

Tvorba moderních webových aplikací pomocí SAPUI5 a oDATA služeb

Diplomová práce

Vedoucí práce:

Ing. Jiří Lýsek, Ph.D.

Bc. Lubomír Filipek

Brno 2015

Děkuji svému vedoucímu, Ing. Jiřímu Lýskovi, Ph.D., za odborné rady a připomínky při vypracování této diplomové práce. Děkuji také mé rodině za obrovskou podporu a trpělivost.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Tvorba moderních webových aplikací pomocí SAPUI5 a oDATA služeb**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 27. prosince 2015

Abstract

Filipek, L. Creation of modern web applications using SAPUI5 and OData services. Thesis. Brno, 2015

The thesis contains comparison JavaScript MVC frameworks. The following is a description of the current system FAIN for the inventory of fixed assets at VUT Brno, describes its shortcomings. It is further explored what is happening in the background on SAP ERP system, where the system FAIN is integrated. Based on these findings, it is designed and implemented OData service at SAP NetWeaver Gateway using ABAP language. And the SAPUI5 application for the inventory of fixed assets, which uses this service. The second designed and implemented SAPUI5 application uses the Northwind OData service. There are described the advantages and disadvantages of SAPUI5 application and previous solution of inventory of fixed assets. In conclusion, there are proposed possible solutions disadvantages of created application.

Keywords

ABAP, ERP system SAP, FAIN, inventory of fixed assets, JavaScript MVC Framework, OData, SAP NetWeaver Gateway, SAPUI5.

Abstrakt

Filipek, L. Tvorba moderních webových aplikací pomocí SAPUI5 a oDATA služeb. Diplomová práce. Brno, 2015

Diplomová práce obsahuje porovnání JavaScript MVC frameworků. Následuje popis stávajícího systému FAIN pro inventuru hmotného majetku na VUT Brno, jsou popsány jeho nedostatky. Dále je prozkoumáno co se děje na pozadí ERP systému SAP, kde je systém FAIN integrován. Na základě těchto poznatků je navržena a implementována OData služba na SAP NetWeaver Gateway v jazyce ABAP. A SAPUI5 aplikace pro inventuru hmotného majetku, která tuto službu využívá. Druhá navržená a implementovaná SAPUI5 aplikace využívá OData službu Northwind. Jsou popsány výhody a nevýhody SAPUI5 aplikace a předchozího řešení inventury hmotného majetku. V závěru jsou navrženy možnosti řešení nevýhod vytvořené aplikace.

Klíčová slova

ABAP, ERP systém SAP, FAIN, inventura hmotného majetku, JavaScript MVC Framework, OData, SAP NetWeaver Gateway, SAPUI5.

Obsah

1	Úvod a cíl práce	13
1.1	Úvod.....	13
1.2	Cíl práce.....	13
2	JavaScript MVC frameworky	14
2.1	MVC.....	14
2.2	Výběr JavaScript MVC frameworku.....	16
2.2.1	Obecné vlastnosti vhodného frameworku	16
2.2.2	Požadované funkce frameworku	18
2.3	SAPUI5: UI Development Toolkit for HTML5	19
2.4	OpenUI5	20
3	Současný systém pro inventuru hmotného majetku na VUT	22
3.1	Funkcionalita na straně systému SAP	22
3.1.1	Tisk etiket.....	23
3.1.2	Inventura	24
3.1.3	Inventurní seznamy	26
3.1.4	Exporty/importy dat.....	27
3.1.5	Zobrazení inventurního seznamu.....	30
3.1.6	Zpracování inventurního seznamu	31
3.1.7	Řešení konfliktů	31
3.2	Funkcionalita mobilního terminálu PDT.....	32
3.2.1	Přihlášení.....	32
3.2.2	Nastavení filtrů.....	32
3.2.3	Zpracování majetku.....	32
3.2.4	Manuální potvrzení majetku.....	34
3.3	Nevýhody systému.....	34
3.4	Co se děje na pozadí	36
4	OData služba	38

4.1	Metadata	40
4.2	Entity Data Model (EDM).....	40
5	SAPUI5 aplikace pro inventuru hmotného majetku	42
5.1	Požadavky na aplikaci.....	42
5.1.1	Funkční požadavky	42
5.1.2	Nefunkční požadavky.....	42
5.2	Entity Data Model (EDM).....	43
5.3	OData Služba v ERP systému SAP	43
5.4	Implementace OData služby v SAP	45
5.4.1	Vytvoření nového projektu.....	45
5.4.2	Vytvoření Entity Type.....	45
5.4.3	Vytvoření Properties (vlastností)	46
5.4.4	Definice navigace.....	46
5.4.5	Vytvoření Entity Set.....	47
5.4.6	Generování podkladových tříd ABAP	47
5.4.7	Registrace a aktivace služby na SAP NetWeaver Gateway.....	48
5.4.8	Implementace metod třídy Data Provider class	49
5.5	Návrh GUI	51
5.5.1	Výběr inventury	51
5.5.2	Výběr umístění	52
5.5.3	Výběr majetku.....	52
5.5.4	Formulář majetku	53
5.5.5	Editace majetku.....	53
5.6	Implementace	53
5.6.1	Views	55
5.6.2	Controllery	56
5.6.3	Apache Cordova	58
6	SAPUI5 aplikace nevyužívající SAP zdroje	60
6.1	Požadavky na aplikaci.....	60
6.2	Entity Data Model (EDM).....	60
6.3	Návrh GUI	60

Obsah	11
7 Závěr	61
8 Literatura	63
A SAPUI5 Runtime & UI Libraries	66
B ERD využívané části databáze v ERP SAP	67
C EDM OData služby pro inventuru majetku	68
D GUI aplikace pro inventuru majetku	69
E Schéma aplikace pro inventuru majetku	74
F Využívaná část EDM OData služby Northwind	75
G GUI aplikace pro zobrazení zaměstnanců	76
H Obsah přiloženého CD	77

1 Úvod a cíl práce

1.1 Úvod

Společnost SAP je světovou jedničkou v poskytování ERP systémů a služeb s tímto souvisejících. Když ale v souvislosti se SAPem začneme uvažovat o user interface experience, každému se hned vybaví na první pohled komplikovaný desktopový klient pro přístup k SAP systému SAP GUI, který je pro neznalého uživatele neintuitivní a bez předchozího rozsáhlého zaškolení jen těžko použitelný. Další user interface technologie SAP jako jsou Classic Dynpro, WebDynpro a Business Server Pages přinášejí stejnou user experience, jen trochu jiným způsobem, tyto nástroje slouží dobře, ale je tu velký prostor ke zlepšení. Společnost SAP si tohoto nedostatku byla vědoma, a proto vytvořila novou technologii pro tvorbu uživatelského rozhraní *User Interface Development Toolkit for HTML5*, zkráceně SAPUI5.

SAPUI5 je platforma pro vývoj bohatých uživatelských rozhraní pro moderní webové business aplikace. Hlavním přínosem je využití JavaScriptu a HTML5 pro generování uživatelského rozhraní. Do webového prohlížeče uživatele je přenesen kód obsahující instrukce, jak má uživatelské rozhraní vypadat. Vykreslení uživatelského rozhraní je tak nezávislé na aplikačním serveru SAP a využívá výkon koncového zařízení. Komunikace se serverem SAP slouží pouze k výměně business dat. Odezva aplikace je tak výrazně kratší než u předchozích webových technologií SAP.

Další důležitou vlastností SAPUI5 je možnost tvorby aplikací pro mobilní zařízení – smartphony a tablety. V dnešní době má téměř každý v kapse smartphone s připojením k internetu. Mobilní aplikace jsou oblíbené a staly se mainstreamem, na kanálu Prima Cool televizní stanice Prima TV se vysílá pořad Aplikace, jehož obsahem jsou novinky ve světě mobilních aplikací. Uživatelská základna pro mobilní aplikace je obrovská. Možnost ovládat některé činnosti v ERP systému prostřednictvím smartphonu je pro uživatele výhodou, protože nemusí kvůli provedení operace vždy sedět u počítače. V případě fyzické inventury hmotného majetku SAPUI5 umožňuje kooperaci mobilního zařízení, které načítá čárové kódy majetků a ERP systému, kde je majetek evidován.

1.2 Cíl práce

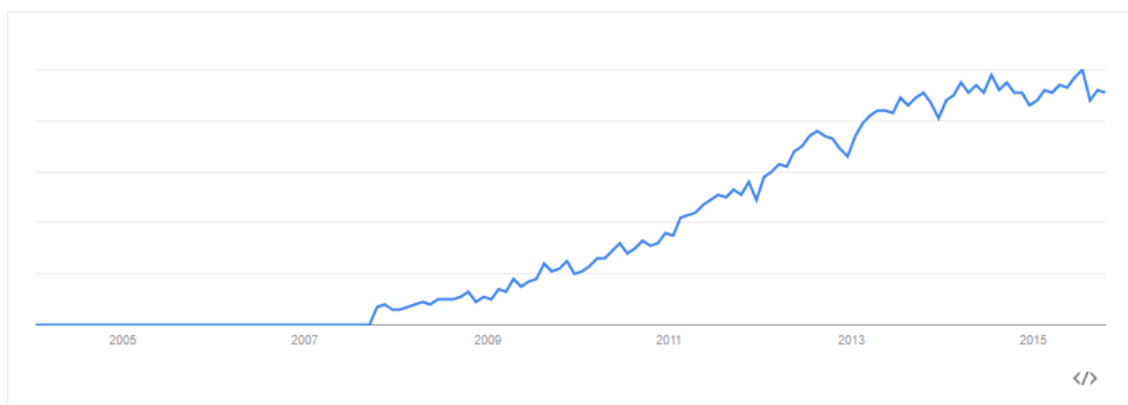
Cílem práce je seznámit se s SAPUI5 frameworkem a s jeho použitím vytvořit dvě aplikace. První aplikace bude sloužit pro inventuru hmotného majetku pomocí čárového kódu na VUT, tato aplikace bude alternativou (front end částí) současného řešení, které dodala firma Oriwin s. r. o. SAPUI5 aplikace bude provádět snímání čárového kódu a okamžitý přenos do SAP (Wi-Fi, mobilní data). Na rozdíl od současného řešení, které využívá Windows Mobile based čtečky čárového kódu, které jsou v offline módu (tzn., nesou si určitý export majetkových dat ze SAP ve své lokální SQL databázi). Druhá aplikace bude sloužit pro demonstraci, že na back endu SAPUI5 aplikace nemusí být systém SAP.

2 JavaScript MVC frameworky

Při psaní webové aplikace od nuly, snadno propadneme pocitu, že bude stačit, když se budeme spoléhat pouze na knihovnu pro DOM manipulaci (jako je jQuery) a několik utility pluginů. Ale ve skutečnosti je velmi pravděpodobné, že se s takovým přístupem brzy ztratíme v zanořené hromadě jQuery callbacků a DOM elementů bez jakékoliv reálné struktury v místě pro naši aplikaci.

Jinak řečeno, výsledkem bude špagetový kód, tedy kód špatně čitelný a obtížně udržovatelný, kde je potřeba pochopit celý kód, abychom pochopili malou část programu.

Z tohoto důvodu se začaly okolo roku 2007 objevovat JavaScript MVC frameworky. Na obr. 1 je vidět zájem v průběhu času. Tyto frameworky při vývoji pomáhají s vytvářením struktury a organizace projektů a zlepšují snadnou udržovatelnost v dlouhodobém horizontu.



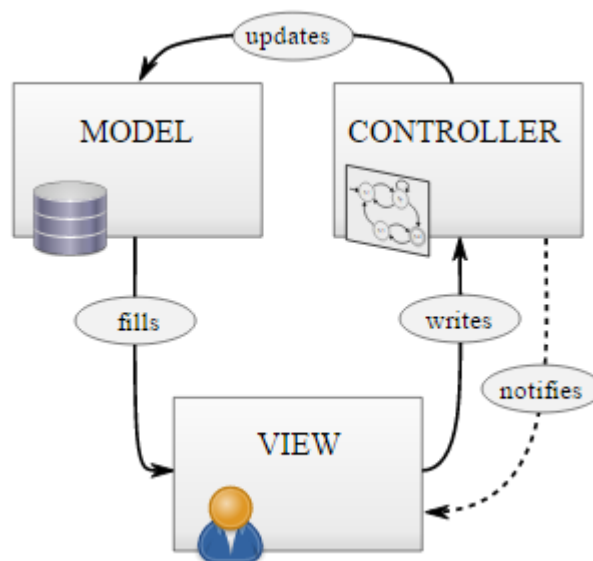
Obr. 1 Zájem v průběhu času, vyhledávací dotaz JavaScript MVC, celosvětově

Zdroj: Google Trends, osa y je bez popisu, Google tuto informaci nezveřejňuje z obchodních důvodů

Jak uvádí Žára (2013) s použitím JavaScript šablon je možné se při vývoji držet softwarové architektury MVC i bez pomoci JavaScript MVC frameworku. I přesto například Steigerwald (2013) doporučuje JavaScript MVC frameworky používat.

2.1 MVC

JavaScript MVC frameworky umožňují vývojářům snadno organizovat kód pomocí vzoru známého jako MVC (Model-View-Controller). MVC je softwarová architektura, která rozděluje aplikaci na tři na sobě nezávislé komponenty: datový model, uživatelské rozhraní a řídicí logika. Na obr. 2 je znázorněna spolupráce komponent MVC.



Obr. 2 MVC Diagram (Model-View-Controller)
Zdroj: Wikimedia Commons, the free media repository

- **Model** – reprezentuje správu vlastních dat, nad nimiž aplikace pracuje: to může být třeba obrázek, textový dokument, databáze uložená na serveru SQL, ... Je dobré si uvědomit rozdíl mezi daty samotnými a jejich správcem (tedy právě tím modelem, o němž zde hovoříme). Třeba takový konkrétní obrázek jsou data; skupina tříd, umožňující obrázky načítat ze souborů a opět do nich ukládat, zjišťovat a měnit jejich atributy, převádět jejich formáty a pracovat s jejich obsahem – to je správce. V běžící aplikaci pak samozřejmě bude základním objektem vrstvy modelu nějaká vhodná instance, jež bude reprezentovat vlastní obrázek a nabízet všechny odpovídající služby.
- **View** – zahrnuje všechny objekty (grafického) uživatelského rozhraní a jejich služby. Uživatel aplikace s ní komunikuje výhradně prostřednictvím těchto objektů; právě ony mu prezentují data z modelu ve vhodné formě, a naopak pouze jejich prostřednictvím uživatel dává aplikaci příkazy, jež určují její další činnost.
View má s modelem společnou tu nejdůležitější věc: *jeho objekty nejsou závislé na konkrétní aplikační logice*. Grafické uživatelské rozhraní aplikace tak můžeme podle potřeby (a podle požadavků zákazníků) snadno kdykoliv měnit, aniž by bylo zapotřebí přitom nějak zasahovat do aplikační logiky (nebo dokonce do modelu).
- **Controller** – mezi datovým modelem a objekty grafického uživatelského rozhraní, jež tvoří vzhled, stojí vrstva controller; právě na její úrovni je implementována funkční logika aplikace, takové věci jako „stiskne-li uživatel tohle tlačítko, provede se támhleta akce, a v tomto textovém poli se zobrazí výsledek“. (Čada, 2009)

JavaScript „MVC“ frameworky, ne vždy striktně dodržují výše popsanou strukturu softwarové architektury MVC. Některé frameworky zahrnují odpovědnosti Controlleru ve View (např. Backbone.js), zatímco jiné přidávají své vlastní dogmatické komponenty. Je to z toho důvodu, že autoři návrhů těchto frameworků byli přesvědčeni, že takové řešení bude efektivnější. Pro tyto frameworky je pak přesnější označení JavaScript MV* framework.

2.2 Výběr JavaScript MVC frameworku

V současné době je na výběr až příliš mnoho možností. Různí JavaScript vývojáři mají různé představy o tom, jak by správně měla být organizována škálovatelná JavaScript aplikace. Proto vznikají stále další frameworky s jinou realizací MV* architektury. Tento stav často se objevujících nových frameworků je nazýván Yet Another Framework Syndrome (nebo YAFS). Na jednu stranu je vidět snaha o inovaci, ale na druhou stranu YAFS může způsobovat zmatek a frustraci, když vývojář chce začít vyvíjet aplikaci, ale nechce manuálně vyhodnocovat mnoho různých možností, aby vybral pro danou aplikaci nejvhodnější framework.

2.2.1 Obecné vlastnosti vhodného frameworku

Pokud se chceme vyhonout problémům „Early adopters“, je vhodné na začátku výběru odfiltrovat frameworky, které nesplňují následující vlastnosti:

- **Framework se prosadil v reálném nasazení** – Ujistíme se, že se framework používá na produkci, vývojáři skutečně vytvořili a nasadili rozsáhlou aplikaci, zároveň se můžeme inspirovat, co je možné s frameworkem vytvořit. Například Backbone.js se používá, mimo jiné v rozsáhlých projektech Sony Entertainment Network a LinkedIn Mobile.
- **Framework má detailní dokumentaci** – Každý solidní framework by měl mít podrobný soubor dokumentace, která provede vývojáře jeho používáním. Nejlepším způsobem, jak u frameworku zjistit, co jeho API podporuje, jak s ním mohou být vytvořeny běžné úkoly nebo komponenty, a jaké jsou nejčastější problémy je právě čtení oficiální dokumentace. Spoléhat se jen na komunitu a vlastní objevování, je často příliš časově náročné.
- **Framework je aktivní** – tedy framework se stále vyvíjí, reaguje na nové trendy a existuje alespoň nějaká záruka, že tomu tak bude i nadále. V případě Open-source frameworků existuje aktivní komunita přispěvatelů projektu, a uživatelů, kteří jsou schopni pomoci, pokud se dostaneme do problémů. Například prostřednictvím komunity Stack Overflow. U proprietárních frameworků vydávání nových verzí a podpora zajišťuje výrobce.
- **Celková velikost frameworku, podpora modulárního systému** – je dobré vědět, jaké má framework závislosti. Často se uvádí pouze celková velikost samotné základní knihovny, ale už se neuvádí velikosti závislých knihoven. Častá je závislost na jQuery.

- **Zdrojový kód aplikace vytvořené pomocí frameworku** – není možné objektivně hodnotit, záleží na preferencích každého vývojáře. Na zkoumání zdrojového kódu vznikl užitečný projekt TodoMVC, dostupný na webu¹. TodoMVC nabízí jednu Todo aplikaci implementovanou ve většině dnes populárních JavaScript MV* frameworků. Výsledná aplikace vypadá vždy stejně, funkcionality je také totožná, takže je jednoduché porovnat syntaxi a strukturu různých frameworků.

V tab. 1 je přehled vlastností a oblíbenosti v systému GitHub SAPUI5 a dalších populárních frameworků.

Tab. 1 Vlastnosti frameworků a jejich oblíbenost v systému GitHub

	SAPUI5	Angular	React	Backbone	Ember
Uvedení	2010	2009	2013	2010	2007
Původ/Základ	jQuery	GetAngular	-	Underscore	SproutCore, Handlebars
Hlavní přispěvatel	SAP	Google	Facebook, Instagram	J. Ashkenas	Y. Katz
Hlavní stránky	SAP	YouTube, Sky store	Netflix, PayPal	Wordpress, LinkedIn	Yahoo
Přispěvatelé	80	1 332	528	276	535
Komity	10 473	7 163	5 550	3 167	11 171
Forky	304	19 551	4 526	5 161	3 194
Hvězdy	672	43 284	29 616	23 179	14 894

Zdroj: developereconomics.com/feature-comparison-of-4-popular-js-mv-frameworks, github.com

Parametry systému GitHub v tabulce ve sloupci SAPUI5 se vztahují k open source verzi frameworku OpenUI5, protože SAPUI5 se v systému GitHub nenachází. Obě verze frameworku budou popsány dále.

- **Uvedení** – rok uvedení, SAPUI5 byl vyvíjen několik let před uvedením pod kódovým názvem *Phoenix*, odtud také pochází logo ptáka. (Kunz, 2013)
- **Původ/Základ** – z čeho framework pochází, nebo na čem je založen.
- **Hlavní přispěvatel** – Hlavní přispěvatel a také autor frameworku.
- **Hlavní stránky** – významné stránky používající framework.
- **Přispěvatelé** – počet přispěvatelů, tento a zbývající řádky tabulky pochází ze statistik systému GitHub. Slouží pro představu jaký je o frameworky zájem a jak jsou oblíbené.
- **Komity** – počet komitů.
- **Forky** – počet verzí forků, které vytvořili vývojáři a pracují na nich.

¹ <http://todomvc.com/>

- **Hvězdy** – počet uživatelů, kteří dali frameworku hvězdičku.

2.2.2 Požadované funkce frameworku

Po odfiltrování frameworků nesplňujících výše uvedené obecné předpoklady se můžeme zaměřit na to, jak dobře zbývající frameworky splňují požadavky našeho konkrétního projektu, který se chystáme vytvořit:

- **UI Vazba (binding)** – pokud dojde ke změně dat v podkladovém Modelu, UI Vazba zajistí automatickou aktualizaci View.
- **OData podpora** – data do Modelů vytvářené aplikace se budou načítat ze SAP backendu pomocí OData služby, je tedy výhodou, pokud Framework podporuje OData out-of-the-box.
- **Směrování (routing)** – navigace uvnitř aplikace je svázána s URL a session historií v prohlížeči. Konkrétní stav aplikace je tedy zaznamenán v URL, typicky v části za znakem #. Při spuštění aplikace pomocí takového URL se bude aplikace nacházet přesně v takovém stavu, v jakém jsme URL uložili. To je výhodné pokud URL uložíme, vytvoříme záložku v prohlížeči, nebo někomu pošleme. Routing většinou také využívá nativní history API prohlížeče.
- **SAP podpora** – vytvářená aplikace bude využívat na backendu SAP systém, SAP poskytuje pokročilé debugovací nástroje.
- **Znovupoužitelné komponenty** – pokud je v plánu vývoj více než jedné aplikace, znovupoužití kódu znamená úsporu práce.
- **UI libraries** – pokud má framework své knihovny UI prvků, které jsou konzistentní v ohledu user experience, pak všechny aplikace postavené pomocí těchto knihoven budou také konzistentní. Designová konzistence všech business aplikací ve firmě je velmi žádaná vlastnost. Ideální stav je, když ve firmě všechny aplikace vypadají stejně a fungují stejně.

Tab. 2 znázorňuje dostupnost požadovaných funkcí u SAPUI5 a dalších populárních frameworků. Jestli je funkce dostupná nebo nedostupná je hodnoceno u frameworků ve stavu tak jak jsou out-of-the-box. Některé funkce je možné do frameworků dodatečně přidávat, například podporu OData služeb lze přidat začleněním knihovny datajs.

Tab. 2 Přehled dostupných funkcí ve frameworkcích

	SAPUI5	Angular	React	Backbone	Ember
UI Vazba	■	■	■		■
OData	■				
Směrování	■	■		■	■
SAP podpora	■				
Komponenty	■	■	■	■	■
UI knihovny	■		■		

Zdroj: developereconomics.com/feature-comparison-of-4-popular-js-mv-frameworks

Z tabulky je možné zjistit, že SAPUI5 je jediný framework, který splňuje všechny požadované funkce a je tedy nejlepší volbou pro vývoj zamýšlených aplikací. Hlavní výhody SAPUI5 jsou v SAP podpoře, OData a usnadnění vývoje aplikací s konzistentním user experience díky UI knihovnám. Pokud se chystáme vyvíjet aplikace, které budou na backendu používat systém SAP, je vhodné použít SAPUI5. Pokud ale vyvíjená aplikace na backendu nebude mít systém SAP, zůstává open source verze OpenUI5 jen výhodou UI knihoven, které ale nemusí každému vyhovovat.

Pokud se podíváme do tab. 1 na čísla z GitHubu, zjistíme, že všechna ve sloupci SAPUI5 jsou řádově nižší, než u ostatních frameworků, kromě počtu komitů. To příliš nesvědčí o velké oblíbenosti a používanosti. Druhý nevyšší počet komitů je pravděpodobně zapříčiněn tím, že SAP na SAPUI5 intenzivně pracuje, poskytuje podporu svým zákazníkům a drtivá většina těchto věcí se promítá i do open source verze OpenUI5. Který framework je nejpoužívanější zjistíme podle počtu forků, zde s obrovským náskokem před všemi ostatními vítězí Angular. Není to moc překvapivé, protože za tímto frameworkem stojí samotný Google. Angular je na tom nejlépe i v počtu hvězdiček a přispěvatelů, nicméně v počtu hvězdiček jej dohání React. Také je potřeba dodat, že React je znevýhodněn tím, že je nejmladší. Ember přestože je nejstarším frameworkem ze srovnání, v počtu hvězdiček a forků skončil na druhém nejhorším místě, z toho můžeme odvodit, že jeho sláva opadá a vývojáři se orientují na jiné frameworky.

2.3 SAPUI5: UI Development Toolkit for HTML5

SAPUI5 je sada vývojářských nástrojů, která slouží k vytváření aplikací s bohatým uživatelským rozhraním pro moderní webové business aplikace. Je to JavaScript UI knihovna stávající se z:

Jádra (core):

- Data binding a modely pro různé datové zdroje, JSON, XML a OData out of the box
- Efektivní engine pro vytváření a aktualizaci HTML ovládacích prvků
- Podpora MVC konceptu

- Podpora deklarativní konstrukce UI a HTML šablon
- Automatické načítání odpovídajících jazykových souborů

Velkého množství ovládacích prvků, které jsou organizovány v několika knihovnách:

- Více než 200 ovládacích prvků, od jednoduchých, jako je *Button* po složité, jako je *sap.m.SplitContainer* s responzivním master-detail uspořádáním a animovanými přechody stránek.
- Možnost tvorby vlastních ovládacích prvků.
- Předdefinované layouty pro typické případy použití.
- Podpora pro přístupnost, ovládání z klávesnice, dotykovou interakci, RTL jazyky.
- Vizuální témata, jsou postavena na LESS, tedy standardizovaném a otevřeném nástroji pro správu CSS. SAP vytvořil pro správu témat GUI nástroj Theme designer. Ale samozřejmě je také možné pracovat přímo s kódem, jako v jakémkoliv jiném projektu používajícím LESS.
- Debugovací nástroj SAP UI5 Diagnostics, který dokonce umožňuje měnit data v modelu a aktualizovat je uvnitř náhledu client View.

SAP používá tuto UI knihovnu pro business aplikace, kde jsou stěžejní standardy jako bezpečnost a podpora, takže jsou integrovány ochrany proti cross-site skriptování a dalším útokům a výkonné nástroje pro analýzu/kontrolu. A samozřejmě SAP hodně investuje do automatizovaných a manuálních testů, od jednotkových testů po porovnání screenshotů. (Kunz, 2013)

Pokud nahlédneme pod kapotu SAPUI5, zjistíme, že je postaven na jQuery a používá dalších 30 open source knihoven, jejich kompletní seznam je dostupný v příloze A.

2.4 OpenUI5

OpenUI5 je bezplatná verze SAPUI5 dostupná od prosince 2013 pod open source licencí Apache 2.0. Zdrojové soubory jsou dostupné na systému GitHub², to znamená následující možnosti:

- procházet, číst zdrojové kódy a přímo sledovat vývoj, který se zde odehrává
- hlásit chyby a navrhopvat vylepšení
- diskutovat nad kódem s ostatními členy komunity
- podílet se na změnách a opravách chyb

² <https://github.com/SAP/openui5/>

SAPUI5 je verze, kterou mohou používat pouze zákazníci SAP s určitým druhem licence. (Samotné použití SAPUI5 je také zdarma, ale jde o to, že tito zákazníci už zaplatili nějaký jiný produkt SAP).

Celé jádro OpenUI5 a SAPUI5 (které obsahuje všechny centrální funkce) je identické. Nejběžněji používané knihovny ovládacích prvků, které obsahují většinu existujících ovládacích prvků, jsou také identické. To vše je nyní možné používat zdarma, při použití OpenUI5.

Rozdíl oproti SAPUI5 je v některých speciálních knihovnách ovládacích prvků, které nejsou (nebo zatím nebyly) uvedeny pod open source licencí.

Další rozdíl se týká podpory, zákazníci používající SAPUI5 mají garance podpory, takže SAP zaručuje opravy různých chyb do určitého termínu od nahlášení zákazníkem. Tyto záruky podpory samozřejmě nejsou zajištěny pro OpenUI5, částečně by to měla vynahrazovat open source komunita. Nicméně, opravy chyb udělané pro SAPUI5 dostane také OpenUI5, protože jsou stejné.

3 Současný systém pro inventuru hmotného majetku na VUT

Na VUT se pro inventarizaci hmotného investičního majetku v současné době využívá systém FAIN – Fixed Assets Inventory (verze 4.1). Systém dodala firma Oriwin s. r. o., používá čárové kódy a RFID tagy a je integrován do informačního systému SAP ERP.

Systém se dělí na dvě části, první část je integrována do systému SAP. Druhou částí je mobilní aplikace, která se používá při snímání čárových kódů majetků na přenosných zařízeních s integrovaným snímačem čárového kódu s operačním systémem Microsoft Windows Mobile. V dokumentaci jsou tato zařízení nazývána *Přenosný datový terminál (PDT)*. Může to být například Motorola MC55.

3.1 Funkcionalita na straně systému SAP

Systém FAIN v ERP SAP používá řadu transakcí začínajících písmeny ZFA, transakce jsou seskupeny do menu ZFA1, které je na obr. 3. Klíčovými objekty inventury jsou Inventura, Inventurní seznam, Položky inventurních seznamů – jednotlivé majetky.



Obr. 3 Menu ZFA1
Zdroj: Uživatelský manuál FAIN 4.1

3.1.1 Tisk etiket

Tisk etikety majetku – transakce ZFATLA. Po nastavení filtrů se zobrazí seznam, ve kterém je potřeba označit majetky, pro které chceme tisknout etikety. Po kliknutí na tlačítko *Tisk etiket majetku* se zobrazí dialog pro výběr tiskárny, na které má tisk proběhnout. Informace o tom, zda etiketa majetku pro daný majetek je již vytištěna, se uloží na kartě majetku. Etiketa majetku obsahuje v čitelné formě následující údaje:

- Popis majetku – přibližně 35 prvních znaků z prvního řádku popisu
- Číslo majetku v SAP
- Podčíslo majetku v SAP

Tisk etikety místnosti – transakce ZFATLM. Postup je obdobný, jako u tisku etikety majetku. Etikety majetků i etikety místností používají 2-D čárový kód standardu PDF417. Etiketa místnosti obsahuje v čitelné formě následující údaje:

- Popis místnosti – přibližně prvních 35 znaků
- Identifikace závodu
- Identifikace budovy
- Identifikace místnosti

Tisk RFID etiket – transakce ZFATLAX pro majetek a ZFATLMX pro místnost. Proces tisku RFID etiket je stejný jako tisk etiket s čárovým kódem, stejný je i vzhled výsledných etiket. RFID etiketa ale obsahuje vedle čárového kódu navíc RFID tag, do kterého tiskárna zapíše ty stejné údaje, které se nacházejí v čárovém kódu. Systém FAIN používá RFID tagy typu UHF 96 bit EPC.

3.1.2 Inventura

V menu Inventura je k dispozici skupina transakcí, které umožňují s inventurou pracovat. Pro každou inventuru v systému existuje záznam o inventuře, inventurní seznamy s inventurními položkami a dostupnými logy. Každá inventura prochází následujícími stavy:

- *Aktivní* – stav po založení nové inventury a během vykonávání inventury
- *Zrušená* – označení neplatné inventury
- *Uzavřená* – inventura po uzavření

V inventuře je možné měnit data, pouze pokud se nachází ve stavu *Aktivní*, v ostatních stavech je možné data pouze číst.

Založení nové inventury – transakce ZFAI01. K provedení fyzické inventarizace pomocí čárových kódů a RFID je zapotřebí založit v SAP inventuru a inventurní seznam. Pro vytvoření inventury musí být vyplněná následující povinná pole:

- *Účetní okruh* – pro který bude inventura vykonána
- *K datu (Datum inventury)* – datum, ke kterému je inventura vykonána
- *Popis inventury* – název inventury na VUT se skládá ze zkratky součásti, čísla součásti v závorkách a roku, příklad: FAVU (11) 2013.
- *Platí od* – datum, od kterého budou platné změny zjištěné během inventury a zapsané do položek časově závislých dat v Kartě majetku.

Po vytvoření inventury se v informačním okně zobrazí systémem vygenerované číslo inventury. Toto číslo inventury se využívá ve všech následujících transakcích systému FAIN.

Zobrazení inventury – transakce ZFAI03. Slouží jen k zobrazení údajů o inventuře. K identifikaci inventury slouží číslo vygenerované při založení nové inventury. Kromě dat vložených během vytváření inventury se zobrazuje také seznam vytvořených Inventurních seznamů a logy důležitých operací. Příklad zobrazení inventury je na obr. 4.

Zobrazení inventury majetku 00000065

Uzavřena

Číslo inventury 65

K datumu 31.08.2012

Pro účetní okruh VVS

Platí od 15.11.2012

Popis FAST (12) 2012

Ř. 1, sl. 1 Ř. 1 - ř. 1 z 1 řádek

Hlavní inventarizační komise

Pozice	Příjmění	Krstní jméno	Funk.

Inventurní seznamy

Inve...	in...	Účok	tiz	st	stpr	dt.zal	čas založení	založil	dt.zm.	čas změny	změnil	
65	1		FREE	A		22.09.2012	15:37:20	BASO		00:00:00		00:00:00

Logy

Inve...	in...	Účok	p...	T-kód	Obraz.	Funkční kód	Uživat.
65			7	Zápisy zn.inv, počet pol.:35807 ,z toho chyb...	1000	WR_INVAM	BASO
65			6	Inv. uzavřena.	ZFAI08	0200	CLOSE
65			5	Zápisy do AM, počet pol.:0 ,z toho chyb:5 .	ZFAI08	0120	WR_AM
65			4	Zápisy do AM, počet pol.:0 ,z toho chyb:5 .	ZFAI08	0120	WR_AM
65			3	Zápisy do AM, počet pol.:16 ,z toho chyb:5 .	1000	WR_AM	BASO
65	1		19	35791 položek ve stavu O bylo odsouhlasen...	ZFAR02	0120	CONF_O
65	1		20	35791 položek ve stavu O bylo odsouhlasen...	ZFAR02	0120	CONF_O

Obr. 4 Zobrazení inventury FAST (12) 2012 pomocí transakce ZFAI03

Změna inventury – transakce ZFAI02. Slouží pro změnu nebo doplnění dat existující inventury. Obrazovka Změna inventury vypadá stejně jako obrazovka Zobrazení inventury, ale pole je možné editovat. Všechny provedené změny jsou logované. Tato transakce slouží také ke zrušení inventury. Zrušenou inventuru není možné měnit, lze jen prohlížet.

Uzavření inventury – transakce ZFAI08. Uzavření inventury je chápáno jako potvrzení ukončení inventurního procesu a po této akci je celá inventura dostupná už jen k prohlížení. Uzavření inventury provádí administrátor na základě písemného

pokynu. Aby bylo možné inventuru uzavřít, musí být splněny následující podmínky:

- Žádné položky ve stavu I (iniciální) se nemohou vyskytovat v žádném inventurním seznamu patřícím k této inventuře
- Všechny položky majetku v každém inventurním seznamu musí být odsouhlaseny
- Žádná položka nesmí být ve stavu konfliktu
- Všechny položky se změněným atributem (stav X) musí mít změny zapsané do karty majetku

Proces uzavření inventury provede následující 3 akce:

- Zapsání změn časově závislých dat do karet majetku FI-AA
- Nastavení stavu inventury na *Uzavřená*.
- Aktualizace data poslední inventury u všech majetků zahrnutých v inventuře (FAIN, 2011)

3.1.3 Inventurní seznamy

Jednotlivé majetky (inventurní položky) jsou seskupené do logických celků – inventurních seznamů. Systém FAIN umožňuje, aby jedna inventura mohla obsahovat více inventurních seznamů. Ale na VUT v Brně se pro každou založenou inventuru používá vždy jen jeden inventurní seznam. Pro každý pracovní úsek je zpravidla založena jedna inventura, ve které je vygenerovaný seznam majetku. (VUT_AM, 2014) Položky jsou v inventuře identifikované hlavním číslem majetku a podčíslem majetku, které je podobné jako v modulu FI-AA v SAP. Inventurní položky mohou mít následující stavy:

- **I** – initial (počáteční) – stav po vytvoření inventurního seznamu.
- **M** – missing (chybějící) – majetky označené jako chybějící během inventury.
- **O** – OK – nalezené majetky bez jakýchkoliv změn v porovnání s kartou majetku v FI-AA.
- **X** – změna – nalezené majetky se změněným atributem (jeden z časově závislých atributů), typicky je to změna lokality (závod, budova nebo místnost).
- **N** – nový – majetky nalezené v inventuře jako nové, s existující kartou majetku v FI_AA. Tento status se objeví, kdy byla oskenovaná nějaká etiketa majetku, která nebyla zahrnuta v seznamu exportovaném do PDT.
- **B** – nový bez čárového kódu – majetky nalezené jako nové, které nejsou označené etiketou a proto nemohly být identifikované na místě jako některé z položek majetku v exportovaném seznamu a nemohly být potvrzené manuálně.
- **V** – deaktivovaný – majetek nasnímaný během inventury (jako nový majetek na PDT) a potom importovaný do systému SAP, kde k tomuto číslu majetku

existuje karta majetku. Majetek je deaktivovaný, pokud pole data deaktivace je vyplněné v kartě majetku. Deaktivované položky mohou být z inventurního seznamu v SAP vymazané. (FAIN, 2011)

Pro každou inventarizovanou položku uloženou v systému SAP se kromě informace o stavu uchovává také informace o způsobu potvrzení. Pokud je inventarizovaná položka potvrzena přečtením čárového kódu z etikety majetku, je hodnota tohoto atributu (*man*) nastavena na „ “ (mezera). V případě, že položka byla potvrzena manuálně na PDT, hodnota je nastavená na „P“. V případě, že položka byla potvrzena manuálně v systému SAP, hodnota je nastavená na „S“. Atribut určující metodu potvrzení nemá žádný vliv na stav inventurní položky. Pouze udává, jak bylo potvrzení vykonáno. (FAIN, 2011)

Nový inventurní seznam – transakce ZFAR01. Transakce se používá k vytvoření inventurního seznamu, který je zakládán společně s vytvořením inventury administrátorem. Výběr majetků může být omezený pomocí filtrů na výběrové obrazovce, na VUT v Brně se pro každý pracovní úsek zakládá jedna inventura, ve které je vygenerovaný seznam majetku podle těchto kritérií:

- Pracovní úsek
- Vyloučení majetkových tříd 1005, 1006, 1019 a 4101 až 4208
- Pouze majetek s podčíslem 0 (výjimky: z inventurního seznamu jsou vyloučena podčísla majetku, pokud na nich není žádná hodnota). (VUT_AM, 2014)

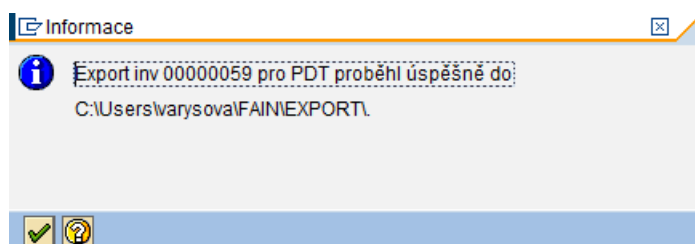
Po vytvoření nového inventurního seznamu mají všechny jeho položky status I – Initial (Počáteční).

3.1.4 Exporty/importy dat

Transakce v tomto menu slouží k přenosu dat mezi systémem SAP a aplikací v PDT terminálu. Aby bylo možné tyto operace provádět, musí PC, na němž spouštíme transakce v SAP GUI, komunikovat s Windows Mobile PDT terminálem. PDT musí být připojeno k PC USB kabelem. Musí být nainstalovány správné ovladače a komunikační software.

Export inventurního seznamu do PDT – transakce ZFAEXP1. Tato transakce slouží pro přenos inventurních položek do PDT terminálu spolu s číselníky potřebnými pro aplikaci na PDT. Do PDT může být exportován buď celý inventurní seznam, nebo jen některé položky. PDT může obsahovat vždy jen jeden aktivní seznam. Před exportem jiného inventurního seznamu (nebo jeho části) do PDT, předcházející exportovaný seznam je potřeba nejprve importovat do SAP. (FAIN, 2011)

Po výběru inventury, případně dalších filtrů, a úspěšném exportu v SAP GUI viz obr. 5.

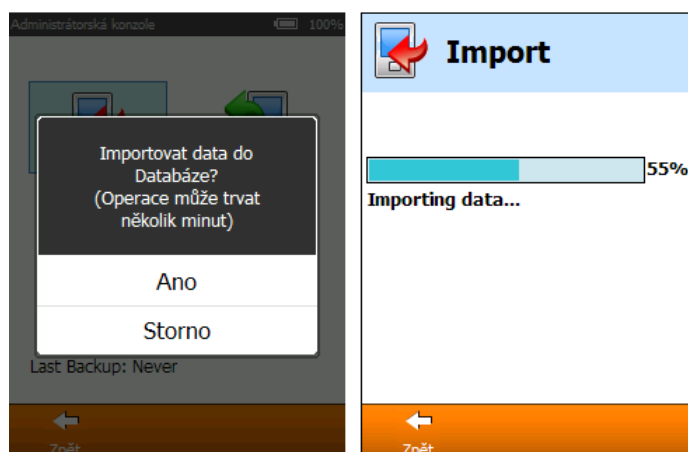


Obr. 5 Úspěšný export v SAP GUI
Zdroj: VUT_AM



Obr. 6 Export dat z PDT
Zdroj: VUT_AM

Je nutné se přesunout k PDT terminálu, po přihlášení pod uživatelem *Admin* se zobrazí obrazovka pro exporty a importy dat. Nejdříve je potřeba zadat „Export“ dat a volbu potvrdit, dále je nutné zadat import dat a volbu potvrdit. Postup je znázorněn na obr. 6 a obr. 7.

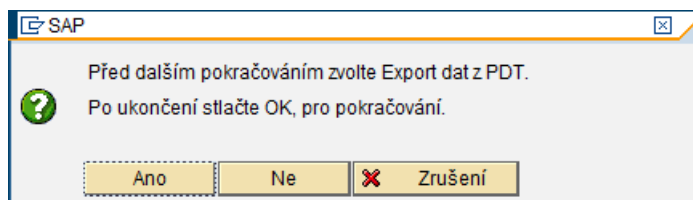


Obr. 7 Import dat z PDT
Zdroj: VUT_AM

Po exportování inventurních dat ze SAP je PDT připravený na použití. Nyní může osoba vykonávající inventuru používat PDT na snímání čárových kódů majetků.

Import inventurního seznamu z PDT – transakce ZFAIMP1. Tato transakce se používá k přenosu dat z PDT terminálu do systému SAP po nasnímání čárových kódů majetků.

Po spuštění transakce se zadá číslo inventury a číslo inventurního seznamu, po potvrzení se zobrazí dialogové okno jako na obr. 8.



Obr. 8 SAP GUI očekává data z PDT
Zdroj: VUT_AM

Dialog uživateli říká, že se má na PDT terminálu přihlásit do obrazovky Admin pro exporty a importy dat a zde provést export (import dat do SAP = export dat z PDT terminálu). Postup na PDT je zobrazen na obr. 6. V zobrazeném dialogu v SAP GUI uživatel zatím nezadá žádnou volbu.

Až po úspěšném exportu dat z PDT terminálu se v zobrazeném dialogu v SAP GUI klikne na volbu „Ano“, SAP GUI převezme data z PDT a zobrazí seznam přenesených dat a statistiku inventury. (VUT_AM, 2014)

Zobrazení importu z PDT – transakce ZFAIMP1VIEW. Tato transakce je určena k zobrazení dat importů z PDT. Všechny seznamy importované z PDT do SAP jsou logované a mohou být zobrazeny později použitím této transakce.

3.1.5 Zobrazení inventurního seznamu

Po několika importech dat z PDT, které představují určitou část zpracování inventurního seznamu je potřeba mít přehled o stavu všech položek inventurního seznamu. Slouží k tomu transakce ZFAR03.

Po vyplnění čísla inventury a čísla inventurního seznamu transakce zobrazí všechny položky patřící k zadanému inventurnímu seznamu. Podle potřeby je možné nastavovat filtry. Příklad zobrazení seznamu je na obr. 9.

FAIN zobrazení položek inv.seznamu								
Inv.maj.	Nákl.stř.	Průs	Sériové číslo	Nákl.zakáz.	SKP	Označení IM	Datum akt.	Množství
10454	12320	12		1100	722020	SW Educational	22.10.2009	1,000
10456	12400	12	321-61	1100	722020	SW TWINS	14.10.2009	1,000
10459	12400	12		1100	722021	SW SoundPlan	25.11.2009	1,000
10460	12400	12		1100	722021	SW DEWESOFT	25.11.2009	1,000
10475	12400	12		1100	722020	SW Support Package	25.08.2010	1,000
10484	12390	12		1100	722020	SW ZWCAD	23.12.2010	1,000
10485	12400	12		1100	722020	SW Visual ODS User	20.12.2010	1,000
10485	12400	12		1100	722020	SW Visual ODS User	20.12.2010	1,000
10496	12431	12		1100	722020	SW program FLOW-3D	22.08.2011	1,000
10497	12431	12		1100	722020	SW Autodesk Education Master Suite 2012	07.11.2011	1,000
10504	12250	12		1100	722020	SW multiPlas University Research lic.	15.12.2011	1,000
10525	12330	12	SW 10724905	1100	722020	SW FLIR Reporter 9.0 Pro	30.08.2012	1,000
10526	12310	12		1100	722020	SW RHEL Red Hat Enterprise	29.08.2012	1,000
301394	12210	12	G7303279	1100	334035	Dataprojektor SANYO	04.11.1997	1,000
301397	12210	12	99J3477B	1100	334035	Datavideoprojektor BENQ	31.10.2003	1,000
301398	12750	12	7J31KT43Y01Z	1100	300212	Počítač ProLiant	28.11.2003	1,000
301399	12210	12	DA 251812/0903	1100	300215	Počítač OfficePro + monitor 18" LCD + monitor 19"	30.11.2003	1,000
301400	12210	12	DA 251813/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301401	12210	12	DA 251 814/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301404	12210	12	DA 251817/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301405	12210	12	DA 251 818/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301406	12210	12	DA 251 819/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301407	12210	12	DA 251820/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301408	12210	12	DA 251821/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301409	12210	12	DA 251822/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301410	12210	12	DA 251823/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301411	12210	12	DA 251824/0903	1100	300215	Počítač OfficePro + monitor 18" LCD	30.11.2003	1,000
301412	12210	12	GNF 6122801	1100	300214	Tiskárna HP LJ	31.12.2003	1,000
301418	12210	12	217739 916	1100	300121	Kopírovací stroj	09.09.1998	1,000
301442	12210	12	DA 177123/0801	1100	300212	Počítač Integra	30.10.2001	1,000
301444	12210	12	H9D033P000191	1100	300214	Scanner UMAX Astra PowerLook	31.12.2003	1,000
301448	12210	12	ZAPEČETĚNO	1100	300217	Soubor síťových prvků	31.12.2001	1,000
301449	12700	12		1100	323033	Kamerový digitální systém	31.12.2001	1,000

Obr. 9 Zobrazení inventurního seznamu

Pro lepší čitelnost seznamu je podkladová barva každého řádku odvozena od stavu položky. Popis použitých barev a jejich významu je na obr. 10.

LEGENDA BAREV:
I - inicialní
M - chybějící
O - nalezen beze změn
X - nalezen, změna atributu
X - nalezen, nepřípustná změna atr.
N - navíc z čár.kódem => existuje karta AM ?
B - navíc bez čár.kódu => bude zařazen ?
V - vyřazen
konflikt po vícenásobném importu

Obr. 10 Legenda barev pro inventurní seznam

Na VUT v Brně se využívají pouze první čtyři stavy. (VUT_AM, 2014)

3.1.6 Zpracování inventurního seznamu

Transakce ZFAR02. Zobrazení inventurního seznamu vypadá stejně, jako na obr. 9. Ale v tomto případě je možné provádět operace:

Status zpracování – nebo také stav položky se mění pomocí PDT, nebo manuálně v SAP GUI pomocí této transakce. Stav jednotlivých položek může být změněn manuálně ze stavu „I“ Iniciální na stavy:

- O – OK (nalezeno)
- X – nalezeno se změnou atributu
- M – chybějící majetek

Změny stavu je možné provádět i hromadně. Po uložení změn v systému SAP bude nastavený atribut manuálního potvrzení v SAP, indikující, že tato změna byla provedena manuálně v systému SAP a ne přímo na PDT. Tento údaj může být viditelný v seznamu inventurních položek. (FAIN, 2011)

Editování poznámek – možnost vybrat nebo změnit předvolenou poznámku nebo změnit volnou poznámku vybrané inventurní položky. Předvolené poznámky jsou předem definované v systému SAP. Volná poznámka je libovolný text, který může být přidělený každé položce inventurního seznamu. Obě dvě poznámky (předvolená i volná) mohou být zadány na PDT nebo v systému SAP.

Odsouhlasení – stavy všech položek inventurních seznamů musí být před uzavřením inventury odsouhlasené. Odsouhlasení položek znamená, že někdo zkontroloval správnost těchto stavů. Stavy majetku jsou řešeny na VUT následovně:

- Pokud se v inventurním seznamu zobrazí majetek podbarvený světle zelenou barvou, je vše v pořádku, tento stav se nemusí nijak řešit.
- Pokud je majetek podbarven světle žlutě, jde o změnu umístění majetku a je nutné posoudit, zda jsou dané změny žádoucí. Jestliže změny nejsou žádoucí, je potřeba je opravit. Jsou-li zachycené změny žádoucí, je vhodné je vyfiltrovat, převést do Excelu a odeslat ke schválení odpovědným osobám.
- Pokud je majetek podbarven tmavě žlutě, jde o konfliktní stav. Tento stav je nežádoucí a nesmí se vyskytovat před uzavřením inventury. Před uzavřením inventury je nutné vyřešit tento stav individuálně s hlavním majetkářem. (VUT_AM, 2014)

3.1.7 Řešení konfliktů

Transakce ZFAICONFL. Slouží k identifikaci a řešení konfliktů mezi inventurními seznamy v rámci jedné inventury. Vzhledem k tomu, že se na VUT v Brně vždy vytváří pouze jeden inventurní seznam pro jednu inventuru, nemá tato transakce využití.

3.2 Funkcionalita mobilního terminálu PDT

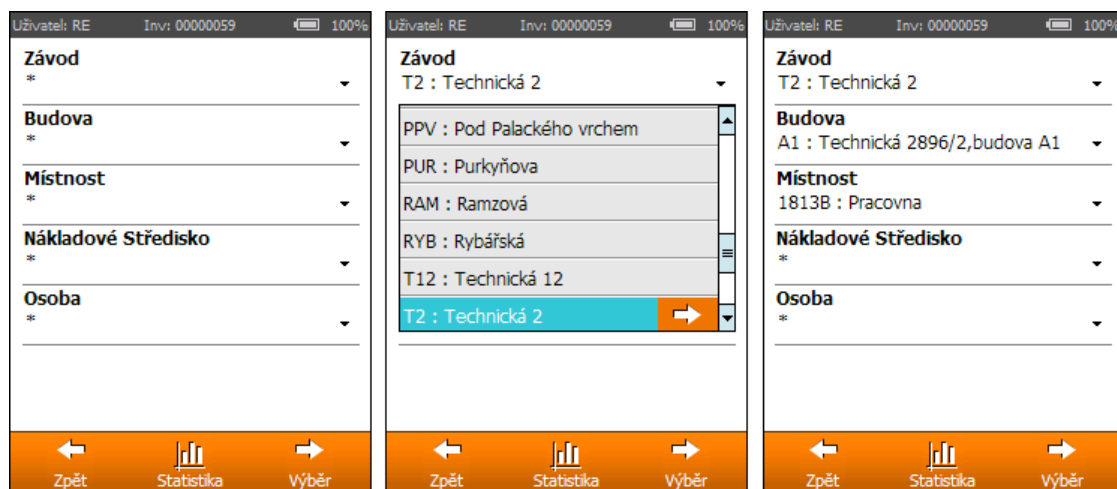
Mobilní terminál PDT je univerzální zařízení s operačním systémem Microsoft Windows Mobile, pro provádění inventury pomocí systému FAIN je potřeba spustit aplikaci *Inventory Management*. Po provedení exportu, importu dat, popsáném v sekci 3.1.4 je terminál připraven ke snímání čárových kódů majetků.

3.2.1 Přihlášení

V aplikaci *Inventory Management* je nutné nejprve provést přihlášení. Pro přihlášení slouží jméno a heslo přiřazené hlavním majetkářem pracovního úseku. Administrace jmen a hesel je prováděna Odborem ekonomických systémů. (VUT_AM, 2014)

3.2.2 Nastavení filtrů

Pokud je ve filtru nastavena hvězdička, není pro tuto položku vytvořeno žádné filtrování. Aby bylo možné snímat čárové kódy majetku, musí být vybrán konkrétní Závod, Budova a Místnost (povinná pole pro spuštění scanneru). Další filtry (Nákladové středisko a Odpovědná osoba) jsou volitelné. Pokud nejsou nastaveny všechny povinné filtry, je seznam majetku otevřen pouze pro prohlížení, není možné zpracovávat majetky. Nastavení filtrů je vidět na obr. 11.

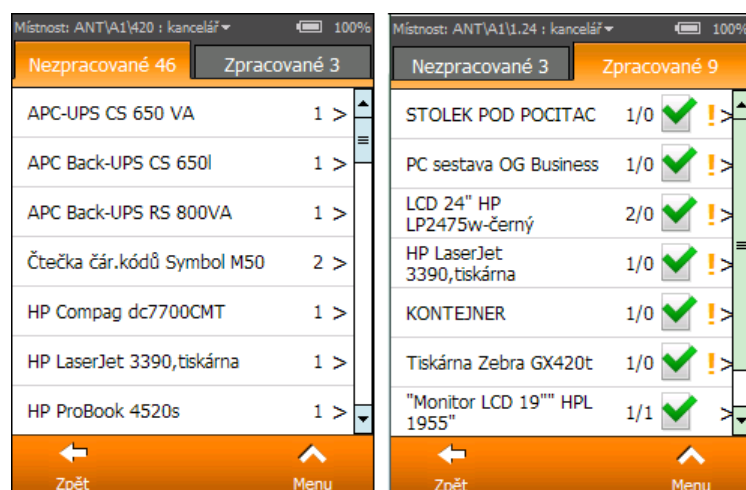


Obr. 11 Nastavení filtrů na PDT

Zdroj: VUT_AM

3.2.3 Zpracování majetku

Po nastavení povinných filtrů a kliknutí na tlačítko Výběr se zobrazí seznam majetku. Nyní je možné zpracovávat majetky buď fyzickým nasnímáním čárového kódu, nebo manuálním potvrzením. Jak můžeme vidět na obr. 12, seznam majetků je rozdělen do dvou skupin – Nezpracované a Zpracované.







Obr. 12 Seznam majetku v místnosti na PDT

Zdroj: Prezentace školení členů inventurní komise VUT Brno

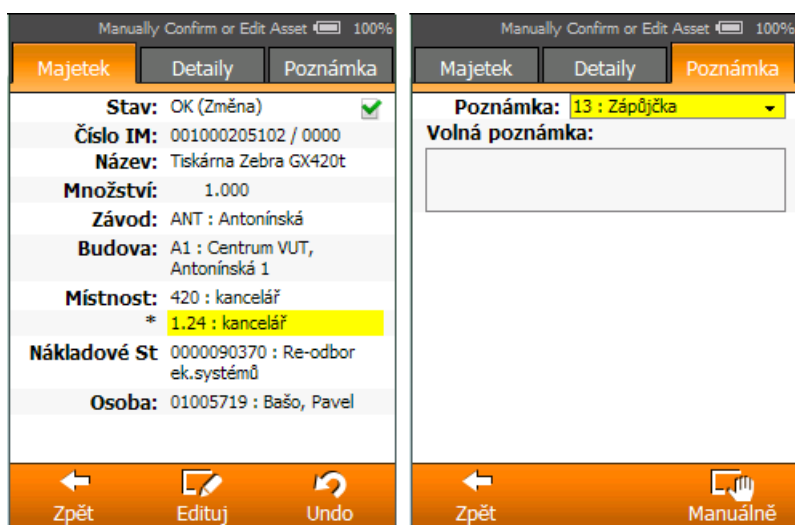
Ve skupině zpracovaných majetků jsou jednotlivé majetky označeny ikonami. První ikona indikuje, jakým způsobem byl majetek potvrzen. Druhá ikona ukazuje stav majetku, vzhled a význam ikon je popsán v tab. 3.

Tab. 3 Vzhled a význam ikon v seznamu majetku na PDT

Význam	Ikona
Potvrzení automaticky	
Potvrzení manuálně	
Změna	
Nový majetek	

Zdroj: VUT_AM

Kliknutím na majetek v seznamu se zobrazí Detail majetku, na této obrazovce je možné zobrazit všechny atributy majetku. Jak vypadá detail majetku na PDT je možné vidět na obr. 13.



Obr. 13 Detail majetku na PDT, změna místnosti, poznámka
Zdroj: Prezentace školení členů inventurní komise VUT Brno

3.2.4 Manuální potvrzení majetku

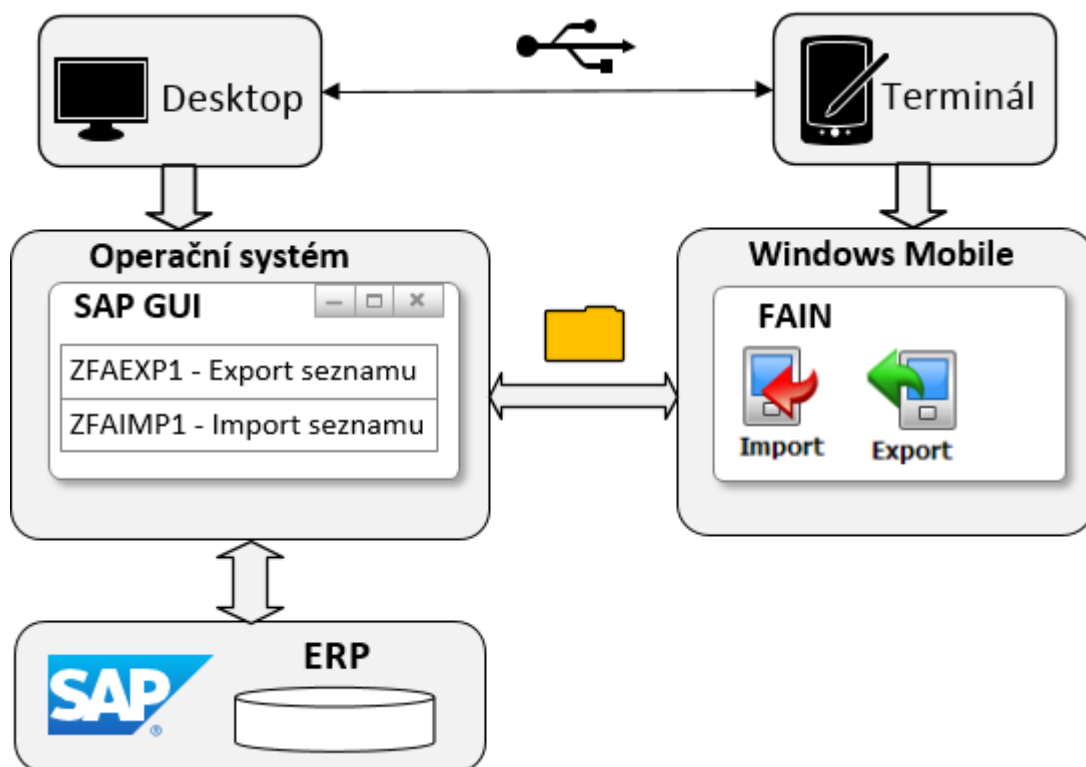
Pokud majetek nelze označit pomocí čárového kódu, je ve výpůjčce a podobně, ale jeho identita může být prokázána, uživatel ho může potvrdit manuálně. Uživatel musí kliknout na detail majetku v seznamu nezpracovaného majetku, vybere manuální potvrzení majetku (volba „Manuálně“), vybere fixní poznámku (důvod, proč je majetek potvrzován manuálně) a opět klikne na volbu „Manuálně“. Pro pevné poznámky jsou možné pouze následující varianty:

- 11 – Chybějící RFID nebo čárový kód
- 12 – Vadný RFID nebo čárový kód
- 13 – Majetek v opravě – doložen servisní list
- 14 – Pronajatý majetek – doložena nájemní smlouva

Jak vypadá manuální potvrzení majetku, můžeme vidět na obr. 13.

3.3 Nevýhody systému

Hlavním problémem systému je, že synchronizace mezi terminálem a SAP probíhá dávkově pomocí souborů, jak je popsáno v sekci 3.1.4 a znázorněno na obr. 14.



Obr. 14 Synchronizace terminálu a ERP SAP přes PC

Terminál PDT tedy pracuje v offline módu, nese si určitý export majetkových dat ze SAP ve své lokální SQL databázi. Je potřeba si uvědomit, že inventura na VUT je velmi rozsáhlá akce, počty položek inventury roku 2012 některých fakult VUT v Brně jsou v tab. 4.

Tab. 4 Počet položek inventury roku 2012 některých fakult VUT Brno

Fakulta	Počet položek inventury roku 2012
FAVU	2 901
FAST	35 812
FSI	41 638
FIT	14 632
FA	3 975
FCH	9 911
FEKT	40 387

Na inventuře pracuje několik inventurních komisí pro fakultu cca 10 – 20. Snadno se může stát, že některé majetky jsou zbytečně snímány několikrát různými členy inventurní komise, kteří pracují s terminálem PDT.

Problematický je i samotný import a export dat popsany v sekci 3.1.4. V manuálu VUT_AM (2014) a také v prezentaci pro školení inventurní komise

(2014) je zvláště uvedeno, že import inventurního seznamu z PDT se využívá pro pravidelné zálohování inventurních dat. Zálohy je doporučeno provádět pravidelně, buď na konci každého dne inventury či po nasnímání většího počtu majetku. *Pokud nebudou provedeny zálohy dat, pak v případě poškození datového souboru, nebude možné poškozená data opravit a celá inventura bude muset být provedena znovu.*

Terminál PDT je speciální HW zařízení s vysokými pořizovacími náklady (jeden terminál cca 50 000,-). Terminály používají dnes již mrtvou platformu Microsoft Windows Mobile a mimo inventuru nemají na VUT v Brně žádné další využití. Používání jedenkrát za rok také zkracuje životnost akumulátoru terminálu.

Windows Mobile aplikace pro PDT terminál je proprietární. Přestože VUT disponuje prostředky na to, aby si mohlo změny v aplikaci provádět samo, není to možné, zdrojové kódy jsou majetkem firmy Oriwin s. r. o. a za každou i malou změnu se musí platit.

3.4 Co se děje na pozadí

Na obr. 15 je seznam a popis tabulek, které byly přidány do ERP SAP implementací systému ZFAIN, systém vedle těchto tabulek využívá i stávající tabulky databáze SAP.

Název tabulky	Krátký popis
ZFAIN	FAIN položky inventur
ZFAIN_ASSETNAMES	Názvy majetkov z importu bez c.k.
ZFAIN_CUST	FAIN - nastavení systému FAIN
ZFAIN_EXPCUST	
ZFAIN_FILESTORE	Úložisko souborů
ZFAIN_FILESTOREH	FAIN hlavička souboru
ZFAINH	FAIN neaktivní položky inventur
ZFAIN_CH_CUST	parametre povolené na změnu
ZFAIN_IMP	Importy
ZFAIN_IMP_A	
ZFAIN_IMPDATA	importovane položky z PDT
ZFAIN_IMPDATA_A	
ZFAIN_IMP_FILT	FAIN položky filtrované při importe
ZFAIN_IMPNOTES	FAIN poznámky k inventure
ZFAIN_IMPNOTES_A	
ZFAIN_INV	FAIN hlavičky inventur
ZFAIN_INVCOMM	
ZFAIN_INVLOG1	FAIN log akcii inventur a zoznamov inv.
ZFAIN_INVNOTES	FAIN poznámky k inventure
ZFAIN_INZ	FAIN hlavičky inventurních seznamů
ZFAIN_IZTYP	FAIN typy
ZFAIN_LOG1	FAIN - log1
ZFAIN_MSGS	FAIN - pevné typy poznámok
ZFAIN_NOTES	FAIN poznámky k inventure
ZFAIN_PRINTERS_C	Zoznam tiskáren pro hromadnou tlač
ZFAIN_PRNALOG	Log tlace etikiet majetkov
ZFAIN_PRRLOG	
ZFAIN_USRPDA	FAIN - PDT users

Obr. 15 Tabulky systému ZFAIN

Při založení nové inventury, popsáno v sekci 3.1.2, vznikne nový záznam v tabulce ZFAIN_INV jako hlavička inventury. ERP SAP má všechny majetky uloženy v tabul-

ce ANLA, časově závislá data majetků jsou uchovávána v tabulce ANLZ. Během zakládání nového inventurního seznamu, popsáno v sekci 3.1.3, transakce podle nastavených parametrů inventury a inventurního seznamu provede spojení tabulek ANLA a ANLZ a výsledek uloží do tabulky ZFAIN. V tabulce ZFAIN mají všechny položky inventury zpočátku v příslušných sloupcích zaznamenán stav I (iniciální) a umístění majetku, které bylo aktuální při uzavření předchozí inventury. Následně probíhá fyzická inventura pomocí terminálů PDT, popsáno v sekci 3.2, aby bylo možné terminály PDT používat, musí se provádět import a export dat, popsáno v sekci 3.1.4. Při přenosu dat do terminálu PDT ze SAP se položky inventury kopírují z tabulky ZFAIN. Při přenosu dat z terminálu PDT do SAP se data ukládají do tabulek ZFAIN_IMP a ZFAIN_IMPDATA. Změny stavu a umístění inventurních položek získané nasnímáním pomocí terminálu se také promítnou do tabulky ZFAIN. Pokud dojde k vícenásobnému nasnímání jednoho majetku, může k tomu dojít omylem, nebo pokud je během inventury majetek přenesen na jiné místo a tam znovu nasnímán, platí jednoduché pravidlo: Poslední nasnímání je platné. Historie se loguje do tabulky ZFAINH. Transakce pro zobrazení a zpracování inventurního seznamu, popsáno v sekcích 3.1.5 a 3.1.6, používají jako kmenová data také tabulku ZFAIN.

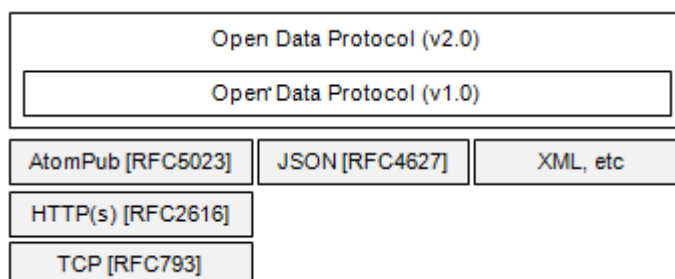
Pokud by aplikace na terminálu přistupovala přímo do tabulky ZFAIN v reálném čase, odpadly by operace importování, exportování a zálohování dat, také by se zamezilo riziku, že dojde k poškození těchto dat a celá inventura bude muset být provedena znovu. Cílem této práce je takovou aplikaci vytvořit. Aplikace kompletně nahradí stávající aplikaci na terminálu PDT, bude postavená na technologii SAPUI5 a bude přistupovat přímo do tabulky ZFAIN prostřednictvím OData služby. Systém pro inventuru na straně SAP zůstane nezměněn, jen transakce pro export a import dat, popsané v sekci 3.1.4, nebudou potřeba.

4 OData služba

Protokol OData (Open Data Protocol) umožňuje vytváření datových služeb založených na REST (representational state transfer), které umožňují webovým klientům provádět CRUDQ operace nad zdroji identifikovanými pomocí Uniform Resource Identifier (URI) a definovanými v datovém modelu použitím jednoduchých HTTP zpráv.

Protokol původně vyvinul Microsoft, verze 1.0, 2.0 a 3.0 jsou uvolněny pod Microsoft Open Specification Promise. Aktuálně nejnovější verze 4.0 byla schválena, jako standard prostřednictvím OASIS OData Technical Committee, jejímiž členy jsou BlackBerry, IBM, Microsoft, Red Hat, SAP a další. (OASIS, 2014) Následující popis se vztahuje na verzi 2.0, která bude využívána v aplikacích této práce.

Data přenášená prostřednictvím OData protokolu mohou využívat různé datové formáty běžně používané ve webových technologiích, například Extensible Markup Language (XML), JavaScript Object Notation (JSON), Atom Publishing Protocol (AtomPub) a další. V další úrovni jsou data zabalena do protokolu HTTP, případně do jeho zabezpečené verze HTTPS. Na transportní vrstvě jsou data zabalena do protokolu TCP. Schéma zapouzdření dat je znázorněno na obr. 16.



Obr. 16 Vztah oData k ostatním protokolům včetně čísel jejich RFC dokumentů
Zdroj: <http://odata.org>

Jádrem OData jsou *feeds*, které jsou kolekcemi *Collections* složené ze záznamů *Entries*. Každý *Entry* je identifikovaný klíčem a reprezentuje strukturovaný záznam, který má seznam vlastností *Properties*, ty mohou být komplexního, nebo primitivního typu. *Entries* mohou být součástí hierarchie typů a mohou mít související *entries* a *feeds* odkazované prostřednictvím *Links*.

```

▼<feed xmlns="http://www.w3.org/2005/Atom" xmlns:m="http://schemas.microsoft.com/ado/200
  ▼<id>
    http://alpaca.ro.vutbr.cz:50000/ZLUBOS_INV_SRV/ZfainInv(Inv='80')/Plants
  </id>
  <title type="text">Plants</title>
  <updated>2015-11-27T10:14:13Z</updated>
  ▶<author>...</author>
  <link href="ZfainInv(Inv='80')/Plants" rel="self" title="Plants"/>
  ▼<entry>
    ▼<id>
      http://alpaca.ro.vutbr.cz:50000/ZLUBOS_INV_SRV/Plants(Inv='00000080',Werks='ANT')
    </id>
    <title type="text">Plants(Inv='00000080',Werks='ANT')</title>
    <updated>2015-11-27T10:14:13Z</updated>
    <category term="ZLUBOS_INV_SRV.Plant" scheme="http://schemas.microsoft.com/ado/2007
    <link href="Plants(Inv='00000080',Werks='ANT')" rel="edit" title="Plant"/>
    <link href="Plants(Inv='00000080',Werks='ANT')/Buildings" rel="http://schemas.micro
    <link href="Plants(Inv='00000080',Werks='ANT')/$links/Buildings" rel="http://schema
    ▼<content type="application/xml">
      ▼<m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/meta
        <d:Inv>00000080</d:Inv>
        <d:Werks>ANT</d:Werks>
        <d:Name1>Antonínská</d:Name1>
      </m:properties>
    </content>
  </entry>
  ▼<entry>
    ▼<id>
      http://alpaca.ro.vutbr.cz:50000/ZLUBOS_INV_SRV/Plants(Inv='00000080',Werks='BOZ')
    </id>
    <title type="text">Plants(Inv='00000080',Werks='BOZ')</title>
    <updated>2015-11-27T10:14:13Z</updated>
    <category term="ZLUBOS_INV_SRV.Plant" scheme="http://schemas.microsoft.com/ado/2007
    <link href="Plants(Inv='00000080',Werks='BOZ')" rel="edit" title="Plant"/>
    <link href="Plants(Inv='00000080',Werks='BOZ')/Buildings" rel="http://schemas.micro
    <link href="Plants(Inv='00000080',Werks='BOZ')/$links/Buildings" rel="http://schema
    ▼<content type="application/xml">
      ▼<m:properties xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/meta
        <d:Inv>00000080</d:Inv>
        <d:Werks>BOZ</d:Werks>
        <d:Name1>Božetěchova</d:Name1>
      </m:properties>
    </content>
  </entry>

```

Obr. 17 oData feed obsahující kolekci entit závodů inventury s klíčem 80

Na obr. 17 vidíme výpis feedu ve webovém prohlížeči. Feed je identifikován plnohodnotným URI, můžeme z něj vyčíst, že jsme se na aktuální feed se seznamem závodů dostali použitím odkazu ve feedu se seznamem inventur *ZfainInv*. Inventuru jsme jednoznačně identifikovali pomocí klíče *Inv='80'* a použili link *Plants* a tím jsme získali aktuální feed se seznamem závodů inventury 80.

Kolekce feedu se skládá ze záznamů *entry*, každý *entry* tak představuje jeden *Plant* závod obsažený v inventuře 80. Na začátku *entry* je plnohodnotné URI, pokud jej zadáme do prohlížeče, oData služba vrátí pouze tuto konkrétní *entry*. Dále je specifikován klíč k jednoznačné identifikaci *entry*, můžeme si všimnout, že

ke klíči *Inv* přibyl klíč *Werks*. Dále je uveden seznam link odkazů, naše entita závodu obsahuje odkaz *Buildings*. Takže po identifikaci konkrétního závodu pomocí klíčů *Inv* a *Werks* je možné použít odkaz *Buildings* a oData služba vrátí feed obsahující kolekci budov pro tento závod. Následuje seznam properties (vlastností), vedle klíčů obsahuje entry ještě vlastnost *Name1*, která představuje název závodu.

4.1 Metadata

OData služba poskytuje metadata dokumenty. Aby mohli uživatelé oData služby prozkoumat, co všechno služba nabízí, aniž by museli zkoumat implementaci služby na backendu.

Jednodušší *Service Document* se nachází v root URI služby, obsahuje seznam všech feedů, takže je uživatelé mohou prozkoumat a zjistit jejich adresy.

Přidáním segmentu *\$metadata* do root URI služby získáme *Service Metadata Document*, který popisuje celý datový model, jinými slovy strukturu a propojení všech zdrojů. Jak uvádí (Odata, 2013), *Service Metadata Document* popisuje svá data pomocí termínů EDM použitím XML jazyka pro popis modelů nazvaných Conceptual Schema Definition Language (CSDL). Tento CSDL dokument je pak zabalen použitím formátu EDMX. (Odata, 2015)

4.2 Entity Data Model (EDM)

Hlavním konceptem v EDM jsou entity a asociace. Entity jsou instance *Entity Type* (například Inventura, Závod, Budova a tak dále), které jsou strukturovanými záznamy s klíčem *Entity Key* a složenými z pojmenovaných a typovaných vlastností. *Entity Key* je složen z podmnožiny vlastností v *Entity Type*. *Entity Key* (například *InventuraID*, nebo *BudovaID*) je zásadní koncept pro unikátní identifikaci instancí *Entity Type* a umožnění *Entity Type* instancím fungovat ve vztazích. Entity jsou seskupovány v *Entity Sets* (například Budovy je množina instancí *Entity Type* Budova).

Asociace definují vztah mezi dvěma nebo více *Entity Type* (například Budova patří Závodu). Instance asociací jsou seskupovány v *Association Sets*. *Navigation Properties* jsou speciální vlastnosti v *Entity Type*, které jsou vázány na konkrétní asociaci a mohou být použity k odkazování na asociaci entity.

Položením předchozích definic do OData termínů, feedy vystavované OData službou jsou reprezentovány pomocí *Entity Set*, nebo *Navigation Property* na *Entity Type*, které identifikuje kolekci entit. Například *Entity Set* identifikovaný pomocí URI `http://services.odata.org/OData/OData.svc/Products`, nebo kolekce entit identifikovaná pomocí **Products** *Navigation Property* v `http://services.odata.org/OData/OData.svc/Categories(1)/Products` identifikují feed složený z *Entry* vystavovaný OData službou. (Odata, 2015)

Každý *Entry* OData feedu je popsán v EDM pomocí **Entity Type** a každý *Link* mezi *Entry* je popsán pomocí **Navigation Property**. Převody mezi OData zdroji a prvky EDM jsou popsány v tab. 5.

Tab. 5 Popis OData zdrojů v Entity Data Modelu

OData zdroj	Je popsán v Entity Data Modelu pomocí
Collection	<ul style="list-style-type: none"> • Entity Set • Navigation Property na entity type, která identifikuje kolekci entit
Entry	<ul style="list-style-type: none"> • Entity Type • Poznámka: Entity Type mohou být součástí hierarchie typů
Property of an entry	<ul style="list-style-type: none"> • Primitive, nebo Complex Entity Type Property
Complex Type	<ul style="list-style-type: none"> • Complex Type
Link	<ul style="list-style-type: none"> • Navigation Property definovaná na Entity Type
Service Operation	<ul style="list-style-type: none"> • Function Import

Zdroj: <http://odata.org>

5 SAPUI5 aplikace pro inventuru hmotného majetku

5.1 Požadavky na aplikaci

Aplikace má sloužit jako náhrada původní FAIN aplikace na Windows Mobile terminálu. Systém inventury na backendu zůstává nezměněn, je tedy potřeba zachovat nezměněnou i funkcionální frontend aplikace. *prostor pro změny ve funkcionální nově frontend aplikace byl tímto omezen.

5.1.1 Funkční požadavky

- Aplikace bude online přistupovat do ERP systému SAP, konkrétně do struktur, které byly vytvořeny implementací systému FAIN.
- Uživatel bude používat k přihlášení do aplikace stejné uživatelské jméno a heslo, které používá při přihlašování do ERP systému SAP.
- Aplikace zobrazí seznam všech inventur v systému ZFAIN. U každé inventury budou informace o stavu inventury, datum provádění inventury, případně datum uzavření inventury.
- Uživatel může provádět inventuru hmotného majetku v aktivních inventurách. Uzavřené a zrušené inventury je možné prohlížet.
- Po nastavení umístění, závodu, budovy a místnosti aplikace zobrazí seznam majetků, které by se v tomto umístění měly nacházet. Tento seznam je možné filtrovat podle stavů majetků.
- Nasnímanému majetku je automaticky nastaveno umístění shodné s umístěním nastaveným v aplikaci. Pokud se toto umístění liší oproti umístění v iniciálním stavu majetku, je na to uživatel upozorněn zvýrazněním odlišných položek ve formuláři.
- Uživatel může nastavit nasnímanému majetku libovolné existující umístění.
- Uživatel může nastavit majetku pevný typ poznámky, který koresponduje s typy poznámek v systému ZFAIN.

5.1.2 Nefunkční požadavky

- Aplikaci bude možné provozovat na jakémkoliv zařízení s moderním webovým prohlížečem a digitálním fotoaparátem.
- Aplikace bude responzivní, bude se přizpůsobovat velikosti displeje, bude ji možné používat na mobilním telefonu, tabletu i desktopu.
- Uživatelské rozhraní bude vytvořeno podle zásad pokročilých uživatelských rozhraní. Přehledné, intuitivní a uživatelsky přívětivé.

5.2 Entity Data Model (EDM)

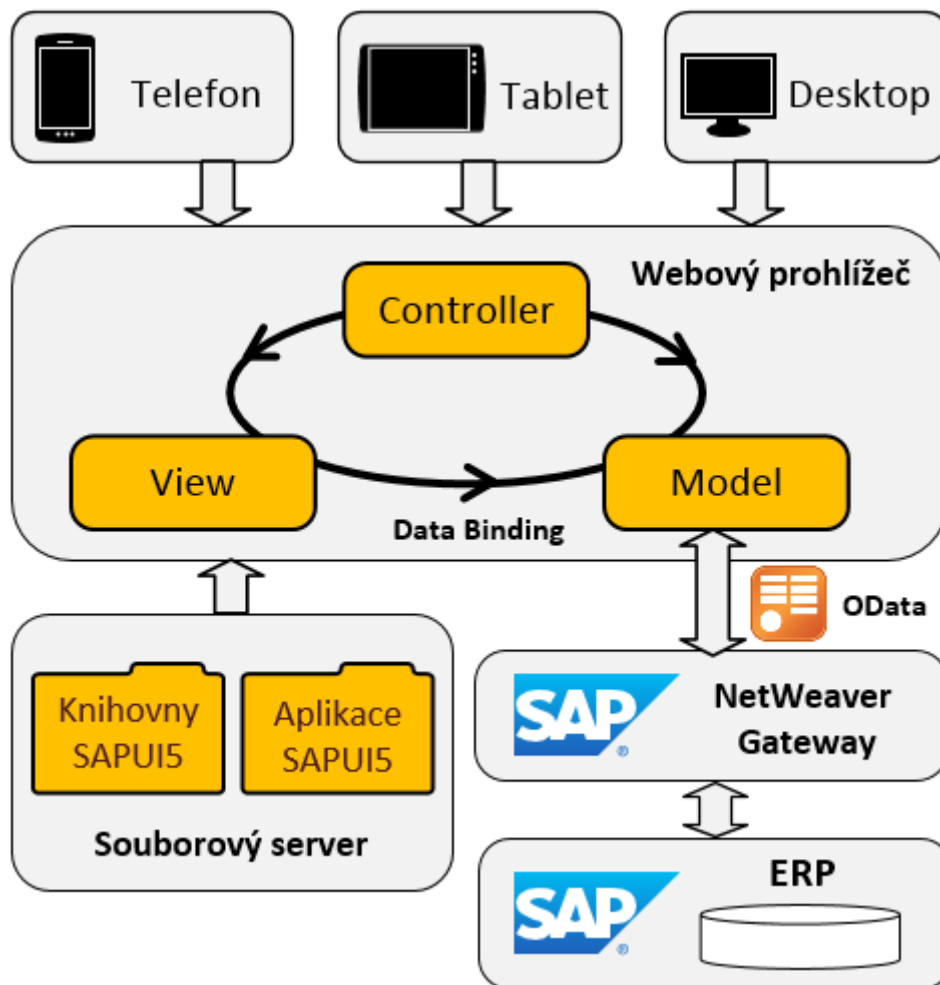
Při návrhu EDM bylo potřeba mít na paměti, že tabulky v databázi SAP obsahují velké množství dat. Jak již bylo uvedeno v tab. 4. Přenesení všech inventurních položek do aplikace by znamenalo ne jen zbytečně obsáhlý datový přenos, ale zejména velkou zátěž pro zařízení, na kterém aplikace poběží. Cílové zařízení bude většinou mobilní, tyto zařízení disponují omezenými systémovými prostředky (výkon CPU, kapacita operační paměti) a jejich nadměrné zatěžování výrazně zkracuje výdrž na baterii. Proto je potřeba OData službu navrhnout takovým způsobem, aby náročné operace, jako je vyhledávání v rozsáhlých tabulkách, filtrování a řazení probíhalo na back-endu. A aby se do aplikace přenášela jen data, která uživatel skutečně potřebuje. Výsledné EDM je v příloze C. Popis jednotlivých entit:

- **ZfainInv** – poskytuje aplikaci seznam všech inventur existujících v systému FAIN. Není potřeba zadávat klíč.
- **Plants** – poskytuje seznam závodů vybrané inventury. V dotazu je nutné zadat klíč inventury.
- **Buildings** – poskytuje seznam budov vybraného závodu. Je potřeba dodat klíč inventury a klíč závodu.
- **Rooms** – poskytuje seznam místností vybrané budovy. Vyžaduje klíč inventury, závodu a budovy.
- **InvItems** – poskytuje seznam inventurních položek vybrané místnosti včetně textových popisků. Je potřeba klíč inventury, závodu, budovy a místnosti. Pro vystavení informací o konkrétním majetku je potřeba klíč klienta, klíč inventury, inventurního seznamu, klíč čísla majetku a podčísla majetku.
- **Zfain** – je odlehčená verze entity InvItem, poskytuje informace o majetcích ve vybrané místnosti, většina informací je v podobě cizích klíčů číselníků, textové popisky chybí. V aplikaci se využívá pro zobrazení seznamu majetků v místnosti, kde většina textových popisků není potřeba. Implementace oproti entitě InvItem obsahuje méně příkazů SELECT. Použití entity Zfain namísto InvItem snižuje zátěž databáze.

5.3 OData Služba v ERP systému SAP

Business data, v našem případě evidence hmotného majetku VUT, jsou uložena v databázi ERP systému SAP. Konkrétně se jedná o verzi mySAP ERP 6 (ECC 6.0), tato verze je z roku 2005 a neobsahuje implementaci služby OData. Systém se může zdát zastaralý, ale takový stav je běžný ve všech firmách používajících rozsáhlý ERP systém SAP. Přechod na novější verzi ERP systému SAP je komplikovaný, zdoluhavý a také nákladný proces. Musí se importovat business data z databáze stávajícího systému do databáze nového systému. Po této operaci hrozí, že business logika, stávající se z programů napsaných pro starší verzi systému, nebude fungovat na novém systému korektně. Z tohoto důvodu se upgrade ERP systé-

mu s business data provádí v delších časových intervalech, zhruba 10 let. Potřebujeme tedy dobře fungující ERP systém, ale zároveň chceme využívat nejnovější technologie. Řešením je SAP NetWeaver Gateway. V našem případě verze EHP1 NetWeaver 7.31, SPS07, z roku 2013. Je to server, který neobsahuje business data, ale může si je vyžádat z ERP systému, na který je napojen prostřednictvím tRFC. SAP NetWeaver Gateway přímo neobsahuje business data, proto je přechod na novější verzi relativně snadný, tudíž je únosné provádět upgrade v kratších časových intervalech například jeden, nebo dva roky.



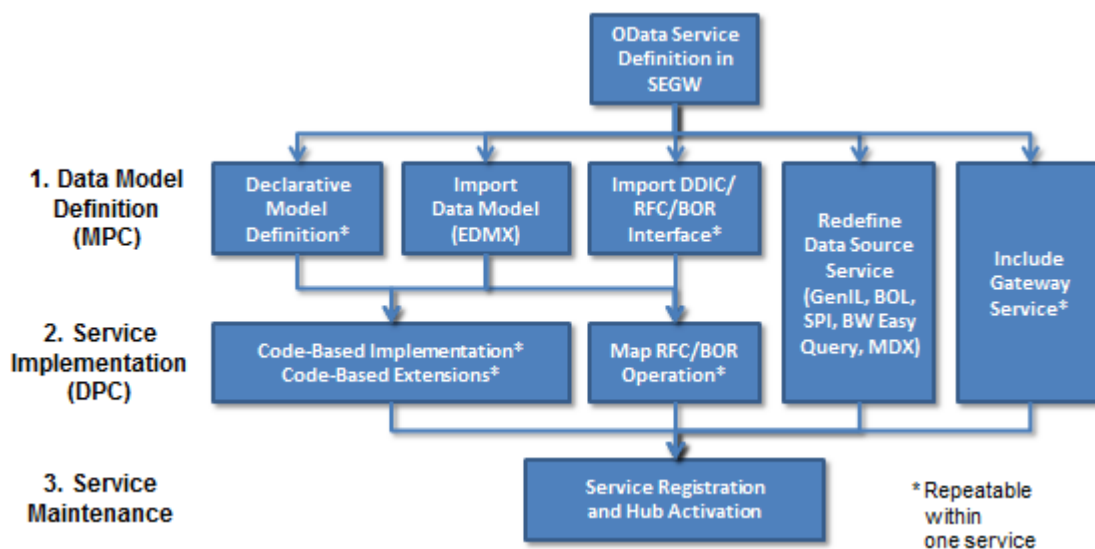
Obr. 18 Schéma propojení SAPUI5, OData a SAP ERP
Zdroj: <http://help.sap.com>, upraveno.

Na obr. 18 je schéma spuštění a činnosti SAPUI5 aplikace. Po otevření webového prohlížeče a zadání správné URI se do prohlížeče načtou ze souborového serveru soubory SAPUI5 knihovny a soubory samotné aplikace. V prohlížeči se vytvoří Model, View a Controller. Model bude odesílat požadavky ke čtení a zápisu dat pomocí OData služby na SAP NetWeaver Gateway. Implementace OData služby na Gateway bude komunikovat prostřednictvím tRFC se systémem SAP, který má požadovaná

business data ve své databázi. Data doručená do Modelu jsou prostřednictvím Data Bindingu vázána na ovládací prvky ve View. View zobrazuje data uvnitř ovládacích prvků. Controller reaguje na akce vyvolané uživatelem.

5.4 Implementace OData služby v SAP

Gateway služba je implementována vytvořením dvou ABAP tříd – Model Provider class a Runtime Data Provider class. (SAP SE, 2011) Jak je vidět na obr. 19, definici tříd je možné provést více způsoby. V našem případě to bude deklarativní definice modelu a implementace programovým kódem v jazyce ABAP.



Obr. 19

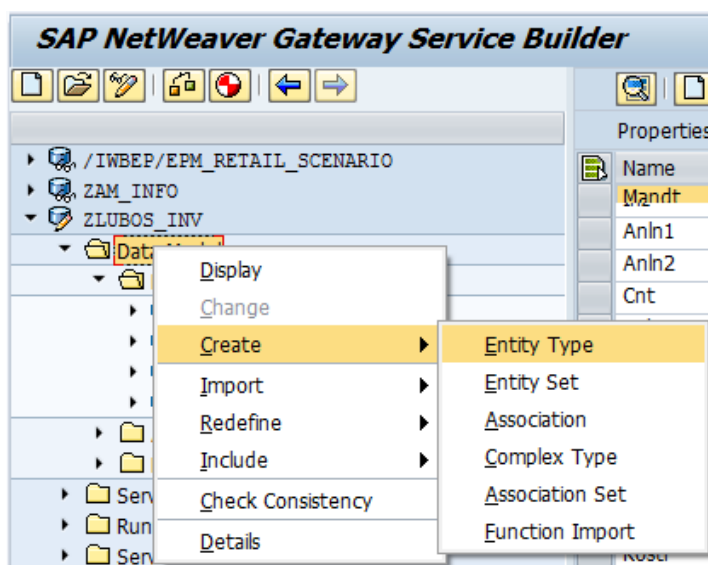
Zdroj: <http://scn.sap.com> The new SAP NetWeaver Gateway Service Builder: Build new OData Services in 3 Quick Steps

5.4.1 Vytvoření nového projektu

Pomocí SAP GUI se přihlásíme do systému SAP ERP. Zadáním transakčního kódu *SEGW* spustíme nástroj SAP NetWeaver Gateway Service Builder. Vlevo je zobrazen seznam služeb, které již na SAP NetWeaver existují. Tlačítkem Create Project vytvoříme novou službu.

5.4.2 Vytvoření Entity Type

V nově vytvořené službě je potřeba nejprve definovat Entity Type, z kontextového menu složky Entity Types stačí vybrat Create a Entity Type, viz obr. 20.



Obr. 20 Vytvoření Entity Type

V dialogovém okně stačí zadat název Entity Type, název musí odpovídat svému Entity Type v EDM viz příloha C.

5.4.3 Vytvoření Properties (vlastností)

Nyní je možné definovat vlastnosti, ze kterých bude Entity Type složeno, názvy musí odpovídat názvům v EDM. Maximální délka a datový typ vlastnosti musí korespondovat s maximální délkou a datovým typem odpovídajícího sloupce v databázi SAP, viz obr. 21.

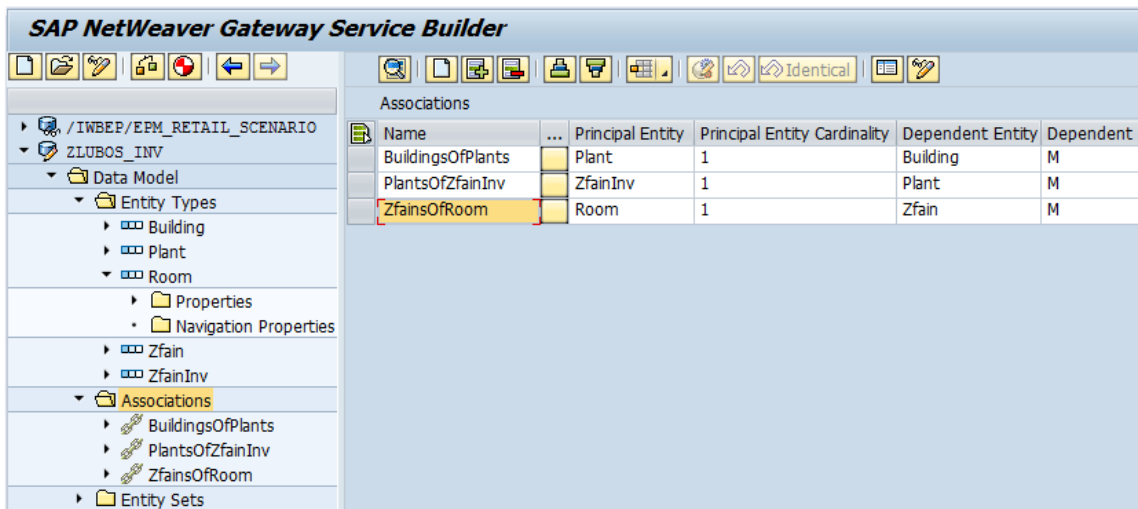
 The screenshot shows the 'Properties' dialog box in SAP NetWeaver Gateway Service Builder. It contains a table with columns: Name, Schlüssel, EDM-Coretyp, Genauigkeit, Sk..., M..., Einh..., and Bezeic... (with several empty columns). The 'Raumn' property is highlighted in yellow.

Name	Schlüssel	EDM-Coretyp	Genauigkeit	Sk...	M...	Einh...	Bezeic...	...	KompTyp	ABAP-Feldname
Inv	<input checked="" type="checkbox"/>	Edm.String	8	0	0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			T	INV
Werks	<input checked="" type="checkbox"/>	Edm.String	4	0	0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			T	WERKS
Stort	<input checked="" type="checkbox"/>	Edm.String	10	0	0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			T	STORT
Raumn	<input checked="" type="checkbox"/>	Edm.String	8	0	0		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			T	RAUMN

Obr. 21 Vlastnosti Entity Type

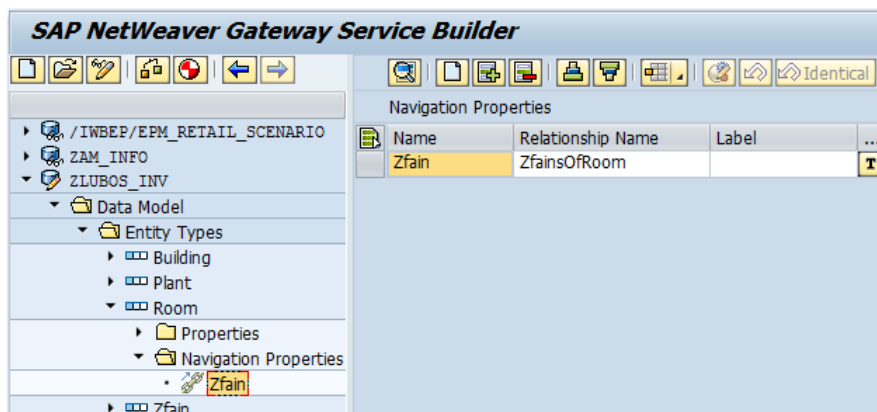
5.4.4 Definice navigace

Nyní definujeme vztah 1:M mezi Entitou Room a Zfain, jak to definováno v EDM. Ve složce Associations definujeme novou asociaci, zadáme název, řídicí a závislou entitu a příslušné kardinality, viz obr. 22.



Obr. 22 Definice asociace

Dále je potřeba nastavit Navigaci pro entitu Room. Chceme mít možnost přejít od konkrétně identifikované Místnosti na seznam (Entity Set) všech majetků, které se v této místnosti nacházejí. V Entity Set Room přejdeme do složky Navigation Properties a vytvoříme novou navigaci, název bude sloužit jako klíč v URI služby. V Relationship Name zvolíme asociaci ZfainsOfRoom, kterou jsme definovali výše.



Obr. 23 Definice navigace

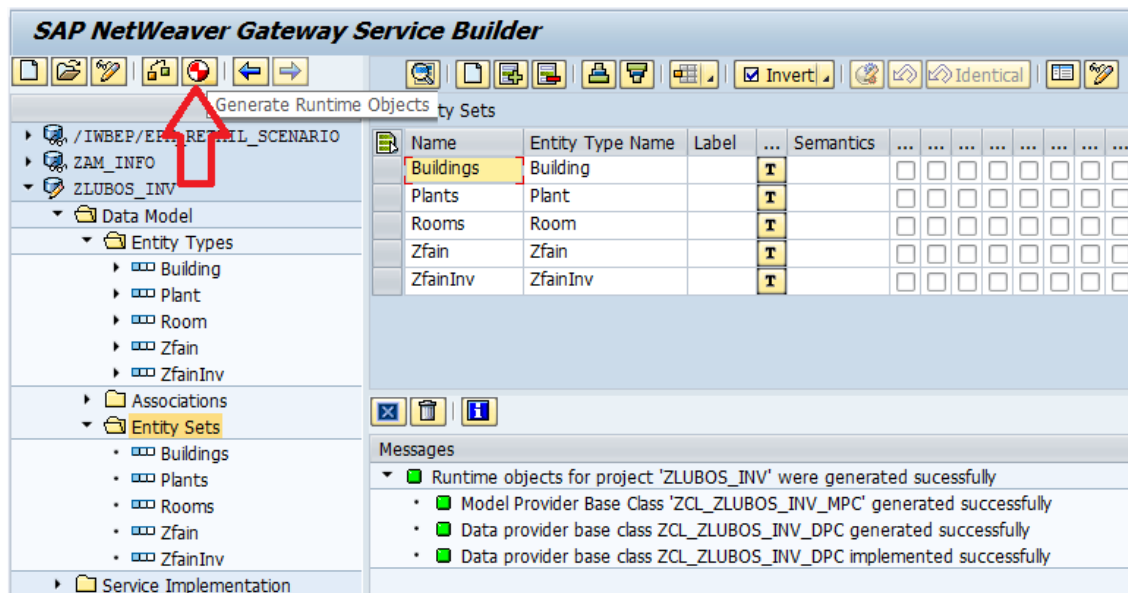
5.4.5 Vytvoření Entity Set

Vytvoříme Entity Set s názvem Rooms na základě Entity Type Room, který jsme definovali v předchozích krocích. Ve složce Entity Sets přidáme nový záznam.

5.4.6 Generování podkladových tříd ABAP

Aby bylo možné službu používat, musí se nejprve vygenerovat podkladové třídy ABAP. Při generování tříd se zohledňují všechna nastavení, která jsme specifikovali v Gateway Service Builderu. Pokud provedeme další změny v Gateway service bui-

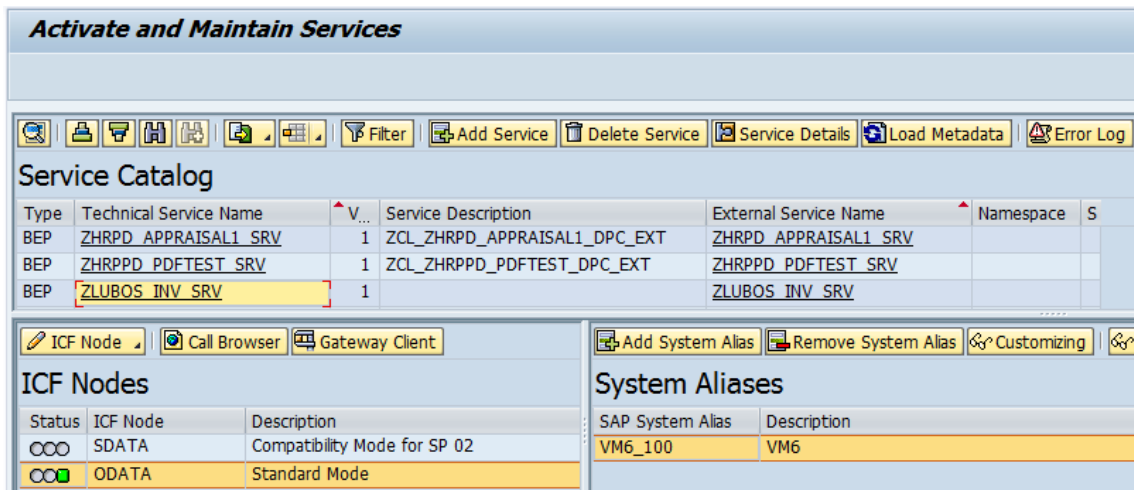
lderu po vygenerování tříd, a chceme, aby se tyto změny promítly do fungující služby, musíme generování spustit znovu. Generování se spouští tlačítkem Generate Runtime Objects, po dokončení generování se zobrazí log, pokud všechno proběhne v pořádku, budou všechny indikátory svítit zeleně, viz obr. 24.



Obr. 24 Vytvoření Entity Set a generování podkladových objektů

5.4.7 Registrace a aktivace služby na SAP NetWeaver Gateway

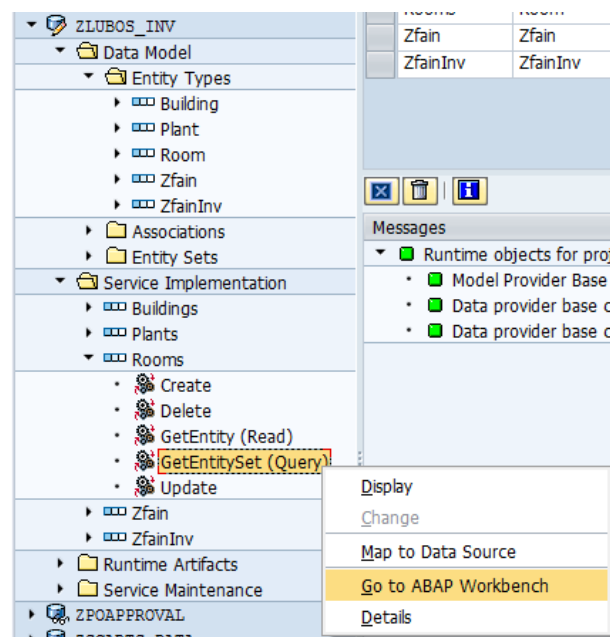
Aby bylo možné službu používat, musíme ji nejprve aktivovat. To znamená přidat ji do seznamu služeb, které jsou dostupné prostřednictvím SAP NetWeaver Gateway. Ve složce Service Maintenance vytvořené služby klikneme na tlačítko Register, SAP GUI zobrazí upozornění o přesměrování na systém SAP NetWeaver Gateway, v zobrazeném dialogu Add Service zadáme název služby, stejný název bude sloužit k identifikaci služby na serveru SAP NetWeaver Gateway. Pokud je služba aktivní, ve sloupci Status svítí kontrolka zeleně, jak je vidět na obr. 25.



Obr. 25 Aktivace služby na SAP NetWeaver Gateway

5.4.8 Implementace metod třídy Data Provider class

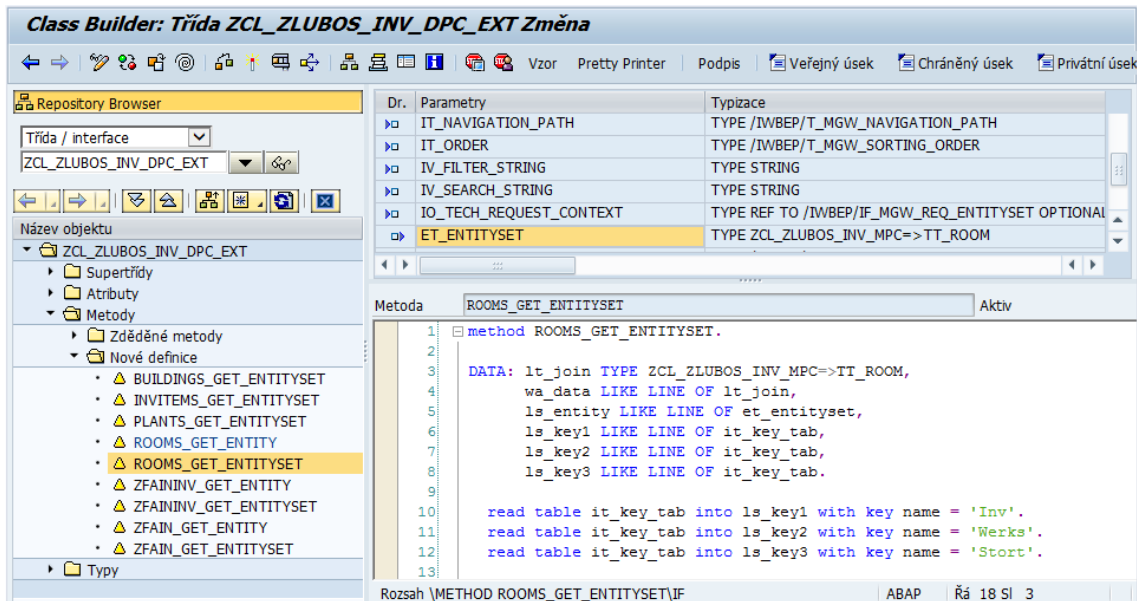
Nyní je potřeba implementovat metody, které zajistí načtení dat z databáze. ERD využívané části databáze ERP SAP je v příloze B. Rozbalíme složku Service Implementation a pak položku *Rooms* a vyhledáme položku „Get EntitySet (Query)“, z jejího kontextového menu zvolíme položku Go to ABAP Workbench. Postup je zachycen na obr. 26.



Obr. 26 Přechod do Class Builderu

Ve stromu objektu vlevo rozbalíme složku *Metody* a pak *Zděděné metody*, vyhledáme metodu `ROOMS_GET_ENTITYSET`, z kontextového menu vybereme *Redefinice*.

Tím se metoda přesune do složky *Nové definice* a otevře se editor zdrojového kódu, viz obr. 27.



Obr. 27 Metoda ROOMS_GET_ENTITYSET v Class Builder

Implementace metody ROOMS_GET_ENTITYSET:

```
method ROOMS_GET_ENTITYSET.
```

```
DATA: lt_join TYPE ZCL_ZLUBOS_INV_MPC=>TT_ROOM,
      wa_data LIKE LINE OF lt_join,
      ls_entity LIKE LINE OF et_entityset,
      ls_key1 LIKE LINE OF it_key_tab,
      ls_key2 LIKE LINE OF it_key_tab,
      ls_key3 LIKE LINE OF it_key_tab.
```

```
read table it_key_tab into ls_key1 with key name = 'Inv'.
read table it_key_tab into ls_key2 with key name = 'Werks'.
read table it_key_tab into ls_key3 with key name = 'Stort'.
```

```
SELECT zmst~raumnt zf~raumn zf~inv zf~werks zf~stort
FROM zfain as zf
JOIN zmistnostt as zmst
on zf~werks eq zmst~werks
AND zf~stort eq zmst~stort
AND zf~raumn eq zmst~raumn
into TABLE lt_join
WHERE zf~inv EQ ls_key1-value
AND zf~werks EQ ls_key2-value
```

```
AND zf~stort EQ ls_key3-value.

SORT lt_join by raumn.
DELETE ADJACENT DUPLICATES FROM lt_join comparing raumn.

LOOP AT lt_join INTO wa_data.
  ls_entity-werks = wa_data-werks.
  ls_entity-raumn = wa_data-raumn.
  ls_entity-inv = wa_data-inv.
  ls_entity-stort = wa_data-stort.
  ls_entity-raumnt = wa_data-raumnt.

APPEND ls_entity to et_entityset.
ENDLOOP.
```

Entita Rooms čerpá data z tabulek ZFAIN a ZMISTNOSTT databáze SAP. Z tabulky ZFAIN potřebujeme vyfiltrovat seznam unikátních místností, v tabulce ZMISTNOSTT se pak nachází textový popis místnosti.

Po definici proměnných jsou načteny parametry z tabulky it_key_tab. Tabulka it_key_tab je naplněna klíči zadanými v URI OData služby. Následuje příkaz SELECT. OData služba má poskytovat seznam unikátních místností, nabízí se použití výrazu DISTINCT, ale jak uvádí Radhakrishnan (2015) je z hlediska výkonu lepší variantou načíst všechny místnosti do interní tabulky, seřadit je, a pak pomocí příkazu DELETE ADJACENT DUPLICATES smazat duplicitní záznamy. Pro zlepšení výkonu také není použit výraz INTO CORRESPONDING FIELDS of Table, sloupce jsou vypsané za výrazem SELECT ve shodné sekvenci, v jaké jsou v tabulce et_entityset. Správná sekvence byla zjištěna spuštěním metody v debuggeru.

5.5 Návrh GUI

Obrázky aplikace se nacházejí v příloze D. Aplikaci budou používat členové inventurní komise, kteří před každou inventurou procházejí školením. Během školení inventurní komise jsou uživatelé mimo jiné detailně seznámeni s ovládáním aplikace.

Aplikace slouží primárně ke snímání majetků, může také sloužit k prohlížení uzavřených a zrušených inventur. Vzhledem k rozsahu inventury, nastíněné v tab. 4, byl kladen maximální důraz na to, aby uživatel mohl snímat jeden majetek za druhým a ničím se zbytečně nezdržoval.

5.5.1 Výběr inventury

Na této obrazovce, viz obr. 30, potřebuje uživatel vybrat tu správnou inventuru. Rozhodovat se bude podle stavu. Inventura musí být aktivní, v uzavřených a zrušených inventurách nelze provádět změny, lze jen prohlížet. Z tohoto důvodu jsou aktivní inventury označeny zelenou ikonou. Dalším rozhodujícím faktorem je ná-

zev inventury. Na VUT Brno je název inventury zpravidla složen ze zkratky fakulty, čísla součásti v závorkách a roku, příklad: *FAVU (11) 2013*. Předpokládáme, že uživatel byl na poradě informován, na jakých inventurách má pracovat. Důležité je také datum inventury, aby se uživatel omylem nesnažil pracovat na inventuře se stejným názvem z předchozích let. Pokud byla inventura již uzavřena, je zobrazeno datum uzavření. Pokud došlo ke změně v inventuře, je zobrazeno datum poslední změny. Také se zobrazuje, pro uživatele nepříliš zajímavé, číslo inventury, podle tohoto čísla jsou inventury v tabulce seřazeny, tím je zajištěno, že na předních pozicích budou inventury týkající se aktuálního roku.

5.5.2 Výběr umístění

Poté, co uživatel vybere inventuru, přepne se typ aplikace z jednooknové na dělené okno. A v rámci celé aplikace dojde k zanoření o jednu úroveň. Viz obr. 31. Levá část děleného okna slouží k nastavení umístění a pravá část slouží k zobrazení seznamu majetků, později k zobrazení a editaci konkrétního majetku.

První dropdown menu slouží k výběru závodu, druhý slouží k výběru budovy, následuje seznam místností. V záhlaví okna je uživatel informován o inventuře, kterou si vybral na předchozí obrazovce, dále se v záhlaví nachází tlačítko pro vyvolání dialogového okna, kde je možnost návratu na obrazovku s výběrem inventury. Výběr inventury je operace, kterou ve většině případů uživatel provede jen jedenkrát za den. Z tohoto důvodu je možnost přechodu zpět na výběr inventury zanořena o další úroveň pomocí dialogového okna. Nabízí se řešení s nižší kognitivní zátěží – tlačítko, které nebude vyvolávat dialogové okno, ale přímo provede akci zpět na výběr inventur. Při manipulaci s dotykovým zařízením existuje možnost, že bude tlačítko stisknuto omylem. Pokud tlačítko zobrazí dialogové okno, stačí zrušit akci. Ale v případě že tlačítko provede rovnou akci zpět na seznam inventur, bude muset uživatel znovu vybrat inventuru a poté závod, budovu a místnost. Nebo použít tlačítko vpřed webového prohlížeče, to ale většinou není na mobilních prohlížečích snadno dostupné. Dialogové okno tedy zajišťuje, že se uživatel vrátí na výběr inventury pouze v případě, že takovou akci vědomě vyvolá.

5.5.3 Výběr majetku

V pravé části děleného okna je v tabulce zobrazen seznam majetků v místnosti. Na této obrazovce je aplikace připravena snímat majetky. Stisknutím tlačítka skenovat se spustí fotoaparát pro snímání čárového kódu. Nebo lze majetek vybrat v tabulce. Obsah tabulky je možné filtrovat podle stavu majetků, uživatel tak má lepší přehled o průběhu zpracování inventury v místnosti. Protože na mobilních telefonech bude panel s umístěním minimalizován, jak je vidět na obr. 34. Je v záhlaví okna zobrazeno nastavené umístění, uživatel je tak vždy informován o nastaveném umístění. Pro uživatele je tato informace velmi důležitá, protože se může stát, že se bude v místnosti nacházet majetek, který má správně být umístěn někde jinde.

5.5.4 Formulář majetku

Jak můžeme vidět na obr. 32. Při provádění inventury kontroluje uživatel u každého majetku dva atributy. První je stav majetku, druhý je umístění majetku. Pokud je otevřený majetek v iniciálním stavu, je mu automaticky nastaven stav O – ok, nalezeno. Tato situace nastává u většiny majetků, proto je tento stav automaticky přednastaven při otevření formuláře nezpracovaného majetku. Při snímání jednoho majetku za druhým toto automatické nastavení omezuje zdržování uživatele zbytečnou manipulací s formulářem. Pokud se uživatel přepne na záložku poznámky, může upřesnit stav majetku výběrem některé pevně definované poznámky z dropdown menu.

Může se stát, že majetek od poslední inventury někdo přemístil do jiné místnosti, nebo i budovy. V takovém případě dojde k naskenování majetku, který se má nacházet v jiném umístění, než je aktuálně nastaveno v aplikaci. Aplikace automaticky nastaví majetku stejné umístění, jaké je definováno v aplikaci. Změněné položky jsou ve formuláři zvýrazněny modrou barvou. Uživatel je o změně umístění také informován zobrazením warning message strip. Kliknutím na tlačítko editace může uživatel měnit umístění a stav majetku. Kliknutím na tlačítko potvrzení dojde k okamžitému odeslání dat do SAP. Kliknutím na zrušení nedojde k žádné manipulaci s daty a zobrazí se seznam majetků v místnosti.

Pokud se v seznamu majetků místnosti nacházejí majetky, které v místnosti chybí. Uživatel je ponechá ve stavu I – iniciální. Tyto majetky mohou být nalezeny později v jiné místnosti. Stav M – chybí se nastavuje v SAP GUI až po ukončení prohlídek místností komisemi všem majetkům, které zůstaly ve stavu I.

5.5.5 Editace majetku

Po kliknutí na tlačítko editace ve formuláři majetku může uživatel libovolně měnit stav a umístění majetku. Jak je vidět na obr. 33, formulář zobrazuje i původní hodnoty. Tlačítko uložit uloží změny, tlačítko zrušení zruší provedené změny. V obou případech se pak aplikace vrátí na zobrazení formuláře majetku.

5.6 Implementace

Schéma souborů aplikace je v příloze E. Aplikace stojí na architektuře MVC. Implementace probíhala v nástroji Eclipse Java EE IDE for Web Developers. Aplikace byla vytvořena v souladu s komponentovým konceptem jako jedna komponenta. To znamená, že aplikace může být spouštěna v jednoduché, samostatné HTML stránce, nebo v širším kontextu Lunchpadu SAP Fiori.

index.html slouží jako kontejner ke spuštění komponenty. Obsahuje následující:

- Meta tagy v hlavičce dokumentu, pro Microsoft Internet Explorer, aby používal nejnovější vykreslovací jádro (EDGE). A nastavení znakové sady dokumentu na UTF-8.

- Bootstrap, načtení a inicializace SAPUI5. Atribut *src* prvního `<script>` tagu říká prohlížeči, kde najít SAPUI5 core knihovny, to inicializuje SAPUI5 runtime a načte dodatečné zdroje. Ovládací prvky SAPUI5 podporují různá témata, nastavíme *sap_bluecrystal* jako výchozí motiv. Specifikujeme požadované knihovny UI *sap.m* obsahující ovládací prvky UI, které potřebujeme ve spuštěné aplikaci. Je nastavena komplexní binding syntaxe. Pro využití nejnovější funkcionality SAPUI5 definujeme verzi kompatibility jako *edge*. Nakonfigurujeme spuštění procesu bootstrappingu asynchronně. To znamená, že prostředky mohou být načítány současně v pozadí z důvodu lepšího výkonu. (SAP SE, 2015) Nakonec je definován jmenný prostor aplikace a cesta k souboru s komponentou.
- Když jsou načteny všechny zdroje a knihovny, SAPUI5 je připraveno, dojde ke spuštění globální události *init*. K této události je zaregistrována anonymní funkce zpětného volání, která se provede, když je bootstrap SAPUI5 dokončen. V anonymní funkci pomocná metoda *sap.ui.core.ComponentContainer* instancuje komponentu vyhledáním souboru *Component.js* ve jmenném prostoru, který jí je předán jako argument. Komponenta je vytvořena uvnitř ovládacího prvku *sap.m.Shell*, ten kontroluje maximální šířku vykreslení aplikace. Nakonec je *sap.m.Shell* umístěn metodou *placeAt* do těla HTML dokumentu.

Component.js obsahuje definici objektu *sap.ui.core.UIComponent*, který představuje celou aplikaci jako komponentu. Ve funkci *init* komponenty jsou vytvořeny modely aplikace a nastaveny na komponentu. Takto nastavené modely jsou pak dostupné ve všech views a jejich controllerech.

V našem případě je nejprve vytvořen OData model pro přístup k OData službě na SAP NetWeaver Gateway. Pro vytvoření OData modelu je potřeba dodat URI služby. Toto URI se nachází v konfiguračním souboru *manifest.json* a je zde načteno pomocí funkce *getMetadata().getConfig()*. Dále je na OData modelu vypnuta funkce dávkového seskupování requestů (batch). Batch je vestavěná optimalizace výkonu OData modelu, která umožňuje poslat několik požadavků najednou. (SAP SE, 2015) Aktuálně používaná verze SAP NetWeaver Gateway na VUT Brno batch funkci nepodporuje. Je to možné jistit zadáním URI služby s parametrem *\$batch* do webového prohlížeče. Proto musí být tato funkce na modelu vypnuta. Po vytvoření je model nastaven na komponentu s názvem *nwProducts*.

Dále je vytvořen lokální JSONModel, bude sloužit k uložení nastaveného umístění v aplikaci. Položky modelu a jejich iniciální hodnoty jsou definovány JavaScript objektem. Po vytvoření je model nastaven na komponentu pod názvem *Location*.

Dojde také k vytvoření modelů specifikovaných v souboru *manifest.json* v sekci *models*. V našem případě je zde specifikován pomocný model pro internacionalizaci (i18n). Specifikace obsahuje název modelu *i18n*, typ modelu *ResourceModel* a cestu ke zdrojovému souboru *i18n.properties*.

Dále se provádí inicializace routeru *sap.ui.core.routing.Router*. V *main* sekci komponenty je instancován root view.

manifest.json slouží jako definice metadat komponenty. Metadata zahrnují konfiguraci na úrovni aplikace a informace o routingu.

formatter.js a **formatterForm.js** obsahují definice formátovacích funkcí.

i18n.properties ve složce *i18n* uchovává texty pro internacionalizační model. Díky tomu, že se všechny texty UI nacházejí v jednom souboru, může být aplikace snadno přeložena do jiných jazyků. SAPUI5 se pokusí načíst nejpřrsnější locale-specific soubor jazykovému nastavení webového prohlížeče, s postupným přechodem až k „catchall“ případu, který představuje soubor *i18n.properties*. (SAP SE, 2013)

5.6.1 Views

SAPUI5 umožňuje definovat UI pomocí JavaScript, XML, JSON, nebo HTML. V našem případě bylo použito XML.

App.view.xml v tomto souboru je definováno root view, které je použito jako výchozí při instancování komponenty. Obsahuje jeden ovládací prvek *sap.m.App*. Tento ovládací prvek slouží pouze jako kontejner pro jednooknovou aplikaci. Do jeho agregace *pages* může router vkládat další views. Naše aplikace někdy používá návrhový vzor děleného okna, k tomu slouží ovládací prvek *sap.m.SplitContainer* definovaný v souboru *Container.view.xml*. Obě tyto view nemají přiřazený controller.

Nejdříve je vždy vytvořen root ovládací prvek *sap.m.App*, pokud je potřeba návrhový vzor jednooknové aplikace, pak router vkládá views do agregace *pages* tohoto ovládacího prvku. Pokud je potřeba návrhový vzor dělené okno, pak router nejprve vloží do agregace *pages* ovládací prvek *sap.m.SplitContainer* a do jeho agregací *masterPages* a *detailPages* může vkládat další views.

Aby bylo zajištěno, že router nejprve vloží ovládací *sap.m.SplitContainer* a budou tak existovat agregace *masterPages* a *detailPages*, musí mít všechny targety, které má router vkládat do těchto agregací definovanou vlastnost *parent*. Ta musí odkazovat na target se souborem *Container.view.xml*.

U Views je nutné dodržovat stromovou strukturu. Ovládací prvky je možné umisťovat jen do agregací nadřazeného prvku. Jinak XML preprocesor nedokáže views zpracovat a aplikace se nespustí. Možné agregace jsou uvedeny v dokumentaci u každého ovládacího prvku.

Home.view.xml obsahuje ovládací prvek *sap.m.Page* a v jeho agregaci *content* pak ovládací prvek *sap.m.Table*. Table slouží k zobrazení seznamu inventur, je to jediné view, které router vkládá do agregace *pages* a využívá tak jednooknový typ aplikace.

Master.view.xml obsahuje dva ovládací prvky *sap.m.Select*, ty slouží jako dropdown choosery pro výběr závodu a budovy. Seznam místností je zobrazen v ovládacím prvku *sap.m.List*.

Detail.view.xml je určeno jen k vyplnění plochy agregace *detailPages*, v době, kdy uživatel ještě nezvolil umístění.

SuperDetail.view.xml obsahuje ovládací prvek *sap.m.IconTabBar*, ten slouží k filtrování majetků. Majetky se zobrazují v ovládacím prvku *sap.m.List*, který je umístěn v agregaci *content* zmiňovaného *IconTabBar*.

SuperDetailForm.view.xml obsahuje ovládací prvek *sap.m.Page* s prázdnou agregací *content*. Naplnění obsahu stránky probíhá až za běhu aplikace, controller tohoto view podle potřeby vkládá do agregace *content* fragmenty *Display.fragment.xml*, nebo *Change.fragment.xml*. V agregaci *footer* se pak nachází *sap.m.Toolbar* a několik tlačítek *sap.m.Button* pro manipulaci s formulářem.

Display.fragment.xml zobrazuje panel *sap.m.IconTabBar* se záložkami Vlastnosti a Poznámka. Agregace *content* záložky vlastnosti obsahuje formulář *sap.ui.layout.form.SimpleForm* pro zobrazení všech vlastností majetku. Každá vlastnosti je zobrazena pomocí *sap.m.Label* a *sap.m.Text*. Formulář zobrazuje aktuální vlastnosti majetku a zvýrazněně vlastnosti stavu a umístění, které byly změněny, a uživatel se je chystá potvrdit. Zvýraznění je dosaženo nastavením jiné třídy ve vlastnosti *class*. Zobrazování položek při změně zajišťuje výraz ve vlastnosti *visible*. Tento výraz načte data ze dvou různých modelů a porovná je, podle výsledku pak nastaví vlastnost *visible* na hodnotu *true* nebo *false*. Poznámka obsahuje dropdown menu *sap.m.Select* s pevně danými typy poznámek *sap.ui.core.Item*.

Change.fragment.xml zobrazuje stejné vlastnosti majetku, jako *Display.fragment* a navíc umožňuje nastavit nový stav a umístění pomocí dropdown menu *sap.m.Select*.

MainMenu.fragment.xml tento fragment je zobrazen při stisknutí tlačítka v záhlaví *Master.view.xml*. Obsahuje dialogové okno *sap.m.Dialog*. Položka *sap.m.ActionListItem* slouží pro návrat na seznam inventur.

NotFound.view.xml toto view router použije, když se žádné jiné view nemitane s požadavkem na zdroj.

5.6.2 Controllery

BaseController.js třída Base Controller představuje základ, ze kterého všechny ostatní controllery budou dědit. Funkce implementované v Base controlleru jsou opakovaně použity jeho podtřídami. Funkce *getRouter* slouží pro snadnější přístup ke komponentám routeru v celé aplikaci. Event handler *onNavBack* kontroluje, jestli je předchozí hash hodnota v historii aplikace. Pokud ano, tak přesměruje na předchozí hash pomocí nativního History API prohlížeče. V případě, že neexistuje žádný předchozí Hash použije se router k navigaci na routu *appHome*. (SAP SE, b2015)

Home.controller.js event handler *onListItemPressed* se spustí, když uživatel vybere ze seznamu inventuru. Z parametru *oEvent* je načten objekt vybrané položky. Číslo a název inventury se uloží do *Location* modelu a pak se použije router k navigaci na routu *master*, předá se konfigurační objekt do *navTo* metody s povinným parametrem *invId* naplněným správným číslem inventury. Router vždy zajistí, že jsou specifikovány všechny povinné parametry, jak jsou uvedeny v patternu routy, jinak navigace není provedena a v konzoli prohlížeče se vypíše chyba.

Master.controller.js ve funkci *onInit* je z routeru načtena routa *master* a je k ní připojena event handler funkce *_onRouteMatched*, zde je možné přistupovat k parametru *arguments* z parametru *oEvent*, který obsahuje všechny parametry patternu routy. Jméno povinného parametru *invId* odpovídá *{invId}* v definici patternu

routy *master*, tím je také zajištěno zobrazení čísla inventury v URL. Pomocí funkce *bindAggregation()* je vygenerován seznam závodů do dropdown menu. Funkce přebírá název agregace, šablonu generovaných položek. Cestu k entitě v datovém modelu, zde je použito číslo inventury z patternu routy. OData model načítá data ze služby asynchronně, aby bylo možné přistupovat k načteným závodům je potřeba použít událost *dataReceived*. V této události je pomocí stejné funkce *bindAggregation()* vygenerován seznam budov z prvního načteného závodu. Stejným způsobem je naplněn také seznam místností. Funkce *onSelectPlant* a *onSelectBuilding* jsou spuštěny, když uživatel vybere závod nebo budovu v příslušném dropdown menu. Dojde k naplnění seznamu místností, případně i dropdown menu budov stejným způsobem, jak bylo popsáno ve funkci *_onRouteMatched*. Event handler *onRoomListItemPressed* je spuštěn, když uživatel zvolí místnost, ve které chce provádět inventuru. Nejprve dojde k uložení dat o umístění do *Location* modelu a pak je pomocí routeru spuštěna navigace na routu *superDetail*. Funkce *onOpenMainDialog* je aktivována stisknutím tlačítka menu. Pomocná funkce *_getDialog* instancuje *MainMenu.fragment.xml*. Protože fragmenty nemají svůj controller, je předán odkaz na aktuální controller objekt a event *onBackToHomePage* vyvolaný ve fragmentu po stisknutí tlačítka je zpracován v tomto controlleru. Spustí se navigace na routu *appHome*, to způsobí přechod na obrazovku se seznamem inventur.

SuperDetail.controller.js ve funkci *onInit* je z routeru načtena routa *superDetail* a je k ní připojena event handler funkce *_onRouteMatched*, zde je vygenerován seznam majetků pomocí funkce *bindAggregation()*. Klíče číslo inventury, id závodu, id budovy a id místnosti potřebné pro dotaz na OData službu jsou načteny z patternu routy. Funkce *handleIconTabBarSelect* je spuštěna při každém výběru filtru nad seznamem majetků. Filtrování pracuje tak, že je odeslán dotaz na OData službu s příslušným filtrem za klíčem *\$filter* v URI. Filtrování tak probíhá na back endu a není jím zatěžováno koncové zařízení. Když uživatel vybere majetek, spustí se funkce *onSelectProperty*. Dojde ke spuštění navigace na routu *superDetailForm*. Funkce *doScan* je spuštěna po kliknutí na tlačítko Scan. Zavolá se funkce *scan* z Cordova pluginu *barcodeScanner*. Pokud je čárový kód úspěšně naskenován, použije se číslo a podčíslu majetku k navigaci na routu *superDetailForm*.

SuperDetailForm.controller.js ve funkci *onInit* je načtena routa *superDetailForm* a je k ní připojena event handler funkce *_onRouteMatched*. Dále je zavolána pomocná funkce *_showFormFragment*, která zajistí zobrazení formuláře majetku načtením fragmentu *Display.fragment.xml*. Funkce *onExit* je volána při opuštění view *SuperDetailForm*, slouží pro uvolnění zdrojů zabraných zobrazováním fragmentů. Ve funkci *_onRouteMatched* je funkcí *bindElement()* vytvořena vazba do view. Je použit model *nwProducts*, ovládací prvky používající tuto vazbu zobrazují původní stav a umístění majetku. Klíče pro OData dotaz jsou brány z patternu routy. Dále je vytvořen nový OData model, URI je stejné, jako u modelu *nwProducts*. Tentokrát je ale binding mode nastaven na *TwoWay*. To umožňuje modelu ukládat data o majetku na backend pomocí metody POST. Model je nastaven na aktuální view jako nepojmenovaný. Následuje nastavení vazby do view metodou *bindElement()*. Ovládací prvky vázané na tento nepojmenovaný model reprezentují nový stav a umís-

tění majetku. Kvůli asynchronnímu načítání dat do modelu je použita funkce *attachRequestCompleted()*. Nejprve dojde k porovnání původního umístění majetku v modelu *nwProducts* s umístěním nastaveným v aplikaci v modelu *Location*. Informace, zda došlo ke změně umístění majetku je uložena do modelu *Location*. Pokud je původní stav majetku iniciální, je do nepojmenovaného modelu nastaven stav 0 – nalezeno nebo stav X – nalezeno, změna umístění. Událost *handleEditPress* je vyvolána po stisknutí tlačítka Editovat. Nejdříve pomocná funkce *_toggleButtonsAndView* načte fragment *Change.fragment.xml* a zobrazí správná tlačítka v zápatí stránky. Použitím funkce *bindAggregation()* jsou naplněny dropdown menu pro výběr nového umístění majetku, podobně jako v *Master.controller.js*. Funkce *handleSavePress* je spuštěna, pokud uživatel potvrdí změny, které provedl v editovatelném formuláři. Z každého dropdown menu musí být manuálně načten textový popis a uložen do nepojmenovaného modelu, aby se pak správně zobrazil ve formuláři *Display*. Také je kontrolováno nastavení umístění uložené v nepojmenovaném modelu s umístěním nastaveným v aplikaci v *Location* modelu. Událost *handleCancelPress* nastane v případě, že uživatel zruší změny provedené ve formuláři *Change*. Všechny změny provedené na modelu s *TwoWay* binding mode jsou shromažďovány v batch group s názvem *changes*, která je nastavena na *deffered* (odloženo). To umožňuje resetovat změny bez odeslání nového requestu na backend, aby se znovu načetly stará data. Metodou *resetChanges()* je možné resetovat-obnovit všechny změny. (SAP SE, c2015) Po smazání všech změn musí být znovu provedena inicializace, jako v metodě *attachRequestCompleted()*. Když uživatel potvrdí majetek, spustí se funkce *onConfirm*. Metoda *submitChanges()* provede odeslání dat uložených v nepojmenovaném modelu na backend. A pro návrat na seznam majetků je použita funkce *onNavBack()*.

5.6.3 Apache Cordova

Apache Cordova je open-source vývojový Framework pro mobilní aplikace. Umožňuje využít standardní webové technologie, jako HTML5, CSS3 a JavaScript pro vývoj cross-platform aplikací a tím se vyhnout vývoji v nativních jazycích jednotlivých mobilních platforem. (APACHE, 2015)

Cordova používá šablony nativních aplikací, mají přes celý displej jednu WebView komponentu, ve které běží dodaná webová aplikace. Do Apache Cordova se vloží webová aplikace a pak je možné generovat nativní aplikace pro mobilní platformy.

Důležitý je systém pluginů, který zpřístupňuje nativní API zařízení (fotoaparát, geolokaci, souborový systém, akcelerometr, a další) do JavaScriptu uvnitř WebView. Silnou stránkou je možnost napsat pluginy vlastní (stačí napsat JavaScriptový wrapper a nativní implementaci pro všechny platformy, které chceme podporovat). (Hujer, 2014)

V našem případě potřebujeme plugin, který spustí fotoaparát zařízení a ve vytvořeném snímku rozpozná čárový kód a jeho obsah předá do naší SAPUI5 aplikace. Takovou funkcionalitu obsahuje PhoneGap Plugin BarcodeScanner, jeho zdro-

kové kódy jsou dostupné na GitHubu³. BarcodeScanner podporuje platformy Android, iOS, Windows Phone a BlackBerry. Umí rozpoznat a dekodovat obsah čárových kódů různých typů, nachází se mezi nimi i typ PDF417, právě tento typ je používán k označování majetků na VUT Brno.

³ <https://github.com/phonegap/phonegap-plugin-barcodescanner>

6 SAPUI5 aplikace nevyužívající SAP zdroje

Tato aplikace je pouze ukázková, má demonstrovat, že SAPUI5 aplikace nemusí jako zdroj dat nutně využívat OData službu implementovanou v ERP systému SAP. Jak již bylo zmíněno v sekci 2.4, pokud na back endu aplikace není využíván nějaký produkt společnosti SAP, musí být použit framework OpenUI5 namísto SAPUI5. Jinak by došlo k porušení licenčních podmínek společnosti SAP.

6.1 Požadavky na aplikaci

Aplikace zobrazí seznam všech zaměstnanců. Bude možné vybrat zaměstnance a zobrazit objednávky, za které je daný zaměstnanec odpovědný. Bude možné zobrazit adresu a poznámky k zaměstnanci.

6.2 Entity Data Model (EDM)

Aplikace využívá veřejně dostupnou OData službu na adrese services.odata.org⁴. Tato služba vystavuje data známé testovací databáze Northwind. Aplikace z celé služby využívá jen dvě entity: Employee a Order. Obsah entit je na obr. 35 v příloze F.

6.3 Návrh GUI

Aplikace je typu dělené okno, jak je vidět na obr. 36 v příloze G. Levá část okna zobrazuje seznam zaměstnanců. V záhlaví okna se nachází vyhledávací pole. V seznamu zaměstnanců je možné vyhledávat podle příjmení. Po výběru zaměstnance se v pravé části okna načte stránka, která v horní části obsahuje jméno zaměstnance a datum od kterého zaměstnanec ve firmě pracuje. Následuje panel se třemi záložkami. Jako výchozí je zvolena záložka Objednávky. Obsahuje tabulku s objednávkami, za které je zaměstnanec odpovědný. Každá objednávka obsahuje datum objednání, id, jméno zákazníka, jméno dopravce, cílovou zemi, město a cenu dopravy v EUR. Záložka Adresa obsahuje kompletní adresu zaměstnance.

⁴ <http://services.odata.org/V2/Northwind/Northwind.svc/>

7 Závěr

V práci bylo provedeno porovnání několika oblíbených JavaScript MVC frameworků a SAPUI5. Ukázalo se, že SAPUI5 přebírá osvědčené vlastnosti těchto frameworků a navíc má oficiální podporu od společnosti SAP, která je světovou jedničkou v poskytování ERP systémů a služeb s tímto souvisejících. Společnost SAP vyvinula SAPUI5 jako nástupce dosavadní, dnes již zastaralé, UI technologie WebDynpro. Lze tedy předpokládat, že SAP bude v budoucnu SAPUI5 podporovat a dále rozvíjet. Zejména kvůli podpoře je nejvhodnější pro vývoj business aplikace nad ERP systémem SAP zvolit SAPUI5.

Dále byl prozkoumán stávající systém FAIN pro inventuru hmotného majetku, který je integrován ERP systému SAP na VUT Brno. Byly identifikovány nedostatky mobilních terminálů, které tento systém využívá. Podařilo se vytvořit aplikaci, která je plnohodnotnou náhradou původních Windows Mobile terminálů systému FAIN. Nejprve bylo potřeba zjistit, jak pracuje systém FAIN na pozadí během procesu inventury majetku. Jaká data se ukládají do kterých struktur v ERP SAP. Na základě těchto poznatků bylo možné navrhnout EDM model OData služby a následně provést její implementaci na serveru SAP NetWeaver Gateway v jazyce ABAP. Nad datovými entitami vystavovanými OData službou pak byla vytvořena SAPUI5 aplikace.

Hlavní výhoda SAPUI5 aplikace oproti předchozímu řešení s Windows Mobile terminály spočívá v okamžitém online zpracování jednotlivých položek inventury. Odpadá tak problém s konzistencí dat a riziko, že dojde k poškození souborů a celá inventura se bude muset dělat znovu. Funkcionalita původního řešení zůstala zachována, chybí jen možnost vytvoření volné poznámky k majetku. Tuto funkci je možné do vytvořené aplikace přidat. Otázkou je, zda má smysl se tím zabývat. Funkce nebyla příliš využívána, během 6 let provozu systému FAIN bylo vytvořeno pouze 1506 volných poznámek. Navíc drtivá většina těchto volných poznámek vznikla zbytečně, protože popisují stavy, které je možné popsat pevnou poznámkou.

Díky online přístupu k business datům je možné v SAPUI5 aplikaci okamžitě přepínat mezi jednotlivými inventurními seznamy, Windows Mobile terminál může pracovat jen s jedním inventurním seznamem, pro přepnutí na jiný inventurní seznam je nutné připojit terminál USB kabelem k PC a v SAP GUI provést export a import dat. Nyní je možné k inventuře majetku používat téměř jakýkoliv smartphone nebo tablet, podmínkou je webový prohlížeč a fotoaparát. Původní řešení vyžaduje drahý specializovaný terminál s dnes již mrtvou platformou Windows Mobile, takové zařízení nemá mimo inventuru další využití. Další nevýhodou je proprietárnost aplikace, přestože VUT Brno má kapacity na to, aby si mohlo dělat požadované změny v aplikaci samo, není to možné, protože zdrojové kódy nejsou k dispozici.

Jedinou nevýhodou SAPUI5 aplikace oproti původnímu terminálu je rychlost snímání majetků. Terminál poskytuje lepší user experience při snímání čárového kódu, nebo RFID tagu. Smartphonu, nebo tabletu trvá delší dobu, než fotoaparát

zaostří čárový kód a provede jeho rozpoznání. Také nutnost během ostření držet zařízení v naprostém klidu je při celodenním snímání majetků unavující. Rychlejší a pohodlnější snímání majetků pomocí smartphonů a tabletů je možné dosáhnout označením majetků NFC tagy. Nebo připojením hardwarové čtečky čárových kódů. Dnes už také existují terminály s aktuální platformou android, na nich je možné používat vytvořenou SAPUI5 aplikaci. Rychlé a pohodlné snímání čárových kódů a RFID tagů zajistí jejich speciální hardware.

Druhá aplikace slouží pouze k demonstraci, že SAPUI5 aplikace nemusí jako zdroj dat nutně využívat OData službu implementovanou v ERP systému SAP. Aplikace zobrazuje data z veřejně dostupné OData služby, která poskytuje data ze známé testovací databáze Northwind.

8 Literatura

- APACHE. *Apache Cordova*. Overview [online]. 2015. [cit. 2015-11-29]. Dostupné z: <http://cordova.apache.org/docs/en/5.4.0/guide/overview/index.html>
- ČADA, O. *Objektové programování: naučte se pravidla objektového myšlení*. 1. vyd. Praha: Grada, 2009, 200 s. Průvodce (Grada). ISBN 978-80-247-2745-5.
- FAIN *Uživatelský manuál - SAP*. FAIN – Fixed Assets Inventory (verze 4.1). 2011. [cit. 2015-11-30].
- HUJER, M. *Cordova a Sencha Touch aneb mobilní aplikace pomocí webových technologií*. Zdroják: o tvorbě webových stránek a aplikací [online]. 2014. [cit. 2015-10-11]. Dostupné z: <https://www.zdrojak.cz/clanky/cordova-sencha-touch-mobilni-aplikace/>
- KUNZ, A. *What is OpenUI5 / SAPUI5 ?*. SAP Community Network [online]. 2013. [cit. 2015-10-15]. Dostupné z: <http://scn.sap.com/community/developer-center/frontend/blog/2013/12/11/what-is-openui5-sapui5>
- OASIS *OASIS Approves OData 4.0 Standards for an Open, Programmable Web*. [online]. 2014. [cit. 2015-11-26]. Dostupné z: <https://www.oasis-open.org/news/pr/oasis-approves-odata-4-0-standards-for-an-open-programmable-web/>
- ODATA *Overview (OData Version 2.0)*. OData - the Best Way to REST [online]. 2015. [cit. 2015-11-27]. Dostupné z: <http://www.odata.org/documentation/odata-version-2-0/overview/#ODataBasics>
- RADHAKRISHNAN, G. *ABAP Performance tips*. Contributor Corner - SCN Wiki [online]. 2015. [cit. 2015-12-4]. Dostupné z: <http://wiki.scn.sap.com/wiki/display/Community/ABAP+Performance+tips/>
- SAP SE. *How to Write an OData Channel Gateway Service. Part 2 - The Runtime Data Provider Class*. SAP NetWeaver How-To Guide [online]. 2011. [cit. 2015-11-29]. Dostupné z: <http://scn.sap.com/docs/DOC-16326/>
- SAP SE. *Tutorials – Walkthrough*. SAPUI5 SDK - Demo Kit [online]. 2015. [cit. 2015-11-29]. Dostupné z: <https://sapui5.netweaver.ondemand.com/#docs/guide/3da5f4be63264db99f2e5b04c5e853db.html>
- SAP SE. *Tutorials – Navigation and Routing*. SAPUI5 SDK - Demo Kit [online]. b2015. [cit. 2015-11-29]. Dostupné z: <https://sapui5.netweaver.ondemand.com/#docs/guide/1b6dc d39a6a74f528b27ddb22f15af0d.html>

- SAP SE. *Essentials – Models*. SAPUI5 SDK - Demo Kit [online]. c2015. [cit. 2015-11-29]. Dostupné z: <https://sapui5.netweaver.ondemand.com/#docs/guide/e1b625940c104b558e52f47afe5ddb4f.html>
- SAP SE. *Application Best Practicess*. UI Development Toolkit for HTML5 (SAPUI5) - SAP Library [online]. 2013. [cit. 2015-11-29]. Dostupné z: http://help.sap.com/saphelp_nw74/helpdata/en/5c/a68e6e62e6464a8103297fbc19cd9c/content.htm?frameset=/en/f6/998380aeb94af6917eff27aa5cc14c/frameset.htm¤t_toc=/en/5c/be4e5b4a19479a92b1d32ff23b7b63/plain.htm&node_id=46&show_children=false/
- STEIGERWALD, D. *JavaScriptové MVC frameworky (reakce na článek Ondřeje Žáry)*. Zdroják: o tvorbě webových stránek a aplikací [online]. 2013. [cit. 2015-10-11]. Dostupné z: <https://www.zdrojak.cz/clanky/javascriptove-mvc-frameworky-reakce-na-clanek-ondreje-zary/>
- VUT_AM *INVENTURA POMOCÍ ČÁROVÝCH KÓDŮ A RFID*. Dokumentace ekonomického systému SAP na VUT v Brně. 2014. [cit. 2015-11-29].
- ŽÁRA, O. *JavaScriptové MVC frameworky*. Zdroják: o tvorbě webových stránek a aplikací [online]. 2013. [cit. 2015-10-11]. Dostupné z: <https://www.zdrojak.cz/clanky/javascriptove-mvc-frameworky/>

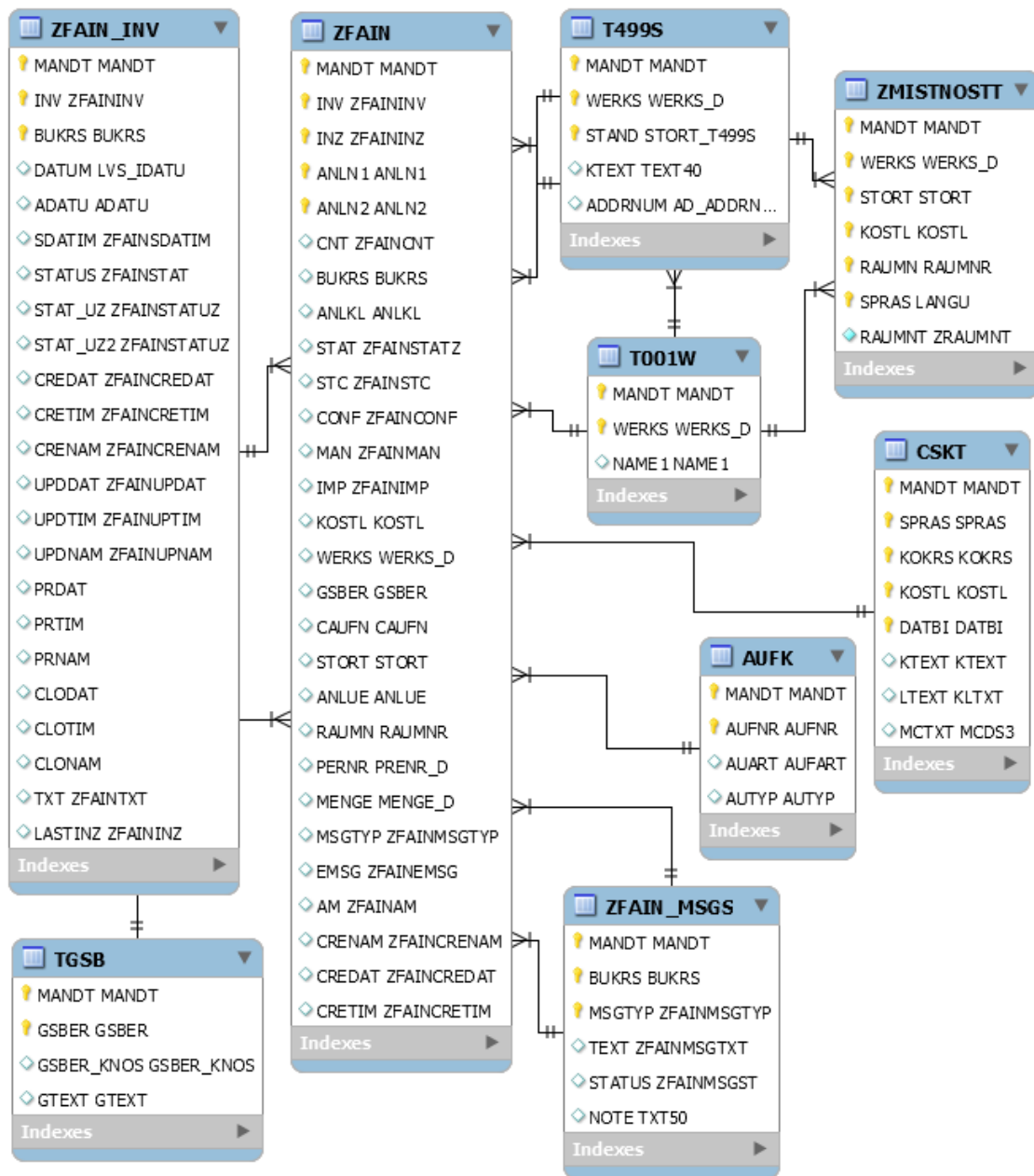
Přílohy

A SAPUI5 Runtime & UI Libraries

Open source software třetíh stran:

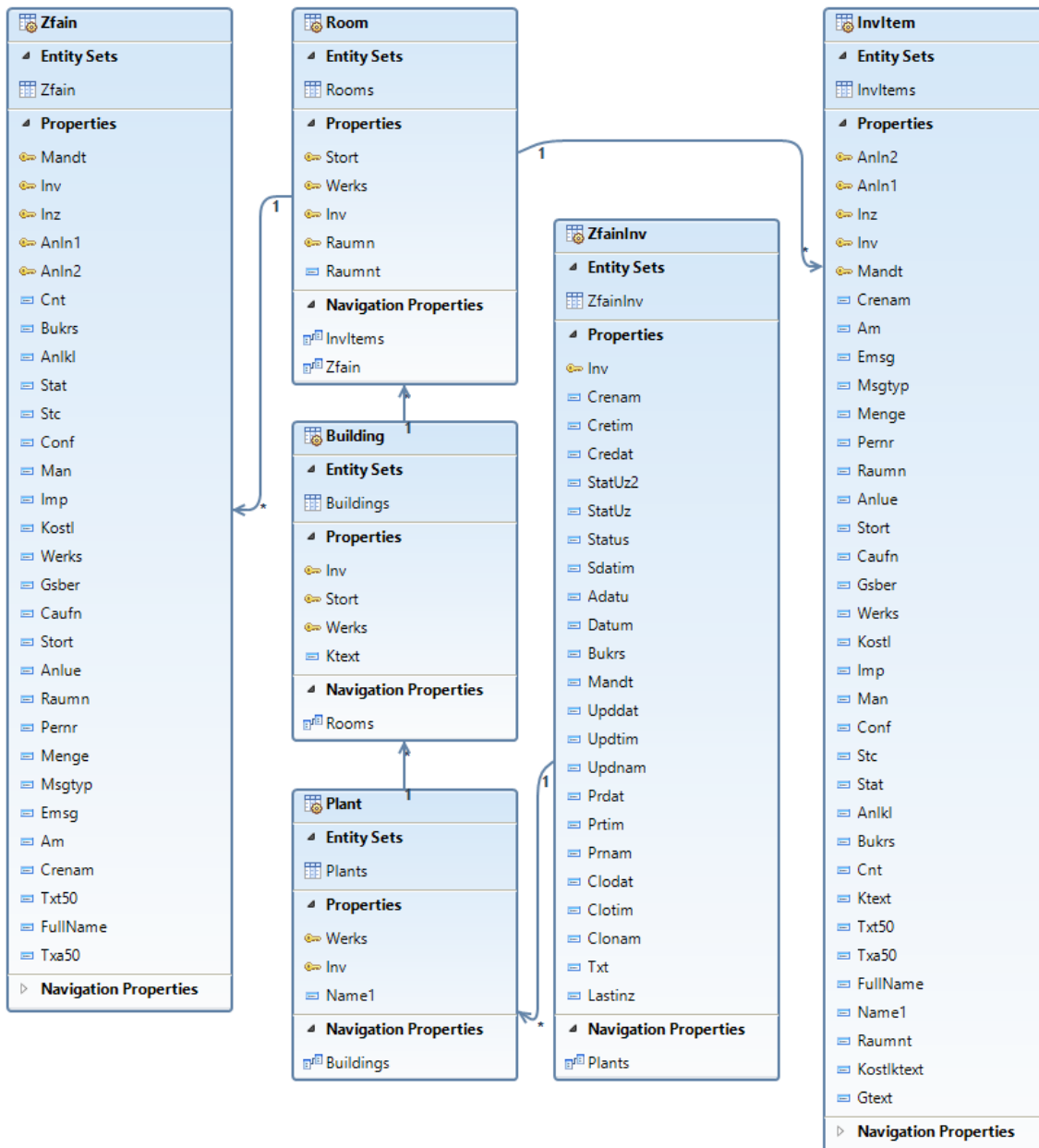
jQuery, QUnit, jQueryUI
jQuery Mobile
Qunit Composite
JUnit Reporter for QUnit
cubiq.org – iscroll
cubiq.org – swipeview
Mobify.js
Zynga Scroller
LESS
DataJS
D3
JSZip
Google-Caja JS HTML Sanitizer
Unicode Common Locale Data Repository
MobiScroll
SinonJS
URI.js
punycode.js
Flexie.js
handlebars
requireJS
crossroads.js
hasher.js
JS-Signals
blanket.js
Array.prototype Polyfills
unorm.js
Top Down Operator Precedence
es6-promise

B ERD využívané části databáze v ERP SAP



Obr. 28 ERD využívané části databáze v ERP SAP

C EDM OData služby pro inventuru majetku



Obr. 29 EDM OData služby pro inventuru majetku

D GUI aplikace pro inventuru majetku

Home

Inventories

Inventory	Status	Created by	Inventory Date	Update	Close Date
test upravy fain 00000083	Active	BASO	20. 5. 2013		
FEKT - stěhování 02/2013 00000082	Active	BASO	15. 2. 2013		
POZ-BUD-ST(2) 2012 00000081	Closed	BASO	31. 8. 2012	15. 11. 2012	18. 2. 2013
ICV(60) 2012 00000080	Closed	BASO	31. 8. 2012	15. 11. 2012	18. 2. 2013
KaM 2012 00000079	Closed	BASO	31. 8. 2012	15. 11. 2012	9. 5. 2013
POZ-BUD-ST 2012 00000078	Canceled	BASO	31. 8. 2012	8. 10. 2012	
RE(72,74,90) 2012 00000077	Closed	BASO	31. 8. 2012	15. 11. 2012	18. 2. 2013

Obr. 30 Výběr inventury

FEKT - stěhování 02/2013

PUR: Purkyňova

01: Purkyňova 464/118

066: Aula - M

100: Chodba-přízemí

103: Chodba-3. poschodí

104: chodba

121: server, PC cluster

126: laboratoř

127: laboratoř

128: LABORATOR

133: depozitář

136: laboratoř

Purkyňova/Purkyňova 464/118/126: laboratoř

Scan

All items

Unprocessed

Processed

HSM ca LUNA PCI-3000 | 000000314270/0000

Server X3050 | 000000314271/0000

IPSec HS500 | 000000314274/0000

Soubor zabezpečovacích systémů -7ks SZS | 000000314278/0000

Datový rozvaděč 800/600 45u | 000000314526/0000

Zátěžový tester | 000000316983/0000

PC | 001000153300/0000

PC | 001000153301/0000

PC | 001000153302/0000

+

Obr. 31 Výběr umístění a seznam majetků ve vybrané místnosti

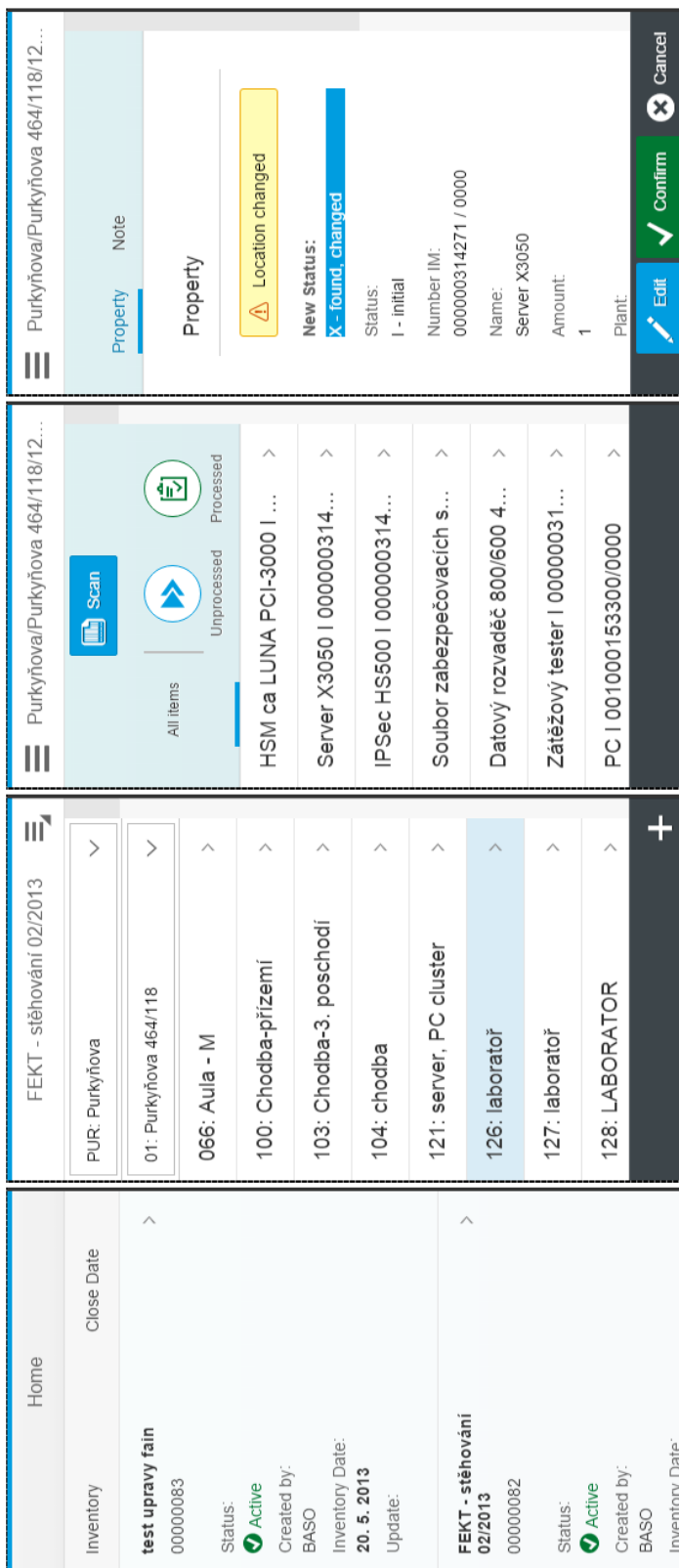
FEKT - stěhování 02/2013	Purkyňova/Purkyňova 464/118/126: laboratoř
PUR: Purkyňova	Property
01: Purkyňova 464/118	Note
066: Aula - M	Property
100: Chodba-přizemí	<p>⚠ Location changed</p> <p>New Status: X - found, changed Status: I - initial Number IM: 000000314271 / 0000 Name: Server X3050 Amount: 1 Plant: PUR: Purkyňova Building: 01: Purkyňova 464/118 New Room: 127: laboratoř Room: 126: laboratoř Cost Cen.: 0000018350: Ú. telekomunikací Person: 01002159: Zeman Václav doc. Ing. Ph.D. Work area: 18: Fakulta elektr. a kom.technol.</p>
103: Chodba-3. poschodí	
104: chodba	
121: server, PC cluster	
126: laboratoř	
127: laboratoř	
128: LABORATOR	
133: depozitář	
136: laboratoř	
Edit Confirm Cancel	

Obr. 32 Detail majetku

FEKT - stěhování 02/2013		Purkyňova/Purkyňova 464/118/126: laboratoř	
PUR: Purkyňova	>	Status: X - found, changed	>
01: Purkyňova 464/118	>	New: X - found, changed	>
066: Aula - M	>	Number IM: 000000314271 / 0000	>
100: Chodba-přízemí	>	Name: Server X3050	>
103: Chodba-3. poschodí	>	Amount: 1	>
104: chodba	>	Plant: PUR: Purkyňova	>
121: server, PC cluster	>	New Plant: PUR: Purkyňova	>
126: laboratoř	>	Building: 01: Purkyňova 464/118	>
127: laboratoř	>	New Building: 01: Purkyňova 464/118	>
128: LABORATOR	>	Room: 126: laboratoř	>
133: depozitář	>	New Room: 127: laboratoř	>
136: laboratoř	>		>

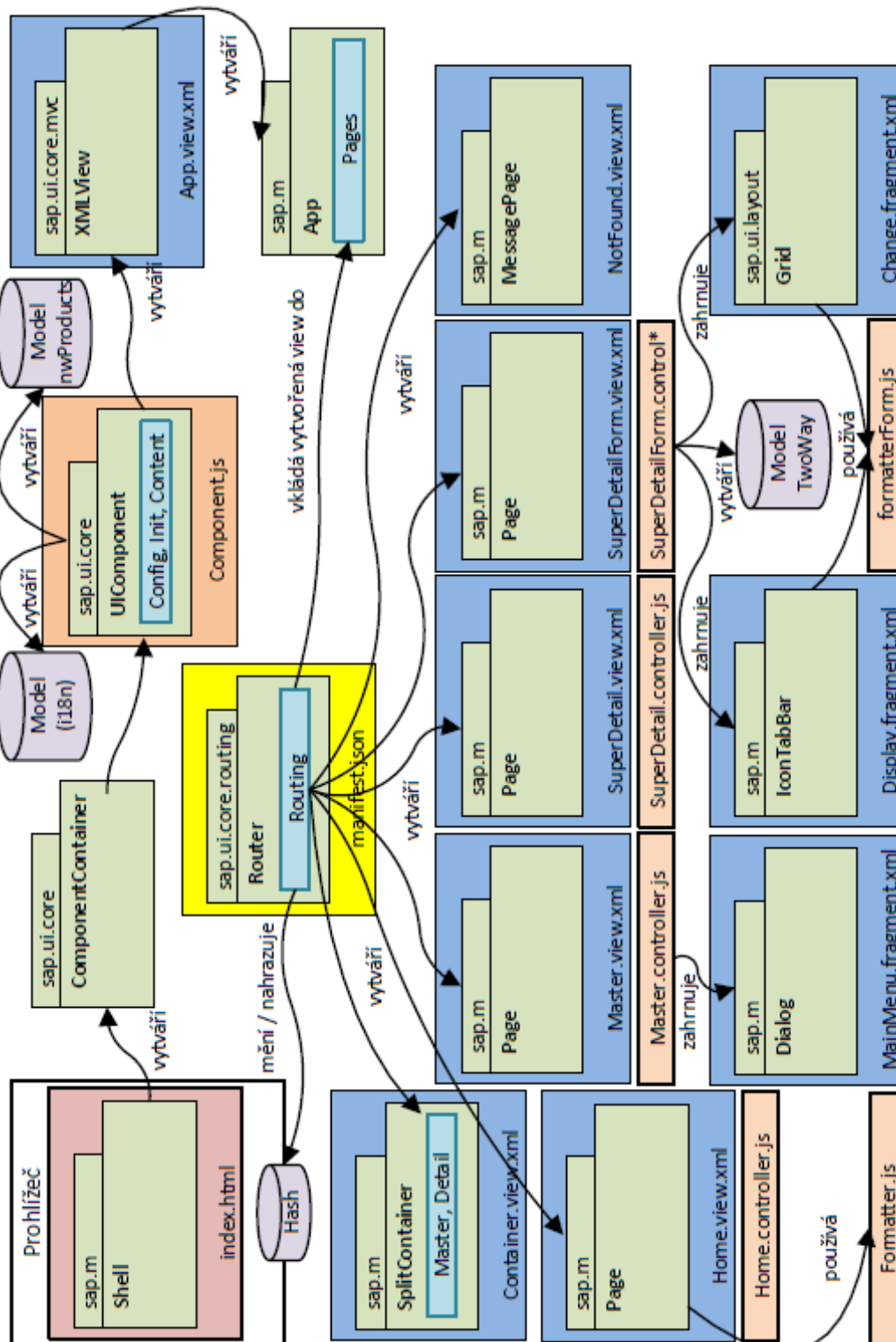
Save Cancel

Obr. 33 Editace majetku

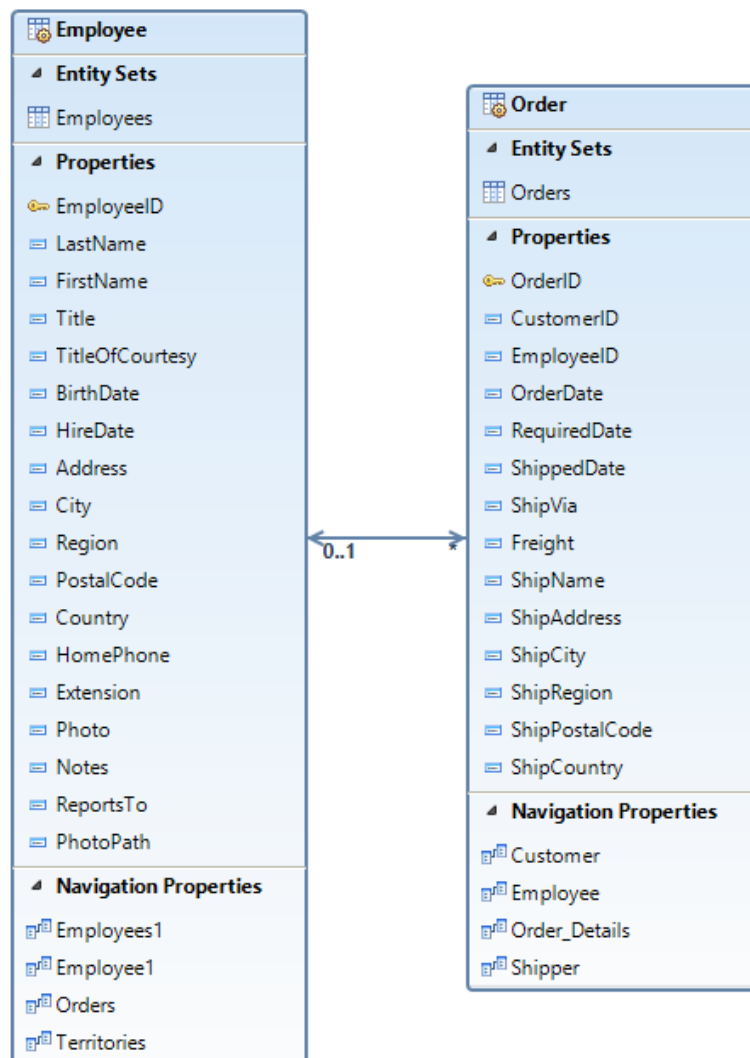


Obr. 34 Aplikace na mobilním telefonu Apple iPhone 5

E Schéma aplikace pro inventuru majetku



F Využívaná část EDM OData služby Northwind



Obr. 35 Využívaná část EDM OData služby Northwind

G GUI aplikace pro zobrazení zaměstnanců

The screenshot displays a web application interface for managing employees. It is divided into two main panels: 'Employees' and 'Employee'.

Employees Panel:

- Search bar: Hledání
- List of all employees
- Employee list: Nancy Davolio, Andrew Fuller, Janet Leverling (selected), Margaret Peacock, Steven Buchanan, Michael Suyama, Robert King, Laura Callahan.

Employee Panel:

Employee: Janet Leverling

Employee hire date: 1. 4. 1992

Employee ID: 3

Navigation tabs: Orders, Address, Notes

Orders Table:

OrderDate, ID	Customer	ShipName	Destination	Freight
8. 7. 1996 10251	VICTE	Victuailles en stock	France > Lyon	41,34 EUR
10. 7. 1996 10253	HANAR	Hanari Carnes	Brazil > Rio de Janeiro	58,17 EUR
15. 7. 1996 10256	WELLI	Wellington Importadora	Brazil > Resende	13,97 EUR
26. 7. 1996 10266	WARTH	Wartian Herkku	Finland > Oulu	25,73 EUR

Obr. 36 Aplikace pro zobrazení zaměstnanců

H Obsah přiloženého CD

CD přiložené k této práci obsahuje následující adresáře:

- **aplikace** – zdrojové kódy aplikací
- **odata** – zdrojové kódy implementace OData služby
- **text** – diplomová práce ve formátu PDF