XXXX

add title page

(from regulations)

XXXX

add annex 1 xor 2 (formular de inscriere a temei lucrarii)

(from regulations with signatures, scan)

XXXX

add Annex 6/6bis - copyright declaration

(from regulations with signature, scan)

XXXX

add Annex 5/5bis - academic honesty declaration.

(from regulations with signatures, scan)

Cuprins

[Introducere 19](#_Toc42244321)

[1 Introducere in metodologia agilă 21](#_Toc42244322)

[1.1 Metodologia de lucru agil 21](#_Toc42244323)

[1.2 Platforme agile 22](#_Toc42244324)

[1.2.1 Jira 22](#_Toc42244325)

[1.2.2 Bitrix24 22](#_Toc42244326)

[1.2.3 ClickUp 23](#_Toc42244327)

[2 Instrumente software folosite 24](#_Toc42244328)

[2.1 Baza de date 24](#_Toc42244329)

[2.1.1 MySQL 24](#_Toc42244330)

[2.2 Serverul 24](#_Toc42244331)

[2.2.1 Java 24](#_Toc42244332)

[2.2.2 SpringBoot 24](#_Toc42244333)

[2.2.3 Hibernate 25](#_Toc42244334)

[2.3 Interfața grafică 26](#_Toc42244335)

[2.3.1 JavaScript 26](#_Toc42244336)

[2.3.2 ReactJS 26](#_Toc42244337)

[3 Arhitectura aplicației 27](#_Toc42244338)

[4 Aplicația finală 29](#_Toc42244339)

[Concluzii 31](#_Toc42244340)

[Bibliografie 33](#_Toc42244341)

[Annex 1 35](#_Toc42244342)

[Annex 2 37](#_Toc42244343)

List of figures

[Figura 4.1: image template ro - XXXX schimba titlul 35](#_Toc12795531)

[Figure 4.1: image template eng XXXX change title 35](#_Toc12795532)

List of tables

Tabelul 4.1: table template ro XXXX schimba titlul 35

Table 4.1: table template eng XXXX change title 35

List of equations

[Ecuaţia 4.1: equation template ro XXXX schimba titlul 36](#_Toc12795535)

[Equation 4.1: equation template eng XXXX change title 36](#_Toc12795536)

List of acronyms and abbreviations

API – Application Programming Interface

CSS – Cascading Style Sheets

HTML – Hyper Text Markup Language

HQL – Hibernate Query Language

JDBC – Java Database Conectivity

JSX – JavaScript XML

REST –Representational state transfer

# Introducere

Lucrarea de față își propune să proiecteze o platformă web care să poată gestiona proiectele, sarcinile de lucru aferente, facilitarea unui bun control al tuturor necesităților tehnice și facilitarea unei comunicări eficiente între conducere, angajați și elevi.

Alegerea temei a fost influențată de importanța unei comunicări eficiente în cadrul unei echipe, în cadrul unui departament și în cadrul unei organizații în general. Pentru a obține o comunicare eficientă, în facultatea noastră, consider că este nevoie, în primul rând, de o modalitate ușoară și accesibilă tuturor, în special studenților, pentru a raporta problemele întâmpinate către responsabilii facultății. În al doilea rând este nevoie de un sistem de gestiune a tuturor departamentelor, proiectelor și sarcinilor de lucru care să permită o mai bună viziune de ansamblu, atât pentru cei care se ocupă cu mentenanță, cât și pentru conducerea facultății.

Urmând câteva idei și principii din metodologiile de lucru ale echipelor de dezvoltatori, am dorit ca aplicația să permită șefilor de departament sau conducerii, să asigneze câteva sarcini de lucru, pentru angajații care se ocupă de mentenanță sau de partea tehnică. Aceștia, la rândul lor, vor putea să înregistreze progresul făcut în cadrul unei sarcini, să introducă un număr de ore lucrate, să aproximeze numărul de ore necesare pentru îndeplinirea sarcinii. Toate aceste detalii vor fi vizibile pentru manager. Astfel, aplicația va oferi o viziune clară de ansamblu, asupra modului în care evoluează un anumit proiect sau sarcina de lucru, ceea ce va permite o mai buna prioritizare și eventuale întrajutorări intre membrii echipei.

Pe lângă posibilitatea de management și gestiune al proiectelor și sarcinilor de lucru, aplicația trebuie să permită studenților facultății să raporteze anumite probleme sau nereguli întâmpinate în timpul folosirii oricărei resurse puse la dispoziție de facultate. Imediat ce este raportată problema, un mail cu conținutul mesajului studentului ar trebui să fie trimis automat către șeful departamentului, acesta trebuind să verifice problema și să decidă mai departe ce este de facut.

Platforma finala va conține 2 sub-aplicații, cu funcționare distinctă, dar care vor interacționa în strânsă legătură. Pe deoparte va exista aplicația server, numita si backend, ce are rolul de a interacționa cu baza de date, prin operații de tip creare, update sau ștergere. Serverul va trebui să pună la dispoziție astfel câteva puncte finale, care atunci când sunt invocate, să producă modificările dorite asupra bazei de date.

Pe de altă parte va exista interfața cu utilizatorul care sa permită modificarea bazei de date si a informațiilor din aceasta într-un mod facil si prietenos pentru utilizatorul de rând. Interfața cu utilizatorul, prin acțiunile efectuate de către utilizator va invoca acele puncte finale ale serverului, pentru crearea, modificarea sau ștergerea unei informații din baza de date.

În ceea ce privește tehnologiile folosite, aplicația va folosi MySQL ca sistem de gestiune al bazei de date, un server care să gestioneze interacțiunea dintre baza de date și interfața cu utilizatorul construit folosind Java ca limbaj de programare, împreună cu instrumentele specializate de dezvoltare Spring Boot și Hibernate. Interfața cu utilizatorul va fi elaborată folosind ReactJS, o librărie specială de Java Script care permite crearea de GUI-uri dinamice întru-un mod facil. Pe lângă Java Script, se vor folosi JSX, un limbaj descriptiv foarte asemănător cu HTML, dar și CSS pentru stilizare si aranjare in pagină.

# Introducere in metodologia agilă

## Metodologia de lucru agil

Metodologia de lucru agil este un model de gestionare si organizare a unui proiect de tip software, care susține o dezvoltare și reglementare din aproape in aproape. Practic o aplicație întreaga este spartă în probleme mai mici, ce trebuie tratate pe rând in vederea livrării celui mai bun produs posibil.

În dezvoltarea aplicațiilor software, se întâmplă frecvent să apară întârzieri datorită problemelor neprevăzute ce pot apărea sau datorită modificării dorințelor clientului. Acestea din urma apar deoarece este dificil la început să se descrie cu exactitate toate cerințele unei aplicații. De aceea sunt necesare întâlniri recurente intre dezvoltatorii aplicației si client pentru a discuta eventuale probleme întâmpinate si direcțiile de urmat pe viitor.

Metodologia agilă de lucru promovează împărțirea unei probleme în subprobleme care să fie apoi planificate pe durate scurte, pentru a putea trata cât mai bine fiecare impediment întâmpinat. Planificând un proiect doar pe termen scurt permite o flexibilitate foarte mare în ceea ce privește dezvoltarea aplicației. Aceste planificări pe termen scurt se împart in cicluri de dezvoltare numite sprinturi sau iterații. De regula la începutul fiecărui sprint, există un număr de ședințe fixate, în vederea planificării sarcinilor de lucru. Acestea se negociază intre dezvoltatori si client in funcție de dificultatea sarcinilor de lucru, timpul alocat, dar și în funcție de bugetul disponibil.

|  |
| --- |
| XXXX pune poza aici |
| Figura 1.Metodologia de lucru agil |

De regulă când vine vorba despre etapele de lucru ale metodologiei agile, putem împărții tot procesul in 6 pași:

* Discutarea cerințelor clientului cu dezvoltatorii și stabilirea specificațiilor aplicației de comun acord.
* Planificarea bugetului si perioadei de dezvoltare din punct de vedere al cantității de lucru dar și a timpului.
* Crearea arhitecturii, logicii interne și realizarea unor scheme de urmărit pentru dezvoltarea interfețelor de lucru cu utilizatorii, la nivel conceptual.
* Dezvoltarea propriu zisă, urmând planul de lucru stabilit.
* Testarea aplicației sau a modificărilor aduse dea lungul sprintului și darea in folosință a noii versiuni.
* Ședințe cu scopul de a analiza ce a fost bine, ce a fost mai puțin bine, ce se poate schimba și cum. Toate acestea se bazează pe monitorizarea din timpul sprintului, și au în vedere îmbunătățirea modului in care lucrează echipa si eventuale ajustări pentru viitor.

## Platforme agile

### Jira

Jira este probabil cel mai utilizat instrument software de monitorizare a progresului unui proiect, folosit de către echipele de dezvoltatori de aplicații care lucrează în sistem agil. Dintre facilitățile oferite de către Jira reamintim:

* Planificarea sprintului, incluzând aici descrierea sarcinilor de lucru precum și împărțirea lor între membrii echipei.
* Urmărirea progresului fiecărei sarcini de lucru, în vederea prioritizării lor sau pentru întrajutorarea între membrii echipei.
* Urmărirea progresului pentru întregul proiect în vederea livrării unui produs finit cât mai apropiat de cerințele clientului.
* Crearea de rapoarte în timp real sau la final de sprint pentru îmbunătățirea performanței echipei și pentru ajustarea sarcinilor de lucru pe viitor.
* Integrarea cu alte instrumente de dezvoltare deja bine cunoscute și folosite la scară larga în întreaga lume, cum ar fi BitBucket sau Confluence.

În ceea ce privește prețurile Jira este gratis pentru o echipa de până la 10 persoane, având la dispoziție doar anumite facilități. În varianta gratis Jira oferă doar 2 GB de stocare online și nu pune la dispoziție suport direct, permisiuni avansate sau acces anonim, iar integrarea este permisă doar pentru anumite instrumente, contra cost. Varianta plătită oferă suport 24/7, spațiu de stocare nelimitat online, acces anonim, securitate sporita, informații speciale pentru administrator, permisiuni avansate si multe altele.

### Bitrix24

Bitrix24 este o soluție de management al proiectelor, sarcinilor si sub-sarcinilor de lucru. Oferă suport pentru un număr nelimitat de proiecte. Oferă facilități asemănătoare cu cele mentionate în cadrul punctului 1.1, dintre care reamintim: urmărirea progresului pentru sarcinile de lucru si pentru proiecte, managementul documentelor aferente unui proiect, instrumente de colaborare intre membrii echipei, precum mesagerie instant videoconferință. Este posibilă stocarea informațiilor atât online cat si pe propriul server.

Bitrix24 este o soluție gratis pentru echipe de maxim 12 persoane. În cazul în care echipele sunt mai mari, este disponibil un abonament lunar de 99$ pentru un număr nelimitat de persoane.

În plus Bitrix24 este o soluție open source code și open API.

### ClickUp

ClickUp este un alt instrument de management și gestionare pentru proiecte și sarcini de lucru, care în varianta gratis oferă un număr nelimitat de utilizatori, număr nelimitat de proiecte și sarcini de lucru, suport 24/7. Pe lângă aceste facilități, variantele plătite oferă spațiu de stocare nelimitat, pentru fișierele echipei, posibilitatea adăugării invitațiilor care să aibă permisiuni temporare și restrânse, setarea permisiunilor avansate, dar și posibilitatea alegerii și schimbării paletei de culori, precum și a modului de aranjare in pagina.

# Instrumente software folosite

## Baza de date

### MySQL

MySQL este un sistem de gestiune a bazelor de date relaționale, produs de compania suedeza MySQL AB și distribuit sub Licența Publică Generală GNU. Este cel mai popular SGBD open-source la ora actuală, fiind o componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP).[1]

Deși este folosit foarte des împreună cu limbajul de programare PHP, cu MySQL se pot construi aplicații în orice limbaj major. Există multe scheme API disponibile pentru MySQL ce permit scrierea aplicațiilor în numeroase limbaje de programare pentru accesarea bazelor de date MySQL, cum ar fi: C, C++, C#, Java, Perl, PHP, Python, FreeBasic, etc., fiecare dintre acestea folosind un tip specific API.[1]

În ceea ce privește aplicația noastră, folosind Java ca limbaj de programare pentru partea de server, vom avea nevoie de API-ul JDBC care oferă o interfață standard pentru conectarea și interacțiunea cu orice bază de date de tip relațional.

## Serverul

Dintre instrumentele software folosite in dezvoltarea serverului amintim:

* Java – Limbaj de programare
* SpringBoot – framework de Java folosit in configurarea întregii aplicații de server
* JPA – Java Persistence API, folosit pentru interacțiunea cu baza de date

### Java

Java este pe deoparte un limbaj de programare de nivel înalt, orientat pe obiecte, similar cu C++, dar care oferă facilități mai avansate și mai usor de folosit. În plus Java este independent de platformă, orice aplicație sau cod Java compilat poate rula pe toate platformele, cu ajutorul JVM, având astfel avantajul portabilității.

Este un limbaj de programare foarte sigur deoarece tot codul Java este ulterior transformat in bytecode, acesta din urma fiind imposibil de înțeles de către un om.

### SpringBoot

SpringBoot este un framework Java, open-source, care facilitează crearea de micro-servicii pentru dezvoltatori. Micro serviciile sunt o arhitectura care ajuta la crearea și livrarea de servicii care lucrează independent. Practic fiecare serviciu rulează pe un proces separat și de sine stătător.

SpringBoot aduce cu sine multe avantaje dintre care reamintim:

* Configurații minime prin simpla adăugare a adnotațiilor
* Oferă un server deja integrat, ce nu mai necesita prea multe configurări
* Creează un mediu de dezvoltare care în timp este ușor de scalat
* Gestionare ușoară a dependințelor
* Procesare rapida si management facil pentru endpoint-urile de tip REST

### Hibernate

Hibernate este un instrument software folosit in maparea dintre obiecte si entitățile asociate dintr-o bază de date. Practic prin Hibernate se pot trata toate entitățile din baza de date, folosind clase, iar interacțiunea cu sau dintre acestea folosind metode.

Conectorul Java (JDBC) oferă o interfață standard pentru conectarea și interacțiunea cu orice bază de date de tip relațional, permițând scrierea de intergoari direct în codul de Java. Totuși pentru simplificare și mentenanță acest gen de abordare nu este deseori preferat. Soluțiile au fost Hibernate Query Language (HQL), care este un limbaj de scriere de interogări asemănător cu SQL, dar si Java persistence API (JPA), acesta din urma fiind o implementare a Hibernate, care pune la dispoziția dezvoltatorilor de aplicații câteva metode deja implementate, facilitând interacțiunea cu baza de date.

## Interfața grafică

### JavaScript

### ReactJS

# Arhitectura aplicației

# Aplicația finală

# Concluzii

# Bibliografie

Made by Mihai A

XXXX add here

1. <https://ro.wikipedia.org/wiki/MySQL>
2. ccc
3. ddd

# Annex 1

optional annexes as needed.

# Annex 2

optional annexes as needed