**TITLUL LUCRĂRII DE LICENŢĂ**

LUCRARE DE LICENŢĂ

|  |  |  |
| --- | --- | --- |
|  | Absolvent: | **Popirtan Vlad-Mihai** |
|  |  |  |
|  | Coordonator științific: | **Prof. Dr. Ing. Madalin Neagu** |

**2022**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
| DECAN, |  | | DIRECTOR DEPARTAMENT, | |
| **Prof. dr. ing. Liviu MICLEA** |  | **Prof. dr. ing. Rodica POTOLEA** | |

Absolvent: **Popirtan Vlad-Mihai**

**TITLUL LUCRĂRII DE LICENŢĂ**

1. **Enunțul temei:** *Scurtă descriere a temei lucrării de licență și datele inițiale*
2. **Conținutul lucrării:** *(enumerarea pârților componente) Exemplu: Pagina de prezentare, aprecierile coordonatorului de lucrare, titlul capitolului 1, titlul capitolului 2,… titlul capitolului n, bibliografie, anexe.*
3. **Locul documentării**: *Exemplu*: Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
4. **Consultanți**:
5. **Data emiterii temei:** 1 noiembrie 2021
6. **Data predării:** 8 iulie 2022

|  |  |
| --- | --- |
| Absolvent: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |
| Coordonator științific: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Declarație pe propria răspundere privind**

**autenticitatea lucrării de licență**

Subsemnatul(a)**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**, legitimat(ă) cu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ seria \_\_\_\_\_\_\_ nr. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
CNP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, autorul lucrării \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_elaborată în vederea susținerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ din cadrul Universității Tehnice din Cluj-Napoca, sesiunea \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ a anului universitar \_\_\_\_\_\_\_\_\_\_, declar pe propria răspundere că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării, și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

In cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

|  |  |  |
| --- | --- | --- |
| Data  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Nume, Prenume  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
|  |  | Semnătura |

**Instrucțiuni generale**

**De citit înainte** (paginile de aici până la cuprins se vor elimina din versiunea finală):

1. Cele trei pagini anterioare (foaie de capăt, foaie sumar, declarație) se vor lista pe foi separate (nu fată-verso), fiind incluse în lucrarea listată.

Foaia de sumar (a doua) necesită semnătura absolventului, respectiv a coordonatorului.

Pe declarație se trece data când se predă lucrarea la secretarii de comisie.

1. Pe foaia de capăt, se va trece corect titulatura cadrului didactic îndrumător (consultați pagina de unde ați descărcat acest document pentru lista cadrelor didactice cu titulaturile lor).
2. **Cuprinsul** începe pe pagina nouă, impară (dacă se face listare fată-verso).

Toate capitolele încep pe o pagină nouă. Numerotarea paginilor începe de la 1 la primul capitol (**Introducere)**. Pentru actualizarea cuprinsului, folosiți meniul *References: Table of Contents->Update table*.

1. Vizualizați (și în timpul editării, dacă e cazul) acest document după ce activați vizualizarea simbolurilor ascunse de formatare (click pe pictograma **** din *Home/Paragraph*).
2. Fiecare capitol începe pe pagină nouă. Acest lucru este realizat folosind simbolul de formatare *Section Break* cu opțiunea *Next Page*, care a fost folosit la sfârșitul fiecărui capitol. Dacă ștergeți din greșeală simbolul, îl puteți pune înapoi folosind submeniul *Breaks* al meniului *Layout*.
3. Folosiți stilurile predefinite (Headings, Figura, Tabel, Normal, etc.)
4. Mărimea hârtiei este A4, iar marginile paginilor sunt 25.4 mm toate.
5. Respectați restul instrucțiunilor din fiecare capitol.
6. Înainte de salvarea ca fisier pdf, alegeți din meniul *Options* opțiunile din figura următoare



Cuprins

[Capitolul 1. Introducere – Contextul proiectului 1](#_Toc66962577)

[1.1. Contextul proiectului 1](#_Toc66962578)

[1.1.1. Exemplu de titlu subsecțiune 1](#_Toc66962579)

[Capitolul 2. Obiectivele Proiectului 3](#_Toc66962580)

[Capitolul 3. Studiu Bibliografic 5](#_Toc66962581)

[Capitolul 4. Analiză și Fundamentare Teoretică 7](#_Toc66962582)

[Capitolul 5. Proiectare de Detaliu si Implementare 9](#_Toc66962583)

[Capitolul 6. Testare și Validare 11](#_Toc66962584)

[Capitolul 7. Manual de Instalare si Utilizare 13](#_Toc66962585)

[Capitolul 8. Concluzii 15](#_Toc66962586)

[Bibliografie 17](#_Toc66962587)

[Anexa 1 19](#_Toc66962588)

# Introducere – Contextul proiectului

Tehnologia este un aspect ce evoluează din ce in ce mai rapid in zilele noastre, lucru ce influențează in mod direct viața de zi a oamenilor. Deși exista si aspecte negative ale acestei evoluții, trebuie sa amintim ca tehnologia aduce in principal beneficii asupra noastră. Multe lucruri ce in trecut păreau imposibile, precum comunicarea zilnica cu prietenii sau familia de la distanta, acum se pot realiza cu ajutorul câtorva apăsări de butoane prin intermediul apelurilor telefonice, sau prin folosirea rețelelor de socializare.

Un alt rezultat al acestei evoluții este automatizarea industriala. Prin înlocuirea oamenilor cu roboti industriali, se elimina, sau cel puțin diminuează, probleme precum riscul erorilor umane de calcul, riscul accidentarilor in timpul operațiilor, inconsistenta adusa de stările diferite in care se poate afla o persoana umana, sau nevoia de buget crescut alocat salariilor angajaților. Cantitatea produsa de industrii a devenit si ea mai mare odată cu automatizarea proceselor, robotii industriali neavând nevoie de pauze, aceștia fiind funcționali pe tot parcursul zilei.

## Contextul proiectului

Deși am precizat ca folosirea roboților industriali si automatizarea sarcinilor reduce nevoie de personal uman, întrebarea care a apărut este „Cine creează această automatizare?”. Astfel a apărut domeniul IT. Domeniul IT a devenit unul foarte căutat de studenți din mai multe motive:

* copii sunt atrași de calculator încă de la o vârstă frageda, datorita jocurilor video;
* este un domeniu dificil, perfect pentru persoanele dornice de noi provocări, fie acestea create de necunoscutul tehnologiei, sau de dorința de performanta;
* Este recunoscut ca unul din cele mai bune plătite domenii in care poți activa;

Așa ca orice persoana poate fi atrasa de ideea de a deveni programator. Însă principala calitate pe care trebuie sa o dețină un programator este gândirea algoritmica. Deși nu este un lucru dificil sa înveți un limbaj de programare, sau sa urmezi niște pași care sa te duca la îndeplinirea sarcinii, acest lucru nu te va duce prea departe in cariera ta, atat pe plan financiar, cat si in propria satisfacție. Vor veni momente in care pentru a te face remarcat trebuie sa fii in stare sa gândești un sistem de la zero, sau sa aduci o soluție ce va beneficia produsul, ridicându-l deasupra competiției.

Intr-un studiu realizat folosind Stack Overflow Jobs [1], reiese ca pe lângă cunoștințele in domeniul programării (ex: Limbajul de programare, librarii, unelte de development), companiile sunt interesate de așa zisele soft skills, caracteristicile unui individ legate de comportamentul acestuia (ex: Comunicare, lucrul in echipa, abilitățile de lider). In urma analizării descrierii a douăzeci de mii de postări de recrutare, pe lângă abilitățile de comunicare si abilitați legate de colaborare, cuvintele „analytical”, „problem solving” sau „deliver” sunt prezente in una din zece postări, cuvinte ce denota abilitatea de rezolvare a problemelor menționată mai sus.

O alta calitate pe care trebuie sa o aibă un programator este abilitatea de a lucra rapid. Cum nu fiecare proces este menit succesului din prima încercare, un programator bun trebuie sa fie in stare sa încerce mai multe soluții pentru problema in cauza, lucru care ar lua foarte mult timp in cazul unui programator lent.

Datorita celor menționate mai sus, putem trage concluzia ca orice persoana poate deveni un programator bun, atât timp cat este dispus sa sacrifice timpul necesar pregătirii acestor cerințe. Însă calitățile prezentate se pot dobândi doar prin exercițiu. Abilitatea de a rezolva probleme se poate antrena prin abordarea a diferitor probleme, in timp ce abilitatea de a scrie cod rapid si eficient este doar ajutata de practicarea regulata a programării.

Pentru a expune copiii la domeniul IT, scoli si licee au introdus materia de de calcul in programa școlară, inca din primii ani ai elevului. Anglia a introdus materii de calcul chiar din grădiniță, in Australia copii sunt expuși la tehnologii digitale din primul an școlar, in timp ce in Finlanda, materii de calcul sunt introduse încă din primul an de școală [2].

## Problema

O mare parte din întrebările de interviu pentru companiile software, este reprezentata de probleme de algoritmica, pentru a observa gândirea candidatului. Deși școlile fac eforturi sa introducă materii de informatica in programa școlară, timpul limitat care este alocat acestor materii, nu poate pregăti un elev pentru așteptările companiilor atunci când recrutează.

După cum am menționat mai sus, prin pachetul salarial atractiv pe care îl oferă acest domeniu, programarea poate stârni interesul unor oameni ce nu au terminat o facultate de specialitate. Sunt oameni ce nu doresc sa investească anii necesari pentru a termina o facultate de specialitate, iar lipsa de resurse educaționale ii poate descuraja din a afla o posibila pasiune din care sa câștige si un trai. La fel ca orice domeniu, exista manuale si cursuri online, însă acestea nu conțin o diversitate ridicata, se axează in principiu pe conceptele de baza ale programării si nu pun accentul pe optimizarea soluțiilor.

## Abordarea problemei

Titlul proiectului reprezintă o platforma online, gratuita si valabila oricui dorește sa își testeze abilitățile de programare. Fiecare soluție va fi rulata pe un set de teste, specifice fiecărei probleme, ce asigura corectitudinea completa a soluției, investigând situații specifice, la care utilizatorul va fi nevoit sa gândească suplimentar pentru a le găsi si trata.

Cum fiecare persoana este familiara cu internetul si aplicațiile web, titlul proiectului dorește să facă utilizarea sa plăcută pentru orice utilizator, acesta nefiind nevoit sa instaleze diferite compilatoare sau aplicații suplimentare pentru a isi testa soluția la problema. Utilizatorul va beneficia de un editor de text, integrat in pagina aplicației, care sa se ocupe de funcționalitatea de „syntax highlighting”.

Fiecare soluție va fi verificata din punct de vedere al corectitudinii, atât in cadrul rezolvării problemei propuse, dar si din punct de vedere sintactic. Aceasta va fi verificata si din punct de vedere al timpului de rulare si al memoriei utilizate, fiecare problema având aceste constrângeri predefinite.

Orice utilizator poate rezolva problema aleasa, in limbajul sa de programare preferat, atata timp cat acesta se afla in optiunile valabile: Java, Python, C/C++.

# Obiectivele Proiectului

În acest capitol se prezintă tema propriu zisă (sub forma unei teme de proiectare/cercetare, formulată exact, cu obiective clare – 2-3 pagini – și eventuale figuri explicative).

Reprezintă cca. 10% din lucrare.

# Studiu Bibliografic

Documentarea bibliografică are ca obiectiv prezentarea stadiului actual al domeniului/sub-domeniului în care se situează tema. În redactarea acestui capitol (în general a întregului document) se va ține cont de cunoștințele acumulate la disciplinele dedicate din semestrul 2, anul 4 (Metodologia Întocmirii Proiectelor, etc.), precum si la celelalte discipline relevante temei abordate.

Acest capitol reprezintă cca. 15% din lucrare.

Referințele se scriu în secțiunea *Bibliografie*. Formatul referințelor trebuie sa fie de tipul *IEEE* sau asemănător. Introducerea și formatarea referințelor în bibliografie, respectiv citarea în text, se poate face manual – nu recomandăm, întrucât sunt mai greu de gestionat corect – sau folosind instrumentele de lucru menționate în ultimele paragrafe din acest capitol.

In secțiunea *Bibliografie* sunt exemple de referințe pentru articol la conferințe [1] [2], articol în jurnal [3], sau cărți [4]. Referințele spre aplicații sau resurse online (pagini de internet) trebuie sa includă cel puțin o denumire sugestivă pe lângă URL ca în [5], plus alte informații dacă sunt disponibile (autori, an, etc.) și data ultimului acces. Referințele care prezintă doar link spre resursa online se vor plasa în footer-ul paginii unde sunt referite.

Citarea referințelor în text este obligatorie, vezi exemplul de mai jos (în funcție de tema proiectului se poate varia modul de prezentare a metodei/aplicației).

În articolul [1] autorii prezintă un sistem pentru detecția obstacolelor în mișcare folosind stereo viziune și estimarea mișcării proprii (nu e adevărat despre articolul 1). Metoda se bazează pe …*trecerea în revistă a algoritmilor, structurilor de date, funcționalitate, aspecte specifice temei proiectului* *etc*….. Discuție: *avantaje – dezavantaje*.

În capitolul 4 al [4] *se prezintă* …..

Începând cu **MS Word 2007** se poate folosi sistemul integrat de gestiune bibliografiei: *References* submeniul *Citations & Bibliography.* Mai multe informații se găsesc în documentația online de la MS Office.

# Analiză și Fundamentare Teoretică

Împreună cu capitolul următor trebuie să reprezinte aproximativ 60% din lucrare.

Scopul acestui capitol este să explice principiile funcționale ale aplicației implementate. Aici descrieți soluția propusă din punct de vedere teoretic - explicați și demonstrați proprietățile și valoarea teoretică:

* algoritmi utilizați și/sau propuși,
* protocoale utilizate,
* modele abstracte,
* explicații/argumentări logice ale soluției alese,
* structura logică și funcțională a aplicației.

NU SE FAC referiri la implementarea propriu-zisă.

NU SE PUN descrieri de tehnologii preluate cu copy-paste din alte surse sau lucruri care nu țin strict de proiectul propriu-zis (materiale de umplutură).

# Proiectare de Detaliu si Implementare

Împreună cu capitolul precedent reprezintă aproximativ 60% din total.

Scopul acestui capitol este să documenteze aplicația dezvoltată în așa fel încât dezvoltarea și întreținerea ulterioară să fie posibilă. Cititorul trebuie să poată identifica funcțiile principale ale aplicației din ceea ce este scris aici.

Capitolul ar trebui sa conțină (nu se rezumă neapărat la):

* schema generală aplicației,
* descriere a fiecărei componente implementate, la nivel de modul,
* diagrame de clase, clase importante și metode ale claselor importante.

# Testare și Validare

Aplicatia fiind una impartita pe module, am putut testa separat componentele. Pentru inceput, am folosit unealta Postman pentru a verifica serviciul de backend, testand raspunsul aplicatiei la evenimente ce ar putea aparea in timpul rularii

# Manual de Instalare si Utilizare

În secțiunea de Instalare trebuie să detaliați resursele software și hardware necesare pentru instalarea și rularea aplicației, precum și să descrieți pas cu pas procesul de instalare. Instalarea aplicației trebuie să se poată face folosind instrucțiunile date aici.

Utilizarea aplicației trebuie descrisă din punctul de vedere al utilizatorului, fără a menționa aspecte tehnice interne. Folosiți capturi ale ecranului și explicați pas cu pas interacțiunea cu persoana care execută instalarea. Folosind acest manual, o persoană ar trebui să poată instala și utiliza produsul vostru.

# Concluzii

Acest capitol va ocupa circa 5% din lucrare.

Capitolul ar trebui sa conțină (nu se rezumă neapărat la):

* un rezumat al contribuțiilor voastre
* o analiză critică a rezultatelor obținute
* o descriere a posibilelor dezvoltări și îmbunătățiri ulterioare

# Bibliography

|  |  |
| --- | --- |
| [1] | Montandon, João & Politowski, Cristiano & Silva, Luciana & Valente, Marco & Petrillo, Fabio & Guéhéneuc, Yann-Gaël. (2020). What Skills do IT Companies look for in New Developers? A Study with Stack Overflow Jobs. Information and Software Technology. 10.1016/j.infsof.2020.106429. |
| [2] | Szabo, Claudia & Sheard, Judy & Luxton-Reilly, Andrew & Simon, Beth & Becker, Brett & Ott, L.M.. (2019). Fifteen Years of Introductory Programming in Schools: A Global Overview of K-12 Initiatives. 1-9. 10.1145/3364510.3364513. |
| [3] | G. Cachon și M. Lariviere, „Supply chain coordination with revenue sharing contracts: strengths and limitations,” *Management Science,* vol. 51, pp. 30-44, 2005. |
| [4] | C. P. Pfleeger, S. L. Pfleeger și J. Margulies, Security in Computing, 5th Edition, Pearson, 2015. |
| [5] | Software Freedom Conservancy, „The Selenium Browser Automation Project,” [Interactiv]. Available: https://www.selenium.dev/. [Accesat 17 martie 2021]. |

După ce ați inserat/actualizat bibliografia selectați întregul tabel și aplicați stilul *Biblio.*  Stilul *Normal* are indentare la începutul paragrafelor și, de aceea nu veți obține formatul ca mai sus fără acest pas.

# Anexa 1

…

Secțiuni relevante din cod

…

Alte informații relevante (demonstrații etc.)

…

Lucrări publicate (dacă există)

etc.