**Raport tehnic Proiect-SPT**

**Studenți : Enache Mihai**

**Ciocoiu Radu**

**Teodorescu Adrian**

**Hozu Marius Gabriel**

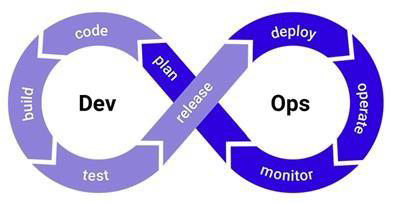
**Vlăsceanu Ovidiu**

**Manole Alexandru**

**Profesor coordonator :**

**Horia Alexandru Modran**

* Pentru modul de dezvoltare software am ales DevOps



Câteva avantaje ale metodologiei DevOps:

1. **Creșterea vitezei:** cu un model DevOps flexibil, tehnologia este optimizată în funcție de necesitățile curente ale ciclului de viață. În multe cazuri, DevOps utilizează cele mai recente tehnologii machine learning și inteligenţa artificială pentru a obține viteză. De fapt, un termen recent inventat, AIOps, se referă la utilizarea AI în operațiunile IT. DevOps pune accentul și pe automatizare și integrarea/livrarea continuă, scutind personalul de o serie de activități manuale, pentru a se concentra asupra inovării. În ceea ce privește dezvoltarea, inginerii pot să își atingă obiectivele privind codarea mai rapid sau să colaboreze mai eficient. Pe partea de operațiuni, administratorii de sistem pot utiliza frameworkuri de automatizare pentru a aloca și actualiza cu ușurință noi aplicații și infrastructuri.
2. **Calitate:** datorită creșterii vitezei, DevOps deschide noi posibilități de creștere a calității și fiabilității. Acest lucru începe, în ceea ce privește dezvoltarea, cu o colaborare mai rapidă și instrumente mai bune pentru rezolvarea și integrarea problemelor. În ceea ce privește operațiunile, actualizările mai mici și mai frecvente permit o mai mare stabilitate, ceea ce îmbunătățește calitatea generală a experienței utilizatorilor.
3. **Securitate:** utilizarea DevOps oferă mai multe niveluri pentru îmbunătățirea securității generale. Din motive practice, DevOps include uneori integrarea echipelor de securitate. Astfel, se creează un model denumit DevSecOps, care se ocupă și de securitate, în afară de dezvoltare și operațiuni. Pentru a detalia mai mult, viteza DevSecOps permite corecții, audituri și analize rapide, cu ajutorul inteligenței artificiale, conformității automate și gestionării automate a resurselor.
4. **Scalabilitate:** Resursele flexibile, automatizarea și capacitatea de a susține ciclul complet de dezvoltare a software-ului înseamnă că infrastructura este pregătită pentru a susține scalarea după necesități. Deoarece scalarea atinge numeroase elemente, susținerea unei viziuni holistice asupra tehnologiei ajută la scalarea eficientă, gestionând totul, de la gestionarea resurselor până la implementarea corecțiilor.

* Rolul fiecărui membru al echipei:

* + - * **Team Leader**: Enache Mihai-Sebastian
        + Descriere:

Acționează ca un fel de mentor și ajută echipă să se concentreze pe task-uri, să îndeplinească cerințele proiectului și să se asigure că acesta este realizat la timp

* + - * + Responsabilități:

Ghidarea echipei în a realiza cu succes proiectul

Prevenirea și rezolvarea de conflicte ce pot apărea pe durata realizării proiectului

* + - * **Project Manager** : Hozu Marius Gabriel
        + Descriere:

Trebuie să știe “cine, ce, unde, când și de ce” în planificarea proiectului

* + - * + Responsabilități:

Planifică, organizează, se ocupă de buget și execută livrarea produsului final

Supraveghează echipa

* + - * **Developer** : Vlăsceanu Ovidiu
        + Descriere:

Acesta se ocupă în mare parte cu scrierea de cod și dezvoltarea produselor software

* + - * + Responsabilități:

Dezvoltarea de feature-uri

Înștiințează Project Manager-ul de starea proiectul

* + - * **Software Architect** : Teodorescu Adrian-Iulian
        + Descriere:

Definește arhitectura per-total a sistemului (a proiectului în sine) și realizează alegeri în legatură cu design-ul acestui sistem

* + - * + Responsabilități:

Definirea arhitecturilor tehnică și funcțională a sistemului

Dezvoltarea majorității componentelor ce au importanță ridicată

* + - * **Ux/Ui designers** : Ciocoiu Radu-Mihail
        + Descriere:

Sprijină echipa în realizarea produsului final și se asigură că se utilizează practicile necesare ce duc la realizarea proiectului

* + - * + Responsabilități:

Analizează funcționalitățile fundamentale necesare utilizatorilor

Definește modelul de navigare și arhitectura informațională

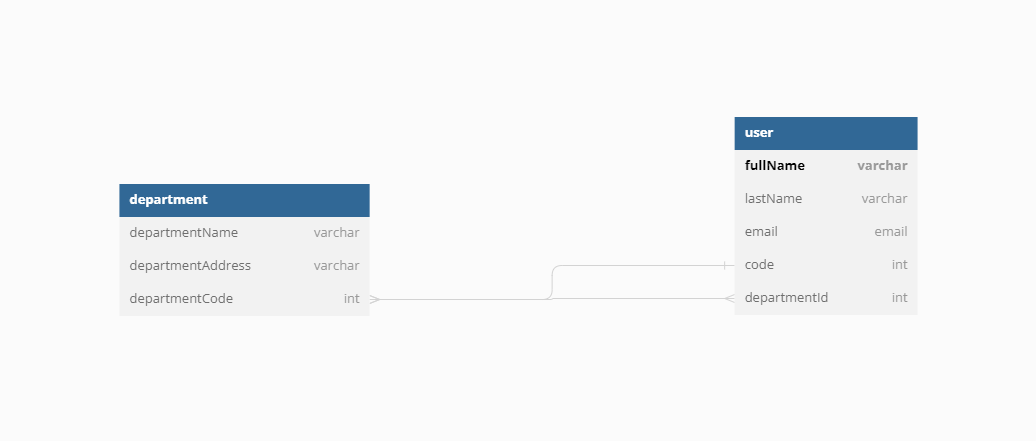
* + - * **Tester** : Manole Alexandru-Emil
        + Descriere:

Tester-ul asigură faptul că soluțiile software ating cerințele impuse și că acestea sunt conform standardelor de calitate

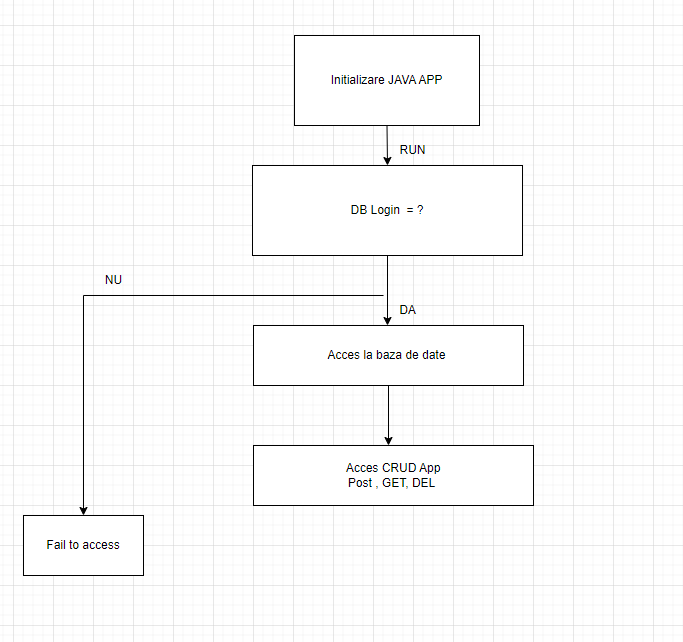
* + - * + Responsabilități:

Înțelegerea feature-urilor ce trebuiesc impuse proiectului

Crearea și executarea de teste de tip test case, unit test, etc. pentru a detecta bug-uri sau defecte

  
**Structura tabelelor si conexiuniile dintre ele**

**Diagrama de clasa**

****