Bow

Techical report

Botanical Garden web manager

Technical report

Authors

Andrei Andrica , Bogdan Suciu

Abstract

Botanical web manager is an application whose goal is to give access to the public to complex information on the collection of plants which can be found in a specific botanic garden. Up until now it was only possible to gain such knowledge by being present at the place, and was often impossible to gain access to the information you were being display and often forced to spend the whole time on researching/documenting each object that was posing an interest. By migrating the information into a web application, it will allow each visitor to retrieve information on a specific plant at any given moment and from any location.

General purposee

The application is intended to give free access to the information on all the plants of the given botanical garden. It also included the possibility of updating it with new information. The user can perform a complex search on the subject that it is interested in, and given the option of exporting the information in various formats such as CSV, XML. ;The search can be performed on multiple criteria, such as English name/Latin name, description, origin, the location where the plants can be found, the option of displaying images of it, reproduction, benefits, the purpose its servers (ornament, scent, medicinal), the environment where it grows (open field, greenhouse).

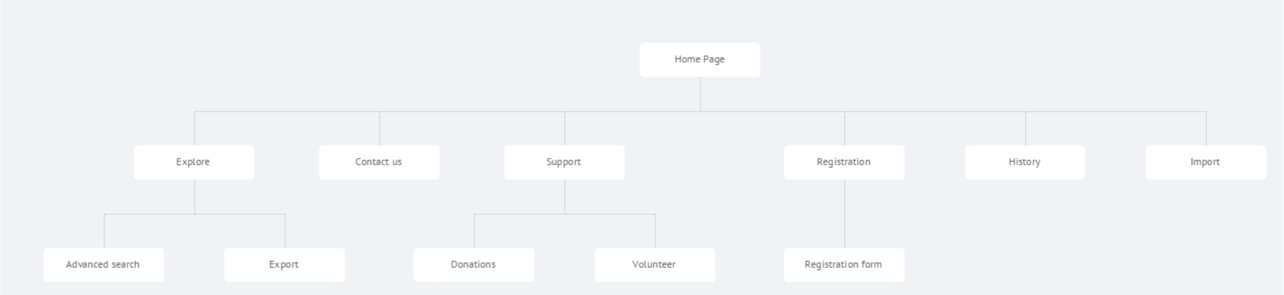
Requirements

1. The application is intended to run on both portable (android, IOs, windows mobile) and standing devices

2. It can be accessed by anyone for free, authentication is not required but recommended and it can only be done via social networks
3. There will be 3 classes of user: visitor/regular, data-entry and super user
4. Certain modules of the application should be restricted to regular user
5. Provide a history of the user actions, such as displaying documents that have been either imported/exported
6. Provide a friendly and attractive interface for the visitor/ regular user
7. The applications should be hosted 24/7 on a dedicated server
8. The implementation should be made so it keeps the amount of resources required to a minimum
9. It is preferred that during the implementation the following resources are used: HTML, CSS, bootstrap framework, JavaScript and it libraries and PHP

Application scheme

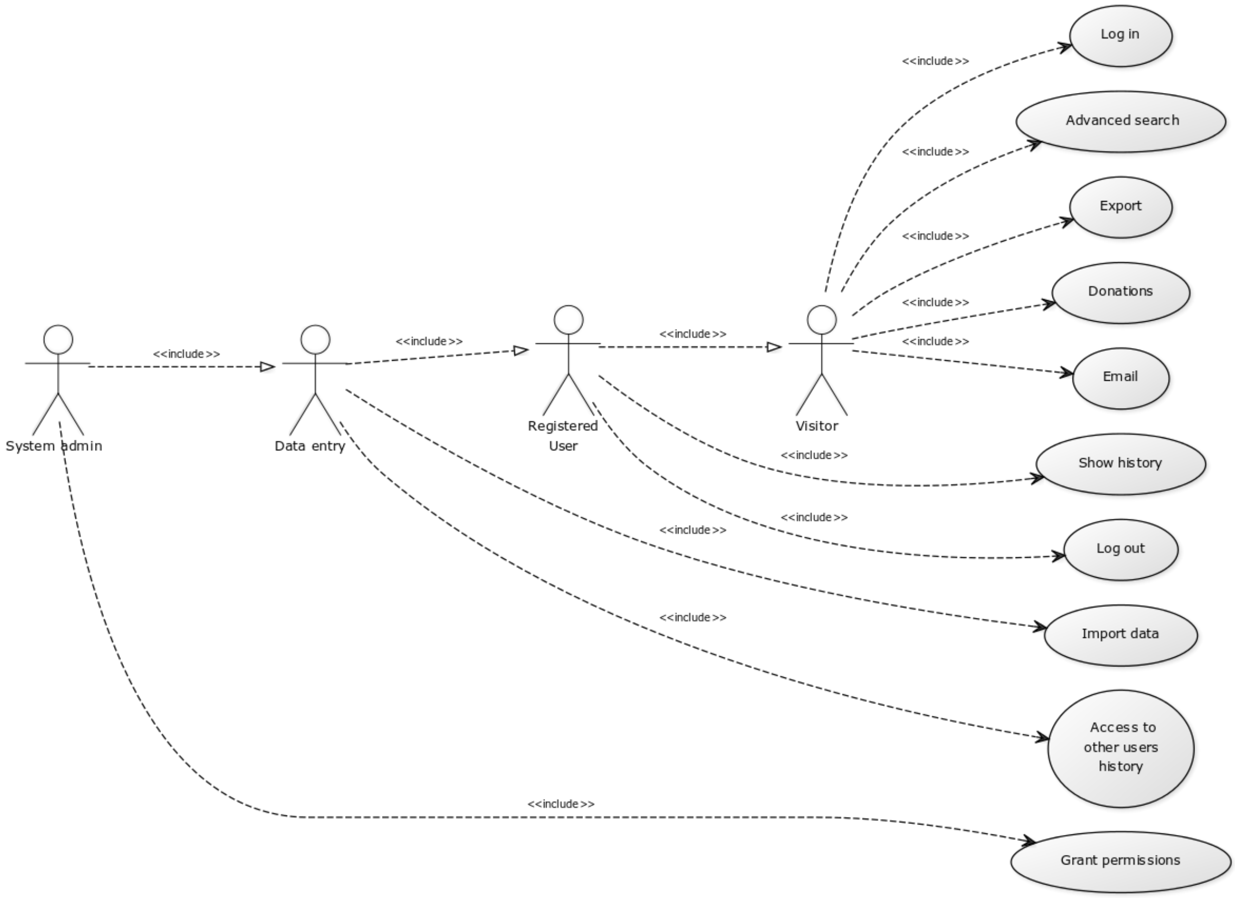
We have devised the following schema which will help in building the frame of the application and based on which we have built the application modules



Application schema

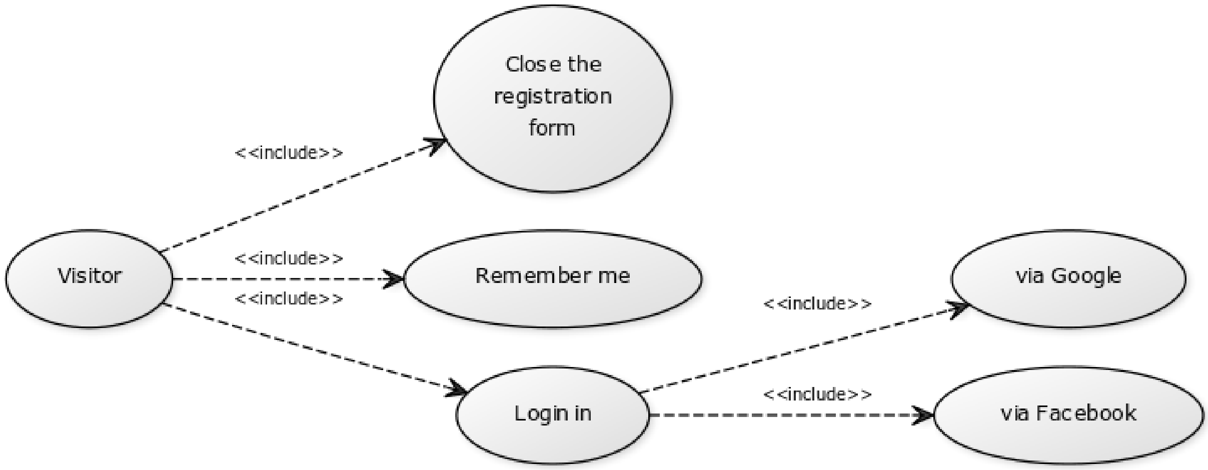
Use cases

Users inheritance



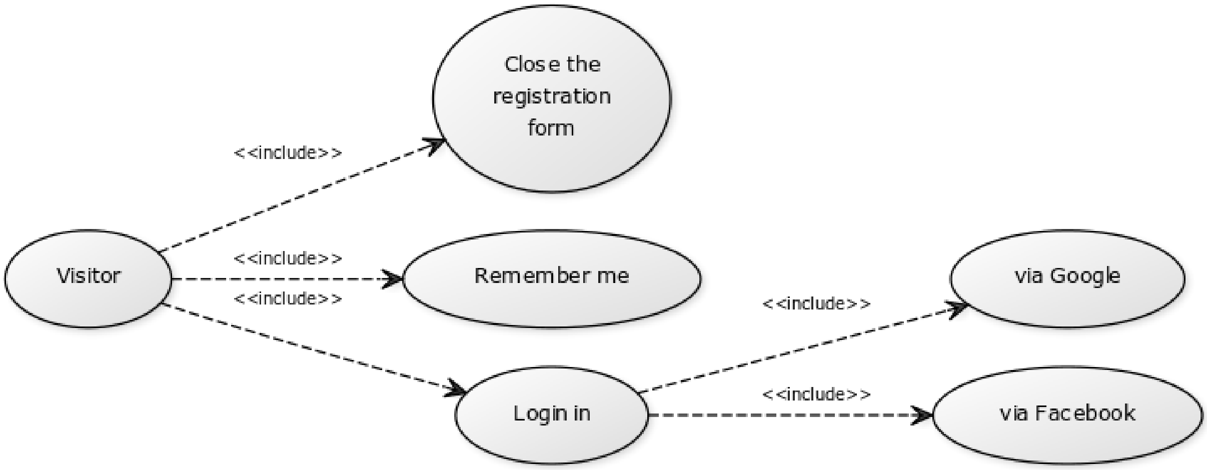
User inheritance

Log in module



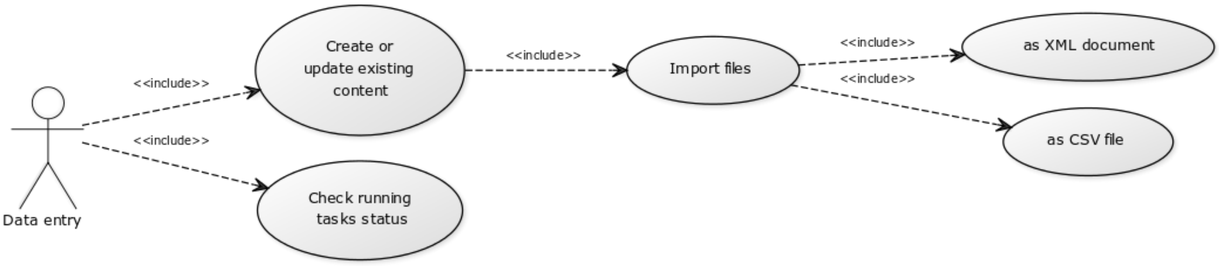
Log module

Export module



Exp module

Import module



Imp module

Application modules

User types

The application users will be divided into 4 categories:

1. Visitors
2. Users
3. Data-entry
4. System administrator

Search module

There will be two types of search options available:

1. Quick search
2. Advanced search

The quick search can be performed after the user types a keyword in the search bar, which is located at the top-side of the screen and it will display the top results that match his interests.

The advanced search will allow the user to perform advanced filtering, based on multiple criteria which can be selected via checkboxes located in a panel, on the left-side of the screen.

Both search function should only be available from the Explore module.

Sign in

The registration can only be done via social networks, Facebook and Google. In order to do that, the unregistered user must action the Sign in button located at the top-right of the page. A pop-up window will appear providing with two option from which it can choose. One of the preconditions for this to happen, is the user should have an active Facebook or Google+ account.

Export

This feature will not constrain the user to be logged in the application, meaning even visitors will have access to it. The option will only be available in the Explore windows. The information must be exported either as a CSV file or XML document.

Contact us

The module will display a map with the address of the botanical garden and a simple form, which can be accessed by non-registered users as well, and it will have the option of sharing ideas, messages to an email address.

Support us

The mentioned module will offer two functions, as follows.

First is the possibility of supporting by donating money. The user can do that viawww.patreon.com

Second, it will allow for volunteering activities. To access this function, the user must be logged in.

Import module

This feature will only be available for a certain class of users: data-entry and system administrator. The person will be able to import data into the database in bulk without it having any technical major skills, this should be done through a user interface which will focus on functionality and not design. The import will be done by using CSV files XML documents. There will be two import options available, creating a new item and updating an existing one. When creating an item, there will be a minimum number of fields required for the task to be completed successfully. The module will also display any running tasks and its current status.

History

Only available to registered users, it will display all the documents that have been exported by them in the past, and also provide the option of searching within a given date range.

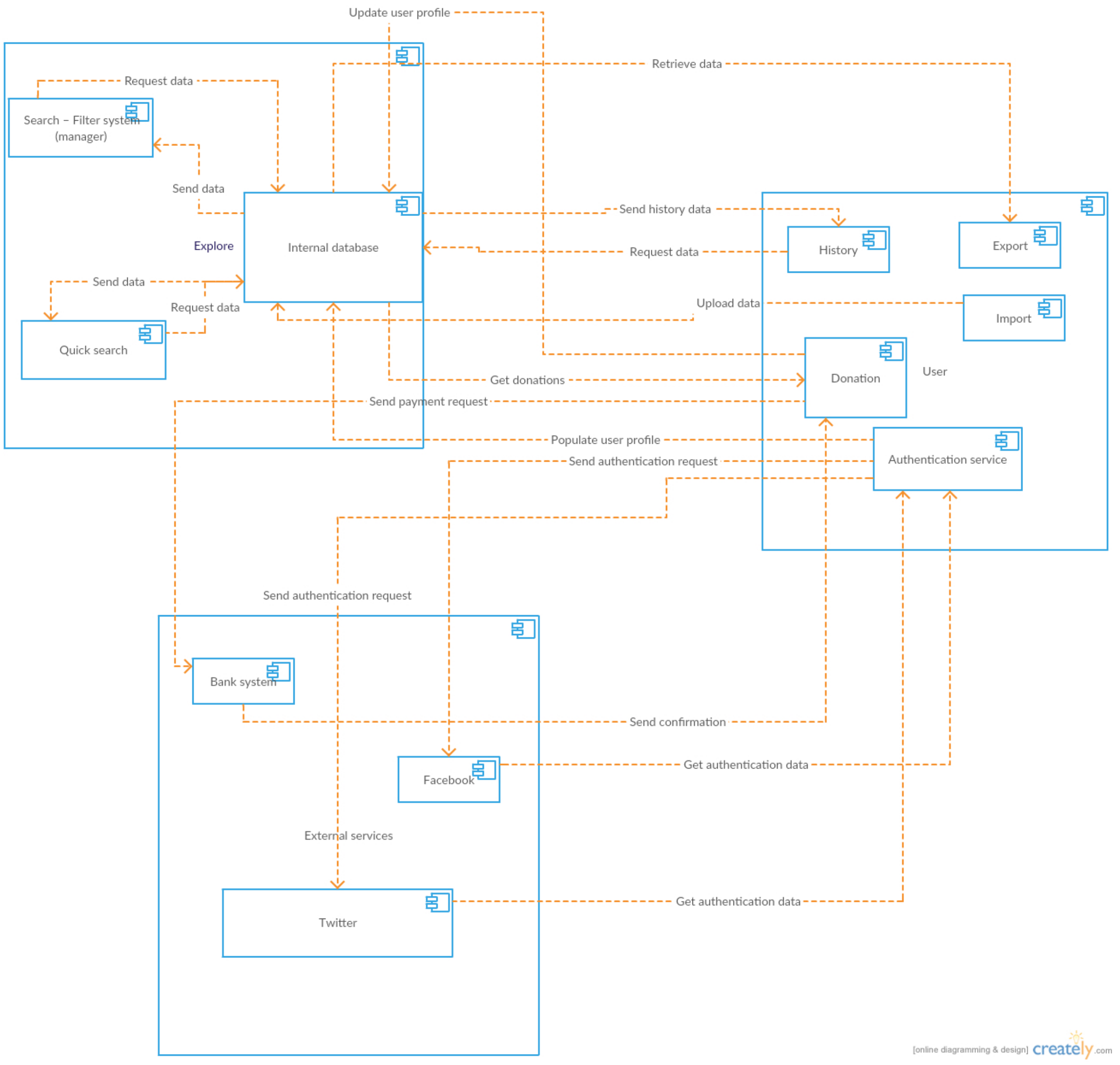
It will also display a feature, only available for data-entry users and above, that will allow to look up on the exports/imports done by other users.

Data structures

A data structure represents a specialized format for storing and organizing data. The general data structure types include arrays, files, records, tables, threes and so on. Data structures are designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In a more general sense, the concept of data structure dovetails with that of virtual objects and virtual reality. As it is a more elaborately arranged by developers and others, data becomes more functional allowing the emergent of virtual reality and becoming a core concept of many technological advances.

To better understand the flow of data and how the modules interact with each other we have designed the component diagram, which shows the flow of data and information between the modules of the application.

Component diagram



Component diagram

The application is divided into three separate modules that communicate with each other to achieve our end goal, a fully functional application based on external services. The three modules are User, Explore and External Services.

The User module contains functionality such as authentication, history, export and import all of them being an important part of the further improvement, development and success of the website. By implementing user module in a manner easily understood by the end-user, we stimulate the user to visit the site again, therefore increasing the rate of return and our income.

The Explore module contains the core data needed for the website to work. It contains quick search, search-filter system (manager) and internal database. These functionalities further increase the likelihood of the end-user returning by offering an easy way of obtaining the information that they require.

The External Services module contains functionality implemented with the help of APIs. This module includes a

Facebook API, Twitter API, and a payment system. These services offer authentication functionality and the possibility for the end-users to donate.

Submodules

A submodule is as the name suggests a composing part of the module. It helps create the full functionality of the module by implementing specific data operations, which in the end produce a result for the end-user. They are the key element of any application as they can be added to a module if the requirements of the application changed and some of them can be removed without changing the overall functionality of the application.

The first functionality that the user encounters when logging into a site is usually authentication. For our application this is submodule is part of the User module and as the name suggests implements the how the user will authenticate on our website. This is done with the help of other submodules such as Facebook API and Twitter API and introducing a new registration in our internal database.

Another important functionality is represented by the Import submodule, which is also part of the User module. With the help of this submodule, the authenticated user is able to introduce new data into the system and storing it in another submodule from the Explore module, which is Internal database.

The export submodule offers the possibilities for either registered or not registered user to export data from the Internal database.

This implementation is helped by two other submodules of the Explore module, which are Quick search and Search-Filter system, which help the user find the information necessary or search if the information is already available on the website or if they should add it themselves.

Last but not least the Donation submodule which implements a functionality of donating for the further development and improvement of the application.

External data sources and use of external services (APIs)

In simple terms, APIs or Application Programming Interfaces are sets of requirements that govern how an application can talk to another. They are what make it possible to move information between programs. On the web, APIs make it possible for big services like Facebook for example to let other applications take advantage of what they can offer.

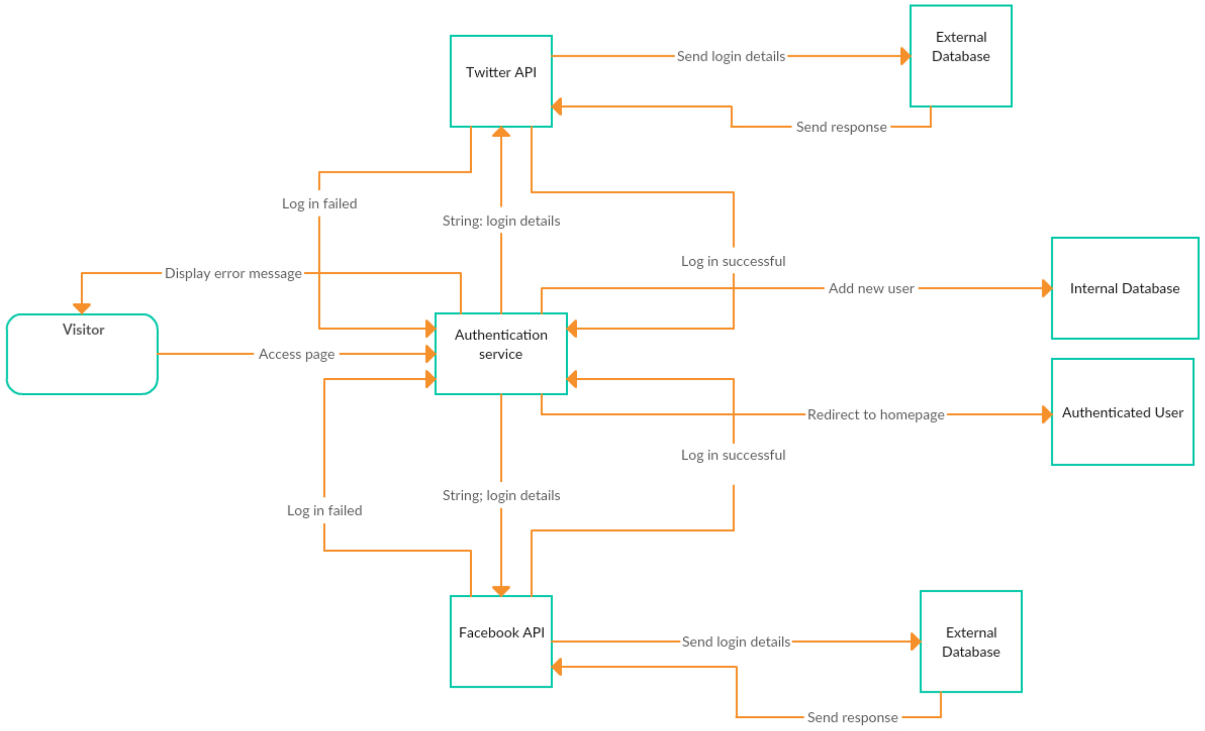
They do this by exposing some of a programs internal functions to the outside world in a limited fashion that does not damage the overall security of the designed software. By doing that it makes it possible for applications to share data between them and take actions on one and anothers behalf without requiring developers to share the code.

APIs are especially important because they dictate how developers can create new apps that implement functionalities of big Web services or social networks, reducing time needed to develop and implement it from scratch.

In one sense, then, APIs are great time savers and offer user convenience in many cases (such as ours) by offering the possibility of signing into many applications and web sites by using their Facebook id.

For our application, we implemented two APIs, which provide functionality to our authentication submodule. Our authentication services communicates with the API which in return communicates with their database. If the response is negative than authentication service displays an error message. If the response is affirmative than the visitor is added to the Internal Database and redirected to the homepage as an authenticated user.

Authentication Data Flow



Authentication Data Flow

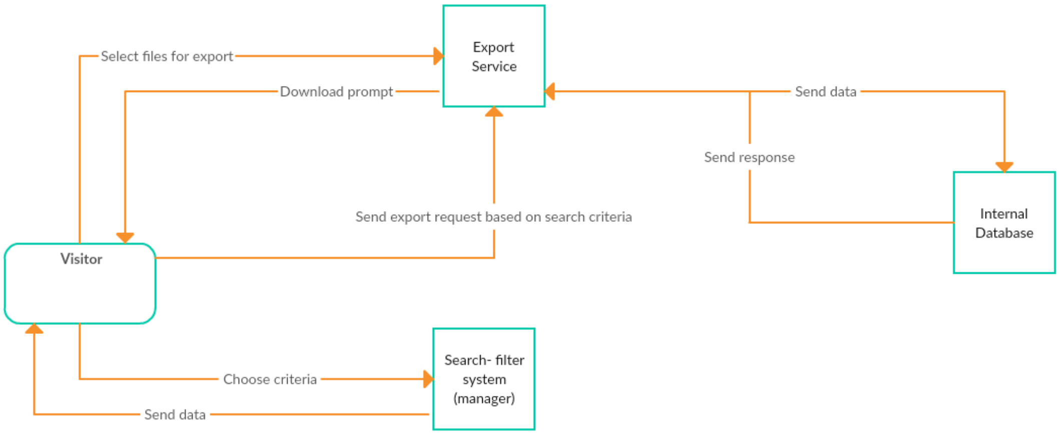
Data flows

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

In our case we have two additional data flows except for the one mentioned previously.The export data flow which represents the flow of data needed for the end-user to be able to export a file containing the flowers that he needs. The visitor chooses the criteria for what plants he would like to see in the advance search and filter system then he is given the data he requested. Based on the informationhe gets he then sends an export request to the export service. The export service sends the data to the internal database and receives a response. After this the user/visitor is prompted to download the file and the flow is complete.

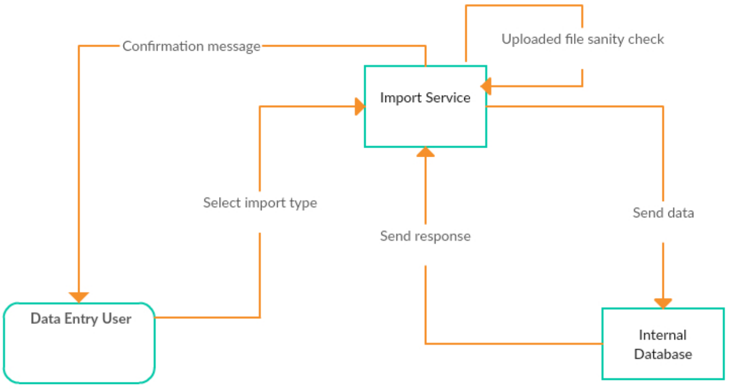
Export Data Flow



Export Data Flow

The import data flow works in a similar way. The data entry user selects the import type CSV or XML, the import services checks that the file is supported by the sistem and sends the data to the internal database. The internal database sends the response and after that the import service sends the confirmation message to the user.

Import Data Flow



Import Data Flow

References

http://readwrite.com/2013/09/19/api-defined/

https://en.wikipedia.org/wiki/Data_flow_diagram

http://18f.github.io/API-All-the-X/pages/benefits_of_apis