System Administrator

User Types

Our project has one user type standard user. He can do several actions an our platform.

User is able to see the home page, to search packages, to download the script and marking packages for installing.

|  |  |  |
| --- | --- | --- |
| **As a(an) Standard User** | **Want to See the main page** | **So that I can know more about project** |
| As a(an) Standard User | Browse in package list | Choose what I need to install |
| As a(an) Standard User | Search in package list | Easily find one specific package |
| As a(an) Standard User | Mark packages to install | Download selected packages and dependences |
| As a(an) Standard User | Download the script | Install the packages |

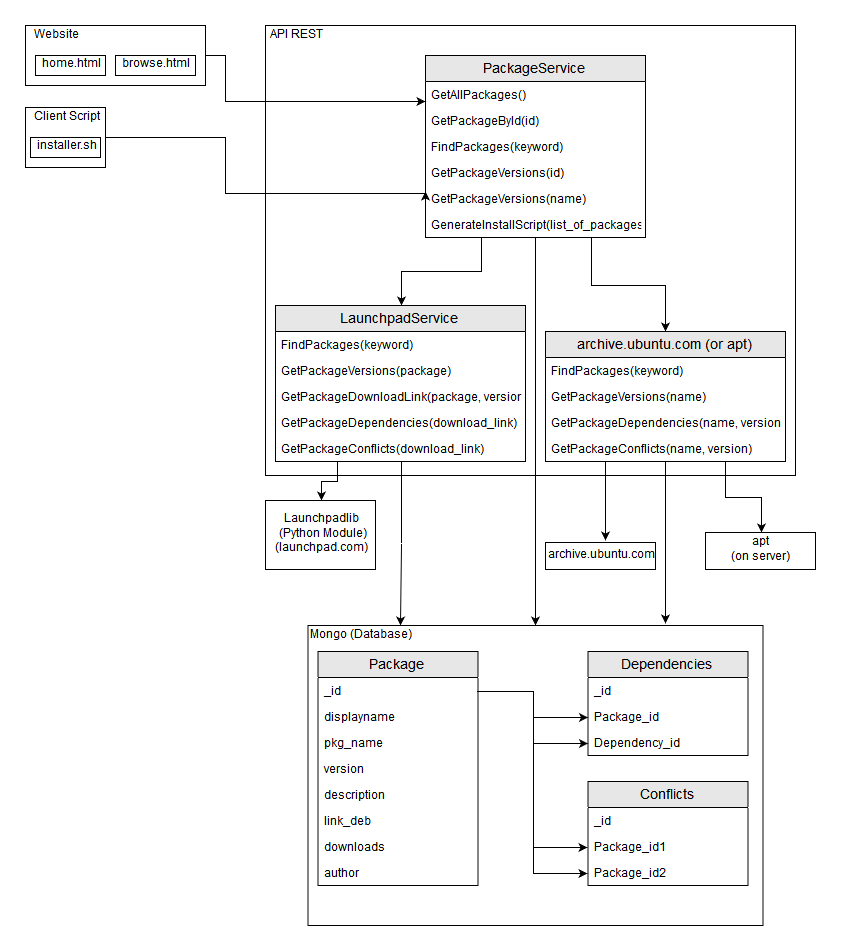
Use cases

Here are a possible use cases that a user may want to know more about:

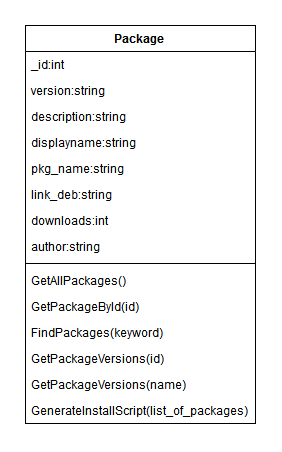
* Somebody wants to see more details about a specific package. So he opens our web aplication, he goes in Browse Page, selects the ubuntu version and searches the package by name. In the displayed list he can see the description of the package, the number of downloads and the latest version of the package. If he wants an older version he can click on the button "all versions" and a list will be displayed containing all versions of the package.
* Somebody wants to install a number of packages on his local machine. He opens the web aplication, opens the Browse Page, he searches for every package and marks them accordingly. After selecting those packages, he can download and run the installer script which will take care of the rest.

Tehnical Details

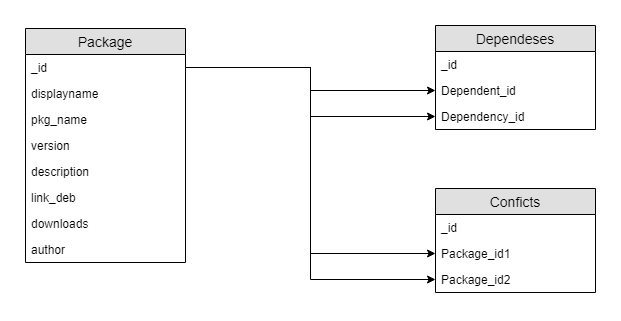
*Arhitectural Diagram*



*Class Diagram*



*Database Diagram*



*Tehnologies:*

* *html*
* *css*
* *javascript*
* *flask*
* *python 2.7*
* *archive.ubuntu.com*
* *mongoDB*
* *apt*

*Details:*

*We are going to use html, css and javascript for frontend, in scope of stylizing and creating.(html for document strucutre and css designing and javascript for frontend processing)*

*Python 2.7 will be used to create a REST API for the aplication. The deb packages will be downloaded from archive.ubuntu.com and mongoDB is going to store the packages information(names, dependencies and conflicts).*

Tasks

The tasks have been distributed to the team members as below:

* Create Database(Andrei)
* Database Acces from API(Andrei)
* Create script for database maintenance(periodicaly checks for packages on launchpad(Eduard)
* Apt module(Andrei)
* Launchpad module(Radu) – Replaced by archive.ubuntu.com
* Document a method of obtaining packages from archive.ubuntu.com and similar website(Radu)
* Package Service(Eduard)
* Integration(Radu, Andrei, Eduard)
* Frontend -> REST API (eduard)