

TripInspire Architecture

1. Descriere

TripInspire este o aplicatie Web care ofera sugestii de calatorie pe baza unor constrangeri/preferințe, de exemplu: “doresc o vacanță activă de 7-10 zile la o temperatură mai mare de 20°C la mare” sau “doresc un city break în Europa care să coste mai puțin de 200 EUR de persoană”. Sistemul va gestiona diverse criterii de selectare. Pe baza acestor criterii, se vor monitoriza prețurile biletelor de avion pentru destinațiile ce se potrivesc, alertandu-se (via browser notifications) utilizatorii când criteriile sunt îndeplinite. Un utilizator poate avea una sau mai multe căutări de vacanțe active.

2. Alegerea temei

Am ales acest proiect deoarece consideram ca o astfel de aplicatie este foarte utila in viata de zi cu zi. Prin softul nostru utilizatorii sunt pusi la curent cu cele mai potrivite vacante in functie de preferinte/constrangeri.

3. Utilizatori

Guest: utilizator neautentificat, cu drept de acces exclusiv la datele publice ale aplicatiei.

Paginile ce vor putea fi accesate de acesta sunt: Home, Account, News, About.

Prin urmare, utilizatorul Guest poate realiza urmatoarele actiuni:

- poate crea un cont prin completarea formularului de Signup din pagina Account;
- poate sa isi formeze o idee asupra functionalitatilor aplicatiei (enuntate in Home);
- se poate autentifica printr-un cont deja creat prin completarea formularului din pagina Account, folosind un username si o parola.

User: utilizator cu drept de modificare a datelor din aplicatie asociate contului sau.

Are acces la toate paginile din aplicatie.

- poate completa unul sau mai multe formulare in care isi seteaza preferintele unei vacante (porneste cautari).
- poate dezactiva, activa sau sterge cautari din My Stuff;
- poate primi notificari legate de vacantele active;

Admin: utilizator cu drept de gestiune a tuturor informatiile detinute de aplicatie.

Are acces la orice informatie inregistrata in aplicatie.

Detine drept de stergere de conturi de utilizator si a oricarei informatii inregistrate in baza de date.

4. Use case-uri

1 Utilizatorul acceseaza aplicatia, serverul ruleaza corect

1.1 Intra pe pagina Account si completeaza formularul de SignUp

1.1.1 Inputurile sunt invalide => Cont neinregistrat

1.1.2 Inputurile sunt corecte => Cont inregistrat cu success

1.2 Intra pe pagina Account si completeaza formularul de LogIn

1.2.1 Inputurile sunt invalide => Eroare la login

1.2.2 Inputurile sunt corecte => Cont autentificat cu success

1.3 Intra pe pagina News si vede top 10 cele mai accesibile destinatii

1.2.2.1 Userul completeaza formularul din Home (creaza o cautare noua)

1.2.2.1.1 Date invalide => Mesaj de avertizare

1.2.2.1.2 Date corecte => Se creaza o vacanta noua care este adaugata in baza de date si in pagina MyStuff. Primeste notificari cand se gasesc bilete de avion pentru cautarile sale

1.4 Userul (deja logat) acceseaza MyStuff

1.4.1 Apasa pe 'Activate/Deactivate' => Activeaza/dezactiveaza o cautare

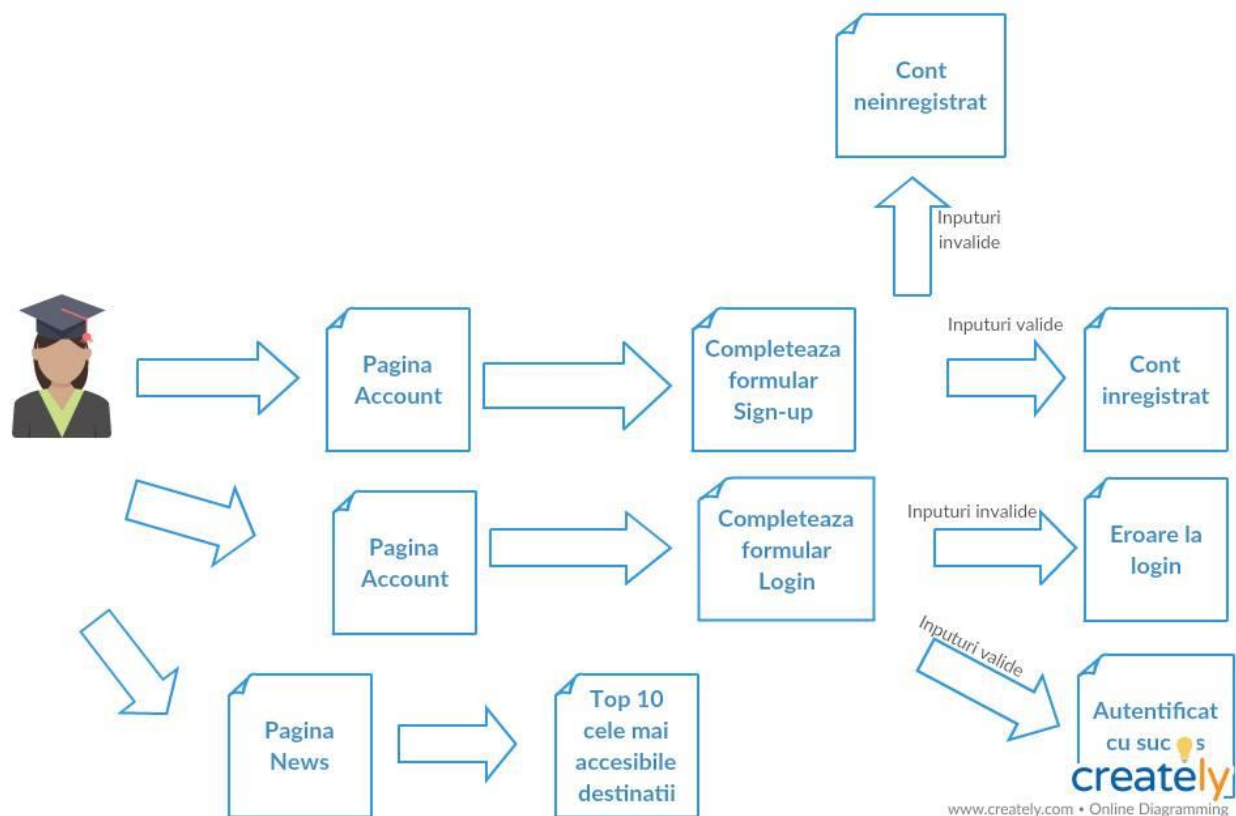
1.4.2 Apasa pe 'Delete' => Opreste complet cautarea si sterge vacanta

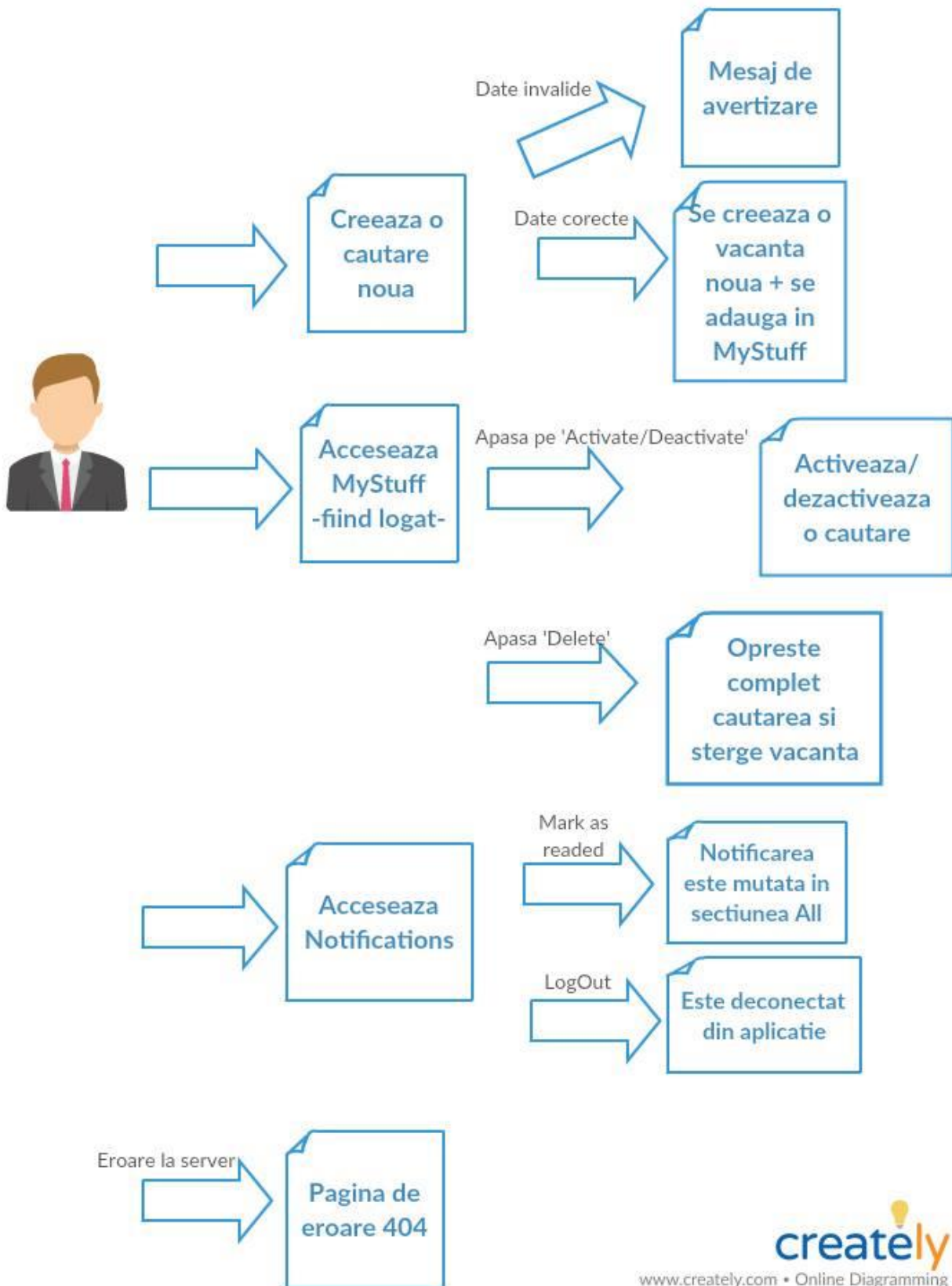
1.5 Userul acceseaza Notifications

1.5.1 Din sectiunea Unread apasa pe 'Mark as readed' => Notificarea este mutata in sectiunea All

1.6 Userul apasa pe LogOut => Este deconectat din aplicatie

2 Utilizatorul acceseaza aplicatia, apare o eroare la server => 404 not found





5. Tehnologii

Basic: HTML5, CSS3, PHP, JavaScript

Baza de date: mysql (localhost)

Scriptul BD (tripdb)

```
create table users (  
    id int(20) not null AUTO_INCREMENT PRIMARY KEY,  
    username varchar(256) not null,  
    email varchar(256) not null,  
    password varchar(256) not null    );  
  
create table profileimgs (  
    id int(20) not null AUTO_INCREMENT PRIMARY KEY,  
    iduser int(20) not null,  
    status int(10) not null,  
    extension varchar(10) not null    );  
  
create table trips (  
    id int(20) not null AUTO_INCREMENT PRIMARY KEY,  
    iduser int(20) not null,  
    name varchar(256) not null,  
    status int(10) not null,  
    interests varchar(256) not null,  
    fromcity varchar(256) not null,  
    maxbudget int(20) not null,  
    flight varchar(256) not null,  
    leaving datetime not null,  
    coming datetime not null,  
    mindeg int(20) not null,
```

```

maxdeg int(20) not null,

sky varchar(256) not null,

humidity int(20) not null      );

create table notifications(

id int(20) not null AUTO_INCREMENT PRIMARY KEY,

idtrip int(20) not null,

iduser int(20) not null,

fromcity varchar(256) not null,

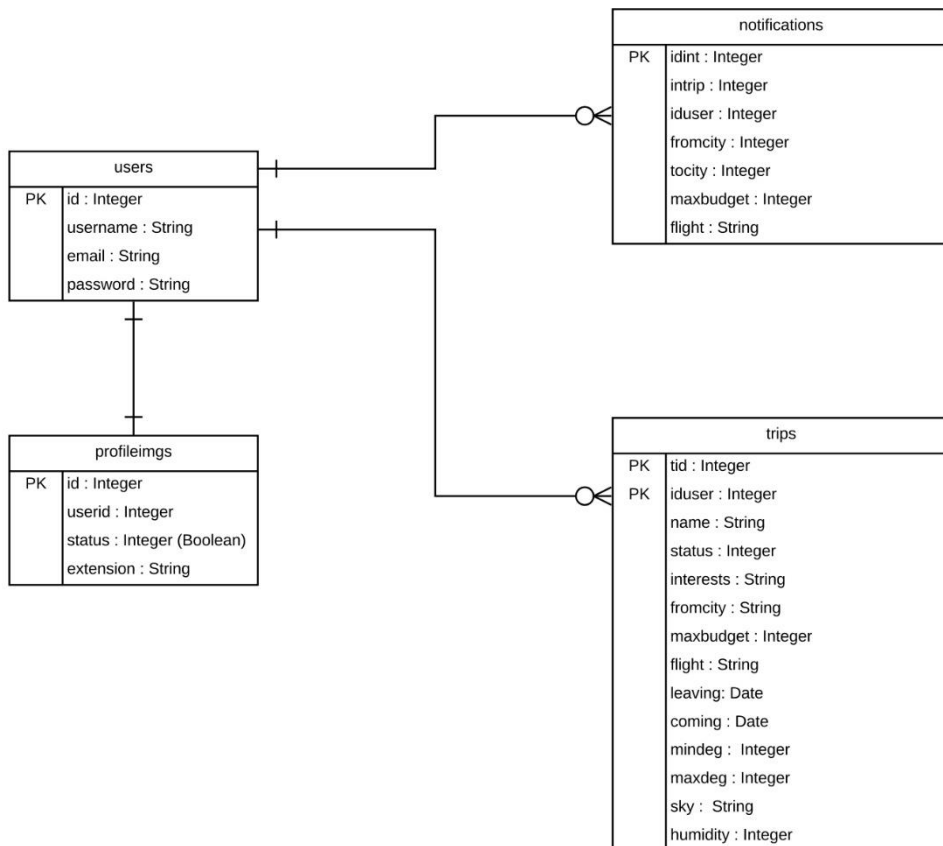
tocity varchar(256) not null,

maxbudget int(20) not null,

flight varchar(256) not null    );

```

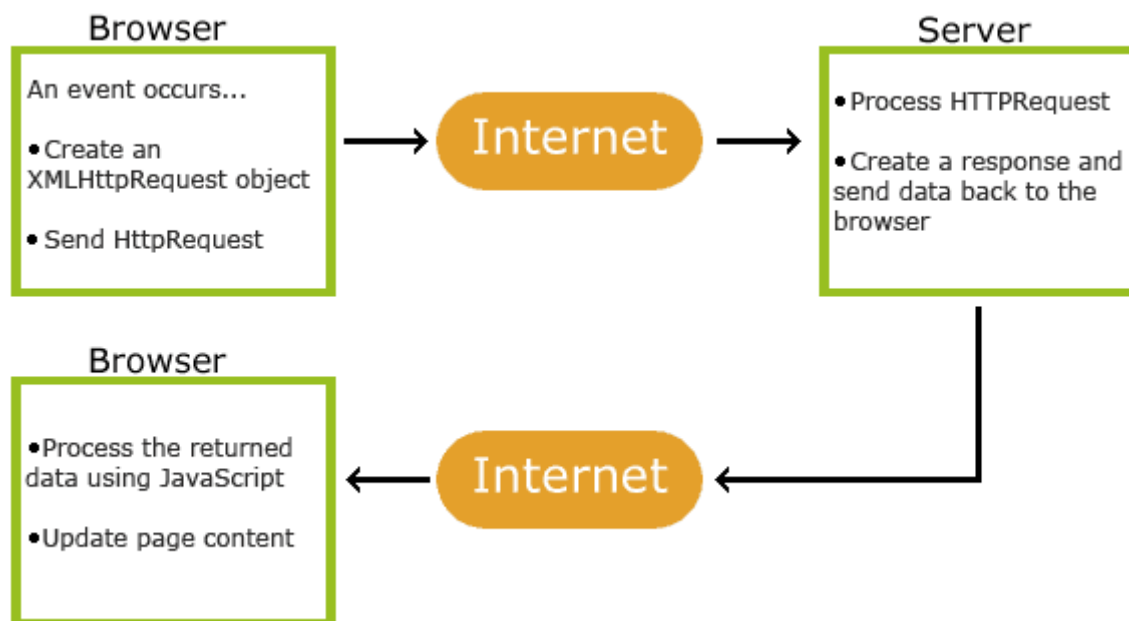
Schema relationala



AJAX (Asynchronous JavaScript And XML):

Este folosit pentru apeluri asincrone catre controllerul care face job-ul de baza (cauta bilete).

Cautarile trebuie facute in background, la un interval specific de timp, fara a intrerupe rulara altor task-uri, prin urmare vom trimite/primi date catre/de la server in mod asincron, fara reload al paginii. La fel se intampla sic and userul modifica o parte din date (de ex. sterge o vacanta).



MVC Pattern (Model-View-Controller)

Model:

- lucreaza cu DB(tripdb): insereaza, sterge linii, face update tabelor;

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 initializeaza 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 - callModelForTriplInsert() - 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()

Descrierea problemei principale:

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 – 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.