**State Diagrams**

**StateDiagram-DataRequest.png**

The above diagram is used to describe the behavior of the system when data is requested by a third party. Very briefly, there are two types of data request that can be made by third parties; which can be named as *“Individual Data Request”* and *“Anonymous Data Request”*. According to the type of request, the system undergoes different states.

When a request is made by a third party, system enters an identification state in which it identifies type of the request.

If it is an individual data request, system first enter to “Verifying SSN” state and check the SSN provided by the third party. If SSN is not valid, then system will end up with the state of “Reject” and terminate the request. If the provided SSN is valid, system will enter to the state of “In-List Confirmation” and checks the name of the third party inside the pre-confirmed list of the individual. If the third party is found, system will enter to an “Accept” state and provide the requested data. On the other hand, if the company was not found, then system will enter to “Ask Individual” state to get a request approval from individual. If individual approves the request of his/her data, system will enter to the state of “Accept” and provide individual data to the company. On the contrary, if individual does not approve the request, system will end up with “Reject” state and terminate the data request.

If it is an anonymous data request, system will first enter “Search” state and query the requested data. After finding the requested data, it will check the anonymity constraint of data. To check the anonymity constraint of data, system determines the number of records in the result, if the number of individuals whose data are used for the request are less than 1000 then anonymity constraint is reported as not satisfied. If anonymity constraint holds, system will enter to “Accept” state and give the requested data. On the other hand, if the anonymity constraint is not satisfied, system will enter to “Reject” state and terminate the request of third party.

This is the overall state diagram for the service; *Data4Help*

**StateDiagram-DataMonitor.png**

The above diagram is used to describe the behavior of the system for the service; *AutomatedSOS.*

In this service, system constantly monitors individual data and enters to the state of “Data Comparison”. On this state, it compares the health records of the individual with a predefined threshold.

If it detects an anomaly between the threshold value and individual data, it first enters a new state to locate the individual, then it enters to another consecutive state to find the third party that is closest to individual and then another consecutive state to notify this third party for the emergency situation of the individual.

If no anomaly is detected, then simply system will continue with “Data Monitoring” state and continue to monitor the data of individual.

**StateDiagram-TrackRun\_Organizer.png**

The above diagram is used to describe the behavior of the system when *Data4Run* is used by organizers.

In this service, firstly the usage purpose of organize is identified and system decides its following state according to this.

If the organizer wants to define a path for a run, then the system will enter into an “Organizing Run” state and will request path and time of the run. Then, it will validate and save the path. After saving the path, system will generate a unique code for the run and will pass to the “Confirm” state to finalize the process of run organization.

On the other hand, if the organizer wants to track the athletes, system will move to “Tracking Athlete” phase and ask for the QR code. Then, it will go on with the “verification” state and verify the given code. According to the result of verification, it will either enter to “Accept” or “Reject” state.

**StateDiagram-TrackRun\_Athlete.png**

The above diagram is used to describe the behavior of the system when *Data4Run* is used by athletes. Very briefly, athletes will use this system to enroll for the runs.

In this service an athlete is able to perform two different operations; either participate to a run or display the map of the run that he/she is being currently enrolled. And according to the choice of the athlete, system will enter to two different states.

When an athlete wants to use the service for enrolling to a run, system will request the run code from athlete and then pass to “validation” state to verify the provided QR code. If the code is not valid, system will move to the “Reject” state and terminate the process. On the other hand, if the provided code is verified, then system will enter to a new state in which it will check the starting time of the run. If the run has not already started, then system will go on with the state of “Accept” and let the athlete enroll to the run. However, if the run already started, system will go on with the “Reject” state and terminate the process

And when an athlete wants to use the service for displaying the map of the run that the athlete is being currently enrolled, system will enter to “Display” state and display the map.

**StateDiagram-TrackRun\_Spectator.png**

The above diagram is used to describe the behavior of the system when *Data4Run* is used by spectators. Very briefly, spectators will use this system to track the athletes on a map.

In this service, system will first request the code of the run from spectator, after requesting it, system will pass to the “Validation” state in order to validate the correctness of the code. If the given code is correct, then system will continue with the “Accept” state and display the map to the spectator. However, if the code is not correct, then it will go on with the “Reject” state and terminate the process.