAPS system - technicke detaily. Emailová zpráva. [obrázek]. Antonín Škopec, 2016-05-05. [cit. 2016-05-11]. `http://www.dotnetportal.cz/clanek/140/Seznameni-a-instalace-Microsoft-SQL-Serveru`.

Příloha D

Obsah přiloženého CD

```
readme.txt ..... stručný popis obsahu CD
src
    impl ..... zdrojový kód implementace
    thesis ..... zdrojová forma práce ve formátu LaTeX
image ..... adresář s obrázky z práce
text ..... text práce
    thesis.pdf ..... text práce ve formátu PDF
```