ARHITECTURA APLICATIEI LOST PETS

1.Use case:

Aplicatia este accesata de user prin **Home page**, in acesta regasim butoanele de Log in si de **Register**,userul poate da click pe ele si este automat trimis pe pagina de **Log in/Register**, pagina de **Register** poate fii accesata doar de la pagina principala si odata cu introducerea datelor si apasarii butonului submit userul este trimis pe pagina **Log in**, in aceasta userul are posibilitatea de a-si completa datele si la finalizarea acestui pas este trimis pe pagina **Workspace**, in aceasta pagina poate vedea anunturi cu toate animalele pierdute, schimbarea locatiei animalelor pierdute, marcarea oricarui animal cu “Found Pet”, in sectiunea de notificari userul va primi anunturi cu toate animalele pierdute din zona in care se afla, acesta mai are posibilitatea de a-si schimba datele de user,si de a se deloga, tot in **Workspace** se pot gasi diferite statistici legate de zone ( de ex: in ce zona sunt pierdute cel mai frecvent animale), si mai dispune de butonul “Add a lost Pet”, odata apasat acesta userul este trimis pe pagina **Formular** in care poate poate completa cu diferite date ale animalului pierdut.





2.Diagrama Bazei de date3. Patternul MVC

Exemplu de comunicare intre view, controller si model in timpul interactionarii cu aplicatia:

In momentul logarii, ne aflam in pagina Log in, introducem date in casutele username si password si dam submit, in acest moment componenta controller primeste de la view aceste date, username-ul si parola pe care le-am introdus, controllerul paseaza aceste date modelului iar modelul verifica in baza de date, mai exact in tabela user daca aceste date introduse sunt unice, daca da, atunci modelul ia din tabela datele necesare afisarii urmatorului page, le paseaza controllerului, iar acesta le da mai departe view-ului pentru ca acesta sa afiseze pagina Workspace, daca nu, atunci controllerul da mai departe un mesaj de eroare care este mai apoi afisat in view, in pagina Log in.

