		Name: Register patient **Preconditions: Patient should open the app** **Postconditions: Patient appears in the database** **Actors: Patient**
Main Scenario	**Step**	**Action**
	1	User is asked for his personal information
	2	User types his personal information
	3	Information is verified in order to be successfully registered
	4	User is registered into the database with its correspondent information
Extensions	**Step**	**Branching Action**
	4_1	A registered doctor will be assigned randomly to the patient, by searching a random id of a doctor already registered
	4_2	When a patient registers, his id document is automatically set as his username
	4_3	An id is automatically asigned to the patient
		Name: Register doctor Preconditions: Doctor should open the app Postconditions: Doctor appears in the database Actors: Doctor
Main Scenario	**Step**	**Action**
	1	User is asked for his personal information
	2	User types his personal information
	3	Information is verified in order to be successfully registered
	4	User is registered as a doctor into the database with its correspondent information
	Step	**Branching Action**
	4_1	An id is automatically assigned to the doctor
	4_2	When a doctor registers, his id document is automatically set as his username
		Name: Log in Preconditions: User is registered in the app with an username and a password Postconditions: User gets a menu with options Actor: Patient, doctor or director
Main Scenario	**Step**	**Action**
	1	System asks for username and password
	2	User types username and password
	3	System validates user's username and password in order to be successfully logged in.
	4	System displays the menu of options.

Extensions	**Step**	**Branching action**
	4_1	System detects if the user is a doctor, a director or a patient, and it leads to a different menu depending on what it is
		Name: Log Out Preconditions: User should have logged in before Postconditions: App closes Actor: Doctor, patient, director
Main Scenario	Step	Action
	1	Select option log out on the menu
		Name: Add vaccine Preconditions: the vaccine that is going to be added must not exist in the database Postconditions: vaccine added Actor: director
Main Scenario	**Step**	**Action**
	1	Director manager is asked fot the vaccine information
	2	Director manager types the vaccine's name and number of doses
	3	System validates vaccine's information in order to be successfully inserted into de database.
	4	Vaccine is registered into the database with its correspondent information
Extensions	**Step**	**Branching action**
	3_1	The vaccine already exists in the database, it cannot be added
	4_1	An id is automatically assigned to the vaccine
		Name: Add condition Preconditions: the condition that is going to be added must not exist in the database Postcondition: condition added Actor: director
Main Scenario	**Step**	**Action**
	1	Director manager is asked for the condition's name
	2	Director manager types the condition's name
	3	System validates the condition's information in order to be successfully inserted into the database
	4	Condition is registered into the database.
Extensions	**Step**	**Branching Action**
	3_1	The condition already exists in the database, it cannot be added
	4_1	An id is automatically assigned to the condition
		Name: Add disease Preconditions: the disease that is going to be added must not exist in the database Postconditions: disease added Actor: director
Main Scenario	**Step**	**Action**
	1	Director manager is asked for the name of the disease he wants to register

	2	Director types the name of the disease
	3	Disease is inserted into the database
Extensions	**Step**	**Branching Action**
	3_1	If the disease already exists in the database, it cannot be added.
	3_1	An id is automatically assigned to the disease
		Name: Remove doctor Preconditions: the doctor that is going to be remove must exist on the database Postconditions: doctor removed, all the patients assigned to that doctor are now not assigned to any doctor
Main Scenario	**Step**	**Action**
	1	Director manager is asked for the name of the doctor that wants to remove from the database
	2	Director manager types the name of the doctor that want to remove
	3	System validates that there is a doctor with that name.
	4	System gives a list of the doctor with that name and asks director to type the id of the one he wants to remove.
	5	Director types the id of the doctor he wants to remove
	6	Doctor is removed from the database.
Extensions	**Step**	**Branching Action**
	6_1	All the patients that were assigned to that doctor are now not assigned to any doctor
	4_1	System looks for all the doctor with that name in the database and returns a list
		Name: Remove patient Preconditions: the patient that is going to be remove should exist on the database Postconditions: patient removed Actor: director
Main Scenario	**Step**	**Action**
	1	Director manager is asked for the name of the patient he wants to remove
	2	Director types the name of the patient he wants to remove
	3	System validates that there is a patient registered with that name
	4	System gives a list of the patients that have that name, and asks the director to type the id of the one he wants to remove
	5	Director types the id of the patient he wants to remove
	6	Patient is removed from the database
Extensions	**Step**	**Branching Action**
	4_1	System looks for all the patients in the database with that name and returns a list.
		Name: Remove vaccine Preconditions: the vaccine that is going to be remove should exist on the database Postconditions: vaccine removed

		Actor: director
Main Scenario	**Step**	**Action**
	1	Director manager is asked for the name of the vaccine he wants to remove
	2	Director types the name of the vaccine he wants to remove
	3	Vaccine is removed from the database
		Name: Assign condition to vaccine Preconditions: vaccine and condition must exist on the database, thay can not be assigned Postconditions: condition is assigned to vaccine, if a patient has that condition he cannot put on that vaccine.
Main Scenario	**Step**	**Action**
	1	Director is asked for the name of the condition
	2	Director types the name of the condition
	3	System looks for the id of the condition with that name
	4	System asks for the name of the vaccine
	5	Director types the name of the vaccine
	6	System look for the id of the vaccine with that name
	7	Condition is assigned to vaccine in the database by their ids
		Name: Check all vaccine supplies Preconditions: Doctor has to click the option Postconditions: Doctor receives a list with all the vaccines and go back to menu Actor: doctor
Main Scenario	**Step**	**Action**
	1	Database returns a list with all the vaccines that are registered
		Name: Check vaccines of a patient Preconditions: patient should have a list of vaccines he has to put Postconditions: Doctor receives a list with the vaccines of one of his specific patients and go back to menu Actor: doctor
Main Scenario	**Step**	**Action**
	1	System asks for the name of the patient he wants to see his vaccines
	2	Doctor types the name of the patient
	3	System validates that there is a patient with that name
	4	System looks for all patients registered with that name and returns a list
	5	System asks the doctor to type the id of the patient
	6	System looks for all vaccines of the patient with that id and returns a list.
Extensions	**Step**	**Branching Action**
	3_1	If there is not a patient with that name, it asks for a valid name
		Name: Select patients Preconditinos: doctor should have patients assigned Postconditions: doctor receives a list with all his patients and go back to menu

		Actor: Doctor
Main Scenario	**Step**	**Action**
	1	System selects doctor's id and looks for all the patients That are assigned to him
		Name: Select appointments being a doctor Preconditions: Doctor should have some appointments set Postconditions: Doctor receives a list with all the appointments and go back to menu Actor: Doctor
Main Scenario	**Step**	**Action**
	1	System will be able to display a list of all the doctor's future appointments
		Name: Search all vaccines a patient has on Preconditions: Patient should have put on some vaccines Postconditions: Patients gets a list of vaccines and go back to menu Actor: patient
Main Scenario	**Step**	**Action**
	1	System detects the patient's id and all vaccination records associated with the registered patient will be fetched and shown
		Name: Search all vaccines a patient has to put Preconditions: Patient should have some vaccines still to put Postconditions: Patients gets a list of vaccines and go back to menu Actor: patient
Main Scenario	**Step**	**Action**
	1	System detects the patient's id and looks for all vaccines that he still has to put on
		Name: Check my appointments Preconditions: The patient should have set some appointments Postconditions: Patient gets a list of the appointments he has Actor: Patient
Main Scenario	**Step**	**Action**
	1	System detects the patient's id an looks for all the appointments he Has schedule
		Name: Schedule an appointmet Preconditions: The date of the appointment that is going to be set should be empty, there should be a vaccine and a doctor for that appointment Postconditions: Appointment setted Actor: Patient
Main Scenario	**Step**	**Action**
	1	System asks the patient the name of the disease
	2	Patient introduces the name of the disease
	3	System looks for the disease id by its name
	4	System selects the vaccines that are for that disease, according to the patient's medical conditions
	5	System asks for the date the patient wants to set the appointment at
	6	Patient types the date of the appointment in the specified format
	7	An appointment is registered with all this information

Extensions	Step	Branching action
	4_1	System selects the patient’s id and looks for all his conditions, and then looks for a vaccine that is compatible with his conditions, and that it is for the disease specified.
	6_1	Patient choses a day to schedule a medical appointment, that date is no longer available, it indicates him to introduce another one.
	6_2	The system traduces the format that is typed by the patient, to a format compatible with the database.
		Name: Cancel an appointment Preconditions: The appointmetn that is going to be cancelled should exist Postconditions: Appointment removed from the database Actor: Patient
Main Scenario	**Step**	**Action**
	1	System gives to the patient a list of all the appointments he has
	2	Patient is asked the id of which of the displayed appointments want to cancel
	3	Patient types the id of the appointment
	4	Appointment is removed
		Name: Update medical conditions Preconditions: Patient should click on the menu option Postconditions: It leads to another menu Actor: Patient
Main Scenario	**Step**	**Action**
	1	Patient is asked if he wants to add new conditions or delete conditions
	2	Patient inserts which option he wants to do
		Name: Add new conditions Preconditions: The patient should not have the conditions that are going to be added Postconditions:_ Condition assigned to patient Actor: Patient
Main Scenario	**Step**	**Action**
	1	Patient is asked how many conditions he wants to add
	2	Patient inserts the number of conditions that he wants to add
	3	Patient is asked the name of the conditions one by one
	4	Patient types the name of the conditions one by one
	5	All conditions that the patient has types are assigned to him
		Name: Delete condition Preconditions: The condition that is going to be deleted should be assigned to the patient Postconditions: Condition is not assigned to patient anymore Actor: Patient
Main Scenario	**Step**	**Action**
	1	Patient is asked for the name of the condition that he wants to remove
	2	Patient types the name of the condition he wants to remove
	3	System looks for the condition and removes it from the patient

FUNCTIONAL REQUIREMENTS
REQ-1: System registers
REQ-2: System allows a log in
REQ-3: System removes
REQ-4: System assigns
REQ-5: Application can be used by doctor, director and patient.
REQ-6: Doctor should access to database information
REQ-7: Doctor should be able to see patient's private information
REQ-8: Doctor should be able to see all his information
REQ-9: Patient should be able to see all his information
REQ-10: Patient is the one that controls the appointments
REQ-11: Patients should be able to modify his information
REQ-12. There should be one unique director
REQ-13. Each patinet should be assigned to one doctor
REQ-15: If a patient has a condition, he cannot put a specific vaccine
REQ-16. Each appointment should be at different dates.

NON FUNCTIONAL REQUIREMENTS
REQ-17: System automatically adds an id to a patient when this one registers.
REQ-18: System automatically adds an id to a doctor when this one registers.

REQ-19: System automatically adds an id to a vaccine when this one is registered
REQ-20. System automatically adds an id to a condition when this one is registered.
REQ-21: System automatically adds an id to a disease when this one is registered.
RE1-22: System automatically assigns a doctor that is registered in the application to a patient when the patient registers.
REQ-23: The application automatically detects by the username and password if the user is a doctor, a patient or a director and it leads to one menu or another depending on that.
REQ-24: The system traduces the format that is typed by the patient when typing the date of the appointment, to a format compatible with the database.
REQ-25: When a patient types the disease he want to put a vaccine of, the application automatically looks for a vaccine that is compatible with the disease and with the condition.
REQ-26: When a user registers, his username is automatically set up as his id document.