**SEAL**

**Abstract**

Software architecture of the SEAL Web Application that involve the structure, modeling, usage and technologies.

**Keywords**

Web Application, User, API, JavaScript, Architecture

**Introduction**

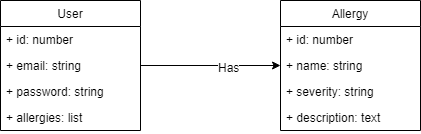
SEAL, Seasonal Allergy Web Notifier, is an application meant to notify users about possible allergy alerts according to personal profile. Also, the human user can both visualize articles about latest allergy alerts in newsfeed and view statistics about reported data.

**Technologies**

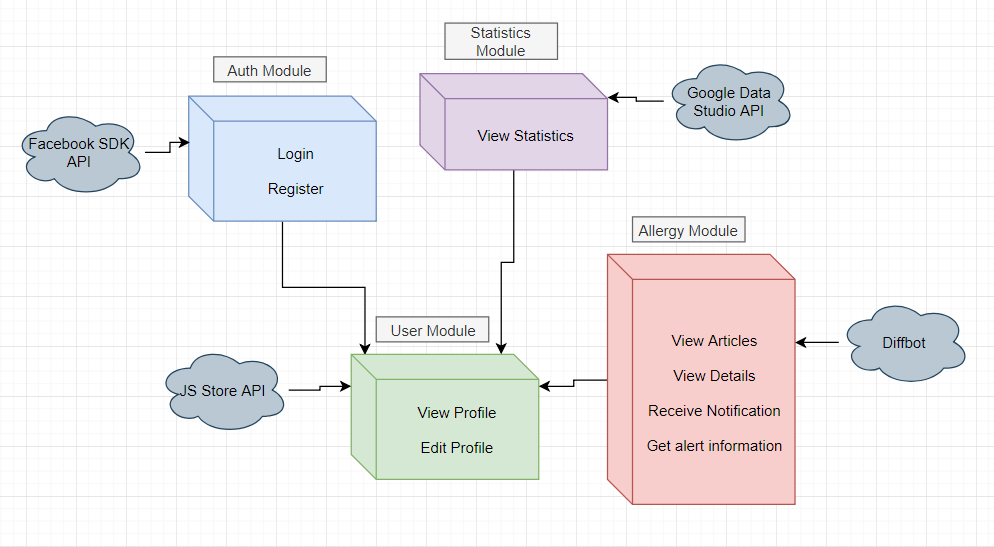
* JavaScript : Often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside with HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. We choose JS for the dynamic view of our concepts and data representation usage. Also, its DOM (Document Object Model) structure makes both the manipulation of HTML structure and stylization easier.

**Diagrams**

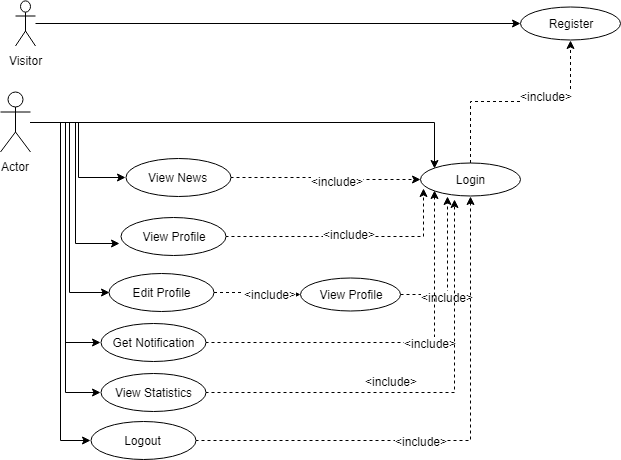
* **Architectural Diagram**
* **Class Diagram**



* **Module Diagram**



* Auth Module: This module manages the Login and Register features. We will use the Facebook SDK with JavaScript API to implement the Login with Facebook functionality. We will need the user list of social friends to send them all notifications about possible allergies.
* Statistics Module: Handles the representation of some information about the allergy flows. We will use the Google Data Studio API to implement the representation of data.
* Allergy Module: This module handles everything about allergy including viewing articles, viewing more details about them, receiving notification with certain frequency and importance and getting information about allergy alert from a specific platform with a web crawler. We will use Diffbot API to look at the microdata of the health platform and extract what we need for an allergy notifier.
* User Module links all the rest module because the user is the one that activates each module. This module will manage the View Vrofile and Edit Profile information and we will manage the data base with JS Store API.
* **Use-Case Diagram**

****

**Use Cases**

* Authentication:  The visitor can create an account and register on SEAL. Filling out the fields, accessing the REGISTER button.
* Login: The user will firstly register and then login to his personal account after filling the fields and accessing the LOGIN button.
* View Profile: After Login, the user will access the Profile Page where he can view his profile information.
* Edit Profile: After Login, the user will access the Profile Page where he can edit his profile information by accessing the EDIT button.
* View allergies: Login, access the News Page and view articles.

* View allergy detail: Login, access the News Page and view articles, accessing the SHOWMORE button.

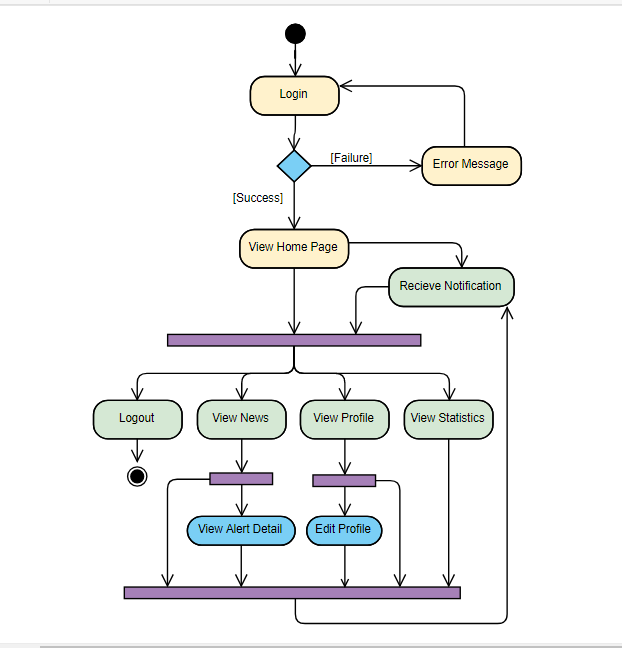
* Receive notification: Login, receive notification dynamic according both to user’s personal profile and allergy alerts dynamically loaded in a suggestive pop-up.
* **Activity Diagram**

Activity diagram is another important diagram in UML to describe the dynamic aspects of a system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another.

Here we captured the dynamic behavior of the application. The activity starts with Login and ends with Logout. The main characteristic of the diagram is the fact that the user can receive a notification at any point in his activity flow.

****