

Gymnázium, Praha 6, Arabská 14

Předmět Programování, vyučující Jan Lána

Aplikace pro sdílení knih

maturitní projekt



2020

Viktor Vašek 4.E

Obsah

| | |
|--|----|
| Úvod | 5 |
| Anotace | 5 |
| Annotation | 5 |
| Zadání projektu | 5 |
| Předmluva k práci | 6 |
| Použité nástroje | 7 |
| Google Web Toolkit | 7 |
| Eclipse | 7 |
| Apache Tomcat | 7 |
| Zdrojový kód | 7 |
| Struktura kódu | 7 |
| Diagram uživatelských obrazovek | 8 |
| Vzhled webové stránky | 9 |
| Login | 10 |
| Login klientská část | 10 |
| Přenos informací z formuláře na server | 11 |
| Login serverová část | 11 |
| Logout | 11 |
| Register klientská část | 11 |
| Register serverová část | 11 |
| Upload Souborů | 12 |
| Upload Souborů klientská část | 12 |
| Přenos souboru z klientské části na server | 12 |
| Upload Souborů serverová část | 12 |
| Ukládání dat o souboru | 12 |
| Sdílení souborů | 13 |
| Sdílení souborů klientská část | 13 |
| Sdílení souborů serverová část | 13 |

| | |
|----------------------------------|----|
| Stahování souborů | 14 |
| Stahování souborů klientská část | 14 |
| Stahování souborů serverová část | 14 |
| Uložení webového nastavení | 14 |
| Instalace | 14 |
| Nezbytné programy pro instalaci | 14 |
| Konfigurace cest | 15 |
| Spuštění aplikace | 16 |
| Uživatelská dokumentace | 16 |
| Login a Register | 16 |
| Nahrávání souborů | 17 |
| Sdílení souborů | 17 |
| Stahování souborů | 17 |
| Závěr | 18 |

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V dne

Viktor Vašek

Úvod

Anotace

Cílem práce je vytvoření webové aplikace pro sdílení e-knih. Jedná se o server-klient aplikaci, kde má každý uživatel svůj vlastní účet, který si musí v aplikaci vytvořit. Aplikace umožňuje nahrávání souborů na server a jejich sdílení s jinými uživateli. Každý uživatel může na serveru skladovat své soubory, nebo si může ze serveru stáhnout soubory s ním sdílené.

Annotation

The aim of the work was to create a web application for sharing e-books. It is a server-client application, where each user has their own account, which they must create in the application. The application allows you to upload files to the server and share them with other users. Each user can store their files on the server or download files shared with them from the server.

Zadání projektu

Uživatel se přihlásí na server pomocí aplikace a provede login(bude možné provést reset hesla).

Po přihlášení na server bude mít možnost odeslat knihy na server a zobrazit své soubory(které tam již poslal).

Dále bude mít uživatel možnost sdílet soubory s ostatními uživateli

Každý uživatel bude moci přidat knihu do veřejné knihovny, která bude viditelná pro veškeré uživatele, nebo ji bude moci přidat do své soukromé knihovny, která bude viditelná jen pro něj, nebo ji může sdílet s vybranými uživateli, jako sdílení souborů(do veřejné knihovny půjde přidat pouze čtečkové formáty jako např. pdf).

Do veřejné knihovny bude možné přidat pouze nové soubory(takže nebude možné, že se tam vyskytuje 2X stejná kniha ve stejném formátu).

Aplikace bude moci také otevřít některé knihy na dálku bez stahování, nebo si je bude moci uživatel stáhnout.

Předmluva k práci

Mou motivací pro tvorbu této práce bylo několik faktorů. Jedním z nich byla touha vytvořit vlastní webovou aplikaci pro sdílení souborů. Původně jsem měl představu o tom, že vytvořím něco podobného jako je ulož.to, pro sdílení knih. Během vývoje se vzhled aplikace začal značně měnit oproti mé původní představě, což bylo vytvořit online knihovnu. Finální verze aplikace je použitelná pro sdílení knih, ale neviděl jsem důvod proč by přes aplikaci nebylo možné sdílet i jiné soubory. Ve finální verzi je navíc oproti předpokládané funkcionalitě možné aplikaci použít pro posílání libovolných souborů konkrétním uživatelům na dálku a jediné co je pro tuto funkci potřeba je rychlé a jednoduché vytvoření účtu.

Použité nástroje

Google Web Toolkit

Jedná se o open source framework původně vyvinutý společností Google. Framework je určený pro vývoj klient-server aplikací. Další jedinečnou funkcí GWT je možnost psaní frontendu v javě, který je pomocí ant souborů překládán na javascript. Tato funkce pro mě byla jedním z rozhodujících důvodů, proč jsem si vybral GWT před jinými frameworky. Konkrétní verze kterou jsem použil je verze GWT-2.8.2

Eclipse

IDLE Eclipse jsem si vybral k vývoji díky několika výhodám které nabízí. Jednou z očividných výhod je fakt, že má import pro GWT a celkově je s frameworkem velmi kompatibilní. Další výhodou Eclipse je masivní množství přidáných funkcí co nabízí, což mi značně zjednodušilo vývoj a snížilo množství času co jsem strávil u command linu. Verze eclipse co jsem použil pro vývoj je verze eclipse IDE 2019-09

Apache Tomcat

Původně jsem neměl v plánu Tomcat použít, ale během vývoje jsem začal mít problémy s předinstalovanými servery pro deployment v eclipse a Tomcat jsem shledal použitelnějším. Více o Tomcatu se zmíním více v kapitole instalace, pro vývoj jsem použil verzi - 7.0.96

Zdrojový kód

Struktura kódu

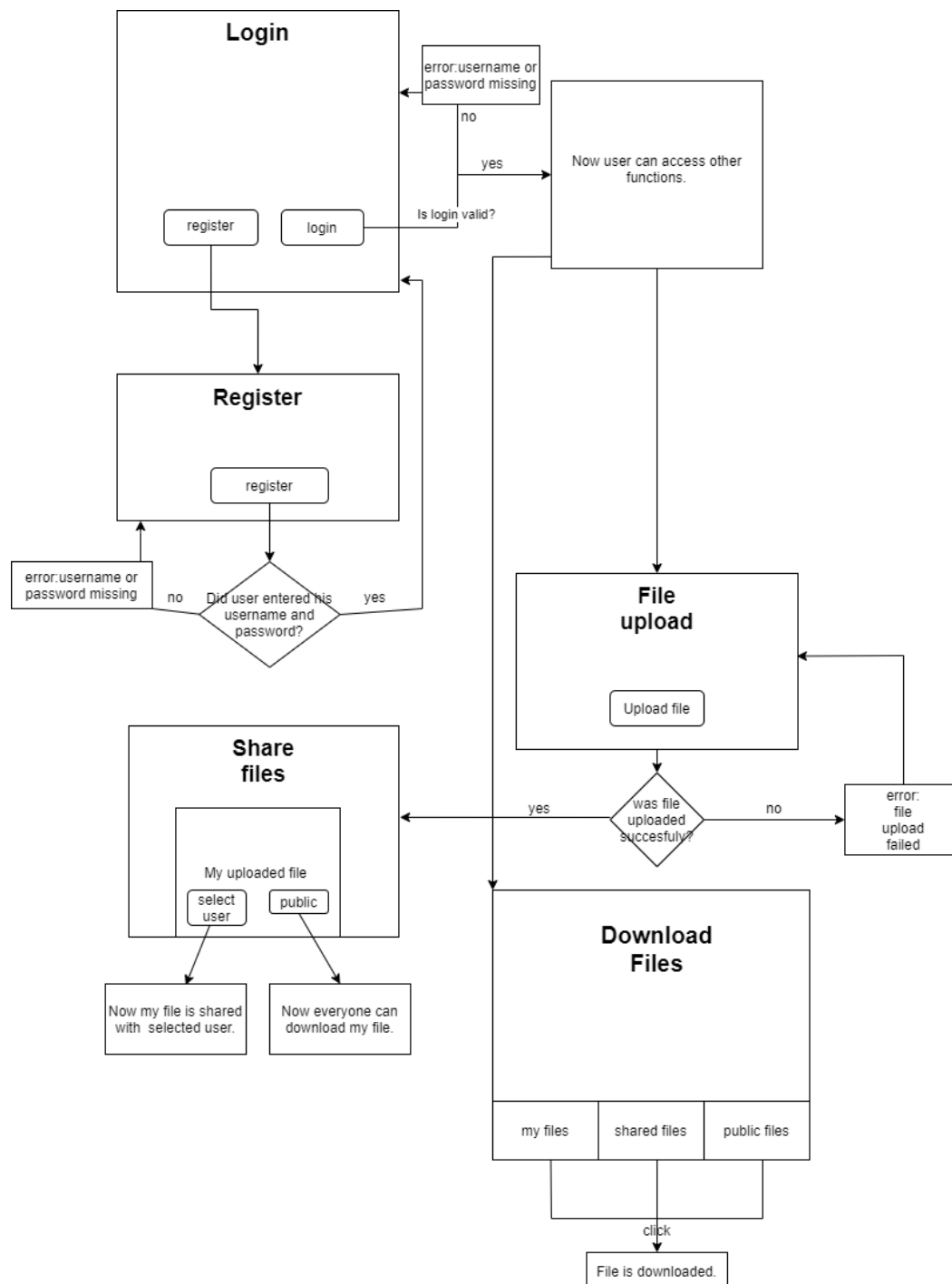
Aplikace je rozdělena na dvě hlavní části, klientskou část a část serverovou. V klientské části se nachází hlavní klientská třída FileSharingApp. Dále se zde nachází několik

pomocných interfaců. Serverová část obsahuje třídy zabývající se zpracováním dat získaných z klientské části a o jednotlivých funkcích se zmíním níže. Sdílená část zdrojového kódu obsahuje konstanty sdílené klientskou i serverovou částí.

Diagram uživatelských obrazovek

Diagram nacházející se o stránku dále zobrazuje jednoduchou ilustraci jednotlivých obrazovek, co jsou v aplikaci a jak jsou na sebe vázané

Diagram byl vytvořen pomocí programu: <https://app.diagrams.net/>



Vzhled webové stránky

Uživatelské rozhraní je nastavené třídě FileSharingApp. Každé okno má svůj link a container.

```
36 public class FileSharingApp implements EntryPoint {
37     private static Logger logger = Logger.getLogger(FileSharingApp.class.getName());
38     private static final String homeLinkId = "homeLink";
39     private static final String loginLinkId = "loginLink";
```

Poté se pomocí GWT funkce anchor vygenerují v horní části stránky linky, které při kliknutí odkáží na příslušný text uložený v HTML dokumentu. V případě že se jedná o stránku s formulářem, nebo tlačítka tak také odkáže na funkci která tyto objekty generuje.

```
Anchor.wrap(RootPanel.get(FileSharingApp.LoginLinkId).getElement()).addClickHandler(new ClickHandler() {
    @Override
    public void onClick(ClickEvent event) {
        FileSharingApp.logger.log(Level.INFO, "Login page link clicked");
        navigateToTab(FileSharingApp.LoginLinkId, FileSharingApp.LoginContainerId);
        configureLoginContainerContent();
    }
});
```

Nastavení vzhledu písma a tlačítek je uloženo v css souboru z kterého jsou tyto hodnoty využívány.

Login

Aplikace si uchovává hodnotu o stavu přihlášení uživatele. Pokud není uživatel přihlášen, tak mu aplikace neumožní přístup k dalším funkcím, jako je například FileUpload.

Uživatelské informace jsou uloženy v třídě user store v Hash mapě. V případě že se uživatelské jméno s heslem shodují, tak se hodnota o přihlášení změní nenulovou. Poté má uživatel přístup k dalším funkcím aplikace. Uživateli se také přidělí jeho cookie, podle které se identifikuje vlastnictví souborů. V případě, že se jedná o první přihlášení tak se cookie vygeneruje.

Login klientská část

Uživateli se vygeneruje formulář do kterého je možné zadat uživatelské jméno a heslo.

V okamžiku kdy kdy uživatel klikne, tak se provede GWT funkce submit. V případě že uživatel nevyplní jeden z boxů, tak se funkce submit neprovede.

V případě neúspěšného loginu se uživateli napíše zpráva o selhání loginu a musí se zkusit znovu zalogovat. V případě úspěchu je uživatel zalogován a místo loginScreen se zobrazí PostLoginScreen, která umožňuje uživateli odhlášení.

Přenos informací z formuláře na server

Informace o uživatelském jméně a heslu jsou přenášeny pomocí interfacu AuthenticationService a AuthenticationServiceAsync. Informace uložené v těchto interfacech jsou poté vytaženy v třídě AuthenticationServiceImpl, která dané informace zpracuje.

Login serverová část

Pro login je použito 5 tříd AuthenticationServiceImpl, která zajišťuje kontrolu údajů z login formuláře. SessionStore, která zajišťuje ukládání sessionID. UserStore, který slouží k ukládání uživatelského jména a hesla pomocí Hash mapy.

```
private static Map<String, String> userStore;  
static {  
    userStore = new HashMap<String, String>();  
    userStore.put("user1", "Password01");  
    userStore.put("user2", "Password02");  
    userStore.put("user3", "Password03");  
    userStore.put("user4", "Password04");  
}
```

zmiňovaná Hash mapa

CustomHttpServlet, který slouží ke komunikaci s klientskou částí aplikace, managementu Cookies a jiných údajů, které jsou později používány pro identifikaci vlastníka souborů a LoginController, který zajišťuje přiřazení příslušné cookie uživateli, a také zajišťuje pomocí servletu odpověď o úspěchu či neúspěchu loginu klientské části aplikace.

Logout

Po úspěšném loginu se načte logout obrazovka, s možností se odhlásit. Poté co uživatel klikne na logout link, tak se zavolá serverová třída LogoutController, která pomocí Servletu odstraní cookie a odhlásí uživatele.

Register klientská část

V klientské části se vygeneruje formulář velmi podobný formuláři pro login. Na tento formulář se uživatel dostane pomocí linku v login formuláři. Po úspěšném vyplnění formuláře se opět zobrazí login obrazovka.

Register serverová část

Pro přenos informací z formuláře jsou opět použity interfacy AuthenticationServiceAsync a AuthenticationServiceImpl. Dále jsou použity dvě třídy. RegisterController, který pomocí servletu komunikuje s klientskou částí aplikace a zpracovává výsledky metody addUser z userStore. V případě, že metoda addUser proběhne úspěšně tak pošle klientské části aplikace potvrzení o úspěšné registraci. V třídě userStore se nachází metoda addUser, která zkontroluje, jestli uživatel vyplnil formulář, nebo si nevolil již používané username. Poté přidá uživatele do Hash mapy.

Upload Souborů

Poté co se uživatel úspěšně zaloguje, tak může použít funkci FileUpload

Upload Souborů klientská část

Klientská část má opět formulář s možností kliknout na panel upload. Poté co uživatel klikne na tento panel tak se provede gwt funkce upload, která umožní vybrat libovolný soubor z prohlížeče souborů.

Přenos souboru z klientské části na server

Přenos zajišťuje interface FileUploadService a FileUploadService. Pro přenos je také použita gwt knihovna RemoteService

```
package com.google.gwt.fileSharingApp.client;

import com.google.gwt.user.client.rpc.RemoteService;
import com.google.gwt.user.client.rpc.RemoteServiceRelativePath;

/**
 * The client-side stub for the RPC service.
 */
@RemoteServiceRelativePath("uploadFile")
public interface FileUploadService extends RemoteService {
    String uploadFile(String name) throws IllegalArgumentException;
```

FileUploadService interface

Upload Souborů serverová část

Poté co se soubor dostane na server tak je zpracován třídou FileUploadController. První krok co třída provede je že zavolá metodu getAuthenticationdetails. Tato metoda porovná jestli se sessionId a cookie shodují, čímž se potvrdí identita uživatele. Poté si třída zjistí jméno aplikace. Dále si vytáhne zaslaný soubor a uloží ho do složky uploadedFiles, kde vytvoří adresář se stejným jménem jako má uživatel co soubor zaslal. Pokud daný adresář neexistuje tak se vyrobí a do daného adresáře se soubor uloží. V případě, že jsme již soubor se stejným jménem uložili, tak se předchozí soubor smaže.

Ukládání dat o souboru

Kromě samotného ukládání souboru na disk tak se také zavolá metoda třídy FileStore addFileForUser. Data o souborech jsou uložena ve formátu JSON pomocí JSONObject. Třída UserFile obsahuje stejnojmenný konstruktor. O každém souboru jsou ukládány 4 informace. Jméno souboru, vlastník souboru, jestli je soubor sdílený s jiným uživatelem nebo je veřejný. Třída UserFile také obsahuje metody pro ověření těchto údajů, které jsou používány v jiných třídách. Tyto informace o souborech jsou uloženy v třídě FileStore v Hash mapě filestoreDB odkud jsou jinými třídami zpřístupňovány.

```
private static Map<String, ArrayList<UserFile>> fileStoreDB;
static {
    fileStoreDB = new HashMap<String, ArrayList<UserFile>>();
}
```

zmiňovaná Hash mapa

Sdílení souborů

Tuto část aplikace je možné použít po nahrání prvního souboru. Pro uživatele je možné vybrat sdílení s jiným uživatelem co používá aplikaci nebo aplikaci zveřejnit pro všechny.

Sdílení souborů klientská část

Klientská část aplikace načte JSON tabulku ze serveru, co obsahuje data o sdílení souborů. Tuto tabulku obsahující JSON informace se převede na arrayList složený ze stringů.

```
JSONArray userFiles = responseAsJSON.get("userFiles").isArray();
for (int i = 0; i < userFiles.size(); i++) {
    final String fileName = userFiles.get(i).isObject().get("fileName").isString()
        .toString().replace("\\", "");
    final String sharedBy = userFiles.get(i).isObject().get("owner").isString()
        .toString().replace("\\", "");
    final boolean isPublic = userFiles.get(i).isObject().get("public").isBoolean()
        .booleanValue();
    final JSONArray sharedWith = userFiles.get(i).isObject().get("sharedWith")
        .isArray();
    List<String> sharedWithAsList = new ArrayList<String>();
    for (int j = 0; j < sharedWith.size(); j++) {
        sharedWithAsList.add(sharedWith.get(j).isString().stringValue());
    }
}
```

zmíněná část kódu

Poté je vytvořen ListBox a CheckBox, což jsou gwt třídy umožňující vytvářet uživatelské rozhraní pro frontend. V Listboxu se poté porovnají data o uživateli co mají již daný soubor sdílený a uživatele, kteří daný soubor sdílený nemají a pro uživatele co soubor ještě nemají sdílený je nabídnuta možnost sdílení. V CheckBoxu se pouze nastaví formátování a hodnota pro zaškrtnutí. Poté se přidá Update button, jehož onClick akce aktualizuje data v tabulce, z které jsme daná data původně vytáhli podle změn co uživatel provedl.

Sdílení souborů serverová část

Serverovou část pro sdílení souborů řeší třídy ShareFilesTabDataController a UpdateShareSettingsController. Obě třídy používají stejný autentizační protokol, který je zmíněn v kapitole Upload souborů serverová část. V třídě ShareFilesTabDataController se nachází metody sloužící jako protiváha k metodám v klientské části co extrahují JSON data z Hash mapy fileStoreDB do stringů a polí, také se z této třídy pomocí servletu posílají

nezbytné data do klientské části kde jsou upravena. Třída UpdateShareSettingsController slouží k implementaci změn z klientské, části a aktualizuje do Hash mapy fileStoreDB.

Stahování souborů

Tato část aplikace umožní uživateli stahovat soubory ze serveru, tak že poté co načte soubory ke kterým má uživatel přístup, tak k nim vytvoří download link.

Stahování souborů klientská část

Opět se načtou data o souborech ze serveru podobně jako u klientské části sdílení souborů. Poté co se získají data o souborech vlastněných uživatelem, sdílených souborech a souborech veřejných, tak se vygenerují v příslušné kategorii podle jmen souborů.

Stahování souborů serverová část

Opět je použit stejný autentizační protokol, který je zmíněn v kapitole Upload souborů serverová část. Třída posílá klientské části data o sdílení a jménech souborů.

```
final String fileName = request.getParameter("fileName");  
final String sharedBy = request.getParameter("sharedBy");
```

zmíněná část kódu

Dále se nastaví cesta do složky s uloženými soubory. Poté se pomocí knihovny java.nio a java.io pošle obsah složky klientovi.

Uložení webového nastavení

Veškeré Controllery mají nastavené servlety v souboru web.xml v tomto formátu:

```
<servlet>  
  <servlet-name>loginServlet</servlet-name>  
  <servlet-class>com.google.gwt.fileSharingApp.server.LoginController</servlet-class>  
</servlet>  
<servlet-mapping>  
  <servlet-name>loginServlet</servlet-name>  
  <url-pattern>/filesharingapp/login</url-pattern>  
</servlet-mapping>
```

Instalace

Nezbytné programy pro instalaci

ApacheTomcat můžete stáhnout na: <https://tomcat.apache.org/download-70.cgi>

Dále je nezbytné nainstalovat GWT knihovny <http://www.gwtproject.org/download.html>

Odkazy na instalaci dalších nezbytných knihoven:

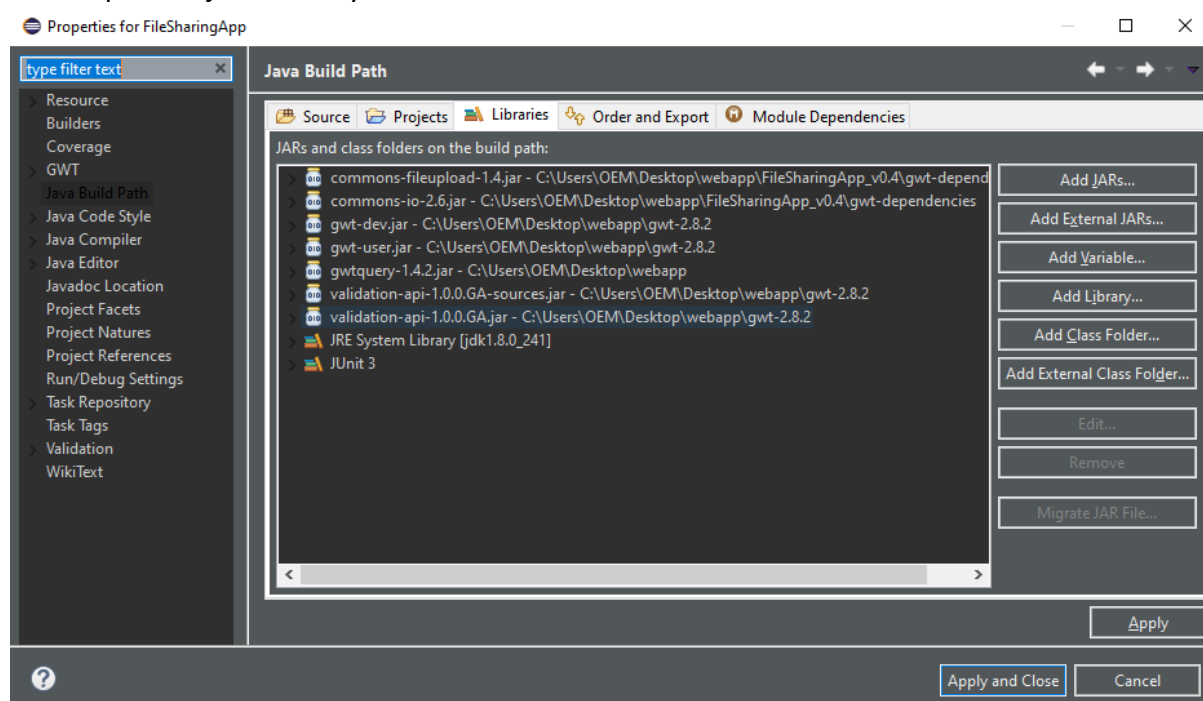
<https://jar-download.com/artifacts/commons-fileupload/commons-fileupload/1.4/source-code>

http://commons.apache.org/proper/commons-io/download_io.cgi

<https://jar-download.com/artifacts/javax.validation/validation-api/1.0.0.GA/source-code>

Konfigurace cest

Každé knihovně je potřeba nastavit cestu k aplikaci, já jsem použil eclipse, ale je možné použít i jiné metody.



ukázka nastavení cest pochází z Eclipse <https://www.eclipse.org/>

Pro úspěšné spuštění aplikace je dále nutné změnit cesty ke knihovnám v souboru build.xml

Dále je nezbytné v Tomcatu vytvořit adresář uploadedFiles. Poté Tomcatu v adresáři webapps vytvoříme adresář filesharingapp.

Po rozbalení FileSharingApp najdete adresář war a přepokopírujte veškerý obsah adresáře do nově vytvořeného adresáře v Tomcatu filesharingapp.

V případě problémů se spuštěním navštivte návod na používání Tomcatu: <https://tomcat.apache.org/tomcat-7.0-doc/windows-service-howto.html>

Spuštění aplikace

Aplikaci spustíme pomocí příkazové řádky příkazem:

cd C:\Users\OEM\Desktop\webapp\apache-tomcat-7.0.96\bin - *zde zadejte vaši cestu k Tomcatu*

catalina.bat run

Tímto příkazem se spustí aplikace na webové adrese <http://localhost:8080/filesaringapp/>,

Uživatelská dokumentace

Aplikace vás přivítá domovskou obrazovkou, která popisuje k čemu aplikace je.

Login a Register

Z tohoto okna klikněte na login link. Na této obrazovce se můžete přihlásit pokud jste si již vytvořili vlastní účet jestli účet nemáte klikněte na register link a zaregistrujte se. Poté se můžete přihlásit a budete mít přístupné veškeré funkce aplikace. Jestli jste přihlášení, tak se vám v pravém horním rohu objeví vaše uživatelské jméno.

File Sharing Application

[Home](#)[Login](#)[Upload Files](#)[Share Files](#)[Download Files](#)

Login with your user name and password

Not having an account? [Register](#) please

Please enter your username

Please enter your password

Nahrávání souborů

Klikněte na link Upload Files. Zde klikněte na tlačítko upload files. Otevře se vám prohlížeč souborů, zde vyberete soubor který chcete nahrát. Místo zprávy soubor nevybrán se vám zobrazí jméno vašeho souboru. Klikněte na tlačítko submit, čímž nahrajete váš soubor. V případě úspěšného nahrání se objeví textová zpráva File upload success.

Logged in as a

[Home](#) [Login](#) [Upload Files](#) [Share Files](#) [Download Files](#)

Upload your files

Select the file you would like to upload and hit submit

Sdílení souborů

Poté co nahrajete soubor, tak je viditelný pouze pro vás. Jestli chcete využít aplikaci pouze jako vzdálené úložiště, tak funkci Share Files nepoužívejte. V případě, že chcete soubor sdílet s konkrétním uživatelem tak klikněte na link Share Files, poté vyberete daného uživatele kliknutím na okno, kde je defaultně vybrán user1 a kliknete na tlačítko Update. Jestli chcete sdílet váš soubor pro veškeré uživatele, tak zaškrtněte checkbox public a klikněte na tlačítko Update.

File Sharing Application

Logged in as a

[Home](#) [Login](#) [Upload Files](#) [Share Files](#) [Download Files](#)

Share files with other users

You can share your files with specific users or make them public

Your files

view.html

☐ Public

Stahování souborů

Klikněte na link Download Files. Zde se vám zobrazí vaše soukromé soubory. Soubory sdílené s vámi od jiných uživatelů a soubory veřejné. soubor stáhnete tím, že na něj kliknete.

Tím zahájíte stahování souboru, který se vám zobrazí v dolní záložce prohlížeče, kde ho můžete otevřít. Stáhnutý soubor také naleznete v stáhnutých souborech na vašem počítači.

Download files

You can download [here](#) your files or files shared with you

Your files

[view.html](#)

Shared files

[odpovědi test.txt](#)
[notes \(1\).txt](#)

Public files

[kvety-zla.pdf](#)

Závěr

Snad mohu říct, že se mi podařilo splnit skoro vše v zadání co jsem slíbil. Jedním z největších problémů, co jsem měl byl proces učení používání GWT frameworku. GWT framework sice umožňuje psát frontend v javě, ale používá velké množství vlastních knihoven, které je nezbytné umět používat, což mi zabralo více času než by nejspíše mělo. Avšak s celkovým vzhledem aplikace a uživatelským rozhraním jsem relativně spokojen. Za svůj největší neúspěch považuji fakt, že se mi nepodařilo vytvořit reset hesla. Avšak v ostatních ohledech je aplikace zcela funkční a spolehlivá. Mezi jeden z nedostatků aktuální verze patří způsob jakým jsou soubory ukládány a v budoucnu zvažuji připojení databáse pro jejich ukládání. Další věcí která by šla vylepšit je zabezpečení, protože aktuální způsob ukládání souborů by byl v praktickém využití velmi náchylný k virovému útoku. Celkově jsem se svou prací spokojen a myslím si že je relativně solidní.