Use Case Name:	Open the app	**ID:**	
Actors:	Owner of the phone		
Description:	User's identification / Confirm the user identity		
Trigger:			
Preconditions:	1. Download and install the app 2. Internet has to be active		
Normal course:	1. If internet is not active, ask for an internet connexion 2. Ask the user to turn on the GPS 3. Check if there are data on the phone (E1) 4. Send a crypted message with the user's infomation 5. The server send back a hash and an ID to confirm the authentification 6. The hash and the ID are stored in the phone.		
Postconditions:	1. The user is registered in the database 2. The time of his last connexion is registered in the database 3. The request page/home page is displayed		
Exceptions:	E1: No data on the phone - Register use case E2: Data from the server doesn't match the data from the phone - The application will display 2 windows: - On the left, a window with the information from the phone - On the right, a window with the information from the server - The user choose the data he want to keep - The server or the phone will update its data and send back the host ID		

Priority:	

Use Case Name:	Register	**ID:**	
Actors:	Owner of the phone		
Description:	Gather the needed information, store it locally and send it to the server		
Trigger:	Open the app		
Preconditions:	1. Open the app		
Normal course:	1. The app will display a form which contains: - name*, surname - age*, gender (male, female, other)* - mail* - phone number (can be pick from the phone) - medical providers, hospitals (the user can provide 3 different answers) - health problems - submit button 2. The user fill the form and tap on the submit button (E1) (E2) 3. Send a crypted message with the user's infomation 4. The server will update its database and will send back a hash and an ID 5. The hash and the ID are stored in the phone.		
Postconditions:	1. The request page will be displayed		
Exceptions:	E1: Fields with a star are mandatory - A alert will be display "Please fill all the form with a star/asterisk E2: If there are data on the server that partially match the sent message - The application will display 2 windows: - On the left, a window with the information from the phone - On the right, a window with the information from the server - The user choose the data he want to keep - The phone data will be update - Send a crypted message with the user's infomation - The server will update its database and will send back a hash and an ID - The hash and the ID are stored in the phone.		

Priority:	

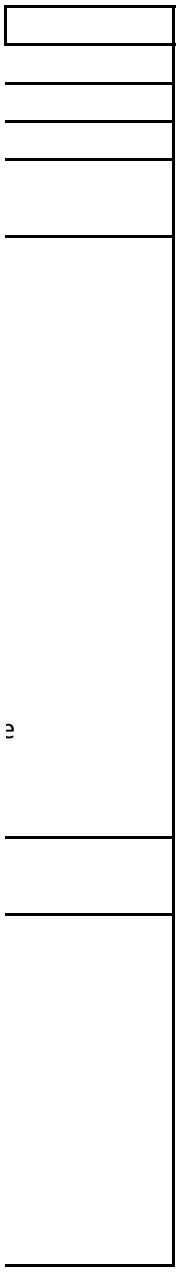
Use Case Name:	Handle pending request	**ID:**	
Actors:	Owner of the phone		
Description:	Allow the user to edit and send a request that has been sent unsucessfuly		
Trigger:			
Preconditions:	1. Open the app 2. Register 3. A request is stored in the phone		
Normal course:	1. Check if there are a pending request on the phone 2. Ask the user to send or not the request again. The request's information will be dis 3. Go to "Request from the patient" or "Request from a bystander" 4. All the field will be filled with the previous answers		
Postconditions:	1. Display		
Exceptions:	E1: The user tap "no" - The pending request will be delete - Display the request page		

Priority:	
splayed on the screen (E1)	

Use Case Name:	Request from the patient	**ID:**	
Actors:	Owner of the phone as a patient		
Description:	Request an ambulance from a patient		
Trigger:	Tap on the button "Request for myself"		
Preconditions:	1. Open the app 2. Register		
Normal course:	1. Define the location: Map, satellite view (E1) - Minimum: latitude, longitude - Other fields: Street number, street, room, floor, building, district (list form) 2. Fill the symptoms list 3. Send the request + hash and ID (E2) 4. The server will send back "something" which display a confirmation message stored		
Postconditions:	1.		
Exceptions:	E1: The GPS doesn't work or the user choose to write the adress 1. The user will have to fill the adress fields 2. The location will be display on the screen - The satellite view will be display on the screen 3. Fill the symptoms list 4. Send the request 5. The server will send back "something" which display a confirmation message stored E2: The hash doesn't match 1. The request will be stored in the phone 2. The application will ask the user to restart the app		

Priority:	
in the phone	
in the phone	

Use Case Nan	Request from the patient	**ID:**		**Priority:**
Actors:	Owner of the phone as a bystander			
Description:	Request an ambulance from a bystander			
Trigger:				
Preconditions:	1. Register completed 2. Tap on the button "Request for someone"			
Normal course:	1. Ask for patient information - name, surname - age (at least an appreciation) - gender (male, female, other) 1. Define the location: Map, satellite view (E1) - Minimum: latitude, longitude - Other fields: Street number, street, room, floor, building, district (list form) 2. Fill the symptoms list, "bystander" version - Bleeding ? Breathing ? Conscious? 3. Send the request + hash and ID (E2) 4. The server will send back "something" which display a confirmation message stored in the phone			
Postconditions:	1.			
Exceptions:	E1: The GPS doesn't work or the user choose to write the adress - The user will have to fill the adress fields - The location will be display on the screen - The satellite view will be display on the screen E2: The hash doesn't match 1. The request will be stored in the phone 2. The application will ask the user to restart the app			



Use Case Name:	Update profile	**ID:**	
Actors:	Owner of the phone		
Description:	Allow an user to modify his information		
Trigger:	Tap on the button "profile" -> edit		
Preconditions:	1. Open the app		
Normal course:	1. A form will be display with the user' information 2. The user can modify these information by tapping on the right field 3. Click on the update button 4. Send information to the server 5. Server will store the information and send back a new host ID 6. The app will store the host ID in a local storage 7. The app will be display a confirmation message		
Postconditions:	1.		
Exceptions:	E1: Tap on the cancel button or return button - Cancel the modification		

Priority:	