Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Nástroj pro integraci uživatelských účtů

Plzeň 2014 David Janíček

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 8. května 2014

David Janíček

Abstrakt

Cíle bakalářské práce jsou eliminace redundantního zadávání dat o uživatelích, optimalizace správy integrovaných aplikací a minimalizace rizika vzniku chyby zapříčiněné lidským faktorem, to vše v IT prostředí zákazníka. Těchto cílů bude dosaženo pomocí konsolidace uživatelských dat v jednotlivých aplikacích a automatizací správy uživatelských účtů, tj. zakládání, aktualizace a rušení uživatelských účtů v jednotlivých aplikacích. Bude vytvořen softwarový nástroj, jehož implementace zajistí komunikaci aplikací prostřednictvím definovaných datových rozhraní a transformaci dat do požadovaných formátů. Konsolidovaná uživatelská data budou uložena v databázi integračního SW a bude možné je spravovat prostřednictvím webového rozhraní. Výrazným přínosem je, že konsolidace bude možná vytvořením nového rozhraní na integračním softwaru, čímž je zajištěna dlouhodobá udržitelnost spravovaného prostředí.

Abstract

The aim of this bachelor thesis is the elemination of redundant user data insertion, the management optimalization of the particular applications and the risk minimalization of errors made by human, all of it in the customers's environment. These goals will be achieved by the consolidation of user data stored in the applications and by the automatization of user accounts lifecycle in the applications, specifically creation, update and deletion of the accounts in the particular applications. An software tool will be created and it's implementation will maintain communication among the applications through the defined data interfaces and will transform data to the required formats. The consolidated user data will be stored in the database of the implemented software and there will be a possibility to manage data through a web interface. The main contribution will be the consolidation through the formation of the new interface in the software tool which will provide longterm maintainability of the managed environment.

Obsah

Seznam obrázků

Seznam tabulek

1 | Úvod

Cílem této bakalářské práce je navrhnout, implementovat, otestovat a nasadit produkční verzi nástroje pro integraci uživatelských účtů, který se bude starat o synchronizaci uživatelských dat mezi vybranými aplikacemi u konkrétního zákazníka. Nástroj by měl být přizpůsoben speciálním potřebám zákazníka, na druhou stranu by měla být znatelná snaha navrhnout strukturu programu obecně a nezávisle na specifickém prostředí odběratele, aby se dal snadno a rychle aplikovat jinde. Z toho plyne, že nástroj není považován za jednorázový a v budoucnu je předpokládáno jeho nasazení i u jiných zákazníků.

Bakalářská práce byla vytvářena pro externí firmu KOBOZ SERVICE, s.r.o.[1], která realizuje nasazení integračního nástroje jako jednu z částí svého projektu pro Vyšší odbornou školu a Střední školu veterinární, zemědělskou a zdravotnickou v Třebíči[2] (dále jen škola, zákazník). Cílem bakalářské práce bylo úspěšně provést tuto část projektu.

Před samotným návrhem nástroje je nejprve nutné se detailně seznámit s aktuálním řešením u zákazníka, definovat a prozkoumat aplikace účastnící se integrace, porozumět stávajícímu procesu toku dat a identifikovat jeho chyby a problémy. Postup analýzy je popsán v kapitole 2 Analýza prostředí. Po analýze prostředí zákazníka jsou definovány algoritmy výměny dat a navrhnut nový proces fungování integrace aplikací a synchronizace dat s využitím vyvíjeného projektu v kapitole 3 Návrh řešení. Následuje fáze samotné implementace aplikace, to znamená popis použitých nástrojů pro vývoj, jak implementace probíhala a hlavně jak byly aplikace definované v analýze integrovány a napojeny na celý proces. Těmto záležitostem se věnují kapitoly 4 Nástroje pro implementaci, 5 Fáze implementace a 6 Integrace aplikací. Nakonec je shrnut přínos celého projektu v kapitole 7 Zhodnocení.

[1] domovská adresa `http://koboz.cz/`

[2] více informací o škole na adrese `http://www.szstrebic.cz/`

1.1 Motivace

Hlavní motivací pro realizaci této bakalářské práce je právě možnost spolupráce s uvedenou externí firmou, protože se nejedná o teoretický školní projekt, ale o vývoj skutečné použitelné aplikace, která bude fungovat v reálném prostředí a měla by přinést pro zákazníka značné ulehčení a zlepšení procesu zadávání uživatelských dat. S problémy tohoto typu se nesetkává pouze tento konkrétní zákazník, proto by mohly být zajímavé vyhlídky budoucí možné expanze s projektem i do některých jiných školských prostředí.

2 | Analýza prostředí

Analyzování současného stavu synchronizace dat, softwarový a datový audit se uskutečnil přímo v prostředí zákazníka, kterým je Vyšší odborná škola a Střední škola veterinární, zemědělská a zdravotnická Třebíč, ve středu 12. června 2013.

2.1 Integrované aplikace

Proces definování integrovaných aplikací probíhal v součinnosti s IT pracovníkem třebíčské školy, a to na základě smlouvy mezi firmou KOBOZ SERVICE, s.r.o. a školou, kde už jsou vyjmenovány aplikace, jež mají být součástí integrace. Ke každé aplikaci bylo potřeba získat přístupové údaje, využívané poté i pro integrační nástroj, specifikovat obsažená data a jejich souvislost s daty v jiných softwarech. Zároveň bylo potřeba zjistit, jestli má daná aplikace sloužit jako producent dat, tzn. data jsou aktuální a použitelná v ostatních systémech, nebo jako konzument dat, tj. data jsou kopií dat z jiného zdroje a nepřinášejí žádné nové informace. Uvedený IT pracovník byl pro tento proces nepostradatelnou součástí, nikdo jiný neměl na škole v daném okamžiku lepší přehled o využívaných softwarech a stavu dat. Z analýzy vyplynuly následující aplikace.

2.1.1 Bakaláři

Software Bakaláři je určen pro školní administrativu, pro účely této práce pak zvláště jeho sekce Evidence žáků a zaměstnanců. Webové stránky aplikace uvádějí, že zpracovává vedle osobních údajů zejména klasifikaci žáků. Propracovaný systém zápisu a účinných kontrol dovoluje udržovat data v lepším stavu než klasická ruční evidence. Z karty žáka pak lze vyčíst veškeré potřebné informace - osobní údaje, údaje o rodičích, kompletní klasifikaci za celou školní docházku a podobně.[3]

Kromě osobních údajů žáků a klasifikace software obsahuje zařazení žáků do tříd a jednotlivých skupin třídy (např. rozdělení chlapci/děvčata) s vazbou na třídní učitele, a v neposlední řadě také osobní údaje o učitelích včetně jejich aprobací, pracovního zařazení a jiné. Navíc má systém propracovaný model archivace a zneplatnění uživatelů. Data v programu Bakaláři ručně vyplňuje asistentka vždy v okamžiku, kdy dojde k určité změně - nabírají se noví žáci, studenti nebo zaměstnanci, osoba školu z různých důvodů opustí nebo dojde ke změně osobních údajů. Mimo asistentky má přístup do administrace Bakalářů už jen IT pracovník.

Data v aplikaci byla identifikována jako úplná a především aktuální, tím se myslí, že žádný jiný systém na škole neměl adekvátně ucelené údaje jako Bakaláři. Výjimkou byla čísla čipových karet důležitých k činnostem jako je tisk, vydávání obědů v jídelně a docházkového systému pro zaměstnance (viz sekce 2.1.5). Informace tohoto typu se v softwaru vůbec nevyskytovaly, v základním nastavení programu dokonce nebylo ani nalezeno odpovídající pole, kam by se mohly ukládat. Bakaláři ovšem nabízejí funkcionalitu přidání atributů do databáze, bylo tedy rozhodnuto, že se vytvoří atribut *"číslo čipu"* a data budou ručně doplněna, v první fázi primárně pro učitele - tisk se totiž týká hlavně jich. Doplněním se dosáhlo i žádoucího vedlejšího efektu kontroly správnosti a aktuálnosti čipů.[3]

Škola disponuje verzí programu *13/14* s databází uloženou v datových souborech, takzvaných **DBF** (dBase file) souborech. Neprobíhá žádný import do aplikace Bakaláři, data jsou vkládána pouze ručně, tj. neexistovalo napojení na jiné systémy školy. Po doplnění čipů obsahují Bakaláři kompletní potřebná data. Program byl proto zvolen jako primární producent dat, z pohledu integračního procesu nebude sloužit jako konzument dat, protože již není nutné vkládat žádné jiné informace.

Analýza softwaru Bakaláři probíhala formou ručního procházení databázových souborů (DBF), k prohlížení bylo nutné nainstalovat program pro čtení DBF formátu. Bylo prozkoumáno několik desítek menších i větších souborů, nakonec bylo zjištěno, že údaje potřebné k synchronizaci se nachází v celkem 5 hlavních tabulkách (UCITELE, ZACI, TRIDY, SKUPINY, WEB). V integrační aplikaci bude nutné naprogramovat čtení z těchto DBF souborů, protože Bakaláři neposkytují výstup ve strojově zpracovatelné podobě.

[3] více informací o přidání pole v Bakalářích na `http://www.bakalari.cz/znalosti/znalosti.aspx?l=ed1&p=AIEAA&m=EVID&o=KONFIG`

2.1.2 Active Directory

Active Directory (AD) je adresářová služba obsažená v systému Windows Server. Active Directory zahrnuje adresář, v němž jsou uloženy informace o distribuovaných prostředcích, stejně jako o službách, díky nimž jsou tyto informace užitečné a dostupné. Všechny verze systému Windows Server od systému Windows 2000 podporují Active Directory.[9, str. 23] Active Directory implementuje protokol LDAP (Lightweight Directory Access Protocol).

Původní stav počtu dat ve službě byl velmi nízký, Active Directory bylo používáno z pohledu zaměstnanců, žáků a studentů jen k přihlašování se v počítačových učebnách (respektive k pracovním stanicím) do místní sítě, takže se vystačilo s údaji přihlašovací jméno (v Active Directory atribut `sAMAccountName`), heslo (`unicodepwd`), jméno (`givenName`), příjmení (`sn`) a údajem do kdy je účet aktivní. Účty žáků a studentů byly ručně přidávány jako členové do globální skupiny `studenti`. U učitelů byl navíc vyplněn údaj e-mailová adresa (`mail`) sloužící k napojení na Microsoft Exchange Server, službu poskytující e-mailovou komunikaci. Učitelé byli členové globální skupiny `ucitele`. Ke každému účtu přísluší domovský adresář, tj. na serverovém úložišti existuje dedikovaný adresář, do kterého má přístup pouze daný uživatel a administrátor, uživatelé zde uchovávají svá osobní data.

Z hlediska pouhého ověřování vůči službě Active Directory by se vystačilo s hodnotami, které se vyskytovaly v původním stavu. Active Directory je ale velmi rozšířený nástroj, umožňuje pojmout mnohonásobně větší množství informací a hlavně mnoho aplikací na něj poskytuje napojení (integraci), v rámci tohoto projektu se jedná právě o aplikace SafeQ a Moodle, jak je podrobněji popsáno v dalších sekcích. Z tohoto důvodu slouží Active Directory zároveň jako primární konzument dat z integrační aplikace i jako producent dat (jakýsi přenašeč) pro aplikace Moodle a SafeQ, kvůli kterým bude potřeba synchronizovat do Active Directory mnohem více dat.

Škola v Třebíči disponuje doménovým řadičem na operačním systému Windows Server 2003 a emailovým serverem Microsoft Exchange Server 2003.

2.1.3 Moodle

Moodle je open-source e-learningový nástroj (systém pro podporu výuky) nabízející jednotný, robustní, bezpečný a integrovaný systém k vytváření personalizovaných výukových prostředí, kterých na světě vznikly desítky tisíc. Těší se důvěry malých i velkých institucí a organizací, mezi něž se řadí Shell, London School of Economics, State University of New York, Microsoft a Open University. Moodle používá přes 65 miliónů uživatelů na akademické i firemní úrovni, což z něj dělá světově nejpoužívanější výukovou platformu.[7]

Moodle nebyl původně na škole v Třebíči využíván (ani nainstalován), jeho implementace probíhala souběžně s vývojem této integrační aplikace, byla nasazena verze `2.5.1+ (Build: 20130815)`. Moodle pro svou činnost potřebuje uživatele (studenty a učitele), učitelé vytváří obsah e-learningových kurzů a studenti ho pak využívají ke studiu. To vyžaduje vložení informací jako jména, přihlašovací údaje a hlavně e-mailové adresy, přes ně probíhají notifikace o událostech v systému (například nový obsah kurzu apod.). Dále je nutné rozdělit uživatele, především studenty, do globálních skupin, pro něž se tvoří kurzy - nazývají se kohorty (anglicky cohorts). V prostředí školy skupiny přesně odpovídají třídám (a jejím skupinám) tak, jak jsem popsal výše v sekci o softwaru Bakaláři, tato rozdělení je tedy žádoucí přenést do Moodle.

Moodle bude v první fázi implementace pouze konzumentem dat, nebude produkovat nová data, která by bylo potřeba synchronizovat do jiných aplikací. Do budoucna by se dalo uvažovat například o importu známek z online testů do aplikace Bakaláři.

2.1.4 YSoft SafeQ

YSoft SafeQ je tiskové řešení, které přináší kontrolu nad tiskem, kopírováním a skenováním v podniku. V prostředí zákazníka běží verze systému `3.6`. Tisk je primárně určen jen pro zaměstnance školy (hlavně učitele). Do SafeQ se uživatel identifikuje pomocí čipové karty, obsahující číslo čipu, a to na terminále umístěném na tiskovým/kopírovacím zařízení. Software zkusí ve své databázi vyhledat poža-

dované číslo čipu a spojit ho se odpovídajícím kontem, které obsahuje kompletní historii tisku, stav konta, a také jméno, příjmení a ostatní údaje o osobě. Všechna tato data byla zadávána ručně přes webové rozhraní, které Ysoft SafeQ poskytuje. Nejdůležitější částí integrace bude přenést čísla čipů ze softwaru Bakaláři do interní databáze SafeQ. SafeQ nepřináší nové informace, které by bylo potřeba předávat do ostatních integrovaných aplikací, splňuje tak definici konzumenta dat.

2.1.5 Další aplikace

Původně měl být integrován i software Jídelna, ale vzhledem k tomu, že se jedná jen o lokální instalaci (tj. aplikace není na serveru, ale pouze na jednom počítači v jídelně, který není přístupný pořád) a navíc jde o velmi zastaralý software bez možnosti importu dat (jinak než přímým nebezpečným zápisem do DBF souborů aplikace), byl z této fáze projektu vyjmut. To způsobilo problém, protože právě tento software je nejlepším zdrojem čísel čipů na škole, a to z toho důvodu, že prakticky každý držitel čipu zároveň navštěvuje jídelnu. Dalšími systémy na škole jsou software Karas (správa studentů ubytovaných na internátu školy) a software Docházka (evidence příchodů a odchodů zaměstnanců do práce), z kapacitních i časových důvodů ale nakonec součástí integrace nejsou. V současné době se však chystá druhá fáze nasazení integrační aplikace, která už postihne úplně všechny systémy zákazníka.

2.2 Stávající řešení

Původní stav „synchronizace“ dat na škole byl závislý na ručním zadávání (a tedy neustálém opisování) informací do všech aplikací. Proces by se dal popsat takto: škola přijala nového žáka nebo učitele, ten byl zaveden do softwaru Bakaláři (klasické údaje jako iniciály, místo narození, třída, rodné číslo a přibližně 50-100 dalších položek). V případě, že chtěl navštěvovat jídelnu, sekretářka mu vydala čip s unikátním číslem (když se jednalo o učitele, dostal čip vždy, protože se používá pro tisk i docházkový systém). Osoba pak musela s čipem dojít do jídelny, kde příslušná obsluha zadala znovu údaje (včetně čísla čipu) do softwaru Jídelna. Učitel byl zaveden také v docházkovém systému (zadávala znovu sekretářka). Poté nastávala část práce pro administrátora sítě, ten zavedl uživatele do souboru ve formátu Excel (jméno,

příjmení, login, heslo, platnost), na který měl navázaný skript v jazyku Visual Basic for Applications (VBA), jež importoval osobu do Active Directory, čímž osoba získala přístup do školní sítě (přihlašování se na počítače v učebnách apod.). Zároveň administrátor sítě zavedl uživatele i do softwaru pro tisk SafeQ (opět jméno, příjmení, číslo čipu, stav konta). Pokud osoba měla být ubytována na internátu školy, administrátor sítě opakoval zadávání i do tohoto systému (software Karas). Mohla nastat i situace, kdy na internátu bydlí osoby, které nejsou součástí ani jednoho z předchozích systémů, protože internát slouží i jako ubytovna. V těchto případech většinou osoby žádají administrátora sítě o přístup do Active Directory, aby se mohly přihlašovat do školní sítě.

Při odchodu osoby ze školy musel proběhnout proces opačný, tj. zneplatnění uživatele. Velmi často nastávaly situace, kdy se na jeho účty zapomnělo a ty pak zůstávaly nečinné v aplikacích (často právě v Active Directory). To s sebou nese i bezpečností rizika, kdyby si osoba byla vědoma, že její účty jsou stále aktivní, mohla by například přistupovat do školní sítě, i přesto že už takovým právem disponovat nemá.

3 | Návrh řešení

Navrhované řešení musí pokrýt přenos dat mezi aplikacemi, které jsou uvedeny v tabulce 7.1.

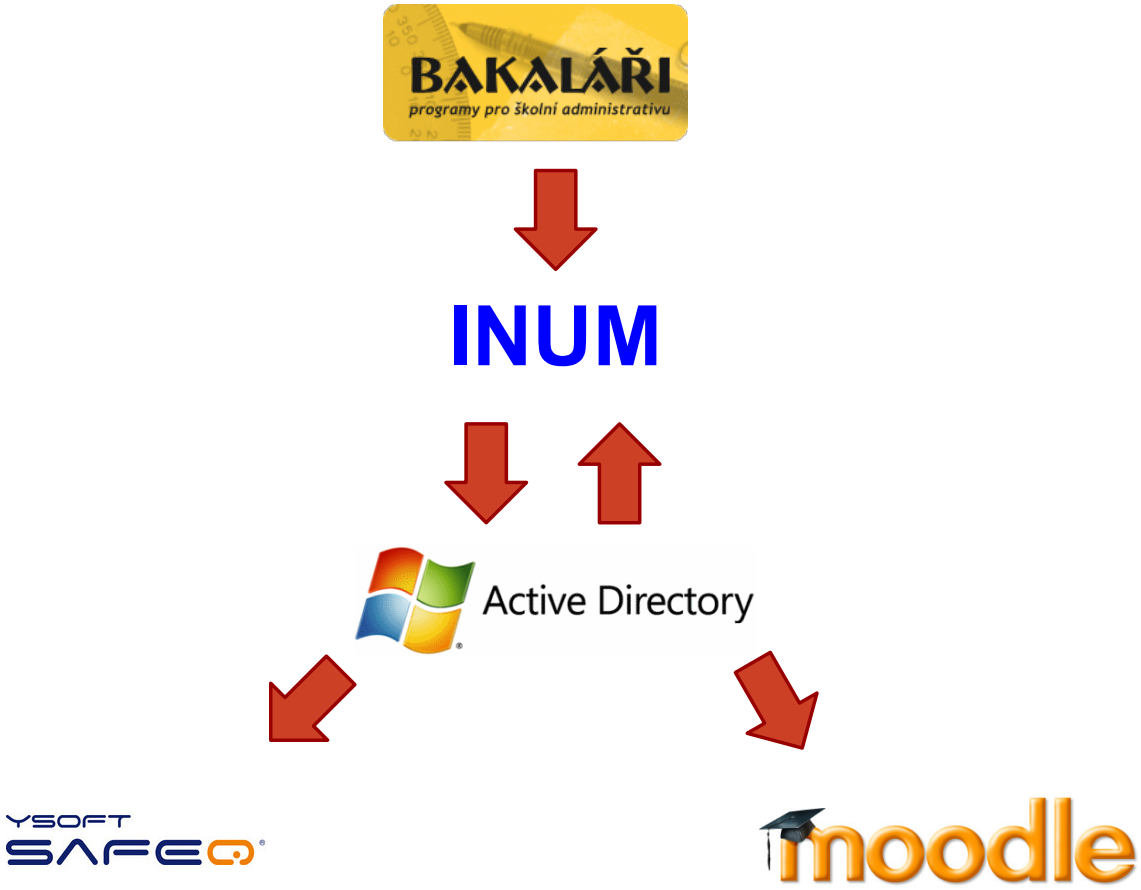
Tabulka 3.1: Přehled integrovaných aplikací

aplikace	**verze**	**vazba přes**
Bakaláři	13/14	INTERN_KOD
Active Directory	AD pro Windows Server 2003	sAMAccountName
Moodle	2.5.1+	sAMAccountName
SafeQ	3.6	cn

Pro každou integrovanou aplikaci je nutné určit vazbu mezi daty, která obsahuje, a daty, která jsou obsažena v aplikaci, na kterou se napojuje. Tato vazba musí být unikátní, nezaměnitelná s žádnou jinou vazbou. Po analýze jednotlivých aplikací byl navržen princip toku dat, s ohledem na to, jaká data aplikace obsahují a hlavně jaké jsou možnosti napojení aplikací na jiné programy. Jeho vizualizace je ilustrována na obrázku 3.1[4].

Program Bakaláři neposkytuje napojení na externí systém, proto bylo zvoleno napojení na vyvíjenou integrační aplikaci - INUM (Integrated User Manager - Nástroj pro integraci uživatelských účtů). Při analýze jeho databázových souborů bylo zjištěno, že tabulky `ZACI.dbf` (data o žácích a studentech) a `UCITELE.dbf` (data o učitelích) obsahují pětimístný primární klíč v atributu `INTERN_KOD` (řetězec znaků) unikátní v rámci kontextu databáze, je totiž používán pro vazbu na jiné tabulky (například učitel na třídu jako třídní učitel, žák na skupinu atd.). Tabulky `TRIDY.dbf` a `SKUPINY.dbf` obsahují primární klíč `KOD_TRID` respektive `KOD_SKUP`. Těchto unikátních identifikátorů bude využito při načítání dat do databáze integrační aplikace,

[4]obrázek vytvořen pomocí aplikace Google Drawing, loga převzata z:
`http://www.gjp-me.cz/uploads/images/aktuality/101217-u-dp-uprav/340.jpg`
`http://enowsoftware.com/assets/images/active-directory-logo.png`
`http://www.sharp-cee.com/cps/rde/xbcr/scee/productimages/260_img-P-document-solution-yssq-960.jpg`
`http://files.gybnp.webnode.cz/200001682-0682b077cd/moodle-icon.jpg`



Obrázek 3.1: Návrh datového toku

kde bude pro každou entitu odpovídající sloupec `bakalari_id`. Pokud bude v databázi již řádek s konkrétním unikátním klíčem, bude se záznam upravovat, pokud ne, bude vytvořen nový záznam. Tabulky `ZACI.dbf` a `UCITELE.dbf` obsahují navíc ještě sloupec `DELETED_RC`, který značí, zda je záznam považován za smazaný. Po určité době smazané záznamy z tabulek mizí a archivují se do jiných tabulek (v případě skupin a tříd mizí rovnou) - když se pak nenajde záznam při synchronizaci dat v tabulkách, bude považován za smazaný a nastaví se tak příslušný atribut v integrační databázi. Z tabulky integrační aplikace záznamy mizet nebudou, pouze budou mít nastaven příznak smazán.

Active Directory bude napojeno také na integrační aplikaci, aby se mohla načtená data z Bakalářů přenášet do něj. V AD je unikátních atributů více, byl zvolen atribut `sAMAccountName` (neboli přihlašovací jméno - login; řetězec znaků). Ten sice z definice unikátní být nemusí, administrátor má ale na škole nastaven proces tak, že `sAMAccountName` je stejný jako atribut `cn` (common name), a ten už unikátní je. V databázi INUM bude veden atribut `ldap_id`, který bude obsahovat právě tuto hodnotu. Login se generuje podle logiky nastavené administrátorem, bude se tedy muset tento algoritmus zapracovat do integrační aplikace a současně najít způsob, jak co nejjednodušeji a efektivně importovat již existující loginy a hesla z Active Directory do INUM, protože synchronizace bude prováděna ze strany aplikace. Na obrázku 3.1 je šipka toku dat i z Active Directory do INUM - `sAMAccountName` totiž není nejlepší atribut pro vazbu dat, ten v AD představuje atribut `objectGUID` (unikátní 16B (bajtový) řetězec znaků, měl by být unikátní celosvětově). Tento atribut bude ukládán do databáze INUM a v budoucnu se možná přes něj budou vyhledávat objekty v Active Directory. Pokud bude synchronizovaný objekt v Active Directory existovat, budou se pouze měnit jeho atributy podle dat v databázi INUM, pokud nebude existovat, vytvoří se nový. V případě objektů s příznakem smazaný - pokud objekt neexistuje v AD, nebude se přidávát, pokud existuje, zneplatní se daný objekt (nebude se mazat, ale např. uživatel nebude schopen se přihlásit).

Moodle nabízí modul pro integraci s Active Directory, vazba bude probíhat také přes atribut `sAMAccountName`. Logiku vytváření, aktualizace a zneplatnění objektů bude řídit sám modul. To samé platí pro Ysoft SafeQ s tím rozdílem, že vazba bude konfigurována přes atribut `cn`. Moodle i SafeQ budou iniciovat synchronizaci dat vlastním zabudovaným plánovačem.

4 | Nástroje pro implementaci

V této kapitole je stručně popsáno, které nástroje byly použity pro vývoj (implementaci) integrační aplikace. Nejdříve se popis věnuje v sekci Aplikační vrstva nástrojům od projektu Spring[5], které tvoří jádro aplikace, následně programům starajícím se o uchovávání a bezpečnost dat v sekci Datová vrstva, poté prvkům jež napomáhají k zobrazovaní dat v sekci Webová vrstva a serverům zajišťujícím běh celé aplikace v sekci Serverová vrstva. Byly použity i některé jiné nástroje, které se do výše uvedeného rozdělení nehodily, ty jsou popsány ve vlastních sekcích. Jsou uvedeny jen nejdůležitější programy, integrační nástroj obsahuje velkou spoustu dalších projektů, jako například Google Guava[6]. Většina nástrojů je open-source a bezplatných.

4.1 Aplikační vrstva

Aplikační vrstva ovlivňuje celý chod integračního nástroje, řídí logiku aplikace, stará se o tok dat a možnosti, jak k nim přistupovat. Dalo by se říci, že se jedná o základní kámen celé implementace.

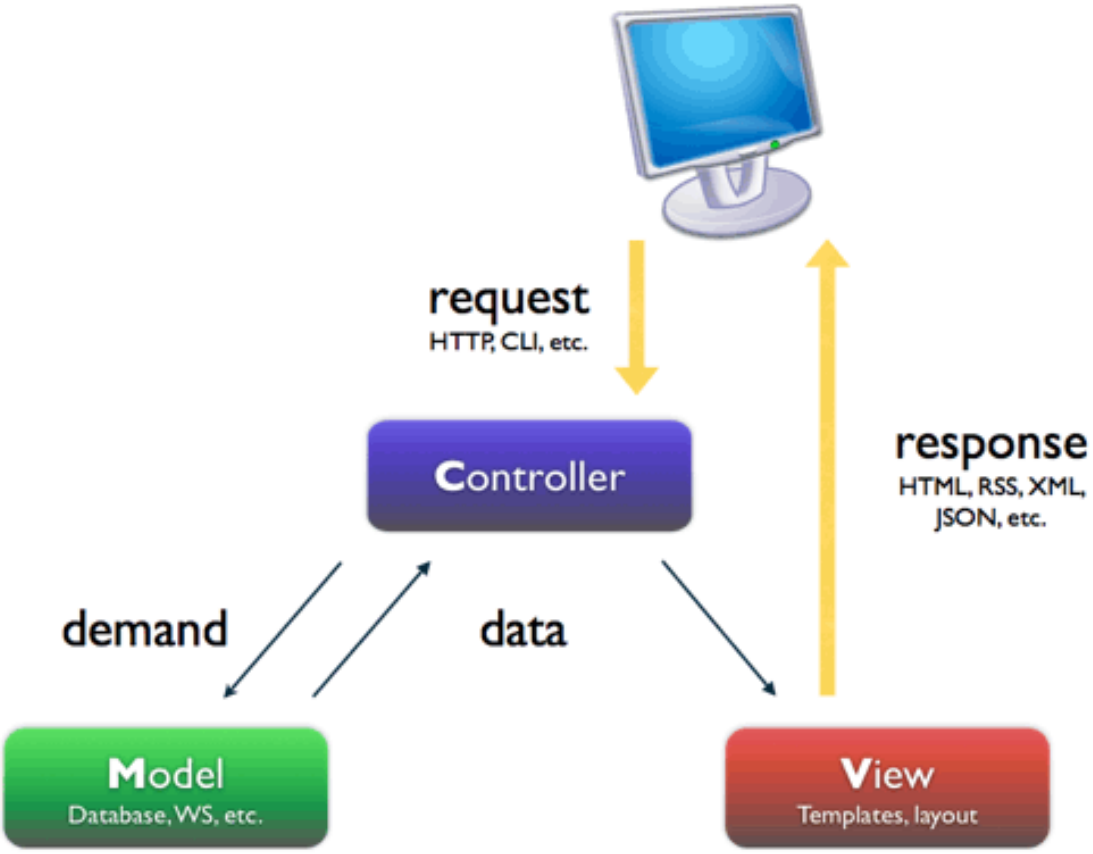
4.1.1 Spring Framework

Spring Framework je rámec pro tvorbu aplikací v programovacím jazyku Java od firmy Pivotal. První jeho verze byla publikována již v roce 2003 programátorem Rodem Johnsonem s cílem ulehčit vývojovým týmům vytváření jednoduchých (ale i složitých), přenositelných, rychlých a flexibilních aplikací podle standardu J2EE. Obsahuje velké množství nejrůznějších částí a vývojových větví, na druhou stranu je modulární, to znamená, že vývojář může jednoduše používat jen ty části, které

[5]URL `http://spring.io/`
[6]URL: `https://code.google.com/p/guava-libraries/`

jsou k dosažení jeho cílům potřebné.[5]

Z množství modulů nabízených Spring Framework je používán Spring Core (funkce jádra), Spring AOP (možnost aspektově orientovaného programování), Spring JDBC a Spring ORM (práce s databázemi) a hlavně Spring Web MVC, tato část poskytuje podporu pro MVC architekturu webové aplikace. MVC (Model-View-Controller) je známý návrhový vzor, princip jeho fungování lze vidět na obrázku 4.1. Všechny moduly jsou v integrační aplikaci použity ve verzi Spring `3.2.4.RELEASE`.



Obrázek 4.1: MVC architektura

4.1.2 Spring Security

Spring Security je aplikace sloužící k ověřování a autorizaci přístupu a je považována za standard zabezpečování aplikací vyvíjených ve Spring Framework. Je

velmi dobře integrována se Spring Web MVC.[1] V synchronizační aplikaci kontroluje přístup (přihlášení) do webové aplikace, aby nedošlo k neoprávněnému čtení nebo změně dat. V databázi jsou totiž velmi citlivé údaje (rodná čísla apod.), je tedy žádoucí, aby byl přístup poskytnut pouze administrátorům, v tomto projektu se jedná o vývojáře aplikace a IT pracovníka školy. Kdyby nebyla aplikace zabezpečena, mohl by se na její webové rozhraní připojit prakticky kdokoliv znalý její URL. Použitá verze Spring Security je `3.1.4.RELEASE`.

4.1.3 Spring LDAP

Spring LDAP je důležitý modul pro správu a komunikaci s prostředím Active Directory. Jedná se Java knihovnu ke zjednodušení operací s LDAP (Lightweight Directory Access Protocol - protokol, na němž je postaveno mimo jiné i Active Directory) ve stylu návrhového vzoru JdbcTemplate. Rámec poskytuje pro uživatele sadu obvyklých úkonů, jako je vyhledávání v kontextu adresáře, procházení přes vrácené výsledky, šifrování/dešifrování dat, filtrovaní apod.[2] V integrační aplikaci je použita verze `1.3.2.RELEASE`.

4.2 Datová vrstva

Následující nástroje se starají o uchovávání, získávání a bezpečnost dat v integrační aplikaci.

4.2.1 Hibernate ORM

Hibernate ORM (Object/Relational Mapping) představuje framework pro tvorbu aplikací, jejichž data musejí být uchována permanentně, tzn. nejedná se jen o online data s krátkou dobou životnosti. Hibernate značně ulehčuje ukládání těchto dat do příslušné databáze, poskytuje nástroje pro získávání dat zpět a hlavně zajišťuje jejich mapování na POJO (Plain Old Java Object) objekty, to znamená, že data uložená obvykle v relačních databázích (tabulkové) převede na data držená v Java objektech. Hibernate ORM implementuje standard JPA (Java Persistence

API) a dovede velmi úzce spolupracovat s aplikačním rámcem Spring Framework.[4] V integrační aplikaci je využívána verze Hibernate `4.2.4.Final`.

4.2.2 MySQL

Pro uchovávání dat je nutné použít databázi, v tomto projektu se jedná o relační databázi Mysql ve verzi `5.5`.

4.2.3 Jasypt

Jasypt[7] je knihovna v jazyce Java umožňující šifrování dat, je schopna ovládat jak symetrické, tak asymetrické šifrování a hashování dat. Její snadné napojení na Hibernate umožní například ukládat hesla uživatelů do databáze v zašifrované podobě, takže v případě nepředpokládaného úniku (to ale neznamená, že se nemůže stát) dat z databáze se nedostane útočník k citlivým informacím.

4.3 Webová vrstva

Webová vrstva se skládá převážně z nástrojů zobrazujících potřebné informace pro administrátory integrační aplikace na rozhraní internetového prohlížeče, takže umožňuje aplikace ovládat a konfigurovat přes webové rozhraní stejně jako zobrazovat data uložená v databázi.

4.3.1 Twitter Bootstrap

Twitter Bootstrap[8] je v poslední době trendem různorodých webových aplikací, což se stává čím dál více terčem kritiků tohoto projektu, argumentují totiž tím, že všechny portály mají stejný vzhled a navíc nenutí uživatele, kteří ho použijí, příliš rozumět kaskádovým stylům a jiným elementárním dovednostem programátora

[7] URL: `http://www.jasypt.org/`
[8] URL: `http://getbootstrap.com/`

webového rozhraní. V této problematice zastávám také spíše kritický názor, na druhou stranu přínos této integrační aplikace nemá být ve spektakulárním a efektním uživatelském designu, ale v její funkcionalitě a efektivitě, takže pro potřeby konfigurace přes web je Bootstrap velmi vyhovující a usnadňující další rozvoj aplikace.

Twitter Bootstrap přináší responzivní (měnící se dle šířky zobrazujícího zařízení) design orientovaný hlavně na mobilní zařízení, aby se aplikace vytvořené pro klasické rozlišení monitorů daly ovládat i přes tato zařízení s menšími zobrazovacími schopnostmi. Poskytuje vlastní CSS (kaskádové) styly a využívá rozšířeného javascriptového frameworku jQuery[9]. Používána je verze Twitter Bootstrap `3.0.3`, jQuery `v1.11.0` a pluginy pro Bootstrap jako např. Bootbox nebo bootstrap-datetimepicker.

4.3.2 Freemarker

Freemarker[10] je šablonovací jazyk podobný původním JSP (Java Server Pages) či Velocity. Slouží pro generování převážně HTML stránek, v třívrstvé architektuře MVC se jedná o písmeno V (view). Je schopen zobrazovat Java objekty předané v modelu, a naplňovat tak obsah generovaného výstupu, lehce se pomocí něho oddělí aplikační logika od zobrazování dat. Integrační aplikace obsahuje Freemarker verze `2.3.20`.

4.3.3 Sitemesh

Sitemesh[11] představuje program implementující *Decorator pattern* (návrhový vzor dekorátor), který v tomto konkrétním případě říká, že lze jednoduše rozdělit HTML kód stránky na části, které se opakují (například hlavička, patička), a části, které jsou unikátní pro každou stránku (její obsah). Pomocí Sitemesh se vytvoří tzv. dekorátor, jímž se obalí každá určená stránka po jejím generování, takže nemusíme opakující se části kopírovat na více míst v kódu, ale máme je pouze na jednom místě. Případná změna dekorátoru je pak podstatně jednodušší a rychlejší na provedení.

[9] URL: `http://jquery.com/`
[10] URL: `http://freemarker.org/`
[11] URL: `http://wiki.sitemesh.org/wiki/display/sitemesh/Home`

4.4 Serverová vrstva

Aplikace napsané ve Spring Framework potřebují pro svoji činnost webový server, který se bude starat o předávání požadavků ze strany klienta, například zadání příslušné URL aplikace do webového prohlížeče. Použity jsou dva takové servery, jeden při vývoji a druhý při implementaci v produkčním prostředí zákazníka.

4.4.1 Jetty

Jetty[12] server je použit pro vývoj integrační aplikace, a to z důvodu jeho jednoduchosti a nenáročnosti na systémové prostředky. Jedná se o HTTP (Hypertext Transfer Protocol) server a Java Servlet kontejner, operováno je s verzí `7.6.4`.

4.4.2 Apache Tomcat

Apache Tomcat[13] je použit v produkčním prostředí třebíčské školy. Nevyužívá se Jetty server, protože Apache Tomcat velmi dobře a úzce spolupracuje s Apache[14], což je open-source HTTP server. Na Apache serveru běží v produkčním prostředí například Moodle (projekt napsaný v programovacím jazyku PHP), a tak by bylo zbytečné nasazovat další HTTP server, navíc by nastal problém s přesměrováním portů atp. Apache Tomcat je také jako Jetty Java Servlet kontejner, u zákazníka je nasazen Apache verze `2.4` a Apache Tomcat verze `7.0.50`.

4.5 Maven

Apache Maven[15] je software pro správu vyvíjeného projektu, řídí proces kompilace zdrojového kódu, lze definovat závislosti na ostatních Java knihovnách, generování dokumentace a jiné. Konfigurace projektu probíhá pomocí souboru `pom.xml`

[12]URL: `http://www.eclipse.org/jetty/`
[13]URL: `http://tomcat.apache.org/`
[14]URL: `http://httpd.apache.org/`
[15]URL: `http://maven.apache.org/`

v kořenovém adresáři projektu. Pokud v souboru uvedeme závislost na cizí knihovně, Maven se postará o proces stažení příslušné knihovny z tzv. repository (úložiště) a přidá ji do kontextu projektu, poté lze knihovnu využívat kdekoliv v aplikaci. Umožňuje také lepší správu všech použitých závislostí a snadnou výměnu nebo jejich aktualizaci.

4.6 Powershell

"Powershell se pomalu stal neodmyslitelnou součástí operačního systému Windows a dalšího aplikačního softwaru společnosti Microsoft. Nebylo by na tom nic divného, kdyby se nejednalo o technologii, která byla ještě před pár lety pro Windows poměrně netypická: textové příkazové rozhraní, správcovská textová konzola, zkrátka shell."[6, str. 14] Důvody, proč bylo rozhodnuto použít v rámci implementace integrační aplikace právě Powershell, jsou v některých ohledech nedostatečné možnosti programovací jazyku Java a příslušných Java knihoven v operacích s prostředím Windows, Active Directory a v operacích se souborovým systémem.[8] Skripty napsáné pro Powershell lze pouštět a ovládát z prostředí aplikace a plně nahradí nedokonalosti Javy a pomohou vyřešit problémy, například práva uživatelů Active Directory k přístupu ke složkám.

Powershell pro Windows Server je běžně dostupný až od verze Windows Server 2008, takže v produkčním prostředí se musel nainstalovat balíček od Microsoft s názvem Windows Management Framework Core[16], který obsahuje verzi Windows Powershell 2.0 a ještě musel být přidán balíček příkazů pro ovládání Active Directory (není ve verzi Windows Server 2003) od firmy Quest[17].

[16]dostupný z: `http://support.microsoft.com/kb/968930`
[17]dostupný z: `http://www.quest.com/powershell/activeroles-server.aspx`

5 | Implementace nástroje

Nástroj pro integraci uživatelských účtů je nazván anglicky *Integrated User Manager*, odtud zkratka INUM.

5.1 Vývoj aplikace

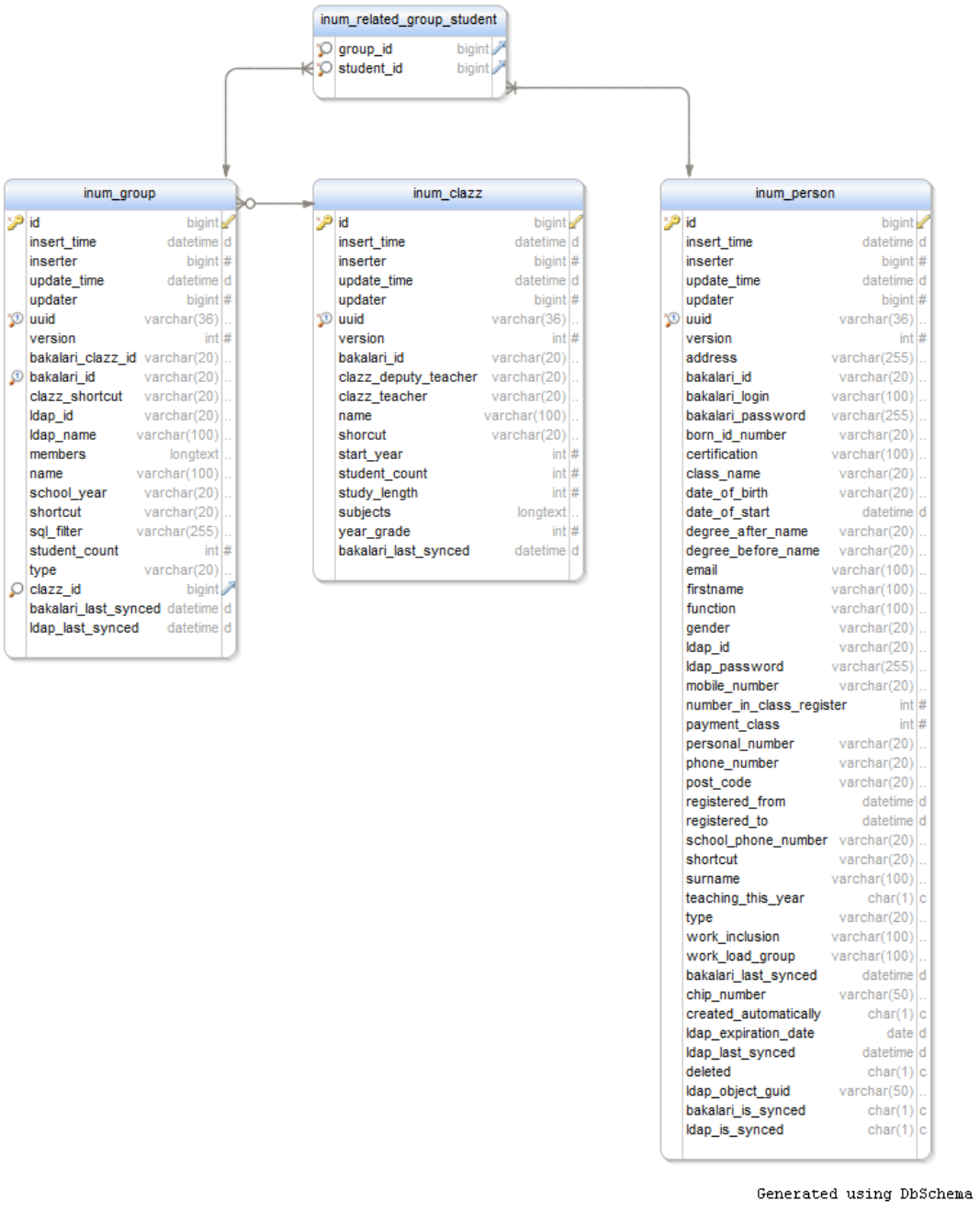
Vývoj pilotní verze aplikace probíhal v období od konce srpna 2013 do října 2013, pracovním prostředím byl osobní počítač. Na obrázku 5.1 je vidět historie vkládání změn do verzovacího systému GIT.

Obrázek 5.1: Graf historie počtu změn ve verzovacím systému (z aplikace Gitblit)

Začalo se vytvořením nového projektu v Java frameworku Spring, založením MySQL databáze pro účely vývoje a napojení databáze přes projekt Hibernate ORM. Následoval návrh databázového modelu popsaný v následující sekci.

5.1.1 Datový model

Datový model z velké části kopíruje strukturu databáze programu Bakaláři. Na obrázku 5.2 je vidět relační model pro databázové entity osoba (*Person*), třída (*Clazz*) a skupina (*Group*) a vazby mezi nimi. V příloze A je uveden kompletní databázový model.



Obrázek 5.2: Datový model

U každého atributu bylo potřeba pečlivě zvážit jeho datový formát, aby při přenosu dat do integrovaných aplikací nemuselo docházet ke zbytečné konverzi mezi formáty. Jednotlivé atributy uvedených tabulek byly jeden po druhém hodnoceny,

zda má, nebo by mohl v budoucnu mít, využití při integraci s ostatními systémy. Tak vzniklo v aplikaci INUM momentálně mnoho nadbytečných informací (konkrétně třeba úvazky učitelů), ale na druhou stranu se například ukázalo, že původně zdánlivě zbytečný atribut `pracovní zařazení`, který představuje číselnou hodnotu, je velmi důležitou součástí algoritmu rozlišování interních a externích učitelů. Dalo by se tvrdit, že aplikace INUM slouží jako přechodné úložiště dat z Bakalářů, to je ale jenom jeden úhel pohledu. Údaje totiž při synchronizaci jsou zpracovávány a ukládány v aplikaci pod vhodnou strukturou (příkladem může být zařazení žáka do skupiny – v Bakalářích může být uložen jako výčet unikátních klíčů osob v textovém poli, v aplikaci INUM je výčet přetvořen na vazbu 1:N mezi osobou a skupinou). Navíc databáze INUM obsahuje důležité vazební atributy jak na program Bakaláři, tak na Active Directory, což Bakaláři nemají.

5.1.2 Získání dat

Po návrhu a implementaci datového modelu se začalo pracovat na prvotní integraci s primárním producentem dat, programem Bakaláři. Způsob napojení je detailně popsán v kapitole 6 Integrace aplikací. Pro vývojové účely se musely zkopírovat databázové soubory z produkčního prostředí na místní disk vývojového počítače, aby v případě, že by se provedl neplatný zápis do souborů nebo se jakkoliv jinak poškodily, nebyla ovlivněna produkční data. Poté došlo k načtení údajů do databáze INUM.

Následovala příprava dat pro napojení Active Directory. Administrátor měl k dispozici soubor ve formátu Microsoft Excel, obsahující sloupce jméno, příjmení, login, heslo a platnost, jehož pomocí v kombinaci s VBA skriptem zaváděl uživatele do Active Directory. Neexistoval však unikátní klíč, podle kterého by šlo přiřadit uživatelům v databázi odpovídající login a heslo. Byl tak vytvořen SQL skript, který přidával položkám v databázi nejdříve přes klíč „příjmení, jméno“ a pak přes klíč „příjmení“ login a heslo pro Active Directory. Tímto krokem se vyřešila většina uživatelů (celkem se z Bakalářů načetlo přibližně 900 osob, z toho okolo 500 aktivních – ty byly brány v potaz). Zbývalo vyřešit duplicity (přibližně 20 uživatelů, řešeno ručně) a uživatele, kteří se nespárovali (zbytek, poměrně velká část, opět ručně). Zde bylo vidět, že při pouhém ručním zakládání uživatelů do jednoho excelovského

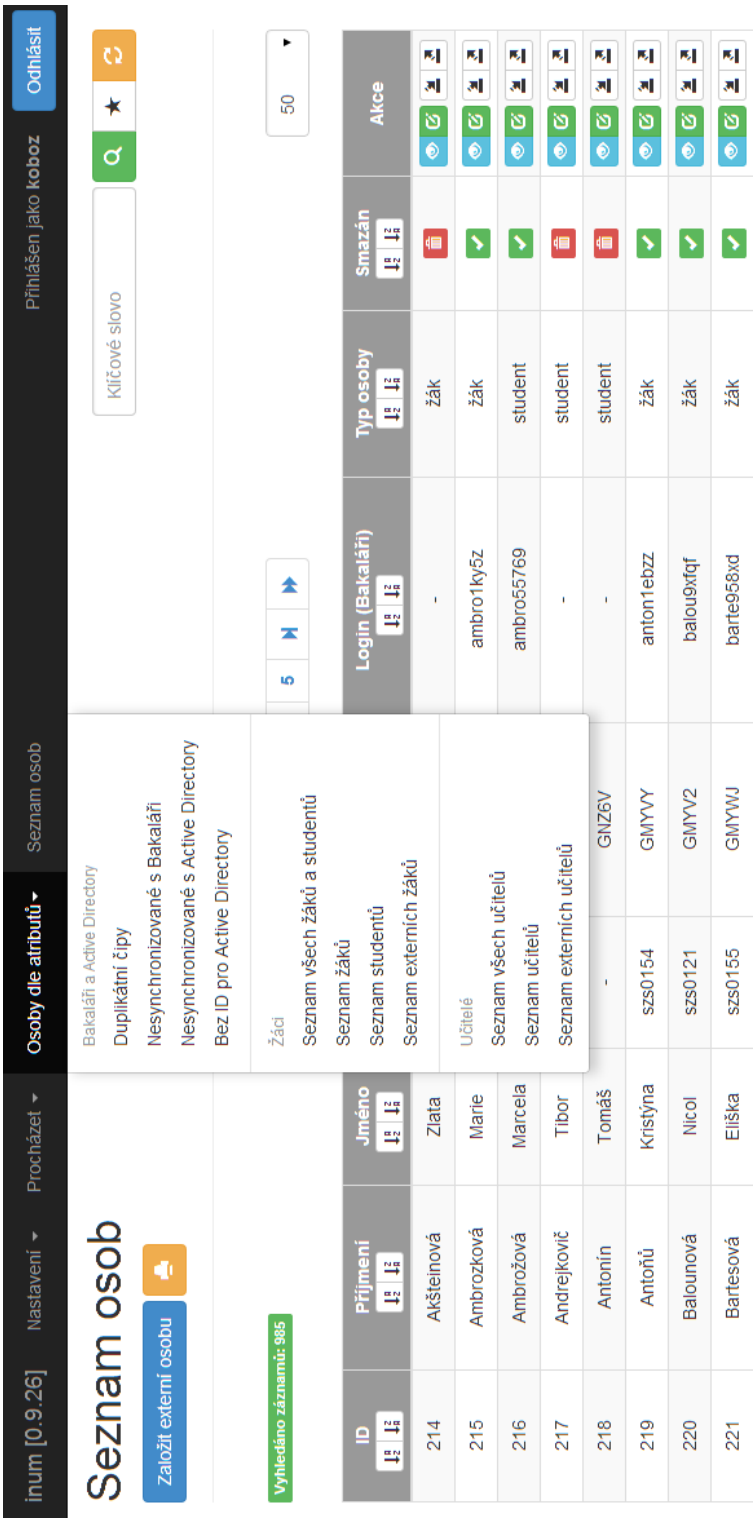
souboru docházelo, i přes pečlivost administrátora, k chybám a překlepům stejně jako k duplicitám (dvojí až trojí loginy). Nakonec tedy byly k dispozici údaje pro synchronizaci do Active Directory, pokračovalo se napojením samotného Active Directory popsaném kapitole 6 Integrace aplikací.

5.1.3 Další vývoj

Pro vývojové prostředí bylo nutné co nejvíce nasimulovat produkční prostředí, potřebný tak byl Microsoft Windows Server povýšený na doménový řadič. Byla zvolena forma virtualizace, kde se přes program VirtualBox server nainstaloval, nastavil a rozběhl. Pak už nic nebránilo přenosu dat z databáze INUM do Active Directory. Všem objektům v Active Directory bylo přidáno do názvu `_dev` na znamení, že se jedná o vývojovou verzi aplikace.

Zbývalo naprogramovat webové rozhraní. Pro tyto účely byl použit projekt Twitter Bootstrap v kombinaci se šablonovacím jazykem Freemarker, který velmi napomohl ke snadné implementaci vizualizace databázových dat. Dalším nutným krokem bylo také zabezpečit přístup k aplikaci přihlášením pomocí Spring Security. K aplikaci INUM mají přístup pouze administrátoři. Ti můžou osobám, které se synchronizují z Bakalářů do aplikace, na webovém rozhraní upravovat pouze ty údaje, které nejsou v Bakalářích (jako například platnost v doméně, login a heslo do domény – pozor, ze specifikace odlišný login do Bakalářů a do Active Directory), z toho důvodu, že by se data pravidelně přepisovala daty z Bakalářů; ručně přidaným externím osobám lze upravovat vše.

Na požádání zákazníka byly přidělány funkcionality webového rozhraní, jako například možnost tisknout sestavu login a heslo pro konkrétní osobu nebo seznam osob (třída, skupina) nebo různé seznamy, např. osob s duplikátními čísly čipů, případně vyhledávání v objektech. Na webovém rozhraní je možnost spustit tlačítkem jak hromadnou synchronizaci všech osob, tak jednotlivé osoby, a to v obou směrech Bakaláři-INUM a INUM-Active Directory. Na obrázku 5.3 je ukázka jedné z obrazovek webového rozhraní aplikace INUM.



Obrázek 5.3: Screenshot webového rozhraní aplikace

5.2 Testování

Testování probíhalo již v prostředí zákazníka, přibližně dva měsíce v intervalu listopad-prosinec 2013. Aplikace INUM už byla napojena přímo na produkční databázi programu Bakaláři, aby se ale neovlivnil chod IT procesů školy, v Active Directory byla vytvořena organizační jednotka (OU) `Test` a všechny objekty byly synchronizovány do ní s příponou názvu `_test` - nedocházelo tak ke konfliktům s již existujícími objekty v AD. Na testovací OU bylo nastaveno napojení aplikace Moodle, která též běžela v testovacím režimu. SafeQ zatím kvůli nejistým výsledkům integrace napojováno nebylo. V součinnosti s administrátorem zákazníka se ladilo chování programu, analyzovala se přenášená data a opravovaly se případné chyby.

5.3 Produkční verze

Při nasazování produkční verze aplikace stačilo překonfigurovat informaci v databázi na prostředí z `TEST` na `PRODUCTION` a změnit opět v konfiguračních položkách umisťování objektů v rámci adresáře Active Directory. Objekty se pak začaly synchronizovat s produkčními daty. Po úspěšném přechodu na produkční verzi integrační aplikace se přenastavil Moodle také na produkční data a SafeQ bylo nakonfigurováno pro synchronizaci s Active Directory, jak je popsáno v kapitole 6 Integrace aplikací.

Jako aplikační server pro produkční prostředí slouží kontejner Apache Tomcat. Celá aplikace běží na školním serveru v Třebíči, kde je nainstalován operační systém Microsoft Windows Server 2012 R2. Jiný server, který slouží jako doménový kontroler, je vybaven operačním systémem Microsoft Windows Server 2003 a správcem emailových schránek Microsoft Exchange Server 2003. Nasazování aplikace do provozu muselo být provedeno dvakrát, protože v původním stavu nám byl poskytnut server se starými Windows 2003, v půlce ledna 2014 se potom z technických důvodů nasazoval nový server s operačním systémem Windows 2012 R2.

6 | Integrace aplikací

V následujících sekcích je detailně popsána integrace jednotlivých aplikací, konkrétně způsob jak probíhala, které problémy se při ní vyskytly, jak musely být aplikace pro komunikaci nastaveny a další převážně technické údaje.

6.1 Integrace - Bakaláři

Aplikace Bakaláři byla v kapitole Analýza prostředí definována jako primární producent dat, proto bylo napojení datového zdroje na integrační aplikaci prioritou. Bakaláři na škole v Třebíči mají svá data uchována v DBF souborech (ještě jsou verze programu, které používají MSSQL databáze - Microsoft SQL Server), bylo tedy nutné najít vhodný nástroj, který dokáže číst DBF soubory a operovat s nimi v prostředí jazyku Java.

Existuje velké množství open-source Java knihoven pro čtení a zápis DBF souborů, například *DANS DBF Library*[18] nebo *jdbf*[19], jejich problém je však individuální přístup k danému souboru, tj. v jeden okamžik lze přistupovat pouze k jednomu souboru, nelze tak provádět výběr přes více souborů, jako jsme zvyklí při dotazování na data přes SQL - Structured Query Language (JOIN, neboli spojení tabulek - výběr z více tabulek). Vzhledem k povaze dat v aplikaci Bakaláři, kde se často vyskytují vazby mezi jednotlivými entitami jako například žák-třída, žák-skupina, skupina-třída a jiné, bylo vhodné najít alternativu umožnující dotazování nejlépe právě přes SQL.

Takového výsledku lze dosáhnout pomocí JDBC-to-ODBC bridge (mostu). ODBC (Open Database Connectivity) je obecné rozhraní definující přistupování k databázovým zdrojům. Vývojáři ODBC mají za cíl vytvořit rozhraní nezávislé na použitém databázovém systému a operačním systému. JDBC (Java Database Connectivity)

[18]URL: `http://dans-dbf-lib.sourceforge.net/`
[19]URL: `https://github.com/iryndin/jdbf`

je rozhraní pro programovací jazyk Java, které poskytuje metody pro dotazování a změnu dat v databázích, hlavně pak relačních. JDBC-to-ODBC bridge přináší přístup v JVM (Java Virtual Machine) prostředí pro jakýkoliv databázový zdroj dosažitelný pomocí ODBC rozhraní.[11] Bylo by samozřejmě výhodnější využít jen JDBC rozhraní bez použití JDBC-to-ODBC bridge, takové ovladače se však nepovedlo najít v bezplatné verzi, existují jen placené verze jako například *StelsDBF*.[20]

Ke čtení (i zápisu) do DBF souborů už pak stačí nainstalovat na server, na kterém běží aplikace, ovladač Microsoft Visual FoxPro ODBC Driver[21]. Ovladač je napsán pro 32b (bitové) operační systémy. Prostředí zákazníka však disponuje 64b operačním systémem, tento problém se dá vyřešit použitím stávající 32b verze ovladače s tím omezením, že celá integrační aplikace bude spuštěna pomocí 32b JVM, podrobněji viz `http://saltydogllc.com/?p=356`.V integrační aplikaci je poté nutné uvést následující konfiguraci JDBC zdroje:

```
"Url"="jdbc:odbc:Driver={Microsoft Visual FoxPro Driver};
       SourceType=DBF;
       SourceDB=d:/dev/inum/bakalari/evid/data;
       Exclusive=No;Collate=Machine;NULL=NO;
       DELETED=NO;BACKGROUNDFETCH=NO;"
"DriverClassName"="sun.jdbc.odbc.JdbcOdbcDriver"
"ConnectionProperties"="charSet=WINDOWS-1250"
```

Obsah `"Url"` je takzvaný **connection string** (řetězec k připojení ke zdroji), určuje který ovladač se má k připojení použít (již výše zmiňovaný Microsoft Visual FoxPro Driver) a hlavně cesta ke složce, ve které jsou uloženy DBF soubory, může se jednat o cestu k místní složce nebo cestu v síti.[22] `"DriverClassName"` je název Java třídy určující používaný JDBC ovladač a `"ConnectionProperties"` definuje specifické parametry připojení, v tomto případě jaké použít kódování pro čtení a zápis do DBF. Před použitím tohoto kódování byl problém s českou diakritikou například ve jménech osob.

[20] URL: `http://www.csv-jdbc.com/stels_dbf_jdbc.htm`

[21] Dostupný z `http://download.microsoft.com/download/vfoxodbcdriver/Install/6.1/W9XNT4/EN-US/VFPODBC.msi`

[22] Více o connection strings na `http://www.connectionstrings.com/visual-foxpro/` nebo `http://www.devlist.com/ConnectionStringsPage.aspx`

Tímto je vytvořeno spojení do databáze aplikace Bakaláři a lze pak jednoduše psát SQL dotazy pro získání potřebných dat, například následující dotaz vrátí seznam všech skupin včetně potřebných parametrů:

```
SELECT
 g.KOD_TRID clazz_id,
 c.ZKRATKA clazz_shortcut,
 g.KOD_SKUP group_id,
 g.CLENOVE members,
 g.NAZEV name,
 g.SKOLNI_ROK school_year,
 g.ZKRATKA shortcut,
 g.FILTR sql_filter,
 g.POCET_ZAKU student_count,
 g.TYP type
FROM
 dbf/SKUPINA.DBF g
LEFT JOIN dbf/TRIDY.DBF c ON g.KOD_TRID = c.KOD_TRID
ORDER BY clazz_shortcut;
```

Klauzule `FROM dbf/SKUPINA.DBF g` a `LEFT JOIN dbf/TRIDY.DBF c` obsahuje přímé odkazy na DBF soubory, ze kterých je čteno, a to relativně vůči cestě zadané v connection stringu. Ze Spring Framework je pak využíván `JDBCTemplate` a jeho metody jako např. `jdbcTemplate.query(sqlGroups, new BakalariGroupMapper())`, která výsledek výše uvedeného SQL příkazu přemapuje na seznam Java objektů definovaných ve třídě `BakalariGroupMapper`.

6.2 Integrace - Active Directory

Integrace Active Directory byla jednodušší než integrace softwaru Bakaláři, protože byl použit projekt Spring LDAP verze `1.3.2.RELEASE`, který značně ulehčuje implementaci, konfiguraci a výměnu dat mezi integrační aplikací INUM a doménovým řadičem. Pro konfiguraci přístupu k serveru s Active Directory je nutné v souboru

`/src/main/filters/production_filter.properties` (respektive při vývoji a testování v souboru `/src/main/filters/development_filter.properties`) uvést následující údaje:

```
cfg.ldap.url=ldaps://192.168.56.102:636
cfg.ldap.base=dc=inum,dc=development
cfg.ldap.readonly=false
cfg.ldap.user.dn={DN administrátora}
cfg.ldap.user.password={heslo administrátora}
```

Všechny parametry jsou povinné, `cfg.ldap.base` udává bázi vůči které budou probíhat následná vyhledávání a změny objektů v AD, `cfg.ldap.readonly` zda bude z AD pouze čteno nebo se bude i zapisovat (*false* = i zápis) a `cfg.ldap.user.dn` s `cfg.ldap.user.password` jsou přístupové údaje administrátorského účtu, jimiž se přihlašuje integrační aplikace. Nejdůležitějším údajem je `cfg.ldap.url`, tedy IP adresa nebo jméno serveru, na kterém je nainstalován doménový řadič. Jeho část *ldaps* (a také číslo portu *636*) mimo jiné definuje, že se bude připojovat přes zabezpečený (LDAPS - anglicky LDAP over SSL) protokol, kvůli přenášení hesel uživatelských účtů do AD se jedná o nezbytnou část, pokud by totiž byl použit nezabezpečený protokol (klasický LDAP), Active Directory nepovolí přenášení hesel. K použití LDAPS je potřeba nainstalovat SSL (Secure Sockets Layer) certifikát doménového kontroleru do úložiště Javy. K tomuto účelu byl použit nástroj InstallCert[23], podrobný popis postupu instalace certifikátu lze nalézt na `http://www.opentox.org/tutorials/q-edit/how-to-install-ssl-certificates`/. Po tomto nastavení pak stačí uvést v XML souboru aplikačního kontextu integrační aplikace následující Java objekty (Beans):

```
<bean id="ldapContextSource"
  class="org.springframework.ldap.core.support.LdapContextSource">
    <property name="url" value="${cfg.ldap.url}"/>
    <property name="base" value="${cfg.ldap.base}"/>
    <property name="userDn" value="${cfg.ldap.user.dn}"/>
    <property name="password" value="${cfg.ldap.user.password}"/>
```

[23]Dostupný z `https://code.google.com/p/java-use-examples/source/browse/trunk/src/com/aw/ad/util/InstallCert.java`

```
    <property name="baseEnvironmentProperties">
     <map>
         <entry key="java.naming.security.authentication"
           value="simple"/>
         <entry key="java.naming.ldap.attributes.binary"
           value="objectGUID"/>
        </map>
    </property>
    <property name="pooled" value="false"/>
</bean>

<bean id="ldapTemplate"
  class="org.springframework.ldap.core.LdapTemplate">
    <property name="contextSource" ref="ldapContextSource"/>
    <property name="ignorePartialResultException" value="true"/>
</bean>
```

Pomocí objektu `ldapTemplate` lze následně snadno přistupovat a dotazovat se do AD, příkladem může být, aby AD vrátilo seznam všech objektů typu uživatel s uvedeným přihlašovacím jménem (`ldapId` je parametr obsahující požadované přihlašovací jméno):

```
ldapTemplate.search(
  "",
  "(&(objectClass=user)(sAMAccountName=" + ldapId + "))",
  new LdapPersonContextMapper()
);
```

Příkaz funguje na podobném principu jako JDBC template uvedený v integraci aplikace Bakaláři, tj. získanou odpověď od serveru AD přetransformuje na seznam objektů definovaných ve třídě `LdapPersonContextMapper`. Tímto způsobem je integrační aplikace schopna číst data z AD, což je využíváno v kódu k tomu, aby se zjistilo, jestli daný synchronizovaný objekt již v AD existuje. Existuje-li, pak je určeno, že se budou jeho atributy jenom aktualizovat následujícím voláním metody:

```
ldapTemplate.modifyAttributes(
  {DN objektu získaného čtení,},
  {mapa klíč-hodnota atributů ke změně}
);
```

a pokud objekt v AD ještě neexistuje, voláním následující metody se vytvoří:

```
ldapTemplate.bind(
  {DN nového objektu},
  null,
  {mapa klíč-hodnota atributů k vytvoření}
);
```

Při vytváření nových objektů je třeba dbát na unikátnost jejich DN, jinak Active Directory nepovolí zapsání. V databázi INUM je pro tento účel vedeno pole `ldap_id` pro entity `Person` (osoba - žák, student, učitel) a `Group` (skupina), které je jedinečné a lze z něj snadno sestavit požadované DN.

Do Active Directory jsou synchronizovány údaje o osobách (jména, adresy, mobilní čísla, emaily, čísla čipů apod.), údaje o zařazení osob (student, žák, učitel, externí učitel, externí žák) a údaje o příslušnosti ke skupinám (třídy, skupiny – podmnožiny tříd). Každá osoba je samostatným objektem v Active Directory (objekt *user* či *person*), je součástí globální skupiny (objekt *group*), která ji identifikuje vzhledem k pracovnímu zařazení (student, žák, učitel, externí učitel, externí žák; patří právě do jedné takové skupiny) a žáci či studenti jsou pak ještě členové jiných skupin (objekt group) podle toho, do které třídy nebo podskupiny třídy patří (mohou patřit do více skupin).

Další důležitou součástí integrace Active Directory je vytváření domovských adresářů pro uživatele domény. K tomu velmi dobře posloužil nástroj Powershell 2.0 pro Windows, jak je již zmíněno v sekci pojednávající o Powershellu. K ovládání skriptů napsaných v tomto skriptovacím jazyce z Javy slouží třída umístěná v cestě `src\main\java\cz\davidak\utils\PowershellScriptHandler.java`, která zvládá pouštět skripty zadané přímo řetězcem znaků nebo je načítat z konfigurační položky aplikace. Zde je uveden příklad, jak se nastavuje právo plné kontroly pro administrátory na konkrétní složce pomocí Powershellu:

```
$acl = Get-Acl {cesta ke složce}
$ruleForAdmins = New-Object
  System.Security.AccessControl.FileSystemAccessRule(
    "\Administrators", "FullControl",
    "ContainerInherit, ObjectInherit", "None", "Allow")
$acl.AddAccessRule($ruleForAdmins)
Set-Acl $homeFolderPath $acl
```

Tento kód je uložen v konfiguraci aplikace, lze ho na webovém rozhraní libovolně měnit. V sekci o Powershellu je také řečeno, že některé operace s Active Directory nelze pohodlně udělat nebo vůbec provádět z Javy, jeden takový příklad je nastavení nemožnosti změny hesla pro uživatele, což je vzhledem k pravidelné synchronizaci hesel z integrační aplikace do Active Directory žádoucí efekt. Pomocí Powershellu a sady příkazů Quest pro Powershell se tohoto nastavení dá dosáhnout následujícím způsobem:

```
Get-QADUser $personCN -Service "{jméno serveru}" |
  Set-QADUser -AccountExpires $accountExpires
Get-QADUser $personCN -Service "{jméno serveru}" |
  Add-QADPermission -Account SELF,Everyone
    -ExtendedRight "User-Change-Password"
    -Deny -ApplyTo ThisObjectOnly
```

Novější verze systému Windows Server (od 2008) mají Powershell nainstalovaný v základním nastavení a poskytují podobnou sadu příkazů jako nástroje Quest s tím rozdílem, že práce s nimi je pohodlnější a tohoto konkrétního efektu nemožnosti změnit heslo lze dosáhnout jednodušší cestou.

Poslední důležitou částí, kterou bylo potřeba při integraci Active Directory vyřešit, je zakládání e-mailových schránek pro učitele v prostředí Microsoft Exchange Server. Tento program ukládá své nastavení právě v doméně Active Directory, tj. i údaje o schránkách lze najít mezi atributy AD. Opět, pokud bychom operovali s novější verzí Microsoft Windows Server (2008 a výš), bylo by vytváření schránek jednoduché pomocí Powershell skriptů, v produkčním prostředí je však nasazen operační systém Windows Server 2003, tak bylo nutné najít alternativní řešení. Java

knihovna s operacemi pro Exchange Server neexistuje, došlo se však k závěru, že k vytvoření schránek stačí do Active Directory pomocí Spring LDAP přenášet tyto 4 údaje - `mail`, `mailNickName`, `displayName` a hlavně `homeMDB`. První určuje celou e-mailovou adresu včetně zavináče, druhý část adresy před zavináčem, třetí jméno osoby vlastnící e-mailovou schránku a čtvrtý, nejdůležitější, je umístění schránek v adresáři Active Directory. Cesta k poslednímu údaji byla získána z analýzy dříve vytvořených Exchange účtů a je jednou z konfiguračním položek integrační aplikace. Tento způsob vytváření vyplynul z testování, inspirací byly tipy ze stránky SelfADSI[24].

6.3 Integrace - Moodle

Integrace e-learningového nástroje Moodle spočívala v nastavení propojení dat s Active Directory. To zahrnuje 2 části - zaprvé ověřování přihlašování uživatelů do Moodle a zadruhé synchronizaci globálních skupin (v Moodle nazývaných kohortů) se skupinami vytvořenými v Active Directory z příslušnosti žáků a studentů do tříd a skupin tříd (z hlediska programu Bakaláři).

První část je zajištěna modulem pro Moodle *LDAP*, screenshot z nastavení tohoto modulu je v příloze B. Princip přihlašování uživatelů do Moodle spočívá v důvěře ze strany Moodle k jiné autoritě, v tomto případě Active Directory. Pokud se uživatel prokáže správným přihlašovacím jménem a heslem, Active Directory signalizuje, že uživatel je ověřený a Moodle ho tedy vpustí do svého systému. Pokud Moodle uživatele ještě neznal, vytvoří pro něj záznam ve vlastní databázi a načte příslušné informace (e-mailová adresa, jméno, příjmení), a když uživatel patří do globální skupiny AD *ucitele*, udělí mu roli tvůrce kurzu. V prostředí zákazníka je nastaveno, že si Moodle neukládá hesla uživatelů, a tak ověřování vůči Active Directory probíhá při každém přihlašování do Moodle. Výhoda tohoto nastavení je, že pokud se změní heslo uživatele v Active Directory, nemusí se měnit i v Moodle, ale uživatel se automaticky ihned ověřuje novým heslem.

Synchronizace globálních skupin v Moodle aneb kohortů (anglicky cohorts) je složitější, protože se nejedná o standardní plugin jako *LDAP*. Integrace probíhá

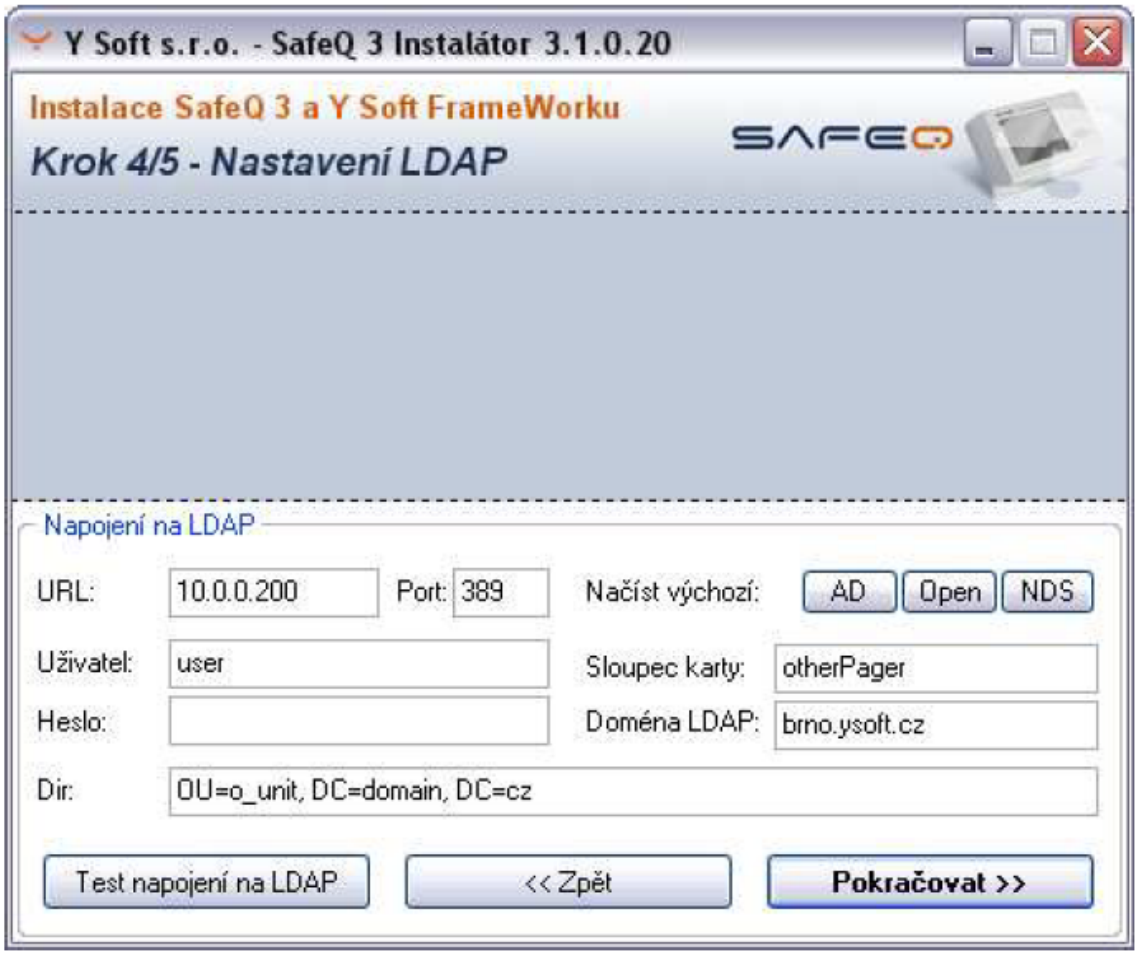
[24]URL: `http://www.selfadsi.org/create.htm`

pomocí externího PHP skriptu[25], který je nutné umístit do složky v adresáři Moodle `%MOODLE_HOME%/www/auth/cas`. Skript poté každou noc synchronizuje skupiny s Active Directory a přiřazuje do nich uživatele ze své interní databáze. Tato synchronizace značně ulehčuje práci tvůrcům kurzů (učitelé), kteří nemusejí ručně přidělovat žáky nebo studenty podle příslušnosti ke skupinám, ale pouze při vytváření kurzu vyberou požadovanou skupinu, pro kterou je kurz určen, a uživatelé už jsou přiřazeni automaticky.

6.4 Integrace - YSoft SafeQ

Software pro tisk SafeQ poskytuje také integrační modul pro napojení k Active Directory, jediným problémem bylo tedy určit, které uživatele má do své lokální databáze synchronizovat a poté vkládat do Active Directory číslo čipu. Výběr uživatelů se uskutečňuje na základě „search filteru“ do Active Directory, který vybere jen uživatele, kteří mají vyplněn atribut číslo čipu a zároveň atribut, který uvádí, že je daná osoba synchronizována s integračním nástrojem. Active Directory neposkytuje ve svém základním schématu žádný atribut k podobným účelům jako číslo čipu, a přidání takového atributu do schématu není triviálním úkolem, proto byl použit běžný, ale pro účely školy nepoužívaný, atribut `title`. Dalším důvodem k výběru tohoto atribut je fakt, že ho lze zobrazit v přehledu programu *Uživatelé a počítače služby Active Directory*. Na obrázku 6.1 je vidět jedna z obrazovek nastavení propojení SafeQ na protokol LDAP - Active Directory.

[25] domovská stránka `https://github.com/patrickpollet/moodle_local_ldap` případně `https://tracker.moodle.org/browse/MDL-25011`



Obrázek 6.1: Nastavení propojení SafeQ - LDAP, převzato z [10]

7 | Zhodnocení

Integrační aplikace INUM je od konce ledna 2014 nasazena v produkčním prostředí zákazníka, kterým je Vyšší odborná škola a Střední škola veterinární, zemědělská a zdravotnická Třebíč. Od uvedení do produkčního provozu příliš změn v logice nebo chování aplikace neproběhlo, nepřišlo se totiž na žádné závažné chyby, které by se musely akutně opravit. Proces zadávání uživatelů se značně zjednodušil, v současné situaci je nutné zadávat data pouze do softwaru Bakaláři, pak už přenos dat mezi integrovanými aplikacemi probíhá automaticky.

Integrační část je hotová, občas se však najdou podněty pro změnu. Z nedávných případů lze uvést například přidání algoritmu pro hlídání a upozorňování na duplicitní čipy v databázi, aby nedocházelo k vícenásobnému výskytu a chybám plynoucím z těchto duplikátů. Tyto návrhy jsou postupně zapracovávány do aplikace v souladu s požadavky administrátora v Třebíči. V současnosti už se aplikace stará o výměnu dat mezi čtyřmi hlavními aplikacemi v produkčním prostředí. Plně nahradila zmíněný VBA skript pro vytváření uživatelů do Active Directory, navíc administrátor sítě nyní může pročistit samotné úložiště, protože už ví, které osoby jsou aktivní (synchronizovány) a které jsou jen pozůstatkem z dřívější doby. Dále mohou uživatelé přistupovat do Moodle, stačí jim k tomu stejné přihlašovací údaje jako do domény. Jsou zařazeni do svých kurzů a mohou nyní naplno využívat možností e-learningu. SafeQ je také napojen a plně synchronizován.

Každou noc ve 3:15 proběhne načtení informací o žácích, studentech, učitelích, třídách a skupinách do interní databáze integrační aplikace, jež zabere velmi málo času, necelou minutu. Následuje ve 3:45 aktualizace dat z databáze do adresářové služby Active Directory, která už trvá déle, přibližně 40 minut, a to protože probíhá spouštění Powershell skriptů a také protože synchronizace probíhá úplně na všech uživatelích, včetně neaktivních. Z tohoto pohledu proběhne v nejbližší době změna algoritmu pro výběr uživatelů, kteří se mají synchronizovat, aby se podstatně snížila doba přenosu. Na druhou stranu se nepředpokládá růst objemu dat, takže se aktuální stav dá považovat za konstantní, časový interval by se tedy do budoucna

neměl zvyšovat. Přenášejí se data o uživatelích a skupinách, ke kterým uživatelé přísluší - třídy pro žáky a studenty, a pak také globální skupiny rozdělující uživatele na učitele, externí učitele, studenty, žáky a další externí osoby. Synchronizací všech uživatelů se dosáhlo efektu, že administrátor školy snadno určí, které osoby jsou synchronizovány (tj. jsou v databázi INUM) a které osoby už by se v Active Directory neměly vyskytovat (jsou pozůstatkem z původní ručního a nekontrolovaného zakládání uživatelů).

Poté probíhá proces synchronizace globálních skupin (kohortů) a informací o uživatelích do databáze Moodle, o jeho časování se stará plánovač zakomponovaný v e-learningovém systému - začátek je ve 4:00, právě proto by mělo smysl zkrátit čas synchronizace do Active Directory nebo změnit časování. Synchronizace Ysoft SafeQ je plně v režii softwaru, je iniciována ze strany programu, ne externího systému.

7.1 Výsledky

Integrační aplikace ukládá zprávy o své činnosti do logu, který je uchováván po dobu 30 posledních dní. Takové logy existují dva, jeden klasický souborový log a druhý jen pro významné situace, který je nazván systémový log - ten je uložen v databázi. Pokud nastane v aplikaci výjimka či chyba, odešle se e-mail s textem chyby do schránek uživatelů nastavených jako administrátoři aplikace. Poslední chyba nastala 4.2.2014, takže už 3 měsíce neustálého provozu běží aplikace bezchybně.

Výsledná aplikace se jmenuje Nástroj pro integraci uživatelských účtů, proto je v tabulce 7.1 uveden přehled počtů synchronizovaných uživatelských účtů obsažených v databázi nástroje. Všechny entity až na externí žáky jsou do databáze načteny z programu Bakaláři, externí žáci se zadávají přes webové rozhraní aplikace.

7.2 Budoucí vývoj

Vývoj integrační aplikace není zdaleka u konce, naopak se plánuje intenzivní rozvoj. Ten by mohl spočívat v napojení čím dál většího množství aplikací, vytvoření konektorů na ně a možnosti větší konfigurovatelnosti aplikace, tak aby se při

Tabulka 7.1: Synchronizované účty (stav k 4.5.2014)

objekt	**počet aktivních**	**počet neaktivních**
žák	354	91
student	138	156
učitel	62	81
externí učitel	22	2
externí žák	79	0
celkem	**655**	**330**

případném nasazení do jiného prostředí nemusel příliš měnit zdrojový kód. Potenciál pro integraci jiných programů je díky robustnosti aplikace postavené na Spring Framework velký, at' už se bude jednat o zbývající aplikace v prostředí zákazníka (Karas, Jídelna, Docházka) nebo například integrace nástrojů od firmy Google (Google Apss - kalendář, kontakty, e-maily). INUM totiž již nyní obsahuje velké množství informací o uživatelích, tudíž už většinou stačí jen najít způsob jak do cílové aplikace data synchronizovat (jak ji integrovat).

8 | Závěr

Nástroj pro integraci uživatelských účtů bezproblémově funguje v produkčním prostředí školy po dobu více jak třech měsíců, od konce ledna 2014. Aplikace rapidně zlepšuje přehlednost a pořádek dat, velmi snižuje redundanci jejich zadávání do různých systémů, zjednodušuje správu, zamezuje chybám vznikajícím při ručním vkládání informací a hlavně zvyšuje bezpečnost prostředí, zamezuje totiž přístup do aplikací uživatelům, kteří už nejsou součástí školy. V současné době existují v prostředí pouze 2 vstupní body pro zadávání dat - software Bakaláři a samotná integrační aplikace. Každá integrovaná aplikace nyní obsahuje aktuální, pravidelně synchronizované údaje a informace jen potřebné pro svůj běh, žádné nadbytečnosti.

V rámci nasazení aplikace proběhlo i školení administrátora na ovládání a konfiguraci nástroje. Zákazník projevil spokojenost s výsledkem integrace, důkazem může být objednávka na integraci zbylých školních systémů, které byly vyloučeny z první fáze projektu. Realizace této části by měla proběhnout v nejbližších týdnech.

INUM má velký potenciál rozvoje a implementace do jiných prostředí, většinou středních škol, které se potýkají se stejnými problémy jako škola v Třebíči. Nasazení do cizího prostředí bude spíše otázkou konfigurace, než programování nové logiky aplikace, nástroj byl navržen s důrazem na přenositelnost. Byl použit multiplatformní jazyk Java a projekty jako Spring Framework či Apache Tomcat, které lze použít jak v prostředí Windows, tak i například na serverech pod distribucí Linux.

Z osobního hlediska považuji za přínos možnost zkusit si navrhnout a naprogramovat nástroj pro skutečného zákazníka a spolupracovat s lidmi pohybujícími se v IT prostředí. Musel jsem se kvůli implementaci naučit pracovat s velkým množstvím nástrojů a myslím, že budu schopen v budoucnu tyto cenné zkušenosti využít u jiných projektů.

Seznam použitých zkratek

AD - Active Directory

DBF - dBase file

DN - Distinguished Name

HTTP - Hypertext Transfer Protocol

INUM - Integrated User Manager

JDBC - Java Database Connectivity

JVM - Java Virtual Machine

LDAP - Lightweight Directory Access Protocol

ODBC - Open Database Connectivity

ORM - Object/Relational Mapping

POJO - Plain Old Java Object

SQL - Structured Query Language

SSL - Secure Sockets Layer

URL - Uniform Resource Locator

VBA - Visual Basic for Applications

Literatura

[1] Alex, B.; Taylor, L.: *Spring Security Reference Documentation*. Spring Security, 2013, [online]. [navštíveno 27.4.2014]. Dostupné z: `http://docs.spring.io/spring-security/site/docs/3.1.4.RELEASE/reference/springsecurity.html`

[2] Arthursson, M.; Sandberg, U.; Dalquist, E.; aj.: *Spring LDAP - Reference Documentation*. Spring LDAP, 2014, [online]. [navštíveno 27.4.2014]. Dostupné z: `http://docs.spring.io/spring-ldap/docs/1.3.2.RELEASE/reference/html/`

[3] Bakaláři: *Evidence, Klasifikace, Vysvědčení*. 2014, [online]. [navštíveno 17.4.2014]. Dostupné z: `http://www.bakalari.cz/evidence.aspx`

[4] Hibernate: *Hibernate Reference Documentation*. Hibernate, 2014, [online]. [navštíveno 27.4.2014]. Dostupné z: `http://docs.jboss.org/hibernate/orm/4.2/manual/en-US/html/`

[5] Johnson, R.; Hoeller, J.; Donald, K.; aj.: *Spring Framework Reference Documentation*. Spring Framework, 2013, [online]. [navštíveno 27.4.2014]. Dostupné z: `http://docs.spring.io/spring/docs/3.2.4.BUILD-SNAPSHOT/spring-framework-reference/htmlsingle/#spring-introduction`

[6] Malina, P.: *Jak vyzrát na Powershell 2.0*. Brno: Computer Press. a.s., 2010, ISBN 978-80-251-2732-2.

[7] Moodle: *About Moodle*. Moodle project, Březen 2014, [online]. [navštíveno 27.4.2014]. Dostupné z: `http://docs.moodle.org/26/en/About_Moodle`

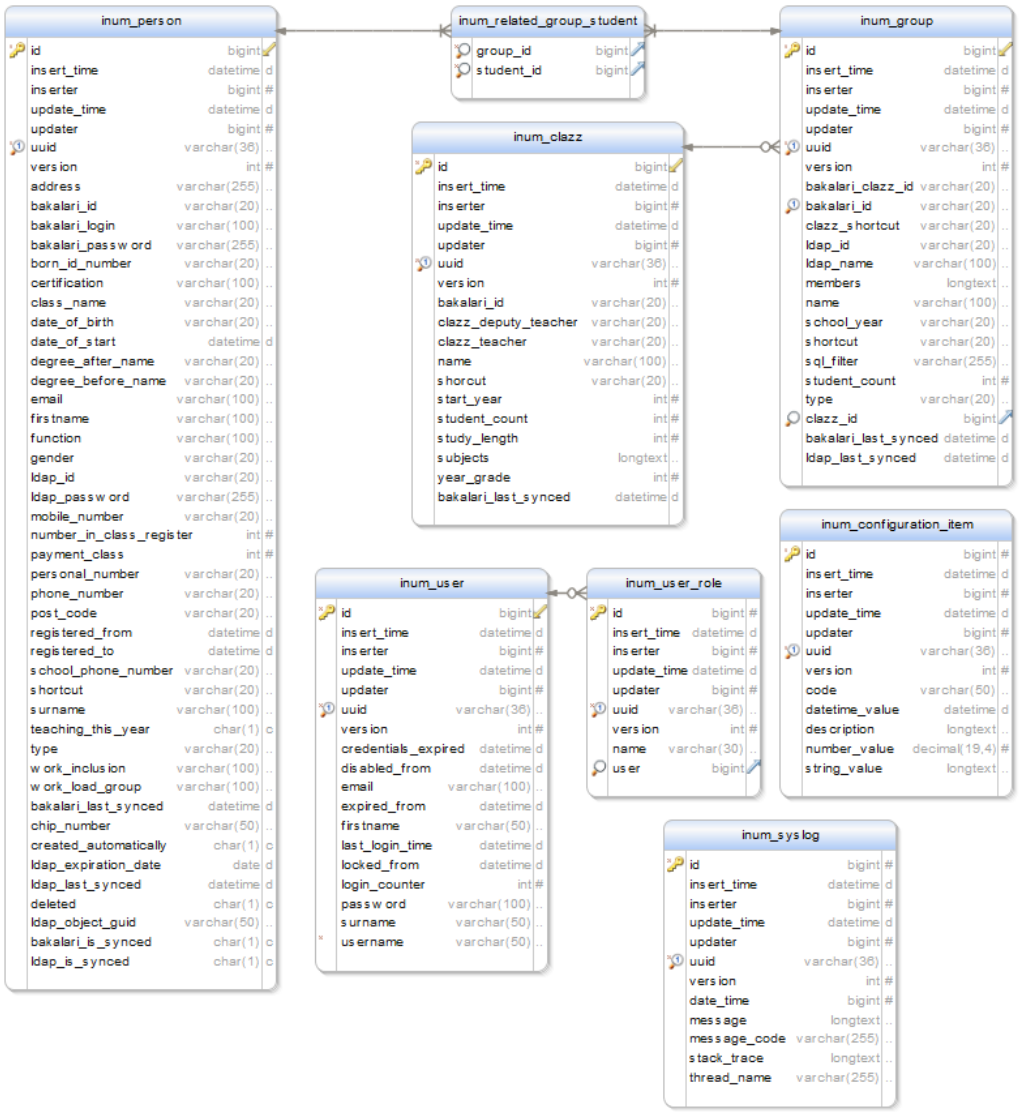
[8] stackoverflow: *How to manage AD ACLs using Java*. stackoverflow, 2013, [online]. [navštíveno 20.4.2014]. Dostupné z: `http://stackoverflow.com/questions/12784606/how-to-manage-ad-acls-using-java`

[9] Stanek, W. R.: *Active Directory - Kapesní rádce administrátora*. Brno: Computer Press. a.s., 2009, ISBN 978-80-251-2555-7.

[10] Vybíral, F.: *SafeQ: příručka administrátora v3.0*. Y Soft, s.r.o., 2005, [online]. [navštíveno 27.4.2014]. Dostupné z: `https://helpdesk.minolta.cz/hds/DownloadFile.jsp?id=4422`

[11] Wikipedia: *Java Database Connectivity*. Březen 2014, [online]. [navštíveno 27.4.2014]. Dostupné z: `http://en.wikipedia.org/wiki/Java_Database_Connectivity`

A | Kompletní datový model



B | Integrace Moodle s AD

Nastavení serveru LDAP

URL hostitele	192.168.4.1	Zadejte URL serveru LDAP. Například 'ldap://ldap.naseskola.cz/' nebo 'ldaps://ldap.naseskola.cz/'
Verze	3	Verze protokolu LDAP, kterou používá váš server.
Use TLS	Ne	Use regular LDAP service (port 389) with TLS encryption
Kódování LDAP	UTF-8	Specifikujte kódování, používané serverem LDAP. Nejpravděpodobněji utf-8, MS AD v2 implicitně používá takové platformy kódování, jako cp1252, cp1250 atd.
Page Size	999	Make sure this value is smaller than your LDAP server result set size limit (the maximum number of entries that can be returned in a single query)

Nastavení bind

Skrýt hesla	Ano	Má se zabránit ukládání hesel v databázi Moodlu? Zvolíte-li "ano", nebudou hesla v databázi ukládána.
Jednoznačné jméno (DN)	koboz@szstrebic.local	Chcete-li používat nějaký účet (bind-user) k vyhledání uživatelů, specifikujte jej zde. Příklad: 'cn=ldapuser,ou=public,o=org'
Heslo	•••••••••	Heslo pro bind-user

Nastavení prohledávání uživatelů

Typ uživatele	MS ActiveDirectory	Vyberte, jak jsou uživatelé ukládáni v LDAP. Toto nastavení zároveň určuje, jak se bude chovat vypršení hesla (login expiration), přihlášení z milosti (grace logins) a tvorba uživatelů.
Kontexty	cn=users,dc=szstrebic,dc=local	Seznam kontextů, ve kterých se nacházejí uživatelé. Jednotlivé kontexty oddělujte středníkem. Příklad: 'ou=uzivatele,o=naseskola; ou=dalsi,o=naseskola'

C | Obsah přiloženého CD

- /
  - `install` ..... složka s implementovaným nástrojem
    - `foxpro-oledb` ..... instalační balíček s ovladači pro čtení z DBF souborů
    - `install-cert` ..... Java nástroj pro instalaci certifikátu do úložiště JVM
    - `UnlimitedJCEPolicyJDK7` ..... balíček pro JVM, zprovoznění Jasypt
    - `apache-tomcat-7.0.53-windows-x86.zip` ..... webový server
  - `other` ..... ostatní soubory
    - `sync_cohorts.php` ..... PHP skript pro Moodle
  - `text` ..... složka s textem bakalářská práce
    - `inum.pdf` ..... text práce ve formátu PDF
    - `latex` ..... složka se zdrojovými soubory ve formátu LaTeX
  - `tool` ..... složka s implementovaným nástrojem
    - `inum` ..... zdrojové kódy integrační aplikace
    - `instalace.txt` ..... návod pro instalaci nástroje