|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| upb.png | Universitatea Politehnica București | acs.png | Facultatea de Automatică și Calculatoare | CS-Logo-transp.png | Departamentul de Calculatoare |

**Dezvoltarea aplicațiilor Internet**

Documentare aplicație DAI  
Aplicație pentru concedii

Versiunea 0.1

Autor: Simona Roboiu

Seria: ISI1

# Cuprins

[I. Cuprins 2](#_Toc472726782)

[II. Introducere 3](#_Toc472726783)

[II.1. Subiectul abordat 3](#_Toc472726784)

[II.2. Organizarea documentului 3](#_Toc472726785)

[III. Realizarea aplicației 4](#_Toc472726786)

[III.1. Analiza cerințelor 4](#_Toc472726787)

[III.1.1. Gestiunea utilizatorilor 4](#_Toc472726788)

[III.1.2. Funcționalitățile aplicației 4](#_Toc472726789)

[III.1.3. Monitorizare și jurnalizare 4](#_Toc472726790)

[III.2. Proiectarea aplicației 4](#_Toc472726791)

[III.2.1. Tehnologii și produse folosite 5](#_Toc472726792)

[III.2.2. Decizii de proiectare 10](#_Toc472726793)

[III.3. Manualul de utilizare 10](#_Toc472726794)

[III.3.1. Acces public 10](#_Toc472726795)

[III.3.2. Utilizator autentificat 10](#_Toc472726796)

[III.3.3. Administrator 12](#_Toc472726797)

[III.4. Instalarea aplicației 12](#_Toc472726798)

[III.4.1. Cerințe preliminare 12](#_Toc472726799)

[III.5. Testarea aplicației 12](#_Toc472726800)

[IV. Concluzii și dezvoltări ulterioare 13](#_Toc472726801)

[IV.1. Concluzii 13](#_Toc472726802)

[IV.2. Dezvoltări ulterioare 13](#_Toc472726803)

[V. Bibliografie 14](#_Toc472726804)

[VI. Anexe 15](#_Toc472726805)

# Introducere

Aplicatia dezvoltata pentru disciplina "*Dezvoltarea aplicațiilor Internet*" este o unealta utila companiilor pentru gestiunea concediilor angajatilor. Aceasta vine atat in ajutorul companiilor, cat si al angajatilor. Ajutorul este de ambele parti deoarece la nivel de companie se pot adauga zile nationale libere si se pot gestiona zilele de concediu ale angajatilor, iar la nivel de angajat, fiecare isi poate adauga zile de concediu (de odihna, medical, neplatit, special), dar si vizualiza starea lor in timp.

Acest proiect este o aplicatie web, creata cu tehnologii actuale, larg raspandite si in continua dezvoltare, astfel ca se pot adauga sau schimba relativ usor functionalitati in aplicatie. De asemenea, stadiul actual al proiectului a fost atins printr-o munca redusa fata de folosirea altor tehnologii, dar in acelasi timp poate asigura securitatea si functionalitatea corecta a ceea ce este implementat.

## Subiectul abordat

Companiile au nevoie de aplicatii care sa le usureze munca si birocratia cererilor printate de concediu, asa ca am ales sa implementez o aplicatie web pentru gestiunea concediilor angajatilor. In functie de companii, politica numarului de zile de CO permise pe an pentru un angajat este diferit, dar si modul cand acestea isi pierd valabilitatea de pe un an pe altul, si nu in ultimul rand sa nu uitam de procedura de aprobare a cererilor. In mare parte aplicatia raspunde adaptiv acestor probleme si incearca sa le rezolve intr-o maniera cat mai clara, simplista si corecta.

Aplicatia dispune de 4 tipuri de utilizator: administrator – cel care gestioneaza angajatii si utilizatorii aplicatiei (adauga, sterge si cauta angajati), angajat contabil – cel care detine ultimul nivel de prelucarare a cererilor de concediu (aproba sau respinge cereri, adauga zile nationale libere in sistem), angajat cu rolul de manager care este de fapt primul nivel in procesul de aprobare a unei cereri (isi poate creea cereri de concediu pentru sine, poate vizualiza starea lor, poate aproba sau respinge cererile angajatilor aflati in subordinea lui) si angajat normal – fara alte functii speciale de manager sau contabil (creeaza cereri de concediu si le vizualizeaza starea acestora).

Proiectul tine cont de zilele de sambata si duminica, dar si de sarbatorile nationale introduse in sistem astfel incat o perioada de concediu sa nu includa si aceste zile ca zile platite sau neplatite de CO.

Tinand cont de existenta utilizatorilor in aplicatie, in mod obligatoriu aplicatia are un sistem de autentificare. Acesta este pe baza de nume de utilizator si parola. Un utilizator se creeaza in acelasi moment cand se creeaza un angajat, prin completarea campurilor de catre administrator.

## Organizarea documentului

Pana acum a fost prezentata ideea aplicatiei si ce functionalitati sunt implementate la nivel general, astfel incat sa se creeze o imagine de ansablu asupra a ceea ce ofera. In continuare va fi prezentat si modul cum a fost construita aplicatia, ce contine in mod concret, care sunt pasii pentru un utilizator. Apoi vor fi descrise tehnologiile folosite si motivele pentru care au fost alese, dar si deciziile de proiectare luate initial si de-a lungul dezvoltarii. De asemenea, pentru o utilizare in mod corect este nevoie si de o descriere a modului de functionare si utilizare in functie de tipul de utilizator, care sunt restrictiile si care sunt elementele permise. Iar pentru a putea beneficia de toate aceste functionalitati este necesara instalarea unor instumente de lucru, de compilare si rulare a aplicatiei ce vor fi detaliate intr-un capitol ulterior. Toate aceste detalii vor creea imaginea complete a acestui proiect.

In final, concluziile vor aduce inca un plus de informatie, alaturi de imbunatatirile ulterioare posibile.

# Realizarea aplicației

## Analiza cerințelor

### Gestiunea utilizatorilor

Utilizatorii sunt de 4 tipuri: administrator – cel care gestioneaza angajatii si utilizatorii aplicatiei (adauga, sterge si cauta angajati), angajat contabil – cel care detine ultimul nivel de prelucarare a cererilor de concediu (aproba sau respinge cereri, adauga zile nationale libere in sistem), angajat cu rolul de manager care este de fapt primul nivel in procesul de aprobare a unei cereri (isi poate creea cereri de concediu pentru sine, pate vizualiza starea lor, poate aproba sau respinge cererile angajatilor aflati in subordinea lui) si angajat normal – fara alte functii speciale de manager sau contabil (creeaza cerei de concediu si le vizualizeaza starea acestora). Initial in sistem este adaugat prin script sql utilizatorul cu rol de administrator. Ulterior acesta va folosi aplicatia pentru a adauga pe rand angajatii in baza de date, dar si contul de utilizator asociat fiecaruia. In functie de rolul asignat, utilizatorul va accesa apoi un anumit tip de pagini web specifice permisiunilor lui.

### Funcționalitățile aplicației

Aplicatia are mecanismul de autentificare in sistem pe baza de nume de utilizator si parola. Apoi in functie de rolul detinut de utilizator i se deschide o interfata specifica functiilor pe care le poate executa. Utilizatorul cu rol de angajat normal poate adauga cereri de concediu pentru o zi sau o perioada, isi poate vizualiza cererile de pana la momentul respectiv si starea lor. Angajatul poate adauga zile de cncediu atata timp cat nu si-a folosit toate zilele disponibile. In plus, pentru acest angajat mai sunt create 2 pagini web inca far functionalitati implementate: una pentru schimbarea parolei si inca una pentru editarea unor informatii de profil (nume de utilizator si pe viitor poza).

Managerul are in plus fata de utilizatorul normal si posibilitatea sa vizualizeze cererile facute de angajatii pe care el ii are in subordine si sa le gestioneze (aprobare sau respingere).

Contabilul vine cu inca o functionalitate, aceea de a adauga in baza de date zile nationale libere.

Admistratorul este cel care poate sa creeze un cont pentru un angajat, sa efectueze operatii de cautare asupra tuturor angajatilor din sistem in functie de anumite criterii de cautare, sa inactiveze un cont si implicit un angajat, dar si sa reactiveze (sunt anumite cazuri in care este nevoie), iar ulterior va fi realizata si functia de editare a informatiilor despre un angajat.

### Monitorizare și jurnalizare

Datele pot fi vizualizate in baza de date de catre administrator. De asemenea in log-ul serverului sunt afisate informatii precum cererile sql catre baza de date (fara parametrii in clar), dar si posibilele erori aparute pe parcursul utilizarii aplicatiei.

## Proiectarea aplicației

Initial a fost creata o schema cu fluxul urmat in aplicatie: cele 3 tipuri principale de utilizatori (fara administrator) si optiunile pe care le poate avea. Acesta se poate observa in figura urmatoare. In imagine nu este inclus si fluxul pentru administrator, care este o entitate separata cu alte functii.

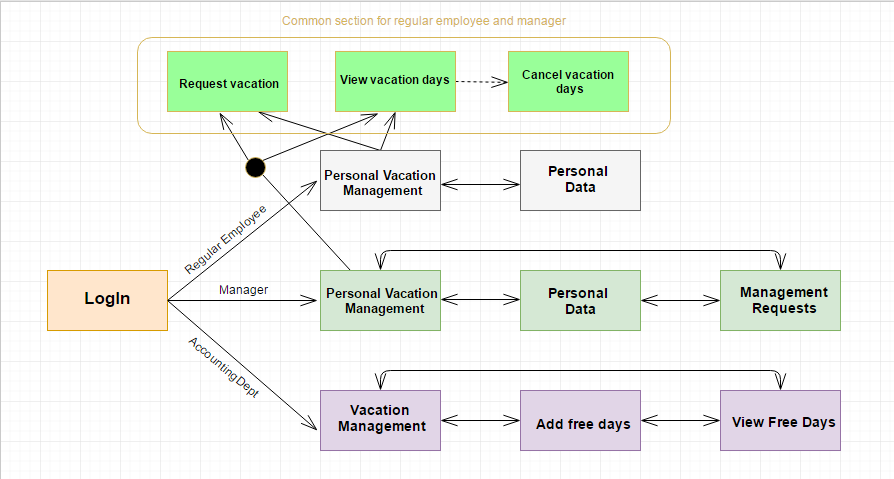


Figure Flux in aplicatie

### Tehnologii și produse folosite

Datorită evoluției rapide a tehnologiilor este de dorit să se folosească medii de dezvoltare și utilitare cât mai noi, dar și stabile și cu o comunitate larg dezvoltată în mediul online. Oferta de astfel de produse este destul de variată, astfel că uneori printre condițiile de selecție a unei tehnologii se numără și preferința strictă a dezvoltatorului.

Analizând cerințele aplicației realizate, diverse criterii de proiectare dar și preferințele personale am creat o aplicație Java Enterprise Edition (Java EE), am folosit un framework open-source numit Primefaces pentru interfața grafică, un server de aplicații Wildfly, o bază de date de la Oracle, versiunea Oracle Express, iar pentru rezolvarea dependențelor la compilare și la rulare am decis să utilizez Apache Maven. Mediul de dezvoltare utilizat a fost Eclipse. În secțiunile următoare există o descriere detaliată a tehnologiilor folosite și a motivelor care au condus la alegerea lor.

Table Tehnologii folosite

| **Tehnologie** | **Descriere scurtă** |
| --- | --- |
| JSF | Tehnologie de programare utilizata pe partea de server care permite crearea unei metode dinamice si independenta de platforma pentru aplicatii web. |
| PrimeFaces | Framework open-source și gratuit pentru dezvoltarea interfețelor grafice în aplicații Java EE bazate pe Java Server Faces. |
| Oracle | Sistem de gestiune al bazelor de date relaționale ce include facilități din tehnologia orientată obiect. |
| Hibernate | Soluție ORM, o implementare a interfeței JPA care permite dezvoltatorilor să creeze mai ușor aplicații care operează cu informații stocate într-o bază de date. |
| Java EE | Limbaj de progamare de nivel înalt, orientat pe obiecte, bazat pe încapsulare, moștenire, polimorfism. |
| Apache Maven | Instrument gratuit și open source care ajută dezvoltatorul de aplicații pe tot procesul de creare a unui proiect software. |
| Jboss Wildfly | Server de aplicații, inițial cunoscut sub numele de JBoss. |

#### JSF

Pentru realizarea interfeței grafice a unei aplicației în care să se afișeze informații venite de la server (din back-end) în cazul tehnologiei Java se poate folosi Java Server Faces, o specificație Java care simplifică construcția interfețelor prin reutilizarea componentelor în pagină.

JSF a fost creat având la bază modelul Swing, dar și alte framework-uri pentru interfețe grafice, oferind dezvoltatorului de aplicații posibilitatea să abstractizeze cererile, răspunsurile și limbajele de adnotare și să se axeze mai mult pe componente, evenimente, interacțiuni, backing bean-uri, simplificându-i în acest mod munca.

O pagină JSF este de fapt o pagină xhtml care definește în head lista cu bibliotecile pentru folosirea tagurilor specifice, iar în corpul fisierului (eng. body) se afla reprezentarea grafică a componentelor din bibliotecile declarate.

Tehnologia Java Server Faces este bazată pe arhitectura Model-View-Controller utilă pentru a separa logica aplicației de nivelul prezentare. O aplicație JSF rulează într-un container web. Componenta destinată View-ului este compusă din paginile JSF din aplicație.

Pentru paginile web, interacțiunea pe partea de client se poate face utilizând HTML sau CSS pentru crearea conținutului static și pentru aspectul paginii și JavaScript pentru a schimba dinamic parți din conținut, însă de multe ori este nevoie de o interacțiune cu baza de date pentru a afișa anumite informații și trebuie ajuns la nivelul EJB-urilor. Acest lucru se poate face prin JSF pe partea de server. Aplicația JSF este o aplicație standard web care primește cereri și creează răspunsuri HTML folosind diverse componente, evenimente, ascultători.

#### PrimeFaces

Primefaces este un framework open-source și gratuit pentru dezvoltarea interfețelor grafice în aplicații Java EE bazate pe Java Server Faces. Principalele caracteristici sunt: simplitatea în utilizare și performanța aplicațiilor dezvoltate folosind Primefaces. Popularitatea acestuia depășește competitorii RichFaces, IceFaces, ADF Faces.

Are peste 100 de componente grafice (editoare HTML, dialoguri, autocompletare, grafice, tabele, etc), permite folosirea tehnologiei AJAX pentru creșterea interactivității paginilor web, dispune de validare pe partea de client, are predefinite inclusiv teme pentru intefața grafică din care dezvoltatorul poate alege. Complexitatea componentelor nu împiedică utilizarea lor deoarece este ascunsă utilizatorului, iar flexibilitatea acestora a fost păstrată.

Pentru a putea crea interfețe folosind tag-urile din biblioteca Primefaces trebuie realizată o configurație minimalistă. Este nevoie de un singur jar care se poate descărca de pe site-ul oficial primefaces, iar apoi trebuie introdus jarul în classpath și inclus spațiul de nume în fișierul xhtml. Cum eu am folosit Maven, am realizat puțin diferit configurația, în sensul că am introdus dependențele în fișierul pom.xml așa cum este explicat și pe site, iar Maven s-a ocupat de rezolvarea acestora. Până la versiunea Primefaces 3.5 trebuie scris în pom.xml și locul public (eng. repository) prin URL unde se poate găsi jar-ul pentru că nu există în depozitul central Maven, însă de la versiunea 4.0 această problemă este rezolvată.

Un avantaj al utilizării acestei biblioteci este documentația foarte bine realizată. Site-ul oferă explicații detaliate cu exemple de utilizare, iar toate aceste informații le putem obține și offline prin cartea în format pdf care cuprinde toate exemplele de pe site dar și informații suplimentare. Mai mult, există o comunitate bine dezvoltată de utilizatori Primefaces unde se pot obține lămuriri și ajutor în caz de nevoie.

#### Oracle

Pentru modulul de gestiune al aplicației am folosit baze de date relaționale. Acestea se bazează pe modelul relațional fundamentat pe teoria seturilor și logica matematică în care datele sunt văzute ca niște atribute stocate în tabele între care se pot stabili relații. Astfel se asigură o redundanță minimă și controlată, independența datelor la nivel logic și fizic, adaptabilitate mare la schimbări, eficiență, regăsirea ușoară și rapidă a datelor.

Oracle este un sistem de gestiune al bazelor de date relaționale ce include facilități din tehnologia orientată obiect. Prima dată a fost dezvoltat de firma Oracle Corporation în anul 1977. La momentul actual, Oracle este cel mai mare furnizor de software de gestiune a datelor.

Având experiență de lucru cu bazele de date de la Oracle, am decis să utilizez o versiune gratis de la Oracle. Soluția aleasă este Oracle Database 11g Express Edition, o versiune simplificată a serverului Oracle. Aceasta a fost lansată în anul 2007, având câteva facilități în plus față de versiunea anterioară Oracle 10g: gestionare mai eficientă a datelor, simplificare a sistemelor prin stocarea datelor în baza de date cu Oracle SecureFiles. Așa cum este descris pe site-ul Oracle SecureFiles reprezintă un mecanism ce oferă cea mai bună soluție pentru stocarea conținutului fișierelor precum imagini, audio, PDF-uri, foi de calcul. În mod tradițional, datele relaționale sunt stocate într-o bază de date, iar datele nestructurate sunt stocate ca fișiere în sistemele de fișiere. SecureFiles a fost special creat pentru a oferi performanțe ridicate pentru fișierele de date, comparabile cu cele ale fișierelor de date tradiționale, păstrând în același timp avantajele bazei de date Oracle. SecureFiles oferă o arhitecură ce conține ce e mai bun din ambele soluții pentru stocarea datelor nestructurate.

Oracle 11g îmi permite să folosesc limbajul SQL pentru accesarea datelor, oferă un driver de conectare la aplicația Java și permite operarea cu baza de date prin intermediul mediului integrat de dezvoltare SQL Developer care este de fapt o interfață grafică pentru utilizatori și administrator prin care se pot realiza operațiile dorite foarte ușor. Alegerea nu are o importanță majoră pentru aplicația dezvoltată de mine deoarece se poate adapta ușor la altă soluție (de exemplu SQL Server). Avantajele au fost cu precădere pentru manipularea și testarea datelor în timpul implementării.

Pentru a utiliza Oracle 11g trebuie descărcat executabilul pentru instalare de pe site-ul Oracle, apoi rulat. Se deschide o interfața grafică care asistă utilizatorul pentru a seta diverși parametri (parola de SYS, de SYSTEM). A mai fost nevoie de descărcarea pachetului pentru SQL Developer. Apoi trebuie urmați pașii din documentația de instalare pentru crearea unui nou utilizator oferindu-i drepturile de acces dorite, pentru crearea unei noi conexiuni și în final a tabelelor necesare din baza de date.

Aplicația Java se conectează la baza de date cu ajutorul serverului de aplicație. Setările le-am realizat prin modificări de conținut în directorul Wildfly. A fost nevoie de un driver de conectare ojdbc6.jar și de un fișier xml asociat jar-ului. Apoi trebuie specificați în fișierul standalone.xml câțiva parametri precum: conexiunea, driverul, url-ul de acces și alte câtevă informații după modelul existent în arhiva Wildfly, informații utile găsindu-se și în mediul online.

#### Hibernate

Hibernate este o soluție ORM, o implementare a interfeței JPA care permite dezvoltatorilor să creeze mai ușor aplicații care operează cu informații stocate într-o bază de date. Poate fi folosit în orice aplicație care suportă JPA, atât pe platformele Java SE cât și Java EE.

Acest framework este o punte de legătură între bazele de date relaționale care stochează informații sub formă tabelară, ce suportă definirea de relații între tabele și aplicațiile Java dezvoltate după modelul obiect care reprezintă de fapt un graf de obiecte interconectate. Hibernate permite crearea de clase persistente implementate după specificațiile tehnicii de programare orientate pe obiect care include moștenirea, polimorfismul, compoziția, ierarhizarea. Pe lângă asigurarea persistenței, Hibernate realizează și interogarea eficientă a datelor din baza de date prin intermediul limbajului Java.

Soluțiile ORM, cum este și cazul Hibernate, abstractizează tehnologia bazei de date folosită. Codul Java scris conform specificațiilor Hibernate funcționează indiferent de produsul specific utilizat pentru a stoca date, fără a mai fi nevoie de alte modificări pentru a rezolva diferențele dintre bazele de date suportate (Oracle, MySQL, DB2, PostgreSQL, FrontBase, etc).

Hibernate este o soluție scalabilă în orice mediu de dezvoltare, flexibilă, stabilă, foarte configurabilă și extensibilă ce asigură calitatea operațiilor efectuate.

#### Java EE

Pentru soft-ul web dezvoltat de mine am folosit platforma Java Platform Enterprise Edition (Java EE), cu versiunea Java 8. Aceasta oferă o interfață pentru programarea de aplicații (eng. Applications Programmng Interface - API) și un mediu de rulare pentru dezvoltarea de programe pentru intreprinderi, servicii de rețea, servicii web, la o scară largă. Se poate folosi pentru lucru cu baze de date prin maparea obiect – relație, pentru servicii distribuite și în general conține elemente ce rulează pe servere de aplicații. Standardele Java EE pentru componente includ Enterprise Java Bean, servleți, Java Server Pages, Java Server Faces și alte tehnologii pentru servicii web. Se pune accentul pe oportunitatea ca dezvoltatorul să se axeze mai mult pe logica aplicației decât pe infrastructură, integrare și alte elemente de configurare.

#### Apache Maven

Apache Maven este un instrument gratuit și open source care ajută dezvoltatorul de aplicații pe tot procesul de creare a unui proiect software. Principalele acțiuni realizate vizează structura inițială a proiectului și rezolvarea dependențelor pentru compilare. Acesta ajută la standardizarea lucrului și la reducerea timpului de a scrie și de a menține scripturi.

Acest tool este implementat în Java, deci este independent de platformă, utilizarea în afara unui IDE necesitând doar descărcarea arhivei, dezarhivarea ei, adăugarea căii către directorul bin în variabila de mediu PATH și existența unei versiuni de JDK (Java Development Kit) către care indică variabila de mediu JAVA\_HOME, apoi se poate lucra direct din linie de comandă.

Maven este utilizat în mediile de dezvoltare în care sunt folosite pentru compilare sisteme precum Ant și GNU make . Acesta simplifică procedeul de construcție și de administrare pentru orice aplicație. Automatizarea procesului de realizare a structurii folderului inițial pentru aplicații Java este posibilă prin arhetipurile disponibile. Pe lângă această caracteristică, tool-ul efectuează compilarea codului sursă și rularea testelor, împachetarea și implementarea produsului final în arhive de tip JAR (Java Archive) sau WAR (web application archive) în funcție de nevoie și tipul aplicației. Convențiile de configurare și arhetipurile disponibile sunt inspirate din modelele de dezvoltare din viața reală, din companii, existând șabloane de creare ce pot fi folosite pentru aproape toate tipurile de aplicații, de la cea mai simplă aplicație Java la cea mai complexă aplicație web.

Instrumentul oferă un sistem de construcție prin fișierul POM.xml (Project Object Model) în care se găsesc informații detaliate despre proiect, dependințe, versiuni, managementul configurării, resurse pentru testare. Acesta este un fișier xml plasat în directorul principal al proiectului. Nu trebuie declarate toate jar-urile de care are nevoie aplicația direct sau indirect, trebuie doar specificată biblioteca și versiunea. Acest mod de lucru prin fișierul POM este o abordare declarativă cu descrierea conținutului și a structurii proiectului, nu prin fișiere de compilare tradiționale sau prin abordarea bazată pe sarcini așa cum este în cazul sistemului Ant.

Într-un proiect se pot folosi chiar și dependențe din alte proiecte prin folosirea Maven-ului, el fiind cel care se ocupă inclusiv de compilarea celorlalte proiecte dacă e nevoie și de rezolvarea dependențelor necesare. Dependențele pot fi încărcate dintr-un sistem local de fișiere, de pe internet sau dintr-un loc public (eng. public repository), din depozitul central Maven deținut de compania Sonatype în care în mod implicit Maven caută bibliotecile cerute.

De neglijat nu este nici posibilitatea de migrare transparentă pentru funcținalități noi și nici faptul că integrarea de plug-in-uri se realizează foarte ușor pentru că ideea de bază a Maven-ului a fost extensibilitatea deoarece s-a dorit ca modulul central Maven să ramână mic.

Un alt avantaj este existența tehnicii de rezolvare a dependențelor în funcție de scopul declarat (eng. dependency scoping) care permite să folosești jar-urile când ai nevoie, implicit scopul este la compilare (compile), dar mai există: provided, runtime (dependențele nu sunt neceare la compilare ci la executie, de exemplu JDBC), test (de exemplu JUNIT), system, import.

Printre beneficiile aduse dezvoltatorului se numără și faptul că acesta se obișnuieste foarte repede cu structura proiectului și îi va fi ușor să dezvolte unul nou, fără să piardă timp pentru a crea noi structuri, scripturi, fișiere de configurare, etc.

Așadar am ales să folosesc Maven pentru că se instalează ușor, mi-a oferit un arhetip potrivit pentru structura aplicației mele, fișierele de configurare pentru stabilirea dependențelor se realizează rapid și intuitiv fiind o mulțime de exemple pe internet, mi-a ușurat munca de găsire a JAR-urilor de care aveam nevoie și de compilare. Eu am folosit Apache Maven versinea 3.3.9 care a necesitat instalarea unui JDK cu versiunea minim 1.7.

#### Jboss Wildfly

Produsul realizat de mine fiind o aplicație web, am avut nevoie de un sevrer care să servească conținut web, așa că am ales WildFly. Acesta este un server de aplicații, inițial cunoscut sub numele de JBoss. În anul 2014 serverul de aplicații JBoss a fost redenumit Wildfly, însă celelalte produse și comunitatea JBoss și-au păstrat numele, făcându-se astfel diferența între cele două versiuni disponibile: community edition și enterprise edition, cea din urmă fiind îmbunătățită doar prin rezolvarea anumitor probleme (eng. bug-fixes), nu prin schimbarea nucleului central. Acum este dezvoltat de Red Hat.

Wildfly este gratis, open-source, dezvoltat în Java și implementat astfel încât să suporte specificațiile Java EE. Oferă un mediu potrivit pentru rularea aplicațiilor pe diverse platforme.

Oferă multiple facilități de lucru cu el. Se instalează ușor, pe site-ul oficial găsindu-se informații detaliate pentru configurare și se integrează în IDE și în aplicație fără să ridice probleme utilizatorilor. Este nevoie de descărcarea arhivei de Wildfly de pe site-ul oficial și dezarhivarea ei, apoi rularea a două fișere de configurare. Există 2 posibilități apoi pentru administrare, din linie de comandă sau din consola de administrare. În Eclipse, IDE-ul folosit de mine, cea mai potrivită metodă de a lucra cu acest server am considerat că este utilizarea ad-on-ului JBoss Tools instalat prin opțiunea Help-> Eclipse MarketPlace din IDE, apoi trebuie adăugat Wildfly ca și server. Totul este intuitiv, iar dacă există nelămuriri sau se întâmpină diverse probleme de instalare sau de configurare, comunitatea vastă care utilizează Wildfly sigur poate oferi răspunsuri pertinente.

Eu am folosit Wildfly versiunea 9. Acesta are un timp de pornire relativ bun, iar oprirea este aproape instantă. Am făcut alegerea potrivită și datorită ușurinței cu care pot să accesez baza de date prin mecanismele pe care le pune la dispoziție. La momentul actual există și versiuni mai noi care asigură îmbunătățiri relevante.

În structura directorului din arhiva cu Wildfly, așa cum sunt menționate în documentația oficială a serverului, regăsim cele mai importante componente:

* appclient (conține fișiere de configurare și de implementare conținut, dar și zonele modificabile de către containerul web al aplicației care rulează pe server)
* bin (scripturi și fisiere de pornire server și configurare, diverse implementări pentru facilități precum: adăugarea unui nou utilizator, rapoarte de diagnoza Java, etc)
* bin/client (jar util celor care nu folosesc Maven)
* docs/schema (fișiere de definiție a shemei XML)
* docs/examples/config (exemple de fișiere de configurare pentru cazuri de utilizare specifice)
* domain
* modules (diverse module din arhitectura modulară a serverului de Wildfly)
* standalone
* welcome-content (conținutul implicit al paginii de pornire)

Aplicația mea a presupus și o conexiune la baza de date așa că pentru realizarea configurării necesare pentru această conexiunea am creat propriul folder după modelul celui existent numit h2database din directorul: \modules\system\layers\base\com.

### Decizii de proiectare

Proiectarea bazei de date a fost un delung proces pentru ca trebuiau de la inceput definite entitatile si relatiile dintre ele. In cele din urma, forma finala a acesteia se poate observa in figura 2.

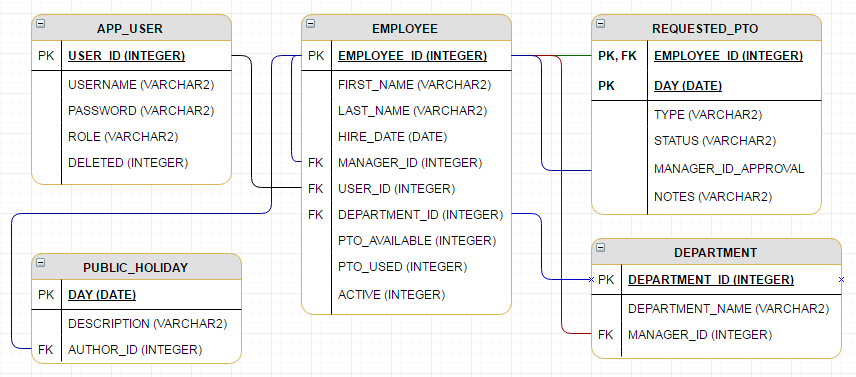


Figure Diagrama bazei de date

Intre cele 5 tabele din baza de date exista relatii atat unu-la-unu cat si unu-la-mai-multi. Singurele relatii de tip unu-la-unu sunt intre employee\_id si manager\_id din tabela employee si intre user\_id din tabela app\_user si tabela employee. Celelalte sunt de tip unu-la-mai-multi.

## Manualul de utilizare

### Acces public

In aplicatie accesul se face doar pe baza de nume de utilizator si parola. Asadar orice accesare a unui link din aplicatie fara sa fii autentificat te va redirecta automat catre pagina de conectare.

### Utilizator autentificat

Utilizatorul de tip angajat normal dispune de o interfata ca in figura urmatoare dupa autentificare. Este pagina web: Dashboard in care isi poate adauga zile de CO, iar in josul paginii este un tabel cu zilele de concediu pe anul respectiv, in functie de selectia utilizatorul: cereri viitoare, cereri de pana la data curenta si cereri in asteptare. In meniu mai sunt 2 elemente prin care angajatul isi va putea schimba parola sau isi poate schimba anumite informatii personale.

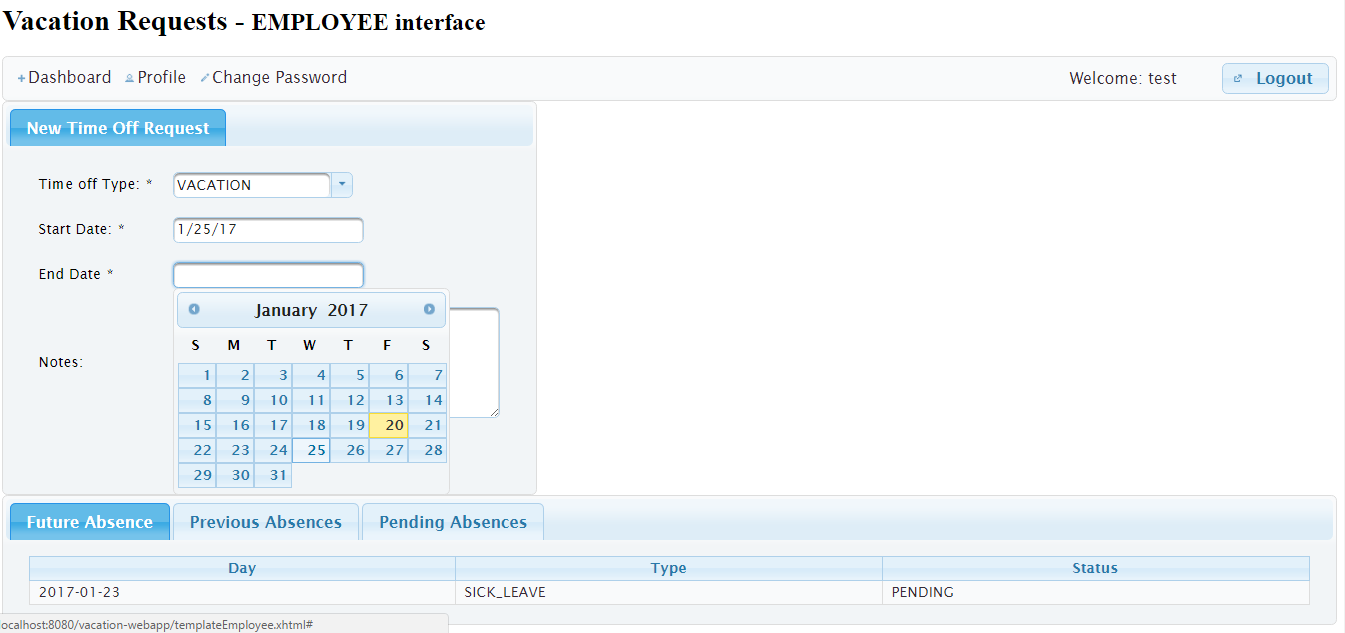


Figure Ecranul pentru angajatul normal

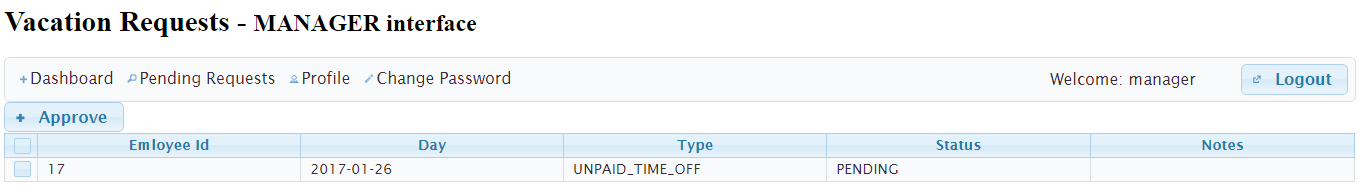


Figure Ecranul pentru aprobarea concediior de catre manager

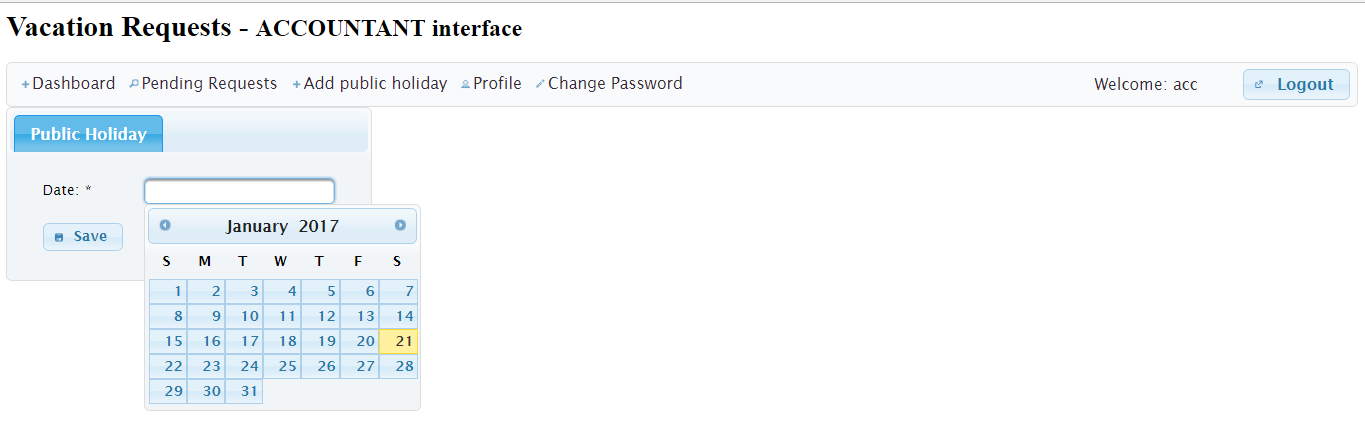


Figure Ecranul de adaugare a zilelor oficiale libere de catre contabil

Fiecare tip de utilizator are o interfata specifica, tipica rolurilor pe care le detine. Interfata este intuitiva, cu un meniu princpal in partea de sus din care poate alege operatiunea dorita. De asemenea numele de utilizator este afisat in partea dreapta sus langa butonul de iesire din aplicatie.

### Administrator

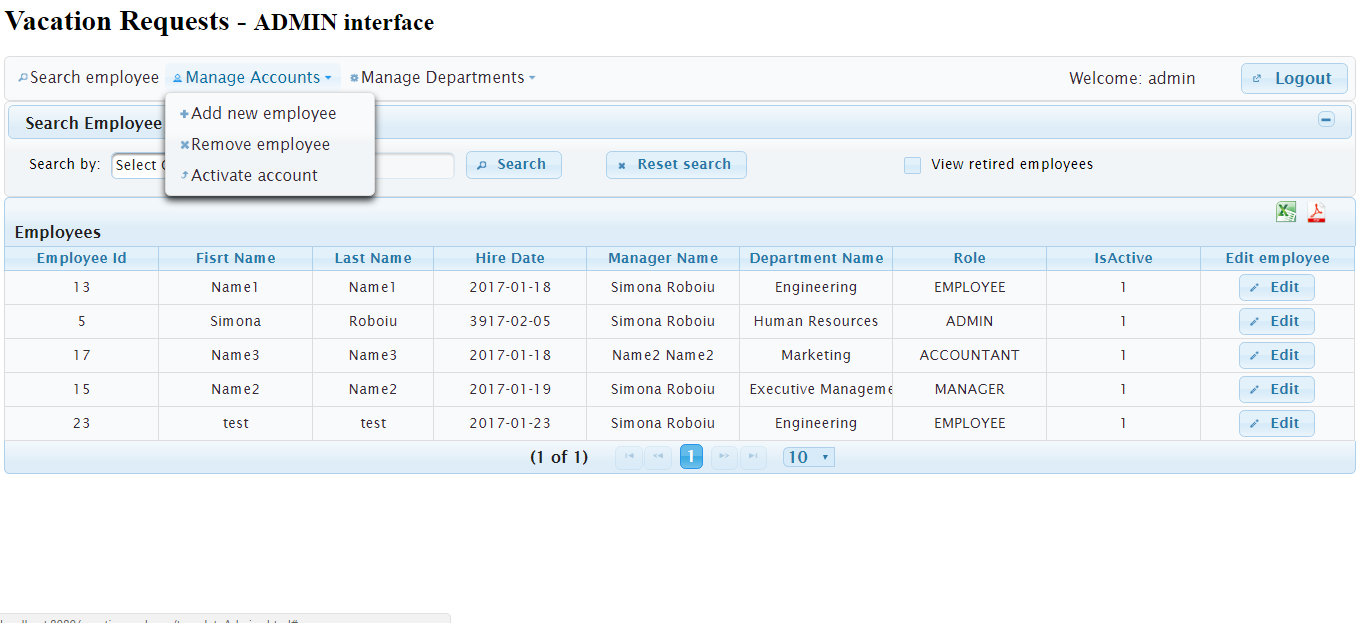


Figure Ecranul pentru administrator

## Instalarea aplicației

In folderul cu sursele aplicatei se afla un fisier Readme.md in care sunt explicatii pasii pentru instalare si configurare. De asemenea, se poate tine cont si de explicatiile din sectiunea Tehnologii folosite.

### Cerințe preliminare

Am folosit sistemul de operare Windows 7 pe 64 de biti cu urmatoarele proprietati: procesor intel core i5, 8GB memorie ram, frecventa 2.5GHz.

## Testarea aplicației

Aplicatia a fost testata manual. Nu am creat teste unitare, nici functionale.

# Concluzii și dezvoltări ulterioare

## Concluzii

Aplcatia este realizata in procent de 80% la momentul actual. Inca trebuie adaugate anumite functionalitati si validari. Desi pare o aplicatie simpla, in realitate sunt destul de multe constrangeri de care trebuie tinut cont in dezvoltarea ei.

## Dezvoltări ulterioare

Trebuie implementate functionalitatile de baza inca neimplementate. In plus se poate adauga functionalitatea noua ca un utilizatorul sa isi poata seta o poza de profil.

# Bibliografie

1. Antonio Goncalves, Beginning Java EE 7, 2013, paginile 314, 350
2. Site-ul ofiacial Apache Maven, online: https://maven.apache.org/
3. Mecanismul de rezolvare a dependențelor pentru Apache Maven (Scopul dependențelor), online: https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html#Dependency\_Scope
4. Tutorial Maven, 19.06.2016, online: http://www.vogella.com/tutorials/ApacheMaven/article.html
5. Wildfly, 2016, online: http://www.wildfly.org/
6. Structura directorului Wildfly , 2016, online: https://docs.jboss.org/author/display/WFLY9/Getting+Started+Guide#GettingStartedGuide-WildFly8DirectoryStructure
7. Tutorial - arhitectura JSP, online: http://www.tutorialspoint.com/jsp/jsp\_architecture.htm
8. Tutorial – arhitectura JSF, online: http://www.tutorialspoint.com/jsf/jsf\_architecture.htm
9. Primefaces downloads, online: http://primefaces.org/downloads
10. SecureFiles - The Next Generation Unstructured Data Management, online: http://www.oracle.com/technetwork/database/features/secure-files/securefiles-160920.html
11. JDBC Product Components, online: http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html
12. Interface EntityManager din API-ul de persistență, pachetul javax.persistence, online: http://docs.oracle.com/javaee/7/api/javax/persistence/EntityManager.html

# Anexe

[Figure 1 Flux in aplicatie 5](#_Toc472726657)

[Figure 2 Diagrama bazei de date 10](#_Toc472726658)

[Figure 3 Ecranul pentru angajatul normal 11](#_Toc472726659)

[Figure 4 Ecranul pentru aprobarea concediior de catre manager 11](#_Toc472726660)

[Figure 5 Ecranul de adaugare a zilelor oficiale libere de catre contabil 11](#_Toc472726661)

[Figure 6 Ecranul pentru administrator 12](#_Toc472726662)

[Table 1 Tehnologii folosite 5](#_Toc472726727)