

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Aplikace pro správu majetku a aktiv organizace

Bc. Lukáš Hervert

Diplomová práce
2018

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2017/2018

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Lukáš Hervert**
Osobní číslo: **I16216**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Aplikace pro správu majetku a aktiv organizace**
Zadávající katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je přiblížit možnosti vývoje aplikací pro správu majetku a aktiv organizace se zaměřením na využití open source portálových řešení.

V teoretické části bude přiblížena problematika a specifika těchto aplikací a představen vývoj a možnosti tvorby portálových řešení. V praktické části bude vytvořena funkční aplikace pro správu majetku a aktiv organizace. Praktická část bude obsahovat způsob, jakým se realizuje vývoj aplikací tohoto typu, analýzu procesů, které budou znázorněny v procesním BPMN diagramu s cílem dosažení maximální bezpečnosti aplikace, popis použitých technologií a popis open source portálového řešení v případě jeho využití pro tvorbu aplikace.

Rozsah grafických prací: 10
Rozsah pracovní zprávy: 60
Forma zpracování diplomové práce: tištěná
Seznam odborné literatury:

JONAS X. Yuan. Liferay Portal Systems Development. Birmingham: Packt Publishing Limited. 515 pages. ISBN: 9781849515986. EAN: 9781849515986.
CHRIS Lloyd. Asset Management. London: ICE Publishing. 268 pages. ISBN: 0727736531. EAN: 9780727736536.
Liferay: Liferay Digital Experience Platform [online]. Liferay: 2017 [cit. 03. 10. 2017]. Dostupné z: <https://www.liferay.com/digital-experience-platform>
Morosystems: Liferay Portál [online]. Morosystems: 2017 [cit. 03. 10. 2017]. Dostupné z: <http://www.morosystems.cz/produkty/liferay-portal/>

Vedoucí diplomové práce: Ing. Miloslav Macháček, Ph.D.
Katedra informačních technologií

Datum zadání diplomové práce: 30. října 2017
Termín odevzdání diplomové práce: 18. května 2018



Ing. Zdeněk Němec, Ph.D.
děkan



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2017

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 5. 2018

Lukáš Hervert

Poděkování

Chtěl bych poděkovat panu Ing. Miloslavu Macháčkovi Ph.D. za trpělivost a ochotu, kterou mi v průběhu zpracování mé diplomové práce věnoval. Dále bych chtěl poděkovat svým rodičům za zázemí a podporu jak během vypracovávání práce tak, během celého studia.

ANOTACE

Tato práce se zabývá vytvořením portálového řešení a aplikace pro správu aktiv a majetku organizace. Jako portálové řešení je zvolena open-source technologie Liferay a aplikace je vytvořena pomocí portletů. V teoretické části jsou vysvětleny základní pojmy, představeny dostupné portály a nastíněn způsob vývoje pro Liferay Portal. Praktická část se věnuje použitým technologiím a implementaci, úpravám a konfiguraci Liferay portálu a aplikace.

KLÍČOVÁ SLOVA

Portál, Portlet, Asset, Management, Java, Liferay, Vaadin, LDAP

TITLE

Asset Management Application

ANNOTATION

This thesis deals with the creation of portal solution and application for Asset Management. Open-source Liferay technology is chosen as a portal solution. The Asset Management application is created using portlets. The theoretical part explains the basic concepts, the available portal solutions and outlines the way of development for Liferay Portal. The practical part deals with used technologies and implementation, modification and configuration of the Liferay portal and the Asset Management application.

KEYWORDS

Portal, Portlet, Asset, Management, Java, Liferay, Vaadin, LDAP

OBSAH

Seznam obrázků	9
Seznam zkratk	10
Úvod	12
1 Teoretická část	13
1.1 Portál	13
1.2 Portlet	14
1.2.1 Životní cyklus	16
1.2.2 Módy	17
1.3 Liferay Portal	18
1.3.1 Enterprise CMS	20
1.3.2 Spolupráce a sociální funkce	21
1.4 Vývoj pro Liferay Portal	22
1.4.1 Portlet	23
1.4.2 Service Builder Portlet	23
1.4.3 Hooks	24
1.4.4 Layout Template	24
1.4.5 Theme	25
1.4.6 Ext	28
1.5 Další portálová řešení	28
1.5.1 IBM WebSphere Portal	30
1.5.2 Microsoft SharePoint	30
1.5.3 Oracle WebCenter Portal	31
1.5.4 SAP Enterprise Portal	32
1.6 Správa majetku a aktiv organizace	33
1.7 Aplikace pro správu majetku a aktiv organizace	36
2 Praktická část	37
2.1 Technologie	37
2.1.1 Vaadin	37
2.1.2 Maven	39
2.1.3 Apache Tomcat	39

2.1.4	LDAP (Lightweight Directory Access Protocol).....	40
2.2	Databáze.....	40
2.3	BPMN	42
2.4	Aplikace	43
2.4.1	Konfigurační soubory v projektu	43
2.4.2	Struktura projektu	45
2.4.3	Hook.....	47
2.5	Instalační příručka.....	47
2.5.1	Instalace a konfigurace OpenLDAP a phpLDAPAdmin	47
2.5.2	Instalace databáze	51
2.5.3	Instalace Liferay portálu	51
2.5.4	Zabalení Asset Management projektu	55
2.6	Konfigurace	55
2.6.1	Data Source	55
2.6.2	LDAP server	56
2.6.3	Liferay Portál	59
2.6.4	Uživatelské skupiny a role.....	63
2.6.5	Ostatní konfigurace.....	64
	Závěr	65
	Použitá literatura	66
	Přílohy.....	70

SEZNAM OBRÁZKŮ

Obrázek 1 - Portálová stránka. Zdroj: [3]	14
Obrázek 2 - Části portletu. Zdroj: [5]	15
Obrázek 3 - Layout Template Editor. Zdroj: [18]	25
Obrázek 4 - Struktura Theme pluginu. Zdroj: [19]	27
Obrázek 5 - Gartner kvadranty. Zdroj: [21]	29
Obrázek 6 - Oracle WebCenter. Zdroj: [26]	32
Obrázek 7 - SAP Enterprise Portal. Zdroj: [28]	32
Obrázek 8 - Příklad životního cyklu aktiva. Zdroj: [30]	34
Obrázek 9 - Vaadin architektura. Zdroj: [35]	38
Obrázek 10 - Databázový model. Zdroj: Vlastní	41
Obrázek 11 - BPMN Úprava konfigurace. Zdroj: Vlastní	42
Obrázek 12 - Přidání nebo úprava assetu. Zdroj: Vlastní	42
Obrázek 13 - Prohlížení assetů. Zdroj: Vlastní	43
Obrázek 14 - liferay-display.xml. Zdroj: Vlastní	44
Obrázek 15 - web.xml. Zdroj: Vlastní	45
Obrázek 16 - Hook tag. Zdroj: Vlastní	47
Obrázek 17 - Přihlašovací stránka LDAP serveru. Zdroj: Vlastní	50
Obrázek 18 - Hlavní stránka LDAP serveru. Zdroj: Vlastní	51
Obrázek 19 - Základní nastavení Liferay portálu. Zdroj: Vlastní	53
Obrázek 20 - Nastavení Liferay databáze. Zdroj: Vlastní	54
Obrázek 21 - Úspěšná konfigurace Liferay. Zdroj: Vlastní	55
Obrázek 22 - Šablona uživatele. Zdroj: Vlastní	58
Obrázek 23 - Výsledná stromová struktura. Zdroj: Vlastní	59
Obrázek 24 - Jméno serveru a výchozí hodnoty. Zdroj: Vlastní	60
Obrázek 25 – Připojení. Zdroj: Vlastní	61
Obrázek 26 – Uživatelé. Zdroj: Vlastní	61
Obrázek 27 - Namapování uživatelů. Zdroj: Vlastní	62
Obrázek 28 - Namapování skupin. Zdroj: Vlastní	63
Obrázek 29 - Přidání obsahu. Zdroj: Vlastní	64

SEZNAM ZKRATEK

BPMN	Business Process Model and Notation
LDAP	Lightweight Directory Access Protocol
CMS	Content Management System
JSR	Java Specification Request
URL	Uniform Resource Locator
IPC	Inter-Process Communication
AJAX	Asynchronous JavaScript and XML
JSON	JavaScript Object Notation
XML	Extensible Markup Language
JEE	Java Platform, Enterprise Edition
JSF	JavaServer Faces
WSRP	Web Services for Remote Portlets
GNU	GNU's Not Unix
SOAP	Simple Object Access Protocol
REST	Representational State Transfer
RSS	Rich Site Summary
API	Application Programming Interface
SSO	Single Sign-On
SOA	Service Oriented Architecture
SEO	Search Engine Optimization
IMAP	Internet Message Access Protocol
SMS	Short Message Service
IDE	Integrated Development Environment
SDK	Software Development Kit
ORM	Object-Relational Mapping
SQL	Structured Query Language
JSP	JavaServer Pages
JSPF	JavaServer Page Fragment
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
IBM	International Business Machines Corporation
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung

EAM	Enterprise Asset Management
DAM	Digital Asset Management
SAM	Software Asset Management
ITAM	Information Technology Asset Management
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
UI	User Interface
AWT	Abstract Window Toolkit
SWT	Standard Widget Toolkit
RIA	Rich Internet Applications
GWT	Google Web Toolkit
DAP	Directory Access Protocol
BSD	Berkeley Software Distribution
JPA	Java Persistence API
WAR	Web Application Archive
JNDI	Java Naming and Directory Interface
ID	Identifier
PHP	Hypertext Preprocessor
DNS	Domain Name System
MDB	Memory-Mapped Database
DN	Domain Name
UUID	Universally Unique Identifier

ÚVOD

Cílem této diplomové práce je přiblížit možnosti vývoje aplikace pro správu majetku a aktiv organizace se zaměřením na využití open-source portálových řešení. Celá práce je postavena na dlouhodobých osobních zkušenostech s vývojem portálových řešení a portletů na míru.

Navržená aplikace a portálové řešení má za cíl usnadnění integrace do struktury organizace. Portálové řešení slouží v tomto případě jako intranet organizace, do kterého jsou zasazeny portlety s aplikací pro správu majetku a aktiv organizace. Aplikace využívá uživatele z portálového řešení a portálové řešení ověřuje uživatele na adresářovém serveru, tím je zajištěna integrace do struktury organizace.

K realizaci práce je vybrán Liferay Portal jako open-source portálové řešení, OpenLDAP jako open-source implementace protokolu LDAP pro přístup a ukládání dat na adresářový server, programovací jazyk Java a Vaadin framework jako technologie pro vývoj portletů. Veškerá data jsou ukládána do Oracle databáze.

Samotná aplikace je navržena tak, aby byla plně konfigurovatelná. Po nasazení do portálu je v administrační části možné nastavit veškeré atributy aplikace podle potřeb organizace. Lze tedy nastavit například pobočky organizace, životní cyklus assetu nebo typy assetů.

V teoretické části práce je pozornost věnována představení základní koncepce portálu, představení portletů včetně kompletního životního cyklu a módů. Dále je představen Liferay Portal jako vybrané portálové řešení a popsány jeho funkce a vlastnosti včetně Enterprise CMS a možností sociálních funkcí. Pro představu jsou dále popsána některá další dostupná portálová řešení, která jsou konkurenty Liferay portálu. V kapitole nazvané Vývoj pro Liferay Portal jsou nastíněny dostupné pluginy, kterými je realizován veškerý vývoj pro Liferay Portal. Konec teoretické části, která slouží zároveň jako úvod do praktické části, je věnován tématu správy majetku a aktiv organizace a samotné aplikaci na toto téma.

V praktické části jsou do detailu vysvětleny technické záležitosti. Na začátek jsou popsány důležité technologie, které jsou použity pro vývoj. Nechybí samozřejmě vyobrazení modelu databáze nebo BPMN diagramy. Další kapitola je věnována aplikaci a popisu konfiguračních souborů, struktury projektu a vylíčení, k čemu v práci slouží Hook plugin. Celá praktická část je zakončena instalační příručkou a konfigurací, které stručně popisuje kroky, které byly prováděny při přípravě infrastruktury.

1 TEORETICKÁ ČÁST

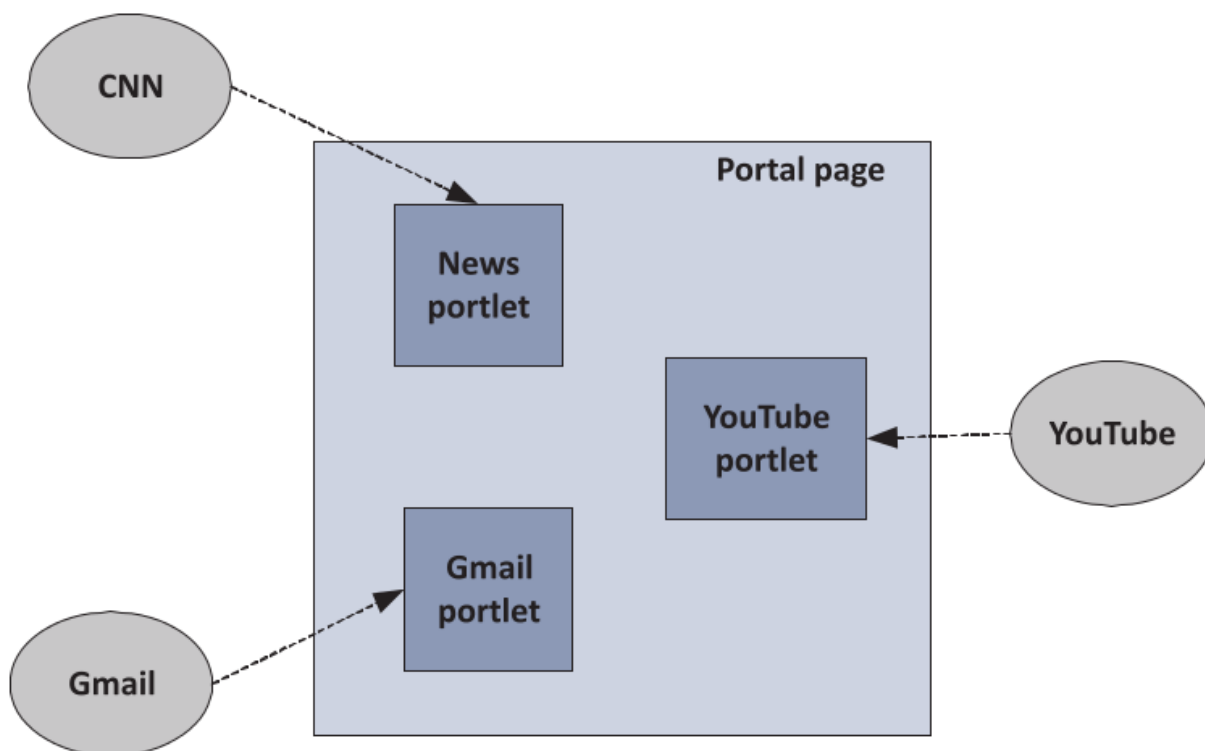
1.1 Portál

Aby bylo možné v další kapitole popsat portlety, je nutné v první řadě porozumět tomu, co je to vlastně portál. Portál není pro uživatele Internetu vůbec nic nového, ale sestavit jeho definici není jednoduché.

Portál je kolekce malých webových aplikací, nazývaných portlety. Portál dále podporuje vlastnosti, jako jsou personalizace, agregace obsahu, autentizace a přizpůsobení. Portlety se v portálu chovají jako okenní webové aplikace a každé okno na webové stránce portálu (nazývané portálová stránka) představuje portlet. [1]

Funkce portálu lze tedy rozdělit na tři hlavní části:

- Portlet kontejner – Portlet kontejner je podobný servlet kontejneru. Každý portlet běží uvnitř portlet kontejneru, který kontroluje celý jeho životní cyklus, poskytuje mu nezbytné prostředky a informace o prostředí. Portlet kontejner je také zodpovědný za inicializaci a zrušení portletů, za posílání požadavků od uživatelů a také za sbírání odpovědí. [2]
- Agregátor obsahu – Jak už bylo zmíněno v definici, jedna z hlavních vlastností portálu je agregování obsahu z různých zdrojů. [2]
- Obecné služby – Jednou z hlavních předností portálu je sada obecných služeb. Služby nejsou součástí specifikace portletu, ale komerční portálové řešení poskytují bohaté sady služeb, aby se odlišili od konkurence. Dvě nejtypičtější služby jsou:
 - Autentizace – Využívá se tzv. „Single sign-on“ přihlašování. To dovoluje přístup ke všem aplikacím po přihlášení do portálu, takže není nutné se přihlašovat ke každé aplikaci zvlášť. [2]
 - Personalizace – Personalizace je v základu možná na dvou úrovních. První je taková, že uživatel portálu se může rozhodnout, jaké chce barvy prostředí nebo jaký styl ikon pro ovládací prvky. Druhá úroveň se týká samotného obsahu, kdy si uživatel může zvolit, jaký obsah uvidí na stránce. Takže pokud je uživatel sportovní fanoušek, tak si na stránku zvolí obsah se sportovní tematikou a naopak odebere například obsah s politickou tematikou. [2]



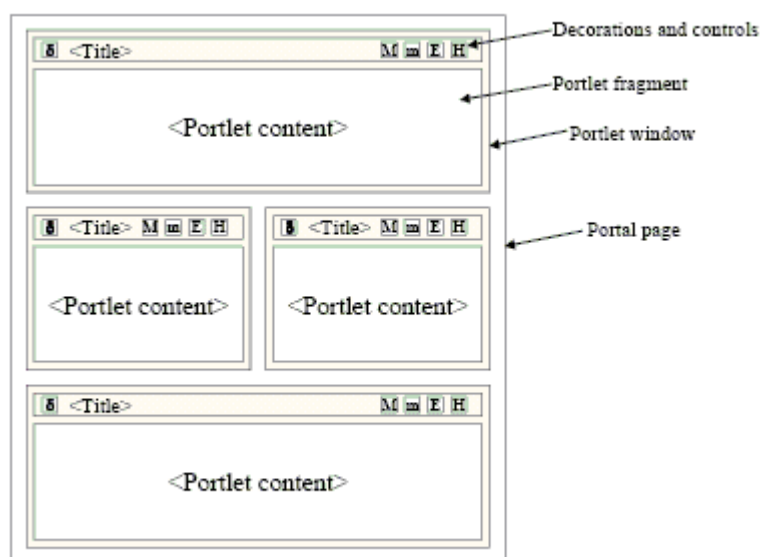
Obrázek 1 - Portálová stránka. Zdroj: [3]

1.2 Portlet

Aby byl portlet vysvětlen zcela správně, tak je možné najít jeho přesnou definici přímo ve specifikaci JSR 186.

"Portlets are web components – like servlets – specifically designed to be aggregated in the context of a composite page. Usually, many portlets are invoked to in the single request of a portal page." [4]

Zjednodušeně řečeno, portlet je webová komponenta (stejně jako servlet) navržena tak, aby byla zasazena do stránky. Portletů může být na jedné stránce více a tedy je možné, že je více portletů vyvoláno jedním požadavkem portálové stránky. [5]



Obrázek 2 - Části portletu. Zdroj: [5]

Každá portálová stránka je složena z jednoho nebo více portletových oken. Každé portletové okno je tvořeno dvěma částmi. První část je dekorační, která se skládá z titulkového pásu, ovládacích prvků a ohraničení portletu. Druhá část je fragment portletu, kde je vykreslována samotná portletová aplikace. [5]

Jak už bylo zmíněno, mezi servlety a portlety je velká podobnost, například:

- Portlety i servlety jsou řízené speciálním kontejnerem.
- Portlety i servlety generují dynamický obsah.
- Portlety i servlety komunikují s webovým klientem přes požadavky a odpovědi. [3] [5]

Ale také se portlety v několika věcech liší:

- Portlety generují pouze fragmenty stránek a ne celé dokumenty.
- Portlety nelze přímo adresovat pomocí URL, nelze tedy nikomu poslat přesnou URL portletu. Je možné pouze poslat URL stránky, na které je portlet obsažen.
- Portlety nemohou generovat libovolný obsah z důvodu, že obsah vygenerovaný portletem bude část portálové stránky. Takže když portálový server bude žádat o obsah ve formátu text/html, tak všechny portlety musí vygenerovat obsah ve formátu text/html. [3] [5]

1.2.1 Životní cyklus

Portlet obsahuje metody:

- init,
- render,
- processAction,
- processEvent,
- serveResource
- a destroy. [6]

Metody reprezentují tzv. portletové fáze. Portletové fáze se liší v závislosti na specifikaci. Specifikace JSR 168 je nejstarší a byla uvedena v roce 2003, specifikace JSR 286 je z roku 2008 a přidává k portletu dvě nové fáze. Jednotlivé fáze portletu jsou tedy:

- init phase (JSR 168),
- destroy phase (JSR 168),
- render phase (JSR 168),
- action phase (JSR 168),
- event phase (JSR 286)
- a resource serving phase (JSR 286). [6]

Metoda init je určena k inicializaci portletu a je volána pouze jednou za celý životní cyklus portletu. Portletový kontejner zavolá metodu init ve chvíli, kdy je portlet načtený a je vytvořena jeho instance. Metoda umožní portletu inicializovat vše potřebné ještě než začne zpracovávat požadavky. V této metodě se obvykle inicializují zdroje jako databázové spojení nebo čtení konfiguračních dat z externího souboru. [3] [6] [7] [8]

Metoda destroy je vyvolána portletovým kontejnerem před odebráním instance portletu z paměti. V této metodě má portlet možnost uvolnit všechny prostředky, které využíval, nebo uložit svůj stav například do databáze nebo do souboru. [3] [6] [7] [8]

Render metoda je vyvolána portletovým kontejnerem, když přijme renderovací požadavek. Metoda má za úkol generovat samotný obsah portletu. [3] [6] [7] [8]

Metoda `processAction` je vyvolána v reakci na přijetí action požadavku. Metoda reaguje na uživatelské akce, které vyústí ve změnu stavu, tedy například odesláním formuláře z klientské části. [3] [6] [7] [8]

Metoda `processEvent`, která byla přidána až ve specifikaci JSR 286 má důležitou roli v implementaci meziportletové komunikaci (IPC). [3] [6] [7] [8]

Další metoda, která byla přidána v JSR 286 je `serveResource`. Tato metoda dovoluje portletu generovat dynamický obsah na základě stavu. Tato metoda je hojně využívána pro AJAX volání a generování obsahu, který je například ve formátu JSON, XML a podobně. [3] [6] [7] [8]

Dále v roce 2017 byla vydána další verze specifikace s označením JSR 362, která se zatím ve velké míře nepoužívá a většina portletů je stále vyvíjena ve verzi JSR 286. JSR 362 ovšem přináší několik vylepšení:

- Podporu JEE 7 vlastností v portletech.
- Sdílení zdrojů mezi portlety.
- Lepší podpora na straně klienta.
- Lepší podpora pro mobilní zařízení.
- Optimalizace Java Server Faces (JSF).
- Podpora pro budoucí WSRP specifikaci.
- Podpora Web Socketů.
- Podpora Open Social Standards.
- Vylepšená komunikace založená na událostech portletu. [9]

1.2.2 Módy

Mód portletu určuje, jaký obsah má samotný portlet generovat. Pro každý mód, se tedy může generovat a zobrazovat jiný fragment. Instruovat portlet, který mód má zrovna zobrazovat je práce portálu. Módy mohou být také měněny v kódu v action fázi portletu. [10]

K dosažení kompatibility, napříč různými portály, definuje specifikace JSR 168 následující módy, které jsou k dispozici v každém portálovém řešení a těmi jsou:

- `view`,

- edit
- a help. [10]

Z těch tří módů je pouze view povinný a vždy ho musí portlet implementovat. Zbylé dva jsou nepovinné. [10]

Ovšem portálové řešení Liferay si přidává několik dalších módů, které je možné využívat v portletech, těmi jsou:

- about,
- config,
- edit default,
- edit guest,
- print
- a preview. [10]

Všechny tyto módy jsou nepovinné a některé mohou být omezeny na konkrétní uživatelskou roli v Liferay portálu. Například mód edit je dostupný pouze pro roli Portal/Site Administrator. [10]

1.3 Liferay Portal

Liferay Portal je podnikový portál vyvinutý v jazyce Java a vydávaný pod licencí GNU Lesser General Public Licence. [11]

Liferay Portal vznikl za účelem poskytnutí portálového řešení neziskovým organizacím již v roce 2000. Postupná rostoucí poptávka zajistila, že v roce 2004 byla založena v americkém Los Angeles společnost Liferay Inc., která pokračovala v rozšiřování Liferay portálu. V dnešní době má Liferay Inc. pobočky kromě USA ještě v Evropě a také Asii. [11]

Liferay je možné pořídit ve dvou různých edicích. První edice je Liferay Portal Community Edition, která je zdarma a open-source, ale není možné k ní pořídit oficiální podporu. Druhá edice je Liferay Portal Enterprise Edition, která je placená. Tato edice je vhodná pro organizace, které potřebují dlouhodobou oficiální podporu přímo od společnosti Liferay Inc. a jejich partnerů. [11]

Liferay má obrovské množství funkcí, které je možné využívat v obou edicích. Mezi funkce patří například:

- Zjednodušený vývoj uživatelského rozhraní – Liferay Portal zjednodušuje vývoj interních i externích webových stránek. Zejména těch, které umožňují uživatelům přihlašovat se k personalizovaným službám nebo pohledům a těm, které vyžadují proces schvalování pro aktualizaci obsahu a integraci nebo agregování více stávajících služeb. Liferay Portal poskytuje jednotnou prezentační vrstvu pro integraci všech podnikových systémů do jediného snadno použitelného rozhraní pro koncové uživatele.
- Flexibilní podniková integrace – Liferay Portal je platforma, která uživatelům, správcům a vývojářům umožňuje integrovat obsah a služby z backendových aplikací. Liferay podporuje několik metod pro integraci, včetně SOAP, REST, RSS, stejně jako proprietární API rozhraní.
- Out-of-the-box nástroje – Liferay Portal poskytuje více nástrojů, které jsou hned k použití, než jiné portálové řešení. K dispozici je více než 60 nástrojů jako jsou Liferay CMS a Liferay Collaboration, které nabízejí publikování obsahu na webu, správu obsahu, spolupráci a sociální síť.
- Secure Single Sign On (SSO) – Liferay Portal dokáže všechny různé systémy zpřístupnit tím, že se přihlásíte jen jednou prostřednictvím zabezpečeného SSO.
- Vlastní pole – Administrátoři mohou přidávat vlastní pole přímo z Liferay portálu bez změny databáze. Takže v případě, že je potřeba extra pole u uživatele, tak je možné ho jednoduše přidat bez úprav v kódu nebo v databázi. Přidávání polí je dostupné u stránek, webového obsahu, blogů, dokumentů, obrázků, záložek, zpráv ve fóru, událostí v kalendáři, stránek ve wiki, uživatelů, uživatelských skupin, organizací a dalších.
- Uživatelské skupiny, organizace a stránky – Uživatelé mohou být intuitivně seskupováni do organizací nebo uživatelských skupin, které jsou napříč organizacemi. To poskytuje flexibilitu a jednoduchou administraci.
- Service Oriented Architecture (SOA) – Liferay Portal je vyvinutý s použitím SOA strategie, která umožňuje integrovat podnikové aplikace.
- Zobrazování obsahu na základě rolí – Portál umožňuje několika různým uživatelům přistoupit na stejnou URL a každému zobrazit unikátní stránku v závislosti na jeho roli, skupině, organizaci nebo osobních preferencích. Liferay poskytuje centrální platformu

pro určování obsahu, včetně toho, kdo může upravovat a publikovat obsah, soubory a aplikace.

- Workflow framework – Liferay umožňuje začlenit workflow do jakéhokoliv vlastního portletu přes workflow API.
- Audit a sledování výkonu – Administrátorům pomáhá při monitorování výkonu portálu, že mají přístup ke klíčovým statistikám pro každý portlet a portálovou stránku. Portálový audit umožňuje administrátorům sledovat a spravovat uživatelské aktivity v portálu.
- Podpora jazyků – Liferay Portal je přeložen do více než třiceti jazyků.
- A mnoho dalších funkcí. [11]

1.3.1 Enterprise CMS

Liferay Enterprise CMS slouží ke správě, integraci a publikování obsahu. Lze využít následujících funkcí:

- Publikování na webu – CMS zahrnuje kompletní správu webového obsahu. Sofistikované funkce umožňují uživatelům používat předem definované šablony, naplánovat publikování či odebírání obsahu a určit, jakým způsobem by ostatní měli obsah před publikací posoudit.
- Připojení více repozitářů z obsahem – Je možné využít kromě základního Liferay repozitáře i jiné repozitáře s obsahem jako například Alfresco nebo Documentum.
- Asset Publisher – Toto je portlet, který je možné vložit na jakoukoliv stránku v portálu a nakonfigurovat tak, že bude zobrazovat veškerá data v portálu. Takže ve chvíli, kdy budou přidána nová data do portálu, Asset Publisher portlet je zobrazí.
- Struktury a šablony – Liferay Portal umožňuje uživatelům snadno vytvářet znovupoužitelné šablony pro webové stránky. To umožňuje uživatelům rychle vytvářet stránky a je možné tímto zajistit, že webové stránky budou mít stejný vzhled na celém webu.
- Integrace s Microsoft Office – Uložiště dokumentů v Liferay portálu je vytvořeno, aby bylo možné využívat SharePoint protokolů a uživatelé tak mohli pracovat s Microsoft Office soubory. Soubory umístěné v lokálním uložení lze automaticky nahrát do cloudu v Liferay portálu.

- Konverze a nahrávání více dokumentů – Uživatelé mohou nahrávat více souborů naráz a také je možné nahrát dokument v jednom formátu a umožnit stažení v jiném formátu.
- Vícehodnotové vyhledávání – Vyhledávání v Liferay portálu umožňuje uživatelům vyhledávat ve všech částech, ke kterým mají přístup, a současně omezuje vyhledávání v částech, kam přístup nemají. Administrátoři si také mohou vybrat, zda využijí k hledání Amazon OpenSearch nebo SOLR, ale mohou být implementovány i jiné vyhledávače.
- Search Engine Optimization (SEO) – Zatímco interní vyhledávač umožňuje vyhledávat data přímo v portálu, Liferay také optimalizuje aktualizace informací v mapě stránek a umožňuje tak, aby nové stránky byly okamžitě vyhledatelné externími vyhledávači.
- Jednoduchá integrace – Webový obsah může být vytvořen ručně přímo v portálu nebo může pocházet z externího zdroje, který je integrován do Liferay portálu. Jednoduše lze obsah integrovat například pomocí REST služeb nebo RSS.
- A mnoho dalších funkcí. [12]

1.3.2 Spolupráce a sociální funkce

Další částí Liferay portálu je část spolupráce a sociálních sítí. I v tomto ohledu poskytuje Liferay velké množství funkcí:

- Wiki – Pro dokumentování důležitých nebo zajímavých informací je k dispozici Wiki. Každá komunita má vlastní Wiki s vlastní sadou oprávnění a kdokoli s právy může upravovat a přispívat do této on-line encyklopedie.
- Nástěnka se zprávami – Nástěnka se zprávami je perfektní řešení pro sdílení nápadů v rámci oddělení nebo projektového týmu.
- Nahlášení zneužití – Funkce nahlášení umožňuje uživatelům nahlásit jakýkoliv nevhodný obsah na nástěnce zpráv, blozích a jakékoliv jiné stránce v portálu.
- Sledování aktivity – Je možné sledovat poslední aktivitu na Liferay blozích, nástěnce zpráv, Wiki a dalších nástrojích. Tato aktivita je zobrazena v portletu Recent Activity.
- Zprávy – Seznam přátel zobrazuje, kdo je zrovna přihlášen do portálu a je možné uživatelům přímo poslat zprávu.

- Email – Pokud uživatel potřebuje poslat email, je možné to udělat přímo z portálu. Liferay portál obsahuje emailového klienta založeného na AJAX technologii, kterého je možné nakonfigurovat pro použití s mnoha IMAP servery.
- Sdílený kalendář – Kalendář umožňuje uživatelům vytvářet, spravovat a vyhledávat události. Události mohou být sdíleny napříč komunitami a je možné nastavit upozornění, které přijdou pomocí emailu, zprávy nebo SMS.
- Ankety – V Liferay portálu je také možné využívat anket s více možnými odpověďmi a sledovat hlasování. Je možné spravovat více anket a ve zvláštním portletu zobrazovat výsledek anket.
- Social Equity – Je nová funkce Liferay portálu, od verze 6, která umožňuje měřit aktivitu uživatelů. Uživatel tedy může získávat určité body za to, že přispívá komunitě, například přidáváním příspěvků (Wiki, blog), hodnocením nebo komentováním, popřípadě prohlížením obsahu. [13]

1.4 Vývoj pro Liferay Portal

Pro vývoj pro Liferay portál je dobré si nainstalovat Liferay IDE, které je ke stažení z oficiálních stránek Liferay. Není to nezbytné, ale usnadní to vývoj aplikací.

V jedné z předchozích kapitol byla řeč o portletech, ale to není jediná věc, která se využívá pro vývoj pro Liferay Portal. Pro Liferay Portal je možné vytvořit následujících sedm druhů pluginů:

- Portlet,
- Service Builder Portlet,
- Hooks,
- Layout Template,
- Theme,
- Ext,
- a Web. [14] [15]

Vybrané důležité pluginy budou popsány v následujících kapitolách.

Každý plugin ještě lze vytvořit různým způsobem. Vytvořit pluginy můžeme pomocí Liferay Plugins SDK nebo pomocí Liferay Maven Pluginu. [14] [15]

Liferay Plugins SDK je založen na technologii Apache Ant, který je možné využít s jakýmkoliv editorem nebo vývojovým prostředím. [14] [15]

Jelikož se Liferay nesnaží určovat, že má být využívána pouze jediná technologie pro vytváření pluginů, tak existuje druhá možnost a tou je Liferay Maven Plugin, který je založený na Apache Maven. Každý vývojář si tedy může vybrat, jaká technologie mu vyhovuje. [14] [15]

1.4.1 Portlet

Portlety byly popsány v jedné z předchozích kapitol.

1.4.2 Service Builder Portlet

Service Builder je nástroj pro generování kódu, který umožňuje vývojářům definovat vlastní objektové modely, které se nazývají entity. Service Builder generuje vrstvu služeb pomocí objektově relačního mapování (ORM), které poskytují separaci mezi objektovým modelem a kódem pro databázi. [16]

Service Builder přebírá jako vstupní soubor XML a generuje nezbytné vrstvy aplikace. Tyto vrstvy poskytují čisté oddělení. [16]

Service Builder generuje většinu kódu, který je potřebný k implementaci operací, jako jsou čtení, aktualizace, mazání a hledání v databázi. Tím umožňuje vývojáři soustředit se na aspekty vyšší úrovně návrhu služeb. [16]

Hlavní výhody Service Builder jsou:

- Integrace s Liferay portálem.
- Automatické generování vrstev (modelové, perzistentní a vrstvy služeb).
- Automatické generování lokálních a vzdálených služeb.
- Automatické generování Hibernate a Spring konfigurací.
- Podpora generování pro vyhledávací metody pro entity a vyhledávací metody, které obsahují oprávnění.
- Vestavěná podpora pro ukládání do mezipaměti.

- Podpora pro vlastní SQL dotazy a dynamické dotazy.
- Ušetří vývojový čas. [16]

1.4.3 Hooks

Liferay Hook je nejvhodnější plugin pro úpravu základních Liferay funkcí. Pokud je to možné, je dobré použít Liferay Hook vždy pro přepisování základní Liferay funkcionality. Je totiž ještě možné využít Ext plugin pro podobné úkony, ale Liferay Hooky je možné nasadit na běžící server a jsou více kompatibilní. [17]

Pomocí Liferay Hooku jdou přepisovat webové zdroje, které se nacházejí ve složce portal-web ve zdrojových kódech Liferay portálu. Je zde možné přepisovat JSP, JSPF, JavaScript soubory, ale i například obrázky. [17]

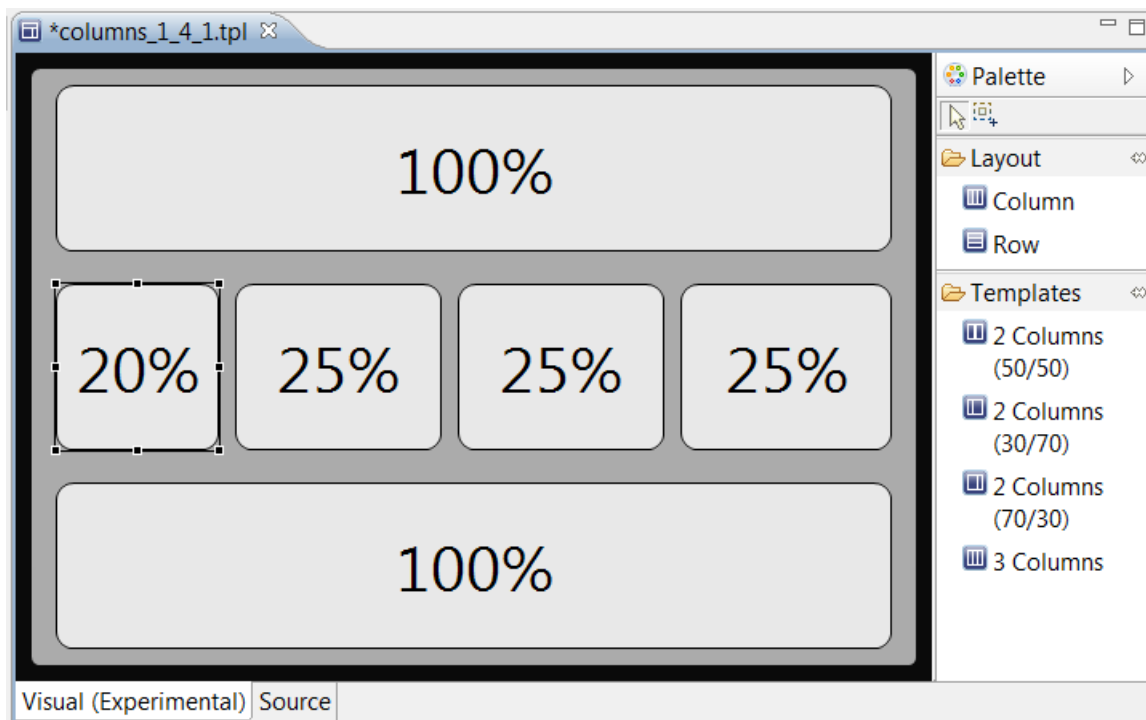
Hooky jsou také užitečné pro spouštění vlastních akcí na běžných portálových událostech, jako je přihlašování uživatele nebo spouštění systému. Tyto akce pro každou událost jsou definované v souboru „portal.properties“, je tedy nutné podědit tento soubor a vytvořit v něm vlastní akci. [17]

Často využívanou vlastností Liferay Hooku je přepisování zpráv nebo popisků v Liferay portálu. Je tedy možné upravit veškerý textový obsah portálu podle potřeb organizace. Realizace je možná pomocí souborů „Language.properties“. [17]

1.4.4 Layout Template

S pomocí šablony, která určuje rozvržení obsahu na stránce, se snadno přizpůsobí Liferay stránka požadovanému obsahu. V šabloně je možné využít technologií, jako jsou například HTML, CSS nebo Velocity. [18]

Pro vytváření šablon je možné využít nástroj Layout Template Editor, který je součástí Liferay IDE. Nástroj obsahuje i vizualizaci tvořené šablony, která je ukázána na obrázku 3. [18]



Obrázek 3 - Layout Template Editor. Zdroj: [18]

Nebo je samozřejmě možné celou šablonu vytvořit, popřípadě upravit, v textovém editoru. [18] Důležitou funkcí šablon je možnost zakomponovat portlet do stránky již při tvoření šablony stránky. To ušetří práci ve chvíli, kdy je známo, že na dané stránce bude vždy určitý portlet na předem definovaném místě, to znamená, že ho tam není nutné pokaždé přidávat ručně. [18]

1.4.5 Theme

K přizpůsobení takzvaného „look and feel“ je vhodný Theme plugin, neboli téma portálu, ve kterém je možné upravit veškerou grafiku portálu. [19]

Theme plugin je možné nasadit do portálu i za běhu serveru a následně v nastavení vybrat nasazené Theme. V případě, že vlastní téma nevyvíjíme, jsou k dispozici různé varianty již předdefinovaných témat. [19]

Pro tvorbu Theme pluginu je nutné znát několik základních technologií:

- CSS – Nové téma je možné vytvořit například pouze změnou existujících CSS souborů. V takovém případě se nejčastěji vychází z jednoho z předdefinovaných témat a jenom se upravuje, aby vyhovovaly dané potřebě.
- Velocity – Velocity je technologie pro šablonování a je důležitá pro vytváření témat, protože v tématech je často nutné přidat nebo upravit šablonu stránky.

- JavaScript – V tématu bývá také potřeba přidat speciální chování, které je nutné implementovat v JavaScriptu. Liferay Portal má svou vlastní JavaScriptovou knihovnu, která se jmenuje AlloyUI a je to rozšíření YUI3 frameworku od společnosti Yahoo.
- XML – Znalost XML se předpokládá z důvodu, že spousta nastavení Theme pluginu je právě v XML formátu. [19]

Výsledná struktura Theme pluginu potom může vypadat jako na obrázku 4. [19]

- `deep-blue-theme/`
 - `docroot/`
 - `WEB-INF/`
 - `liferay-plugin-package.properties`
 - `_diffs/` - subfolders not created by default.
 - `css/`
 - `images/`
 - `js/`
 - `templates/`
 - `css/`
 - `application.css`
 - `base.css`
 - `custom.css`
 - `dockbar.css`
 - `extras.css`
 - `forms.css`
 - `layout.css`
 - `main.css`
 - `navigation.css`
 - `portlet.css`
 - `images/`
 - (many directories)
 - `js/`
 - `main.js`
 - `templates/`
 - `init_custom.vm`
 - `navigation.vm`
 - `portal_normal.vm`
 - `portal_pop_up.vm`
 - `portlet.vm`

Obrázek 4 - Struktura Theme pluginu. Zdroj: [19]

1.4.6 Ext

Jak už bylo zmíněno u Hook pluginu, Ext plugin je mocnější verze pro úpravu základních funkcionalit Liferay portálu. Ovšem je nutné také dodat, že nasazením Ext pluginu je možné udělat o mnoho více škody.

Například při přejití na novější verzi portálu, je nutné ručně projít celý Ext plugin a zkontrolovat veškeré změny proti novým zdrojovým kódům v Liferay portálu. V opačném případě není zaručeno, že v dané instanci nevznikne škoda, která způsobí nesprávné fungování nebo úplnou nefunkčnost portálu. [20]

Z vlastní zkušenosti doporučuji využít Ext plugin pouze v krajním a nezbytném případě. Využívá se hlavně, pokud je nutné upravit funkcionalitu, která je umístěna v Java souborech. Je to z důvodu, že Hook plugin na rozdíl od JSP souborů nebo služeb, které definují určité rozhraní, nezvládne překrýt. Ale takové úpravy se provádí velice výjimečně, spíše je možné se setkat s úpravami na úrovni JSP souborů.

Dále je možné pomocí Ext pluginu definovat v Liferay portálu vlastní třídy nebo vytvořit novou implementaci jakékoliv Liferay beanu deklarované ve Spring souborech. [20]

Ext plugin nelze nasadit za běhu serveru, ale vždy je nutné provést restart, aby byly veškeré změny promítnuty do instance Liferay portálu. Také je potřeba mít na paměti, že není oficiální cesta jak Ext plugin odstranit z instance portálu. [20]

1.5 Další portálová řešení

Společnost Gartner pravidelně analyzuje nejvýznamnější portálová řešení na trhu. Studie této společnosti se nazývá „Magic Quadrant for Digital Experience Platforms“. Nejnovější data jsou dostupná k prosinci 2017, která byla uveřejněna v lednu 2018 a z nich bude čerpáno. [21]

V rámci studie je poskytnut graf, který obsahuje čtyři kvadranty. V pravém horním rohu je nejvýznamnější kvadrant s výstižným názvem „Leaders“. Při postupu proti směru hodinových ručiček se dále nacházejí v grafu kvadranty „Challengers“, „Niche Players“ a „Visionaries“. [21]

Celkem se ve studii objevilo 21 portálových řešení. [21]



Obrázek 5 - Gartner kvadranty. Zdroj: [21]

Z obrázku lze vyčíst, že v kvadrantu „Leaders“ se nalézá Liferay portál jako jedno z nejlepších řešení na trhu. Ovšem je nutné zmínit několik dalších konkurentů, kterými jsou:

- IBM,
- Adobe,
- Sitecore,
- Microsoft,
- Oracle,
- Salesforce
- Acquia,

- SAP
- A OpenText. [21]

Těchto devět řešení jsou nejbližšími konkurenty Liferay portálu a v následujících kapitolách budou některá řešení představena. [21]

1.5.1 IBM WebSphere Portal

WebSphere je integrační softwarová platforma od IBM. Součástí platformy je middleware infrastruktura (server, služby a nástroje), která je potřebná pro napsání, spuštění a monitorování webových aplikací. WebSphere poskytuje spolehlivý, flexibilní a robustní integrační software. [22]

WebSphere Portal je možné upravit podle potřeb organizace nebo uživatelů. Je možné přizpůsobit vzhled portálu, aby vyhovoval standardům organizace a upravit obsah stránek pro uživatele podle byznys pravidel a uživatelských profilů. Uživatelé mohou dále upravovat jejich vlastní vzhled portálu. Uživatelé mohou přidat portlety na stránky, uspořádat je a vybrat barevné schéma. [22]

Portlety jsou hlavním stavebním kamenem WebSphere portálu. Portlety zajišťují přístup k mnoha aplikacím, službám a webovému obsahu. WebSphere portál obsahuje bohatou nabídku portletů. Kromě portletů v katalogích IBM Lotus a WebSphere Portal Business Solutions, je také k dispozici několik portletů třetích stran. Portál podporuje Java Standard API (JSR 286), takže je možné vytvořit vlastní portlety. [22]

Další věci jsou widgety, které jsou zahrnuty přímo do WebSphere portálu. Widgety jsou vysoce interaktivní uživatelské komponenty, které jsou napsané v jazyce JavaScript. [22]

Studie od Gartneru vyčítá WebSphere portálu přehnanou komplexitu na rozdíl od ostatním řešení na trhu. [21]

1.5.2 Microsoft SharePoint

SharePoint je enterprise informační portál k vytváření webů organizací. Může být nakonfigurován jako intranet, extranet nebo internetová stránka. [23]

Jako SharePoint může být označován jeden nebo více produktů, mezi které patří:

- SharePoint Online,

- SharePoint Server a
- SharePoint Foundation. [24]

SharePoint Online je cloudové služba. Místo instalace a nasazení SharePoint serveru si mohou organizace předplatit SharePoint Online. [24]

SharePoint Server je serverové řešení SharePointu. Organizace si mohou nasadit a spravovat na vlastním serveru. Obsahuje všechny služby SharePoint Foundation a nabízí také funkce, jako jsou správa podnikového obsahu, business intelligence, podnikové vyhledávání, osobní weby a informační kanál. [24]

SharePoint Foundation již není ve verzi SharePointu z roku 2016 k dispozici. Ovšem byla to základní technologie pro všechny weby služby SharePoint. Dříve se SharePoint Foundation jmenovalo Windows SharePoint Services. [24]

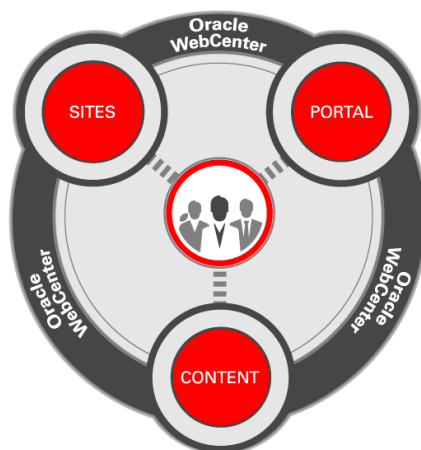
Gartner na adresu SharePointu podotýká, že rychlý vývoj produktu zapříčiňuje řadu stížností od zákazníků na nedokončené vlastnosti systému a potíže s vysvětlováním nových vlastností zaměstnancům. [21]

1.5.3 Oracle WebCenter Portal

Oracle WebCenter Portal je úplný, otevřený a integrovaný podnikový portál, který umožňuje organizacím poskytnout interní nebo externí řešení podnikového portálu. [25]

WebCenter portál umí agregovat a prezentovat informace z aplikací Oracle Cloud Applications nebo i z aplikací třetích stran. Portál bývá dodáván jako součást balíku Oracle WebCenter, který obsahuje ještě produkty Oracle WebCenter Sites a Oracle WebCenter Content. [25]

Dle Gartneru je jednou z nevýhod WebCenter portálu jeho nasazení. Uvádí, že si zákazníci často stěžují na komplexnost nasazení a více než průměrný implementační čas. [21]

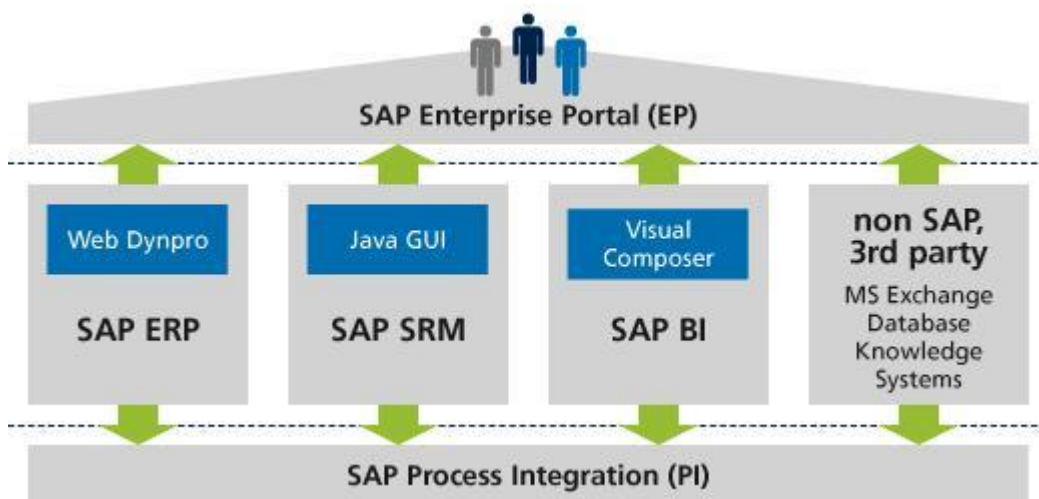


Obrázek 6 - Oracle WebCenter. Zdroj: [26]

1.5.4 SAP Enterprise Portal

SAP Enterprise Portal, dříve též SAP NetWeaver Portal, nabízí jeden bod přístupu jak k informačním zdrojům, podnikovým aplikacím, informačním repozitářům, databázím a službám uvnitř i mimo organizaci – vše je integrováno do jednoho systému. [27]

Gartner ve své studii uvádí, že SAP Enterprise portál je neatraktivní pro organizace, které neinvestují i do dalších produktů společnosti SAP. [21]



Obrázek 7 - SAP Enterprise Portal. Zdroj: [28]

1.6 Správa majetku a aktiv organizace

Správa majetku a aktiv organizace je poměrně náročná a významná část managementu organizace. Pro přesné vysvětlení toho, co je to správa majetku a aktiv organizace, nazývané anglicky též Asset Management, je vhodné uvést následující výstižnou definici:

"Asset managementem se míní systematické a koordinované činnosti a postupy, kterými organizace v průběhu životního cyklu optimálně a trvale spravuje svůj majetek a aktiva, jejich související stav a výkonnost, rizika a výdaje, za účelem dosažení svých organizačních strategických plánů." [29]

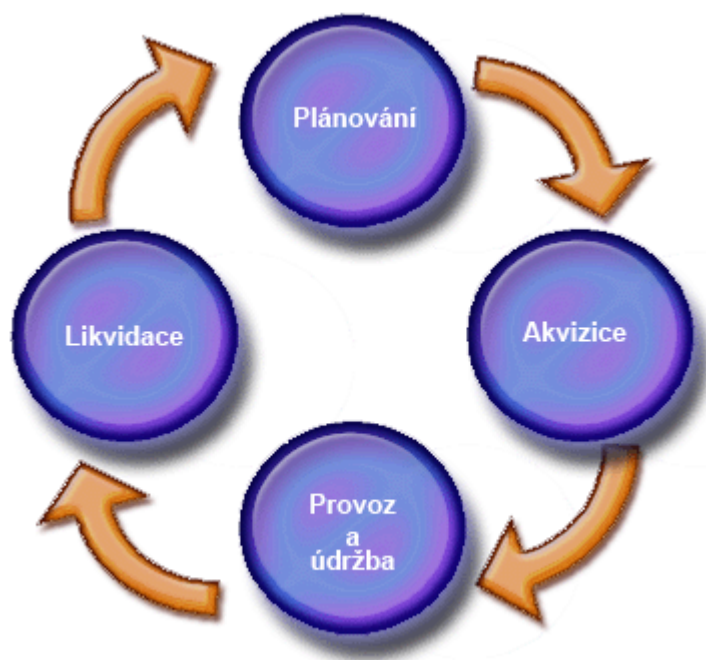
Z definice vyplývá, že Asset management má organizaci vždy zajistit optimální hodnotu majetku, tedy jak komerční, estetickou, funkční či jinou. Tato hodnota by měla v budoucnu organizaci přinášet maximální přínos, ať už v krátkodobém či dlouhodobém časovém horizontu. [29] [30]

Úkolem manažera, který spravuje majetek a aktiva organizace, je ve většině případů udržování majetku v co nejvyšší hodnotě, ovšem může to být udržování i jiné hodnoty než té finanční. [29] [30]

Asset managementu rozlišuje dva druhy aktiv:

- (Operating) asset – Nebo také provozní aktivum, které generuje organizaci přidanou hodnotu. Může to být například firemní vůz k nákladní dopravě.
- Liability – Toto je aktivum, které naopak negeneruje přidanou hodnotu, ale snižuje ji. Například osobní firemní vůz. [29] [30]

Aktivum, též asset, které je fixní, tj. hmotné dlouhodobé, může být jak provozní, tak liability podle účelu používání. Podle toho by měla být také nastavena správa aktiv. Správou aktiv je myšleno nastavení vhodných fází životního cyklu aktiva. Ukázka jednoduchého životního cyklu aktiva je na obrázku 8. [29] [30]



Obrázek 8 - Příklad životního cyklu aktiva. Zdroj: [30]

Aktiva organizace se také velmi často sdružují do skupin, které mají společná pravidla pro řízení:

- pozemky, půda,
- budovy,
- infrastruktura,
- stroje a zařízení,
- a podobně. [29] [30]

U Asset managementu je tedy možné mít dva základní cíle:

- Správná správa majetku – Tento cíl je velice prostý a jde o to, aby aktiva byla efektivně provozovatelná, tedy schopná poskytovat definovanou službu.
- Správné provozování aktiv – To cílí na to, že by provozování aktiv mělo přinášet plánovaný efekt prostřednictvím poskytovaných služeb. [29] [30]

Aktiva jsou také omezována legislativními předpisy (interní i externí), ale i zdroji (rozpočet, úvěry, lidské zdroje, ...).[29] [30]

Asset management lze také dělit do několika kategorií:

- Enterprise Asset Management (EAM) – Správa aktiv napříč odděleními, zařízeními, obchodními jednotkami a geologickým umístěním. [31]
- Digital Asset Management (DAM) – DAM je systém pro správu obsahu (CMS), který centrálně skladuje a spravuje všechny digitální soubory produkované organizací. Umožňuje organizaci ovládat a centrálně spravovat digitální obsah a data, která jsou sdílána uživateli. [32]
- Software Asset Management (SAM) – SAM je využíván na nákup, rozpočtování, správu a použití softwaru v organizacích. [33]
- Information Technology Asset Management (ITAM) – ITAM je typ managementu, který je přímo svázán s podnikovou IT infrastrukturou. S ITAM je možné revidovat celkový soupis hardwaru a softwaru organizace a učinit komplexní rozhodnutí o získávání, použití a dalších aspektech týkajících se životního cyklu aktiva. [34]

S rozvojem informačních technologií už každá větší organizace potřebuje ke svému fungování IT Asset Management aplikaci, který ji vedení informací o hardwaru a softwaru. Lze v ní vést nejen evidenci informačních technologií, ale také dodavatelských smluv, kontraktů nebo spravovat licence k softwaru. ITAM často vychází ze základů metodiky ITIL. [35]

ITAM aplikace může obsahovat řadu funkcionalit a vlastností jako například:

- evidence softwaru,
- evidence hardwaru,
- správa licencí,
- softwarový audit,
- monitoring provozu IT infrastruktury,
- evidence vazeb,
- hlídání termínů a upomínání,
- hlídání shody (Compliance Management),
- evidence smluv k hardwaru a softwaru,
- ukládání souvisejících dokumentů,
- provozní deník IT infrastruktury (IT Assets Tracking),

- preventivní údržba IT infrastruktury (IT Assets Preventive Maintenance),
- IT Service Management (ITSM)
- a další. [35]

1.7 Aplikace pro správu majetku a aktiv organizace

Aplikace pro správu majetku a aktiv organizace, která byla vyvinuta v praktické části této práce, se soustředí na univerzálnost použití a na integraci do portálového řešení Liferay.

Integrace do portálového řešení byla vybrána záměrně. V dnešní době jsou organizace často nuceny využívat spousty různých systémů a to se postupem času stává nepohodlné.

Vyvinutá aplikace proto sází na to, že téměř každá organizace využívá intranet a je tedy vhodné takový intranet využít k tomu, aby všechny aplikace byly do něj zahrnuty. Je to usnadnění pro uživatele i pro správce infrastruktury, jelikož vše se nachází na jednom místě.

Aplikace nerozlišuje, pro jaký typ aktiva bude využívána. Je vyvinuta tak, aby byla plně konfigurovatelná a tak i univerzální. Konfigurovat se dají základní parametry, jako jsou aktiva, životní cyklus nebo umístění, tedy například pobočky organizace.

Je samozřejmě dodržován přístup k jednotlivým částem aplikace podle oprávnění uživatelů. Díky integraci se veškeré nezbytné údaje získávají z portálového řešení a není nutné držet duplicitně, vzhledem k infrastruktuře organizace, některá data. Pro příklad je možné uvést, že samotná aplikace nedrží žádná data o uživateli a všechny získává z portálového řešení, včetně oprávnění, údajů o uživateli a další.

V rámci praktické části se také realizovala instalace, konfigurace a napojení portálu na autentizační server. Realizace proběhla z důvodu možnosti ukázat plnou integraci portálu i aplikace do infrastruktury organizace. V tomto stavu se při přihlášení do portálového řešení ověřuje uživatel proti centrálnímu autentizačnímu serveru, kde jsou uloženy veškeré údaje o uživateli v organizaci. Tento autentizační server je možné využít i pro jiné systémy v infrastruktuře a umožní usnadnění přístupu do všech systémů napříč organizační infrastrukturou, díky stejným přihlašovacími údajům.

V praktické části je k nalezení vše do detailu rozepsané včetně využitých technologií, postupů, konfigurací, instalací a popisů řešení.

2 PRAKTICKÁ ČÁST

2.1 Technologie

2.1.1 Vaadin

Vaadin je javovský framework pro webový vývoj, který je navržen ke snadnému vytváření a správě vysoce kvalitního uživatelského rozhraní. [36]

Počátky Vaadin frameworku se datují do roku 2000, kdy se skupina vývojářů dala dohromady a vytvořila skupinu IT Mill. Tým měl za cíl vytvořit nové programovací paradigma. [37]

Knihovna se nejprve jmenovala Millstone Library. První verze byla použita pro vývoj aplikace pro mezinárodní farmaceutickou společnost. To bylo již v roce 2001. V roce 2006 byla vydána nová generace knihovny nazvaná IT Mill Toolkit Release 4. V této verzi byl představen úplně nový engine založený na AJAXu. Ten umožňoval vyvíjet AJAX aplikace bez starání se o komunikaci mezi klientem a serverem. [37]

V roce 2007 byla vydána verze IT Mill Toolkit 5, která posunula celý projekt dále k AJAXu. Klientská část, která vykresluje uživatelské rozhraní, byla kompletně přepsána s použitím GWT. Tato verze byla vydána pod licencí Apache License 2. [37]

V roce 2009 byl IT Mill Toolkit přejmenován na Vaadin framework a vydána verze 6. [37]

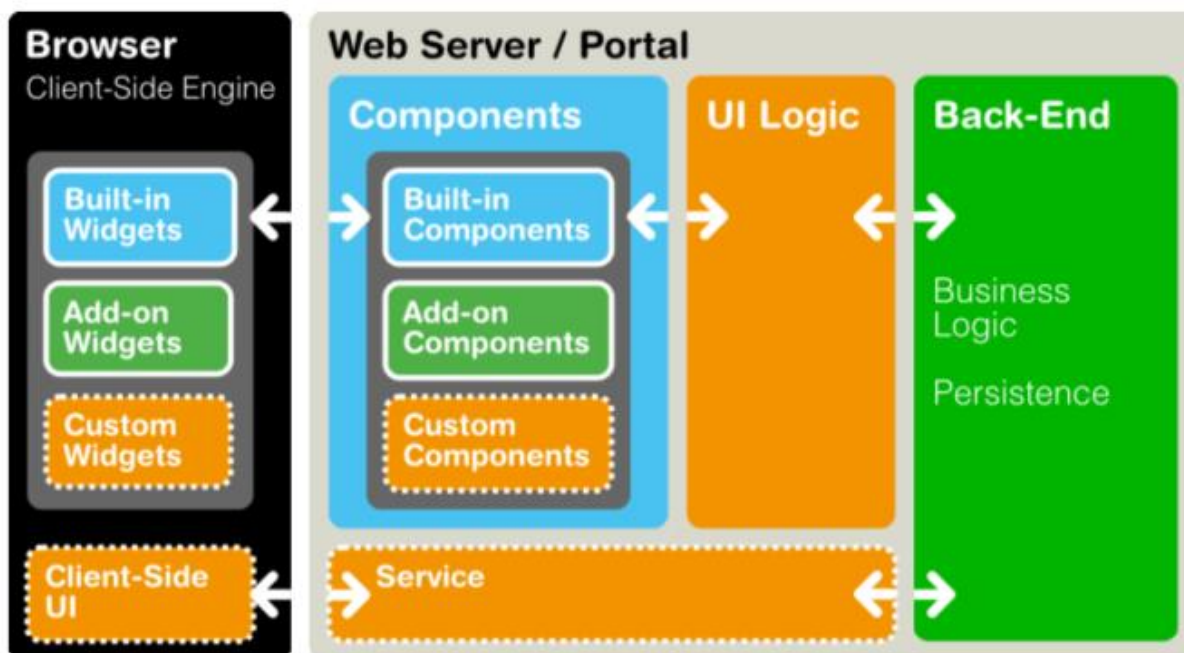
V dnešní době je nejvíce využívána verze 7 a 8. V obou verzích došlo k velkým změnám Vaadin API. [37]

Vaadin framework s přibývajícími léty získává stále více aktivních vývojářů a také ocenění. V roce 2014 byl vyhlášen jako nejlepší javovský webový framework pro byznys aplikace (Java Web UI Framework for Business Applications). [37]

Když se zaměříme na technickou část frameworku, tak Vaadin podporuje dva rozdílné typy programovacích modelů: server-side (serverová část) a client-side (klientská část). Serverově orientovaný model programování nabízí více možností a díky němu můžeme programovat webové aplikace téměř jako aplikace desktopové. Programování je podobné jako v technologiích s konvenčním Java toolkitem, tedy například AWT, Swing nebo SWT. Ale jednodušeji. [36]

Zatímco tradiční webové programování je zábavný způsob jak strávit čas učením se nových technologií, tak ve Vaadinu je možné být produktivní a soustředit se na aplikační logiku aplikace. Serverová část Vaadin frameworku se postará nejen o správu uživatelského rozhraní ve

webovém prohlížeči, ale i o AJAX komunikaci mezi webovým prohlížečem a serverem. Při použití Vaadin frameworku téměř není nutné se učit a zabývat se přímo technologiemi na straně webového prohlížeče jako jsou HTML nebo JavaScript. [36]



Obrázek 9 - Vaadin architektura. Zdroj: [35]

Obrázek 9 ilustruje základní architekturu webových aplikací vytvořených ve Vaadin frameworku. Serverová část aplikační architektury se skládá ze server-side frameworku a client-side engine. Engine se spouští ve webovém prohlížeči jako JavaScriptový kód a má na starosti vykreslování uživatelského rozhraní a posílání uživatelských interakcí na server. Část na obrázku nazvaná jako „UI logic“ je spouštěna jako javovský servlet v javovském aplikačním serveru. [36]

Jelikož je client-side engine spouštěn jako JavaScript ve webovém prohlížeči není potřeba pro aplikace vytvořené ve Vaadin frameworku žádný plugin pro webový prohlížeč. To dává Vaadin frameworku výhodu nad frameworky založenými na Flash, Java Appletech nebo jiných pluginech. Vaadin spoléhá na podporu Google Web Toolkitu, který je optimalizován pro široké spektrum webových prohlížečů a tak se vývojáři nemusí bát o podporu v jakémkoliv webovém prohlížeči. [36]

Protože HTML, JavaScript a další webové technologie jsou v podstatě neviditelné v aplikační logice, tak je možné považovat webový prohlížeč pouze za tzv. tenkého klienta. Tenký klient zobrazuje uživatelské rozhraní a odesílá uživatelské události na server. [36]

Vaadin využívá AJAX technologii, která mu umožňuje vytvořit Rich Internet Applications (RIA), které jsou responzivní a interaktivní stejně jako desktopové aplikace. [36]

Navíc je ještě možné vyvíjet na klientské části pomocí vytváření nových widgetů v Javě, které běží čistě na straně klienta a v prostředí prohlížeče. Toto je možné, protože Vaadin framework na klientské straně obsahuje Google Web Toolkit (GWT), který poskytuje kompilátor z Javy do JavaScriptu, který je následně možné spustit ve webovém prohlížeči. Tímto přístupem Vaadin framework využívá Javu jak na serverové části, tak na klientské části aplikace. [36]

Vaadin framework má své cíle a filozofii. Jednoduše lze říct, že Vaadin framework chce být nejlepší možný nástroj pro vytváření byznys aplikací. Proto má Vaadin několik pravidel:

- Správný nástroj pro správný účel. Vaadin je navržen pro vytváření webových aplikací. Není navržen pro vytváření webových stránek nebo reklamních prezentací. Na takové účely je vhodné zvolit jiné nástroje, jako jsou například JSP/JSF nebo Flash.
- Jednoduchost a spravovatelnost. Vaadin jde cestou robustnosti, jednoduchosti a spravovatelnosti.
- XML není navrženo pro programování. Webové aplikace jsou velice vázané na deklarativní prezentaci uživatelských rozhraní. Ovšem ve Vaadinu je možné tvořit uživatelské rozhraní programátorsky a tím se stává velice dynamické.
- Nástroj by neměl omezovat práci. V případě, že nějaká komponenta uživatelského rozhraní z frameworku neumožňuje dosáhnout čeho je potřeba, musí být jednoduché přidat novou komponentu do aplikace. [38]

2.1.2 Maven

Maven je nástroj pro řízení a správu projektů napsaných v jazyce Java. Maven byl navržen jako nástroj pro zjednodušení buildů pro projekt Jakarta Turbine. Maven slouží primárně k zjednodušení procesu buildování, poskytnutí jednotného systému buildování, poskytování informací o projektu a transparentní migrace na nové funkce. Maven je licencován pod Apache licenci ve verzi 2.0. [39]

2.1.3 Apache Tomcat

Apache Tomcat je open-sourcová implementace Java Servletu, JavaServer Pages, Java Expression Language a Java WebSocket technologií. [40]

2.1.4 LDAP (Lightweight Directory Access Protocol)

Je softwarový protokol, který umožňuje komukoliv lokalizovat organizace, osoby a další zdroje jako jsou soubory a zařízení v síti ať už se jedná o Internet nebo korporátní intranet. Jedná se tedy o adresářovou informační službu. [41]

LDAP je jednodušší verze Directory Access Protocol (DAP). Je jednodušší, protože základní verze neobsahuje bezpečnostní funkce. Pochází z Univerzity v Michiganu a byl schválen nejméně 40 společnostmi. Využívají ho společnosti jako Microsoft, Novell nebo Cisco. [41]

LDAP struktura je organizována do jednoduché stromové hierarchie skládající se z následujících úrovní:

- Kořen.
- Země.
- Organizace.
- Organizační jednotky (divize, oddělení, atp.).
- Osoby (kromě osob sem patří také soubory, tiskárny, atp.). [41]

V mojí aplikaci je použita volně šiřitelná implementace protokolu LDAP, která se nazývá OpenLDAP. OpenLDAP je uvolněn pod licenci BSD. [42]

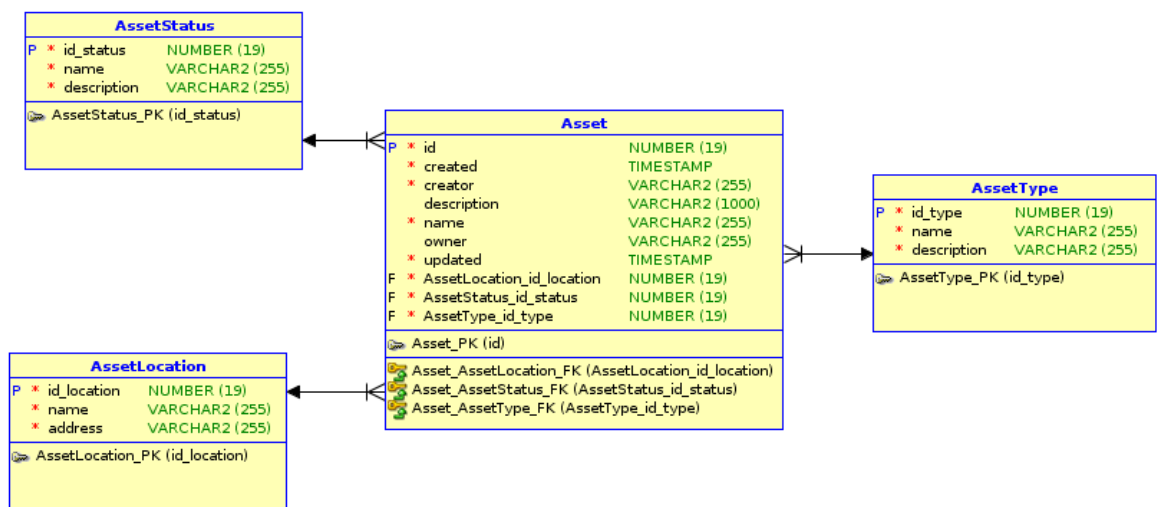
2.2 Databáze

V projektu je využíváno objektově relační mapování (ORM), konkrétně Hibernate framework, který umožňuje automatické vytvoření nebo aktualizaci tabulek v databázi. Tudíž není potřeba žádný SQL skript, který by před spuštěním aplikace vytvořil potřebnou strukturu tabulek v databázi.

Databázový model se skládá ze čtyř tabulek:

- Asset,
- AssetStatus,
- AssetLocation
- a AssetType.

Na ukládání nebo získávání některých dalších informací slouží tabulky, které jsou v Liferay portálu. Například se jedná o práci s uživateli.



Obrázek 10 - Databázový model. Zdroj: Vlastní

Tabulka „Asset“ slouží k uložení samotných assetů a obsahuje atributy:

- `id` (primární klíč),
- `created` (datum, kdy byl asset vytvořen),
- `creator` (uživatel, který asset vytvořil),
- `description` (popis assetu),
- `name` (název assetu),
- `owner` (nynější vlastník assetu, ten komu je přiřazen)
- a `updated` (datum, kdy byl asset naposledy aktualizován).

Tabulka „AssetStatus“ uchovává možné statusy, kterých může asset nabývat a obsahuje následující atributy:

- `id_status` (primární klíč),
- `name` (název statusu)
- a `description` (popis statusu).

Tabulka „AssetType“ obsahuje typy assetů a má tyto atributy:

- `id_type` (primární klíč),
- `name` (název typu assetu)

- a description (popis typu assetu).

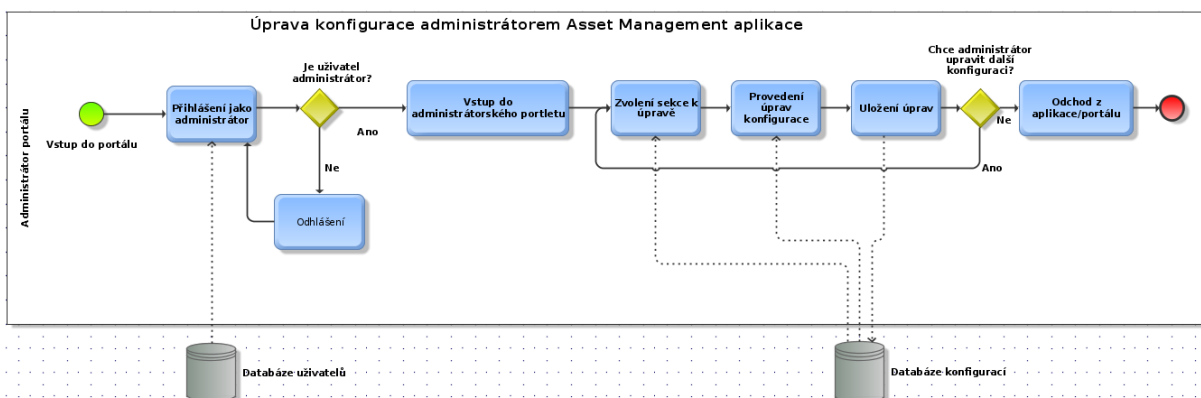
Tabulka „AssetLocation“ slouží k uložení informace, kde se daný asset nachází, tedy například pobočky společnosti. Obsahuje atributy:

- id_type (primární klíč),
- name (název pobočky)
- a address (adresa pobočky).

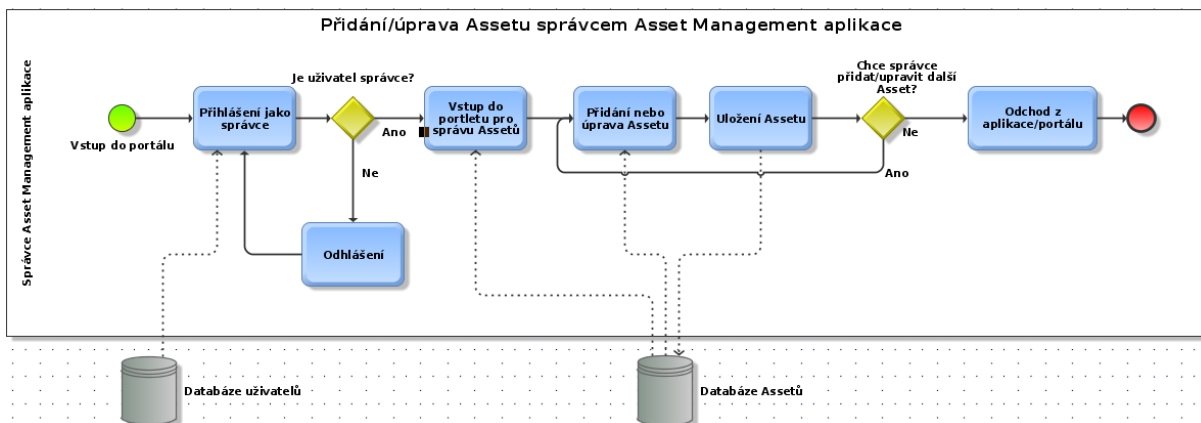
2.3 BPMN

Pro popis vybraných procesů v aplikaci byly využity BPMN diagramy. Jedná se o nejdůležitější procesy a těmi jsou:

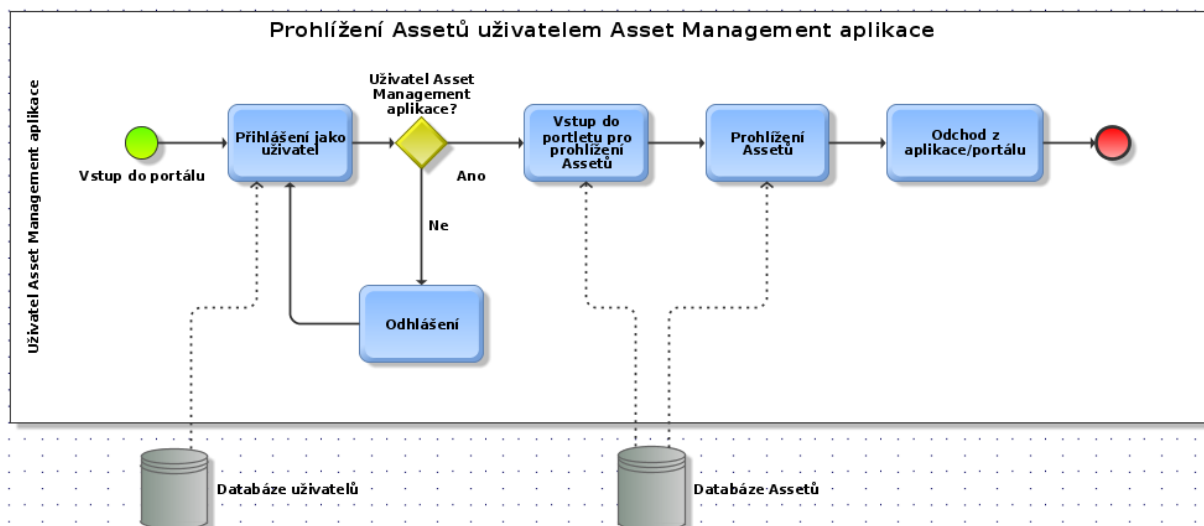
- Úprava konfigurace administrátorem. Obrázek 11.
- Přidání assetu administrátorem. Obrázek 12.
- Prohlížení assetů uživatelem. Obrázek 13.



Obrázek 11 - BPMN Úprava konfigurace. Zdroj: Vlastní



Obrázek 12 - Přidání nebo úprava assetu. Zdroj: Vlastní



Obrázek 13 - Prohlížení assetů. Zdroj: Vlastní

2.4 Aplikace

Aplikace se skládá ze dvou projektů. První projekt jsou samotné portlety s veškerou funkcionalitou a druhý projekt je Liferay Hook, který modifikuje Liferay portál.

2.4.1 Konfigurační soubory v projektu

Projekt s portlety je nazván „AssetManagement“ a nejprve je nutné popsat jeho nastavení, ať už se jedná o nastavení samotného projektu, databáze nebo nastavení vůči Liferay portálu.

V projektu jsou následující důležité konfigurační soubory:

- pom.xml,
- src/main/resources/META-INF/persistence.xml,
- src/main/webapp/WEB-INF/liferay-display.xml,
- src/main/webapp/WEB-INF/portlet.xml,
- src/main/webapp/WEB-INF/liferay-portlet.xml
- a src/main/webapp/WEB-INF/web.xml.

Soubor pom.xml je konfigurační soubor pro Maven. Soubor popisuje základní nastavení projektu jako je jméno, dále groupId nebo artifactId, které jsou nezbytné pro správné fungování Maven projektu. V souboru je také sekce „properties“, kde jsou uvedeny používané verze Vaadin frameworku, Liferay portálu a Liferay maven pluginu. V sekci „dependencies“ jsou zapsány

všechny knihovny, které projekt využívá a potřebuje ke svému správnému fungování. Poslední a důležitá sekce je sekce „packaging“, kde uvedený způsob zabalení projektu. Tento projekt, aby byl snadno nasaditelný do Liferay portálu je zabalován jako WAR.

V souboru persistence.xml je konfigurace Java Persistence API neboli JPA. Je zde tedy nastaveno objektově relační mapování (ORM), které slouží pro ukládání a načítání objektů do a z databáze. Jako implementace JPA standardu je využit Hibernate framework. V souboru je sekce „non-jta-data-source“, ve které je specifikován datový zdroj, který se získá pomocí JNDI a je zapsán ve tvaru „java:comp/env/jdbc/lportal“. Dále je v souboru zapsáno, kterých tříd se bude ukládání do databáze týkat a to je specifikováno v sekcích „class“ a těmi třídami jsou:

- cz.upce.fei.model.Asset,
- cz.upce.fei.model.AssetLocation,
- cz.upce.fei.model.AssetStatus
- a cz.upce.fei.model.AssetType.

Poslední část v souboru je sekce „properties“, kde jsou zdefinovány vlastnosti Hibernate frameworku a specifikován dialekt pro komunikaci s databází.

Soubor liferay-display.xml je konfigurační soubor, který se týká Liferay portálu. V tomto souboru je nakonfigurováno zobrazování portletů, tedy do jaké kategorie se zařadí. V případě, že je potřebné umístit portlet pouze do administrační části Liferay portálu, tak je nutné daný portlet zařadit do kategorie skryté, aby se nezobrazoval ve veřejné části portálu.

```
<display>
  <category name="Asset Management">
    <portlet id="AssetPortlet" />
    <portlet id="ViewPortlet" />
  </category>

  <category name="category.hidden">
    <portlet id="AdministrationPortlet" />
  </category>
</display>
```

Obrázek 14 - liferay-display.xml. Zdroj: Vlastní

Soubor portlet.xml obsahuje definice portletů podle standardu JSR 286. Každý portlet má svou vlastní sekci nazvanou „portlet“ a v ní je několik nezbytných nastavení:

- portlet-name. Jméno portletu, které musí být unikátní a v Liferay portálu se mu také říká portlet ID.

- display-name. Jméno portletu, které se bude zobrazovat v Liferay portálu.
- description. Krátký popis portletu.
- portlet-class. Obsahuje cestu ke třídě, která zpracovává volání do portletu.
- init-param. Inicializační parameter portletu. Obsahuje jméno a hodnotu.
- supports. Zde je uveden podporovaný „mime-type“ a „portlet-mode“. V tomto projektu mají všechny portlety nastaveno na „text/html“ a „view“.

Soubor liferay-portlet.xml je rozšířením portlet.xml pro Liferay portál. Je možné v něm definovat parametry, které jsou specifické pro Liferay portál. Jedním z parametrů je například ikona portletu. Zásadnější parametr je ovšem „control-panel-entry-category“, který s nastavenou hodnotou „site_administration.configuration“ zajistí přidání portletu do konkrétní administrační sekce Liferay portálu. V tomto případě bude portlet přidán do sekce Konfigurace.

Poslední konfigurační soubor je soubor web.xml, což je Deployment Descriptor. V něm je zdefinován servlet, který obsluhuje požadavky. Jelikož v tomto projektu je využívána technologie Vaadin, tak je zdefinován Vaadin servlet, konkrétně „com.vaadin.server.VaadinServlet“. Druhou velice důležitou věcí je nastavení další úrovně abstrakce pro datový zdroj a tedy nastavení parametrů „resource-ref“.

```
<resource-ref>
  <description>lportal DS</description>
  <res-ref-name>lportal</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

Obrázek 15 - web.xml. Zdroj: Vlastní

2.4.2 Struktura projektu

Projekt aplikace je rozdělen do následujících jedenácti balíčků podle obsahu:

- cz.upce.fei.app,
- cz.upce.fei.entitymanagerbean,
- cz.upce.fei.gui.administration,
- cz.upce.fei.gui.administration.location,
- cz.upce.fei.gui.administration.status,

- `cz.upce.fei.gui.administration.type`,
- `cz.upce.fei.gui.asset`,
- `cz.upce.fei.gui.dialog`,
- `cz.upce.fei.model`,
- `cz.upce.fei.portlets`
- a `cz.upce.fei.utils`.

Důležitým balíčkem je balíček `cz.upce.fei.portlets`, kde jsou obsaženy tři třídy pro portlety. Tyto základní třídy následně odkazují do dalších balíčků.

V balíčcích s prefixem `cz.upce.fei.gui.administration` jsou obsaženy třídy pro administrační portlet. Je v nich tedy obsažena celá logika, například přidání, odebrání a úpravy pobočky, typu nebo statusu.

V balíčku `cz.upce.fei.gui.asset` jsou třídy, které pracují s assetem. Jedná se tedy o třídy pro portlet k práci s assety.

Balíček `cz.upce.fei.model` obsahuje třídy, které slouží pro uchovávání informací. S třídami z tohoto balíčku se pracuje v rámci JPA, takže mají potřebné anotace, jako jsou například:

- `javax.persistence.Entity`,
- `javax.persistence.Table`,
- `javax.persistence.Id`,
- `javax.persistence.SequenceGenerator`,
- `javax.persistence.GeneratedValue`,
- `javax.persistence.Column`
- a `javax.persistence.ManyToOne`.

V balíčku `cz.upce.fei.entitymanagerbean` se nalézá třída `EntityManagerBean`, ve které je funkcionální ukládání, mazání a aktualizace objektů. K těmto úkonům je využívám `EntityManager`, který je získán za pomoci `EntityManagerFactory`.

Veškeré zdrojové kódy pro front-end část, tedy HTML soubory, jsou umístěné v „`src/main/resources/cz/upce/fei/gui/`“ a jsou také rozříděné podle obsahu. Jedná se o HTML soubory, které mají nejen běžné HTML tagy, ale také tagy pro Vaadin framework.

2.4.3 Hook

Druhý z projektů je Liferay Hook, který modifikuje zdrojové kódy samotného Liferay portálu. Tento hook upravuje zdrojový soubor Liferay portálu umístěný v „portal-web/dockbar/view_user_account.jspf“, který se stará o zobrazování dockbaru v pravém horním rohu.

V hooku bylo nejprve nutné nakonfigurovat soubor liferay-hook.xml a v něm nastavit cestu k upravenému JSP na „/META-INF/custom_jsps“.

```
<hook>
  <custom-jsp-dir>/META-INF/custom_jsps</custom-jsp-dir>
</hook>
```

Obrázek 16 - Hook tag. Zdroj: Vlastní

Poté se v tomto adresáři vytvořil soubor se stejným jménem a cestou jako je originální soubor v Liferay portálu, který bylo potřeba překrýt. V tomto případě to bylo „META-INF/custom_jsps/html/portlet/dockbar/view_user_account.jspf“. Soubor „view_user_account.jspf“ byl upraven, aby nedovoloval přihlášeným uživatelům nastavovat svůj profil v Liferay portálu.

2.5 Instalační příručka

2.5.1 Instalace a konfigurace OpenLDAP a phpLDAPAdmin

Základní požadavky:

- linuxový systém (tato instalační příručka bude koncipována pro systémy Debian nebo systémy odvozené od Debian jako například Ubuntu),
- Apache webserver,
- PHP.

V první řadě je nutné nainstalovat samotný LDAP server a další nástroje s LDAP serverem spojené. Naštěstí veškeré nutné balíčky jsou k dispozici v repozitářích. K nainstalování se zadají následující příkazy:

```
sudo apt-get update
sudo apt-get install slapd ldap-utils
```

V průběhu instalace se zadá administrátorské heslo pro LDAP server.

Po dokončení instalace se rovnou přejde k rekonfiguraci. Balíček slapd je potřeba nastavit několika důležitými parametry, které se při samotné instalaci nezadávají. Takže je nutné zadat následující příkaz, kterým bude rekonfigurován nainstalovaný balíček:

```
sudo dpkg-reconfigure slapd
```

Po zadání příkazu je nutné vyplnit potřebné parametry, na které se konfigurace bude doptávat. Většinu doptávaných parametrů je možné ponechat v defaultním nastavení. Parametry jsou následující:

- Omit OpenLDAP server configuration? (Vynechat OpenLDAP server konfiguraci?) Je nutné zvolit odpověď No (Ne).
- DNS domain name? (Doménové jméno DNS?) Zde je potřeba vyplnit doménové jméno ve formátu example.org. Například assetmanagement.org.
- Organization name? (Jméno organizace?) Zde je nutné zadat jméno organizace. Například assetmanagement.
- Administrator password? (Administrátorské heslo?) Zde je nutné dvakrát zadat administrátorské heslo.
- Database backend? MDB.
- Remove the database when slapd is purged? (Odstranit databázi, když je odstraněn balíček slapd?) No (Ne).
- Allow LDAPv2 protocol? (Povolit LDAPv2 protokol?) No (Ne).

Po této konfiguraci je LDAP server nakonfigurovaný a běžící. V případě, že je potřeba, aby se připojovali externí klienti, je nutné povolit LDAP port na firewallu následujícím příkazem:

```
sudo ufw allow ldap
```

Pro otestování, že LDAP server běží a funguje správně, je možné zadat ldapwhoami příkaz, který vypíše uživatelské jméno, kterým jsme připojeni:

```
ldapwhoami -H ldap:// -x
```


Výsledkem by měl být výstup v tomto formátu:

Output
Anonymous

„anonymous“ se vypíše z důvodu, že příkaz ldapwhoami byl spuštěn bez zalogování do LDAP serveru.

Tímto je LDAP server připraven.

Pro snazší administraci LDAP serveru je vhodné doinstalovat phpLDAPadmin. phpLDAPadmin je webový interface pomocí kterého lze snadno spravovat LDAP server s využitím grafického uživatelského rozhraní. V opačném případě lze pro administraci použít příkazovou řádku.

PhpLDAPadmin lze opět nainstalovat z repozitářů s využitím následujícího příkazu:

```
sudo apt-get install phpldapadmin
```

Tento příkaz nainstaluje aplikaci, povolí nezbytnou konfiguraci v Apache serveru a restartuje ho. Tímto je web server nakonfigurován, ale je potřeba udělat dodatečné změny jako je konfigurace phpLDAPadmin, aby využíval naši doménu a automaticky nevyplňoval přihlašovací údaje k LDAP serveru.

Tyto změny se provedou v konfiguračním souboru. K těmto změnám je nutné použít root oprávnění a některý z textových editorů:

```
sudo gedit /etc/phpldapadmin/config.php
```

Po otevření souboru je nutné hledat řádku, která začíná „\$servers->setValue('server','name', ...“.
Na této řádce se nastavuje jaké bude zobrazované jméno LDAP serveru. Vyplníme například následovně:

```
$servers->setValue('server','name','Asset Management LDAP');
```

Dále je potřeba upravit další řádek, který začíná „\$servers->setValue('server','base', ...“.
Tento řádek nakonfiguruje kořen LDAP hierarchie. Je zde nutné zadat doménové jméno, které bylo vyplněno při rekonfiguraci slapd balíčku, takže v našem případě to bude assetmanagement.org.
Ale je potřeba to zadat v syntaxi LDAP serveru, bude to tedy ve formátu „dc=assetmanagement,dc=org“ následovně:

```
$servers->setValue('server','base',  
array('dc=assetmanagement,dc=org'));
```

Další řádek ke konfiguraci je defaultně nezakomentovaný a vypadá takto:

```
„$servers->setValue('login', 'bind_id', 'cn=admin,dc=example,dc=com');“
```

Jelikož tato konfigurace vyplňuje administrátorské detaily k přihlášení do webového rozhraní, tak tento řádek se zakomentuje pomocí symbolu „#“. Upravený řádek tedy bude mít tuto podobu:

```
„#$servers->setValue('login', 'bind_id', 'cn=admin,dc=example,dc=com')“
```

Poslední úprava bude spočívat v nezobrazování tolika varovných hlášek, jelikož nejsou důležité. Je tedy nutné upravit v souboru příslušný řádek, aby vypadal následovně:

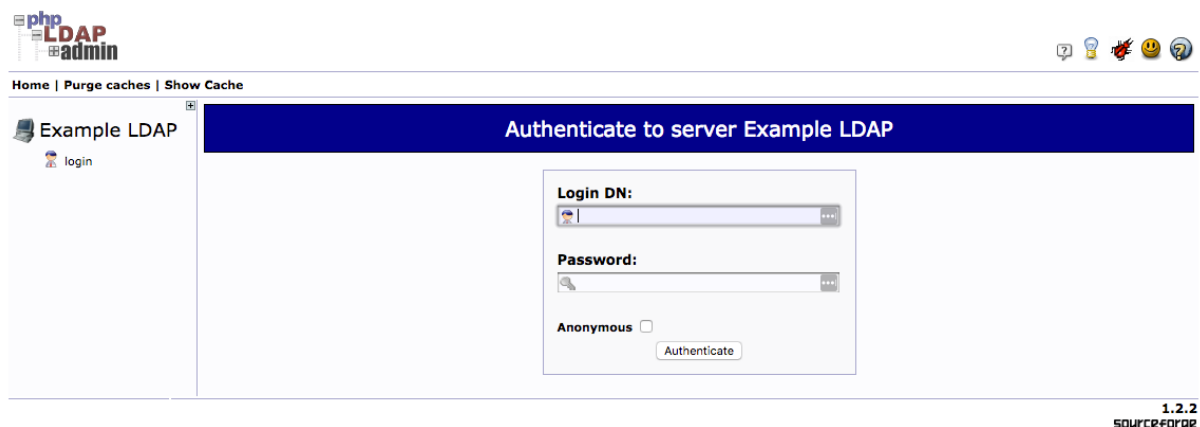
```
$config->custom->appearance['hide_template_warning'] = true;
```

A to je vše, po uložení a zavření souboru je vše nakonfigurováno. Není potřeba nic restartovat, změny se projeví okamžitě po uložení.

Nyní je možné se přihlásit do administrace LDAP serveru. Jak již bylo zmíněno, jedná se o webové rozhraní, je nutné tedy zadat následující URL adresu do webového prohlížeče:

<https://assetmanagement.org/phpldapadmin>

Po načtení stránky je zobrazen přihlašovací formulář. Pokud ne, vybere se v levém menu možnost „login“. Do pole „Login DN:“ je potřeba zadat přihlašovací jméno, ovšem opět v LDAP formátu.



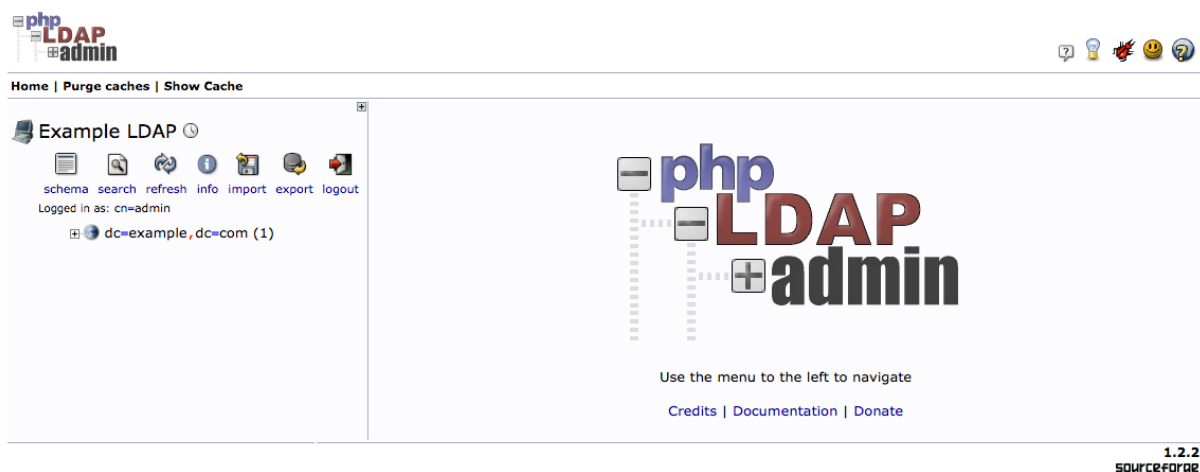
Obrázek 17 - Přihlašovací stránka LDAP serveru. Zdroj: Vlastní

Pro přihlášení jako administrátor bude formát tedy následovný:

cn=admin,dc=assetmanagement,dc=org

A do pole „Password:“ se zadá heslo, které bylo nastaveno při instalaci LDAP serveru.

Po autentizaci je zobrazena hlavní stránka, odkud je možné začít administrovat LDAP server.



Obrázek 18 - Hlavní stránka LDAP serveru. Zdroj: Vlastní

2.5.2 Instalace databáze

Pro aplikaci byla využita databáze od společnosti Oracle, ale je možné využít jakoukoliv jinou dostupnou databázi dle uvážení.

Nejprve je nutné si stáhnout databázi Oracle z webových stránek společnosti Oracle. Lze využít například Oracle Database 11g Express Edition, která je dostupná pro platformy Windows a Linux. Pro instalaci, prvotní nastavení a spuštění doporučuji využít oficiální návod dostupný na webových stránkách společnosti Oracle.

Nic dalšího není potřebné. Samotná aplikace využívá Hibernate framework, který si sám, po připojení k databázi, vytvoří potřebné databázové tabulky. To samé udělá i Liferay portál.

2.5.3 Instalace Liferay portálu

Nejdříve je potřeba stáhnout potřebný Liferay portál ze stránek <https://www.liferay.com/downloads>. Tato aplikace je vyvíjena na Liferay portálu ve verzi 6.2 GA6 v balíku s Tomcat serverem.

Pokud příslušná verze není nalezena na předchozí stránce, je možné ji nalézt na oficiálním sourceforge na adrese <https://sourceforge.net/projects/lportal/files/Liferay%20Portal/6.2.5%20GA6/>.

Po stažení je nutné rozbalit archív a provést několik základních konfigurací před prvním spuštěním.

Konfigurační soubory Apache Tomcat serveru lze nalézt ve složce:

```
$LIFERAY_HOME/tomcat-7.0.62/conf/
```

Kde \$LIFERAY_HOME je cesta k rozbalenému portálu. V této složce je nutné otevřít textový soubor s názvem server.xml textovým editorem následovně:

```
gedit server.xml
```

A po otevření editoru je nutné nalézt a upravit následující řádek:

```
<Connector URIEncoding="UTF-8" connectionTimeout="20000" port="8080"
protocol="HTTP/1.1" redirectPort="8443"/>
```

Úprava bude spočívat ve změně portu. Tato úprava je z toho důvodu, že na portu 8080 běží grafické rozhraní pro Oracle databázi a je tedy nutné zvolit port jiný, například 8880. Řádek po úpravě bude vypadat následovně:

```
<Connector URIEncoding="UTF-8" connectionTimeout="20000" port="8880"
protocol="HTTP/1.1" redirectPort="8443"/>
```

Jelikož pro tuto aplikaci je využívána Oracle databáze a Liferay si s sebou defaultně nenese driver pro Oracle databázi, tak je nutné driver dodat ručně. Lze tak učinit například stažením driveru „ojdbc7“ a nakopírováním do složky:

```
$LIFERAY_HOME/tomcat-7.0.62/lib/ext/
```

Nyní lze provést prvotní spuštění Liferay portálu a to příkazem:

```
./$LIFERAY_HOME/tomcat-7.0.62/bin/startup.sh
```

V případě nedostatečných práv je nutné spustit příkaz:

```
sudo chmod u+x $LIFERAY_HOME/tomcat-7.0.62/bin/startup.sh
```

A poté opakovat spuštění portálu.

Log o průběhu spuštění portálu se jmenuje catalina.out a lze ho nalézt ve složce:

```
$LIFERAY_HOME/tomcat-7.0.62/logs/
```

Po úspěšném spuštění portálu se otevře webový prohlížeč, kde je nutné udělat základní nastavení.

Liferay Basic Configuration

Portal

Portal Name
 For example, Liferay.

Default Language
 Change

Add Sample Data

Administrator User

First Name

Last Name

Email (Required)

Database

Default Database (Hypersonic)
This database is useful for development and demo'ing purposes, but it is not recommended for production use. [\(Change\)](#)

Finish Configuration

Powered By [Liferay](#)

Obrázek 19 - Základní nastavení Liferay portálu. Zdroj: Vlastní

Základní konfiguraci lze udělat následujícím způsobem:

- Portal Name → Asset Management.
- First Name, Last Name → Admin.
- Add Sample Data → odškrtnout.
- Email → admin@liferay.com.

V defaultním nastavení se využívá Hypersonic databáze. Ta je ovšem pouze pro testovací záležitosti a pro produkční prostředí nevhodná. Je nutné tedy databázi přenastavit na nainstalovanou Oracle databázi. Po kliknutí na tlačítko „(Change)“ je potřebné vyplnit do formuláře údaje k databázi.

Database

[« Use Default Database](#)

Database Type

JDBC URL (Required)

JDBC Driver Class Name (Required)

User Name

Password

[Finish Configuration](#)

Obrázek 20 - Nastavení Liferay databáze. Zdroj: Vlastní

Po vyplnění je možné dokončit konfiguraci stisknutím tlačítka „Finish Configuration“ a po vydařené konfiguraci přejít tlačítkem „Go to My Portal“ do samotného portálu, kde ještě následuje odsouhlasení podmínek užívání, nastavení hesla a otázky pro obnovu hesla.

Your configuration was saved successfully.

The configuration was saved in `XXXXXXXXXXXXX/liferay-portal-6.2-ce-ga6/portal-setup-wizard.properties`.

[Go to My Portal](#)

Powered By [Liferay](#)

Obrázek 21 - Úspěšná konfigurace Liferay. Zdroj: Vlastní

Tímto je celá instalace a spuštění Liferay portálu úspěšně dokončeno.

2.5.4 Zabalení Asset Management projektu

Nejprve je nutné přistoupit do složky s projektem na úroveň kde je umístěn soubor pom.xml ve kterém je nadefinován způsob zabalení projektu do archivu. U tohoto projektu je nastaveno balení do formátu WAR. Samotné zabalení aplikace se provede Mavenem pomocí příkazu:

```
mvn clean package
```

Nyní lze nalézt ve složce „target“ soubor s koncovkou WAR, který se použije k nasazení aplikace do Liferay portálu.

Nejprve je ale potřeba stejným způsobem vytvořit balíček s Liferay Hookem, který upravuje chování Liferay portálu.

2.6 Konfigurace

2.6.1 Data Source

Nejprve je nutné nakonfigurovat Data Source neboli zdroj dat pro aplikaci. Pro obecné využití databáze lze nakonfigurovat zdroj dat v aplikačním serveru Tomcat. První soubor, kam je potřeba přidat konfigurace, je server.xml, který je umístěn v:

```
$LIFERAY_HOME/tomcat-7.0.62/conf/
```

V tomto souboru je nutné nalézt tag „<GlobalNamingResources>“ a uvnitř tohoto tagu zakomentovat následující tag:

```

<Resource name="UserDatabase"
    auth="Container"
    type="org.apache.catalina.UserDatabase"
    description="User database that can be updated and saved"
    factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
    pathname="conf/tomcat-users.xml"
/>

```

Po zakomentování se přidá dovnitř tagu „<GlobalNamingResources>“ vlastní zdroj dat, který vypadá následovně:

```

<Resource auth="Container"
    driverClassName="oracle.jdbc.OracleDriver"
    maxActive="20"
    maxIdle="10"
    maxWait="-1"
    name="jdbc/lportal"
    password="..."
    type="javax.sql.DataSource"
    url="jdbc:oracle:thin:@127.0.0.1:1521:xe"
    username="..."
/>

```

Do této konfigurace se přidají údaje o databázi. Je nutné vyplnit pouze parametr „url“, kde se vyplní Connection String do databáze a dále „username“ a „password“, kde se vyplní uživatelské jméno a heslo k databázi.

Dále je nutné upravit soubor context.xml, který je umístěn ve stejné složce jako server.xml. Do souboru context.xml je nutné přidat konfiguraci „Resource Linku“ do tagu „<Context>“. Konfigurace vypadá následovně:

```

<ResourceLink global="jdbc/lportal"
    name="jdbc/lportal"
    type="javax.sql.DataSource"
/>

```

2.6.2 LDAP server

Pro konfiguraci LDAP serveru je nutné přejít do webového rozhraní LDAP serveru, prostřednictvím phpLDAPadmin aplikace, a přihlásit se.

V levém menu je nutné vybrat „Create new entry here“ pro vytvoření potřebné struktury. Pro potřeby této aplikace je potřeba vytvořit skupiny a uživatele. Skupiny budou určovat role v aplikaci a uživatelé uvedení v LDAP serveru budou mít přístup do Liferay portálu.

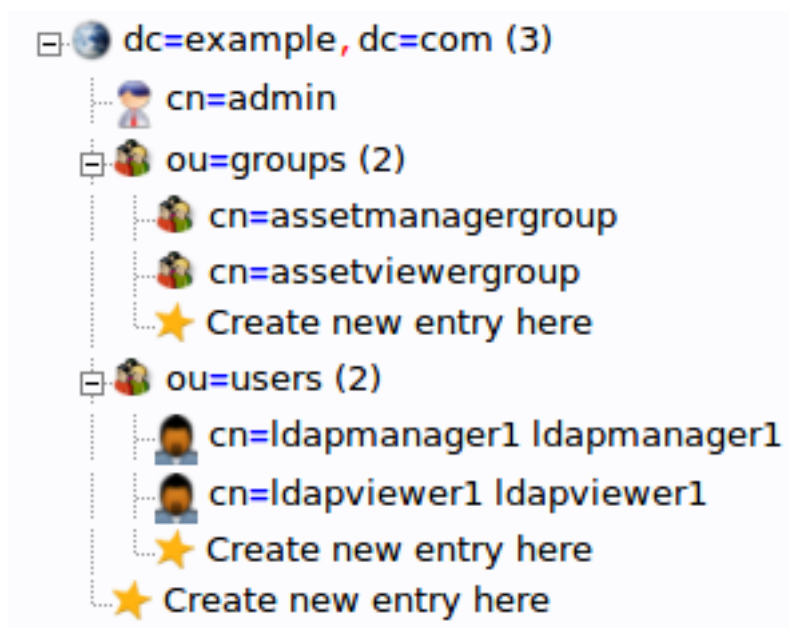
Pro konfiguraci skupin je nutné vybrat možnost „Generic: Organizational Unit“ a zadat název „groups“. Tímto je vytvořena struktura pro skupiny a nyní se vytvoří jednotlivé skupiny tím, že se klikne v levém sloupci na nově vytvořenou položku „ou=groups“ a následně se vybere možnost „Create a child entry“. Nyní při výběru šablony není možné vybrat ani „Generic: Organizational Unit“ ani „Generic: Posix Group“, protože tyto šablony nelze napojit do Liferay portálu. Je tedy nutné vybrat položku „default“ a následně zvolit možnost „groupOfNames“. Takto se vytvoří dvě skupiny, například „assetmanagergroup“ a „assetviewergroup“.

Pro konfiguraci uživatelů lze vybrat opět možnost „Generic: Organizational Unit“ a vytvořit organizační jednotku s názvem „users“, která bude na stejné úrovni jako „groups“. V organizační jednotce je nutné vytvořit potřebné uživatele, které opět nebudou ze standardní šablony, ale po výběru „Default“ je potřeba zvolit možnost „inetOrgPerson“, opět z důvodu kompatibility s Liferay portálem. Po vytvoření všech uživatelů je nutné upravit šablonu, aby v ní byly všechny potřebné položky k namapování do Liferay portálu. Struktura šablony je zobrazena na obrázku 22.

cn	required, rdn
<input type="text" value="ldapviewer1 ldapviewer1"/> * (add value) (rename)	
Email	alias
<input type="text" value="ldapviewer1@liferay.com"/> (add value)	
gidNumber	required
<input type="text" value="501"/>	
givenName	
<input type="text" value="ldapviewer1"/> (add value)	
homeDirectory	required
<input type="text" value="/home/users/ldapviewer1"/>	
objectClass	required
<input type="checkbox"/> inetOrgPerson (structural) <input type="checkbox"/> <input type="text" value="posixAccount"/> <input type="checkbox"/> <input type="text" value="top"/> (add value)	
Password	alias
<input type="password" value="....."/> md5 <input type="button" value="v"/> Check password... (add value)	
sn	required
<input type="text" value="ldapviewer1"/> (add value)	
uidNumber	required
<input type="text" value="1001"/>	
User Name	alias, required
<input type="text" value="ldapviewer1"/> (add value)	

Obrázek 22 - Šablona uživatele. Zdroj: Vlastní

Tímto je hotová konfigurace LDAP serveru. Konečná stromová struktura je zobrazena na obrázku 23.



Obrázek 23 - Výsledná stromová struktura. Zdroj: Vlastní

2.6.3 Liferay Portál

Konfigurace propojení Liferay portálu a LDAP serveru je velice obtížná a časově náročná. Pro konfiguraci v Liferay portálu je nutné přejít do administrace, která se nachází po přihlášení, jako administrátor, pod volbou „Admin“ v levém horním panelu a výběru možnosti „Control Panel“. Po načtení administrace je nutné pokračovat možností „Portal Setting“ v sekci „Configuration“ a následně v pravém menu volba „Authentication“ a záložka „LDAP“.

V této sekci je provedena konfigurace a namapování Liferay uživatelů na uživatele z LDAP serveru. Povolením LDAP autentizace je možné přidat LDAP server k ověřování uživatelů. Pro nastavení ověřování je potřebné zadat správně několik parametrů:

- Server Name → libovolné jméno LDAP serveru.
- Default Values → druh LDAP serveru, zvolí se OpenLDAP.

Connection (Připojení):

- Base Provider URL → URL adresa LDAP serveru včetně portu.
- Base DN → doménové jméno v syntaxi LDAP serveru.

Users (Uživatelé):

- Authentication Search Filter → autentizační filter, který se využije při ověřování uživatelů.
- Import Search Filter → zde je uvedena třída objektu, který bude Liferay portál hledat v LDAP serveru.

User Mapping (Namapování uživatele):

- UUID,
- Screen Name,
- Email Address,
- Password,
- First Name,
- Last Name,
- a Group.

Group Mapping (Namapování skupiny):

- Group Name,
- User.


Server Name

Default Values

- Apache Directory Server
- Fedora Directory Server
- Microsoft Active Directory Server
- Novell eDirectory
- OpenLDAP
- Other Directory Server

Obrázek 24 - Jméno serveru a výchozí hodnoty. Zdroj: Vlastní

Connection

Base Provider URL 

Base DN 

Principal

Credentials

Obrázek 25 – Připojení. Zdroj: Vlastní

Users

Authentication Search Filter 

Import Search Filter

Obrázek 26 – Uživatelé. Zdroj: Vlastní

User Mapping

UUID

Screen Name

Email Address

Password

First Name

Middle Name

Last Name

Full Name 

Job Title

Status

Group

Portrait

Custom User Mapping

Custom Contact Mapping

Test LDAP Users

Obrázek 27 - Namapování uživatelů. Zdroj: Vlastní

Groups

Import Search Filter

Group Mapping

Group Name

Description

User

Obrázek 28 - Namapování skupin. Zdroj: Vlastní

2.6.4 Uživatelské skupiny a role

Po tom, co bylo nastaveno mapování uživatelů z LDAP serveru na uživatele v Liferay portálu, se provede vytvoření uživatelských skupin a rolí. Toto nastavení je k nalezení v administraci portálu v sekci Uživatelů.

Uživatelské skupiny se vytvoří podle skupin v LDAP serveru. Jsou tedy vytvořeny uživatelské skupiny:

- „assetmanagergroup“
- a „assetviewergroup“.

Role se vytvoří celkem dvě a jsou to:

- Asset Admin Role – Role Asset Admin Role je přiřazena uživatelům, kteří zajišťují správu Assetů.
- Asset View Role – Tato role je pro prohlížení Assetů bez možnosti úprav.

Po vytvoření uživatelských skupin i rolí je potřeba docílit nejdůležitější věci. A tou je automatické přiřazování role určité uživatelské skupině ve chvíli, kdy se provede autentizace a tedy přenesení uživatele z LDAP serveru do Liferay portálu.

Tuto konfiguraci se provede v nastavení role, kde se nalezne záložka „Assign Members“ a podzáložka „User Groups“. V této podzáložce se namapují příslušné role na uživatelské skupiny. Tedy roli Asset Admin Role je nutné namapovat na uživatelskou skupinu „assetmanagergroup“ a roli Asset View Role na „assetviewergroup“.

2.6.5 Ostatní konfigurace

Po provedení předchozích konfigurací je Liferay portál připraven, ale chybí to nejdůležitější a to je grafická prezentace uživatelům. Je tedy nezbytné doplnit logo, stránky, umístit portlety nebo webový obsah na stránky.

Změna z defaultního loga portálu na vlastní logo se provede v administraci portálu tentokrát v sekci „Configuration“. V záložce „Portal Settings“ se vybere z pravého menu položku „Display Settings“ a na této stránce se nalezne položka Logo, kde je možné nahrát libovolné logo pro celý Liferay portál.

Pro vytvoření stránek je nutné být administrátor Liferay portálu. Veškerý obsah se přidává na startovací stránce portálu za pomoci velkého plus v levé části obrazovky.



Obrázek 29 - Přidání obsahu. Zdroj: Vlastní

ZÁVĚR

V rámci diplomové práce je vytvořena aplikace pro správu majetku a aktiv organizace, která je zaintegrována do open-source portálového řešení.

Při přípravě práce nebyla zanedbána analýza použitých technologií a tak se nevyskytly větší obtíže. Programovací jazyk Java ve spolupráci s Vaadin frameworkem je pro tento typ projektů vhodně zvolená technologie. Liferay Portal je jedním z nejrozšířenějších open-source řešení pro tvorbu portálových řešení a nabízí nepřehledné množství vlastností a funkcionalit, které usnadní vývoj. Co se týče adresářového serveru OpenLDAP pro uložení uživatelů organizace, tak i přesto, že jsem neměl předchozí zkušenosti s touto technologií, byla konfigurace, po malém zorientování se v problematice, intuitivní.

Vývoj samotné aplikace pro správu majetku a aktiv organizace byl bezproblémový a veškeré dané cíle byly splněny. Aplikace je vyvinuta podle takových zásad, aby bylo možné v budoucnu rozšíření o další funkcionality.

Velká časová náročnost vznikla na problému propojit OpenLDAP server s Liferay portálem. Konfigurace namapování Liferay uživatelů na uživatele uložené na OpenLDAP serveru nebyla snadná, protože ani v oficiální dokumentaci k tomu není dostatečné množství informací. Propojení se povedlo i díky tomu, že je Liferay Portal open-source a bylo tak možné nahlédnout do implementace mapování uživatelů.

Takto navržený projekt lze využít jako základní kámen pro stavbu intranetu nebo extranetu organizace. Celý intranet nebo extranet by bylo možné postupně rozšiřovat o další aplikace, ať už by se jednalo o již vyvinuté aplikace dostupné na Liferay Marketplace nebo aplikace vyvinuté na míru organizaci. Z mé zkušenosti vychází, že se netřeba bát ani velké zátěže, protože je možné Liferay Portal zařadit do clusteru a přidat server k vyrovnávání zátěže. Takto navržený projekt tedy plní veškeré potřeby i pro budoucí rozšíření.

POUŽITÁ LITERATURA

- [1] JONAS X. YUAN. *Liferay portal systems development: build dynamic, content-rich, and social systems on top of Liferay*. 2nd ed. Birmingham: Packt Pub, 2012. ISBN 9781849515986.
- [2] What Is a Portal?. *O'Reilly Media* [online]. 2005 [cit. 2018-03-31]. Dostupné z: http://archive.oreilly.com/pub/a/java/archive/what-is-a-portlet.html?page=2#what_is_a_portal
- [3] SARIN, Ashish. *Portlets in action*. Shelter Island, NY: Manning, c2012. ISBN 978-193-5182-542.
- [4] JSR 168: Portlet Specification. *The Java Community Process(SM) Program* [online]. [cit. 2018-03-31]. Dostupné z: <https://jcp.org/en/jsr/detail?id=168>
- [5] What Is a Portlet. *O'Reilly Media* [online]. 2005 [cit. 2018-03-31]. Dostupné z: <http://archive.oreilly.com/pub/a/java/archive/what-is-a-portlet.html>
- [6] JSR 286 Portlet Life Cycle. *Pro Liferay - A guide for Liferay Developer* [online]. 22. 10. 2014 [cit. 2018-03-31]. Dostupné z: <http://proliferay.com/jsr-286-portlet-life-cycle/>
- [7] Introducing Java Portlet Specifications: JSR 168 and JSR 286. *Oracle* [online]. 16. 3. 2007 [cit. 2018-03-31]. Dostupné z: <http://www.oracle.com/technetwork/java/index-raji-test-141933.html>
- [8] Introduction to JSR 286/ Portlet Specification 2.0. *Liferay Savvy* [online]. Duben 2014 [cit. 2018-03-31]. Dostupné z: <http://www.liferay Savvy.com/2014/04/introduction-to-jsr-286-portlet.html>
- [9] Introduction to JSR 362 / Portlet Specification 3.0. *Liferay Savvy* [online]. Březen 2016 [cit. 2018-03-31]. Dostupné z: <http://www.liferay Savvy.com/2016/03/introduction-to-jsr-362-portlet.html>
- [10] Portlet Modes in Liferay. *Tech Blog* [online]. 15. 7. 2016 [cit. 2018-03-31]. Dostupné z: <http://www.opensource-techblog.com/2016/07/portlet-modes.html>
- [11] Portal Features. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: <https://web.liferay.com/products/liferay-portal/features/portal>

- [12] Enterprise Content Management Features. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: <https://web.liferay.com/products/liferay-portal/features/enterprise-cms>
- [13] Collaboration and Social Features. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: <https://web.liferay.com/products/liferay-portal/features/collaboration>
- [14] The Plugins SDK. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-1/the-plugins-sdk
- [15] Developing Plugins Using Maven. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-1/developing-plugins-using-maven
- [16] What is Service Builder?. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-2/what-is-service-builder
- [17] Hooks. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-1/hooks
- [18] Creating Liferay Layout Templates. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-1/creating-liferay-layout-templates
- [19] Creating Liferay Themes. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-1/creating-liferay-themes
- [20] Ext plugins. *Liferay* [online]. [cit. 2018-03-31]. Dostupné z: https://dev.liferay.com/develop/tutorials/-/knowledge_base/6-1/ext-plugins
- [21] MURPHY, Jim, Gene PHIFER, Gavin TAY a Mike LOWNDES. *Magic Quadrant for Digital Experience Platforms* [online]. 17. 1. 2018 [cit. 2018-03-31]. DOI: G00318713. Dostupné z: <https://www.gartner.com/doc/reprints?id=1-4P1DVJ8&ct=180117&st=sb>
- [22] WebSphere platform. *IBM* [online]. [cit. 2018-03-31]. Dostupné z: https://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/overview/intr_ovr.html
- [23] What is SharePoint?. *AIIM* [online]. [cit. 2018-03-31]. Dostupné z: <http://www.aiim.org/What-is-Microsoft-Sharepoint>

- [24] What is SharePoint?. *Microsoft* [online]. [cit. 2018-03-31]. Dostupné z: <https://support.office.com/en-us/article/What-is-SharePoint-97b915e6-651b-43b2-827d-fb25777f446f>
- [25] ORACLE WEBCENTER PORTAL. Oracle [online]. [cit. 2018-03-31]. Dostupné z: <http://www.oracle.com/technetwork/middleware/webcenter/portal/overview/webcenter-portal-datasheet-427750.pdf>
- [26] Oracle WebCenter. *Oracle* [online]. [cit. 2018-03-31]. Dostupné z: <http://www.oracle.com/technetwork/middleware/webcenter/suite/overview/index.html>
- [27] SAP Enterprise Portal: Getting Started. SAP [online]. [cit. 2018-03-31]. Dostupné z: <https://blogs.sap.com/2012/02/16/sap-enterprise-portal-getting-started/>
- [28] SAP EP (Enterprise Portal). *STechies* [online]. [cit. 2018-03-31]. Dostupné z: <https://www.stechies.com/enterprise-portal-ep/>
- [29] ŠTRUP, Ondřej. Property, Asset nebo Facility management?. TBZ-info [online]. 2. 6. 2011 [cit. 2018-03-31]. Dostupné z: <https://www.tzb-info.cz/epc-energy-performance-contracting/7518-property-asset-nebo-facility-management>
- [30] VOLKO, Vladimír. Co je to: "Asset management"?. *Volko Partners* [online]. [cit. 2018-03-31]. Dostupné z: http://www.volko.cz/new/slovník_vykonnosti.php?ID_term=6
- [31] Enterprise Asset Management (EAM). Techopedia [online]. [cit. 2018-03-31]. Dostupné z: <https://www.techopedia.com/definition/13883/enterprise-asset-management-eam>
- [32] Digital Asset Management (DAM). Techopedia [online]. [cit. 2018-03-31]. Dostupné z: <https://www.techopedia.com/definition/1193/digital-asset-management-dam>
- [33] Software Asset Management (SAM). Techopedia [online]. [cit. 2018-03-31]. Dostupné z: <https://www.techopedia.com/definition/27131/software-asset-management-sam>
- [34] Information Technology Asset Management (ITAM). Techopedia [online]. [cit. 2018-03-31]. Dostupné z: <https://www.techopedia.com/definition/17744/information-technology-asset-management-itam>

- [35] Software pro správu IT majetku (IT Asset management Software). *Sociální síť pro business* [online]. [cit. 2018-03-31]. Dostupné z: <https://managementmania.com/cs/software-pro-spravu-it-majetku-it-asset-management-software>
- [36] Overview. *Vaadin* [online]. [cit. 2018-03-31]. Dostupné z: <https://vaadin.com/docs/v7/framework/introduction/intro-overview.html>
- [37] Background. *Vaadin* [online]. [cit. 2018-03-31]. Dostupné z: <https://vaadin.com/docs/v7/framework/introduction/intro-background.html>
- [38] Goals and Philosophy. *Vaadin* [online]. [cit. 2018-03-31]. Dostupné z: <https://vaadin.com/docs/v7/framework/introduction/intro-goals.html>
- [39] Introduction. *Maven* [online]. [cit. 2018-03-31]. Dostupné z: <https://maven.apache.org/what-is-maven.html>
- [40] Apache Tomcat. *Apache Tomcat* [online]. [cit. 2018-03-31]. Dostupné z: <http://tomcat.apache.org/index.html>
- [41] ROUSE, Margaret. LDAP (Lightweight Directory Access Protocol). *Enterprise Mobile Computing news and information* [online]. [cit. 2018-03-31]. Dostupné z: <http://searchmobilecomputing.techtarget.com/definition/LDAP>
- [42] OpenLDAP. *OpenLDAP* [online]. [cit. 2018-03-31]. Dostupné z: <https://www.openldap.org/>

PŘÍLOHY

Příloha A – Konfigurace portletů v portlet.xml	71
Příloha B – Custom Vaadin Portlet.....	74

PŘÍLOHA A – KONFIGURACE PORTLETŮ V PORT- LET.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-
app_2_0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-
app_2_0.xsd http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
version="2.0">

<portlet>
  <description>Administration of Asset Management.</description>
  <portlet-name>AdministrationPortlet</portlet-name>
  <display-name>Administration Portlet</display-name>
  <portlet-class>
    cz.upce.fei.app.CustomVaadinPortlet
  </portlet-class>
  <init-param>
    <name>UI</name>
    <value>cz.upce.fei.portlets.AdministrationPortlet</value>
  </init-param>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>view</portlet-mode>
  </supports>
  <portlet-info>
    <title>Administration Portlet</title>
    <short-title>
      Administration of Asset Management
    </short-title>
    <keywords>asset, management, administration</keywords>
  </portlet-info>
  <security-role-ref>
    <role-name>administrator</role-name>
  </security-role-ref>
  <security-role-ref>
    <role-name>guest</role-name>
  </security-role-ref>
  <security-role-ref>
    <role-name>power-user</role-name>
  </security-role-ref>
  <security-role-ref>
    <role-name>user</role-name>
  </security-role-ref>
</portlet>

<portlet>
  <description>Asset Portlet.</description>
  <portlet-name>AssetPortlet</portlet-name>
  <display-name>Asset Portlet</display-name>
  <portlet-class>
    cz.upce.fei.app.CustomVaadinPortlet
  </portlet-class>
  <init-param>
    <name>UI</name>
```

```

        <value>cz.upce.fei.portlets.AssetPortlet</value>
    </init-param>
    <supports>
        <mime-type>text/html</mime-type>
        <portlet-mode>view</portlet-mode>
    </supports>
    <portlet-info>
        <title>Asset Portlet</title>
        <short-title>Asset Portlet</short-title>
        <keywords>asset, management</keywords>
    </portlet-info>
    <security-role-ref>
        <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>user</role-name>
    </security-role-ref>
</portlet>

<portlet>
    <description>View Portlet.</description>
    <portlet-name>ViewPortlet</portlet-name>
    <display-name>View Portlet</display-name>
    <portlet-class>
        cz.upce.fei.app.CustomVaadinPortlet
    </portlet-class>
    <init-param>
        <name>UI</name>
        <value>cz.upce.fei.portlets.ViewPortlet</value>
    </init-param>
    <supports>
        <mime-type>text/html</mime-type>
        <portlet-mode>view</portlet-mode>
    </supports>
    <portlet-info>
        <title>View Portlet</title>
        <short-title>View Portlet</short-title>
        <keywords>view, asset, management, read, only</keywords>
    </portlet-info>
    <security-role-ref>
        <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>user</role-name>

```



```
    </security-role-ref>
</portlet>

</portlet-app>
```

PŘÍLOHA B – CUSTOM VAADIN PORTLET

```
package cz.upce.fei.app;

import com.vaadin.server.DeploymentConfiguration;
import com.vaadin.server.ServiceException;
import com.vaadin.server.VaadinPortlet;
import com.vaadin.server.VaadinPortletService;
import com.vaadin.server.VaadinRequest;

public class CustomVaadinPortlet extends VaadinPortlet {

    private static final long serialVersionUID = -13615405654173335L;

    private class CustomVaadinPortletService extends VaadinPortletService {
        private static final long serialVersionUID = -
        6282242585931296999L;

        public CustomVaadinPortletService(final VaadinPortlet portlet,
            final DeploymentConfiguration config) throws ServiceException
        {
            super(portlet, config);
        }

        @Override
        public String getStaticFileLocation(final VaadinRequest request) {
            return request.getContextPath();
        }
    }

    @Override
    protected VaadinPortletService createPortletService(final Deploy-
    mentConfiguration deploymentConfiguration) throws ServiceException
    {
        final CustomVaadinPortletService customVaadinPortletService = new
        CustomVaadinPortletService(this, deploymentConfiguration);
        customVaadinPortletService.init();
        return customVaadinPortletService;
    }
}
```