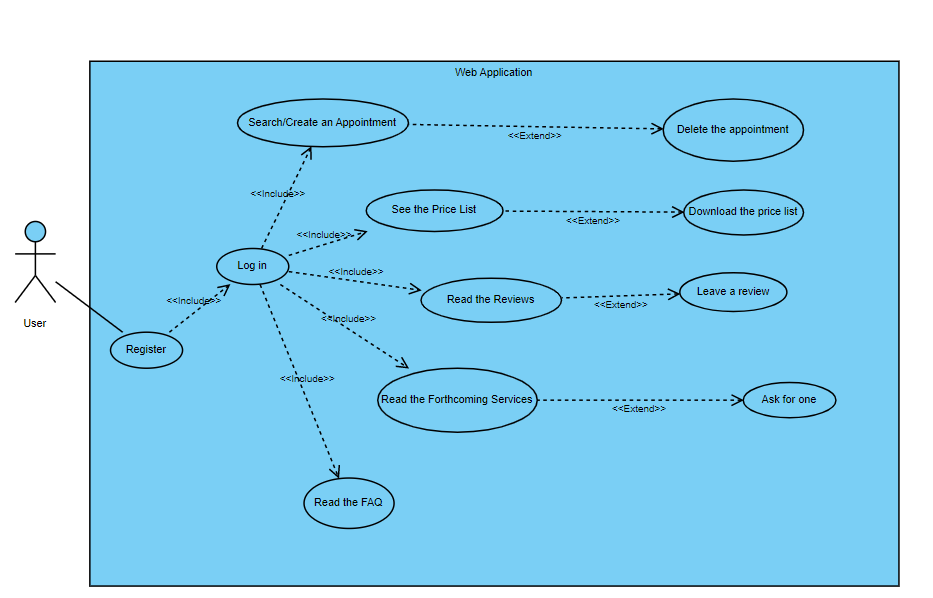
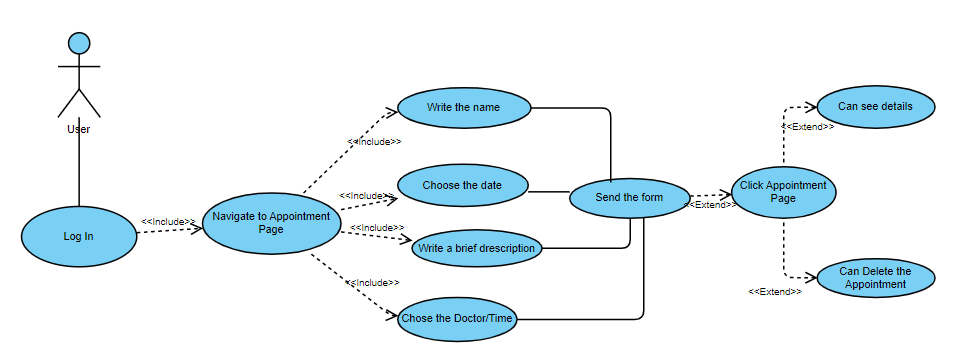
**3. The breakdown of the application**

When this concept of designing and creating a web application designated for medical management showed up in my brain, I knew that it ought to be unquestionably patient-centered. That being so, as far as I am concerned, a use case diagram would be the most convenient way to provide details.



This diagram cares for an overview of the application and what the user can do after the account is created.

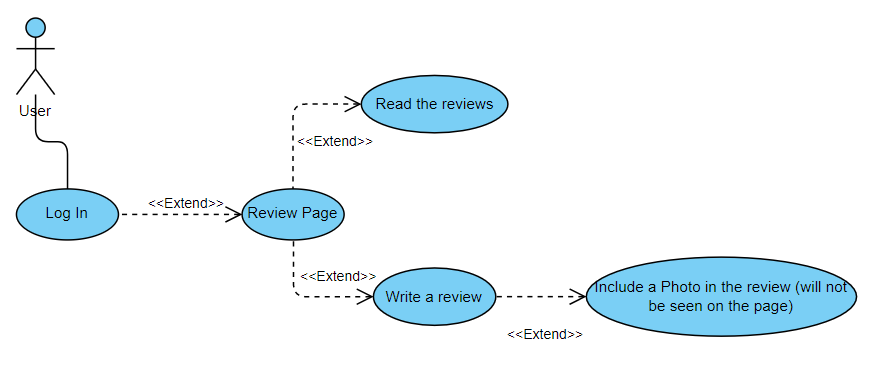


The illustration sketches the process of creating an appointment and the details, the name, the wanted date, a brief description of the problem, and finally, yet importantly the doctor, details that are imperative for the form as a means for the appointment to be created and sent into the local database via the API. Nonetheless, if the user wants so, there is always the ability to see the created appointment and eventually delete it.

Description of the “Appointment” Case

|  |  |
| --- | --- |
| Element of the Use Case | Description |
| Code | Every user will register with the email |
| State | Draft |
| Purpose | Create an appointment |
| Name | Appointment Process |
| Main Actor | User |
| Description | A user wants to appoint a medical consult with a doctor selected upon hers/his specialty |
| Preconditions | The user must have an account and complete the required fields in the form |
| Postconditions | N/A |
| Trigger | Click on Examination in the Menu |
| Basic Flows | The user ought to log in to the site using the credentials, being assumed that the account is already created. Then, the necessary fields displayed on the page need to be completed before the send button is submitted. |
| Alternate Flows | N/A |
| Relationships | <<Include>>/ <<Exclude>> Relationships |
| Frequency of use | It is being used for different situations: control needed/ emergency |
| Business Rules | The user wants to fill out the form |

In another train of thought, based upon the fact that in the present circumstances, reviews matter when a choice is being made, from my personal standpoint, it seemed like a section dedicated to this would be rather appropriate. Nonetheless, the next figure will highlight this slice.



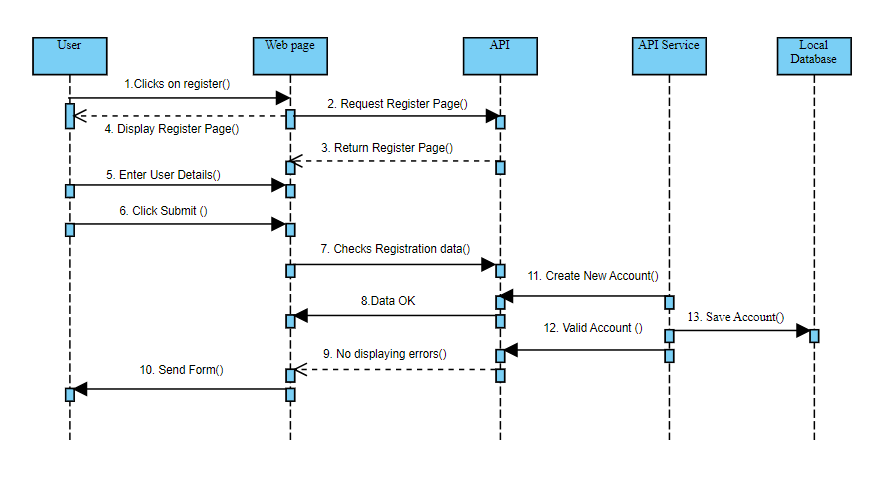
Narration barely and revealing some information, this review section is to a degree rather distinctive, as well as a field for a photo to engage in. In a personal manner, I established that a photo with a doctor or of the clinic would be inclined toward being at least appealing.

Description of the “Review” Case

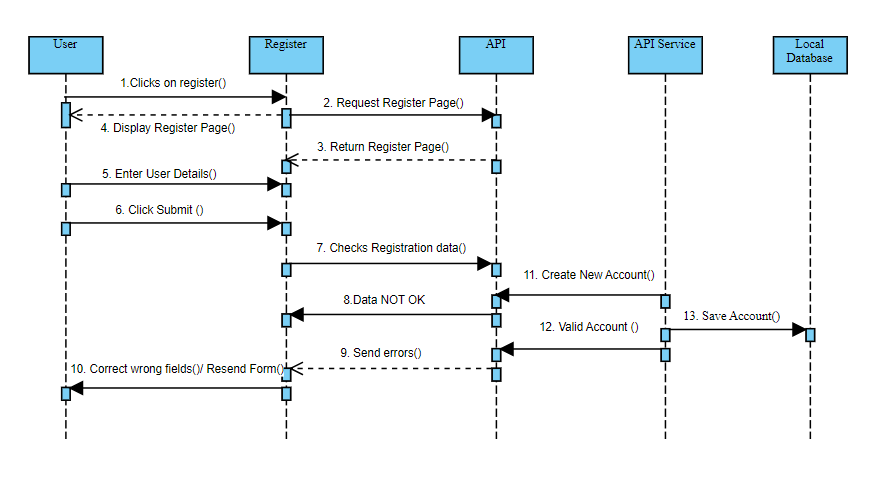
|  |  |
| --- | --- |
| Element of the Use Case | Description |
| Code | Every user will register with the email |
| State | Draft |
| Purpose | Leave a Review on the site |
| Name | Leave a review |
| Main Actor | User |
| Description | A user wants to read the reviews about the clinic/doctors etc. There is always the option to leave a review |
| Preconditions | The user must be logged in to his account |
| Postconditions | N/A |
| Trigger | Click on Reviews in the Menu |
| Basic Flows | If it is wanted, the user can read the reviews from other people; He/she is also empowered at any time to write his/her beliefs and have the possibility to upload a photo with the doctor/clinic, etc. that will not appear in the Review part of the page |
| Alternate Flows | N/A |
| Relationships | <<Exclude>> Relationships |
| Frequency of use | It is being used with a singular purpose: read/rate the clinic |
| Business Rules | The user wants to participate in the overall rating |

Offering a more comprehensive analysis, the very first paramount thing to mention is that my application does not deliver a great number of features when an individual is not registered or logged in to the site, as the majority of the site’s capabilities are interconnected to one’s personal account. This thinking pursues to some extent the concepts in this area of interest, as a virtual profile is essential (Regina Maria, MedLife…).

However, in order to complete the sign-up/log-in process, it is necessary to carefully go through the registration form and ensure that all security requirements are met, which cannot be avoided. There are two feasible outcomes that might occur at the time the send button is clicked: the first states that the registration form security requirements are met, and, evidently, the second scenario occurs when they are not fulfilled. Therefore, when registering, if the prerequisites are met, the piece of information containing the email and password is sent via the API into the local database. Otherwise, the registration form does not get submitted again unless it is modified.



As previously discussed regarding the two scenarios, the figure above considers the first one and presents the stream of the procedure. Nevertheless, the user clicks on the register button from the menu, and while the duty is done in the background in nearly no consequential time, the page is loaded. Next, the form is filled out with the right information, and after the click, it gets sent to the local database in order for it to be stored. At this point, the user can now discover the site of their own choice.



This succeeding scheme is responsible for the circumstance in which the user fails to measure up to the expectations for the account to registered. It is discernible that there is a slight change in the flow of the events, in the end with the same aftermath.