8-6-2019

Framework For Spatiotemporal Visualization of Community Health In a Smart And Connected Community (SCC)

Sharmin Afroz

FRAMEWORK FOR SPATIOTEMPORAL VISUALIZATION OF COMMUNITY HEALTH IN A SMART AND CONNECTED COMMUNITY (SCC)

by

Sharmin Afroz

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Electrical and Computer Engineering

The University of Memphis

August 2019

This thesis is dedicated to my parents, my teachers, my sister, my husband, my uncle, aunty and my cousins for all their love and support they gave me.

ACKNOWLEDGEMENTS

First, I would like to thank my thesis supervisor, Dr. Bashir Morshed for his immense support and guidance throughout my graduate study at the University of Memphis. His consistent supervision gave me a good and clear understanding of the subject matter of my thesis topic. I would also like to thank my professors at the University of Memphis for helping sharpen my knowledge and skills in their taught courses. I am grateful to Dr. Aaron L. Robinson and Dr. Hasan Ali who devoted their precious time through involvement in my thesis committee and provided valuable suggestions.

I would also like to thank the members of the Embedded Systems Advanced Research and Prototyping (ESARP) lab for their knowledgeable and friendly discussion as well as friendship during my research in the lab. I convey special thanks to Dr. Saleha Khatun, Dr. Ankita Mohapatra, Babak Noroozi, Md. Juber Rahman, Md. Sabbir Bin Zaman, Tasnuba Siddiqui, and Haritha Gollakota for their informative comments and suggestions during the past two and a half years.

And finally, I would like to thank my parents for encouraging me throughout my studies to achieve success in my life. I would express my sincere gratitude to them.

PREFACE

One peer reviewed journal paper and conference paper resulting from my research are used as the manuscript of this thesis. Following articles are used to write different chapters in this document:

B. Morshed, B. Harmon, M. Zaman, M. Rahman, **S. Afroz**, and M. Rahman, "Inkjet Printed Fully-Passive Body-Worn Wireless Sensors for Smart and Connected Community (SCC)," *J. Low Power Electron. Appl.*, vol. 7, no. 4, p. 26, 2017.

S. Afroz, B. I. Morshed, "Web Visualization of Temporal and Spatial Health Data from Smartphone App in Smart and Connected Community (SCC)", IEEE International Smart Cities Conference (ISC2), Kansas City, MO, Sept. 16-19, 2018. .

ABSTRACT

Smart and Connected Community (SCC) will use health data of the community members for knowledge generation beyond mobile health (mHealth). Current mHealth only assists individual users to monitor their health status, but do not allow integration and interpretation of collective health data. The objective of this thesis is to exhibit the continuous health status of the community members through a framework of visualization including spatial and temporal plots, such as anonymous user health severity graph, severity flow plot, a severity map view, the cumulative and segmented animation. The framework composes of physiological data collection with smartphones and sharing of anonymous data to SCC health server. Physiological data is sent from the smartphone app in JSON (JavaScript Object Notation) format and stored in the server database. Temporal visualization is presented as graph and flow, whereas spatial visualization utilizes Google Map overlay to display the severity distribution through the color code of severity. Furthermore, an animation mode is developed that displays combined spatiotemporal data over the selected duration in either cumulative or segmented at specified intervals. To implement this, a web-based dynamic server is used. The front end of the server is built with JavaScript JQuery and Ajax, whereas the backend of the server is managed by Hypertext Preprocessor, i.e. PHP, a server-side scripting language. The phpMyAdmin (administration tool for MySQL) stores the JSON data that comes from the smartphone app. To assess the framework, we utilized the MIT-BIH database with pre-recorded data from Arrhythmia patients. We assume each dataset record as a community member (subject). From these records, we classified arrhythmia and measure severity ranging from 0 to 100 considering various severity of arrhythmia (e.g. ventricular tachycardia is the most

severe). These data are then randomized to a different location and fed to the visualization tool for functionally verify and assess the performance of the visualization tool. Furthermore, a survey was conducted to collect feedback about the visualization tool that shows that 81.4% participants in pre-session and 84.75% in post-session provided positive feedback about the visualization of health data. By using this framework, community members can generate collective knowledge that might assist community stakeholders such as the Health Department to improve community health by identifying health issues, developing strategies, and resource allocation.

TABLE OF CONTENTS

List of Figures

List of Tables

I. Introduction

A. Background and Research Objective

The increasing popularity of smartphones, wearables, body sensors, and Internet of Things (IoT) are leading rise to Smart and Connected Communities (SCC). The widespread use of these technologies has huge potential for today's advanced health monitoring systems. Modern computing, communication, and sensor technologies can collect physiological signals, which can be efficiently utilized to improve individual as well as community health. In addition, many people with the chronic disease might be able to reduce visit hospitals with effective everyday monitoring. However, it is difficult to track the disease conditions of various types and engage the participants to share their health data over time due to the lack of visualization tools and privacy concerns. In order to monitor health with mobile phone and to reduce their hospital visits, some body-worn sensor technologies are already incorporated in mobile health (mHealth) technologies. We extend this concept for SCC by incorporating sharing of anonymous severity only that preserves privacy and a framework of visualization tools that provide community-level health status information. In this system, sensor data can be collected through smartphones with our custom app (research conducted by other researchers in our research group), and compute disease severity that can be sent to the server. Web server can be utilized to visualize these health data for the stakeholders (such as community members, city or county health department, non-profit health organizations, hospitals, and clinics). Furthermore, visualization can allow understanding of the conditions of diseases around their community.

Objectives of the thesis are as follows:

1) Develop a web-based framework of spatiotemporal visualization tool for Smart and Connected Community (SCC) for various diseases like arrhythmia, chronic obstructive pulmonary disease (COPD), flu, and sleep apnea.
2) Maps space-time severity of disease events in a community in real-time.
3) Incorporating animation to the visualization to make it more interactive to users.

The thesis statement is to increase user participation with privacy in the visualization tool. It was achieved by the thesis work, which exhibits the continuous health status of the community members through a framework of spatiotemporal visualization. The survey demonstrates the efficacy of the work to fulfil the thesis statement.

B. Proposed Solution

Our proposed solution offers the area of physiological data visualization toolkit for SCC health visualization that still needs to be resolved, for example, i) developing a system to collect physiological data directly from community users and compute disease severity, ii) lack of data sharing and aggregation of data for community level health status monitoring in real-time while keeping privacy, and iii) availability of spatiotemporal visualization of disease severities with respect to community health. This research attempts to fill up this technological gap.

This work proposes a framework for spatiotemporal graphical visualization of the community-based Smart and Connected Community (SCC) Health data through multiple plots and an animation interface. We have developed the SCC-Health web server framework for visualization from scratch for our National Science Foundation (NSF) funded project using a purchased domain (http://sccmobilehealth.com/) and demonstrated full functionally with mock data and real data collected from Memphis community.

The benefits of this proposed framework are:

a) SCC members contribute and gather knowledge about the community health status from the temporal and spatial plots.

b) As personal or identifiable information is not shared, SCC members will likely to be more confident about their privacy and inclined to share anonymized EoI to the server.

c) Stakeholders can also take advantage of the platform by monitoring the most affected diseases, spreads, and trends of the community and efficiently deploy resources.

Our research group has been developing a smartphone app and a novel set of battery-less sensors to collect physiological data from users at home [1]. These data need better visualization in both time and space (spatiotemporal) domains to be fully observable, understandable, and utilizable to the stakeholders. This might also help to understand how the data are correlated with each other, changes over time (for example seasons), and distributed over the community. None of the existing health data web visualization tools meet these requirements. In our work, we only collect and plot the severity of the user's health condition (based on the specific disease) with a code number for the user and provide the information to the overall community with dividing the map in grids such that the user's location cannot be identified. It helps the person to share data to the SCC webserver without worrying for identifiable information. None of the previous work has been conducted for the betterment of the whole community by engaging the individual person. For this, the severities of a person's health data are taken from the smartphone and then sent to the SCC Health web server, and from the server, one can visualize how a person's data are changed through some time interval by temporal plot and also the distribution of the disease in several regions through the spatial plot. The project especially considers the privacy and anonymity of the participants. In the spatial plot, it can be visualized that one's environment is influenced by others. In the smart and connected community, it is not only the sole responsibility of individuals, but the whole community needs to be engaged to lead a healthier life [2].

C. Achievements

- **S. Afroz**, B. I. Morshed, "Web Visualization of Temporal and Spatial Health Data from Smartphone App in Smart and Connected Community (SCC)", IEEE International Smart Cities Conference (ISC2), Kansas City, MO, Sept. 16-19, 2018. .

- B. Morshed, B. Harmon, M. Zaman, M. Rahman, **S. Afroz**, and M. Rahman, "Inkjet Printed Fully-Passive Body-Worn Wireless Sensors for Smart and Connected Community (SCC)," *J. Low Power Electron. Appl.*, vol. 7, no. 4, p. 26, 2017.

- **S. Afroz** and Bashir I Morshed. " An Interactive Web-based Visualization Tool of Community Health Data for a Smart & Connected Community (SCC)." 5th Annual EECE Poster Competition (3rd place). Department of Electrical and Computer Engineering, The University of Memphis, Memphis. 24 Apr. 2017. Poster presentation.

II. Background

In the literature, several applications of web-based visualization are reported. The spatial and temporal plot for web GIS platform are utilized in [3] to monitor and allocate the zero energy settlement in residential building in the district level. For the visualization of England Strategic health data, google map API, Google Earth KML (Keyhole Markup Language) version, and MSN Virtual Earth Map Control version have been exploited to introduce online mapping trends [4]. The work presented in [5] demonstrates a predictive modeling for patients events through web service by integrating a database of 31,855 ICU patients, where the patients are allowed to upload their data from web service and see the visualization of their events. Web visualization of clinical data based on pie charts and continuous calendar data to see the continuity of patients' data has been presented in [6]. To visualize the continuous data of stroke patients, a TimeSpan web visualization is presented in [7] as the data need a deep understanding of intervals time to save life. The work presented in [8] demonstrates a methodology to improve the health behavior of children through data visualization on the web. There are lot of research papers on spatial web visualization to show the vulnerability of climate change on human health [9], [10], [11]. Research work in [12] presents the spatial animation of ovarian cancer throughout the United States. Web based visualization implemented in [13] is an automated tool for querying, filtering, gathering, and demonstrating unstructured articles on disease outbreaks. Like [13], the web-based application developed in [14] is an integrative information processing visualization system that employs modern communication tools and electronic resources to generate an information environment for augmenting the surveillance of growing infectious disease warning to global health. An open access web-based geospatial toolkit is developed in [15] with an interactive interface for the monitoring of dengue fever outbreaks, in space and time with GIS framework. Authors in [16] developed a web-based tool that permits interactive observation of PCHi-C interaction maps and integration with both public and user-defined genomic datasets. Work presented in [17] is a web-based online resource that aims to provide tools for the functional interpretation of large lists of genes. This work also uses the third-part developed database instead of real-time data collection.

Different types of diseases such as, Myocardial Infarction, AV Block, Ventricular Tachycardia and Atrial Fibrillation can be detected from electrocardiogram (ECG) signals with an estimated 300 million ECGs recorded annually [18]. In this thesis, the task of arrhythmia detection has been conducted from the ECG record. ECG records have been collected from the MIT-BIH database. Arrhythmia classification problem by analyzing the ECG signals for arrhythmia have been addressed by utilizing different classification techniques such as the autocorrelation function, frequency domain features, time frequency analysis, and wavelet transform, sequential hypothesis testing, and morphological features [19]. Most of the work in literature is focused on arrythmia detection and classification. In this thesis, the severity ranking of arrhythmia has been considered and demonstrated in developed web-based visualization tool.

III. Health Data Collection, SCC Health Database and Visualization to End User

A. Health Data Collection

Physiological data from community members are collected with sensors. These sensors [20], [21] send data to scanner [22] that parse sensor raw data and send it to a smartphone via Bluetooth [18]. In smartphone, SCCHealth app parses sensor data into EoI value through several severity ranking algorithms [23], [24],[25] and [26]. These EoI values are shared with SCC health server from the SCCHealth app [26], [27].

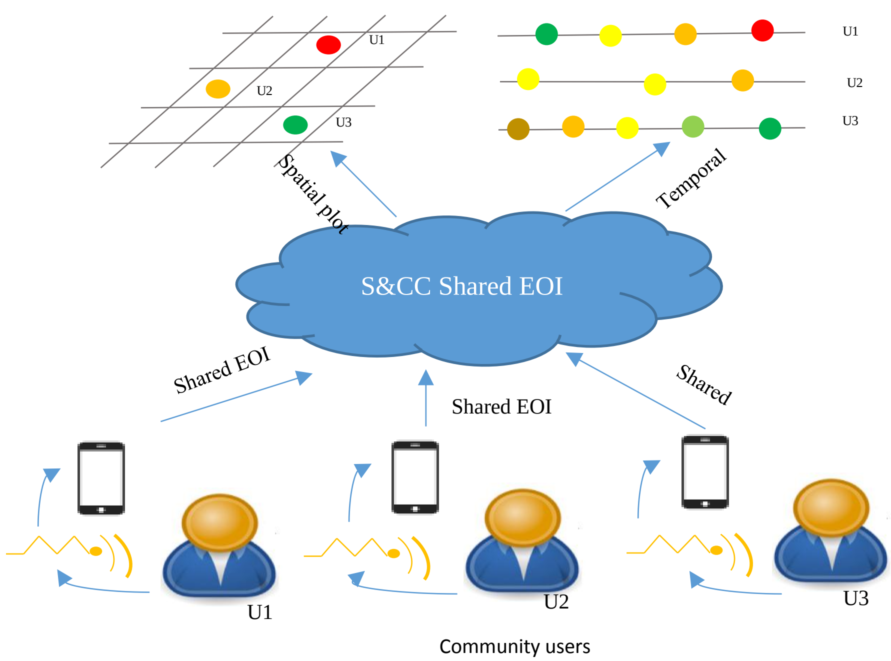


Figure 1. The physiological data flow from participants to SCC Health server.

From the server, registered participants of the community can see the visualization of physiological data through temporal and spatial plot. Figure. 1 demonstrates the physiological data flow from participants to the SCC Health server.

Smartphone app data are sent to the database into JSON (JavaScript Object Notation) format. These data include the participant's hash Id, area code, EoI (Event of Interest), disease type, date time and algorithm type. All participants have a unique Id.

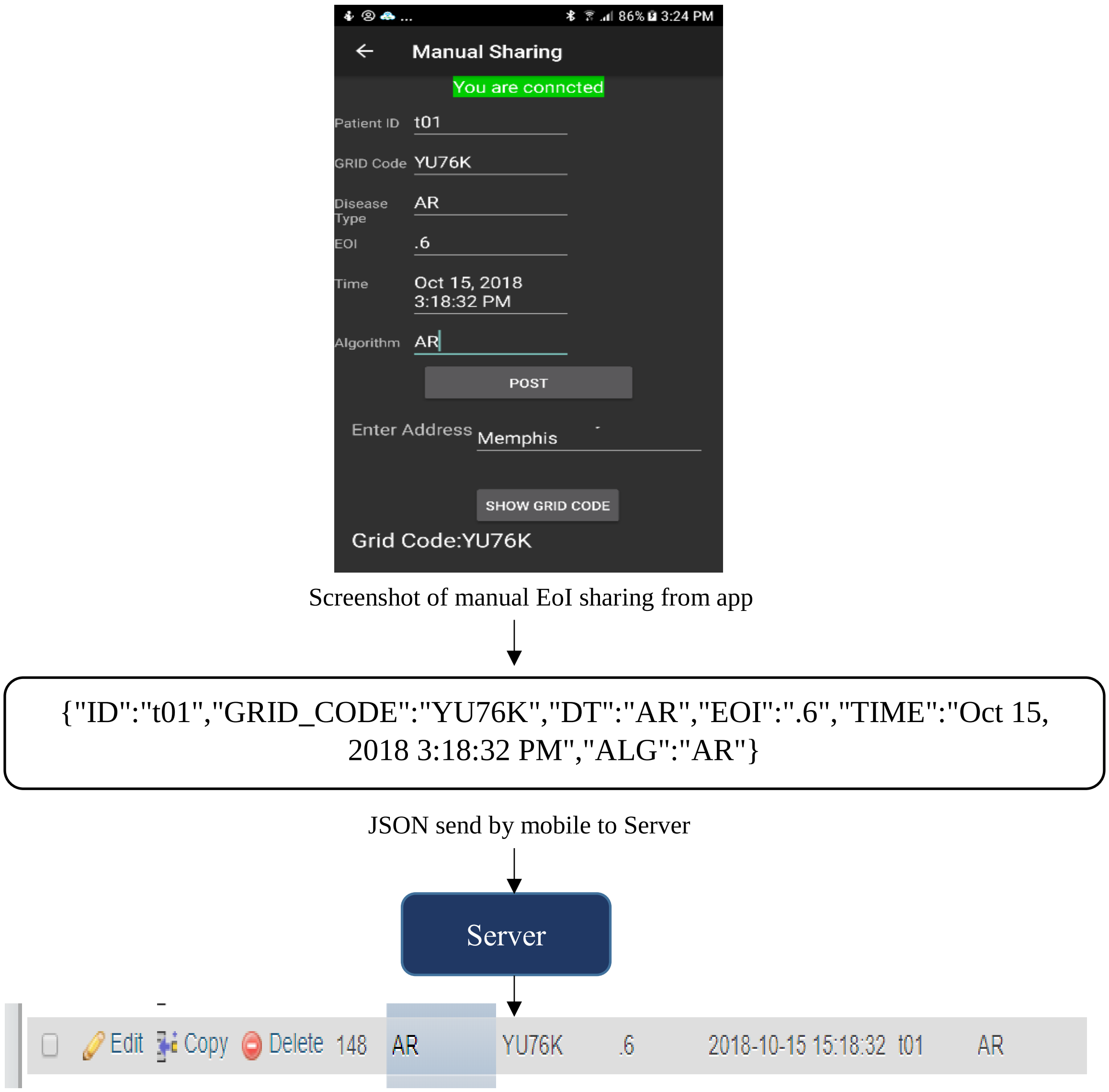


Figure 2. JSON dataflow from a smartphone app to the SCC Health Server

The surrounding Memphis areas are divided into 19 grids with a unique area code name shown in Fig. 3. 15 grids are equal in size but only 1 grid is divided into 4 equal small grids as for its dense population.

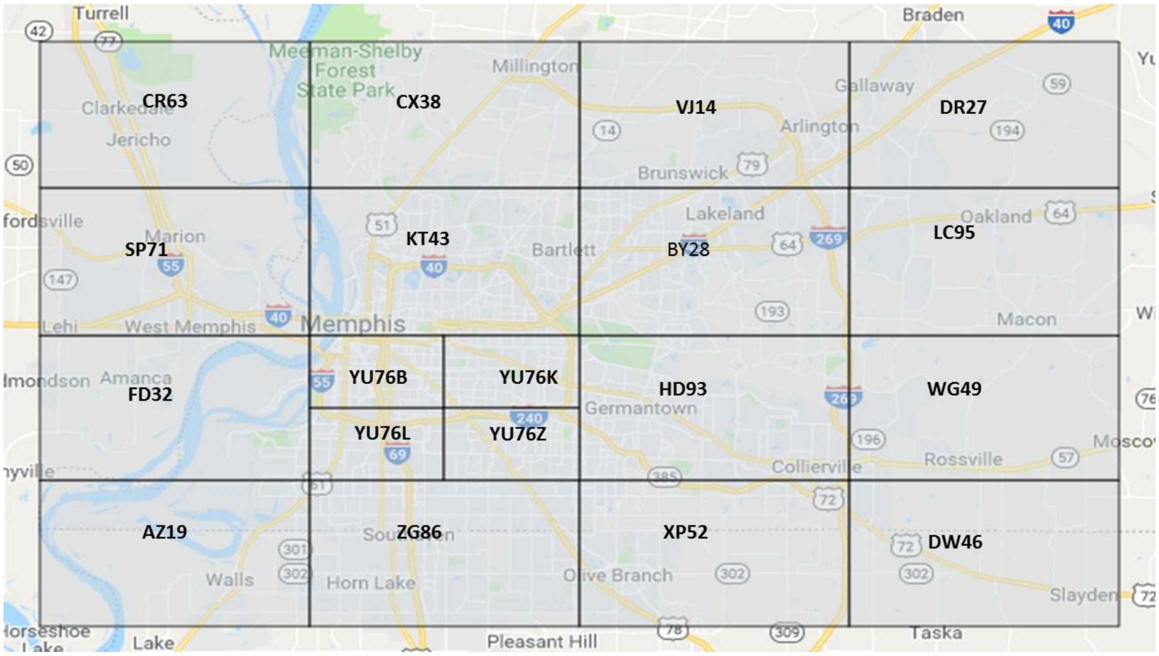


Figure 3. Memphis Grid area with code name

The mobile app has the mechanism to convert the participant's home address into area code. In the SCC Health server, participants share only their area codes rather than their actual address to ensure their privacy, EoI and severity of their disease. Four chronic diseases i.e. Arrhythmia, Chronic Obstructive Pulmonary Disease, Flu, and Sleep Apnea are visualized through SCC Health Web Server. The date and time of the person is the time of their checking health status. Different types of algorithms [23], [24], [25] are used to calculate the severity of the same disease and the user has the facility to choose one of them. So, the choice of algorithm for testing the disease is also included for further feedback.

After completing the testing of a disease in smartphone, if a participant wants to share his/her data to the SCC Health Server, s/he just needs to click a button from app and all information are bundled into a JSON that is shown in Fig. 2 and transferred to the SCC Health web database.

B. *SCC Health Database (Backend)*

MySQL is used as database for SCC Health server. Here phpMyAdmin is an administration tool for MySQL. This database has two tables (Figure. 4): participants' data table and mock data table. Participants' data table consists of directly collected physiological data from the community members and mock data table consists of some mock data that are randomly created.

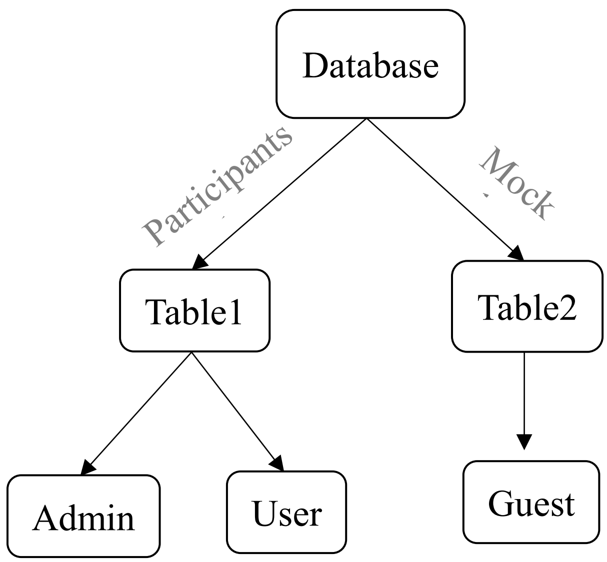


Figure 4. Database Table Access

In the backend, JSON data are encoded with PHP and inserted into the participants' table of the database. If two participants' send data at the same time, server obeys the rule of FIFO (First-In-First-Out) and inserts the data, which has come first and keeps others in a queue.

C. *Visualization to End User (Frontend)*

People from all over the world can access to this visualization. But to visualize the SCC Health data there are three types of login credentials, i.e.

1) **Admin:** Admin login has full access to all physiological data and different types of data management. It has spatial plot (google map overlay), temporal plot (time plot with flow and graph), and all data from the database of the participants' data table. It has the privilege to create login credential for the participants/user. Admin can also add .dex file for the algorithm of detecting diseases. To be logged as an admin, one should have certain credentials.

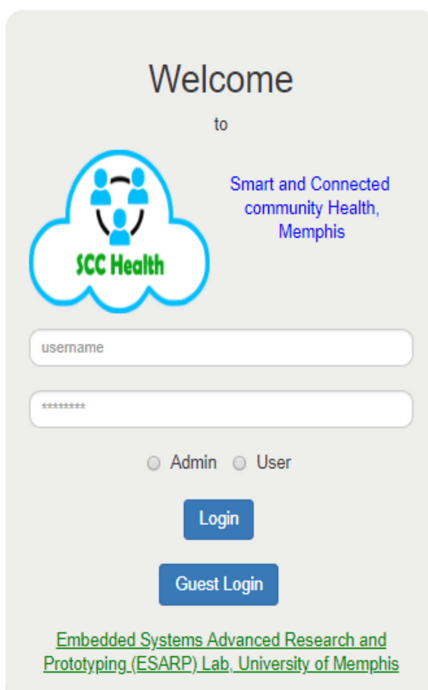


Figure 5. SCCHealth Web Login page

2) **User:** User login has the privilege to visualize the participants' data with spatial plot and two temporal plots. Login credentials are created by the admin for the recruited subjects, and then the subjects can have access as a user.

3) **Guest:** Guest has the same privilege as the user. But the only difference is that guest can visualize spatial plot and temporal plots with mock data. These mock data are arbitrarily entered to test the functionality of the web visualization and is not collected from participants. There is no requirement of login credential for guest login. Anyone can visualize the plot through this login.

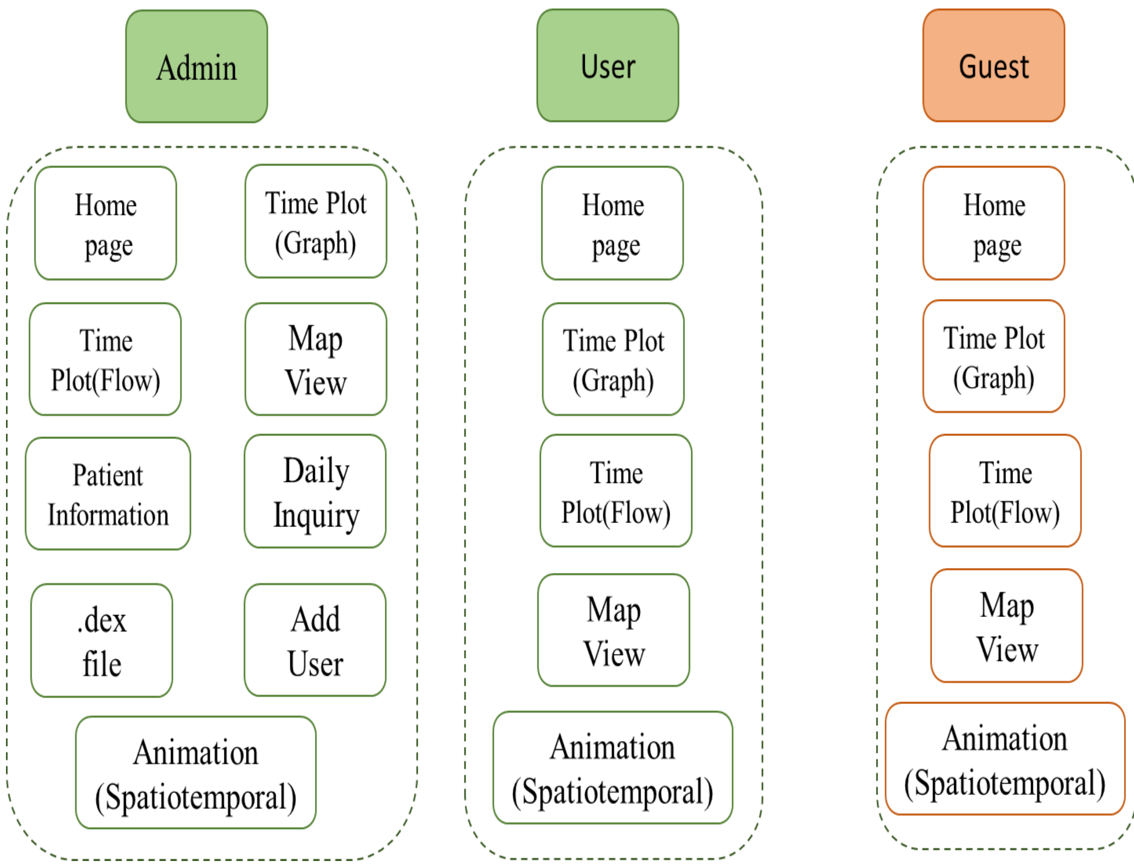


Figure 6. Visualization features to end user

Figure 5 shows SCCHealth server login page where admin and user login require credentials. But on the other hand, guest login requires no credentials so that people around the world can access it and get a concept of how this physiological data visualization works. Figure 6. shows visualization features of admin, user, and guest. Guest can visualize ‘Time plot Graph’, ‘Time plot Flow’, ‘Map view’ and ‘Animation of spatiotemporal’ with mock data. User has similar accessibility of visualization but the only difference with a guest is that user can visualize data with participant data. Admin has similar accessibility like user with some additional features like ‘Patient information’ (where all patients’ information have been accumulated in a table), ‘Daily inquiry’ (where patients’ information within a specific date duration are shown in table), ‘Add User’ (where Admin can create login credentials for user) and ‘.dex file’ (where algorithms for disease detection in mobile are uploaded).

IV. Example Case: Arrhythmia Severity

With the visualization tool different physiological data such as arrhythmia, sleep apnea, COPD (Chronic Obstructive Pulmonary Disease), flu can be viewed in real time. In this thesis, we considered arrhythmia disease to show the effectiveness of the visualization tool.

A. Operation of human heart

The heart is a critical organ that is essentially a pump driven by electrical impulses generated and conducted by internal systems. The heartbeat (contraction) begins when an electrical impulse from the sinoatrial node (also called the SA node or sinus node) moves through it [28]. The normal electrical sequence begins in the right atrium and then spreads throughout the atria to the atrioventricular (AV) node.

The heart has four chambers, two on the right and two on the left:

1) Two upper chambers are called atria (one is called an atrium),

2) Two lower chambers are called ventricles.

It also has four valves that open and close to let blood flow in only one direction when the heart contracts (beats). The four heart valves are:

1) A tricuspid valve located between the right atrium and right ventricle

2) Pulmonary or pulmonic valve, between the right ventricle and the pulmonary artery,

3) Mitral valve, between the left atrium and left ventricle and

4) A aortic valve, between the left ventricle and the aorta.

The cardiac cycle begins with the contraction of both atria to actively fill the main pumping chamber. If the chamber is filled, then the valve opens and drives the blood into arterial system. Sometimes there occur abnormalities during flowing blood and circulation of blood through body. There might be some abnormalities of the SA node to impulse electric signal [29].

Heart is the most important organ of a human body as blood is circulated through the whole body from this organ and the blood with carbon-dioxide come into this organ and then purify from it. Its harmonic rhythm keeps our body healthy and perfect. Without the proper functioning of this organ, our life will fall in danger and sometimes might cause death.

B. Arrhythmia

Arrhythmia is the abnormal rhythm of the heart generated by electrical impulses from the sinusoidal node (SA). Sometimes SA node can't generate regular electrical impulses to the heart and for that reason, heart can't pump blood to the vein in a regular motion. Sometimes SA node generates electrical impulse at a faster speed, so that blood can't go out and in heart properly. So, it causes fast irregularity to beat in heart. For this reason, it causes arrhythmia.

Arrhythmia is classified into two categories: One is based on the frequency of generating the heartbeat in one minute (Tachycardia and Bradycardia) and another is based on where they occur in the body. Sometimes electrical pulse occurs in some other part of the body rather occurring in the SA node. Atrial fibrillation occurs in this situation. Sometimes the ventricular valve can't open properly and for this blood can't go through the ventricles. So, the atrium becomes quiver than the normal size and hinders the flow of blood. Ventricle tachycardia occurs in this situation.

According to [30] the most dangerous (potentially life-threatening) arrhythmias that require evaluation for possible acute and chronic therapy include

1) Sustained ventricular tachycardia in the setting of heart disease

2) Ventricular fibrillation (cardiac arrest)

3) Atrial fibrillation

4) Supraventricular tachycardia

5) Sinus bradycardia (and pauses)

6) Atrioventricular (AV) block

C. *Arrhythmia Database*

To visualize the data in the web server, we use arrhythmia disease. MIT_BIH database is used to collect data for arrhythmia disease. It has 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979 where 60% are inpatient and 40% are outpatient. Among 48 records, there are 25 men aged 32 to 89 years, 22 women aged 23 to 89 years and record 201 and 202 came from the same male subjects. Every record duration is slightly over 30 minutes long. Record of this database are:

`100,101,102,103,104,105,106,107,108,109,111,112,123,114,115,116,117,118,119,121,122,123,124,200,201,202,203,205,207,208,209,210,212,213,214,215,217,219,220,221,222,223,228,230,231,232,233,234.`

There are two groups in this database:

Group 1: 23 records (numbered from 100 to 124)

Group 2: 25 records (numbered from 200 to 234)

Group 2 has more complex arrhythmia than Group 1 such as complex ventricular, junctional, and supraventricular arrhythmias and conduction abnormalities.

In our case, every record in the MIT-BIH database is assumed as a member of a community. The record number is used for the server database as the individual patient ID. For example, 100 is a member of community and he or she has patient ID 100. Though 47 people participate in MIT-BIH database to generate 48 records, mobileHealth server database has 48 members considering the number of records of the MIT-BIH database.

D. *Arrhythmia beat detection*

With the help of ECG-kit every beat of the record is identified. ECG-kit classifies heartbeat into four classes [31]:

- Normal (N)
- Supraventricular (S)
- Ventricular (V)
- Fusion of normal and ventricular (F)

Supraventricular beat occurs in the upper chamber of heart and ventricular beat occurs in lower the chamber of heart. Some abnormal beat happened before normal and ventricular beat is known as fusion beat [32].

Table 1. Beat classification with ECG-Kit

Records	**N**	**F**	**S**	**V**	**Total beats**
100	2239	0	33	1	2273
101	1865	0	0	0	1865
102	142	0	0	2045	2187
103	2070	0	0	9	2084
104	2226	0	1	2	2229
105	2486	2	0	80	2572
106	1507	0	142	378	2027
107	2057	0	0	79	2137
108	1722	0	5	31	1763
109	2506	0	0	26	2532
111	2124	0	0	0	2124
112	2539	0	0	0	2539
113	1775	0	19	1	1795
114	1842	0	5	32	1879
115	1951	0	0	2	1953
116	2302	1	0	109	2412
117	1535	0	0	0	1535
118	2187	0	82	9	2278
119	1543	0	0	444	1987
121	1861	0	0	2	1863
122	2474	0	0	0	2476
123	1514	0	2	1	1518
124	1585	0	0	34	1619
200	1776	2	31	788	2601
201	1291	0	578	91	1963

Table 1 continued: Beat classification with ECG-Kit

Record	**N**	**F**	**S**	**V**	**Total beats**
202	1455	0	630	50	2061
203	1822	70	618	431	2980
205	2594	1	15	46	2656
207	1513	2	110	232	1860
208	1716	186	26	1025	2955
209	2646	0	358	1	3005
210	2393	21	59	173	2650
212	2489	0	1	258	2748
213	2930	63	0	258	3251
214	1918	16	10	252	2262
215	3178	18	26	139	3363
217	2047	0	0	161	2208
219	1982	0	78	64	2154
220	1961	0	87	0	2048
221	1959	0	70	398	2427
222	1705	0	654	124	2483
223	2179	1	54	369	2605
228	1698	0	0	355	2053
230	2256	0	0	0	2256
231	1555	0	14	2	1571
232	215	0	1172	273	1780
233	1970	359	78	672	3079
234	2718	18	18	0	2753

After classifying each beat as N, S, V, F of every record, we estimate the total ventricular beat of each record. Ventricular arrhythmia is a most severe arrhythmia. The presence of three or more ectopic ventricular complexes in a row with 100 BPM than it is considered as ventricular tachycardia. There are 2 types of ventricular tachycardia: sustained and non-sustained ventricular tachycardia. If ectopic beat happens for more than 30s than it is considered as sustained ventricular tachycardia and otherwise it is non-sustained ventricular tachycardia. Non-sustained ventricular tachycardia is considered as benign and sustained ventricular tachycardia is considered as most severe [32]. In this thesis, heart rate is not considered as determining severity ranking for simplicity. But total ectopic beats present in record are considered as a measurement of severity ranking of arrhythmia. Here ventricular beat is considered as the most severe beat and then supraventricular beat, then fusion beat.

E. Severity ranking

We have used five severity ranking classes in this study i.e.:

i. Normal
ii. Mild
iii. Moderate
iv. Severe
v. Very Severe

The record requires several conditions to obtain a severity ranking category i.e. given in Table 2.

Table 2. Records’ required conditions to obtain a severity ranking

Label of Severity Ranking	Conditions
Normal	V = 0 & S = 0
Mild	V <=10&&S<=100
Moderate	(V >10&<=100) &((V+S)>80) & ((V+ S) <400)
Severe	(100<V<=500) & (S+V<1100)
Very Severe	V >500

By following conditions from Table. 2, records with categorized severity with beat count are given in Table 3-7.

Table 3. Normal category records with beat count

Normal Records	**N**	**F**	**S**	**V**	**Total Beats**
101	1865	0	0	0	1865
111	2124	0	0	0	2124
112	2539	0	0	0	112
117	1535	0	0	0	1535
122	2474	0	0	0	2476
230	2256	0	0	0	2256

Table 4. Mild category records with beat count

Mild Records	N	F	S	V	Total Beats
100	2239	0	32	1	2273
103	2070	0	0	9	2084
104	2226	0	1	2	2229
113	1775	0	19	1	1795
115	1951	0	0	2	1953
121	1861	0	0	2	1863
123	1514	0	2	1	1518
220	1961	0	87	0	2048
231	1555	0	14	2	1571
234	2718	18	18	0	2753

Table 5. Moderate Category records with beat count

Moderate Records	N	F	S	V	Total Beats
105	2486	2	0	80	2572
107	2057	0	0	79	2137
108	1722	0	5	31	1763
109	2506	0	0	26	2532
114	1842	0	5	32	1879
118	2187	0	82	9	2278
124	1585	0	0	34	1619

Table 5 continued: Moderate Category records with beat count

Moderate Records	N	F	S	V	Total Beats
205	2594	1	15	46	2656
209	2646	0	358	1	3005
219	1982	0	78	64	2154

Table 6. Severe Category Records with beat count

Severe Records	N	F	S	V	Total Beats
106	1507	0	142	378	2027
116	2302	1	0	109	2412
119	1543	0	0	444	1987
201	1291	0	578	91	1963
202	1455	0	630	50	2061
203	1822	70	618	431	2980
207	1513	2	110	232	1860
210	2393	21	59	173	2650
212	2489	0	1	258	2748
213	2930	63	0	258	3251
214	1918	16	10	252	2262
215	3178	18	26	139	3363
217	2047	0	0	161	2208
221	1959	0	70	398	2427

Table 6 continued: Severe Category Records with beat count

Severe Records	N	F	S	V	Total Beats
222	1705	0	654	124	2483
223	2179	1	54	369	2605
228	1698	0	0	355	2053

Table 7. Very Severe Category Records with beat count

Very severe Records	N	F	S	V	Total Beats
102	142	0	0	2045	2187
200	1776	2	31	788	2601
208	1716	186	26	1025	2955
232	215	0	1172	273	1780
233	1970	359	78	672	3079

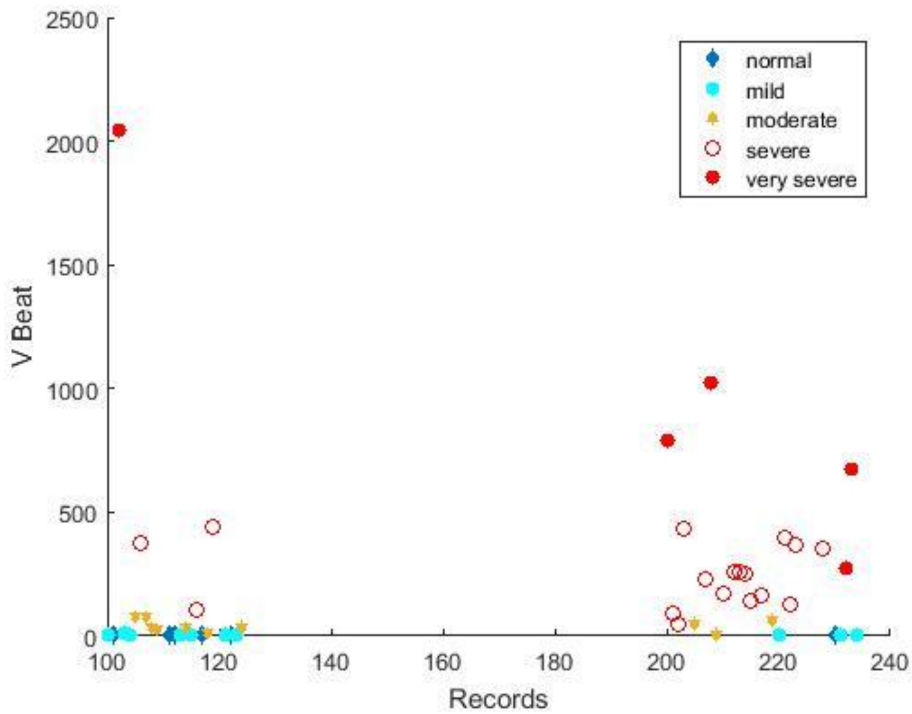


Figure 7: Records VS Ventricular beat

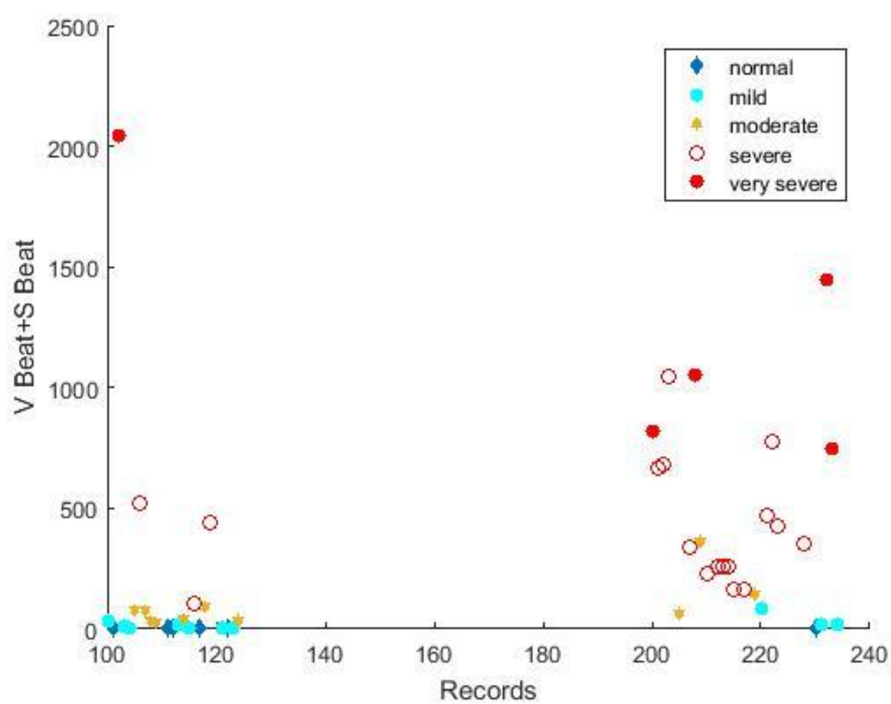


Figure 8: Records VS (Ventricular beat + Supraventricular Beat)

In Figure 7, records vs Ventricular beat and in Figure 8 records vs the sum of ventricular beat and super-ventricular beat have been plotted to visualize how data are associated with each other. It has been shown that only one record is far away from all other records. So, record 102 is discarded from our database and finally, the member of SCCHealth server is 47.

In our case, we need severity value not just normal, mild, moderate, severe and very severe. So, for that case, we weighted our record with some arbitrary value that is alpha value.

$$Severity = N*Alpha_1+F*Alpha_2+S*Alpha_3+V*Alpha_4$$

Here $Alpha_1<Alpha_2<Alpha_3<Alpha_4$

$Alpha_4$ has given more weight than others as it is associated with Ventricular beat.

For choosing Alpha, 11 models have been chosen. After taking weight every model has been normalized.

Table 8. Model parameter with Normalized Value [33]

Model Name	**Parameter**	**Normalized Value**
tenpower	$10^1,10^2,10^3,10^4$	0.0900 0.9001 9.0009 90.0090
multy25	25,50,75,100	10.0000 20.0000 30.0000 40.0000
power2	$2^2,2^4,2^6,2^8$	1.1765 4.7059 18.8235 75.2941
cubic	$2^3,3^3,4^3,5^3$	3.5714 12.0536 28.5714 55.8036
exp	e^2, e^3, e^4, e^5,	3.2059 8.7144 23.6883 64.3914
sqrt	$\sqrt{.1}, \sqrt{1}, \sqrt{10}, \sqrt{100}$	2.1841 6.9068 21.8412 69.0679
fourpower	$4^2,4^3,4^4,4^5$	1.1765 4.7059 18.8235 75.2941

Table 8 continued: Model parameter with Normalized Value [33]

Model Name	Parameter	Normalized Value
powerten	$2^{10}, 3^{10}, 4^{10}, 6^{10}$	0.0017 0.0959 1.7029 98.1995
inversetenth	.1,1,10,100	0.0900 0.9001 9.0009 90.0090
tangent	tan1°, tan22.5°, tan45°, tan67.5°	0.4539 10.7703 26.0018 62.7740
onethird	0.001,33.33,66.66,99.99	0.0005 16.6666 33.3332 49.9997

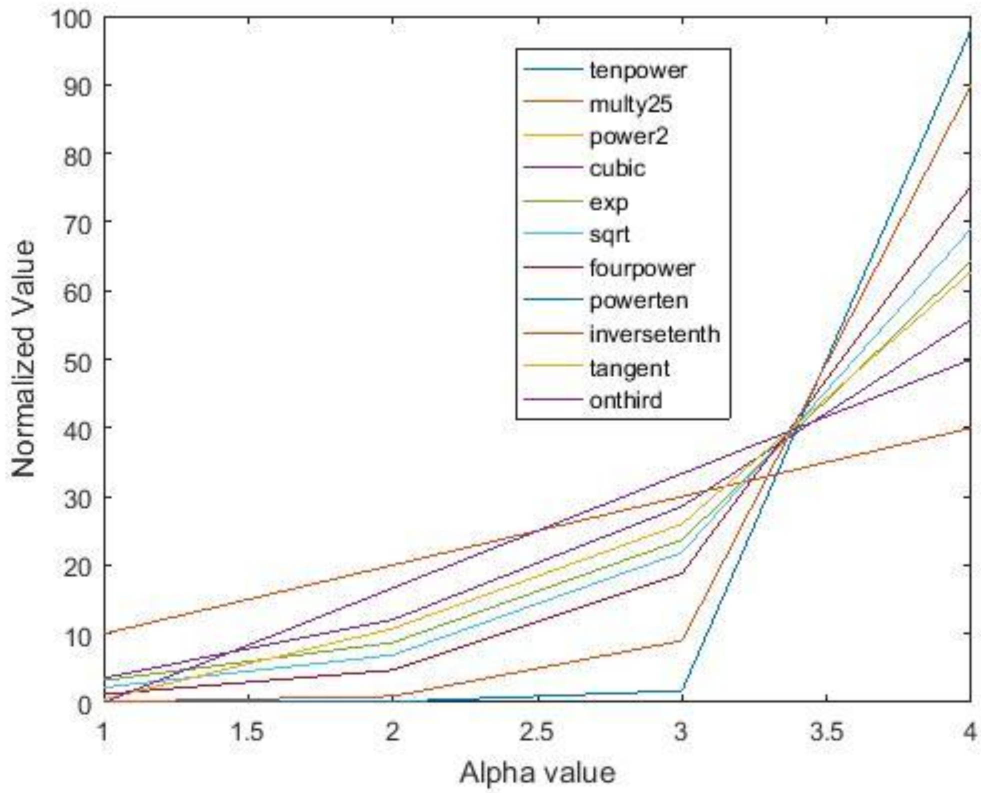


Figure 9. Slope of the normalized model parameter [33]

We check our MIT-BIH database with all models to see the distribution of the data. After giving weight with all model, they are normalized. From Figure 9, it has been shown that model 'tenpower 'has a sharp slope and 'onethird' has a gradual increase in slope. However, all models are checked with normal (Table 3), mild (Table 4), moderate (Table 5), severe (Table 6) and very severe record (Table 7) to compare severity max-min of all models. Table 7 shows the severity max-min comparison of all records for all models.

Table 9. Severity max-min comparison of record value for all models

Severity / Model	Normal		Mild		Moderate		Severe		Very Severe	
	min	**Max**	**Min**	**max**	**min**	**max**	**min**	**max**	**min**	**max**
Power10	0.090009	0.090009	0.15354	0.47927	0.76602	3.4157	3.882	20.183	20.035	84.171
Multi25	10	10	10.031	10.85	10.308	12.393	11.36	18.837	18.22	38.052
Power2	1.1765	1.1765	1.2486	1.9261	1.9376	4.0578	4.3972	17.738	18.211	70.482
Cubic	3.5714	3.5714	3.6249	4.6334	4.1078	6.5672	5.9354	16.681	16.594	52.412
Exp	3.2059	3.2059	3.2685	4.076	3.8341	5.8017	5.9242	16.878	17.721	60.419
Sqrt	2.1841	2.1841	2.2526	3.0192	2.8709	4.9213	5.1276	17.129	17.83	64.725
Fourpower	1.1765	1.1765	1.2486	1.9261	1.9376	4.0578	4.3972	17.738	18.211	70.482
PowerTen	0.001663	0.001663	0.013398	0.42676	0.23702	3.6335	2.8034	21.944	17.352	91.824
InverseTenth	0.090009	0.090009	0.15354	0.47927	0.76602	3.4157	3.882	20.183	20.035	84.171
Tangent	0.45386	0.45386	0.51768	1.5392	1.0938	3.5183	3.2744	15.201	15.905	58.728

In Table 9, some models have overlap with their consecutive severity level. Model power10, multy25, power2, exp, fourpower and inversetenth have no overlap with severity level. But, model cubic, sqrt, powerten, tangent have some overlap with their consecutive severity level. It has been shown that model powerten has the lowest minimum value 0.001663 and multi25 has the highest minimum value 10 in normal severity level. On the other hand, Model multi25 has lowest minimum value 38.052 and powerten has highest maximum value in very severe level. So, Model powerten has the most widest range of 0.001663 to 91.824 and then model inversetenth and power10 have range from 0.090009 to 84.171. But powerten has two overlaps in severity level i.e mild with moderate and moderate with severe level.

As every record is considered as a member of the community, the record is segmented into different parts. Each segmented part contains 500 beats except last part of the record. It contains rest beat of the record. As all records are unequal in size, the number of segmented parts of each record is different than others. Every single segmented part of each record is considered as one-day test data of arrhythmia of that member.

Table 10. Segmentation part of record 100, 101, 103, 104, and 105

Records	N	F	S	V
100	494	0	4	1
100	498	0	2	0
100	489	0	11	0
100	488	0	12	0
100	269	0	4	0
101	500	0	0	0
101	500	0	0	0
101	500	0	0	0
101	365	0	0	0
103	500	0	0	0
103	500	0	0	0
103	492	0	0	6
103	495	0	0	3
103	83	0	0	0
104	497	0	1	2
104	500	0	0	0
104	500	0	0	0
104	500	0	0	0

Table 10 continued: Segmentation part of record 100, 101, 103, 104, and 105

Records	N	F	S	V
104	229	0	0	0
105	485	0	0	15
105	490	0	0	10
105	496	0	0	4
105	460	2	0	36
105	483	0	0	15
105	72	0	0	0

Table 10 shows segmented part of record 100,101,103,104 and 105. All records' segmentation has been shown in index. Record 100,103,104 have five segmented parts whether 101 record has only four and 105 has six.

In the SCCMobileHealth server, color code range of severity is 0 to 100. So, our motive is to find a model that can distribute severity throughout 0 to 100 seeming that it is a real-life distribution. To see the distribution of records' severity level, every model has been plotted through severity level zone vs segmented record numbers. Each severity level length is 20.

Table 11. Severity Lebel with ranking range

Severity level	**Range of Severity ranking**
Normal	0-19
Mild	20-39
Moderate	40-59
Severe	60-79
Very Severe	80-100

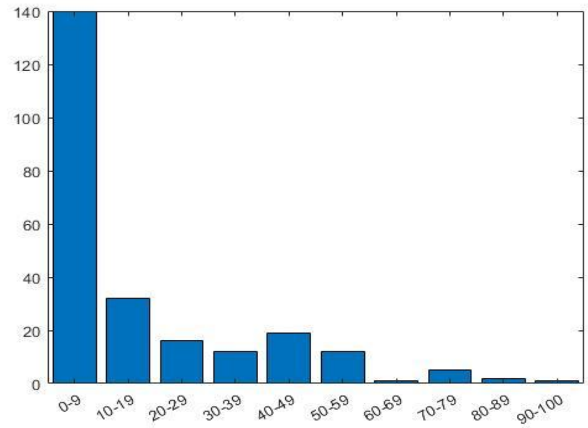


Figure 10. Model Power10’s record distribution

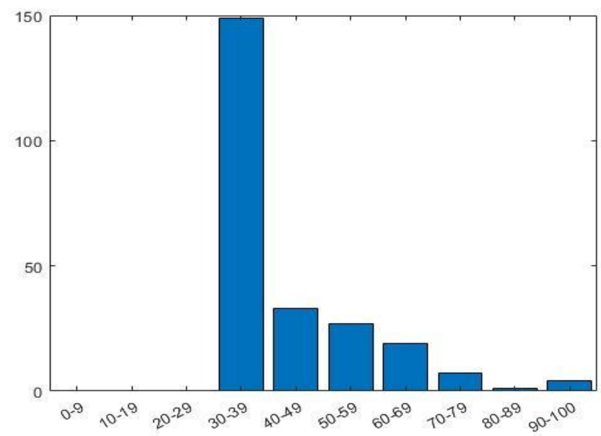


Figure 11. Model Multi25's record distribution

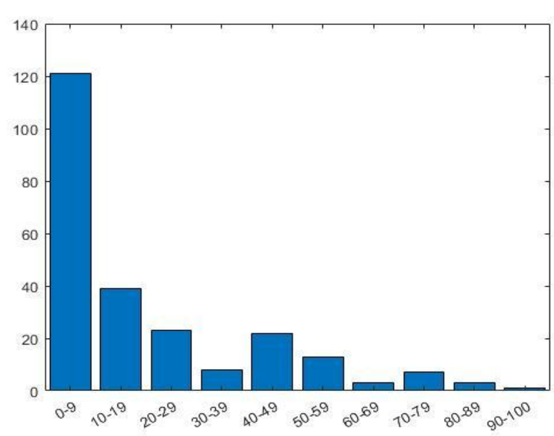


Figure 12. Model Power2's record distribution

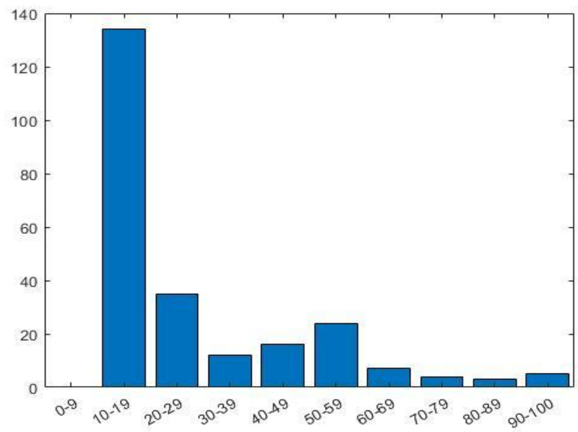


Figure 13.Model Cubic's record distribution

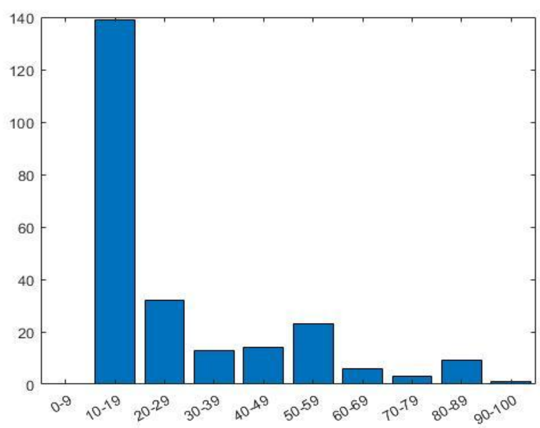


Figure 14. Model Exp's record distribution

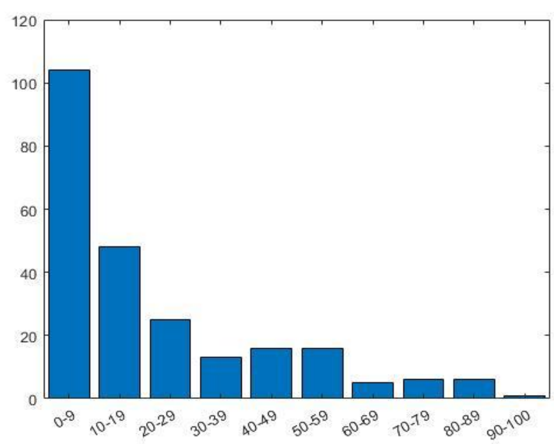


Figure 15. Model Sqrt's record distribution

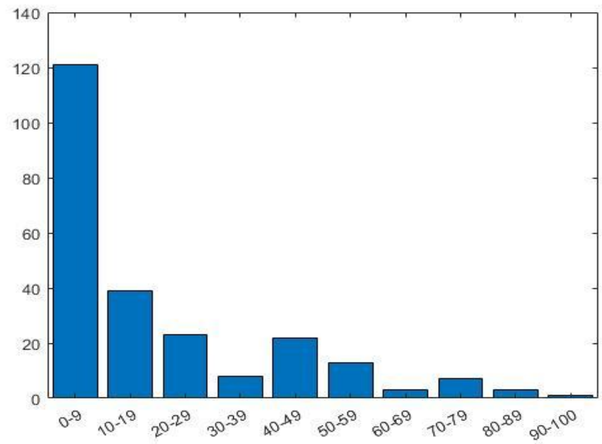


Figure 16. Model Fourpower's record distribution

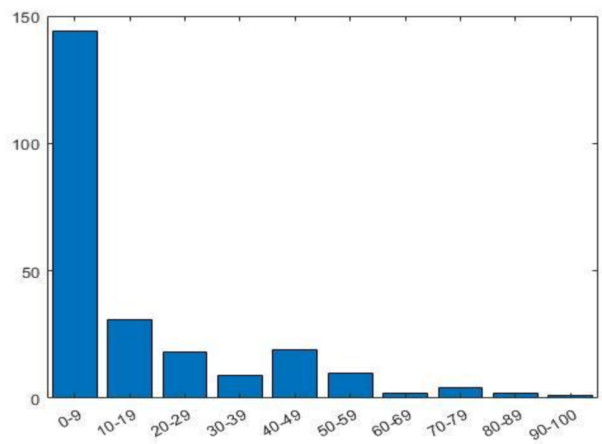


Figure 17. Model Powerten record distribution

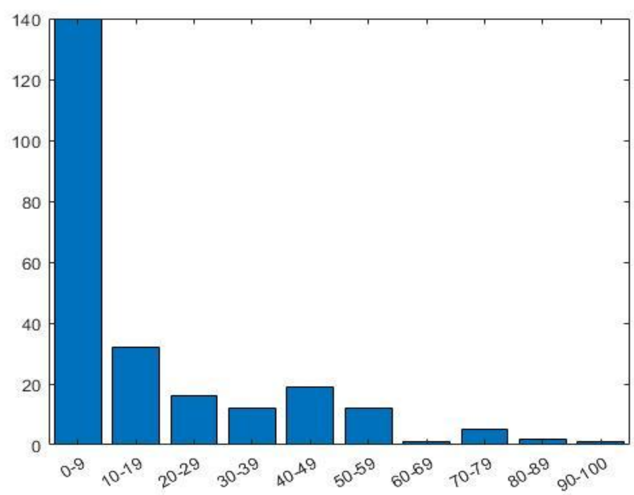


Figure 18. Model InverseTenth's record distribution

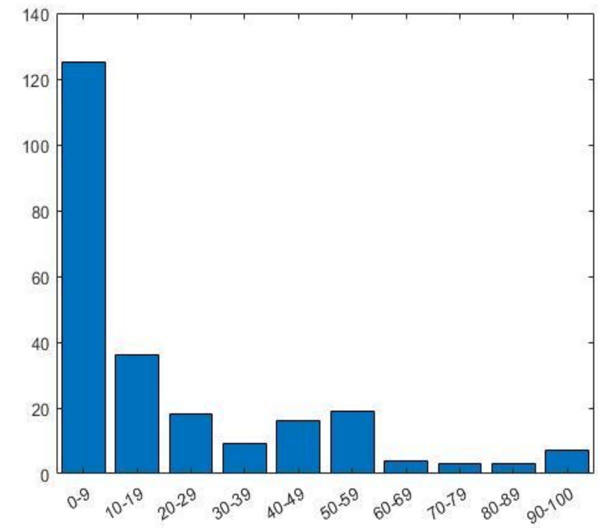


Figure 19. Model Tangent 's record distribution

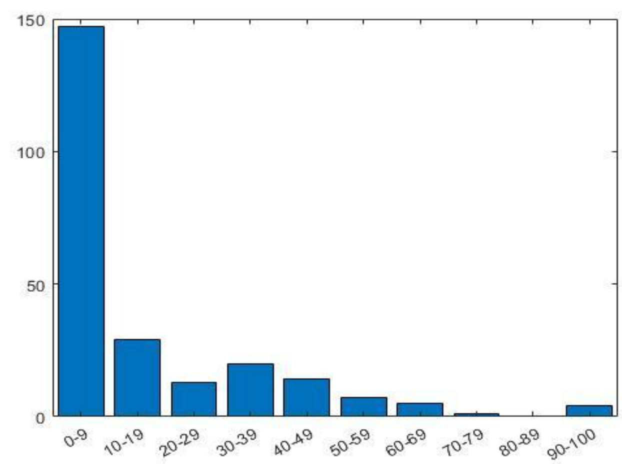


Figure 20. Model One_third's record distribution.

Figure 10-20 show record distribution of Model Power10, Multi25, Power2, Cubic, Exp, Sqrt, Fourpower, Powerten, InverseTenth, Tangent, One_third sequentially. Model Multi25 has no severity in the normal zone. Model Tangent has the highest number in very severe level but has a low number in severe level. On the other hand, model sqrt has the highest number in severe level. Other models have a higher number of records at a normal level. As none of the models is individually good at for alpha value, we average alpha value of model sqrt and tangent model to get a better distribution of record number.

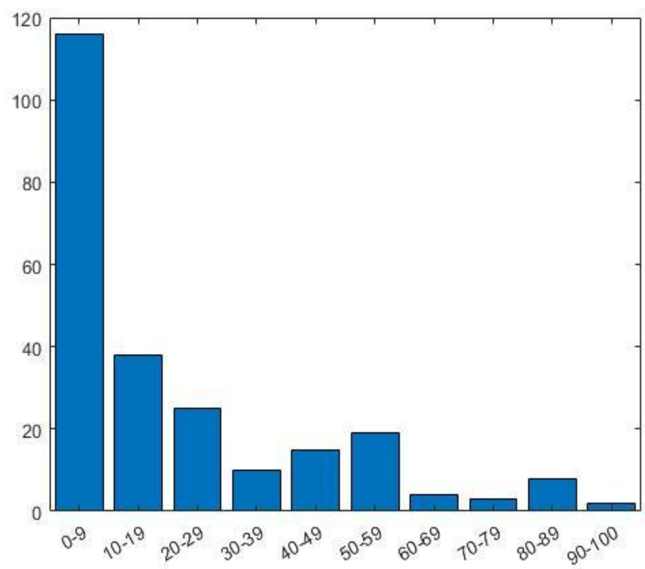


Figure 21. Distribution of record number of Model ‘Sqrt +Tangent’

From Figure 21, it has been shown that this model has a better distribution of segmented number of record than other models throughout 0 to 100 range.

So, our finalized Alpha value has been obtained from Model ‘Sqrt+Tangent’. This finalized Alpha values are used to weight the normal, fusion, supraventricular and ventricular beat of segmented part of the record. From the equation of severity measurement, EoI has been measured for every segmented part of record. Table 12 shows some segmented record with beats and their calculated EoI.

Table 12: Record with calculated EoI from Model 'sqrt+tan'

Record	**N**	**F**	**S**	**V**	**EoI**
100	494	0	4	1	0.050037
100	498	0	2	0	0.043042
100	489	0	11	0	0.0560395
100	488	0	12	0	0.0574835
100	269	0	4	0	0.0507335
101	500	0	0	0	0.0401545
101	500	0	0	0	0.0401545
101	500	0	0	0	0.0401545
101	365	0	0	0	0.0401545
103	500	0	0	0	0.0497725
103	500	0	0	0	0.0401545
103	492	0	0	6	0.0401545
103	495	0	0	3	0.0401545
103	83	0	0	0	0.0401545

F. Assignment of area_code to record

Area_code has been created manually. Records have been assigned to area_code programmatically so that every area_code has at least one record (one member).

Table 13. Area_code with assigned records

Area_code	**Record**
AZ19	106, 115
BY28	111,207, 212,215,222
CX38	101,213,234
CR63	124,220
DR27	203, 217,219
DW46	232
FD32	109, 113
HD93	208,228,230
KT43	103, 122
LC95	112,207,209
SP71	100,223
VJ14	107, 123
WG49	114,221
XP52	116,202,210,214
YU76B	118, 201
YU76K	119,233,200
YU76L	102, 105, 117

Table 13 continued: Area_code with assigned records

Area_code	**Record**
YU76Z	104,231
ZG86	108, 121

Table 13 shows area_code with their corresponding records. Only DW46 area_code has only one record but every other have multiple records.

G. DateTime Selection

Datetime selection to make the database for SCC-websever, we keep first date as 2018-06-01 01:04:25. Then a date duration is picked from 0 to 7 range randomly, add date duration to first date and create its consecutive date programmatically. By following this rule, a consecutive datelist has been whose last date is 2018-10-31 02:12:00.

Now to assign datetime to record from datelist, firstdate of an individual record has been chosen randomly. Then second date of that record is selected again randomly from the rest of date list after its firstdate, thirddate from the rest of date list after its second date and so on. When one record is completed with assigned datetime, another record's first date is chosen again randomly from complete date list except datetime of another record. Here every single datetime is unique. By this way, records' datetime has been created for database.

H. Making JSON for SCCmobileHealth webserver

After creating patient_id (record no), EOI, area_code, datetime, disease type (Arrhythmia (AR)), algorithm type (AR1), a JSON object has been created for each entry of database that is every member's diagnose test data of arrhythmia. Each JSON object has been combined in a JSON array and insert into the SCCmobileHealth webserver through POSTMAN. This POSTMAN inserts data into the database of the webserver through insertjsondb.php which is a php file of hosting server of our SCCmobileHealth.

Example of Json:

[{"DT":"AR","GRID_CODE":"SP71","TIME":"2018-06-01 01:04:25","ID":"100","ALG":"AR1","EOI":"0.014665"},

{"DT":"AR","GRID_CODE":"SP71","TIME":"2018-06-21 02:15:30","ID":"100","ALG":"AR1","EOI":"0.012471"},

{"DT":"AR","GRID_CODE":"SP71","TIME":"2018-06-22 05:08:23","ID":"100","ALG":"AR1","EOI":"0.015647"},

.

.

.

{"DT":"AR","GRID_CODE":"CX38","TIME":"2018-10-31 02:12:00","ID":"234","ALG":"AR1","EOI":"0.011765"}]

V. Visualization

To visualize health data, SCC web Server has two types of visualization, i.e. ‘Temporal plot’ with graph and flow, and another is ‘Spatial plot’ with static and animated view.

A. Temporal Plot

Participants can see EoI in the measurement of severity of persons over time. One is Time plot graph that shows severity in number and another is time plot flow that shows severity through color with the progression of time.

1) Time plot Graph

Time plot Graph shows EoI (0-1) in the Y-axis of the chart and the date and time in X-axis. This plotting has been done with the help of Google JavaScript API of Chart line. Every single line in the X-Y plane represents a single participant’s data. Data collection points are symbolized with a circle and two points are connected through edges to keep the continuity of participant’s data. To differentiate among different individual’s data, multi-colored lines have been exploited. If anyone hovers mouse on line, that line becomes highlighted. When the circles are hovered by mouse, a pop-up message is displayed with date and time of that participant’s testing and with their exact EoI value of that time. The line represents a participant’s gradual report of their health status from zero to one. To see small changes, one can scroll the mouse wheel over the expected area, and they can see the changes in a small granularity of hour. At first, the view of the time plot graph shows the default value. If there is no data in the database, then chart line becomes blank. If there is data, then it takes the oldest date as its first starting date and the last newest date would be the ending date of the chart line. Within these date range and with the default disease type (Flu for actual webserver and for ourselves is Arrhythmia), the time plot graph as a temporal plot of SCC Health are plotted.

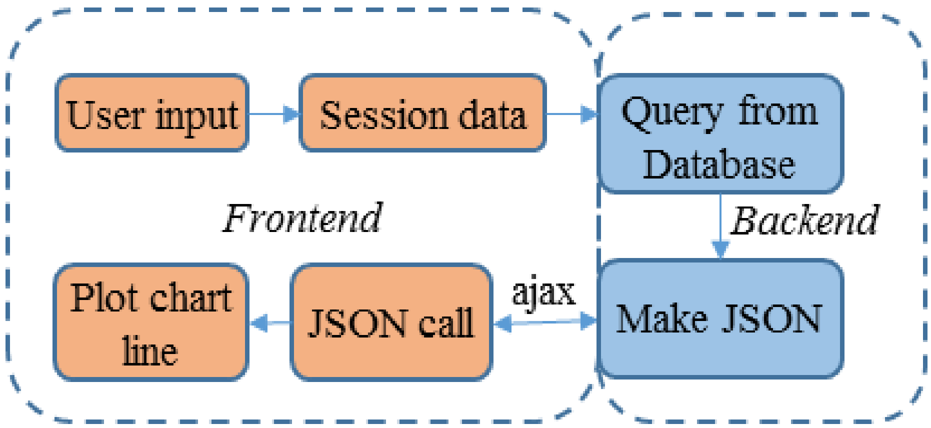


Figure 22. Mechanism of temporal plot at the SCC Health server.

Start date and end date are selected from the input box of calendar and disease type is selected from the option menu of disease. When user clicks the display button, session variables in the frontend PHP grabs the data and sends the data to the backend file. In the backend, with session variable and query of MySQL, data are retrieved from the database. Figure 22 represents mechanism of temporal plot at SCC Health server.

MySQL Query:

```
select patient,area_code,disease_type,datetime,eoi
          from MIT_BHI_Record
             where disease_type = '".$disease_type."'
               and ( datetime >= '".$start_datetime."' and datetime <=
'".$end_datetime."') order by datetime ASC
```

In this SQL query,

`MIT_BHI_Record` is the database table name `disease_type` represents the column of disease from the data table that is selected by the PHP variable `$disease_type`.

`$start_datetime` variable is the user defined firstdate of the datetime duration and

`$end_datetime` represents the last date of user defined date duration.

Then these data are encoded into a JSON array. This JOSN array consists of three JSON object and several JSON array. These three objects come chronologically, and they are startdate, enddate and disease type followed by other JSON array. The structure of JSON array is:

[Datetime, P_1, EOI_1, P_2, EOI_2, P_3, EOI_3 … P_n, EOI_n]

Where 'datetime' is a specific datetime from the database table. When the person P_1, P_2, P_3 … P_n upload their EOI value (EOI_1, EOI_2, EOI_3 … EOI_n respectively). The structure of main JSON array becomes

[startdate, enddate, disease_type, T_1, T_2, T_3, … T_n]

Where T represents sub JSON array with patients corresponding with their EOI value such that

T_1= [$Datetime_1$, P_{11}, EOI_{11}, P_{12}, EOI_{12}, P_{13}, EOI_{13} … P_{1n}, EOI_{1n}]

Every T in the main JSON array comes as an ascending order from startdate to enddate. If there is only one person's EOI in a specific datetime in the database then, only that person has value and other person and their EOI value will be null. The main JSON array is shown below:

[startdate, enddate, disease_type,

[$Datetime_1$, P_{11}, EOI_{11}, P_{12}, EOI_{12}, P_{13}, EOI_{13} … P_{1n}, EOI_{1n}],

[$Datetime_2$, P_{21}, EOI_{21}, P_{22}, EOI_{22}, P_{23}, EOI_{23} … P_{2n}, EOI_{2n}],

[$Datetime_3$, P_{31}, EOI_{31}, P_{32}, EOI_{32}, P_{33}, EOI_{33} … P_{3n}, EOI_{3n}],

.

.

.

[$Datetime_t$, P_{t1}, EOI_{t1}, P_{t2}, EOI_{t2}, P_{t3}, EOI_{t3} … P_{tn}, EOI_{tn}]]

Example of JSON array:

If the startdate is ‘2018-08-17’ and end date is ‘2018-08-20’ then SQL query gives us 10 rows from the above query. The query table is shown below:

Table 14. Query table for time plot Graph

patient	**area_code**	**disease_type**	**datetime**	**eoi**
219	DR27	AR	2018-08-17 01:07:17.000000	0.0560395
217	DR27	AR	2018-08-18 00:18:16.000000	0.11118
122	KT43	AR	2018-08-19 06:15:23.000000	0.0401545
207	BY28	AR	2018-08-19 12:03:24.000000	0.54142
106	AZ19	AR	2018-08-20 05:57:23.000000	0.248955

Table 14 continued: Query table for time plot Graph

patient	area_code	disease_type	datetime	eoi
219	DR27	AR	2018-08-20 06:08:18.000000	0.1003235
212	BY28	AR	2018-08-20 12:12:29.000000	0.261125
112	LC95	AR	2018-08-20 12:21:34.000000	0.0523925
124	CR63	AR	2018-08-20 19:24:17.000000	1

Now the backend php file reconstructs this table as a JSON array to fit into Google JavaScript ‘Line Chart’.

JSON array:

["2018-08-17 00:00:00","2018-08-20 23:59:59","AR",

[2018,7,17,1,7, 17, null, null, null, null, null, null, null,0.0560],

[2018,7,18,0,18,16, null, null, null, null, null, null, 0.1111, null, null],

[2018,7,19,6,15,23, null, null, 0.04015, null, null, null, null, null, null],

[2018,7,19,12,3,24, null, null, null, null, 0.54142, null, null, null, null],

[2018,7,20,5,57,23,0.2489, null, null, null, null, null, null, null, null],

[2018,7,20,6,8,18, null, null, null, null, null, null, null, 0.1003, null],

[2018,7,20,12,12,29, null, null, null, null, null, 0.26112, null, null, null],

[2018,7,20,12,21,34, null,0.0523, null, null, null, null, null, null, null],

[2018,7,20,19,24,17, null, null, null,1, null, null, null, null, null]]

Within date range from '2018-08-17' to '2018-08-20' there are nine distinct person record (i.e. 106, 112, 122, 124, 207, 217, 219, 212, 222). If every sub JSON array is considered as a row of a matrix then the first six columns (C1…C6) are represented by year, month, day, hour, minute and second. The rest of nine columns (C7…C15) are holding the EOI values from persons from 106 to 222 in an ascending order. If a datetime from this array such as 2018-08-19 06:15:23 is picked, then we find that only person 207 has uploaded his or her EOI value at that time. Similarly, person 106 has uploaded at 2018-08-20 05:57:23, person 219 has uploaded twice at 2018-08-17 01:07:17 and 2018-08-20 06:08:18, and so on.

In the frontend, this JSON array are called with the help of Ajax.

```
Code:

var jsonData = $.ajax({
               url:
"http://sccmobilehealth.com/research/admin/admintracker.php",
                dataType: "json",
                async: false
            }).responseText;
```

In the frontend JavaScript, JSON array has been parsed and customized the data for 'Line Chart'. Then with the help of the JavaScript google API key, line chart has been implemented. The work flow of time plot (graph) is given below:

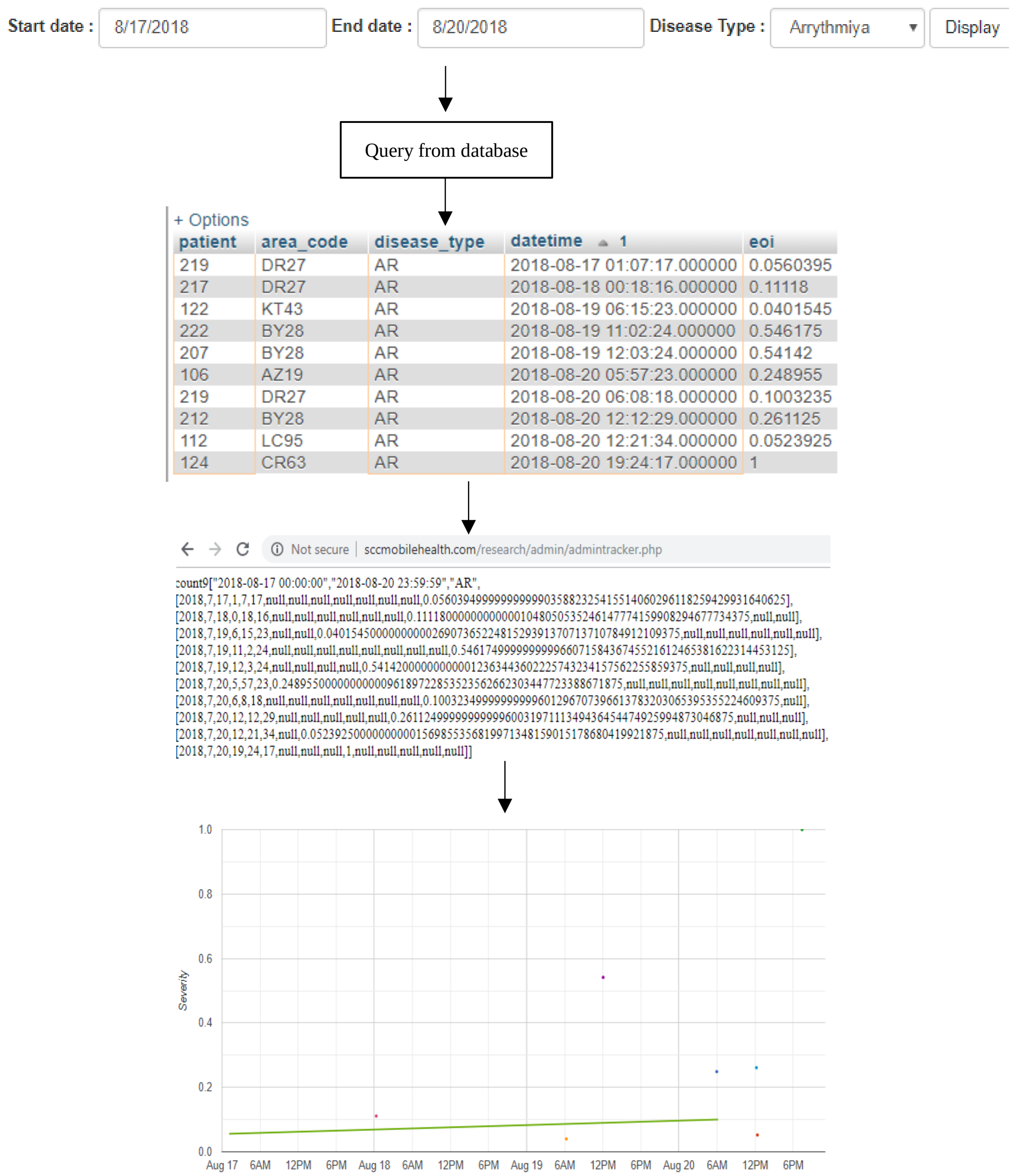


Figure 23. Workflow of timeplot (Graph)

2) Time plot Flow

Time Flow chart represents participant's EoI flow over time in color. X-axis represents time and Y-axis represents number of people. Every horizontal line in the x-y plane represents one person's data. EoI value has been represented by the small colored circle which represents the severity of that participant. EoI values from zero to one are distributed within 0 to 101 color value. Color is ranged from dark green to dark red where dark green represents the severity to low as 0 and dark red represents to high value of severity to 1. Every EoI value collected from database table with the user defined date range is multiplied by 100, retrieve integer value and keep it as an index of color_array.

Color_arrary:

```
[
"#009D00","#00A400","#00AB00","#00B200","#00B900","#00C000","#00C700","#00CE00","#00D500","#00DC00","#00E300"

,"#00EA00","#00F100","#00F800","#00FF00","#07FF00","#0EFF00","#15FF00","#1CFF00","#23FF00","#2AFF00"

,"#31FF00","#38FF00","#3FFF00","#46FF00","#4DFF00","#54FF00","#5BFF00","#62FF00","#69FF00","#70FF00"

,"#77FF00","#7EFF00","#85FF00","#8CFF00","#93FF00","#9AFF00","#A1FF00","#A8FF00","#AFFF00","#B6FF00"

,"#BDFF00","#C4FF00","#CBFF00","#D2FF00","#D9FF00","#E0FF00","#E7FF00","#EEFF00","#F5FF00","#FFFF00"

,"#FFF800","#FFF100","#FFEA00","#FFE300","#FFDC00","#FFD500","#FFCE00","#FFC700","#FFC000","#FFB900"

,"#FFB200","#FFAB00","#FFA400","#FF9D00","#FF9600","#FF8F00","#FF8800","#FF8100","#FF7A00","#FF7300"

,"#FF6C00","#FF6500","#FF5E00","#FF5700","#FF5000","#FF4900","#FF4200","#FF3B00","#FF3400","#FF2D00"

,"#FF2600","#FF1F00","#FF1800","#FF1100","#FF0A00","#FF0000","#F80000","#F10000","#EA0000","#E30000"

,"#DC0000","#D50000","#CE0000","#C70000","#C00000","#B90000","#B20000","#AB0000","#A40000","#9D0000"
]
```

#009D00 #00A400 #00AB00 #00B200 #00B900 #00C000 #00C700
#00CE00 #00D500 #00DC00 #00E300 #00EA00 #00F100 #00F800
#00FF00 #07FF00 #0EFF00 #15FF00 #1CFF00 #23FF00 #2AFF00
#31FF00 #38FF00 #3FFF00 #46FF00 #4DFF00 #54FF00 #5BFF00
#62FF00 #69FF00 #70FF00 #77FF00 #7EFF00 #85FF00 #8CFF00
#93FF00 #9AFF00 #A1FF00 #A8FF00 #AFFF00 #B6FF00 #BDFF00
#C4FF00 #CBFF00 #D2FF00 #D9FF00 #E0FF00 #E7FF00
#EEFF00 #F5FF00 #FFFF00 #FFF800 #FFF100 #FFEA00 #FFE300
#FFDC00 #FFD500 #FFCE00 #FFC700 #FFC000 #FFB900
#FFB200 #FFAB00 #FFA400 #FF9D00 #FF9600 #FF8F00 #FF8800
#FF8100 #FF7A00 #FF7300 #FF6C00 #FF6500 #FF5E00 #FF5700
#FF5000 #FF4900 #FF4200 #FF3B00 #FF3400 #FF2D00 #FF2600
#FF1F00 #FF1800 #FF1100 #FF0A00 #FF0000 #F80000 #F10000
#EA0000 #E30000 #DC0000 #D50000 #CE0000 #C70000 #C00000
#B90000 #B20000 #AB0000 #A40000 #9D0000

Figure 24. Color code

From Color_code array and from Figure 24, it has been shown that yellow (#FFFF00) is middle color and green (#00FF00) as first half middle color and red (#FF0000) as last half middle color.

This Flow chart is made with google ‘Line chart’. The mechanism of retrieving data from database as like the Time plot Graph. The same Google ‘Line Chart’ has been used time plot flow. The differences with graph plot are EoI value is converted in color and the customization of JSON for the line chart.

The structure of JSON array is

[StartDate, EndDate, Disease_type

[Datetime$_1$, P_{Num11}, Color$_{11}$, P_{Num12}, Color$_{12}$, P_{Num13}, Color$_{13}$ … P_{Num1n}, Color$_{1n}$]

[Datetime$_2$, P_{Num21}, Color$_{21}$, P_{Num22}, Color$_{22}$, P_{Num23}, Color$_{23}$ … P_{Num2n}, Color$_{2n}$]

[$Datetime_3$, P_{Num11}, $Color_{31}$, P_{Num32}, $Color_{32}$, P_{Num33}, $Color_{33}$ … P_{Num1n}, $Color_{3n}$]

.

.

.

[$Datetime_n$, P_{Numn1}, $Color_{n1}$, P_{Numn2}, $Color_{n2}$, P_{Numn3}, $Color1_{n3}$ … P_{Numnn}, $Color_{nn}$]]

Here StartDate, Enddate and 'Disease_type' are user's input given by the web server. 'Datetime' of each JSON object is date and time when people do their tests. P_{Num} is neither actual patient number but chronological number of people that comes in database.

SQL query:

```
"select patient,area_code,disease_type,datetime,eoi
          from TestPatient
             where disease_type = '".$disease_type."'
               and ( datetime >= '".$start_datetime."'
               and datetime <= '".$end_datetime."')
               and patient not LIKE 'T%' order by datetime ASC";
```

From the SQL query patient with their corresponding EoI value is retrieved from the database. This EoI value is then multiplied by 100 and only integer value from the multiplied 100 EoI is taken into consideration. These integer EoI values are the index of color code array. After retrieving the color from the color_code_array it is used in the JSON array of chart line.

For example, if user gives input for 'Start date' is '8-11-2018' and End Date is '8-15-2018' and disease type is 'AR' then from MySQL database table looks like:

Edit	Copy	Delete	29	YU76L	AR	0.65033	2018-08-11 03:23:05.000000	105	AR1
Edit	Copy	Delete	223	HD93	AR	0.0401545	2018-08-11 12:23:22.000000	230	AR1
Edit	Copy	Delete	24	YU76Z	AR	0.101701	2018-08-12 01:56:15.000000	104	AR1
Edit	Copy	Delete	169	CX38	AR	0.284055	2018-08-12 04:45:25.000000	213	AR1
Edit	Copy	Delete	182	BY28	AR	0.16275	2018-08-12 17:23:16.000000	215	AR1
Edit	Copy	Delete	237	YU76K	AR	0.0749205	2018-08-13 01:24:04.000000	233	AR1
Edit	Copy	Delete	44	ZG86	AR	0.052414	2018-08-13 04:12:34.000000	108	AR1
Edit	Copy	Delete	170	CX38	AR	0.28056	2018-08-13 06:23:28.000000	213	AR1
Edit	Copy	Delete	187	DR27	AR	0.11991	2018-08-13 09:25:11.000000	217	AR1
Edit	Copy	Delete	14	YU76L	AR	0.0401545	2018-08-14 06:45:15.000000	102	AR1
Edit	Copy	Delete	55	BY28	AR	0.0401545	2018-08-14 16:00:34.000000	111	AR1
Edit	Copy	Delete	19	KT43	AR	0.0401545	2018-08-14 21:33:25.000000	103	AR1
Edit	Copy	Delete	224	HD93	AR	0.910665	2018-08-15 05:34:24.000000	230	AR1
Edit	Copy	Delete	127	DR27	AR	0.0579445	2018-08-15 08:08:24.000000	203	AR1

Figure 25. Screenshot from database of SCCmobileHealth webserver.

Now after running SQL query, it retives the following data table.

Table 15. Data table from time plot (Flow) SQL query

Patient	Area_code	Disease_type	DateTime	EoI
105	YU76L	AR	2018-08-11 03:23:05.000000	0.65033
230	HD93	AR	2018-08-11 12:23:22.000000	0.0401545
104	YU76Z	AR	2018-08-12 01:56:15.000000	0.101701
213	CX38	AR	2018-08-12 04:45:25.000000	0.284055
215	BY28	AR	2018-08-12 17:23:16.000000	0.16275
233	YU76K	AR	2018-08-13 01:24:04.000000	0.0749205
108	ZG86	AR	2018-08-13 04:12:34.000000	0.052414

Table 15 continued: Data table from time plot (Flow) SQL query

Patient	Area_code	Disease_type	DateTime	EoI
213	CX38	AR	2018-08-13 06:23:28.000000	0.28056
217	DR27	AR	2018-08-13 09:25:11.000000	0.11991
102	YU76L	AR	2018-08-14 06:45:15.000000	0.0401545
111	BY28	AR	2018-08-14 16:00:34.000000	0.0401545
103	KT43	AR	2018-08-14 21:33:25.000000	0.0401545
230	HD93	AR	2018-08-15 05:34:24.000000	0.910665
203	DR27	AR	2018-08-15 08:08:24.000000	0.0579445

From the retrieved data table ‘patient’, ‘datetime’, ‘eoi’ column are taken into consideration. This EOI is multiplied by 100 to get the index of color_code from table 16.

Table 16. Making color from EOI for time plot (Flow)

patient	datetime	eoi	eoi*100	Index of color_code _array	Color_code	color
105	2018-08-11 03:23:05	0.65033	65.033	65	FF9600	

Table 16 continued: Making color from EOI for time plot (Flow)

patient	datetime	eoi	eoi*100	Index of color_code _array	Color_code	color
230	2018-08-11 12:23:22	0.04015	04.015	4	00B900	
104	2018-08-12 01:56:15	0.10170	10.170	10	00E300	
213	2018-08-12 04:45:25	0.28405	28.405	28	62FF00	
215	2018-08-12 17:23:16	0.16275	16.275	16	0EFF00	
233	2018-08-13 01:24:04	0.07492	07.492	7	00CE00	
108	2018-08-13 04:12:34	0.05241	05.241	5	00C000	
213	2018-08-13 06:23:28	0.28056	28.056	28	62FF00	
217	2018-08-13 09:25:11	0.11991	11.991	11	00EA00	

Table 16 continued: Making color from EOI for time plot (Flow)

patient	datetime	eoi	eoi*100	Index of color_code _array	Color_code	color
102	2018-08-14 06:45:15	0.04015	04.015	4	00B900	
111	2018-08-14 16:00:34	0.04015	04.015	4	00B900	
103	2018-08-14 21:33:25	0.04015	04.015	4	00B900	
230	2018-08-15 05:34:24	0.910665	91.0665	91	DC0000	
203	2018-08-15 08:08:24	0.0579445	05.79445	5	00C000	

After calculation of color, these color_code are bound with JSON array. Now the JOSN array is looks like:

["2018-08-11","2018-08-15","AR",

[2018,7,11,3,23,5,null,null,null,null,null,null,4,"#FF9600",null,null,null,null,null,null,null,null,null,null,null,null,null,null],

[2018,7,11,12,23,22,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,10,"#00B900",null,null],

[2018,7,12,1,56,15,null,null,null,null,3,"#00E300",null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null],

[2018,7,12,4,45,25,null,null,null,null,null,null,null,null,null,null,null,null,7,"#62FF00",null,null,null,null,null,null,null,null],

[2018,7,12,17,23,16,null,null,null,null,null,null,null,null,null,null,null,null,null,null,8,"#0EFF00",null,null,null,null,null,null],

[2018,7,13,1,24,4,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,11,"#00CE00"],

[2018,7,13,4,12,34,null,null,null,null,null,null,null,null,5,"#00C000",null,null,null,null,null,null,null,null,null,null,null,null],

[2018,7,13,6,23,28,null,null,null,null,null,null,null,null,null,null,null,null,7,"#62FF00",null,null,null,null,null,null,null,null],

[2018,7,13,9,25,11,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,9,"#00EA00",null,null,null,null],

[2018,7,14,6,45,15,1,"#00B900",null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null],

[2018,7,14,16,0,34,null,null,null,null,null,null,null,null,null,null,6,"#00B900",null,null,null,null,null,null,null,null,null,null],

[2018,7,14,21,33,25,null,null,2,"#00B900",null,null,null,null,null,null,null,null,null,null, null,null,null,null,null,null,null,null]]

Here patients are 12 in number and JSON Objects is 14 in number. Only two persons upload their EoI value twice times between 11th Aug, 2018 to 15th Aug, 2018. This JSON Array is called by Ajax call and parsed into front end JavaScript and display the time plot of flow.

B. Spatial Plot (Google Map overlay)

1) Static Map View

The surroundings of our community in google map are divided into 16 grids. The most populated area is again divided into four quads. The grids are assigned with arbitrarily generated area code and each grid has their own rectangle defined by latitude and longitude. The area code is assigned to each participant based upon their home address. All home inside the same grid has unique area code (See Figure 3 in chapter III)

Participants share their EoI values with SCC Health server and EoI of the participant is averaged with other participants' EoI value who live in the same area code as

defined in (1). This adds a layer of privacy, as it is not possible from the visualization, or from the server data to detect any participant's exact home address.

$$EoI_{av}{}^{i} = \left(\sum_{u=1 \; to \; N}\left(\sum_{j=1 \; to \; M} EoI^{j}\right)/\mathrm{M}\right)/N$$

In equation,

i is the grid number,

u is the participant's number,

N is the maximum number of participants in that i'th grid who submit EoI for that particular disease and that particular time and

j is a participants who submit multiple measurements

within that timeframe with last submission as M

Participant can select date interval (i.e. start date and end date) and gives a choice of disease type. After clicking the display button, backend file grabs the data of date interval and disease type. A MySQL query are written to retrieve the EoI value, area code, and participant's ID for the selected date interval and disease type. EoI value is first averaged with the same persons and then this average value is again averaged with the number of participants who live in the same area code.

SQL Query:

```
"select avg(sub.avgeoi) as areaavgeoi, sub.area_code as areacode from
                (select patient,area_code, avg(eoi) as avgeoi
                        from TestPatient
                                where disease_type = '".$disease_type."'
                                and ( datetime >= '".$start_datetime."'
and datetime <= '".$end_datetime."') and patient not LIKE 'T%'
                                group by patient) sub
                                group by sub.area_code";
```

Example of Static Map View:

If user selects start_datetime as '2018-08-11' and end_datetime as '2018-08-22' and disease_type as 'AR', then at first SQL query selects following rows (Table 17) from database table.

Table 17. Table from mapview SQL query

Patient	area_code	disease_type	datetime	eoi
105	YU76L	AR	2018-08-11 03:23:05.000000	0.65033
230	HD93	AR	2018-08-11 12:23:22.000000	0.0401545
104	YU76Z	AR	2018-08-12 01:56:15.000000	0.101701
213	CX38	AR	2018-08-12 04:45:25.000000	0.284055
215	BY28	AR	2018-08-12 17:23:16.000000	0.16275
233	YU76K	AR	2018-08-13 01:24:04.000000	0.0749205
108	ZG86	AR	2018-08-13 04:12:34.000000	0.052414
213	CX38	AR	2018-08-13 06:23:28.000000	0.28056
217	DR27	AR	2018-08-13 09:25:11.000000	0.11991
102	YU76L	AR	2018-08-14 06:45:15.000000	0.0401545
111	BY28	AR	2018-08-14 16:00:34.000000	0.0401545
103	KT43	AR	2018-08-14 21:33:25.000000	0.0401545
230	HD93	AR	2018-08-15 05:34:24.000000	0.910665
203	DR27	AR	2018-08-15 08:08:24.000000	0.0579445
210	XP52	AR	2018-08-16 08:22:18.000000	0.37668
219	DR27	AR	2018-08-17 01:07:17.000000	0.0560395

Table 17 continued: Table from mapview SQL query

Patient	area_code	disease_type	datetime	eoi
217	DR27	AR	2018-08-18 00:18:16.000000	0.11118
122	KT43	AR	2018-08-19 06:15:23.000000	0.0401545
222	BY28	AR	2018-08-19 11:02:24.000000	0.546175
207	BY28	AR	2018-08-19 12:03:24.000000	0.54142
106	AZ19	AR	2018-08-20 05:57:23.000000	0.248955
219	DR27	AR	2018-08-20 06:08:18.000000	0.1003235
212	BY28	AR	2018-08-20 12:12:29.000000	0.261125
112	LC95	AR	2018-08-20 12:21:34.000000	0.0523925
124	CR63	AR	2018-08-20 19:24:17.000000	1

From Table 17 SQL query makes average of EoI value of same patient and then again makes average of EoI value according to area_code.

Table 18. Making average of EOI with their corresponding Area_code

<table>
<tr><th>Patient</th><th>EoI</th><th>Avg EoI</th><th>Area_code</th><th>Avg_EoI</th></tr>
<tr><td>230</td><td>0.0401545</td><td rowspan="2">0.47540975</td><td rowspan="2">HD93</td><td rowspan="2">0.47540975</td></tr>
<tr><td>230</td><td>0.910665</td></tr>
</table>

Table 18 continued: Making average of EOI with their corresponding

Area_code

Patient	area_code	disease_type	datetime	eoi
104	0.101701	0.101701	YU76Z	0.101701
213	0.284055	0.2823075	CX38	0.2823075
213	0.28056			
233	0.0749205	0.0749205	YU76K	0.0749205
108	0.052414	0.052414	ZG86	0.052414
105	0.65033	0.65033	YU76L	0.34524225
102	0.0401545	0.0401545		
215	0.16275	0.16275	BY28	0.3103249
111	0.0401545	0.0401545		
222	0.546175	0.546175		
207	0.54142	0.54142		
212	0.261125	0.261125		
103	0.0401545	0.0401545	KT43	0.0401545
122	0.0401545	0.0401545		

Table 18 continued: Making average of EOI with their corresponding Area_code

Patient	**area_code**	**disease_type**	**datetime**	**eoi**
217	0.11991	0.115545	DR27	0.08389
217	0.11118			
203	0.0579445	0.0579445		
219	0.0560395	0.0781815		
219	0.1003235			
210	0.37668	0.37668	XP52	0.37668
106	0.248955	0.248955	AZ19	0.248955
112	0.0523925	0.0523925	LC95	0.0523925
124	1	1	CR63	1

Table 18 shows how SQL query make average according to patients and area_code. This average value is then multiplied with 100 and then keeps only integer value that is the index of color code array. The color code is ranged from 0 to 100.

Table 19. Table of making color code from avg EOI for map view

Area_code	**Avg_EoI**	**Avg_EoI*100**	**Index**	**Color code**	**Color**
HD93	0.47540975	47.540975	47	E7FF00	

Table 19 continued: Table of making color code from avg EOI for map view

Area_code	**Avg_EoI**	**Avg_EoI*100**	**Index**	**Color code**	**Color**
YU76Z	0.101701	10.1701	10	00E300	
CX38	0.2823075	28.23075	28	62FF00	
YU76K	0.0749205	07.49205	7	00CE00	
ZG86	0.052414	05.2414	5	00C000	
YU76L	0.34524225	34.524225	34	8CFF00	
BY28	0.3103249	31.03249	31	77FF00	
KT43	0.0401545	04.01545	4	00B900	
DR27	0.08389	08.389	8	00D500	
XP52	0.37668	37.668	37	A1FF00	
AZ19	0.248955	24.8955	24	46FF00	
LC95	0.0523925	5.23925	5	00C000	
CR63	1	100.00	100	9D0000	

Hence, every EoI becomes the index of color code and this color code indicates the color of that area code. Table 19 shows how average EoI represents color of grid area.

In the frontend of static mapview, GeoJSON has a featured based property. This property can be modified by color and area code. The backend file makes a JSON bound with area_code and property (Example of GeoJSON is given in appendix). This backend JSON file is named as GeoJSON and this JSON is loaded in frontend as GeoJSON link in loadGeoJson function argument. In the front end this GeoJSON link has been used to change the color of each area code and thus google map overlay represents color of grid code.

GeoJSON:

```
map.data.loadGeoJson('http://sccmobilehealth.com/research/admin/adminjs
onfromdbMap.php');

        map.data.setStyle(function(feature) {
          return ({
            fillColor: feature.getProperty('color'),
            strokeWeight: 1,

            strokeOpacity: 0.8,
            fillOpacity:feature.getProperty('fillOpacity'),
            font: '12px Verdana',
           text: feature.getProperty('areacode'),


          });
        });
```

Here, in GeoJOSN, property of fillcolor (‘fillColor: feature.getProperty('color')’) is responsible to change the color of mapview. Other properties of GeoJSON are kept unchanged.

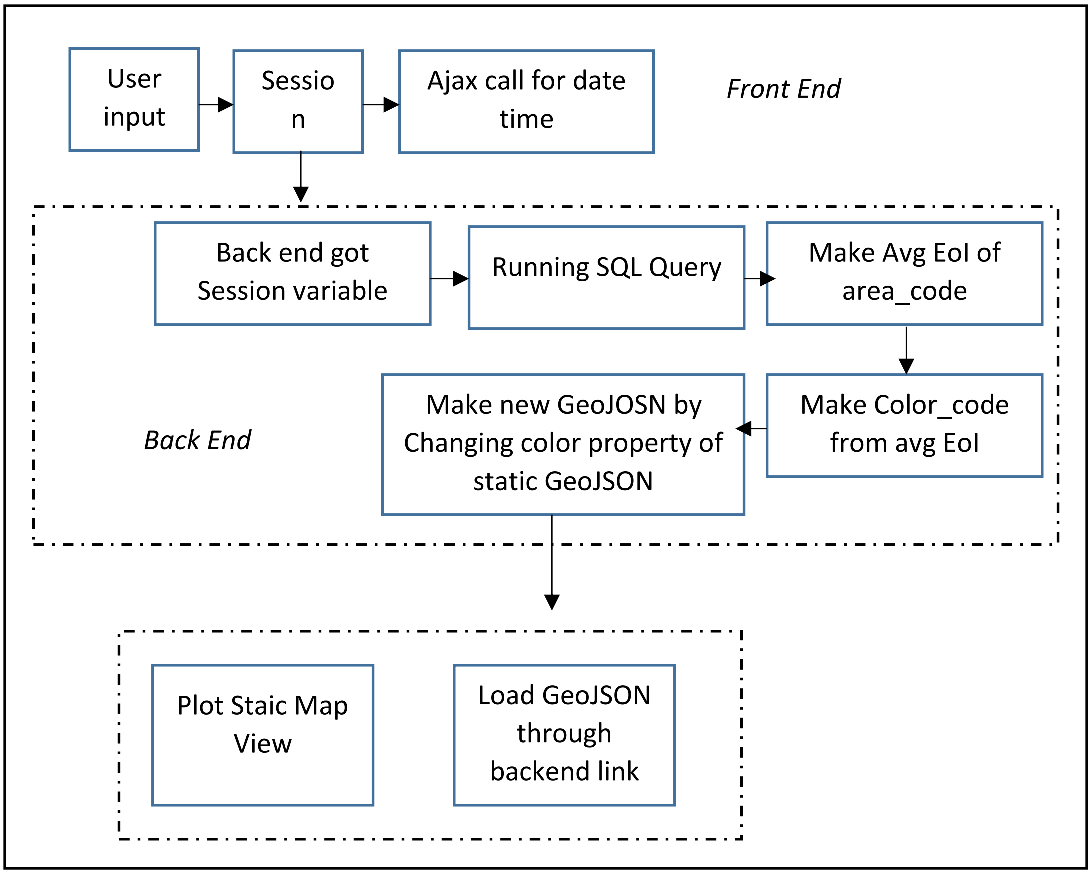


Figure 26. Visualization flow of Static Map View

adminSimpleJsonMapFinal.php is the admin frontend php file and adminjsonfromdbMap.php is the admin backend php file in SCCHealth. web server. Figure 26 shows visualization flow of static map view that helps to plot static map view.

2) Animation of Spatial Plot

To view the result of spatial plot day by day, animation on the google map are taken into consideration.

To visualize animation, one should set following properties,

Startdate- Date from where user want to start animation (default date is the first date of MIT_BHI _record database)

Enddate- Date where animation will stop (Default date is the last date of MIT_BHI_record database)

Disease type- Arrhythmia, Flu, COPD, Sleep Apnea (here always Arrhythmia),

Frame interval- How fast user wants to see the animation (0.1s(default),0.5s,1s,2s)

Date interval- how many days make bin for animation (1(default),2,3…10)

Animation type- cumulative and segmented

When user customizes his/her choice from the properties, s/he has to click the button named as ‘set’ to set all the properties. To see the animation, one should click the button named as ‘Animate’. If someone wants to stop the animation at a certain point, s/he has to click the button named as ‘stop’.

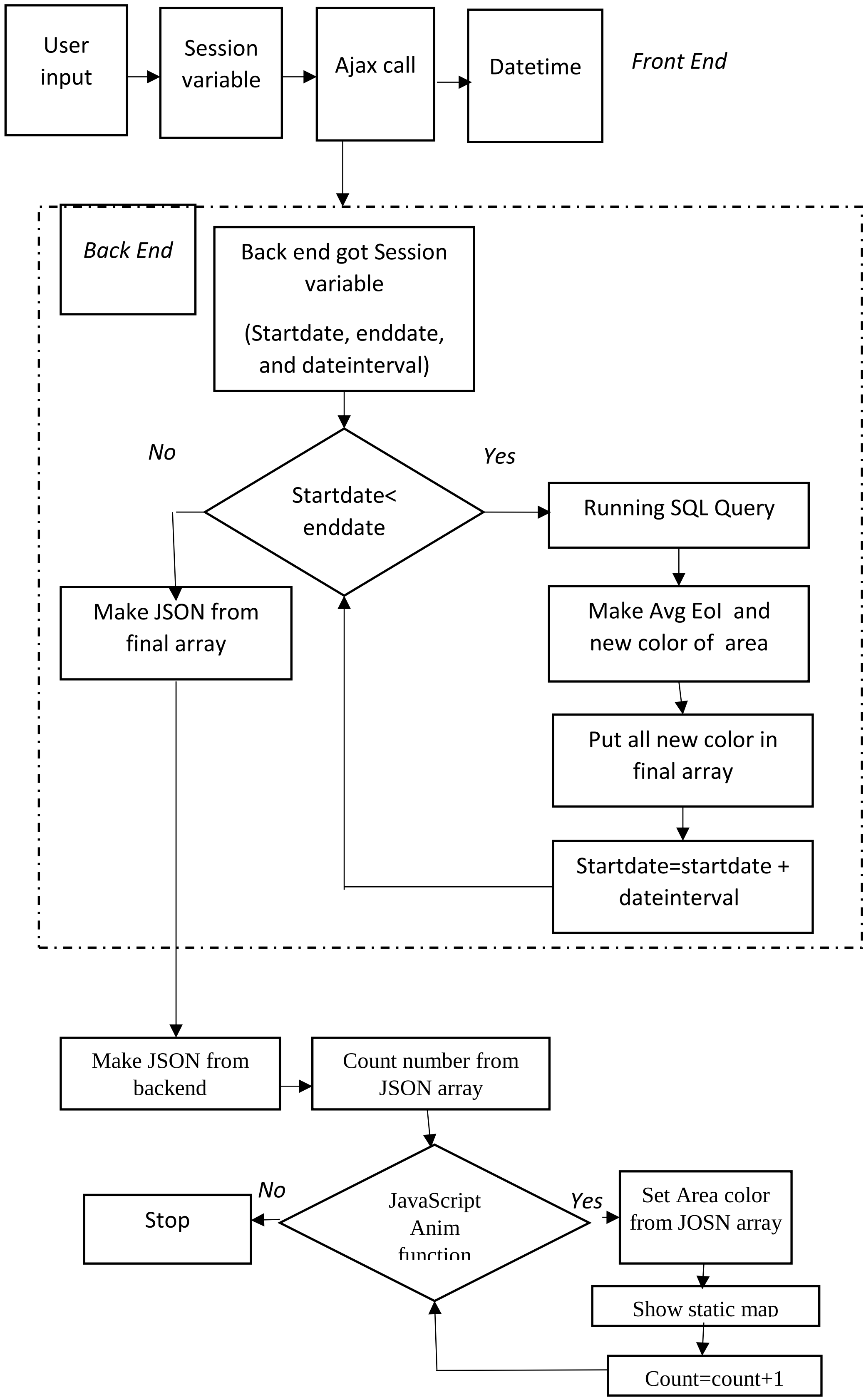


Figure 27.Block Diagram of Animation of Map view

For the animation of map view, user gives their choice through property choice. Then with the help of session variable backend server got the session value. In backend, SQL query makes JSON array by increasing date with date interval given by user. This json consists of color of areacode of each single date or bin of date. After completing JOSN array, it has been called by ajax from frontend. In the frontend this JSON array is parsed and helps JavaScript animation function to render the animation though each date or bin of dates' color of area_code. Thus the animation of map view works. Figure 27 shows how animation of map view works.

There are two types of animation, i.e.

i. Cumulative and

ii. Segmented day-based animation.

The cumulative animation represents the animation from the starting day and then add day depends upon the choice of user. In the default set up, each frame of animation adds next day EoI value and averages them and displays the colored grid on that date slot till end day. In this animation, every single day or the bin of segmented days' EoI values are averaged and represented in the animation.

VI. Results

The web visualization tool has been developed over the last year and deployed (http://sccmobilehealth.com/research/index.php). All functionalities have been implemented and tested. The response time of the first page of the SCC Health server is 0.267 sec. and the total code size is 455 KB.

The sensor data are properly visible through temporal and spatial plot. The guest login allows anyone to see the functionalities and uses mock data for every disease type. In guest login, these data sets can be fully visualized through temporal and spatial plot. The data collected from deployment will be stored in a separate database table named as MIT_BHI_Record. These data sets will be collected from recruited subjects and sensor data collected from the various sensors (e.g. core body temperature). But in this thesis, physiological datasets are used from MIT-BHI database for validation of functionalities. Each record of this database is considered as a member of community. From this database, we measured severity of each record of the database. The recruited subjects and admins will have user name and passwords to observe these data in actual webserver with different privileges.

A. *JSON from SCCHealth App*

JSON collected from mobile app 'SCCHealth' are parsed properly in php backend server. The code for parsing data in backend server is given in Appendix A.

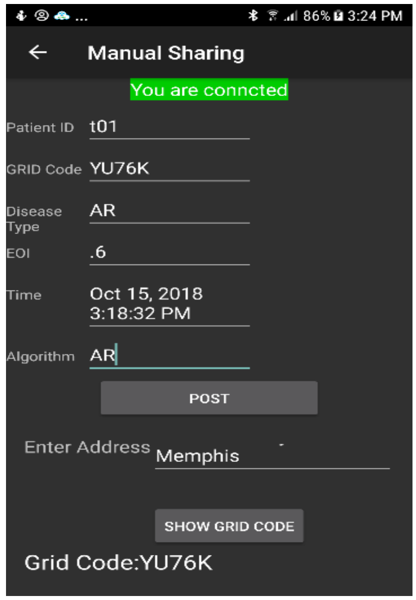


Figure 28. Manual Sharing of EoI value of disease severity from SCCHealth mobile App

{"ID":"t01","GRID_CODE":"YU76K","DT":"AR","EOI":".6","TIME":"Oct 15, 2018 3:18:32 PM","ALG":"AR"}

Figure 29. JOSN of manual sharing EoI of disease severity from console of SCCHealth app

id	area_code	disease_type	eoi	datetime	patient	algorithm
148	YU76K	AR	.6	2018-10-15 15:18:32	t01	AR

Figure 30. SCCHealth web server database

In Figure 28, there is a screenshot of manual sharing of EoI value from SCCHealth app. In this app, it provides all information that is ID is to 1, grid_code is YU76K, Disease type AR, EoI value is 0.6, and time is Oct 15, 2018. When user press button 'POST' on the smartphone app, all information bound in JSON format shown in Figure 29 are sent to the SCC web server. In the web server database, all information is properly columned, as shown in Figure 30, and has similarities with JSON of mobile app.

It has been manually verified that JSON information are properly sent from app for representative data, parsed in webserver, and stored in database for the visualization.

B. Time plot Graph

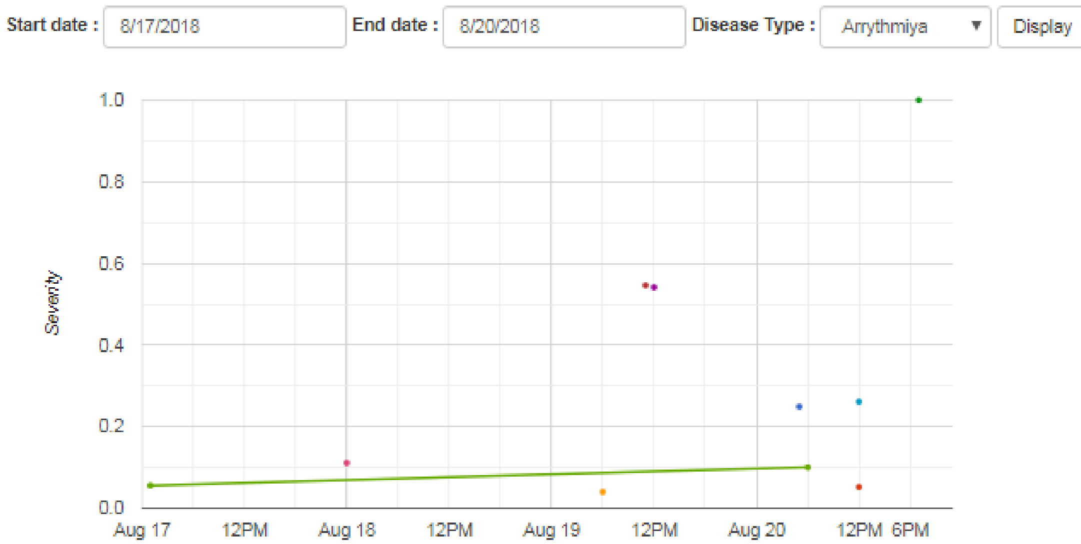


Figure 31. TimePlotFlow with startdate 12-08-2018 and enddate 20-08-2018

From Figure 31, it is shown that only one person has uploaded twice EoI at 2018-08-17 01:07:17 and 2018-08-20 06:08:18 within date range. So, there is only one line shown in the Time Plot Graph. From the JSON of Time plot flow in visualization section it has been mentioned that one person has EoI that is 1. In chart line, one circle has been shown at severity 1. So, it represents that chart line for time plot Graph works properly.

The pop up message that comes when a user hover a EOI valued circle. This pop message shown in the Figure 32 and Figure 33 contains date and time in first row and second row with an arbitary person's number and their corresponding EOI value. This persons number is coming from the row order of json array. The actual person id is not shown here to preserve privacy.

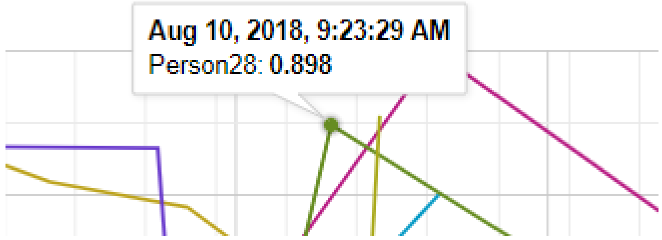


Figure 32. Pop-up message in time plot (Graph) (Zoom version)

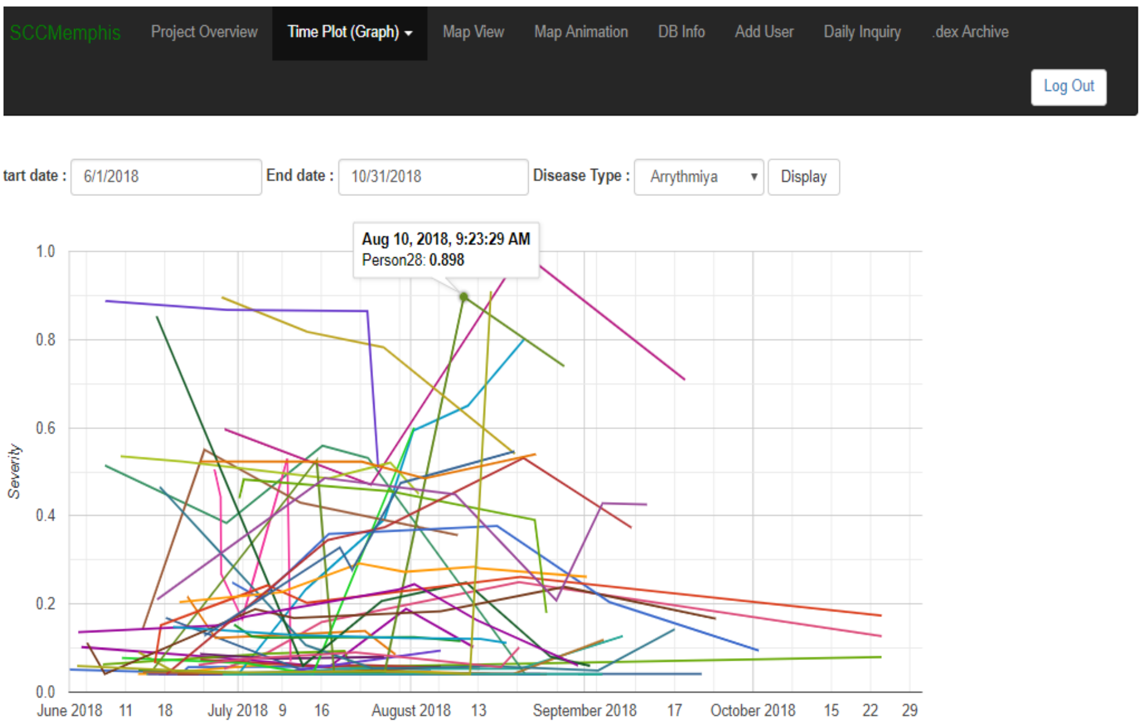


Figure 33. Pop-up message with time plot Graph

The property of 'Line chart' for the Time plot of Graph is given by

```
explorer: {maxZoomIn: .2,
           maxZoomOut: 8,
           zoomDelta:1.5,
           keepInBounds: true}
```

Figure 34 shows zoom out version and Figure 35 shows the zoom in version. The zoom in version helps user to see the EOI value in small granularity.

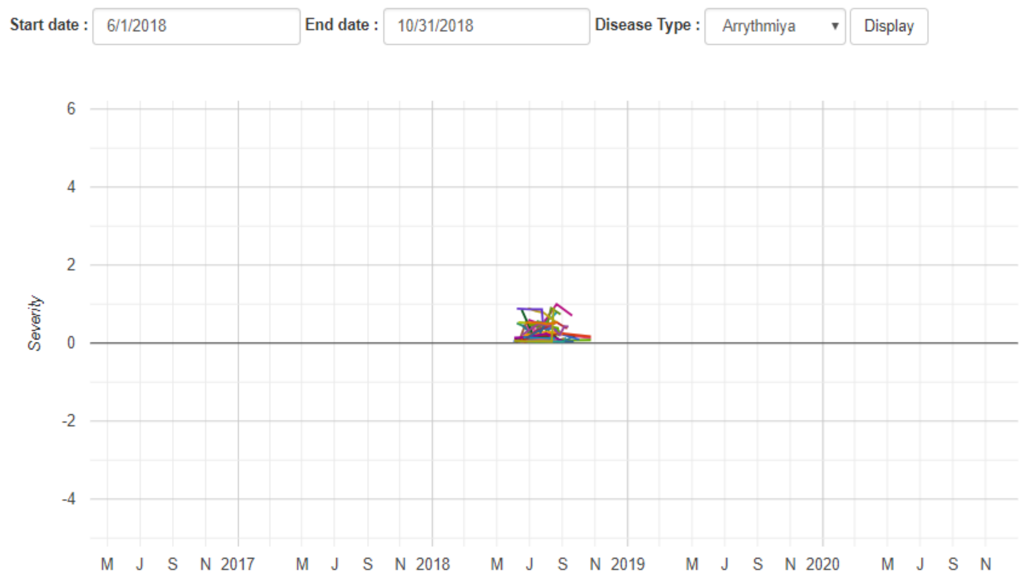


Figure 34. Zoom out version of time plot graph

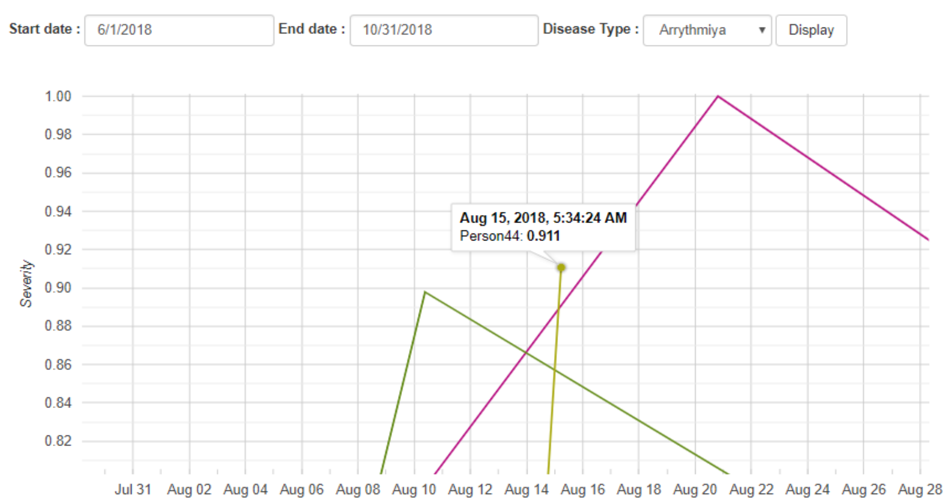


Figure 35. Zoom in version of Time plot Graph

C. Time plot Flow

If Startdate is 11-08-2018 and enddate is 15-08-2018 then Time plot (Flow) shows plot as Figure 36.

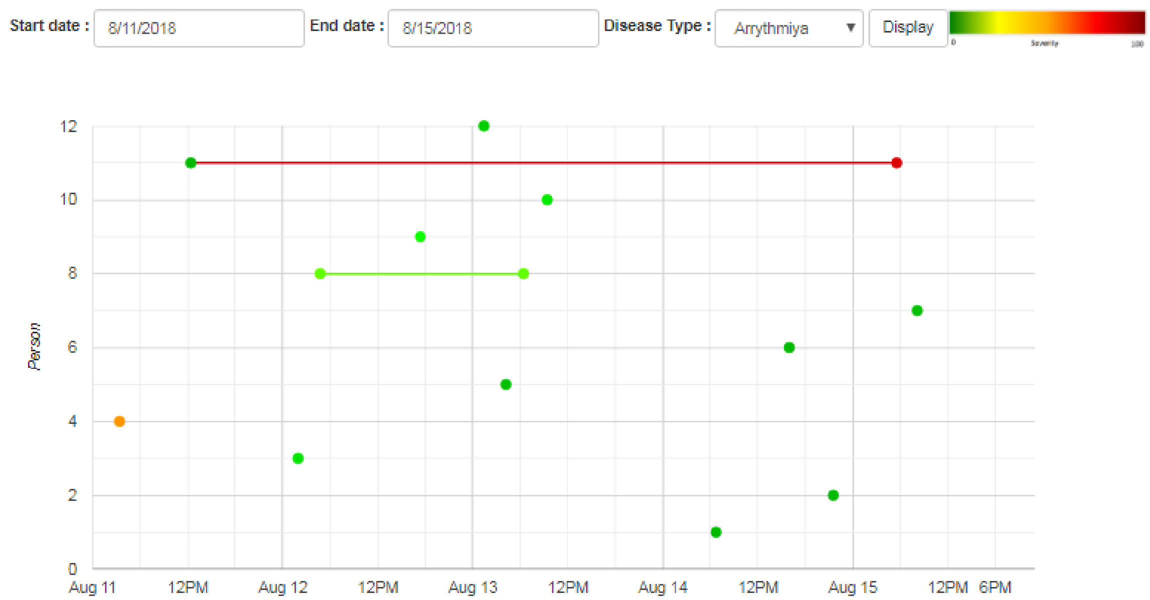


Figure 36. Time plot (Flow)

From the time plot flow, if we consider number 1 person EoI, then we find that he uploaded his data at 13th Aug 2018 and as his circle is almost dark green, it is said that the person is not affected by arrhythmia disease at that time. Person 11 at 11th Aug, 2018 at 12 PM uploaded his test EoI and as the circle is green he is safe from arrhythmia disease. But when he tested his arrhythmic condition at 15th Aug 2018, it was found that the severity looks red. The red circle symbolizes higher severity in arrhythmia disease. From green circle of Aug 11, 2018 to red circle of Aug 15, 2018 the person 11 is suffered from arrhythmia disease and these two circle is connected with a line whose color is similar to the last updated severity that is red. From the line it can be visualize that how many days a person is suffered from a disease. Also, it is visualized how fast member of a community recover from a disease.

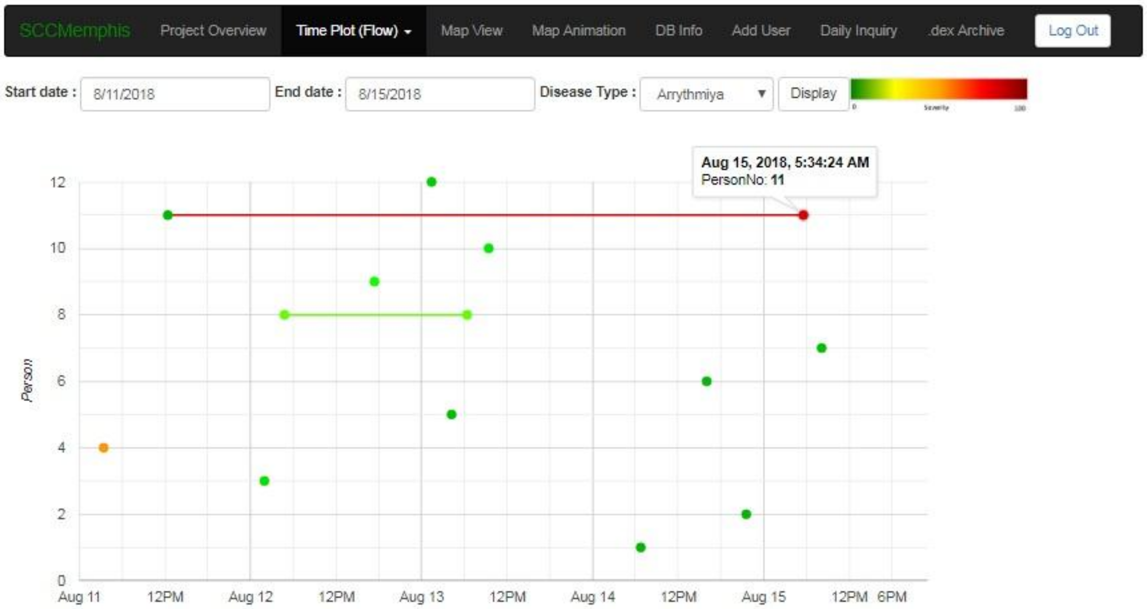


Figure 37. Pop-up Message of Time plot Flow

Figure 37 Shows pop-up message for time plot Flow. The pop-up message has two lines. First line represents test datetime of user and second line represents person no that is automatically created during JSON parsing. It is not persons' ID.

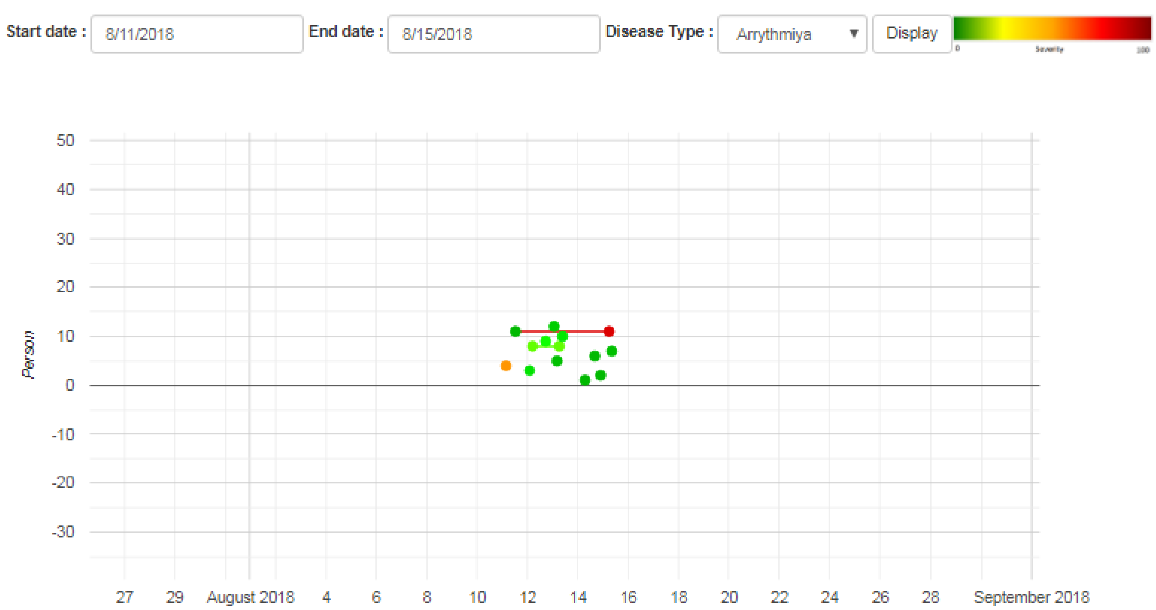


Figure 38. Zoom out version of Time plot Flow

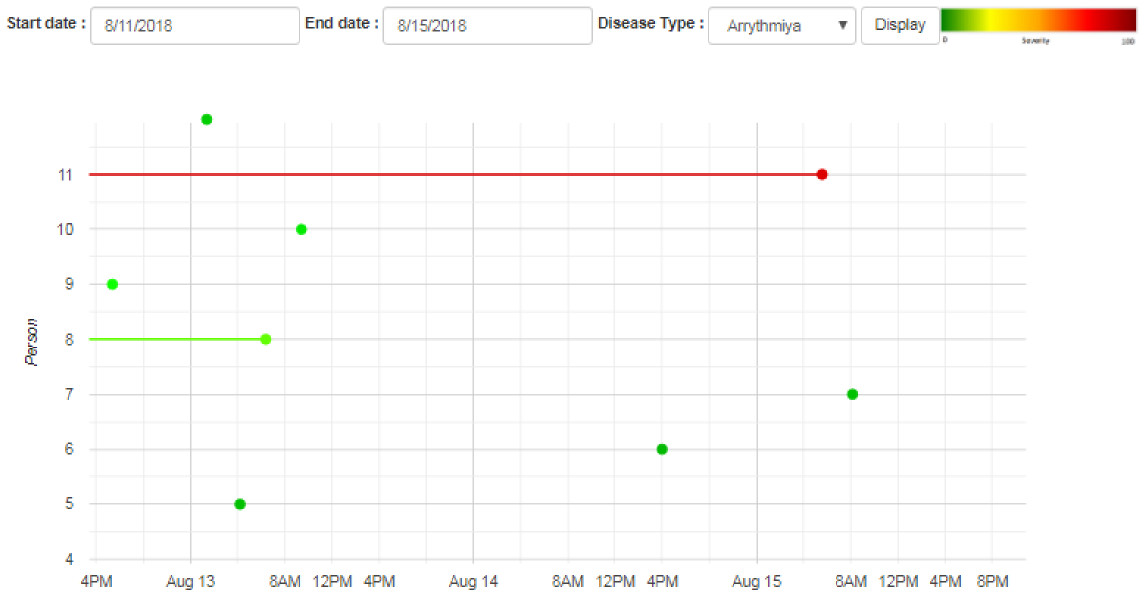


Figure 39. Zoom in version of Time Plot Flow

Figure 38 and Figure 39 represent Time plot flow zoom out and zoom in version respectively. Zoom in version helps user to see data in smaller granulate and Zoom out version helps user to see all data at a glance. It has similar properties as time plot graph for zoom version. So, it represents that time plot flow works properly.

D. Static Map view

Static map view represents condition of part of community through color derived from averaged EoI of that part of community members within user defined date range. Now from visualization chapter if start date is '2018-08-11' and end date is '2018-08-22' and disease type is 'AR' google map plots the following static map for that community members.

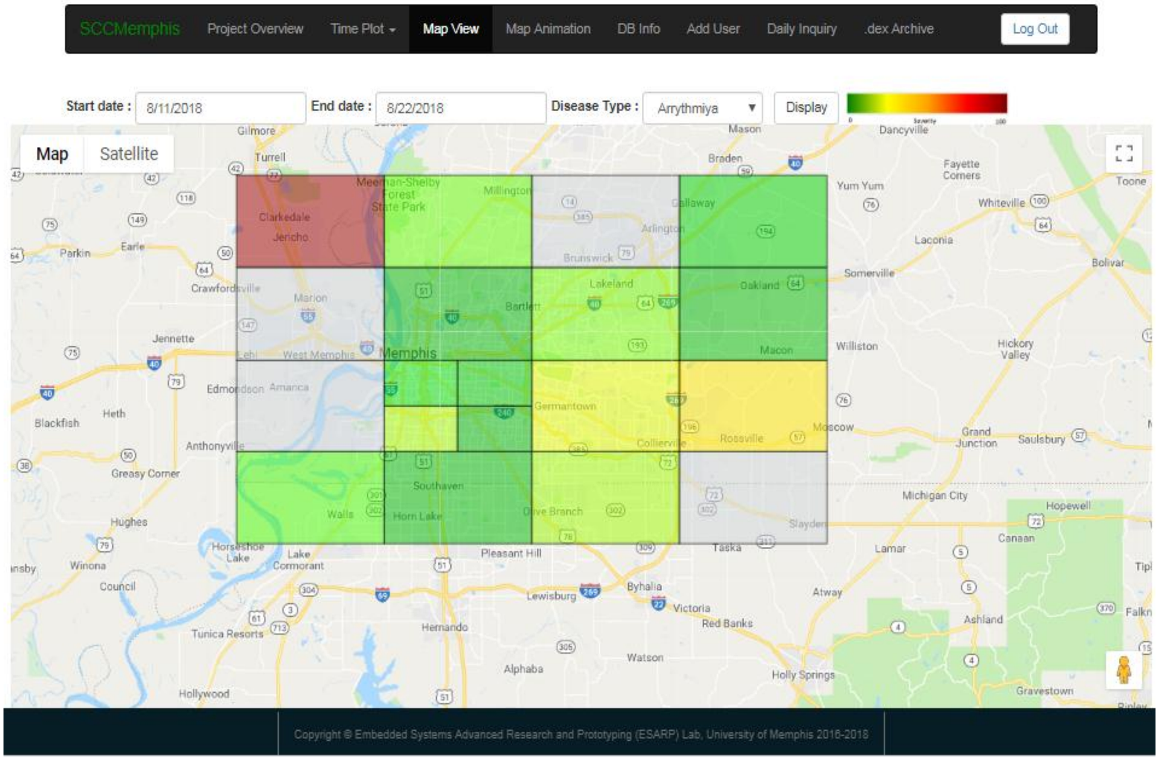


Figure 40. Static map view '2018-08-11' and end date is '2018-08-22'

From Figure 40, it has been shown that there is one grid area that is red colored, 5 dark green area codes, 2 green area, 4 light green grid area and 3 yellow grid area. To compare with Table 18 in visualization chapter, it has also been observed that static map view area color has similarities with JSON averaged grid colored area. So, it represents that static map view works perfectly to visualize community health status within user defined date.

As minimum zoom is 10 and maximum zoom is 12 for static map view, user couldn't stretch it beyond these values. Figure 41 and Figure 42 show zoom in and zoom out version of static map view.

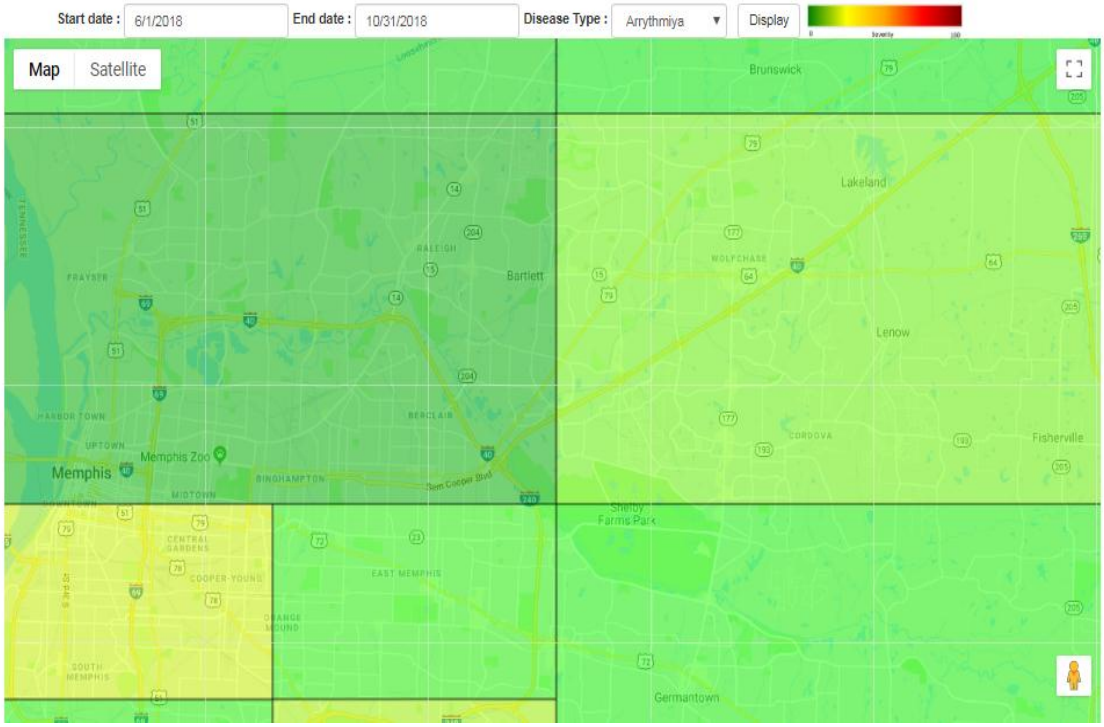


Figure 41. Zoom in version of static map view

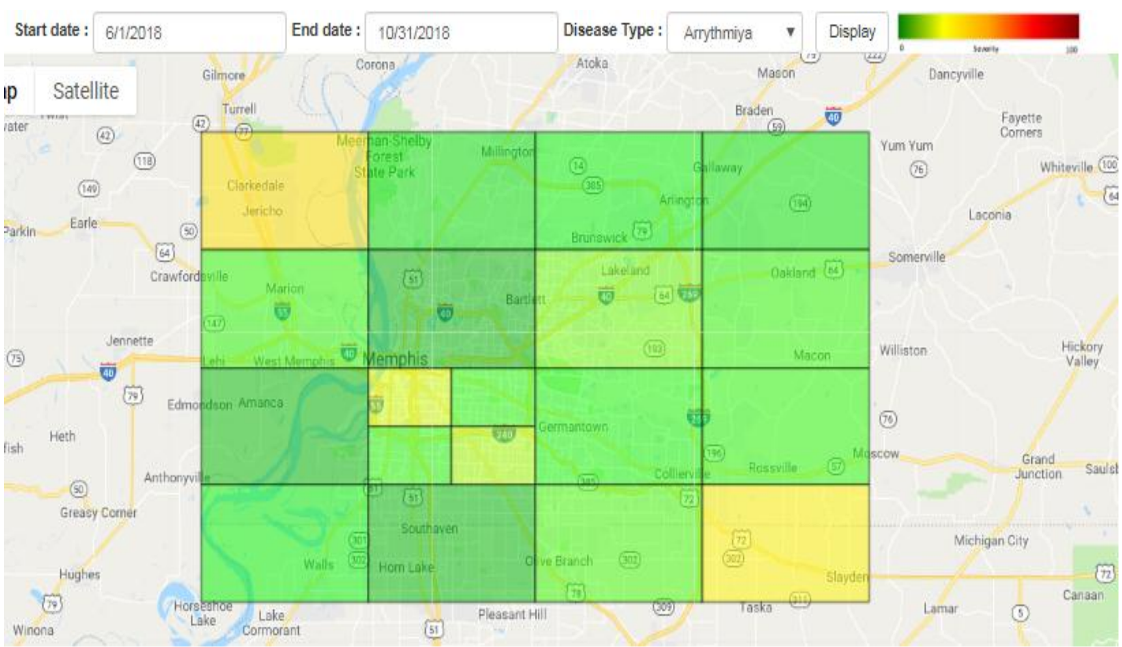


Figure 42. Zoom out version of static map view

E. Animation of Map

Animation of map gives user an understanding the spatial severity of disease in the progression of time. Cumulative animation is described from Figure 43-48.

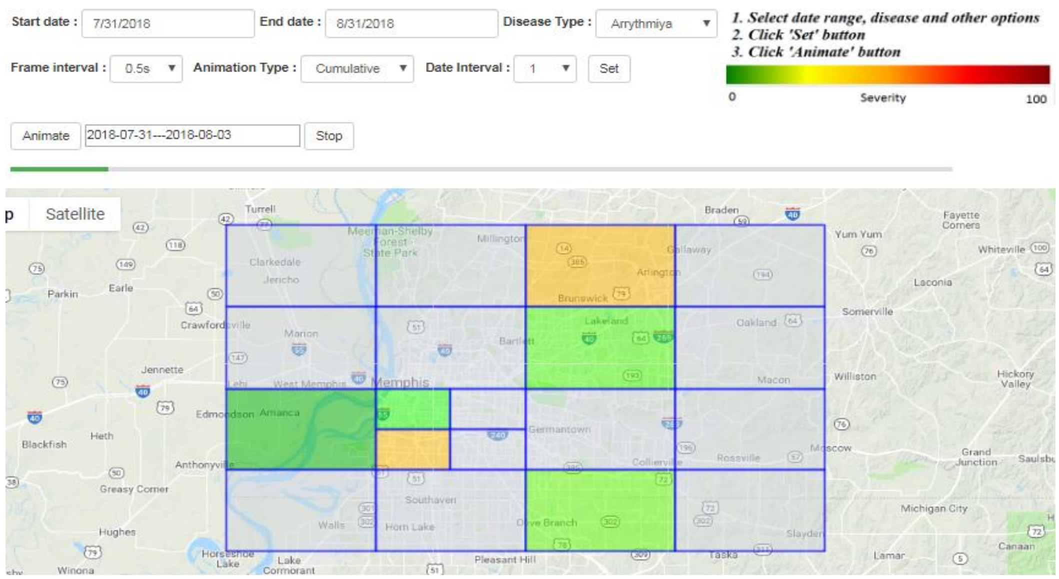


Figure 43. Cumulative animation bin 1 (2018-07-31 to 2018-08-03)

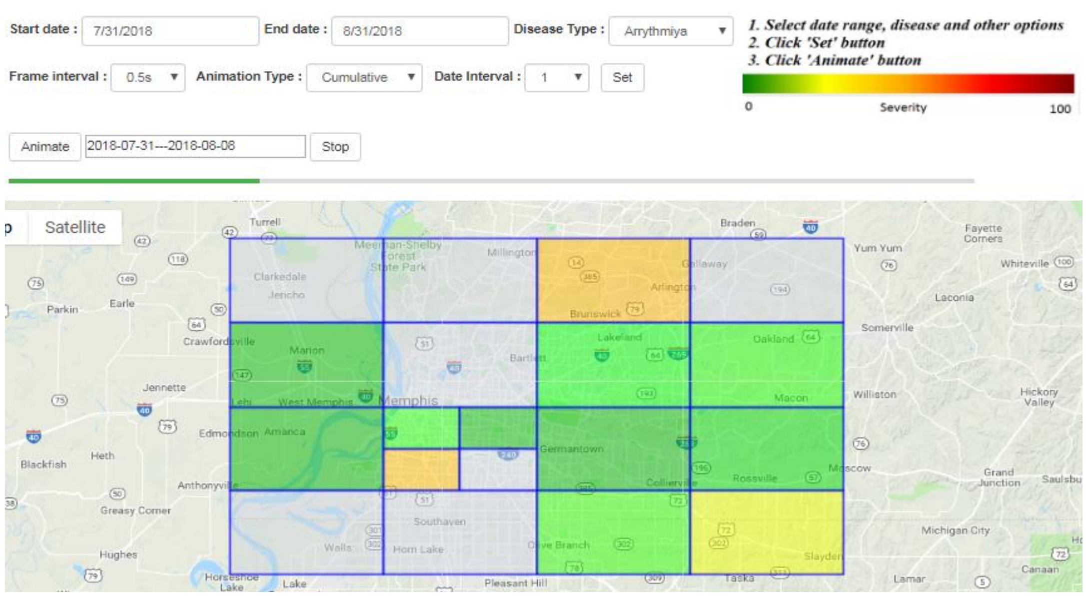


Figure 44. Cumulative animation bin 2 (2018-07-31 to 2018-08-08)

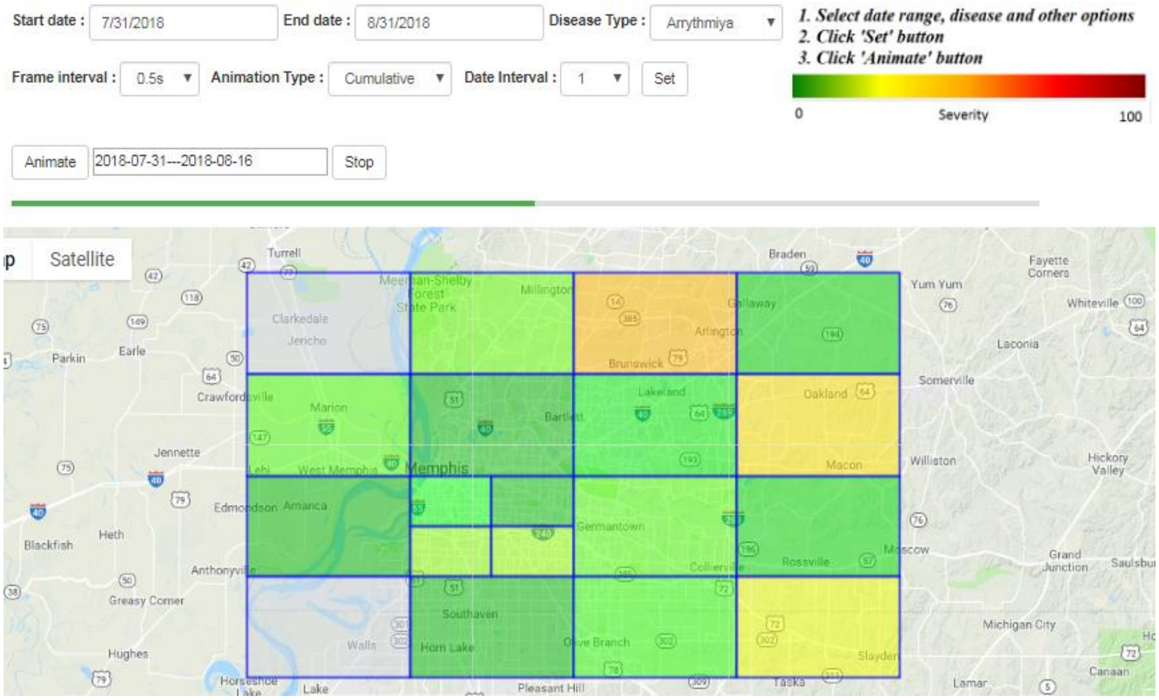


Figure 45. Cumulative animation bin 3 (2018-07-31 to 2018-08-16)

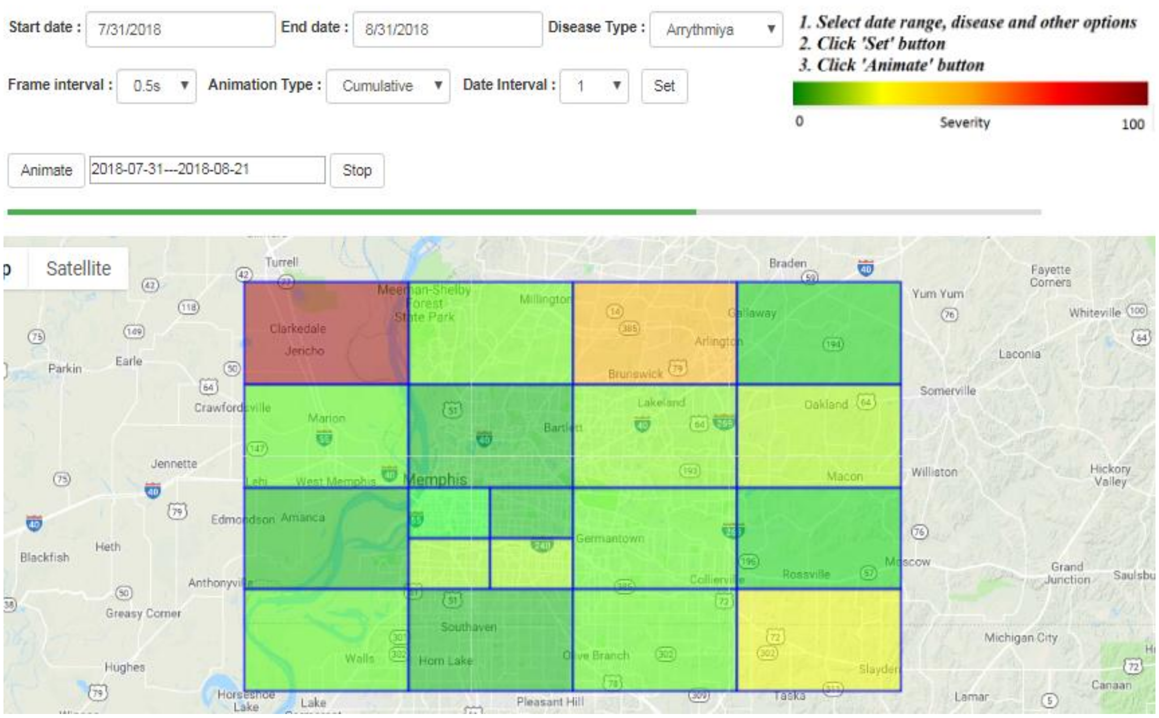


Figure 46. Cumulative animation bin 4 (2018-07-31 to 2018-08-21)

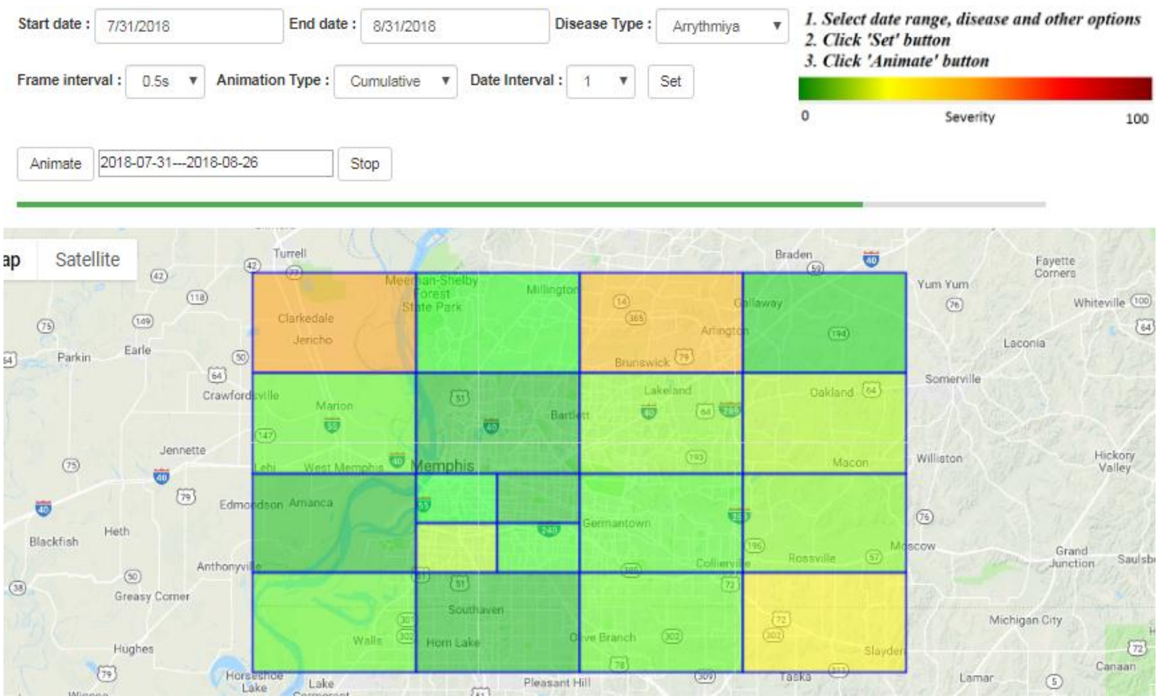


Figure 47. Cumulative animation bin 5 (2018-07-31 to 2018-08-26)

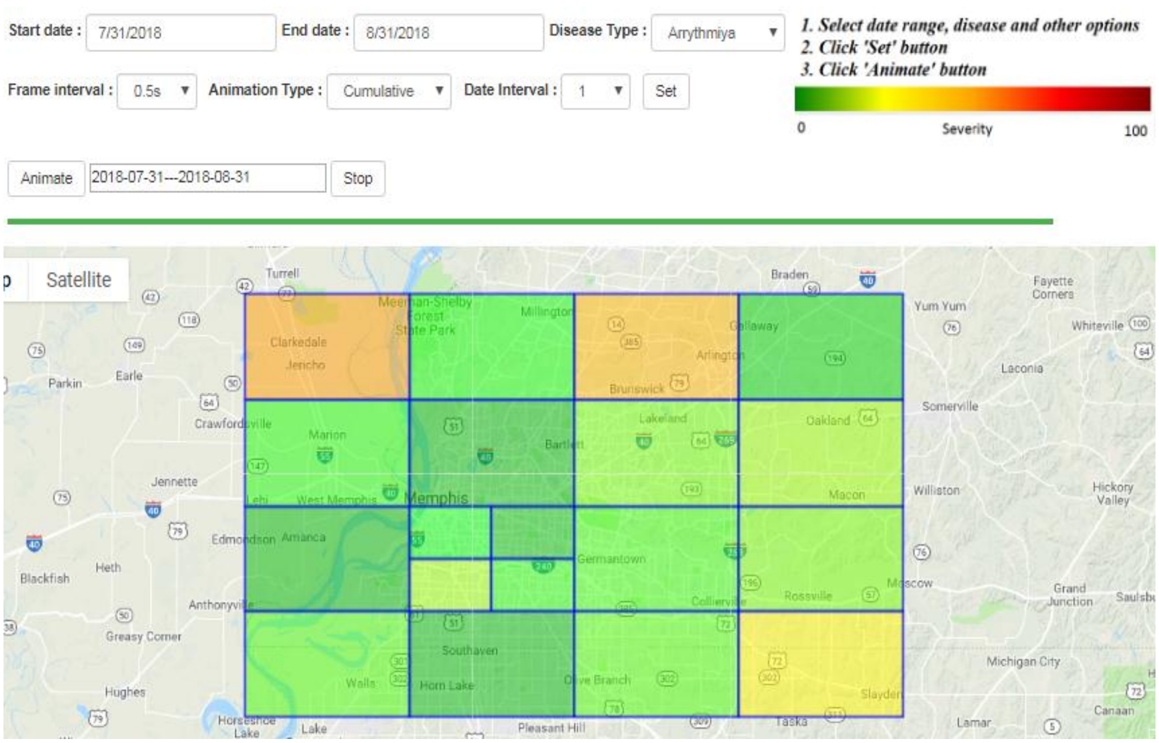


Figure 48. Cumulative animation bin 6 (2018-07-31 to 2018-08-31)

If startdate is 07/31/2018 and enddate is 08/31/2018 and date interval is 1, then in the progress of cumulative animation, six randomly chosen bins are given in Table 20.

Table 20. Six randomly chosen bins for Cumulative animation

Bin with date range	Total Colored grids	Grid count with respective color
Bin 1 (2018-07-31 to 2018-08-03) in Figure 43	5	1 grid dark green, 2 grids light green, 2 yellow grids
Bin 2 (2018-07-31 to 2018-08-08) in Figure 44	12	4 grids dark green, 5 grids light green, 1 yellow, 2 orange grids
Bin 3 (2018-07-31 to 2018-08-16) in Figure 45	17	4 dark green, 2 green, 6 light green, 2 light yellow, 2 yellow, 1 orange
Bin 4 (2018-07-31 to 2018-08-21) in Figure 46	19	4 dark green, 2 green, 7 light green, 3 grenish yellow, 1 yellow, 1 orange, 1 red
Bin 5 (2018-07-31 to 2018-08-26) in Figure 47	19	4 dark green, 1 green, 7 light green, 4 greenish yellow, 1 yellow, 2 orange
Bin 6 (2018-07-31 to 2018-08-31) in Figure 48	19	4 dark green, 1 green, 7 light green, 4 greenish yellow, 1 yellow, 2 orange

From table 20, it has been shown that as number of bin increases the number of colored grid increases. It happens for more community member participation. As more days added with startdate more EoI values have been averaged with previous averaged EoI. In bin 4, there is a red grid. But when days increase in the bin 5, it becomes orange. So, this cumulative animation shows continuous averaged progression of EoI value of community people.

In the segmented animation, only a single day or a bin of day are responsible to display animation. Its EoI value has not been averaged with their previous day or bin of days. It helps user to see the visualization of one single day or one single bin of day. Segmented animation is described in Figure 49-54.

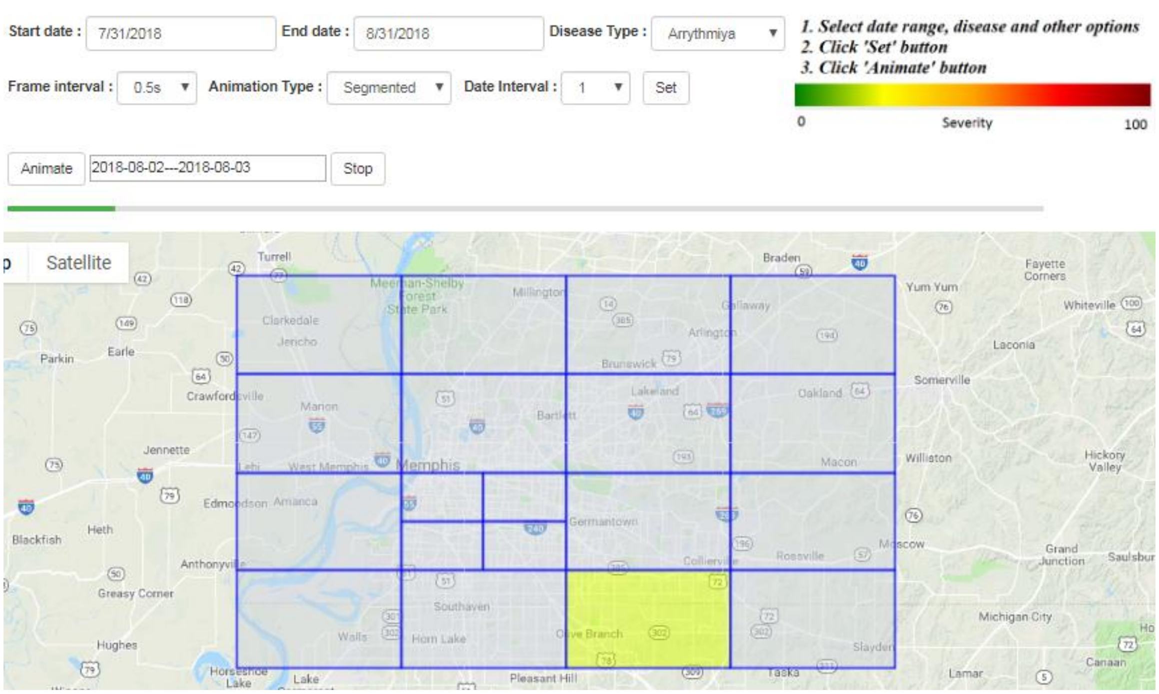


Figure 49. Segmented animation bin 1 (2018-08-02 to 2018-08-03)

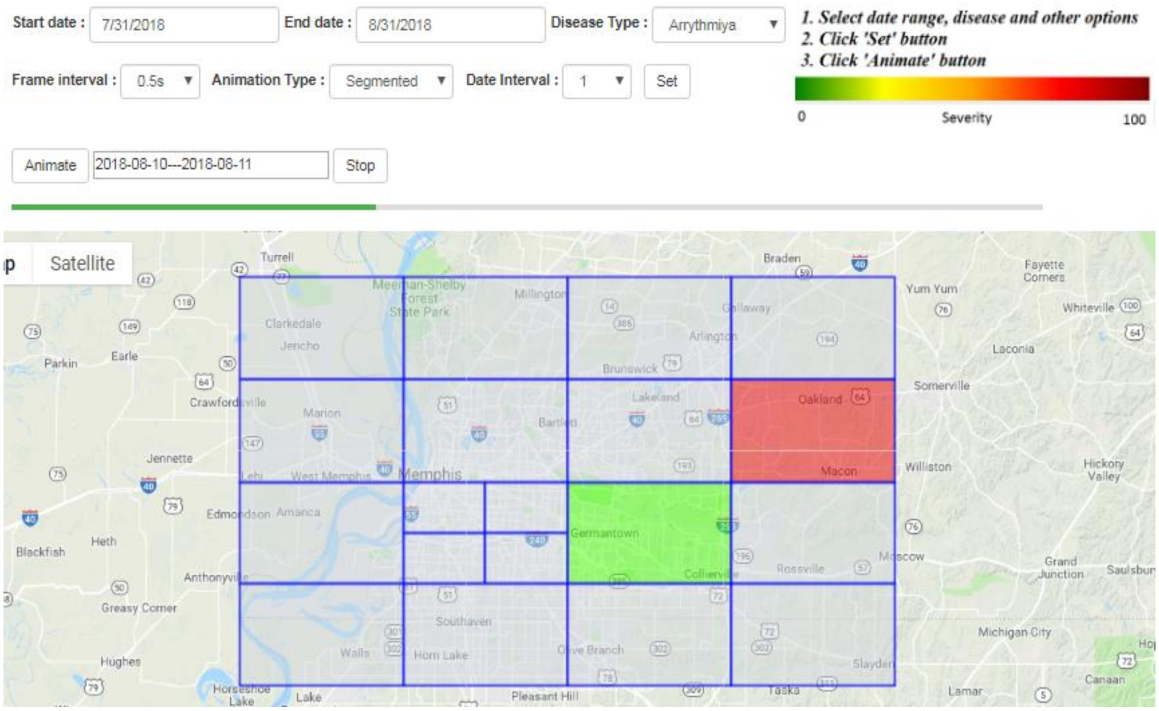


Figure 50. Segmented animation bin 2 (2018-08-10 to 2018-08-11)

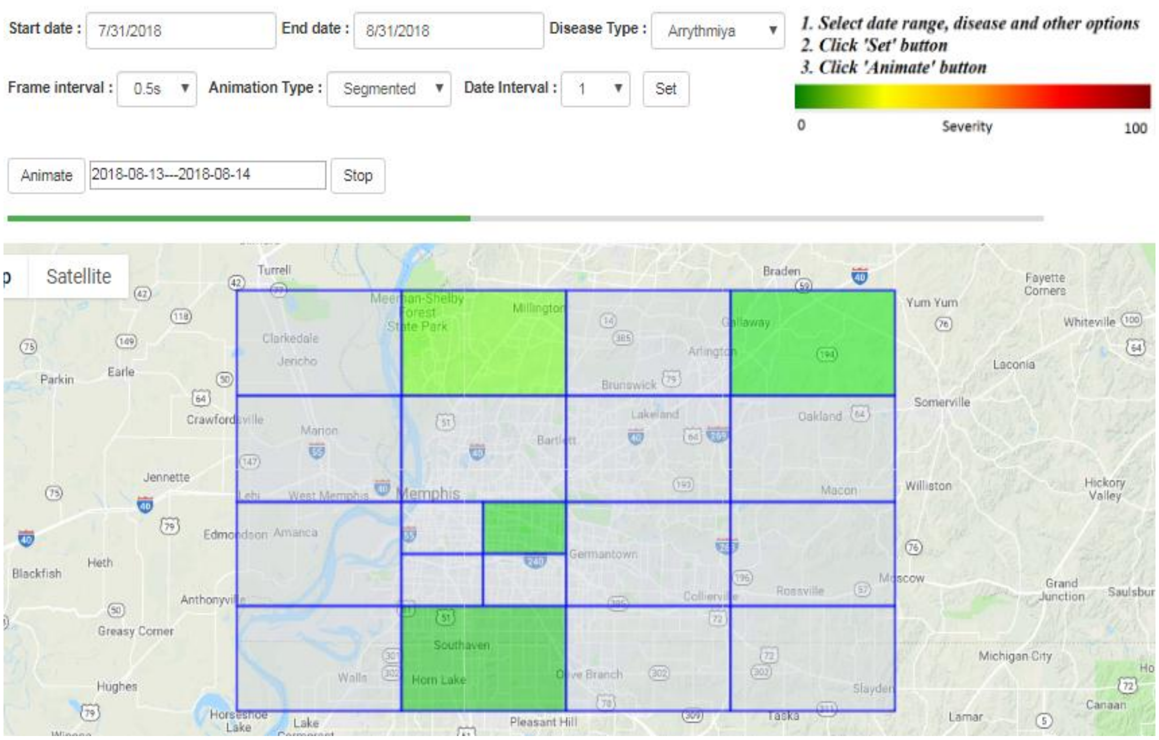


Figure 51. Segmented animation bin 3 (2018-08-13 to 2018-08-14)

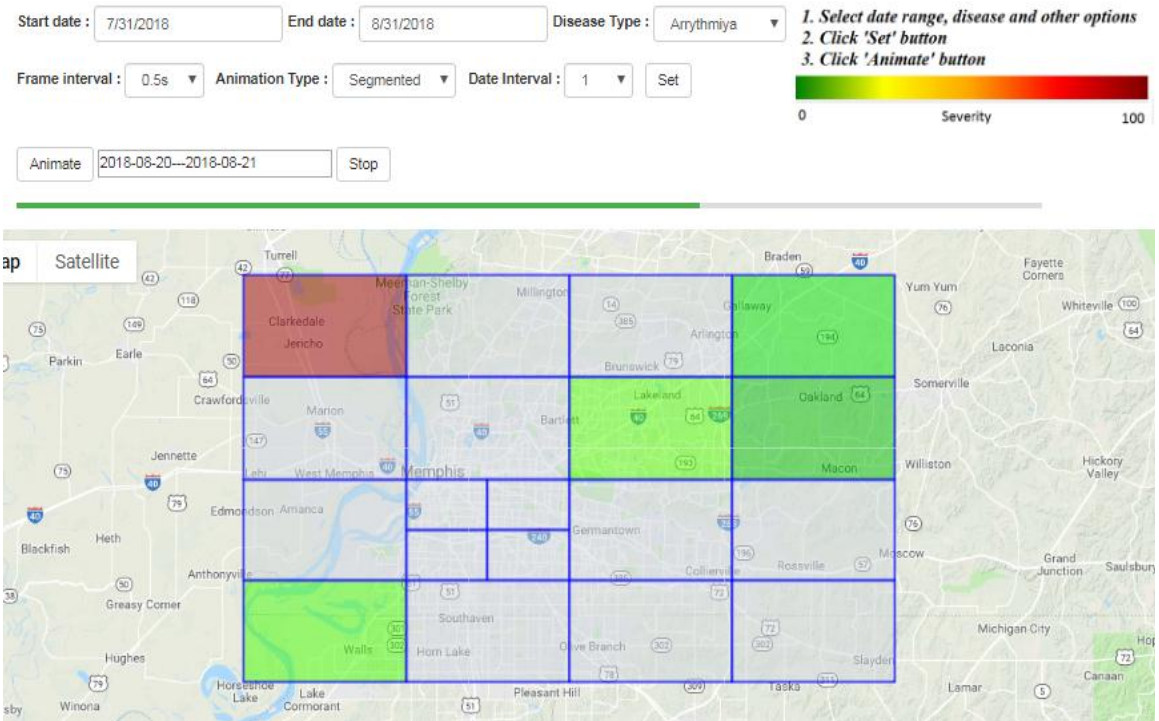


Figure 52. Segmented animation bin 4 (2018-08-20 to 2018-08-21)

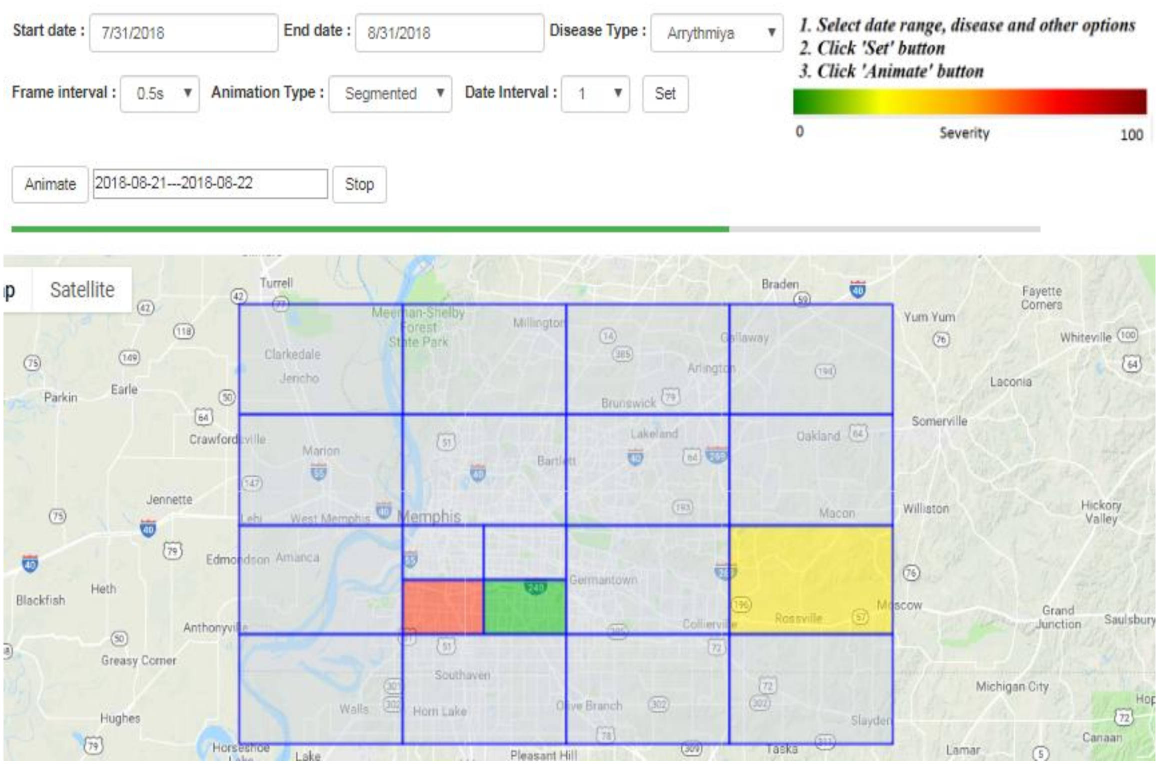


Figure 53. Segmented animation bin 5 (2018-08-21 to 2018-08-22)

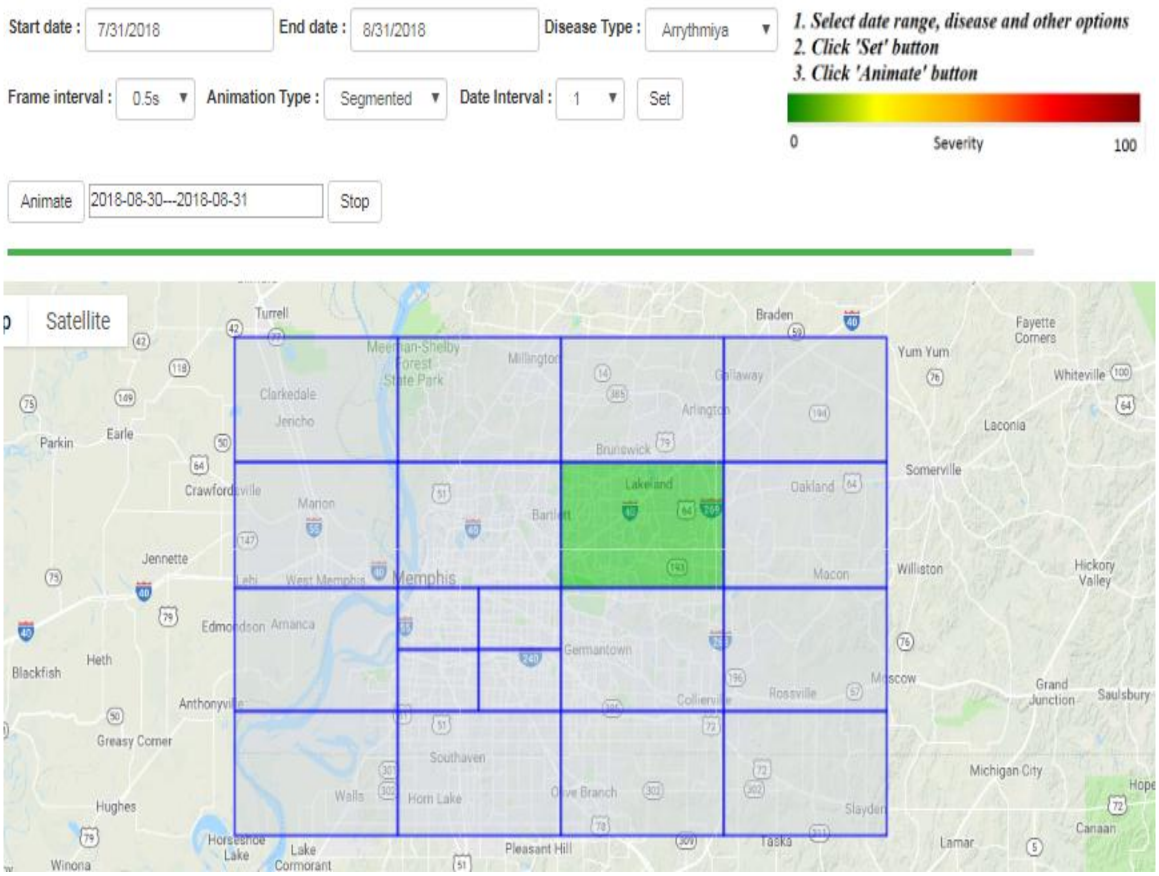


Figure 54. Segmented animation bin 6 (2018-08-30 to 2018-08-31)

If startdate is 07/31/2018 and enddate is 08/31/2018 and date interval is 1, then in the progress of segmented animation, six randomly chosen bins are in the following table

Table 21. Six randomly chosen bin for Segmented animation

Bin with date range	**Total Colored grids**	**Grid count with respective color**
Bin 1 (2018-08-02 to 2018-08-03) in Figure 49	1	1 yellow

Table 21 continued: Six randomly chosen bin for Segmented animation

Bin with date range	**Total Colored grids**	**Grid count with respective color**
Bin 2 (2018-08-10 to 2018-08-11) in Figure 50	2	1 green, 1 red
Bin 3 (2018-08-13 to 2018-08-14) in Figure 51	4	2 dark green, 1 green, 1 light green
Bin 4 (2018-08-20 to 2018-08-21) in Figure 52	5	1 dark green, 1 green, 2 light green, 1 red
Bin 5 (2018-08-21 to 2018-08-22) in Figure 53	3	1 green, 1 yellow, 1 light red,
Bin 6 (2018-08-31 to 2018-08-31) in Figure 54	1	1 green

From Table 21, it has been shown that with the progression of segmented day - based animation, the number of colored grids doesn’t increase. It displays the change of EoI for a single day. Only that day is responsible for that bin animation. From this segmented animation, one can visualize severity change in an individual grid.

F. Arrhythmia Severity detection and ranking:

Our motive in this thesis is to find a suitable model for arrhythmia severity detection and ranking of that severity. This severity should have proper distribution for all record so that the record works as real community member. From chapter IV, we choose Model 'Sqrt +Tangent' in Figure 55 as it distributes record data properly. From the visualization of time plot graph, flow and map view, it has also been observed that

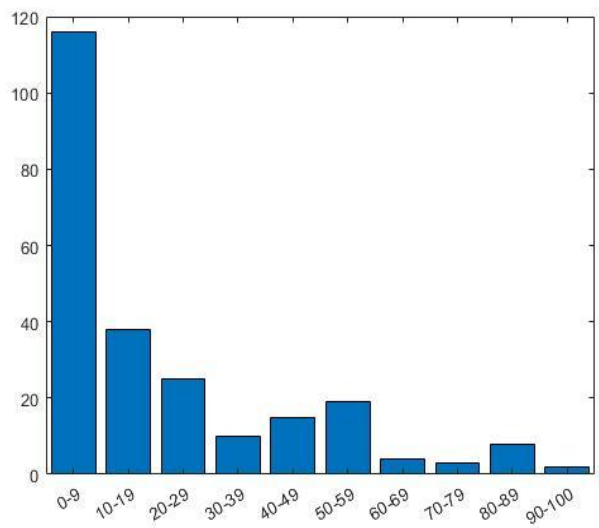


Figure 55. Selected Model 'Sqrt +Tangent' for checking Arrhythmia severity ranking

distribution and detection of severity have been done properly with selected model.

G. Survey Results for the Visualization Tool

In order to verify the effectiveness of the visualization tools, an informal survey is conducted among randomly chosen individuals from Memphis community. The survey consists of two sessions, i.e. pre-session and post-session containing same set of questions. The overall tone of the participants of the survey is positive. Although, in the pre-session survey some participants were not willing to share the physiological sensor-data (30%), but after seeing the visualization tools in the post-session survey they were willing to share their data as they got an understanding that their identity will not be revealed (90%). The participants stated several different ways that this is an effective visualization tool, e.g. “the database will help prevent spreading of a disease by taking proper precaution method”. For temporal plot, a participant described that “Having

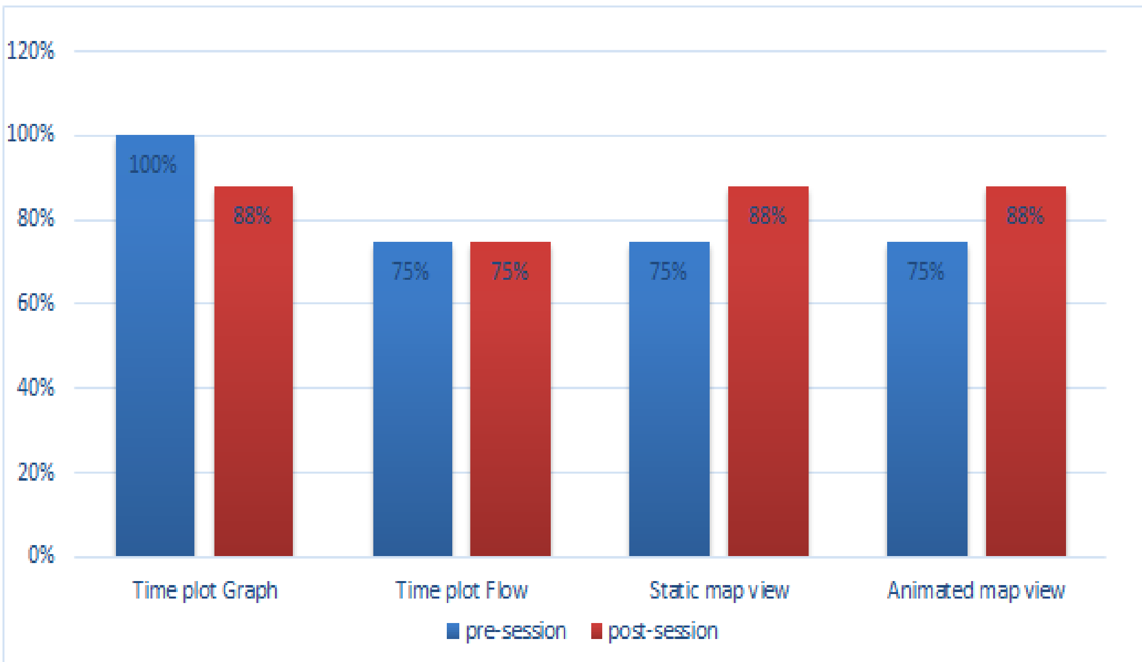


Figure 56. Participants’ responses of likely using the visualization summarized from the survey questionnaire.

access to the tool will help community to stay alarmed and know about what disease is growing”. Another participant commented on spatial plot: “The color code helps and gives us good vision toward the health of the community.” The following plot depicts the outcome (in percentage) of participants regarding how effective the web-based visualization is for the community health status. Figure 56 shows participants responses of survey about questionnaire. More people were keen to participate in the survey.

VII. Conclusions

A. Key Results

This thesis proposes the strategy for spatiotemporal graphical visualization of the community-based Smart and Connected Community (SCC) Health data through multiple plots and animation. We have developed the SCC-Health webserver framework for visualization from scratch and demonstrated full functionally with MIT-BHI database record and mock data. The benefits of this framework are:

a) The SCC members can contribute the physiological data without revealing their identity. They also can gather knowledge about the community health status from the temporal and spatial plots of the developed web-based visualization.

b) Since the personal information is not shared to contribute in the community health improvement, SCC members will likely to be more confident about their privacy and inclined to share anonymized EoI to the server.

c) Web-based visualization developed in this thesis can help stakeholders to monitor the most affected diseases, spreads, and trends of the community and efficiently deploy resources.

B. Discussion

Monitoring health status can be beneficial to the social networking point of view if the health-related social networking is developed while properly maintaining the privacy and security. Through this health social network, one can visualize the condition of region. This can help also the researchers to find the same patterned patients and can easily co-relate with their treatment pattern that help them to find better treatment. With the smart technology, community members can monitor their own diseases through sensors data that can collect required physiological signals. To evaluate community health status, we propose SCC to become connected to a SCC-Health network where they can anonymously monitor community health status in real-time. To make the health web effective, more people need to have interest to share their own EoI data [1].

In this visualization, SCC members can measure and visualize their severity of arrhythmia, COPD, sleep apnea, and flu in terms of EoI (representing disease severity from 0 to 1 encoded with colors). They can also monitor the continuous disease severity in terms of time interval from the temporal flow plot with the colored circles. It can be easily viewed from the plot of flow. Spatial plot represents the averaged severity of a particular area in a time interval. From this visualization, one can learn about disease spread and trends. The change of EoI values can easily be visualized from the animation of spatial plot. From these visualizations, community members can be benefited to lead a healthier life.

This research is being conducted in collaboration with an epidemiology researcher from the School of Public Health at the University of Memphis, and a medical doctor from Memphis area. Incorporation of this health professionals also augments the validity and impact of this research to build a healthy, smart, and connected community.

C. Future Directions

In future, the capability of the developed visualization tool will be enhanced. Adding measurements data to display how a disease is spreading throughout a region will be implemented. Also, forecasting capability about the imminent danger of a disease outbreak will be incorporated in the visualization tool.

References

[1] B. Morshed, B. Harmon, M. Rahman, M. Zaman, S. Afroz, and M. Rahman, "Inkjet Printed Fully-Passive Body-Worn Wireless Sensors for Smart and Connected Community (SCC)," *J. Low Power Electron. Appl.*, 2017.

[2] J. S. Luciano, G. P. Cumming, M. D. Wilkinson, and E. Kahana, "The emergent discipline of health web science," *Journal of Medical Internet Research*. 2013.

[3] K. Gobakis, A. Mavrigiannaki, K. Kalaitzakis, and D. D. Kolokotsa, "Design and development of a Web based GIS platform for zero energy settlements monitoring," in *Energy Procedia*, 2017, vol. 134, pp. 48–60.

[4] M. N. K. Boulos, "Web GIS in practice III: Creating a simple interactive map of England's Strategic Health Authorities using Google Maps API, Google Earth KML, and MSN Virtual Earth Map Control," *International Journal of Health Geographics*, 2005.

[5] R. Chen *et al.*, "ExplICU: A web-based visualization and predictive modeling toolkit for mortality in intensive care patients," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2015.

[6] T. M. Frink, J. V. Gyllinsky, and K. Mankodiya, "Visualization of multidimensional clinical data from wearables on the web and on apps," in *2017 IEEE MIT Undergraduate Research Technology Conference, URTC 2017*, 2018.

[7] M. H. Loorak, C. Perin, N. Kamal, M. Hill, and S. Carpendale, "TimeSpan: Using

Visualization to Explore Temporal Multi-dimensional Data of Stroke Patients," *IEEE Trans. Vis. Comput. Graph.*, 2016.

[8] Ruijie Wang ; Miguel Nacenta ; Xu Zhu ; Qi Dai, "Visualization of health behavior data for children and young adults," in *2017 Computing Conference*, 2017.

[9] M. Mirto, S. Fiore, L. Conte, L. V. Bruno, and G. Aloisio, "A spatial data analysis infrastructure for environmental health research," in *2016 International Conference on High Performance Computing and Simulation, HPCS 2016*, 2016.

[10] M. M. Jankowska, D. Lopez-Carr, C. Funk, G. J. Husak, and Z. A. Chafe, "Climate change and human health: Spatial modeling of water availability, malnutrition, and livelihoods in Mali, Africa," *Appl. Geogr.*, 2012.

[11] A. Houghton, N. Prudent, J. E. Scott, R. Wade, and G. Luber, "Climate change-related vulnerabilities and local environmental public health tracking through GEMSS: A web-based visualization tool," *Appl. Geogr.*, 2012.

[12] S. Purushe, G. Grinstein, M. B. Smrtic, and H. Lyons, "Interactive animated visualizations of breast, ovarian cancer and other health indicator data using weave, an interactive web - Based analysis and visualization environment," in *Proceedings of the International Conference on Information Visualisation*, 2011.

[13] C. C. Freifeld, K. D. Mandl, B. Y. Reis, and J. S. Brownstein, "HealthMap: Global Infectious Disease Monitoring through Automated Classification and Visualization of Internet Media Reports," *J. Am. Med. Informatics Assoc.*, 2008.

[14] Tolentino et al, "Scanning the Emerging Infectious Diseases Horizon - Visualizing

ProMED Emails Using EpiSPIDER," *Adv. Dis. Surveill.*, vol. 169, no. 2, 2007.

[15] E. M. Delmelle, H. Zhu, W. Tang, and I. Casas, "A web-based geospatial toolkit for the monitoring of dengue fever," *Appl. Geogr.*, 2014.

[16] E. C. Schofield *et al.*, "CHiCP: A web-based tool for the integrative and interactive visualization of promoter capture Hi-C datasets," *Bioinformatics,* 2016.

[17] X. Jiao *et al.*, "DAVID-WS: A stateful web service to facilitate gene/protein list analysis," *Bioinformatics*, 2012.

[18] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*. 2019.

[19] M. I. Owis, A. H. Abou-Zied, A. B. M. Youssef, and Y. M. Kadah, "Study of features based on nonlinear dynamical modeling in ECG arrhythmia detection and classification," *IEEE Trans. Biomed. Eng.*, 2002.

[20] B. Noroozi and B. I. Morshed, "Sensitivity optimization of Printed Spiral Coil for Wireless Resistive Analog Passive (WRAP) Sensors using Genetic Algorithm," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2018-July, no. 2, pp. 4653–4656, 2018.

[21] M. Abu-Saude and B. I. Morshed, "Accessing Differential Measures with a Conjugate Coil-Pair for Wireless Resistive Analog Passive (WRAP) ECG Sensors," *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2018-May, pp. 887–891, 2018.

[22] M. S. Zaman and B. I. Morshed, “Design and Verification of a Portable Scanner for Body-Worn Wireless Resistive Analog Passive (WRAP) Sensors,” *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2018-May, pp. 548–553, 2018.

[23] J. Rahman and S. Member, “Exacerbation in Obstructive Sleep Apnea : Early Detection and Monitoring Using a Single Channel EEG with Quadratic Discriminant Analysis *,” pp. 85–88, 2019.

[24] M. J. Rahman, R. Mahajan, and B. I. Morshed, “Severity classification of obstructive sleep apnea using only heart rate variability measures with an ensemble classifier,” *2018 IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2018*, vol. 2018-January, pp. 33–36, 2018.

[25] T. Siddiqui and B. I. Morshed, “Severity Classification of Chronic Obstructive Pulmonary Disease and Asthma with Heart Rate and SpO2 Sensors,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2018-July, no. 1637250, pp. 2929–2932, 2018.

[26] J. Rahman and B. I. Morshed, “SCC Health : A Framework for Online Estimation of Disease Severity for the Smart and Connected Community *,” no. 1637250.

[27] M. Memphis, “Improving Accuracy of Inkjet Printed Core Body WRAP Temperature Sensor Using Random Forest Regression Implemented with an Android App,” vol. 2, no. c.

[28] C. Tso, G. M. Currie, D. Gilmore, and H. Kiat, “Electrocardiography: A Technologists Guide to Interpretation,” *J. Nucl. Med. Technol.*, 2015.

[29] E. J. da S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, “ECG-based heartbeat classification for arrhythmia detection: A survey,” *Comput. Methods Programs Biomed.*, 2016.

[30] D. Rakel, *Integrative Medicine*. 2018.

[31] “ecg-kit a Matlab Toolbox for Cardiovascular Signal Processing,” *J. Open Res. Softw.*, 2016.

[32] G. B. Garcia, Tomas ; T. Miller, *Arrhythmia Recognition: The Art of Interpretation*. 2004.

[33] T. Siddiqui and B. I. Morshed, “Severity Exploratory Model Analysis of Chronic Obstructive Pulmonary Disease and Asthma with Heart Rate and SpO2,” *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2018-May, pp. 913–916, 2018.

Appendices

A. Web Server Code

(i) Getting JSON From Mobile

Insertjsondb.php

```
<?php
$data = json_decode(file_get_contents('php://input'), true);
print_r($data);

$patient_id = $data["ID"];
$area_code= $data["GRID_CODE"];
$disease_type=$data["DT"];
$eoi =$data['EOI'];
$datetime=$data['TIME'];
$algorithm=$data['ALG'];

$sdatetime = new DateTime($datetime);
        $sYear=$sdatetime->format('Y');
        $sMonth=$sdatetime->format('m');
        $sDay=$sdatetime->format('d');
        $sHour=$sdatetime->format('H');
        $sMin=$sdatetime->format('i');
        $sSec=$sdatetime->format('s');

        $new_datetime=$sYear."-".$sMonth."-".$sDay." ".$sHour.":".$sMin.":".$sSec;

$link = mysqli_connect("192.185.4.125", "sscmemph_lab209", "123456", "sscmemph_emdb1");

$sql = "INSERT INTO MIT_BHI_Record(patient,area_code,disease_type,eoi,datetime,algorithm)
         VALUES ('$patient_id', '$area_code','$disease_type','$eoi','$new_datetime','$algorithm');";

  $retval = mysqli_query($link, $sql );

  if(! $retval ) {
    die('Could not enter data: ' . mysqli_error($link));
  }

 if ($link ->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

?>
```

(ii) JavaScript Code for Time plot Graph

Timeplot Flow(Frontend):

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

 <script language="javascript" type="text/javascript"
      src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.0/jquery.min.js">
   </script>
```

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script type="text/javascript">
    $( function() {
   $( "#daytimedatepicker1" ).datepicker();

   } );
    $( function() {
   $( "#daytimedatepicker2" ).datepicker();
   } );


  google.charts.load('current', {packages: ['corechart', 'line']});
  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

   var jsonData = $.ajax({
          url: "http://sccmobilehealth.com/research/admin/admintracker.php",
           dataType: "json",
           async: false
        }).responseText;

    var pp = JSON.parse(jsonData);
    var totalentry=pp.length;
   if(totalentry>3)
    {
     var personnum=pp[3].length-4;       // -4 because first 4 number of each row is for year,month,day,hour
   }
    else
    var personnum=0;

   var data = new google.visualization.DataTable();
    data.addColumn('datetime', 'Time of Day');

/// when there is no person data then add a column to plot the chart
   if(personnum==0)
     data.addColumn('number', "no data is available");
   else{// first 2 number is for startdate and enddate for plot chart line
     var c;
   for(c=1;c<personnum+1;c++)
    {   var ss = "Person"+c;
      data.addColumn('number', ss);

     }
    }

    var mindatestr=pp[0];                    // pp[0] holds data for startdate
    var mindate=new Date(mindatestr);
    var min_year = mindate.getFullYear();
    var min_month=mindate.getMonth();
    var min_day=mindate.getDate();
    var min_hour =mindate.getHours();
   var lastdatestr=pp[1];                    // pp[1] holds data for enddata
```

```
      var lastdate=new Date(lastdatestr);
      var last_year = lastdate.getFullYear();
      var last_month=lastdate.getMonth();
      var last_day=lastdate.getDate();
      var last_hour =lastdate.getHours();
      var disea_type=pp[2];
      var disStartdate=min_month+1+"/"+min_day+"/"+min_year;
       var disEnddate=last_month+1+"/"+last_day+"/"+last_year;

var data2=[];
for(var i=3;i<totalentry;i++)
{
  var temp=[];
  var dd= new Date(pp[i][0],pp[i][1],pp[i][2],pp[i][3]);   //year,month,day,hou
    temp.push(dd);
   for(var j=4;j<personnum+4;j++)                              // after 4 number each row add the num
    temp.push(pp[i][j]);
     data.addRow(temp);
}

    var options = {
       legend : 'none',
        width: '95%',
        height: 400,
         interpolateNulls : true,
          pointSize: 3,

        chartArea: {
          width: '75%',
          height:'80%'
        },
        hAxis: {
          viewWindow: {
            min: new Date(min_year,min_month, min_day, min_hour),
            max: new Date(last_year, last_month, last_day, last_hour)
          },
          gridlines: {
            count: -1,
            units: {
              days: {format: ['MMM dd']},
              hours: {format: ['HH:mm', 'ha']},
            }
          },
          minorGridlines: {
            units: {
              hours: {format: ['hh:mm:ss a', 'ha']},
              minutes: {format: ['HH:mm a Z', ':mm']}
            }
          }

        },
        vAxis: {title: 'Severity'},
        explorer: { maxZoomIn: .2 ,
        maxZoomOut: 8,
        zoomDelta:1.5,
        keepInBounds: true}
```

```
      };

      var chart = new google.visualization.LineChart(
       document.getElementById('chart_div'));

      chart.draw(data, options);

     document.getElementById('daytimedatepicker1').value=disStartdate;
       document.getElementById('daytimedatepicker2').value=disEnddate;
       document.getElementById('disease_type').value=disea_type;
//// set the button through limit

  var setButton = document.getElementById("set");
  setButton.onclick=function()
  {  options.hAxis.viewWindow.min = new Date(min_year,min_month, min_day, min_hour);
     options.hAxis.viewWindow.max = new Date(last_year, last_month, last_day, last_hour );
     chart.draw(data, options);

  };

    }

 var btnlogOut = document.getElementById("logout");
    btnlogOut.onClick=function()
    {
   //  alert("logout");
     header("location: logout.php")
      var timer = setTimeout(function() {

           window.location='logout.php'
        }, 300);
    }
  </script>
```

(iii) Php code for Time plot Graph

TimeplotGraph: Backend

```
<?php
session_start();
{$link = mysqli_connect("192.185.4.125", "sscmemph_lab209", "123456", "sscmemph_emdb1");
   $disease_type = "AR";
   if(isset($_SESSION['daytimedatepicker1'])&&isset($_SESSION['daytimedatepicker2']))
     {
             if(isset($_SESSION['disease_type'])){
                  $disease_type=$_SESSION['disease_type'];}
              $sdatetime = new DateTime($_SESSION['daytimedatepicker1'] );
              $sYear=$sdatetime->format('Y');
              $sMonth=$sdatetime->format('m');
              $sDay=$sdatetime->format('d');
              $start_datetime=$sYear."-".$sMonth."-".$sDay." 00:00:00";

              $edatetime = new DateTime($_SESSION['daytimedatepicker2'] );
              $eYear=$edatetime->format('Y');
              $eMonth=$edatetime->format('m');
              $eDay=$edatetime->format('d');
              $end_datetime=$eYear."-".$eMonth."-".$eDay." 23:59:59";
```

```
                $mindate=$start_datetime;
                $lastday=$end_datetime;
     }
    else
     {
     $Sql2="SELECT MIN(datetime) as EarliestDate, MAX(datetime) as Lastdate
          FROM MIT_BHI_Record where patient not LIKE 'T%'";
          $result2=mysqli_query( $link,$Sql2);
                while( $row = mysqli_fetch_assoc( $result2 ) )
                 {
                  $mindate=$row["EarliestDate"];
                  $lastday =$row["Lastdate"];
                  $start_datetime=$mindate;
                  $end_datetime=$lastday;
                 }
     }

  $mindate=$start_datetime;
  $lastday=$end_datetime;

   $datetime_array=[];
   $sql="select patient,area_code,disease_type,datetime,eoi
         from MIT_BHI_Record
           where disease_type = '".$disease_type."'
             and ( datetime >= '".$start_datetime."' and datetime <= '".$end_datetime."') and patient not LIKE
'T%' order by datetime ASC";

             $result=mysqli_query( $link,$sql);
                while( $row = mysqli_fetch_assoc( $result ) )
                 {
                  $patient_Array[]=$row["patient"];
                  $area_code_Array[]=$row["area_code"];
                  $disease_type_Array[]=$row["disease_type"];
                  $eoi_Array[]=$row["eoi"];

                  $datetime=$row["datetime"];
                  $datetime_array[]=$datetime;
                  $datetime = new DateTime($datetime);
                  $year_Array[]=$datetime->format('Y');
                  $month_Array[]=$datetime->format('m');
                  $day_Array[]=$datetime->format('d');
                  $hour_Array[]=$datetime->format('H');
                  $minute_Array[]=$datetime->format('i');
                  $sec_Array[]=$datetime->format('s');
                  }


 $sql2="select distinct patient
        from MIT_BHI_Record
          where disease_type = '".$disease_type."' and patient not LIKE 'T%'
             and ( datetime >= '".$start_datetime."' and datetime <= '".$end_datetime."')";

  $onlypatient=[];
  $result2=mysqli_query($link,$sql2);
   while($row2=mysqli_fetch_assoc($result2))
```

```
    {$onlypatient[]=$row2["patient"];}

$countdatetime=count($datetime_array);
$countpatient=count($onlypatient);
$myArray=array_fill(0, $countpatient+4, null);

  $finalarray= array();
   array_push($finalarray,$mindate);
    array_push($finalarray,$lastday);
    array_push($finalarray,$disease_type);

   for($i=0;$i<$countdatetime;$i++)
   {
    $patient_id = array_search ($patient_Array[$i], $onlypatient);

       $myArray[0]=(int) $year_Array[$i];
       $myArray[1]= ((int)$month_Array[$i])-1;
       $myArray[2]=(int)$day_Array[$i];
       $myArray[3]=(int)$hour_Array[$i];
       $myArray[$patient_id+4]=(double)$eoi_Array[$i];

       array_push($finalarray, $myArray);

    $myArray=array_fill(0, $countpatient+4, null);
   }
 echo json_encode(array_values($finalarray));

 unset($_SESSION['daytimedatepicker1']);
unset($_SESSION['daytimedatepicker2']);
unset($_SESSION['disease_type']);

 mysqli_close($link);
 }
?>
```

(iv)Timeplot (Flow) Frontend:

```
  google.charts.load('current', {packages: ['corechart', 'line']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

      var jsonData = $.ajax({
              url: "http://sccmobilehealth.com/research/admin/admindatetimeNewdb1.php",
              dataType: "json",
              async: false
           }).responseText;

       var pp = JSON.parse(jsonData);
       var totalentry=pp.length;
       // first 2 number is for startdate and enddate for plot chart line
      if(totalentry>3)
       {
        var personnum=pp[3].length-6;       // -6 because first 6 number of each row is for
year,month,day,hour,min,sec
     }
```

```
       else
       var personnum=0;
       var data = new google.visualization.DataTable();
      data.addColumn('datetime', 'Time of Day');
/// when there is no person data then add a column to plot the chart
      if(personnum==0)
        data.addColumn('number', "no data is available");
      else{

      var c;
      var cc=0;
      for(c=1;c<personnum+1;c+=2)
       {   var ss = "PersonNo";
           cc++;
         data.addColumn('number', ss);
         data.addColumn( {'type': 'string', 'role': 'style'});
        }
       }

       var mindatestr=pp[0];               // pp[0] holds data for startdate
       var mindate=new Date(mindatestr);
       var min_year = mindate.getFullYear();
       var min_month=mindate.getMonth();
       var min_day=mindate.getDate();
       var min_hour =mindate.getHours();


      var lastdatestr=pp[1];                // pp[1] holds data for enddata
      var lastdate=new Date(lastdatestr);
      var last_year = lastdate.getFullYear();
      var last_month=lastdate.getMonth();
      var last_day=lastdate.getDate();
      var last_hour =lastdate.getHours();

       var disea_type=pp[2];


var disStartdate1=min_month+1+"/"+min_day+"/"+min_year ;
       var disEnddate1=last_month+1+"/"+last_day+"/"+last_year;
for(var i=3;i<totalentry;i++)
{
  var temp=[];
  var dd= new Date(pp[i][0],pp[i][1],pp[i][2],pp[i][3],pp[i][4],pp[i][5]);           // year,month,day,hour

    temp.push(dd);
   for(var j=6;j<personnum+6;j++)
   if (j%2==1)
   {var color= 'point { size: 10; shape-type: circle; fill-color: '+pp[i][j] +';}';                          // after 4
number each row add the num
    temp.push(pp[i][j]);
  } else
  {
    temp.push(pp[i][j]);
  }

     data.addRow(temp);
```

```
}
```

(v)TimeplotFlow : Backend

```
<?php

   $datetime_array=[];
   $sql="select patient,area_code,disease_type,datetime,eoi
         from MIT_BHI_Record
            where disease_type = '".$disease_type."'
              and ( datetime >= '".$start_datetime."' and datetime <= '".$end_datetime."') and patient not LIKE
'T%' order by datetime ASC";

                $result=mysqli_query( $link,$sql);
                    while( $row = mysqli_fetch_assoc( $result ) )
                     {
                      $patient_Array[]=$row["patient"];
                      $area_code_Array[]=$row["area_code"];
                      $disease_type_Array[]=$row["disease_type"];
                     $eoi_Array[]=intval($row["eoi"]*100);

                      $datetime=$row["datetime"];
                      $datetime_array[]=$datetime;
                      $datetime = new DateTime($datetime);

                      $year_Array[]=$datetime->format('Y');
                      $month_Array[]=$datetime->format('m');
                      $day_Array[]=$datetime->format('d');
                      $hour_Array[]=$datetime->format('H');
                      $minute_Array[]=$datetime->format('i');
                      $sec_Array[]=$datetime->format('s');

                     }

 $sql2="select distinct patient
        from MIT_BHI_Record
          where disease_type = '".$disease_type."' and patient not LIKE 'T%'
              and ( datetime >= '".$start_datetime."' and datetime <= '".$end_datetime."')";

  $onlypatient=[];
  $result2=mysqli_query($link,$sql2);
   while($row2=mysqli_fetch_assoc($result2))
   {$onlypatient[]=$row2["patient"];}

$countdatetime=count($datetime_array);
$countpatient=count($onlypatient);

$color=array(
"#009D00","#00A400","#00AB00","#00B200","#00B900","#00C000","#00C700","#00CE00","#00D500",
"#00DC00","#00E300"

,"#00EA00","#00F100","#00F800","#00FF00","#07FF00","#0EFF00","#15FF00","#1CFF00","#23FF00","
#2AFF00"
```

```
,"#31FF00","#38FF00","#3FFF00","#46FF00","#4DFF00","#54FF00","#5BFF00","#62FF00","#69FF00",
"#70FF00"

,"#77FF00","#7EFF00","#85FF00","#8CFF00","#93FF00","#9AFF00","#A1FF00","#A8FF00","#AFFF00
","#B6FF00"

,"#BDFF00","#C4FF00","#CBFF00","#D2FF00","#D9FF00","#E0FF00","#E7FF00","#EEFF00","#F5FF0
0","#FFFF00"

,"#FFF800","#FFF100","#FFEA00","#FFE300","#FFDC00","#FFD500","#FFCE00","#FFC700","#FFC00
0","#FFB900"

,"#FFB200","#FFAB00","#FFA400","#FF9D00","#FF9600","#FF8F00","#FF8800","#FF8100","#FF7A00
","#FF7300"

,"#FF6C00","#FF6500","#FF5E00","#FF5700","#FF5000","#FF4900","#FF4200","#FF3B00","#FF3400",
"#FF2D00"

,"#FF2600","#FF1F00","#FF1800","#FF1100","#FF0A00","#FF0000","#F80000","#F10000","#EA0000","
#E30000"

,"#DC0000","#D50000","#CE0000","#C70000","#C00000","#B90000","#B20000","#AB0000","#A40000
","#9D0000"

);

$myArray=array_fill(0, $countpatient*2+6, null);

  $finalarray= array();
   array_push($finalarray,$mindate);
    array_push($finalarray,$lastday);
      array_push($finalarray,$disease_type);

   for($i=0;$i<$countdatetime;$i++)
   {$patient_id = array_search ($patient_Array[$i], $onlypatient);

       $myArray[0]=(int) $year_Array[$i];
       $myArray[1]= ((int)$month_Array[$i])-1;
       $myArray[2]=(int)$day_Array[$i];
       $myArray[3]=(int)$hour_Array[$i];
       $myArray[4]=(int)$minute_Array[$i];
       $myArray[5]=(int)$sec_Array[$i];
       $myArray[$patient_id*2+6]=$patient_id+1;
       $SelectedColor=$color[$eoi_Array[$i]];
       $myArray[$patient_id*2+6+1]=$SelectedColor;
       array_push($finalarray, $myArray);
    $myArray=array_fill(0, $countpatient*2+6, null);
   }
 echo json_encode(array_values($finalarray));
 unset($_SESSION['daytimedatepicker1']);
 unset($_SESSION['daytimedatepicker2']);
 unset($_SESSION['disease_type']);

mysqli_close($link);
?>
?>
```

(vi)Static Map View (Front end):

```
</script>
var map;
     function initMap() {
       map = new google.maps.Map(document.getElementById('map'), {
         zoom: 11,
         center: {lat:35.1495, lng: -89.8610},

         minZoom: 10,
         maxZoom: 12,
           });
    // Load GeoJSON.

var jsonData = $.ajax({
              url: "http://sccmobilehealth.com/research/admin/adminmapdate.php",
              dataType: "json",
              async: false
           }).responseText;


    var pp = JSON.parse(jsonData );
   var mindatestr=pp[0];                // pp[0] holds data for startdate
      var mindate=new Date(mindatestr);
      var min_year = mindate.getFullYear();
      var min_month=mindate.getMonth();
      var min_day=mindate.getDate();
      var min_hour =mindate.getHours();


     var lastdatestr=pp[1];              // pp[1] holds data for enddata
     var lastdate=new Date(lastdatestr);
     var last_year = lastdate.getFullYear();
     var last_month=lastdate.getMonth();
     var last_day=lastdate.getDate();
     var last_hour =lastdate.getHours();

     var disStartdate=min_month+1+"/"+min_day+"/"+min_year;
      var disEnddate=last_month+1+"/"+last_day+"/"+last_year;

     var disea_type=pp[2];

     document.getElementById('datepicker1').value=disStartdate;
      document.getElementById('datepicker2').value=disEnddate;
      document.getElementById('disease_type1').value=disea_type;

       map.data.loadGeoJson('http://sccmobilehealth.com/research/admin/adminjsonfromdbMap.php');

       map.data.setStyle(function(feature) {
         return /** @type {google.maps.Data.StyleOptions} */({
           fillColor: feature.getProperty('color'),
           strokeWeight: 1,

           strokeOpacity: 0.8,
```

```
        fillOpacity:feature.getProperty('fillOpacity'),
        font: '12px Verdana',
       text: feature.getProperty('areacode'),


      });
    });


 var setButton = document.getElementById("set");
  setButton.onclick=function()
  {map.data.loadGeoJson(
        'http://sccmobilehealth.com/research/admin/adminjsonfromdbMap.php');

     map.data.setStyle(function(feature) {
      return
      ({
        fillColor: feature.getProperty('color'),
        strokeWeight: 1,

        strokeOpacity: 0.8,
        fillOpacity:feature.getProperty('fillOpacity')

      });
    });

  };


   }

  </script>
  <script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC4LhavhXIRRnh04ToKVuFhsfQz3RSjOGw&
callback=initMap">
  </script>
```

(vii)Static Map View (Backend):

```
<?php
 session_start();
{
$link = mysqli_connect("192.185.4.125", "sscmemph_lab209", "123456", "sscmemph_emdb1");
$disease_type = "AR";

if(isset($_SESSION['date1'])&&isset($_SESSION['date2']))
{

            if(isset($_SESSION['disease_type1'])){
             $disease_type=$_SESSION['disease_type1'];
              }

             $sdatetime = new DateTime($_SESSION['date1'] );
             $sYear=$sdatetime->format('Y');
```

```
                $sMonth=$sdatetime->format('m');
                $sDay=$sdatetime->format('d');
                $start_datetime=$sYear."-".$sMonth."-".$sDay." 00:00:00";

                $edatetime = new DateTime($_SESSION['date2'] );
                $eYear=$edatetime->format('Y');
                $eMonth=$edatetime->format('m');
                $eDay=$edatetime->format('d');
                $end_datetime=$eYear."-".$eMonth."-".$eDay." 23:59:59";


}
  else
{
  $Sql2="SELECT MIN(datetime) as EarliestDate, MAX(datetime) as Lastdate
        FROM MIT_BHI_Record where patient not LIKE 'T%'";
        $result2=mysqli_query($link, $Sql2);
               while( $row = mysqli_fetch_assoc( $result2 ) )
                {
                 $start_datetime=$row["EarliestDate"];
                 $end_datetime =$row["Lastdate"];

                }

}

$sql="select avg(sub.avgeoi) as areaavgeoi, sub.area_code as areacode from
                        (select patient,area_code, avg(eoi) as avgeoi
                                  from MIT_BHI_Record
                                               where disease_type = '".$disease_type."'
                                               and ( datetime >= '".$start_datetime."' and datetime <=
'".$end_datetime."') and patient not LIKE 'T%'
                                               group by patient) sub
                                               group by sub.area_code";
                                              $result1=mysqli_query($link,$sql);


$area_code_Array=[];
$areaavgeoi=[];
 $length=0;
while( $row = mysqli_fetch_assoc( $result1 ) )
               {   $area_code_Array[]=$row["areacode"];
$areaavgeoi[]=intval($row["areaavgeoi"]*100);
                                              }
$length=count($area_code_Array);

//static 10 colors for 1-10 EOI

$color=array(
"#009D00","#00A400","#00AB00","#00B200","#00B900","#00C000","#00C700","#00CE00","#00D500",
"#00DC00","#00E300"

,"#00EA00","#00F100","#00F800","#00FF00","#07FF00","#0EFF00","#15FF00","#1CFF00","#23FF00","
#2AFF00"
```

```
,"#31FF00","#38FF00","#3FFF00","#46FF00","#4DFF00","#54FF00","#5BFF00","#62FF00","#69FF00",
"#70FF00"

,"#77FF00","#7EFF00","#85FF00","#8CFF00","#93FF00","#9AFF00","#A1FF00","#A8FF00","#AFFF00
","#B6FF00"

,"#BDFF00","#C4FF00","#CBFF00","#D2FF00","#D9FF00","#E0FF00","#E7FF00","#EEFF00","#F5FF0
0","#FFFF00"

,"#FFF800","#FFF100","#FFEA00","#FFE300","#FFDC00","#FFD500","#FFCE00","#FFC700","#FFC00
0","#FFB900"

,"#FFB200","#FFAB00","#FFA400","#FF9D00","#FF9600","#FF8F00","#FF8800","#FF8100","#FF7A00
","#FF7300"

,"#FF6C00","#FF6500","#FF5E00","#FF5700","#FF5000","#FF4900","#FF4200","#FF3B00","#FF3400",
"#FF2D00"

,"#FF2600","#FF1F00","#FF1800","#FF1100","#FF0A00","#FF0000","#F80000","#F10000","#EA0000","
#E30000"

,"#DC0000","#D50000","#CE0000","#C70000","#C00000","#B90000","#B20000","#AB0000","#A40000
","#9D0000"

);

$count_area=0;
$count_area=count($area_code_Array);

$area_color=[];
for($j=0;$j<$count_area;$j++)
{
		$area_color[$j]=$color[$areaavgeoi[$j]];
}

$obj = json_decode(file_get_contents("adminnewstaticjson.json"),true);
   $count_json=0;
   $count_json =count($obj['features']);

 for($i=0;$i<$count_json;$i++)
 {
     $color=$obj['features'][$i]['properties']['color'];
     $json_areacode=$obj['features'][$i]['properties']['areacode'];
     $index_area_code=array_search($json_areacode, $area_code_Array);
 /// get the color index from jsonareacode. give the respective color to that area
   if(is_int($index_area_code))
  {$obj['features'][$i]['properties']['color']=$area_color[$index_area_code];	}
 }

echo json_encode($obj);
}
unset($_SESSION['date1']);
unset($_SESSION['date2']);
unset($_SESSION['disease_type1']);

?>
```

(viii)Animation of map(Frontend):

```
function
animateCircle(pAZ19,pZG86,pXP52,pHD93,pFD32,pSP71,pKT43,pBY28,pYU76L,pYU76Z,pYU76K,pY
U76B,pDW46,pWG49,pLC95,pDR27,pVJ14,pCX38,pCR63,ttyme) {

    //................. initiation.........................//
   if(flaganim==1)
     {
       flaganim=0;
      var ttime=document.getElementById("time").value
     //.................... stop button declare........................//
     var stopButton = document.getElementById("stop");
     var animdate = document.getElementById("animdate");
       // ..................call the json.................//
       var jsonData = $.ajax({
              url: "http://sccmobilehealth.com/research/admin/AdminMapAnimationBack.php",
               dataType: "json",
               async: false
           }).responseText;

      var pp = JSON.parse(jsonData);
      var ppLength=pp.length;
   var count1 = 0;
       //..............Progress bar...........//
      var elem = document.getElementById("myBar");
      var width = 1;
      var widthincrease = 100/ppLength;
    //// ...............animation......................./
  var idf =  window.setInterval(function() {

    stopButton.onclick=function(){
     console.log("stop");
     clearInterval(idf);

      stop();
      flaganim=1;
      }
   if(ppLength<=count1){
    console.log("stop");
    clearInterval(idf);
    flaganim=1;
     }
    else{
              //.................color change.............//
              pAZ19.set("fillColor",pp[count1][0]);
              pZG86.set("fillColor",pp[count1][1]);
              pXP52.set("fillColor",pp[count1][2]);
              pHD93.set("fillColor",pp[count1][3]);
              pFD32.set("fillColor",pp[count1][4]);
              pSP71.set("fillColor",pp[count1][5]);
              pKT43.set("fillColor",pp[count1][6]);
```

```
            pBY28.set("fillColor",pp[count1][7]);
            pYU76L.set("fillColor",pp[count1][8]);
            pYU76Z.set("fillColor",pp[count1][9]);
            pYU76K.set("fillColor",pp[count1][10]);
            pYU76B.set("fillColor",pp[count1][11]);
            pDW46.set("fillColor",pp[count1][12]);
            pWG49.set("fillColor",pp[count1][13]);
            pLC95.set("fillColor",pp[count1][14]);
            pDR27.set("fillColor",pp[count1][15]);
            pVJ14.set("fillColor",pp[count1][16]);
            pCX38.set("fillColor",pp[count1][17]);
            pCR63.set("fillColor",pp[count1][18]);

      width=width+widthincrease ;
      elem.style.width = width + '%';

     var twodates=pp[count1][19]+"---"+pp[count1][20];
            document.getElementById('animdate').value=twodates;

      count1=count1+1;
       }

            }, ttyme);
            //..........end animation....................///
            }
            else
            {console.log("flag......");}
  }
 //................end the animation function..................//

   <script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC4LhavhXIRRnh04ToKVuFhsfQz3RSjOGw&
callback=initMap">
   </script>
```

(ix)Animation of map (Backend):

```
<?php
$animtype="cumulative";
      if(isset($_SESSION['animtype']))
      {$animtype=$_SESSION['animtype'];
      }
$t=1;
if(isset($_SESSION['dateinterval']))
      {$t=$_SESSION['dateinterval'];
      }
$date1=strtotime($start_datetime);

$ind="+".$t." day";
$date2=strtotime($ind,$date1);
$date3=date("Y-m-d", $date2);

$conditiondate=$t-1;
$ddddd="+".$conditiondate." day";
```

```
$sttimeEndDate=strtotime($end_datetime);
$timeEndDate=strtotime($ddddd,$sttimeEndDate);
$sdate=$start_datetime;
$finalarray=array();
for(;;)
{
 if($date2>$sttimeEndDate)
 {
  break;}
 else{

   $sql="select avg(sub.avgeoi) as areaavgeoi, sub.area_code as areacode from
                              (select patient,area_code, avg(eoi) as avgeoi
                                          from MIT_BHI_Record
                                                      where disease_type = '".$disease_type."'
                                                      and patient not LIKE 'T%'
                                                      and ( datetime >= '".$sdate."' and datetime <= '".$date3."')
                                                      group by patient) sub
                                                      group by sub.area_code";

                                                     $result1=mysqli_query($link,$sql);



$area_code_Array=[];
$areaavgeoi=[];
$length=0;
while( $row = mysqli_fetch_assoc( $result1 ) )
                                                      {
                                           $area_code_Array[]=$row["areacode"];
                                           $areaavgeoi[]=intval($row["areaavgeoi"]*100);
                                           areaavgeoi= ".$row["areaavgeoi"]*100;
                                                       }
$length=count($area_code_Array);

$color=array(
"#009D00","#00A400","#00AB00","#00B200","#00B900","#00C000","#00C700","#00CE00","#00D500",
"#00DC00","#00E300"

,"#00EA00","#00F100","#00F800","#00FF00","#07FF00","#0EFF00","#15FF00","#1CFF00","#23FF00","
#2AFF00"

,"#31FF00","#38FF00","#3FFF00","#46FF00","#4DFF00","#54FF00","#5BFF00","#62FF00","#69FF00",
"#70FF00"

,"#77FF00","#7EFF00","#85FF00","#8CFF00","#93FF00","#9AFF00","#A1FF00","#A8FF00","#AFFF00
","#B6FF00"

,"#BDFF00","#C4FF00","#CBFF00","#D2FF00","#D9FF00","#E0FF00","#E7FF00","#EEFF00","#F5FF0
0","#FFFF00"

,"#FFF800","#FFF100","#FFEA00","#FFE300","#FFDC00","#FFD500","#FFCE00","#FFC700","#FFC00
0","#FFB900"

,"#FFB200","#FFAB00","#FFA400","#FF9D00","#FF9600","#FF8F00","#FF8800","#FF8100","#FF7A00
","#FF7300"
```

```
,"#FF6C00","#FF6500","#FF5E00","#FF5700","#FF5000","#FF4900","#FF4200","#FF3B00","#FF3400",
"#FF2D00"

,"#FF2600","#FF1F00","#FF1800","#FF1100","#FF0A00","#FF0000","#F80000","#F10000","#EA0000","
#E30000"

,"#DC0000","#D50000","#CE0000","#C70000","#C00000","#B90000","#B20000","#AB0000","#A40000
","#9D0000"

);

$count_area=0;
$count_area=count($area_code_Array);

$area_color=[];
for($j=0;$j<$count_area;$j++)
{$area_color[$j]=$color[$areaavgeoi[$j]];
}

$all_area_code_array=array("AZ19","ZG86","XP52","HD93","FD32","SP71","KT43","BY28","YU76L","
YU76Z","YU76K","YU76B","DW46","WG49","LC95","DR27","VJ14","CX38","CR63");

$all_area_color=array("#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#C
FD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#
CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC","#CFD8DC");


for($i=0;$i<$count_area;$i++)
{
  $single_code_area= $area_code_Array[$i];

$index_in_all_area_code_array=array_search($single_code_area,$all_area_code_array);

  $all_area_color[$index_in_all_area_code_array]=$area_color[$i];

}
$strtime=strtotime($sdate);
$ssdate=date("Y-m-d", $strtime);

$all_area_color[19]=$ssdate;
$all_area_color[20]=$date3;

array_push($finalarray,$all_area_color);

if($animtype=='segmented')
{$sdate=$date3;}
$d1=strtotime($date3);
$ind="+".$t." day";
$date2=strtotime($ind,$d1 );
$date3=date("Y-m-d", $date2);

if($date2>$sttimeEndDate&&$flag)
{
  $date2=$sttimeEndDate;
  $date3=date("Y-m-d", $sttimeEndDate);
  $flag=0;
```

```
}

}
}
echo json_encode(array_values($finalarray));
mysqli_close($link);

?>
```

(x)Static GeoJSON:

{"type":"FeatureCollection","features":[{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-
90.309543,34.922824],[-90.085642,34.922824],[-90.085642,35.0317400825],[-
90.309543,35.0317400825],[-
90.309543,34.922824]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"AZ19"}},{"type
":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.085642,34.922824],[-
89.861741,34.922824],[-89.861741,35.0317400825],[-90.085642,35.0317400825],[-
90.085642,34.922824]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"ZG86"}},{"type
":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.86174,34.922824],[-
89.861741,35.0317400825],[-89.63784,35.0317400825],[-89.63784,34.922824],[-
89.86174,34.922824]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"XP52"}},{"type"
:"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.63784,35.0317400825],[-
89.861741,35.0317400825],[-89.861741,35.140656165],[-89.63784,35.140656165],[-
89.63784,35.0317400825]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"HD93"}},{"
type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.309543,35.0317400825],[-
90.085642,35.0317400825],[-90.085642,35.140656165],[-90.309543,35.140656165],[-
90.309543,35.0317400825]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"FD32"}},{
"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.309543,35.140656165],[-
90.085642,35.140656165],[-90.085642,35.2495722475],[-90.309543,35.2495722475],[-
90.309543,35.140656165]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"SP71"}},{"t
ype":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.085642,35.140656165],[-
89.861741,35.140656165],[-89.861741,35.2495722475],[-90.085642,35.2495722475],[-
90.085642,35.140656165]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"KT43"}},{"t
ype":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.861741,35.2495722475],[-
89.861741,35.140656165],[-89.63784,35.140656165],[-89.63784,35.2495722475],[-
89.861741,35.2495722475]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"BY28"}},{
"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.085642,35.0317400825],[-
89.9736915,35.0317400825],[-89.9736915,35.08619812375],[-90.085642,35.08619812375],[-
90.085642,35.0317400825]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"YU76L"}}
,{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.085642,35.08619812375],[-
90.085642,35.140656165],[-89.9736915,35.140656165],[-89.9736915,35.08619812375],[-
90.085642,35.08619812375]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"YU76Z"}
},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.9736915,35.140656165],[-
89.861741,35.140656165],[-89.861741,35.08619812375],[-89.9736915,35.08619812375],[-
89.9736915,35.140656165]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"YU76K"}}
,{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.9736915,35.08619812375],[-
89.861741,35.08619812375],[-89.861741,35.0317400825],[-89.9736915,35.0317400825],[-
89.9736915,35.08619812375]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"YU76B"
}},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.63784,34.922824],[-
89.63784,35.0317400825],[-89.413939,35.0317400825],[-89.413939,34.922824],[-
89.63784,34.922824]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"DW46"}},{"type
":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.63784,35.0317400825],[-
89.63784,35.140656165],[-89.413939,35.140656165],[-89.413939,35.0317400825],[-
89.63784,35.0317400825]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"WG49"}},{
"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.63784,35.140656165],[-
89.63784,35.2495722475],[-89.413939,35.2495722475],[-89.413939,35.140656165],[-

89.63784,35.140656165]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"LC95"}},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.63784,35.2495722475],[-89.63784,35.35848833],[-89.413939,35.35848833],[-89.413939,35.2495722475],[-89.63784,35.2495722475]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"DR27"}},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-89.861741,35.2495722475],[-89.861741,35.35848833],[-89.63784,35.35848833],[-89.63784,35.2495722475],[-89.861741,35.2495722475]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"VJ14"}},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.085642,35.2495722475],[-90.085642,35.35848833],[-89.861741,35.35848833],[-89.861741,35.2495722475],[-90.085642,35.2495722475]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"CX38"}},{"type":"Feature","geometry":{"type":"Polygon","coordinates":[[[-90.309543,35.35848833],[-90.085642,35.35848833],[-90.085642,35.2495722475],[-90.309543,35.2495722475],[-90.309543,35.35848833]]]},"properties":{"color":"#CFD8DC","fillOpacity":0.5,"areacode":"CR63"}}]}

B. Code for Arrhythmia Severity Ranking

(i)CollectBeat.m

```
r=[100,101,102,103,104,105,106,107,108,109,111,112,113,114,115,116,117,118,119,121,122,123,124,200
,201,202,203,205,207,208,209,210,212,213,214,215,217,219,220,221,222,223,228,230,231,232,233,234];

 for k=1:48
    rName=[num2str(r(k)),'_ECG_heartbeat_classifier.mat'];
    load(rName);
    len=length(anntyp);
    siz=ceil(len/500);
    record_part=zeros(siz,5);
    e_ind=0;
    f_ind=1;
    for i=1:siz
     if(i==siz)
           e_ind=len;
     else
        e_ind=e_ind+500;
     end
      part=anntyp(f_ind:e_ind)
      N=sum(part=='N')
      F=sum(part=='F')
      S=sum(part=='S')
      V=sum(part=='V')
      record_part(i,:)=[r(k) N F S V];
%       f_ind=e_ind+1;
    end
    filelink=['C:\Users\safroz\OneDrive - The University of
Memphis\research\recordAddition\',num2str(r(k)),'.csv'];
    csvwrite(filelink,record_part);

    dlmwrite('C:\Users\safroz\OneDrive - The University of
Memphis\research\recordAddition\ALLRecord.csv',record_part,'delimiter',',','-append');
end
```

(ii)calculateAlpha.m

```
function alphaM=calculateAlpha(model)

  sumM=sum(model,2);
  sizeM=size(model,2);
  alphaM=zeros(1,sizeM);

   for i=1:sizeM
       alphaM(i)=(model(i)*100)/sumM;
   end
end
```

(iii)generateAlpha.m

```
function alpha=generateAlpha
    tenpowerM=[power(10,1),power(10,2),power(10,3),power(10,4)];
    tenpowerA=calculateAlpha(tenpowerM);
```

```
    multy25M=[25,50,75,100];
    multy25A=calculateAlpha(multy25M);

    power2M=[power(2,2),power(2,4),power(2,6),power(2,8)];
    powerA=calculateAlpha(power2M);

    cubicM=[power(2,3),power(3,3),power(4,3),power(5,3)];
    cubicA=calculateAlpha(cubicM);

    expM=[ exp(2),exp(3),exp(4),exp(5)];
    expA=calculateAlpha(expM);

    sqrtM=[sqrt(.1),sqrt(1), sqrt(10),sqrt(100)];
    sqrtA=calculateAlpha(sqrtM);

    fourpowerM= [power(4,2),power(4,3),power(4,4),power(4,5)];
    fourpowerA=calculateAlpha(fourpowerM);

    powertenM=[power(2,10),power(3,10),power(4,10),power(6,10)];
    powertenA=calculateAlpha(powertenM);

    inversetenthM=[.1,1,10,100];
    inversetenthA=calculateAlpha(inversetenthM);

    tangentM=[tand(1), tand(22.5), tand(45),tand(67.5)];
    tangentA=calculateAlpha(tangentM);

    onethird=[0.001,33.33,66.66,99.99];
    onthirdM=calculateAlpha(onethird);

    %alpha matrix
   c = categorical({'Alpha1','Alpha2','Alpha3','Alpha4'});

alpha=[tenpowerA;multy25A;powerA;cubicA;expA;sqrtA;fourpowerA;powertenA;inversetenthA;tangentA
;onthirdM]';

    plot(c,alpha)

legend('tenpower','multy25','power2','cubic','exp','sqrt','fourpower','powerten','inversetenth','tangent','onthird
')
    %csvwrite('alpha.csv',alpha);
    ylabel('Normalized Value')
    xlabel('Alpha value')
end
```

(iv)AllSeverityNorm.m

```
a = csvread('C:\Users\safroz\OneDrive - The University of Memphis\research\except\AllSeverity.csv');

ALL_max=max(a)';

model_max=ALL_max;

len = length(model_max);
sr=size(a,1);
sc=size(a,2);
model = zeros(sr,sc);
```

```
for i=1:len
    model(:,i) = a(:,i)/model_max(i)*100;
end
csvwrite('C:\Users\safroz\OneDrive - The University of Memphis\research\except\ALLSeverityNorm.csv',model);
```

(v)checkweight.m

```
function indweight= checkweight(data,w)
% figure('name','real data')
% bar(x,y)
% legend('N', 'F', 'S', 'V');


record=data(:,1);
data(:,1) =[];
ml=size(data,2);
data(:,ml)=[];
x=categorical(record);

a = sum(data,2);

rowcount=size(data,1);
columncount=size(data,2);
avgmat=zeros(rowcount,columncount);

for i=1:rowcount
    for j=1:columncount
    avgmat(i,j)=data(i,j)/a(i);
    end
end

mw1=zeros();
for i=1:rowcount
    for j=1:columncount
    mw1(i,j)=avgmat(i,j)*w(j);
    end
end


tw1=avgmat*w;
indweight=tw1;
end
```

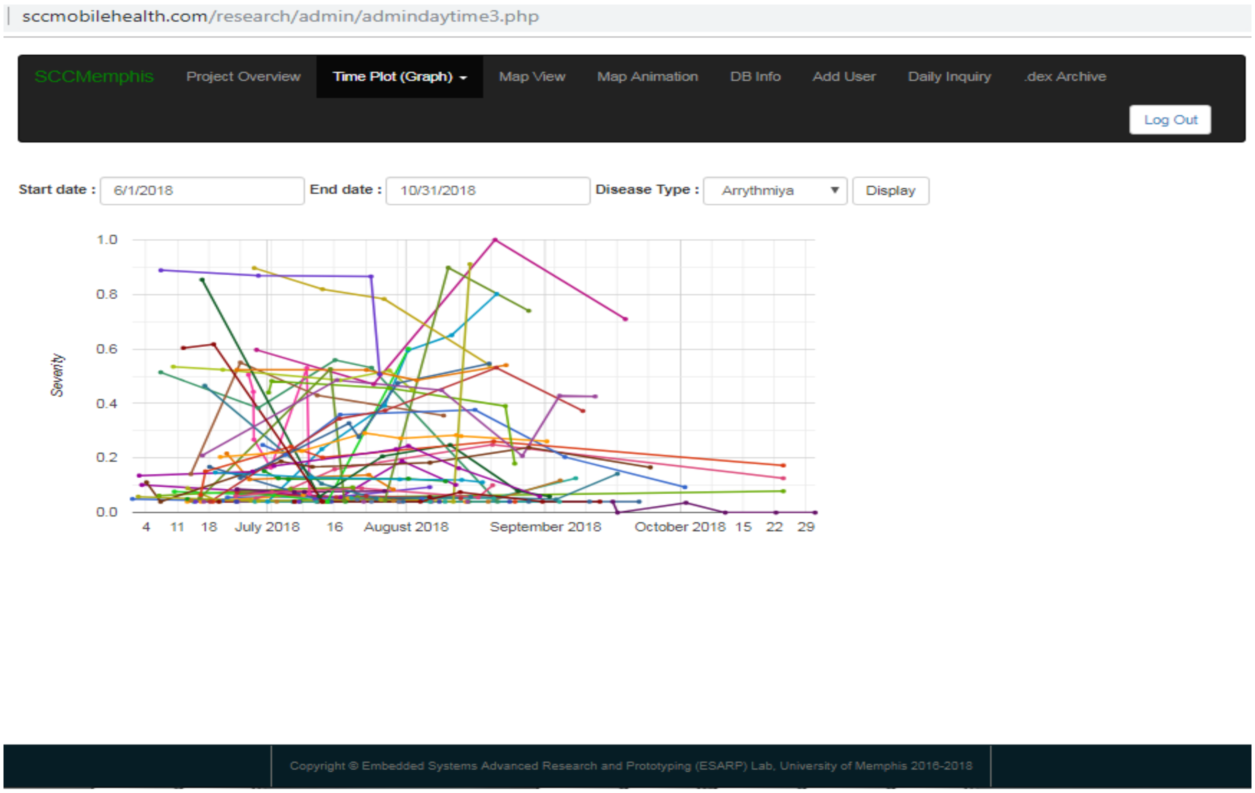
(vi) recordweight.m

```
normaldata = xlsread('C:\Users\safroz\OneDrive - The University of Memphis\research\reocrod with serverity\normal.xlsx');
milddata = xlsread('C:\Users\safroz\OneDrive - The University of Memphis\research\reocrod with serverity\mild.xlsx');
moderatedata = xlsread('C:\Users\safroz\OneDrive - The University of Memphis\research\reocrod with serverity\moderate.xlsx');
severedata = xlsread('C:\Users\safroz\OneDrive - The University of Memphis\research\reocrod with serverity\severe.xlsx');
```

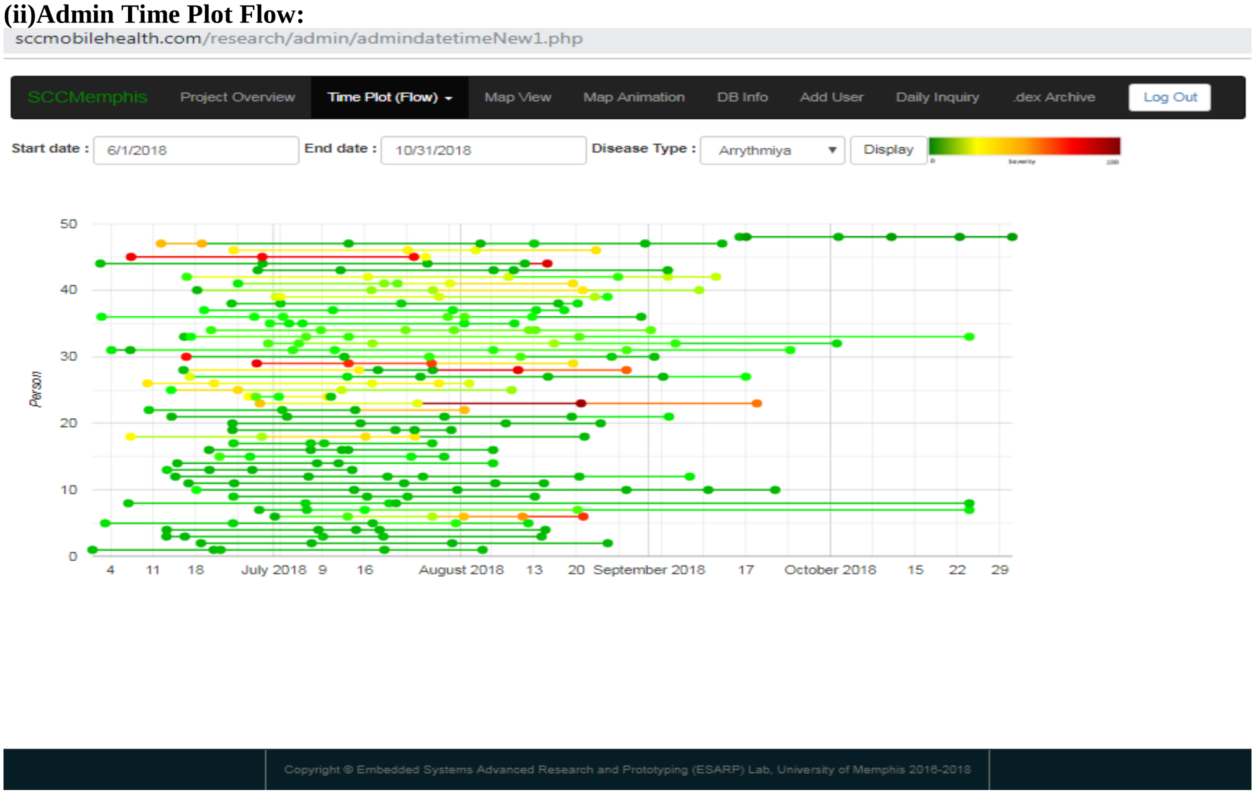
```
veryseveredata = xlsread('C:\Users\safroz\OneDrive - The University of Memphis\research\reocrod with serverity\very severe.xlsx');

w= generateAlpha();
rw=size(w,1);
rc=size(w,2);
indweightnormal=zeros(size(normaldata,1),rc);
indweightMild =zeros(size(milddata,1),rc);
indweightmoderate =zeros(size(moderatedata,1),rc);
indweightsevere =zeros(size(severedata,1),rc);
indweightverysevere =zeros(size(veryseveredata,1),rc);

for i=1:rc
   %normal
   indweightnormal(:,i)= checkweight(normaldata,w(:,i));
   %mild
   indweightMild(:,i) = checkweight(milddata,w(:,i));
   %moderate
   indweightmoderate(:,i) = checkweight(moderatedata,w(:,i));
   %severe
   indweightsevere(:,i) = checkweight(severedata,w(:,i));
   %very severe
   indweightverysevere(:,i) = checkweight(veryseveredata,w(:,i));

end

normal_max_min=[min(indweightnormal);max(indweightnormal)]';
mild_max_min=[min(indweightMild);max(indweightMild)]';
moderate_max_min=[min(indweightmoderate);max(indweightmoderate)]';
severe_max_min=[min(indweightsevere);max(indweightsevere)]';
very_severe=[min(indweightverysevere);max(indweightverysevere)]';

len=size(normal_max_min,2)

model_max_min=[normal_max_min mild_max_min moderate_max_min severe_max_min very_severe]

csvwrite('myFile.csv',model_max_min);
plot(w);

legend('power10','multy25','power2','cubic','exp','sqrt','fourpower','powerTen','inverseTenth','tangent')
```

C. Screen shot of SCCHealth web server

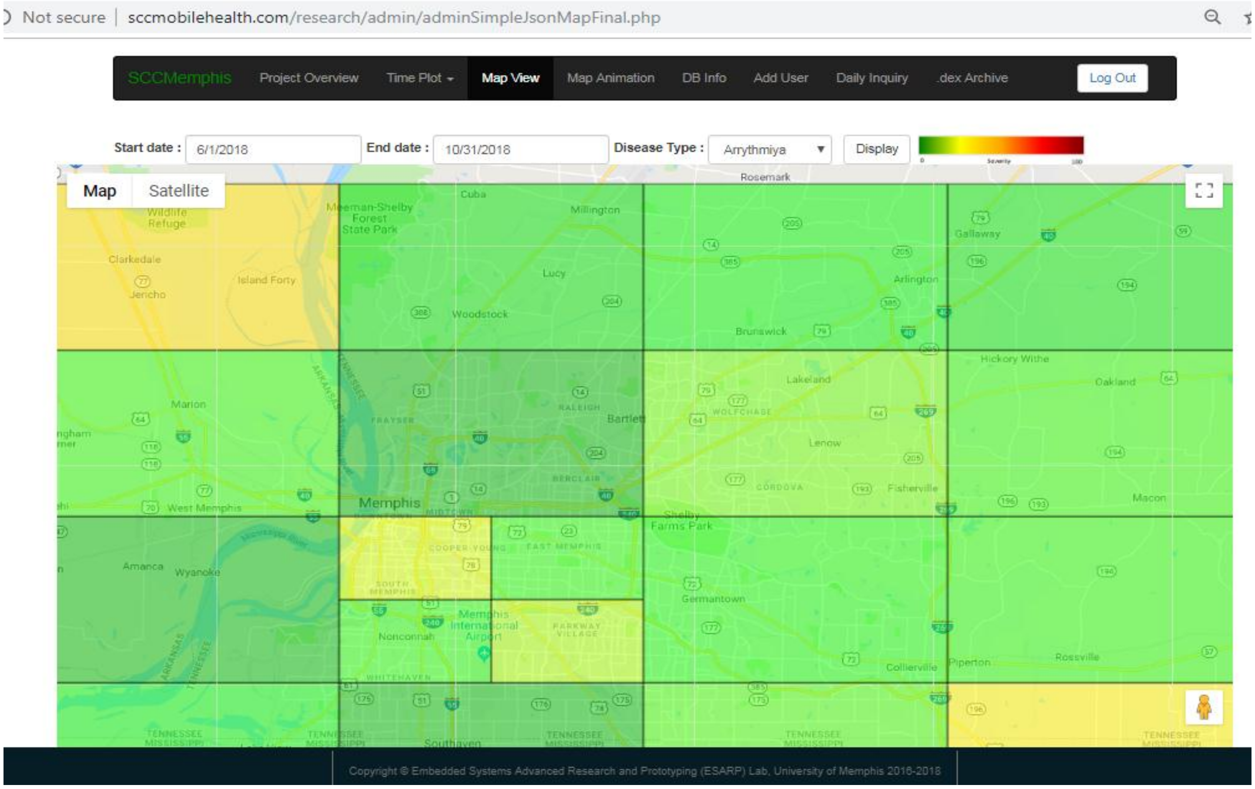
(i)Admin Time plot Graph:



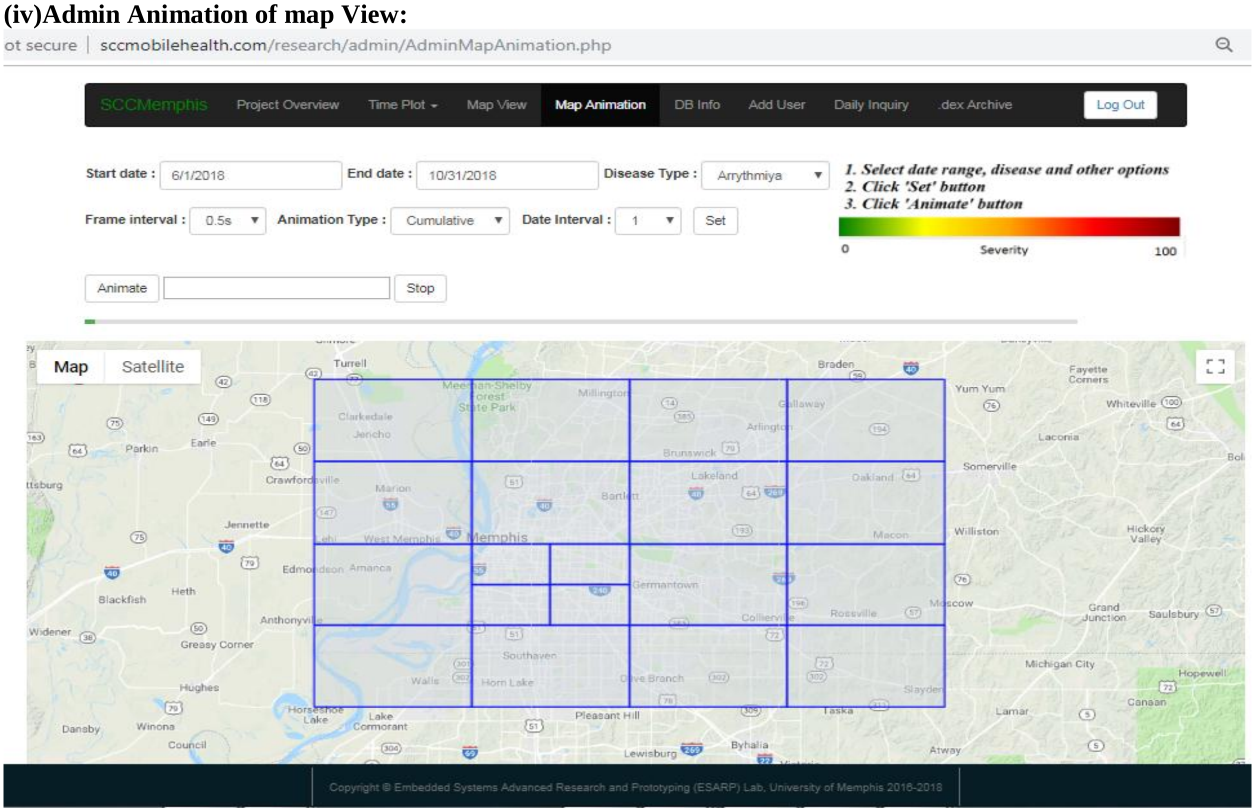
(ii)Admin Time Plot Flow:



(iii)Admin Static map View:



(iv)Admin Animation of map View:



(v)Admin Database info:

sccmobilehealth.com/research/admin/adminAllPatient.php

SCCMemphis | Project Overview | Time Plot ▾ | Map View | Map Animation | DB Info | Add User | Daily Inquiry | .dex Archive | Log Out

The Database info of all patients

Refresh

Download as CSV

Serial No.	Database Id	Patient	Disease_type	EOI	Area code	Date and Time	Algorithm
0	245	234	AR	1.0041E-5	CX38	2018-10-31 02:12:00.000000	AR1
1	34	106	AR	0.12597	AZ19	2018-10-24 00:00:00.000000	AR1
2	37	107	AR	0.0786065	VJ14	2018-10-24 00:00:00.000000	AR1
3	164	212	AR	0.172875	BY28	2018-10-24 00:00:00.000000	AR1
4	244	234	AR	1.0041E-5	CX38	2018-10-22 09:34:50.000000	AR1
5	243	234	AR	1.0041E-5	CX38	2018-10-11 02:56:40.000000	AR1
6	242	234	AR	0.036155	CX38	2018-10-02 08:34:30.000000	AR1
7	158	210	AR	0.093281	XP52	2018-10-02 02:07:25.000000	AR1
8	152	209	AR	0.16595	LC95	2018-09-24 09:23:23.000000	AR1
9	50	109	AR	0.0401545	FD32	2018-09-21 21:23:23.000000	AR1
10	108	124	AR	0.709385	CR63	2018-09-18 20:56:25.000000	AR1
11	241	234	AR	1.0041E-5	CX38	2018-09-17 02:45:20.000000	AR1
12	129	203	AR	0.1421	DR27	2018-09-17 01:06:29.000000	AR1
13	240	234	AR	0.0401545	CX38	2018-09-16 00:35:25.000000	AR1
14	239	233	AR	0.0401545	YU76K	2018-09-13 03:15:02.000000	AR1

(vi)Admin daily inquiry:

sccmobilehealth.com/research/admin/dailyInquiry.php

SCCMemphis | Project Overview | Time Plot ▾ | Map View | Map Animation | DB Info | Add User | Daily Inquiry | .dex Archive | Log Out

The Daily Inquiry

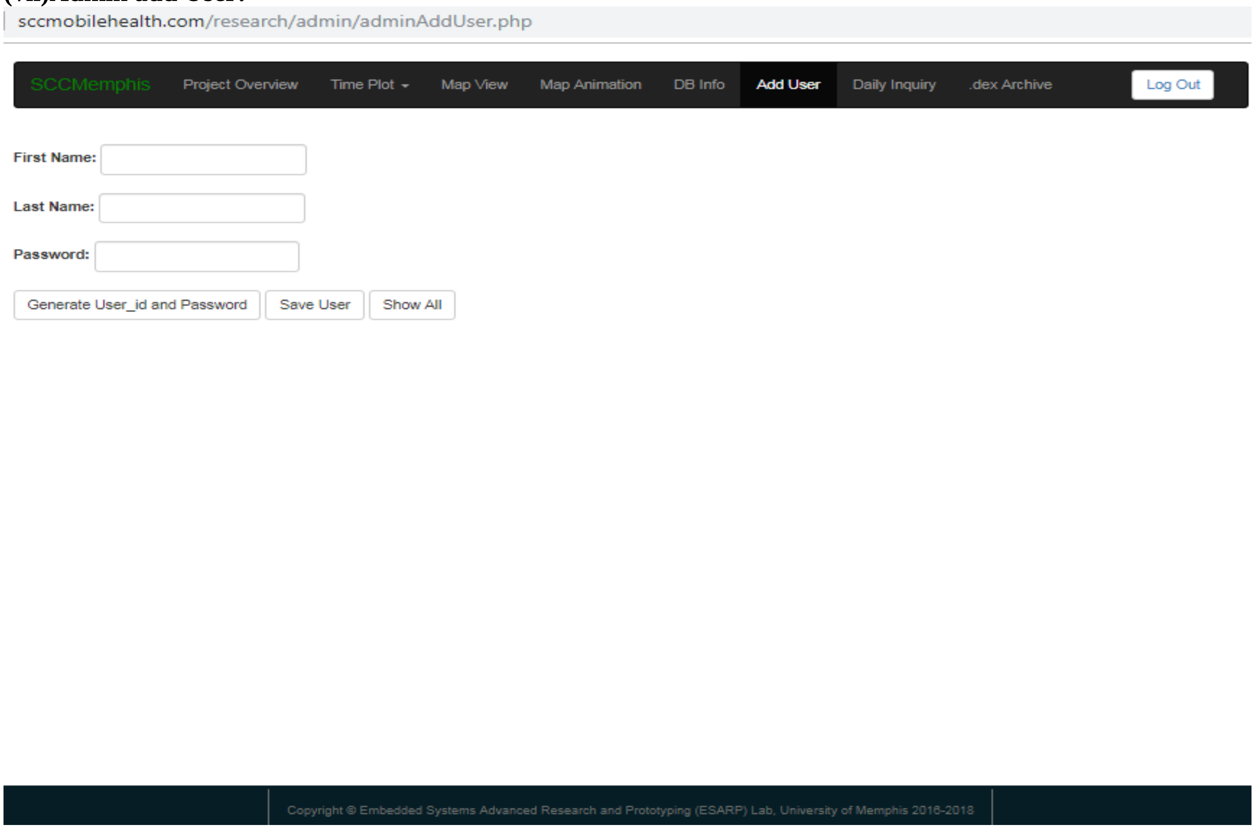
First day of Inquriy : Last day of Inquiry : patientID : All ▾ Search

The patients' information between **07/04/2018** to **07/07/2018** :

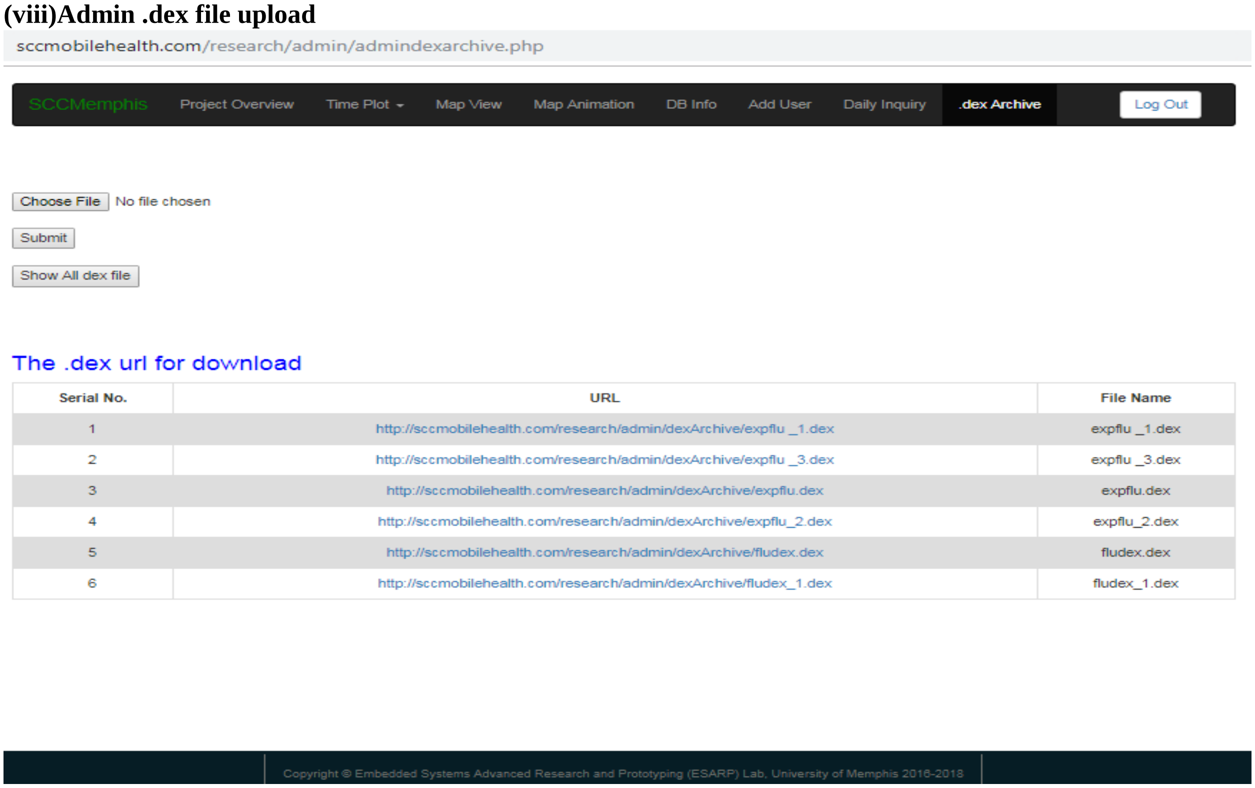
Download as CSV

Serial No.	Database Id	Patient	Disease_type	EOI	Area code	Date and Time	Algorithm
0	148	209	AR	0.18756	LC95	2018-07-04 03:45:38.000000	AR1
1	154	210	AR	0.21179	XP52	2018-07-05 03:34:24.000000	AR1
2	174	214	AR	0.12262	XP52	2018-07-05 18:07:28.000000	AR1
3	38	107	AR	0.086551	VJ14	2018-07-06 06:13:23.000000	AR1
4	161	212	AR	0.24167	BY28	2018-07-06 08:12:23.000000	AR1
5	32	106	AR	0.089191	AZ19	2018-07-06 11:23:43.000000	AR1
6	57	112	AR	0.0401545	LC95	2018-07-06 18:03:23.000000	AR1
7	75	116	AR	0.0401545	XP52	2018-07-07 02:56:25.000000	AR1
8	80	117	AR	0.0755175	YU76L	2018-07-07 02:56:46.000000	AR1
9	7	101	AR	0.0401545	CX38	2018-07-07 06:11:23.000000	AR1

(vii)Admin add User:



(viii)Admin .dex file upload



Serial No.	URL	File Name
1	http://sccmobilehealth.com/research/admin/dexArchive/expflu _1.dex	expflu _1.dex
2	http://sccmobilehealth.com/research/admin/dexArchive/expflu _3.dex	expflu _3.dex
3	http://sccmobilehealth.com/research/admin/dexArchive/expflu.dex	expflu.dex
4	http://sccmobilehealth.com/research/admin/dexArchive/expflu_2.dex	expflu_2.dex
5	http://sccmobilehealth.com/research/admin/dexArchive/fludex.dex	fludex.dex
6	http://sccmobilehealth.com/research/admin/dexArchive/fludex_1.dex	fludex_1.dex

(ix)User Home page:

sccmobilehealth.com/research/user/usertest.php

SCCMemphis | Project Overview | Time Plot (Graph) | Time plot (Flow) | Map View | Map Animation | Log Out

NSF Project Web-portal

Title: EAGER: Events-of-interest Capture Using Novel Body-worn Fully-passive Wireless sensors for SCC

Funding Source: NSF CISE CNS

Project Duration: 2016 - 2018

Project Synopsis:

This NSF funded project aims to develop a new class of battery-less, low-cost, disposable, wireless electronic patch sensors to capture a variety of physiological signals that allows monitoring of their health status through a custom smartphone app and enables sharing of their anonymized health-related events-of-interest towards a smart and connected community (SCC). This will empower users, permit the community stakeholders to assess population health status, reduce the need for frequent hospital visits, and help identify potential individual and community actions to achieve improvement in health status. The project also involves the training of undergraduate and graduate students in interdisciplinary research activities on emerging technologies, and is expected to impact public and private sector efforts to improve healthcare.

Project Personnel:

PI: Dr. Bashir Morshed, Associate Professor, Department of Electrical and Computer Engineering, The University of Memphis

Co-PI: Dr. Brook Harmon, Assistant Professor, School of Public Health, The University of Memphis

Consultant: Dr. M. Rahman, Baptist Minor Medical Center, Memphis, TN

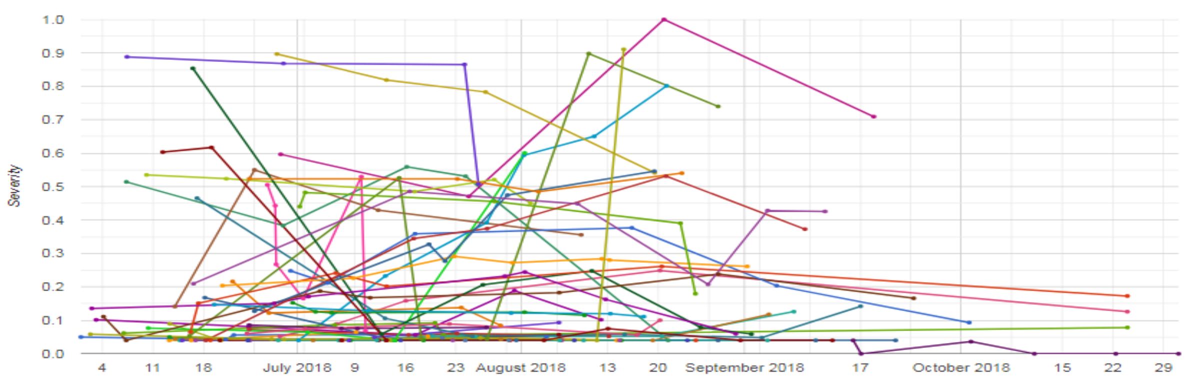
Collaborator: Memphis District of the United Methodist Church (UMC)

(x)User Time plot Graph:

sccmobilehealth.com/research/user/userdaytime3.php

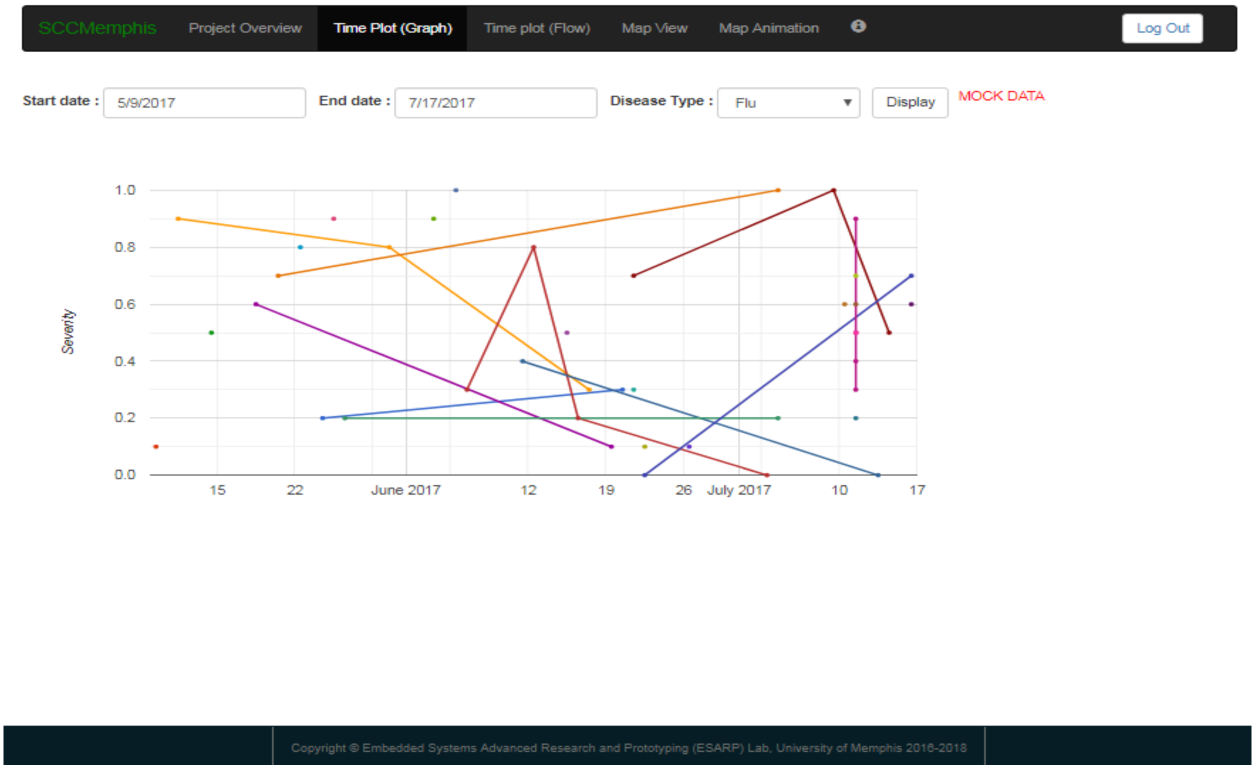
SCCMemphis | Project Overview | Time Plot (Graph) | Time plot (Flow) | Map View | Map Animation | Log Out

Start date : 6/1/2018 **End date :** 10/31/2018 **Disease Type :** Arrythmiya Display



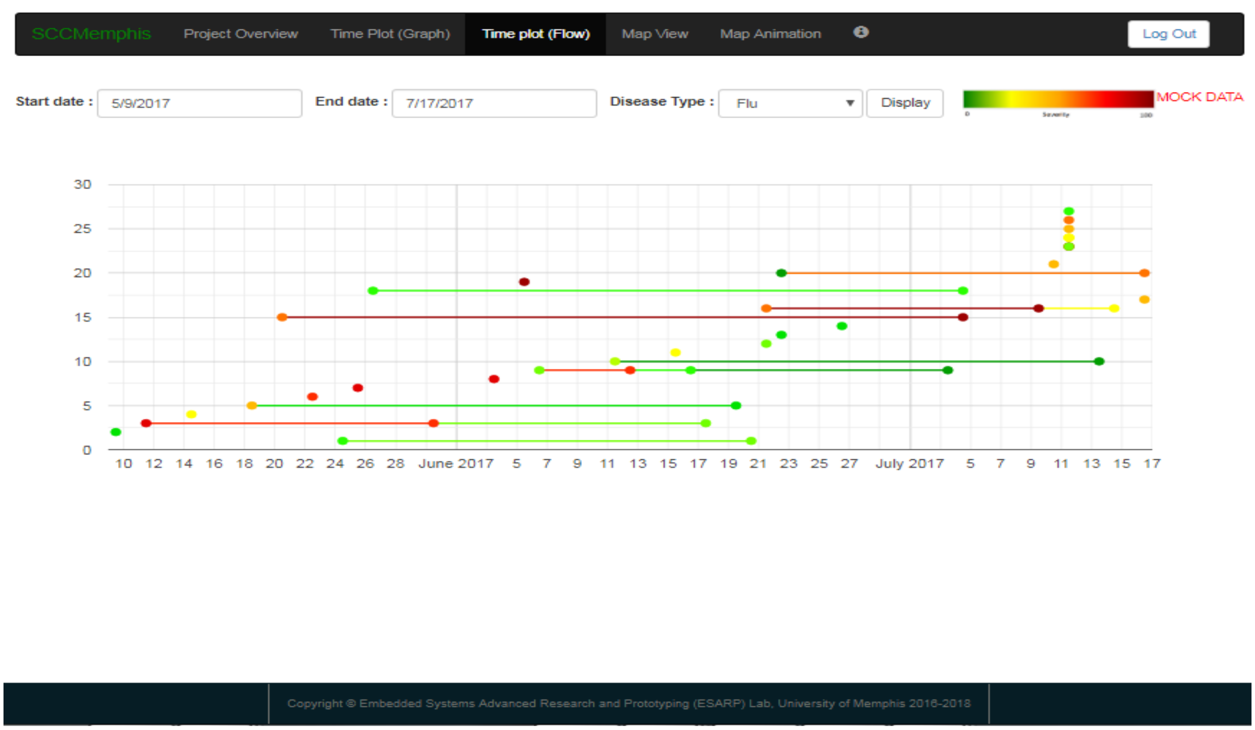
(xi)Guest Time plot Graph:

sccmobilehealth.com/research/guest/daytime3.php



(xii)Guest Time plot Flow:

sccmobilehealth.com/research/guest/datetimeNew1.php



(xiii)Guest Animation: