Assigment 1.2

Aplicatie web ce foloseste paradigma request/reply

Realizata de,

Bursuc Andrei

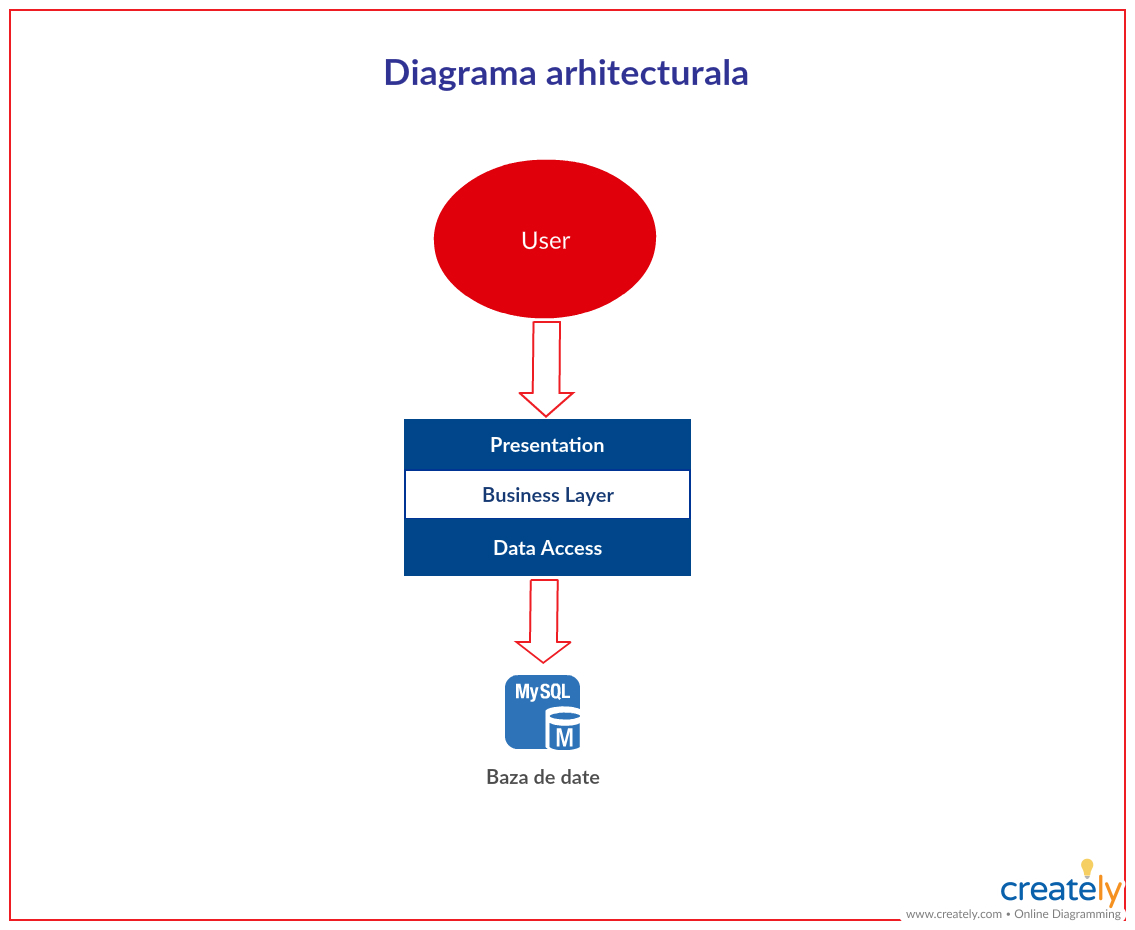
Grupa 30642

## 2. Cerinte

Proiectati, implementati si testati un sistem distribuit pe trei nivele ce permite vizualizarea si editarea zborurilor unui aeroport. Sistemul trebuie sa fie format din urmatoarele nivele: Presentation, Business Layer si Data Access.

## 3. Arhitectura conceptuala

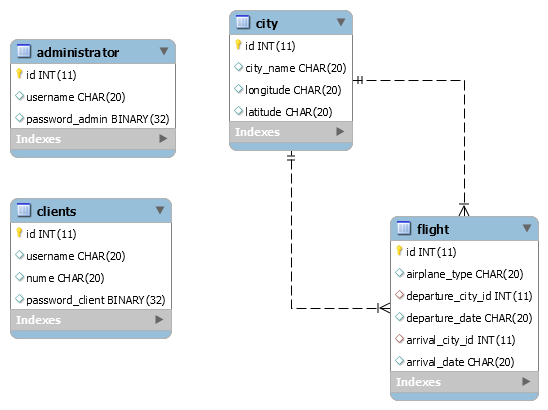
Asa cum specifica si cerinta arhitectura este formata din trei mare nivele care comunica doar unidirectional, in sensul in care in Presentation sunt apelate metode din Business Layer, iar cel din urma la randu lui foloseste metode din Data Acces, intr-un final prin Data Acces se interactioneaza in mod direct cu baza de date prin intermediul Hibernate ORM.



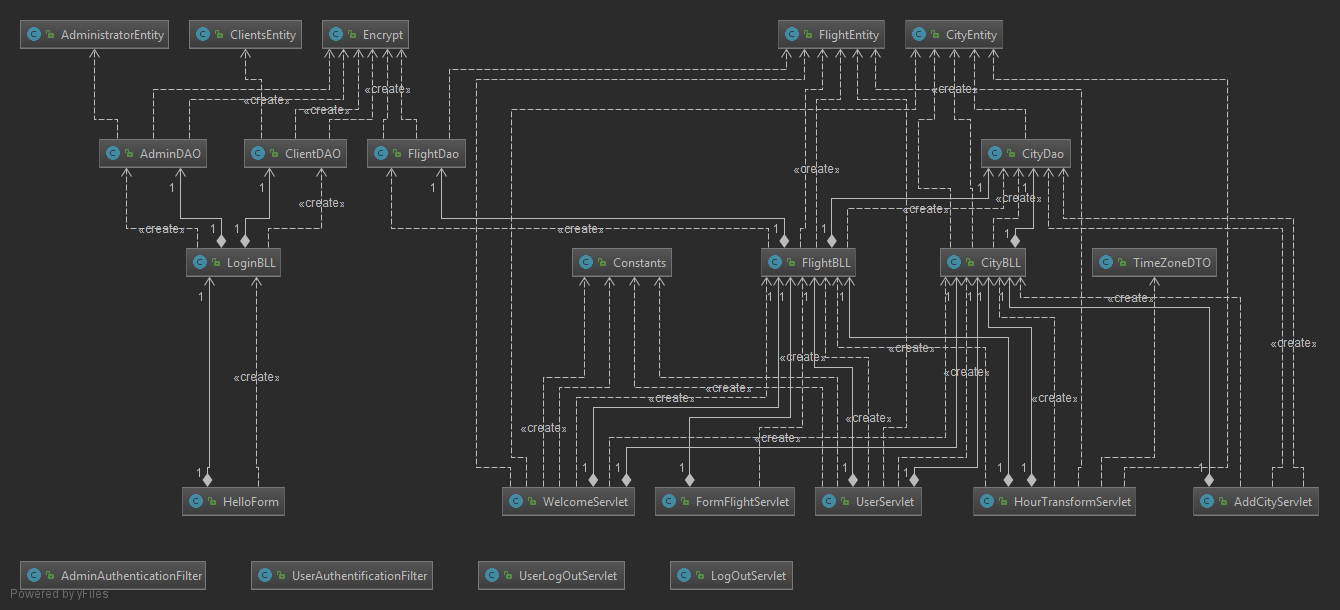
Interfata grafica a utilizatorului este generata prin intermediul HTML si CSS. Practic primul nivel, cel de Presentation este format dintr-un fisier index ce modeleaza pagina de logare si de codul HTML generat de servleturi. Prin intermediul form-urilor din paginile HTML se apeleaza metode din servleturi ce interactioneaza mai departe cu urmatorul nivel, cel de Business Layer. Requesturile trimise de catre paginile HTML sunt fie GET, fie POST, fiecare din acestea doua avand un scop specific. Get-ul se foloseste atunci cand se doreste preluarea de date din baza de date, pe cand Post-ul este folosit atunci cand sunt trimise date ce urmeaza sa modifice baza de date. Urmatorul nivel cel de Business Layer acumuleaza toata logica folosita pentru a efectua actiuni mai complexe folosind metodele din Data Acces, astfel se evita prezenta unei logice mai complexe in nivelul anterior. Ultimul nivel comunica prin intermediul Hiberante ORM direct cu baza de date, incapsuland logica necesara efectuarii query-urile la nivelul bazei de date.

## 3. Proiectarea bazei de date

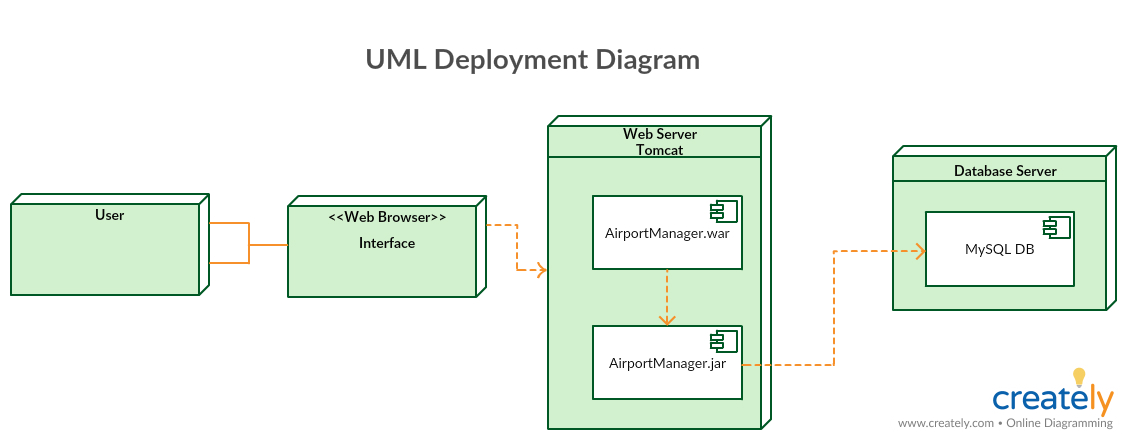
Baza de date este alcatuita din patru tabele: “administrator” contine datele de logare (usernameul si parola encodata cu SHA256 ) pentru administratorii aplicatiei; “clients” este asemanatoare cu tabela anterioara doar ca datele de aici apartin utilizatorilor obisnuiti ce pot doar vizualiza tabela de zboruri; “flights” contine datele zborurilor din baza de date, mai exact tipul aeronavei, orasul de unde decoleaza zborul si ora la care o face, orasul in care acesta aterizeaza si ora la care o face; “city” contine datele legate de orasele intre care se efectueaza zborurile prezente in baza de date, mai exact numele orasului si latitudinea si longitudinea unde acesta se afla. Intre tabele flight si city exista doua relatii one-to-one, un zbor continand doua id-uri de orase, unul pentru cel din care decoleaza si unul pentru cel in care aterizeaza.



## 4. Diagrama UML de clase



## 5. UML Deployment diagram



## 6. Mod de utilizare a aplicatiei

Inainte de pornirea propriu-zisa a aplicatiei trebuie mai intai rulat scriptul de MySQLcu interogarile ce initializeaza baza de date cu care aplicatia comunica.

Ulterior aplicatia se poate deschide in IntelIJ ca si proiect Maven si de acolo ea poate fi rulata pe un server Tomcat ce a fost configurat in prealabil. Serverul poate fi configurat din cadrul meniului Edit Configurations din IntelIJ.

Dupa rularea aplicatiei pe server in functie de portul pe care serverul ruleaza interactiunea cu interfata utlizator a aplicatiei se poate face prin accesarea unei adrese web de tipul: “[http://localhost:8080](http://localhost:8080/)", unde 8080 este numarul portului standard.