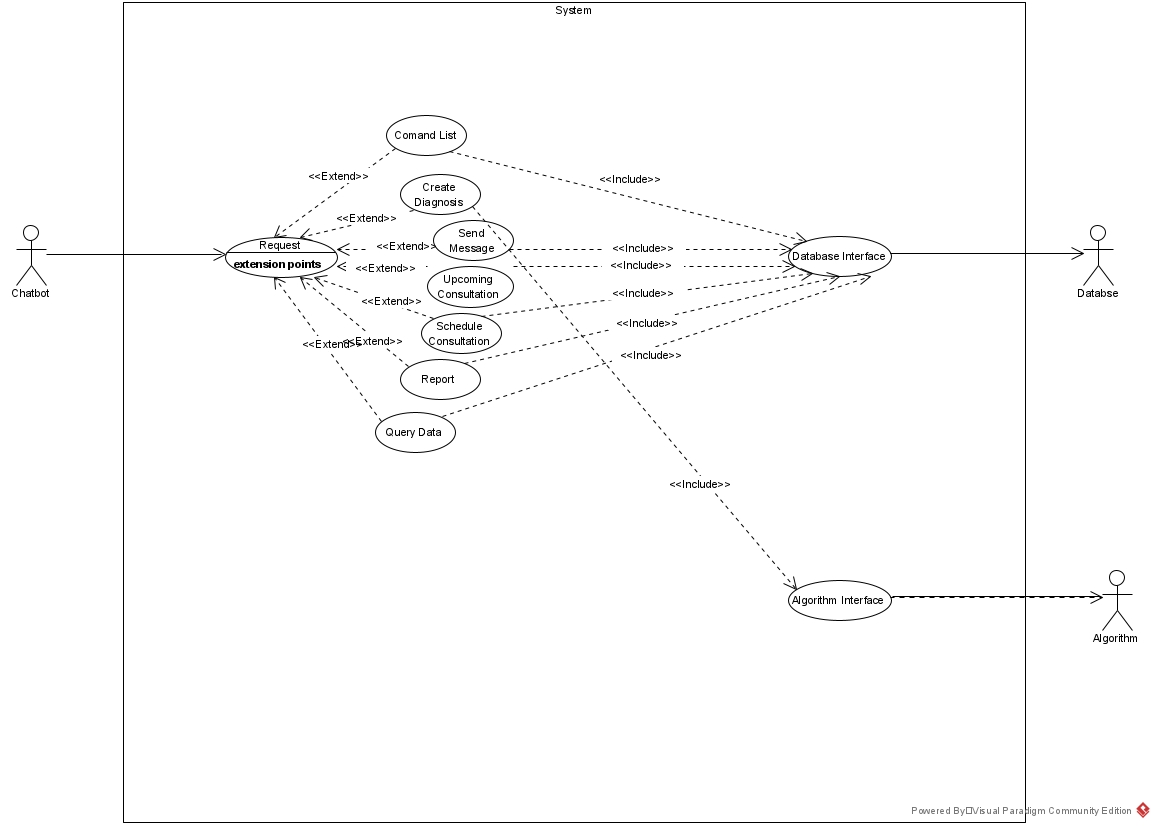
## Use Cases of Server



## Web Client Update Data

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Web Client Update Users Data | | |
| Version | V1 | Created (date): | *12/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client | | |
| Goals | Update data in database about users (doctors, patients), like patients’ diagnosis, patient’s personal data, doctor’s personal data and doctor’s schedules. | | |
| Summary | How Web Client has session logged in, he can raise a request to update data. After request, the server contacts database to apply the modifications, after this the update are applied. Web clients can be doctors, patients, and a system administrator.  Patient have permissions to modify his personal data and his own diagnosis.  Doctor have permissions to modify his personal data, modify the diagnosis of your patients and his schedule.  Administrators have permissions to manage patients and doctors accounts. | | |
| Actors | Web Client (Doctor) (Patient) (Administrator) | | |
| Trigger | Client does API Request to update data | | |
| Precondition | Server is running, Web Client is logged in, have permissions to do the updates, and already has data registered in database. | | |
| Frequency | Rarely | | |
| Postconditions | Data Updated | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Send a request to server |  |
|  |  | Apply the modifications (Move to alternate flow 1 when this flow failed) |

|  |  |  |
| --- | --- | --- |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |

## Web Client Queries Data

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Client Queries Data | | |
| Version | V1 | Created (date): | *12/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client | | |
| Goals | Web Client queries data in database, like patient diagnosis, personal data, and doctor´s schedule and personal data. | | |
| Summary | How Web Client has session logged in, he can raise a request to query data. After request to query what he wants, the system contacts the database to get the data that Web Client has requested.  If user is a Doctor, this one can queries your own data, like personal information, his schedule, and data from your patients, if the users are a Patient, he can query your own data and some data from your doctor. The admin can query data from everyone. | | |
| Actors | Web Client (Doctor) (Patient) (Admin) | | |
| Trigger | Web Client raises API Request to queries data | | |
| Precondition | Server is running, Web Client is logged in, and have permissions to queries that data. | | |
| Frequency | Frequently | | |
| Postconditions | User queries are answered. | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise a query request. |  |
|  |  | Answer queries. (Move to alternate flow 1 when failed.) |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell Web Client that he can´t answer the queries. End |

## Web Client Manage User Information

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Web Client Manage User Information | | |
| Version | V1 | Created (date): | *23/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client (Administrator) | | |
| Goals | Manage all user information. Add new accounts in system and delete accounts. | | |
| Summary | How Web Client has session logged in, he can raise a request to manage user information. After request, the server contacts database to apply the modifications, after this the modifications are applied. Web clients can be administrators.  He has permissions to add new accounts in system, delete accounts, edit accounts and search account in database. | | |
| Actors | Web Client (Administrator) | | |
| Trigger | Client does API Request to update data | | |
| Precondition | Server is running, Web Client is logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Modifications or are applied | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Send a request to server |  |
|  |  | Apply the modifications (Move to alternate flow 1 when this flow failed) |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |

## Web Client Manage Treatment Record

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Web Client Manage Treatment Record | | |
| Version | V1 | Created (date): | *23/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client (Administrator) | | |
| Goals | Manage all treatment records. Add new records in system, delete records and search records | | |
| Summary | How Web Client has session logged in, he can raise a request to manage treatment record. After request, the server contacts database to apply the modifications, after this the modifications are applied. Web clients can be administrators.  He has permissions to add new records in system, delete accounts, and search records in database. | | |
| Actors | Web Client (Administrator) | | |
| Trigger | Client does API Request to manage treatment record | | |
| Precondition | Server is running, Web Client is logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Data Updated | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Send a request to server |  |
|  |  | Search for corresponding record in system (Move to alternate flow 1 when this flow failed) |
|  |  | Apply the modifications (Move to alternate flow 2 when this flow failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the record was not found. End. |
|  |  | Tell client that the request is illegal. End. |

## Web Client Create Diagnosis

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Web Client Create Diagnosis | | |
| Version | V1 | Created (date): | *12/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client (Patient) | | |
| Goals | Web Client create a patient diagnosis. | | |
| Summary | How Web Client has session logged in, he can raise API Request to create a diagnosis. After request, the server contacts algorithm to create a new diagnosis, then algorithm send to database the new diagnosis to be saved in patient record. The diagnosis can be made by patients, the doctor can make the diagnosis by adding the patient's x-ray to the algorithm, and patient can do that too. | | |
| Actors | Web Client | | |
| Trigger | Web Client raises an API request. | | |
| Precondition | Server is running and web client is logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Diagnosis of one client have been made | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Checks if the client's x-ray is valid. (Move to alternative flow 1 when this flow failed) |
|  |  | Algorithm send the new diagnosis to database. (Move to alternative flow 2 when this flow failed) |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that his X-Ray is invalid. End. |
|  |  | Tell client that he can´t save the diagnosis in patient medical record. End. |

## Algorithm Management

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Algorithm Management | | |
| Version | V1 | Created (date): | *12/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client (Administrator) | | |
| Goals | Web Client can manage, make changes in the algorithm, or even change algorithm for another algorithm. | | |
| Summary | Web Client (Administrator) raise API Request to manage the algorithm. After request, the server contacts algorithm interface to apply changes in algorithm, or even change the algorithm for another. | | |
| Actors | Web Client | | |
| Trigger | Web Client raises an API request. | | |
| Precondition | Server is running and web client is logged in. | | |
| Frequency | Rarely | | |
| Postconditions | Changes were made in the algorithm | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  |  |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  |  |
|  |  |  |
|  |  |  |

## Web Client Send Message

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Send Message | | |
| Version | V1 | Created (date): | *12/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web Client | | |
| Goals | Web Client send a message to another client, can be one doctor to send a patient or reverse. | | |
| Summary | How Web Client has session logged in, he can raise a request to send a message to another user, after this the server get in contact with database to get the other user´s contact, then after receiving the contact, message is sent to other user via WeChat. | | |
| Actors | Web Client | | |
| Trigger | Web Client raises an API request. | | |
| Precondition | Server is running and web client is logged in. | | |
| Frequency | Rarely | | |
| Postconditions | Message is sent to destiny | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  | Check if found client contact in database. (Move to alternate flow 2 when failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  | Tell Web Client than can´t find the client contact. End |
|  |  |  |
|  |  |  |

## Web User Sign Up

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Web User Sign Up | | |
| Version | V1 | Created (date): | *22/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web User | | |
| Goals | Web User create new account in system, he can be a patient, doctor, or administrator. | | |
| Summary | Web User raise a request to sign up, after this the server get in contact with database to create new account in system. | | |
| Actors | Web Client | | |
| Trigger | Web Client raises an API request. | | |
| Precondition | Server is running and web client isn´t logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Web client had access to his account | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  | Check data of new account. (Move to alternate flow 2 when failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  | Tell Web Client than can´t create new account. End |
|  |  |  |
|  |  |  |

## Web User Log In

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Web User Log In | | |
| Version | V1 | Created (date): | *22/04/2021* |
| Author | Pedro Nunes | | |
| Source | Web User | | |
| Goals | Login of the web user in system, to have access to the system, its functionalities, and data. | | |
| Summary | Web User raise a request to Log In, after this the server get in contact with database to check user credentials to log in your account. | | |
| Actors | Web Client | | |
| Trigger | Web Client raises an API request. | | |
| Precondition | Server is running and web client isn´t logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Users enter in system | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  | Check Web user credentials. (Move to alternate flow 2 when failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  | Tell Web Client than his credentials wasn´t found in system. End |
|  |  |  |
|  |  |  |

## Android Client Queries Data

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Android Client Queries Data | | |
| Version | V1 | Created (date): | *19/04/2021* |
| Author | Pedro Nunes | | |
| Source | Android Client | | |
| Goals | Web Client queries data in database, like patient record (diagnosis and personal data), and doctor´s personal data. | | |
| Summary | How Android Client has session logged in, he can raise a request to query data. After request to query what he wants, the system contacts the database to get the data that Android Client has requested. If user is a Doctor, this one can queries your own data, like personal information, and medical record from your patients, if the users are a Patient, he can query your own personal data, his medical record, and some data from your doctor. | | |
| Actors | Android Client (Doctor) (Patient) (Administrator) | | |
| Trigger | Android Client raises an App Request. | | |
| Precondition | Server is running and Android Client is logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Android Client queries are answered. | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an App request |  |
|  |  | Answer queries. (Move to alternate flow 2 when failed.) |
|  |  |  |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  |  |
|  |  |  |
|  |  |  |

## Android Client Update Data

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Android Client Update Data | | |
| Version | V1 | Created (date): | *19/04/2021* |
| Author | Pedro Nunes | | |
| Source | Android Client | | |
| Goals | Update data in database about android client (doctors or patients), like patients’ medical record, patient’s personal data, and doctor’s personal data. | | |
| Summary | How Android Client has session logged in, he can raise a request to update data. After request, the server contacts database to apply the modifications, after this the modifications are applied. Android Clients can be doctors, patients:  Patient and Doctor have permissions to update his personal data. | | |
| Actors | Android Client (Doctor) (Patient) | | |
| Trigger | Android Client does App Request to update data | | |
| Precondition | Server is running, Web Client is logged in, and already has data registered in database. | | |
| Frequency | Frequently | | |
| Postconditions | Data Updated | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Send a request to server |  |
|  |  | Apply the modifications (Move to alternate flow 1 when this flow failed) |
|  |  |  |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  |  |
|  |  |  |
|  |  |  |

## App Client Create Diagnosis

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | App Client Create Diagnosis | | |
| Version | V1 | Created (date): | *19/04/2021* |
| Author | Pedro Nunes | | |
| Source | App Client | | |
| Goals | App Client creates a diagnosis. | | |
| Summary | How Android Client has session logged in, he can raise a request to create a diagnosis. After request, the server contacts algorithm to create a new diagnosis, then algorithm send to database the new diagnosis to be saved in patient record.  The diagnosis can be made by patients adding his x-ray to the algorithm. | | |
| Actors | App Client (Patient) | | |
| Trigger | App Client raises an App Request. | | |
| Precondition | Server is running and app client is logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Diagnosis have been made | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
| 1. | Send a request to server |  |
| 2. |  | Checks if the client's x-ray is valid. (Move to alternative flow 1 when this flow failed) |
| 3. |  | Algorithm send the new diagnosis to database. (Move to alternative flow 2 when this flow failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
| 1. |  | Tell client that the request is illegal. End. |
| 2. |  | Tell client that he can´t save the diagnosis in patient medical record. End. |
|  |  |  |
|  |  |  |

## App Client Send Message

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Send Message | | |
| Version | V1 | Created (date): | *19/04/2021* |
| Author | Pedro Nunes | | |
| Source | App Client | | |
| Goals | Web Client send a message to another client, can be one doctor to send a patient or reverse. | | |
| Summary | How Android Client has session logged in, he can raise a request to send a message to another user, after this the server get in contact with database to get the other user´s contact, then after receiving the contact, message is sent to other user via WeChat. | | |
| Actors | App Client | | |
| Trigger | App Client raises an App Request. | | |
| Precondition | Server is running and web client is logged in. | | |
| Frequency | Frequently | | |
| Postconditions | Message is sent to destiny | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  | Check if found client contact in database. (Move to alternate flow 2 when failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  | Tell Web Client than can´t find the client contact. End |
|  |  |  |
|  |  |  |

## Diagnosis Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Diagnosis Comparison | | |
| Version | V1 | Created (date): | *19/04/2021* |
| Author | Pedro Nunes | | |
| Source | App Client | | |
| Goals | App Client receive the comparison between two different diagnosis in a graph | | |
| Summary | How Android Client has session logged in, he can raise a request to get a comparison between two diagnosis. Then the server contacts the databases to get the diagnosis, to do the comparison. | | |
| Actors | App Client | | |
| Trigger | App Client raises an App Request. | | |
| Precondition | Server is running and web client is logged in. | | |
| Frequency | Rarely | | |
| Postconditions | App Client get comparison between two diagnosis | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request |  |
|  |  | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  | Check if found the diagnosis in database. (Move to alternate flow 2 when failed) |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  | Tell App Client than can´t find the diagnosis in database. End |
|  |  |  |
|  |  |  |

## App Client Consult Readings/Videos

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | App Client Consult Readings/Videos | | |
| Version | V1 | Created (date): | *19/04/2021* |
| Author | Pedro Nunes | | |
| Source | App Client | | |
| Goals | App Client consult corresponding readings and videos about scoliosis. | | |
| Summary | How Android Client has session logged in, he can raise a request to get access to readings and videos about scoliosis, server contact the databases to get corresponding information. Readings and videos, can be accessed by patients and doctors. | | |
| Actors | App Client | | |
| Trigger | App Client raises an App Request. | | |
| Precondition | Server is running and web client is logged in. | | |
| Frequency | Rarely | | |
| Postconditions | The corresponding videos and readings are presented to App Client | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise an API request | Check request is valid or not. (Move to alternate flow 1 when failed.) |
|  |  | Check if App Client had medical record, to correspond information to client. (Move to alternate flow 2 when failed) |
|  |  |  |
|  |  |  |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell client that the request is illegal. End. |
|  |  | Tell App Client that he doesn´t have medical record, so he doesn´t have information about scoliosis to get. End |
|  |  |  |
|  |  |  |

## Chatbot Client View Command List

## Chatbot Client Create Diagnosis

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Chatbot Client Create Diagnosis | | |
| Version | V1 | Created (date): | *24/04/2021* |
| Author | Pedro Nunes | | |
| Source | Chatbot Client (Patient) | | |
| Goals | Chatbot Client create a diagnosis. | | |
| Summary | Chatbot Client raise request to create a diagnosis, uploading an x-ray. After request, the server contacts algorithm to create a new diagnosis, then algorithm send to database the new diagnosis to be saved in patient record. The diagnosis can be made by patients. | | |
| Actors | Chatbot Client | | |
| Trigger | Chatbot Client raises a request to upload the x-ray image of his spine. | | |
| Precondition | Server is running and chatbot client send a message for chatbot. | | |
| Frequency | Frequently | | |
| Postconditions | Diagnosis of one client have been made | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
| 1. | Raise an API request |  |
| 2. |  | Checks if the client's x-ray is valid. (Move to alternative flow 1 when this flow failed) |
| 3. |  | Algorithm send the new diagnosis to database. (Move to alternative flow 2 when this flow failed) |
| **Alternative Flow** | *Actor* | *System* |
| 1. |  | Tell client that his X-Ray is invalid. End. |
| 2. |  | Tell the client that he can´t save his diagnosis in database. End. |

## Chatbot Client Send Message

## Chatbot Client Schedule Consultation

## Chatbot Client Report Problem

## Chatbot Client Query user’s information

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Chatbot client Queries user’s Information | | |
| Version | V1 | Created (date): | *12/04/2021* |
| Author | Pedro Nunes | | |
| Source | Chatbot Client | | |
| Goals | Chatbot Client queries data in database, like patient diagnosis and personal data, and doctor information’s, such his schedule, his personal data and his list of patients. | | |
| Summary | Chatbot Client raise request to search information’s in database. After request to query what he wants, the server contacts the database to get the data that chatbot Client has requested.  If chatbot client was a patient, he can search his own data, like his diagnosis, his upcoming consultation, and some data of his doctor.  If he is a doctor, he can search his own data, such his list of patients and your diagnosis, his next schedule and patients messages. | | |
| Actors | Chatbot Client (Doctor) (Patient) (Admin) | | |
| Trigger | Chatbot Client raises a Request to queries data | | |
| Precondition | Server is running and chatbot client send a message for chatbot. | | |
| Frequency | Frequently | | |
| Postconditions | User queries are answered. | | |
| Diagram |  | | |

|  |  |  |
| --- | --- | --- |
| **Basic Flow** | *Actor* | *System* |
|  | Raise a query request. |  |
|  |  | Answer queries. (Move to alternate flow 1 when failed.) |
| **Alternative Flow** | *Actor* | *System* |
|  |  | Tell Web Client that he can´t answer the queries. End |