**UNIVERSITATEA DIN BUCUREȘTI**

**FACULTATEA**

**DE**

**MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA**

**BAZE DE DATE SI TEHNOLOGII SOFTWARE**

**Lucrare de disertatie**

**FlyBird – Sistem informatic si de gestiune al zborurilor**

**Coordonator științific Absolvent**

**Banu Iulia Teodora Vătafu Ștefan**

**București, Iulie 2023**

Cuprins

Contents

[Cuprins 2](#_Toc130332032)

[Abstract 3](#_Toc130332033)

[1.Introducere 4](#_Toc130332034)

[*1.1 Motivarea alegerii temei* 4](#_Toc130332035)

[*1.2 Prezentarea generala a proiectului* 5](#_Toc130332036)

[2. Arhitectura proiectului 7](#_Toc130332037)

[3.1 Frontend 9](#_Toc130332038)

[3.1.1 Angular 9](#_Toc130332039)

[3.1.2 Bootstrap 11](#_Toc130332040)

[3.2 Backend 12](#_Toc130332041)

[3.2.1 Java 12](#_Toc130332042)

[3.2.2 Spring 13](#_Toc130332043)

[3.2.3 Hibernate 13](#_Toc130332044)

[3.3.3 Spring Security 13](#_Toc130332045)

[3.3.4 Spring Rest 14](#_Toc130332046)

[4. Concluzii și implementări viitoare 15](#_Toc130332047)

[Bibliografie 16](#_Toc130332048)

Abstract

Scopul acestui proiect este realizarea unui sistem informatic si de gestiune al zborurilor. Obiectivul este acela de a implemneta o aplicatie web completa, complexa si usor de manevrat pentru toate tipurile de utilizatori, dar si pentru administratorii acesteia.

Această aplicație web va fi împărțită în doua mari categorii: Frontend și Backend utilizându-se pentru fiecare diferite tehnologii care vor fi detaliate mai pe larg în capitolele ce vor urma.

Prima categorie, partea de la Frontend, are scopul de a realiza o interfata dinamica si usor de manevrat pentru utilizatori. Pentru realizarea ei se va folosi un framework de Java Script, Angular, ce se bazeaza pe Typescript si Bootstrap pentru adaugarea componentelor de interfata, dar cuprinde si o serie larga de librarii utile in implementarea de noi functionalitati.

A doua categorie a proiectului, Backend-ul, a fost realizata folosind limbajul de programare Java, folosind pattern-ul arhitectural Spirng MVC (Model-View-Controller) pentru a realiza RESTful Web Services. Împreună cu Spring MVC vom mai folosii tehnologii precum Hybernate care ne va ajuta la maparea domeniilor la o bază de date relațională și framework-ul Spring Security pentru autentificare și autorizare.

Ca și sistem de gestiune a bazelor de date relaționale vom utiliza MySQL care este un SGBD open-source.

1.Introducere

## *1.1 Motivarea alegerii temei*

Motivul alegerii unei aplicatii Web in domeniul aviatic ca si tema de disertatie a aparut datorita domeniului in care am studiat si datorita dorintei de a realiza o astfel de aplicatie utilizand diferite tehnologii care combinate sa produca un produs finit.

Mai mult decat atat aceste aplicatii sunt foarte des intalnite in zilele noastre si acest fapt se datoreaza avantajelor acestora:

* Compatibilitatea dintre sisteme – majoritatea aplicatiilor web trebuie sa indeplineasca o cerinta minima si anume existenta unui browser precum (Micrososft Edge, Google Chrome, Safari), astfel indiferent de sistemul de operare folosit aplicatiile web pot rula cu usurinta.
* Reducerea Costurilor – aplicatiile web pot reduce costurile atat pentru furnizor datorita costurilor mici pentru suport dar si pentru partea de intretinere cat si pentru utilizator deoarece sistemul acestuia are nevoie de specificatii minime pentru a putea utiliza aplicatia.
* Mult mai gestionabil – dupa cum am spus mai sus aplicatiile web necesita cerinte minimme din partea statiilor de lucru ale utilizatorului si impreuna cu avantajul ca acestea sunt instalate doar pe server, actualizarea sistemului este se poate face foarte rapid si foarte simplu.

## *1.2 Prezentarea generala a proiectului*

În această lucrare de disertație am ales să dezvolt un sistem informatic de gestionare al zborurilo, mai exact o aplicație web care se va focusa atât pe utilizatori cât și pe partea de administrare.

Numele aplicației este FlyBird și va avea o pagină principală în care se vor găsi informații generale despre Companie. Navigarea pe pagină se va face din bara de navigare din partea de sus a site-ului. Din această zonă, se poate naviga foarte usor către diferite secțiuni ale site-ului:

Din perspectiva unui utilizator vom avea urmatoarele zone care vor contine diferite functionalitati:

* Info unde:
* userii vor putea descoperii informatii despre avioanele puse la dispozitie de catre companie, cu specificatiile fiecaruia dintre ele;
* detalii despre clasele de travel ( 3 la numar ) : Economy, Premium Economy si Bussines Class;
* Următoarea secțiune este Fly unde:
* se vor afla informatii despre Check-in, statusul zborului si mobile check-in;
* A treia secțiune prezentă este Plan unde:
* se vor afla informatii legate de bagaje si cerinte speciale pentru: animale, persoane cu probleme medicale;
* A patra secțiune este secțiunea de Book unde:
* se poate face booking unui zbor;
* fiecare utilizator isi poate vizualiza zborurile rezervate;
* se regasesc oferte facute de catre companie;
* A cincea sectiune va face parte din pagina de profil a utilizatorului, unde se vor afla date personale despre acesta ca de exemplu: Nume, Prenume, Data nasterii, Oras, CNP, Numar de telefon, Email.
* Ultima zonă este reprezentată de pagina de autentificare și de înregistrare, unde utilizatorii își pot crea contul și ulterior se pot autentifica beneficiind de secțiunile menționate mai sus. To aici, utilizatorii isi vor schimba parola in cazul in care acesta a uitat-o.

Din perspectiva unui administrator vor exista toate paginile pe care le poate vizualiza și utilizatorii cu toate funcționalitățile acestora, însă va avea și o secțiune specială numită Administration care va reprezenta interfața de administrare a aplicației, specială doar utilizatorilor cu rol de admin, unde se vor putea efectua și vizualiza următoarele date:

Pe langă cele menționate mai sus, va exista posibilitatea filtrării entităților în tabelele descrise, astfel reusind să se eficientizeze și să se ușureze procesul de căutare făcut de către administrator în privința anumitor date.

2. Arhitectura proiectului



Figură 1. Arhitectura Apliației Web. Sursă Proprie.

În continuare vom analiza diagrama de mai sus:

Frontend-ul reprezintă componenta aplicației web, pe care utilizatorul o poate vedea și cu care poate interacționa. Această componentă a apliacției este împărțită la rândul ei în două sub-categorii: partea estetică (design-ul) și dezvoltarea implementării interfeței. Partea estetică a aplicației se referă la: meniuri, navigatia platformei, designul butoanelor, fonturile, caruselele cu imagini. În timp ce dezvoltarea aplicației se focusează pe modul în care de exemplu, funcționează butoanele și dacă acestea îndeplinesc corect funcționalitățile pentru care au fost create.

Frontendul nu face logică de business adică nu prelucrează datele. Scopul lui este doar de a colecta datele și de a le afișa utilizatorului sub o formă cât mai prietenoasă. Acesta va face un Request catre Backend acolo unde se face prelucrarea de date.

Pentru a se procesa datele avem nevoie de un server de Backend. Aceasta este componenta care stochează datele și care comunică cu baza de date, comunică cu diferite sisteme de fișiere și poate comunica chiar cu alte sisteme de backend și face toate prelucrările necesare. După aceasta etapa a prelucrărilor, serverul va trimite un Response către aplicația de Frontend, iar aceasta la rândul ei va afișa rezultatele către user.

3. Tehnologii folosite

3.1 Frontend

Pentru implementarea Frontend-ului ne vom ajuta de Visual Studio Code care reprezinta un editor de cod sursa foarte puternic reusind sa ruleze pe toate sistemele de operare. Acest tool este foarte utilizat datorita suportului cu care vine incorporat pentru JavaScript, TypeScript si Node.js. Cu toate acestea a devenit foarte populat si pentru limbaje de programare precum Java, C# sau Python datorita ecosistemului foarte bogdat de extensii pentru aceste limbaje. [[1](#Vis22)]

3.1.1 Angular

Angular este o platforma de dezvolatare construita de catre TypeScript. Acesta este proiectat sa trimita update-urile facute cat mai rapid, avand astfel posibilitatea sa profiti de acestea aproape instant.

Fiind o platforma Angular include:

* Un framework pentru construirea de aplicatii web scalabile;
* O colectie variata de librarii integreate care ajuta la implementarea diferitor caracteristici;
* O multitudine de instumente care te pot ajuta sa iti dezvolti, constuiesti si testezi codul.

Pentru o mai buna intelegere a modului cum functioneaza Angular vom caracteriza mai departe componentele de baza ale acestuia:

* Modulele – Aplicatiile Angular sunt modulare si astfel Angular are propriul sistem de modularitate numit NgModules. NgModules reprezinta containare pentru un anumit bloc de cod dedicat unui domeniu al aplicatiei, sau a unui workflow. Fiecare aplicatie Angular are cel putin un modul, care reprezinta radacina si care este denumita AppModule, reprezentand mecanismul de bootstrap care lanseaza aplicatia.
* Componentele – reprezinta blocurile din care este compusa aplicatia. Componenta include o clasa TypeScript reprezentata de o adnotare @Component(), un template HTML si stilurile. Clasa @Component() specifica urmatoarele informatii:

1. Un selector CSS care defineste modul in care componenta este folosita in template.
2. Un template HTML care indica cum sa fie randata componenta;
3. Un set de stiluri CSS care ajuta la design-ul elementelor HTML.

* Servicii si Dependency Injection – pentru date sau logica care nu sunt impartite intre componente, se va crea o clasa service. O astfel de clasa este imediat precedata de adnotarea @Injectable(). Aceasta adnotare ofera metadatele care permit altor furnizori sa fie injectati ca si dependinte in clasa ta. Cu ajutorul Dependency Injection componentele vor fi mult mai eficiente deoarece nu vor prelua datele de pe server si nu se vor conecta direct la consola aceste sarcini se vor delaga catre servicii. [[2](#Ang22)]



Figură 2. Diagrama Arhitecturala a unei Aplicati Angular [[3](#Ang221)]

3.1.2 Bootstrap

Bootstrap este un framework open source de frontend, care contine elemenete de baza pentru dezvoltarea aplicatiilor web receptive. Acest framework este format pe baza HTML, CSS si JavaScript.

Cu ajutorul Bootstrap se poate detecta dimensiunea ecranului pe care o foloseste utilizatorul, adaptand dimensiunile componentelor aplicatiei web automat. Pe langa aceasta functionalitate de baza, printre componentele incluse in interfetele Bootstrep se regasesc barile de navigatie, carusele de imagini si butoane.

Printre avantajele utilizarii acestui framework se regasesc:

* Fiind responsive, ajuta la performantele aplicatiei web deoarece tot procesul de optimizare este cu mult ameliorat.
* Compatibilitatea cu diferite browsere web;
* Fiind un framework foarte popular documentatia este bogata, iar datorita acestui lucru se pot gasi foarte usor exemple de cod care trebuie doar adaptate la necesitatile fiecaruia.

Exista si mici dezavantaje in folosirea Bootstrap cum ar fi:

* Necesitatea ca browserele sa fie updatate, deoarece daca se foloseste o versiune noua a frameworkului, iar versiunea browserului este mai veche vor exista discrepante.
* Daca se va utiliza un numar mare de functii va rezulta la dimensiuni ale fisierelor foarte mari ceea ce poate duce la ingreunarea proiectului. [[4](#Jor22)]

3.2 Backend

Pentru implementarea Backend-ului ne vom ajuta de IntelliJ IDEA care reprezinta un mediu de dezvoltare integrat (IDE). Este unul dintre cele mai utilizate IDE-uri pentru dezolvatarea aplicatiilor software in limbajul Java.

Printre caracteristicile sale cele mai importante se numara:

* Integrarea cu instrumente de control al versiunii precum Git, SVN;
* Suport pentru dezvoltarea de aplicatii web prin framework-uri popilare precum Spring avand capacitatea de a rula si testa aplicatiile local sau pe servere externe.

3.2.1 Java

Java este un limbaj de programare orientat pe obiecte. A fost lansat pentru prima data in anul 1995 si a devenit foarte popular in lumea dezvoltarii software, fiind utilizat intr-o foarte mare varietate de aplicatii si sisteme.

Este un limbaj de programare compilat si interpretat, ceea ce inseamna ca codul sursa Java este transformat intr-un format intermediar (bytecode) care apoi este interpretat de masina virtuala Java (JVM) pentru a rula programul. Astfel aplicatiile Java pot rula pe aproape orice platforma, fiind foarte usor sa fie portabile.



Figură 3. Arhitectura Java [[5](#Jav)]

3.2.2 Spring

Spring Framework este este un framework open-source pentru dezvoltarea aplicatiilor in Java. Acesta ofera o gama laga de functionalitati care fac ca dezvoltarea aplicatiilor web se fie mai usoara si mult mai eficienta. Printre acestea se numera:

* Inversion of control (Inversiunea de control) si dependency injection (injectia de dependinte). Spring usureaza crearea obiectelor si gestiunea dependentelor dintre acestea prin intermediul unui container care se ocupa cu crearea, configurarea si gestionarea ciclului de viata al acestora.
* Securitatea: Spring pune la dispozitie diferite instrumente pentru securizarea aplicatiiei atat din punct de vedere al autentificarii cat si din punct de vedere al autorizarii.
* Accesul bazei de date: Spring ofera un set de instrumente pentru accesarea bazelor de date, prin intermediul framework-ului Hibernate.

3.2.3 Hibernate

Hibernate ofera o solutie de mapping obiect-relational (ORM) care faciliteaza interactiunea dintre bazele de date relationale si obiectele create in Java. Astfel programatorul lucreaza intr-un mod mai natural, folosind obiecte si metode, in loc sa scrie interogari SQL.

Hibernate este proiectat astfel incat sa se poata integra cu alte framework-uri si librarii precum Srping si Struts.

3.3.3 Spring Security

Spring Security ofera un set de instrumente pentru ajutorul implementarii fucntionalitatilor de autentificare, autorizare si gestionare a sesiunilor in aplicatiile web.

Acesta este folosit pentru a prevenii atacurile de tipul Cross-Site Scripting, SQL injection, Cross-Site Request Forgery.

3.3.4 Spring Rest

Spring MVC este un framework de dezolvare a aplicatiilor web in Java.

Abrevierea MVC vine de la Model-View-Controler ce reprezinta un pattern arhitectural prin care se separa logica in:

* Model – reprezentand obiectele si logica acestora;
* View – reprezentand aspectul vizual al aplicatiei si defineste modul in care datele sunt afisate utilizatorului.
* Controller – gestioneaza interactiunea dintre model si vizualizare. Aici se primesc cereri HTTP de la utilizatori si de face procesarea datelor.

Spring REST ( Representational State Transfer) este o extensie a framework-ului Srping MVC axandu-se pe dezolvatarea serviciilor RESTful in Java. Aceste servicii web reprezinta o modatlitate foarte populara de la construii servicii web care se bazeaza pe protocale HTTP pentru a permite mai usor comunicarea dintre client si server. Prin aceasta extensie serviciile web RESTful pot comunica cu alte aplicatii si sisteme prin intermediul formatelor XML, JSON.



4. Concluzii și implementări viitoare

# Bibliografie

x

|  |  |
| --- | --- |
| [1] | Visual Studio Code. (2022, Aug.) Getting Started. [Online]. <https://code.visualstudio.com/docs> |
| [2] | Angular. (2022, Aug.) Introduction to Angular concepts. [Online]. <https://angular.io/guide/architecture> |
| [3] | Angular. (2022, Aug.) Diagrama Arhitectura Angular. [Online]. <https://angular.io/guide/architecture> |
| [4] | Jordana A. (2022, July) What Is Bootstrap? [Online]. <https://www.hostinger.com/tutorials/what-is-bootstrap/> |
| [5] | JavaTpoint. [Online]. <https://www.javatpoint.com/java-architecture> |

x