**MVC**

*Model*:

* lucreaza cu DB(tripdb): insereaza, sterge linii, face update tabelelor;

*View*:

* preia inputuri/request-uri direct de la user si apeleaza Controllerul; afiseaza contentul specific dup ace primeste raspunsul la cererea facuta;

*Controller*:

* preia cererile de la View, proceseaza datele, apeleaza metodele Modelului si trimite raspunsul inapoi la View sau intializeaza interfata direct;

|  |  |  |
| --- | --- | --- |
| Models | Views | Controllers |
|  | | |
| Model.php  -checkDBConnection() | View.php  -initHeaderLocation() | Controller.php  -constructor(model, view)  -runApp() |
| insert-user.php  - validateUid(uid) : Boolean  - insertIntoDB(uid, mail, hash-pwd) : DbRow | myaccount.php  -form-signup submit | signup-ctrl.php |
| login-user.php  - verifyUid(uid, pwd) : DbRow | myaccount.php  -form-login submit | login-ctrl.php |
|  | header.php | logout-ctrl.php  -sesion\_unset()  -sesion\_destroy() |
| get-set-profile.php  - getProfile(uid) : String  - getPicture(uid) : String  - updatePicStatus() | header.php (  - user-index.php  - user-stuff.php  - user-notifications.php) | profile-render.php  - getUserName() : String  - getUserPic() : String |
| ---------------------------- |  |  |
| modify-trips.php  - insertTripIntoDB(tid, uid, name, **status**, interests, fromcity, maxbudget, flight, leaving, coming, mindeg, maxdeg, sky, humidity)  - get-trip-status(tid, uid)  add-notif.php  - insert-notif(uid, tid) | user-index.php  - new-topic-form submit  add-trip.js  - addNewTrip(event) -> **ajaxFunct()** | new-trip-ctrl.php  - callModelForTripInsert()  - tripSearch(uid, tid){  …  get-trip-status()  …  callTheAPIs()  } |
| modify-trips.php  - change-trip-status() | user-stuff.php  trips-buttons.js  - playWithTrips(event) -> **ajaxFunct()** | moify-trip-ctrl.php |
| get-trips.php  -getTrips(uid) | user-stuff.php | trips-render.php  -getAllTrips() |
| get-notifications.php  -getNotifs(uid) | user-notifications.php | notifications-render.php  -getAllNotifications() |

**Descriere principala:**

Userul face submit la formular => se valideaza toate inputurile la nivel de client (in add-trip.js) => se apeleaza ajaxFunct() care va face un request asincron la server si se ruleaza new-trip-ctrl.php.

Din new-trip-ctrl.php se apeleaza insertTripIntoDB() din modelul modify-trips.php, cu status = 1 apoi se ruleaza la infinit o cautare de bilete pentru trip-ul in cauza. Inainte de fiecare cautare ( apel al functiei tripSearch()) se cere statusul de la modify-trips.php (get-trip-status()) si se verifica statusul.

Daca status = 3 => trip-ul a fost sters => se inchide bucla infinita de cautare

Daca status = 2 => trip-ul este inactiv => bucla inca ruleaza, dar nu se mai cauta

Daca status = 1 => trip-ul este activ =>

- se face API call pentru bilete ( [api.skypicker.com](http://api.skypicker.com/) – kiwi api) [fly\_from, date\_from=date\_to, return\_from=return\_to, flight\_type, price\_to]

- se face API call pentru vreme (<https://openweathermap.org/forecast16> - 16 days weather api) [forecast.temperature.max, forecast.temperature.min, forecast.clouds, forecast.humidity]

Cand raspunsul primit este valid (datele corespund cu cerintele) se apeleaza modelul add-notif.php care va introduce toate datele modelate in tabelul notifications din DB.

Toate informatiile sunt luate din DB cand userul acceseaza contul/una din pagini folosind trip-render.php si notifications-render.php.