**Images:**

**User requests**

**Main / User / Anonymous**

**Q**

**Q**

**Q**

**Airline Company**

**Administrator**

**Customer**

**Q**

**Q**

**Q**

**Q**

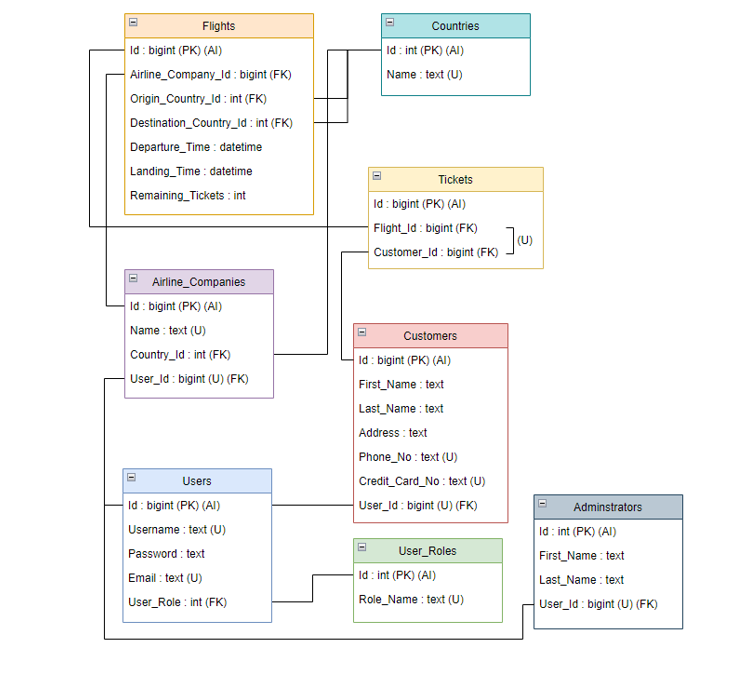
**Q**

**Data layer**

**Q**

**DB**

**DB:**



**דגשים:**

* MySQL
* לבדוק האם ניתן להוריד DATA מהאינטרנט או להתחבר ל-DB קיים
* לשים לב שאותו הלקוח אינו יכול לרכוש פעמיים כרטיס לאותה הטיסה, לכן השילוב של מספר הטיסה ומספר הלקוח הוא Unique. - טבלת **Tickets**.
* טבלת Countries - \*אתגר: הוסף שדה של תמונה (דגל) לכל מדינה
* טבלת Users - (thumbnail) \*אתגר: הוסף שדה של תמונה לכל משתמש
* יש לצרף סקריפטים להכנסת המידע הראשוני.
* בפרויקט יהיה קובץ config ובו נשמור את פרטי הגישה (Connection string) ל-DB שלנו.

**Data layer:**

* **ה-DAL יהיה מחולק ל-2: Functions ו- Models**

**Models:**

* **כפי שמצוין בDB.**

**Functions:**

**get\_airline\_by\_username(\_username text)**

**get\_customer\_by\_username(\_username text)**

**get\_user\_by\_username(\_username text)**

**get\_flights\_by\_parameters(\_origin\_counry\_id int, \_detination\_country\_id int, \_date date)**

**get\_flights\_by\_airline\_id(\_airline\_id bigint)**

**get\_arrival\_flights(\_country\_id int)**

**get\_departure\_flights(\_country\_id int)**

**get\_tickets\_by\_customer(\_customer\_id bigint)**

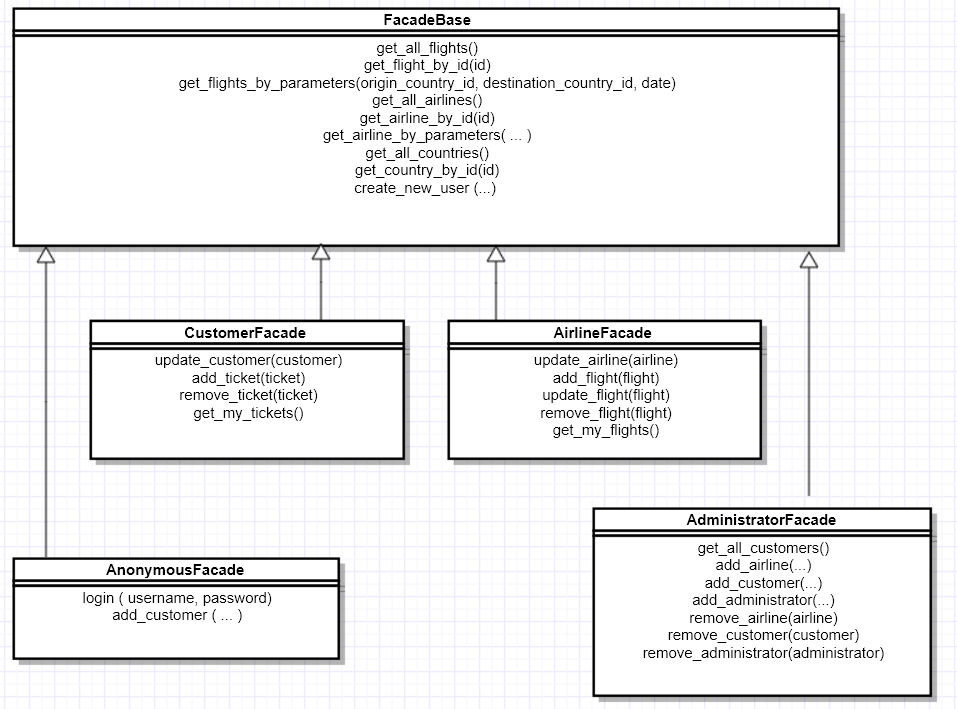
* getAirlinesByCountry(country\_id)
* getFlightsByOriginCountryId(country\_id)
* getFlightsByDestinationCountryId(country\_id)
* getFlightsByDepartureDate(date)
* getFlightsByLandingDate(date)
* getFlightsByCustomer(customer)

**דגשים:**

* יש לבצע את כל פעולות ה-CRUD עבור כל הטבלאות.
* על כל פונקציה שתופעל ב- DAL יש לכתוב לקובץ לוג תיעוד לפעולה
* יש לדאוג להשתמש ב- try except finally ו/או with בהתאם לצורך

!!! להוסיף ולסדר את הפונקציות בכל האפשרויות הקיימות בטבלאות ה DB!!!

**Business Logics:**



**כל אפליקציית** Business Logic **תהיה מורכבת מ- 4 קבצים:**

* **Views**
* **Validations**
* **Façade**
* **ApiViews**
* **כלומר, בחלק הזה נעבד את הנתונים מרמת הrequest עד לרמת המידע המוכן**

**(קובץ Jason).**

**שלבי התהליך בשלב זה הם:**

1. **קבלת מידע**
2. **בדיקה האם המידע תקין**
3. **עיבוד המידע**
4. **שליחת המידע חזרה לviews של ה- Main**

דגשים:

* **מחלקת AnonymousFacade:**תירש ממחלקת **FacadeBase וכל שאר הfasades ירשו מ- AnonymousFacade.**
* **כל fasade יהיה באפליקציה אחרת, ו- AnonymousFacadeיהיה חלק מהאפליקציה Main.**

**יש לעשות דף וולידציות עבור כל אפליקציה** לדוגמא:  
-

- לא ניתן ליצור טיסה (flight) עם שדה remaining\_tickets שלילי (למשל 1-)  
- לא ניתן ליצור טיסה שבה זמן הנחיתה מתרחש לפני זמן ההמראה  
- לא ניתן ליצור טיסה שבה מדינת המקור ומדינת היעד הן אותה המדינה  
- לא ניתן ליצור משתמש (user) עם סיסמא קצרה מ-6 תווים (\*רשות)  
- איסור על חברת תעופה לערוך טיסה של חברה אחרת

- ועוד…

**Main:**

**User**

**Anonymous**

**Main**

Image זה מורכב מ- 3 אפליקציות: Main, User ו- anonymous

**Main:**

אפליקציה שמכילה:

* **Views**
* **FacadeBase**
* **ApiViews**
* האפליקציה הראשית אליה מתקבלות כל הבקשות מאפליקציית המשתמש.

**Anonymous:**

אפליקציה שמכילה:

* **Views**
* **Validations**
* **AnonymousFacade**
* **ApiViews**
* **משתמש ברירת המחדל**

**User:**

אפליקציה שמכילה:

* **Login**
* **Register**
* **factory**
* **משמש כנתב בין כל שאר המשתמשים**

|  |  |
| --- | --- |
| **כרטיסים** | **לקוחות** |
| **1** | **א** |
| **2** | **ב** |
| **3** | **ג** |
| **4** | **ד** |
| **5** | **ה** |
| **6** | **ו** |
| **7** | **ז** |