**Explanations of the Sequence Diagrams**

**Anonymous Data Request**

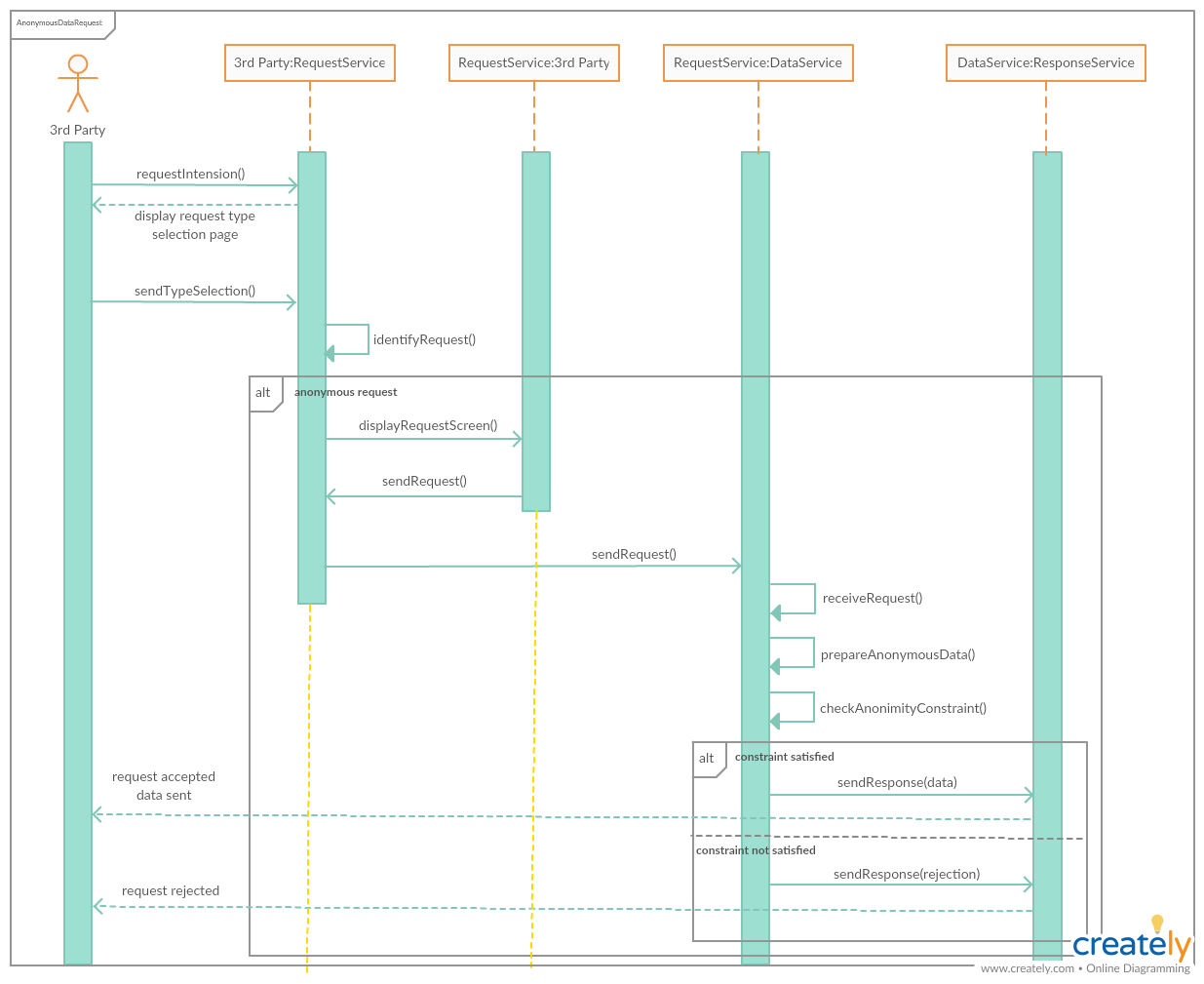
****

Figure 1 - Sequence diagram of anonymous data request

The above figure represents the sequence diagram of Anonymous Data Request. In order to make an anonymous data request, it is first expected from the 3rd party to perform a request intention. By making this request intention, 3rd party informs the system that it wants to request data. Then, this intention of 3rd party is captured by the RequestService and a page asking for the request type; individual or anonymous data is displayed to 3rd party. After this step, the selection of the 3rd party is identified by the RequestService and if the request type is anonymous, the complete version of the request that 3rd party wants to perform is asked. Once again, RequestService receives this request and forwards it to DataService. DataService prepares the requested data and checks its anonymity constraint. If the requested data satisfies the anonymity constraint, then data is sent to the ResponseService in order to be shared with the 3rd party. However, if the anonymity constraint does not satisfy, then the knowledge of request rejection is sent to ResponseService and ResponseService notifies the 3rd party about the rejection of its request.

**AutomatedSOS**

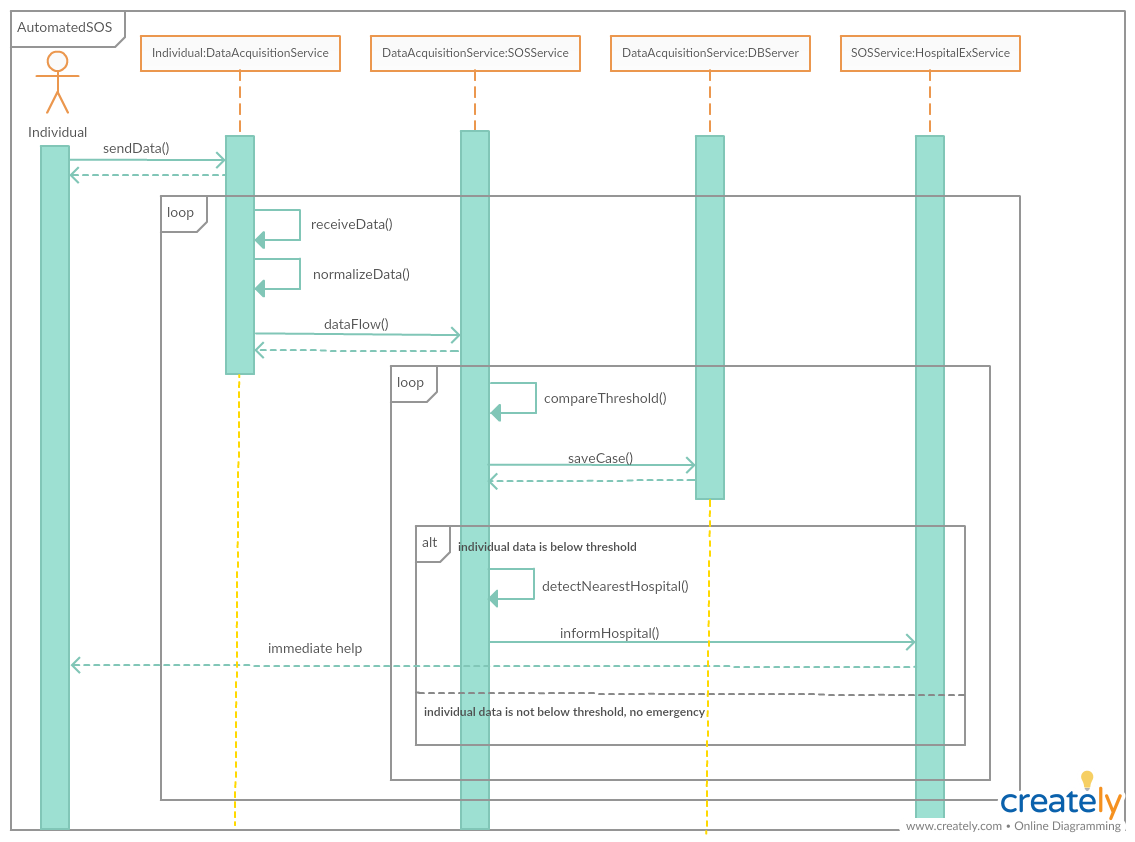


Figure 2 - Sequence diagram of AutomatedSOS

The above figure represents the sequence diagram of AutomatedSOS. On the AutomatedSOS, there is a continuous data flow between the individual and the system.

During the AutomatedSOS, the data of the individual is first received by the DataAcquisitionService. Then, this unstructured data is normalized by the DataAcquisitionService and forwarded to the SOSService in order to decide whether there is an emergency case or not. As the very first step of the SOSService, *compareThreshold()* is invoked and the current health data of the individual is compared with a threshold value. Based on the result of the comparison, SOSService branches into 2 conditions. If it detects an emergency case, it detects the nearest hospital and informs the hospital through an external service called as HospitalExService. On the other hand, if it does not detect an emergency, it will continue to go on with data monitoring.