Database Management system of Emergency Response Drones equipped with defibrillator for reviving human heart

A Project Report submitted by

Darshak Shah, Devarsh Shah, Sreevathsan Athinarayanan, Vinod Thiagarajan

Northeastern University, Boston

In partial fulfillment

for the course, of

INFO 6210

Database Management and Database Design Report

Authors Note

We sincerely thank **Prof Chaiyaporn Mutsalklisana** for accepting our proposal to build a database management system along with a front-end system to be used in a real-time environment. We also thank TA's Kaushik Padmanabhan, Thiyagarajan Anandan for helping us during project by ensuring logical flow, innovativeness in the project and pushing our problem-solving methodologies. We hereby grant permission to use (and adapt) this document for learning and research purposes. You may not sell this document either by itself or in combination with other products or services. Third, if you use this document, you use it at your own risk. The document's accuracy and safety have been thoroughly evaluated, but they are not guaranteed.

Abstract

A database management system has been developed for an emergency response drone to treat patients who have heart failures or are likely to have heart failures in about half an hour for persons living in remote locations as well. The current available drone system or medical help system doesn't cater to remote locations or doesn't predict heart failures in advance. The system also sends a hospital team or an air ambulance to the location of the patient to recover the heart completely, as the defibrillator can only temporarily revive the heart. The system calculates heart factor based on age, heart rate, systolic pressure and diastolic pressure with references from American Heart Association. Defibrillator uses adhesive pads which are much better and the use for strips has been eliminated. Path of traversing has been optimized as we used Dijkstra's algorithm to help our drones reach the patient in a faster way. Many other problems which the drone system faces were addressed through this project or at least were attempted to solve.

Keywords: defibrillator, heart rate, systolic pressure, diastolic pressure, Dijkstra

Table of Contents

INTRODUCTION

The diagram below indicates the logical flow of the project with the person data being fed in by patient who uses the device which measures both hear rate and pressure. Hence the device can be used my multiple users. The person’s health details along with location are lively transmitted to the base station which monitor’s person heart rate by calculating heart risk factor and predicting heart failures before half an hour of the actual failure. The drone is then triggered off from the deployment centre to the patient through an optimized path. The patient’s emergency contact is alerted and communities nearby are alerted to reach the patient. The Defibrillator is equipped with ECG and intelligent enough to distinguish the type of heart attack and provide necessary energy to the patient.The system has IR sensor to detect the patient’s location during night as themographs can easily detect human body based on temperature variations. Additionally we have weather station to tell if its snowing or raining in cases when drone may not work so that we can send an air ambulance or a hospital team to the patient’s location.

PROBLEM DEFINITION AND MOTIVATION

The total number of deaths occurring due to heart failure is around 17 million per year as depicted below. Each year ambulance services save around just 8 percent of them. Our solution of drones travelling with speed of 35mph and reaching destination within 3-5 minutes helps increase survival rate to 80 percent. Apart from this drone technology will continue to grow with increased networks worldwide in providing internet, delivering medicines and groceries, evade traffic in emergency situations, improving safety in difficult to reach places, aerial photography making our lives bigger and better.

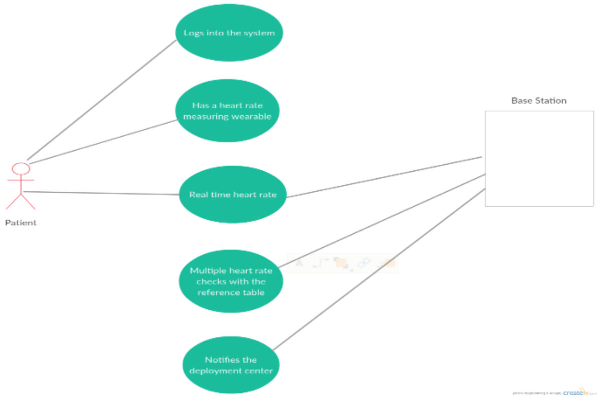
EXISTING AND PROPOSED SOLUTION

Existing Solution	Proposed Solution
1)It doesn’t have a heart failure prediction system to detect or predict failures even before actual occurrence 1) Some other person is required to call for help 2) The drone reaches only after patient is fainted 3) System exists only in urban areas 4) Technology is very costly 5) No proper navigation module or path optimization is present 6) Drones cannot predict weather changes	1) Per medical research, that whenever a person is going to get a cardiac arrest, the heart rate fluctuation starts around 30-45 minutes before the attack. 2) Since we are getting the real-time updates of the person’s heart rate, the drones will be sent 30 minutes before the patient is going to get a heart attack, which increases the chances of the person to be saved. 3) No one needs to call for the emergency, the data will automatically show when the person needs help. 4) Emergency contact people will automatically be notified. 5) A proper algorithm for the shortest traversal path and time. 6) Introduction of this technology even to the rural and remote areas. 7) patient’s location is shared with base station to know the weather and hence help them act accordingly

Key Assumptions

1) Every person using the device will be wearing a smart device and a pressure meter that gives a real-time update of the heart rate and blood pressure.
2) The patient must be in an area where there are people around who can be notified about the emergency as the defibrillator is not automatic
3) The emergency contact people or communities nearby will be given a pre demo of the system or know how to use a defibrillator

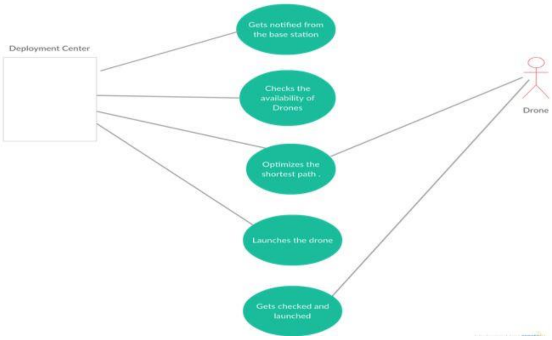
USE Case Diagrams


Logs into the system
Has a heart rate measuring wearable
Real time heart rate
Multiple heart rate checks with the reference table
Notifies the deployment center
Patient
Base Station

1) Patient logs into the system

2) He has heart-measuring device

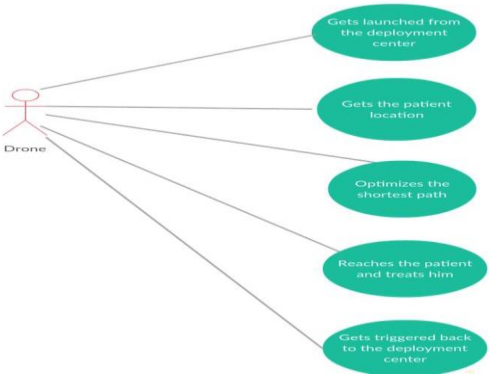
3) Real-time heart rate is calculated and compared with reference data in case of erratic fluctuations. The deployment center is notified


Deployment Center
Gets notified from the base station
Checks the availability of Drones
Optimizes the shortest path
Launches the drone
Gets checked and launched
Drone

1) The deployment center gets notified from the base station

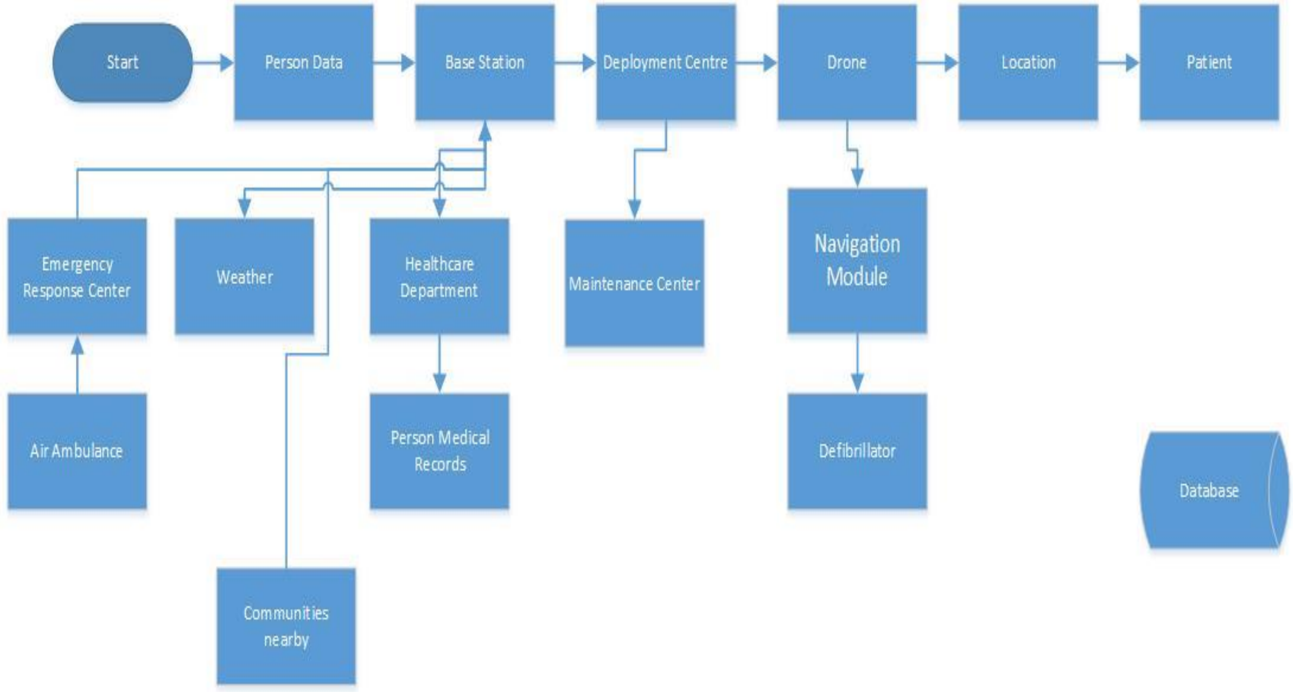
2) Drone availability is checked and launched

3) Shortest path is optimized using Dijkstra algorithm


Gets launched from the deployment center
Gets the patient location
Optimizes the shortest path
Reaches the patient and treats him
Gets triggered back to the deployment center
Drone

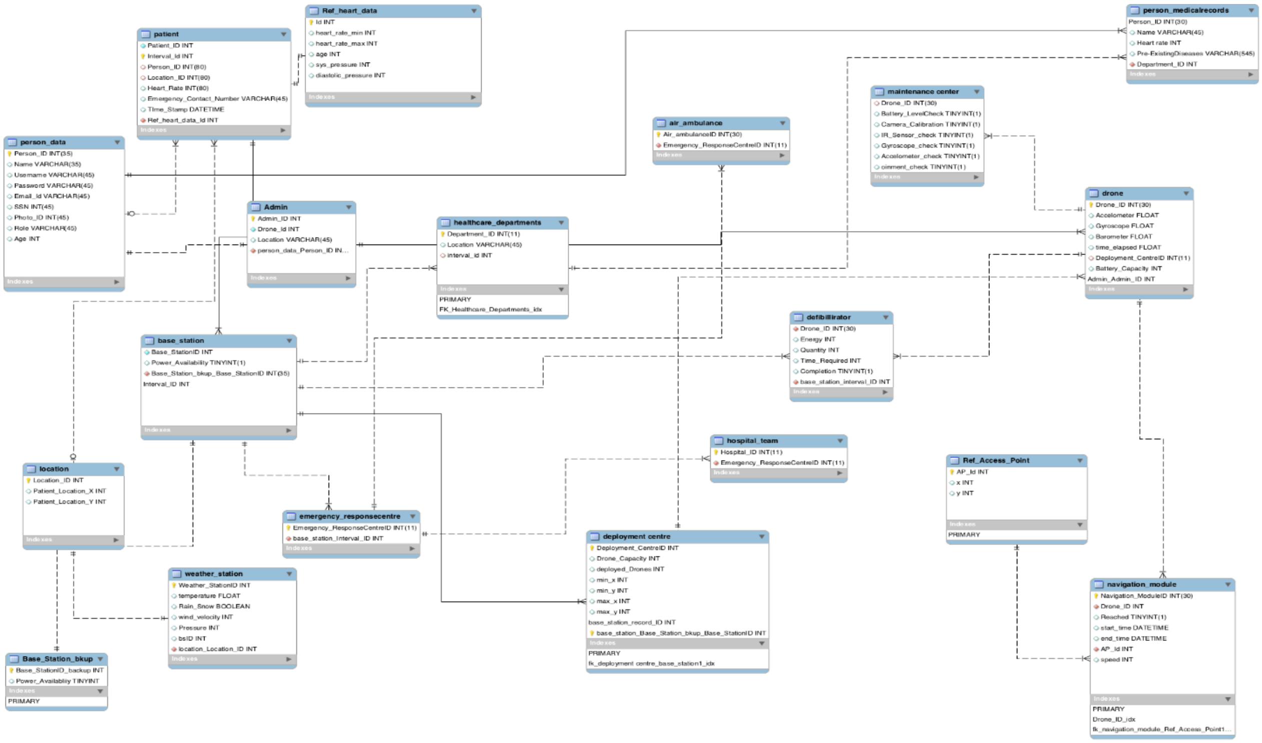
1) The drone reaches the patient treats and triggers back to the deployment center

LOGICAL FLOW DIAGRAM



The above diagram represents the logical flow of database system

ENTITY RELATIONSHIP DIAGRAM



Person Data:

1) We created a login system for Sign-in and Sign-up of users using the device so as multiple people can use the same device avoiding extra costs.

2) The attributes for sign up includes **Name, Password, Email-id, Username, SSN, Photo** and for signing in we will take **username and password.**

Patient:

1) The Patient table will be linked to the Person Table then live heart parameters such as Systolic pressure, diastolic pressure, heart Rate.

2) The person's location or GPS coordinates is also sent to the base station.

Base Station:

1) It is the focal part of the entire system. Here as soon as the base station gets a signal from the logged in individual, apart from receiving data from the person the base station retrieves medical records from healthcare department about individual’s history of illness, allergy, diseases prone to the area he lives in etc. The base station will trigger the drone to move autonomously to the location where the patient is located by providing the parameters such as time, distance and shortest/optimized path to the drone.

2) The base station will also have details of weather reports helping in optimizing the travel conditions of drone. The base station will have a backup in case of emergency shutdown.

Drone:

The drone is one of the integral part of the system which travels to the patient location to perform the first aid with the help of data from **weather station, patient medical records and location via the base station table** and equipped with a defibrillator.

Deployment Centre:

1) The Deployment Centre table has attributes such as **number of drones, battery levels of drones, drone id.**

Maintenance Centre:

1) This centre is the final safety check for drones to get launched. Here quality check and testing of drones is done if the drone does pass the test, it will be sent back to maintenance centre and

2) The attributes present in this table are drone id, battery level, accelerometer check, cog check, barometer check, led check, camera check.

Healthcare department:

It will contain attributes such as address of **healthcare department, department id** which links to both base station and person medical data.

Person Medical data:

From the SSN or Person ID, we will retrieve all medical records of the person, pre-existing diseases to the base station through healthcare department.

Weather Station:

- Weather Id as there are a many station to link to base station.
- Weather station has address as an attribute as well.
- Weather station provides parameters wind velocity, temperature, pressure, rainfall probability, snow prediction etc.

Emergency Response Centre:

After the drone is sent, an emergency response team is sent to get the patient to the hospital. Apart from the data given to drones, the Emergency Response team will also have distance to go the patient's location. This is done because heart can only be temporarily revived using defibrillator.

Navigation Module:

The drone will leverage the navigation module to transit to the person location. **The x, y and z co-ordinates, air pressure, velocity of drone,** are the parameters which help in drone navigation.

Defibrillator:

The defibrillator is dispatched from the drone to perform defibrillation on the victim. **The defibrillator works based on the vital sign records,** intelligent to determine the energy to be given to patient.

Reference Table:

It will contain the parameters like **age, heart rate, blood pressure** with which the base station will check when fluctuations will be caused in the real-time heart rates or blood pressure as it is linked to the patient table and health factor is calculated as well.

Admin:

Admin is a user operating at base station who can see all happenings of the system.

ID: The nodes which must be followed to reach the destination

NORMALIZATION

Database normalization, or simply **normalization**, is the process of organizing the columns (attributes) and tables (relations) of a relational **database** to reduce data redundancy and improve data integrity.

3NF

Third Normal form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So this *transitive functional dependency* should be removed from the table and also the table must be in **Second Normal form**. For example, consider a table with following fields.

Our EER model is in the 3NF form as it does not have any transitive functional dependency and is in the 2nd Normal Form.

Student_Detail Table :

Student_id	**Student_name**	**DOB**	**Street**	**city**	**State**	**Zip**

In this table Student_id is Primary key, but street, city and state depends upon Zip. The dependency between zip and other fields is called **transitive dependency**. Hence to apply **3NF**, we need to move the street, city and state to new table, with **Zip** as primary key.

New Student_Detail Table :

Student_id	**Student_name**	**DOB**	**Zip**

Address Table :

Zip	**Street**	**City**	**state**

JOINS

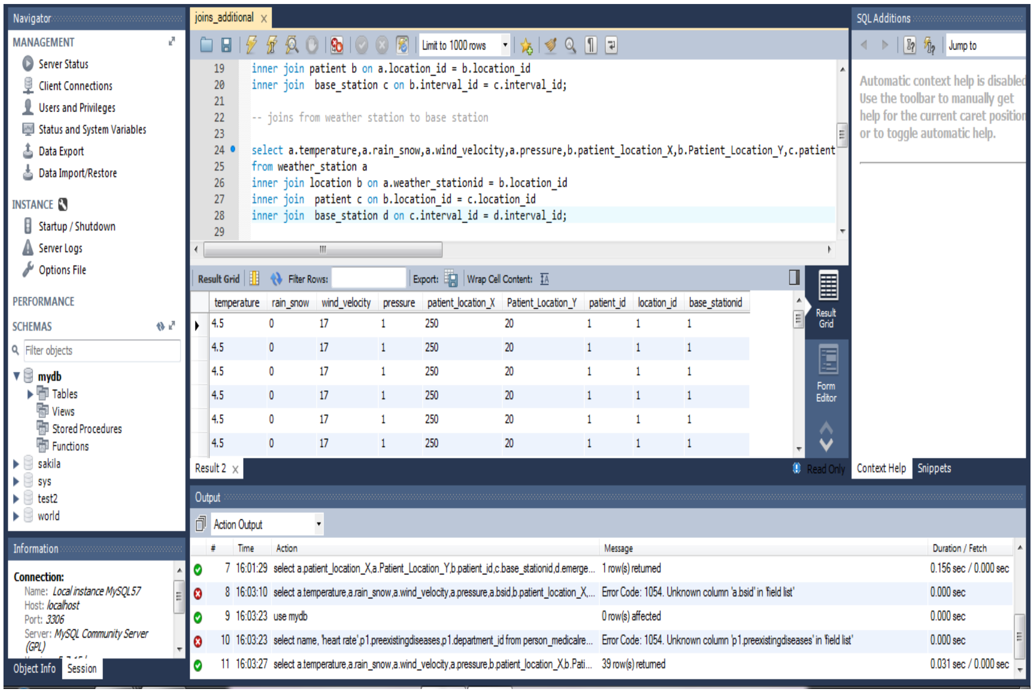
SQL Join is used to fetch data from two or more tables, which is joined to appear as single set of data. SQL Join is used for combining column from two or more tables by using values common to both tables. **Join** Keyword is used in SQL queries for joining two or more tables. Minimum required condition for joining table, is **(n-1)** where **n**, is number of tables. A table can also join to itself known as, **Self Join**.

The following are the types of JOIN that we can use in SQL.

- Inner
- Outer
- Left
- Right

Inner Join

- We are using inner join in our project to join weather station to base station.
- To find the patient location and weather at that location we are joining the 2 tables and forming a SQL query to find the relations.
- We are using Inner joins instead of Sub query at some places to maximize performance when there are lot of sub queries.



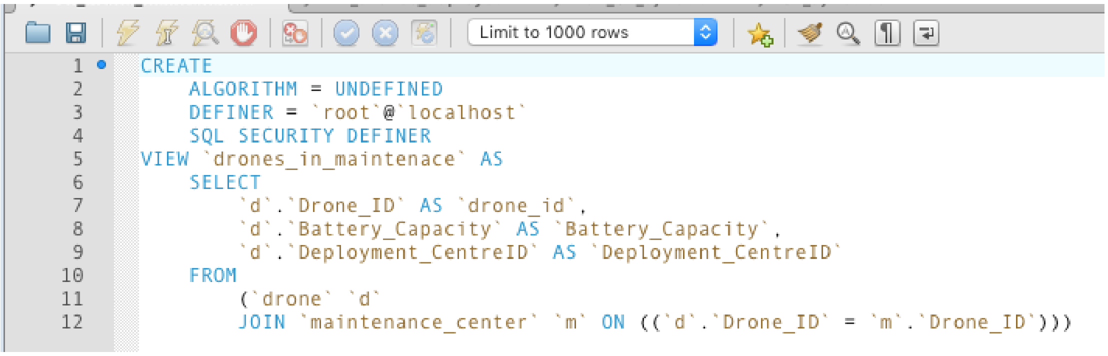
```sql
inner join patient b on a.location_id = b.location_id
inner join  base_station c on b.interval_id = c.interval_id;

-- joins from weather station to base station

select a.temperature,a.rain_snow,a.wind_velocity,a.pressure,b.patient_location_X,b.Patient_Location_Y,c.patient
from weather_station a
inner join location b on a.weather_stationid = b.location_id
inner join  patient c on b.location_id = c.location_id
inner join  base_station d on c.interval_id = d.interval_id;
```

temperature	rain_snow	wind_velocity	pressure	patient_location_X	Patient_Location_Y	patient_id	location_id	base_stationid
4.5	0	17	1	250	20	1	1	1
4.5	0	17	1	250	20	1	1	1
4.5	0	17	1	250	20	1	1	1
4.5	0	17	1	250	20	1	1	1
4.5	0	17	1	250	20	1	1	1
4.5	0	17	1	250	20	1	1	1

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Connection:
Name: *Local instance MySQL57*
Host: *localhost*
Port: *3306*
Server: *MySQL Community Server (GPL)*

```
 #  Time      Action                                                                        Message                                                              Duration / Fetch
 7  16:01:29  select a.patient_location_X,a.Patient_Location_Y,b.patient_id,c.base_stationid,d.emerge...  1 row(s) returned                                          0.156 sec / 0.000 sec
 8  16:03:10  select a.temperature,a.rain_snow,a.wind_velocity,a.pressure,a.bsid,b.patient_location_X,...  Error Code: 1054. Unknown column 'a.bsid' in 'field list'   0.000 sec
 9  16:03:23  use mydb                                                                      0 row(s) affected                                                    0.000 sec
10  16:03:23  select name, 'heart rate',p1.preexistingdiseases,p1.department_id from person_medicalre...  Error Code: 1054. Unknown column 'p1.preexistingdiseases' in 'field list'   0.000 sec
11  16:03:27  select a.temperature,a.rain_snow,a.wind_velocity,a.pressure,b.patient_location_X,b.Pati...  39 row(s) returned                                         0.031 sec / 0.000 sec
```



```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `drones_in_maintenace` AS
    SELECT
        `d`.`Drone_ID` AS `drone_id`,
        `d`.`Battery_Capacity` AS `Battery_Capacity`,
        `d`.`Deployment_CentreID` AS `Deployment_CentreID`
    FROM
        (`drone` `d`
        JOIN `maintenance_center` `m` ON ((`d`.`Drone_ID` = `m`.`Drone_ID`)))
```

The above sub-query tells number of drones in maintenance center.

ALGORITHMS USED

Launching of drones

(i) Base station is linked to the patient whose data is lively received by the base station

(ii) Here heart rate records more than 10 times are recorded and compared with reference heart table to anticipate heart failures.

(iii) The drone is then triggered off from the deployment center to location of patient

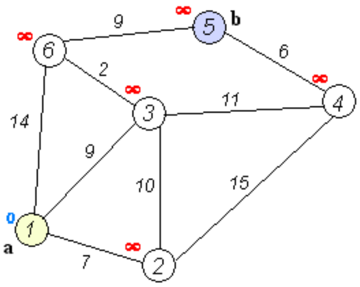
Drone traversal

(i) The drone will be using the Dijkstra's Algorithm, for the shortest path navigation.

(ii) Every deployment center will have 5 nodes associated with it.

(iii) These nodes will help the drone to reach the patients location.

Dijkstra's Algorithm



Let the node at which we are starting be called the **initial node**. Let the **distance of node *Y*** be the distance from the **initial node** to *Y*. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

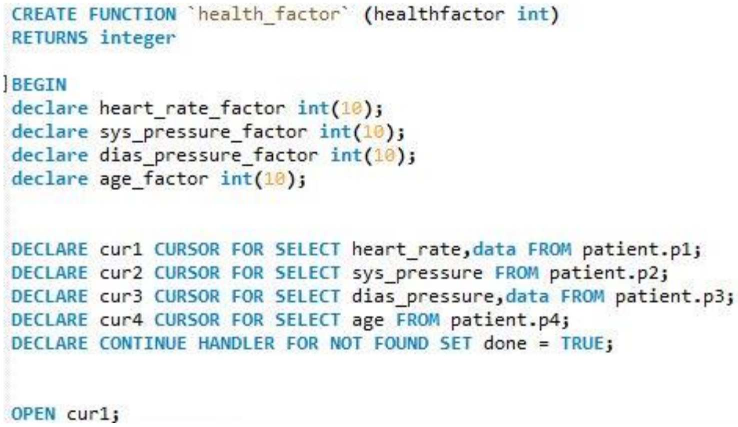
1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Set the initial node as current. Mark all other nodes unvisited. Create a set of all the unvisited nodes called the *unvisited set*.
3. For the current node, consider all its unvisited neighbours and calculate their *tentative* distances. Compare the newly calculated *tentative* distance to the current assigned value and assign the smaller one. For example, if the current node *A* is marked with 6, and the edge connecting it with a neighbour *B* has length 2, then the distance to *B* (through *A*) will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.
4. When we are done considering all the neighbours of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the *unvisited set* is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step 3.

Special cases addressed by Dijkstra

1) If there exist a situation where a patient requires a drone at a location near to the place where one drone is already treating another patient, a second drone is launched from the deployment center, but if the first drone is already near to the person, completes treating the first patient, the drone will return to the nearest node to know whether the second drone has passed through that area or not.
2) If the second drone has passed that node, this drone will be diverted to another patient if not, the first drone will go to the second patient, and the second drone’s speed will be reduced by 5mph and as soon as it reaches that node, the node will divert it to some another patient.

FUNCTIONS

A function is a stored program that you can pass parameters into and then return a value. In our project, we used functions to calculate **health rate factor** and display it on Dashboard for admin to monitor his health. We used functions to calculate average heart rate and blood pressure.

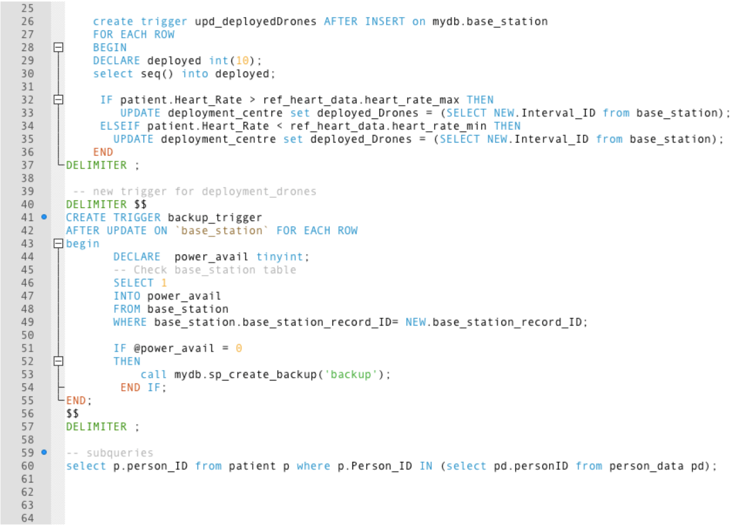


```
CREATE FUNCTION `health_factor` (healthfactor int)
RETURNS integer

BEGIN
declare heart_rate_factor int(10);
declare sys_pressure_factor int(10);
declare dias_pressure_factor int(10);
declare age_factor int(10);


DECLARE cur1 CURSOR FOR SELECT heart_rate,data FROM patient.p1;
DECLARE cur2 CURSOR FOR SELECT sys_pressure FROM patient.p2;
DECLARE cur3 CURSOR FOR SELECT dias_pressure,data FROM patient.p3;
DECLARE cur4 CURSOR FOR SELECT age FROM patient.p4;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;


OPEN cur1;
```

Triggers

A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, or DELETE statement) is performed on a specified table. Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail.

1) **Triggers are used for deploying drones**
2) **After Update Backup trigger for the drone to go back to the deployment center**



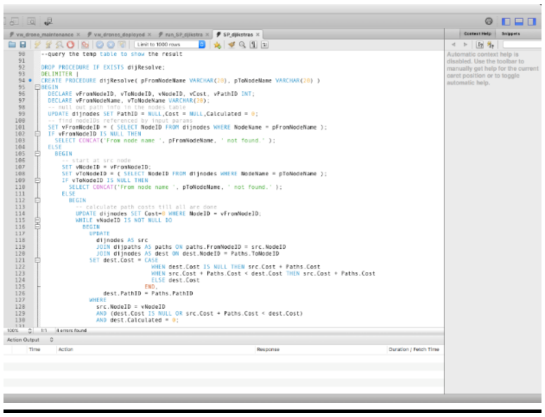
```
    create trigger upd_deployedDrones AFTER INSERT on mydb.base_station
    FOR EACH ROW
    BEGIN
    DECLARE deployed int(10);
    select seq() into deployed;

     IF patient.Heart_Rate > ref_heart_data.heart_rate_max THEN
        UPDATE deployment_centre set deployed_Drones = (SELECT NEW.Interval_ID from base_station);
     ELSEIF patient.Heart_Rate < ref_heart_data.heart_rate_min THEN
       UPDATE deployment_centre set deployed_Drones = (SELECT NEW.Interval_ID from base_station);
    END
DELIMITER ;

 -- new trigger for deployment_drones
DELIMITER $$
CREATE TRIGGER backup_trigger
AFTER UPDATE ON `base_station` FOR EACH ROW
begin
       DECLARE  power_avail tinyint;
       -- Check base_station table
       SELECT 1
       INTO power_avail
       FROM base_station
       WHERE base_station.base_station_record_ID= NEW.base_station_record_ID;

       IF @power_avail = 0
       THEN
           call mydb.sp_create_backup('backup');
        END IF;
END;
$$
DELIMITER ;

-- subqueries
select p.person_ID from patient p where p.Person_ID IN (select pd.personID from person_data pd);
```

Triggers are written to update the base station about weather conditions of the patients location in order for further action like sending off an air ambulance or a hospital team

STORED PROCEDURES

A procedure (often called a stored procedure) is a subroutine like a subprogram in a regular computing language, stored in database. A procedure has a name, a parameter list, and SQL statements.

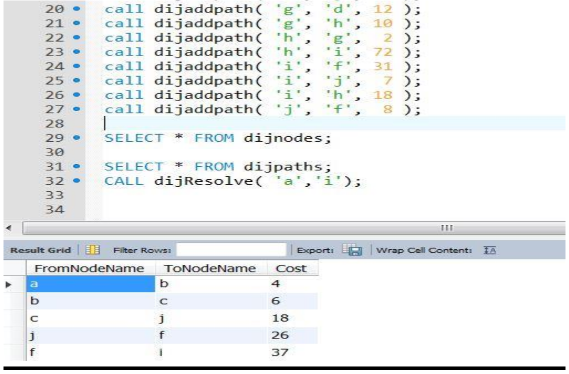
1) Stored Procedures to populate valid nodes and paths.
2) Stored Procedures for Reference medical table and Patient table.
3) Stored Procedures for Full Database backup in form of .csv.
4) Stored Procedures for Incremental back up of Each Table.



```
--query the temp table to show the result

DROP PROCEDURE IF EXISTS dijResolve;
DELIMITER |
CREATE PROCEDURE dijResolve( pFromNodeName VARCHAR(20), pToNodeName VARCHAR(20) )
BEGIN
  DECLARE vFromNodeID, vToNodeID, vNodeID, vCost, vPathID INT;
  DECLARE vFromNodeName, vToNodeName VARCHAR(20);
  -- null out path info in the nodes table
  UPDATE dijnodes SET PathID = NULL,Cost = NULL,Calculated = 0;
  -- find nodeIDs referenced by input params
  SET vFromNodeID = ( SELECT NodeID FROM dijnodes WHERE NodeName = pFromNodeName );
  IF vFromNodeID IS NULL THEN
    SELECT CONCAT('From node name ', pFromNodeName, ' not found.' );
  ELSE
    BEGIN
      -- start at src node
      SET vNodeID = vFromNodeID;
      SET vToNodeID = ( SELECT NodeID FROM dijnodes WHERE NodeName = pToNodeName );
      IF vToNodeID IS NULL THEN
        SELECT CONCAT('From node name ', pToNodeName, ' not found.' );
      ELSE
        BEGIN
          -- calculate path costs till all are done
          UPDATE dijnodes SET Cost=0 WHERE NodeID = vFromNodeID;
          WHILE vNodeID IS NOT NULL DO
            BEGIN
              UPDATE
                dijnodes AS src
                JOIN dijpaths AS paths ON paths.FromNodeID = src.NodeID
                JOIN dijnodes AS dest ON dest.NodeID = Paths.ToNodeID
              SET dest.Cost = CASE
                                WHEN dest.Cost IS NULL THEN src.Cost + Paths.Cost
                                WHEN src.Cost + Paths.Cost < dest.Cost THEN src.Cost + Paths.Cost
                                ELSE dest.Cost
                              END,
                dest.PathID = Paths.PathID
              WHERE
                src.NodeID = vNodeID
                AND (dest.Cost IS NULL OR src.Cost + Paths.Cost < dest.Cost)
                AND dest.Calculated = 0;
```

4 errors found

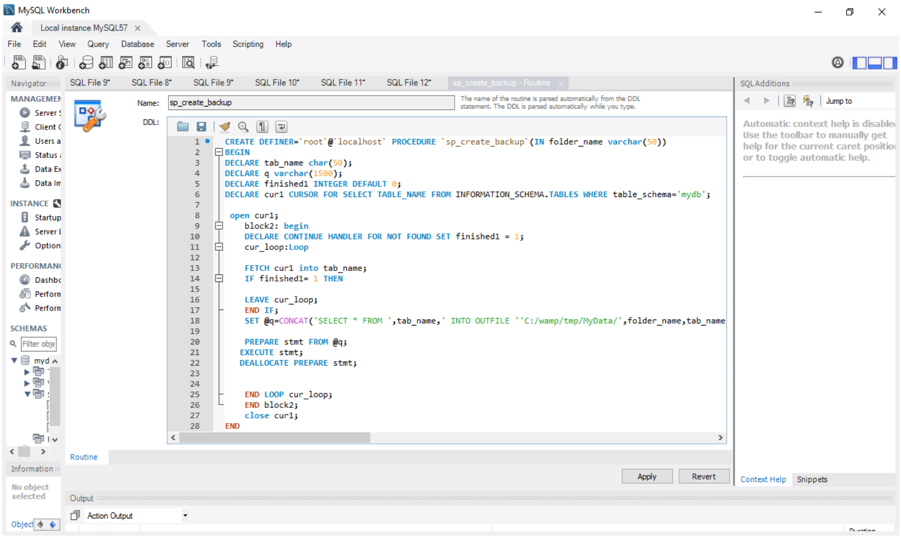
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.



```
call dijaddpath( 'g', 'd', 12 );
call dijaddpath( 'g', 'h', 10 );
call dijaddpath( 'h', 'g',  2 );
call dijaddpath( 'h', 'i', 72 );
call dijaddpath( 'i', 'f', 31 );
call dijaddpath( 'i', 'j',  7 );
call dijaddpath( 'i', 'h', 18 );
call dijaddpath( 'j', 'f',  8 );

SELECT * FROM dijnodes;

SELECT * FROM dijpaths;
CALL dijResolve( 'a','i');
```

FromNodeName	ToNodeName	Cost
a	b	4
b	c	6
c	j	18
j	f	26
f	i	37

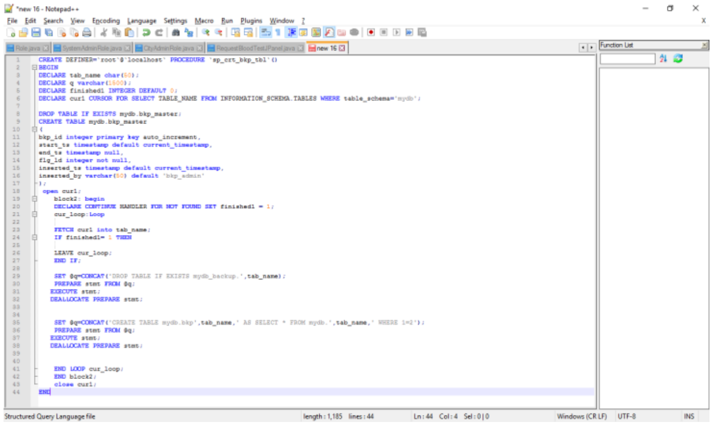
The above figure shows execution of Dijkstra algorithm from navigation module to nodes ID indicating the path to be followed. It is based on weights ie. Distance, speed and time.



Name: sp_create_backup

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_create_backup`(IN folder_name varchar(50))
BEGIN
DECLARE tab_name char(50);
DECLARE q varchar(1500);
DECLARE finished1 INTEGER DEFAULT 0;
DECLARE cur1 CURSOR FOR SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE table_schema='mydb';

 open cur1;
    block2: begin
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished1 = 1;
    cur_loop:Loop

    FETCH cur1 into tab_name;
    IF finished1= 1 THEN

    LEAVE cur_loop;
    END IF;
    SET @q=CONCAT('SELECT * FROM ',tab_name,' INTO OUTFILE ''C:/wamp/tmp/MyData/',folder_name,tab_name

    PREPARE stmt FROM @q;
   EXECUTE stmt;
   DEALLOCATE PREPARE stmt;


    END LOOP cur_loop;
    END block2;
    close cur1;
END
```

The above diagram represents a stored procedure for full backup of database



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_crt_bkp_tbl`()
BEGIN
DECLARE tab_name char(50);
DECLARE q varchar(1500);
DECLARE finished1 INTEGER DEFAULT 0;
DECLARE cur1 CURSOR FOR SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE table_schema='mydb';

DROP TABLE IF EXISTS mydb.bkp_master;
CREATE TABLE mydb.bkp_master
(
bkp_id integer primary key auto_increment,
start_ts timestamp default current_timestamp,
end_ts timestamp null,
flg_id integer not null,
inserted_ts timestamp default current_timestamp,
inserted_by varchar(50) default 'bkp_admin'
);
 open cur1;
    block2: begin
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished1 = 1;
    cur_loop:Loop

    FETCH cur1 into tab_name;
    IF finished1= 1 THEN

    LEAVE cur_loop;
    END IF;

    SET @q=CONCAT('DROP TABLE IF EXISTS mydb_backup.',tab_name);
    PREPARE stmt FROM @q;
   EXECUTE stmt;
   DEALLOCATE PREPARE stmt;


    SET @q=CONCAT('CREATE TABLE mydb.bkp',tab_name,' AS SELECT * FROM mydb.',tab_name,' WHERE 1=2');
    PREPARE stmt FROM @q;
   EXECUTE stmt;
   DEALLOCATE PREPARE stmt;


    END LOOP cur_loop;
    END block2;
    close cur1;
END
```

The above diagram represents stored procedures for incremental backup to be performed every day.

Fact Sheet

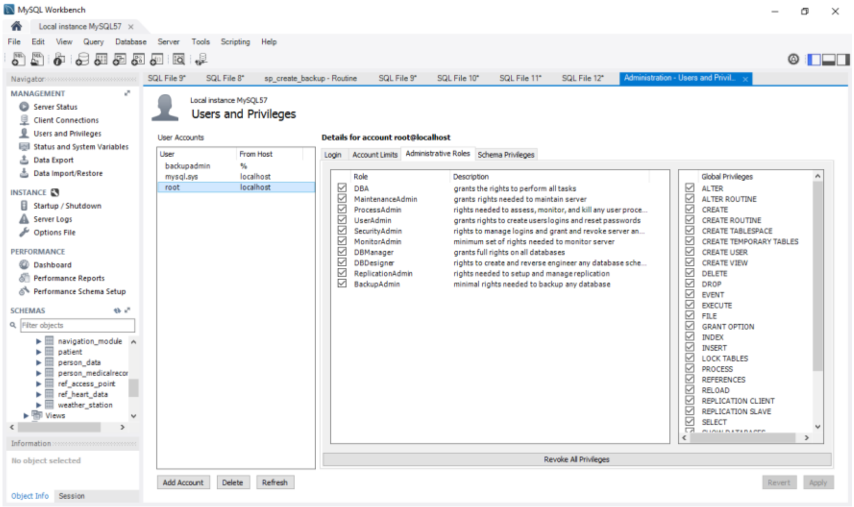
Components	Count
Stored Procedures	4
Triggers	2
Views	8
Joins, subqueries	11
Functions	1

User and Privileges

A user **privilege** is a right to execute a particular type of SQL statement, or a right to access another user's object. The types of privileges are defined by Oracle.

User, on the other hand, are created by users (usually administrators) and are used to group together privileges or other roles. They are a means of facilitating the granting of multiple privileges or roles to users.

In our system, we have given access to the following in our system as represented in the picture below, we had also demonstrated it during our presentation.



Local instance MySQL57

Users and Privileges

User Accounts

User	From Host
backupadmin	%
mysql.sys	localhost
root	localhost

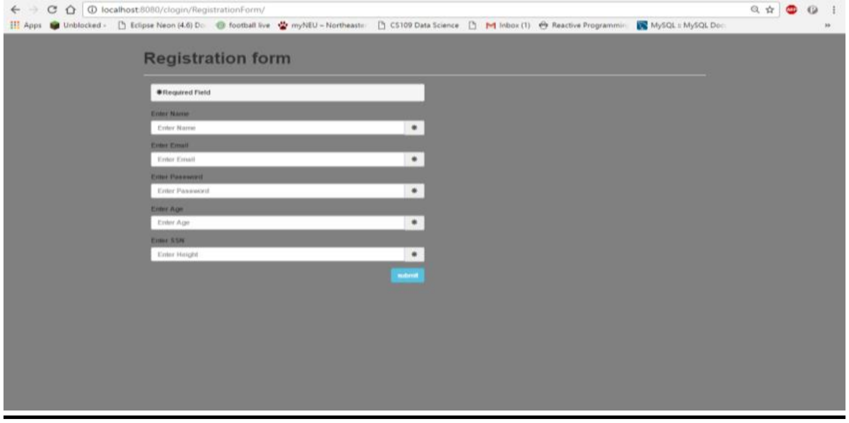
Details for account root@localhost

	Role	Description
☑	DBA	grants the rights to perform all tasks
☑	MaintenanceAdmin	grants rights needed to maintain server
☑	ProcessAdmin	rights needed to assess, monitor, and kill any user proce...
☑	UserAdmin	grants rights to create users logins and reset passwords
☑	SecurityAdmin	rights to manage logins and grant and revoke server an...
☑	MonitorAdmin	minimum set of rights needed to monitor server
☑	DBManager	grants full rights on all databases
☑	DBDesigner	rights to create and reverse engineer any database sche...
☑	ReplicationAdmin	rights needed to setup and manage replication
☑	BackupAdmin	minimal rights needed to backup any database

	Global Privileges
☑	ALTER
☑	ALTER ROUTINE
☑	CREATE
☑	CREATE ROUTINE
☑	CREATE TABLESPACE
☑	CREATE TEMPORARY TABLES
☑	CREATE USER
☑	CREATE VIEW
☑	DELETE
☑	DROP
☑	EVENT
☑	EXECUTE
☑	FILE
☑	GRANT OPTION
☑	INDEX
☑	INSERT
☑	LOCK TABLES
☑	PROCESS
☑	REFERENCES
☑	RELOAD
☑	REPLICATION CLIENT
☑	REPLICATION SLAVE
☑	SELECT

User Interface

The UI was designed using MyPHPadmin and BootStrap



Registration form

●Required Field

Enter Name

Enter Name ●

Enter Email

Enter Email ●

Enter Password

Enter Password ●

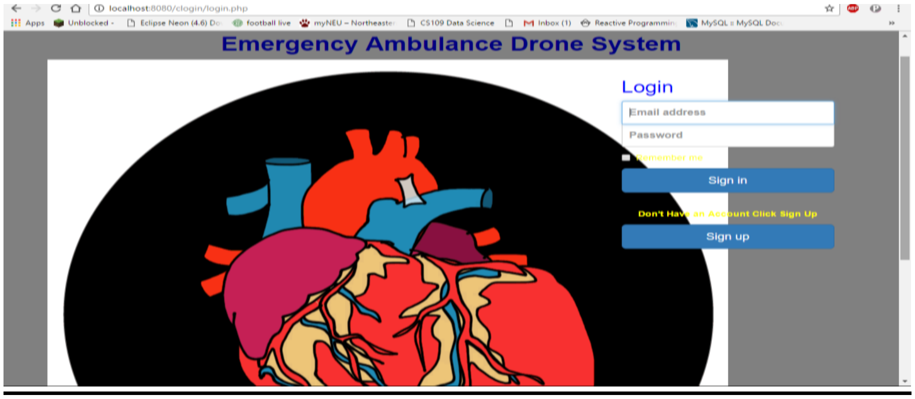
Enter Age

Enter Age ●

Enter SSN

Enter Height ●

The user details are taken in for registration and using the device in the above picture.



Emergency Ambulance Drone System

Login

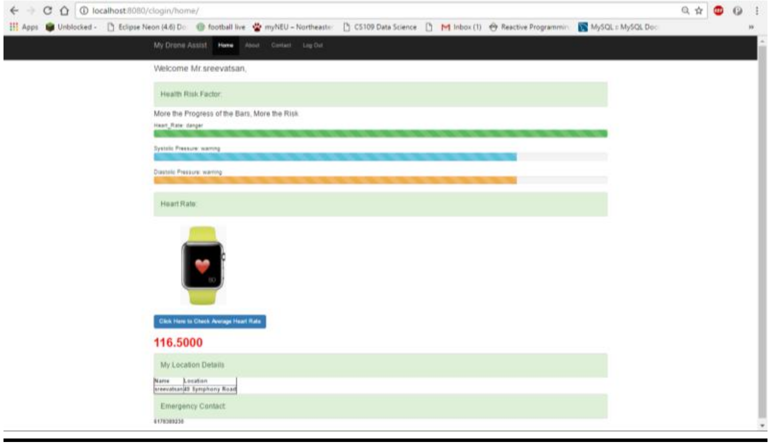
Email address

Password

□ Remember me

Don't Have an Account Click Sign Up

The login details in the above picture.



Welcome Mr.sreevatsan,

Health Risk Factor:

More the Progress of the Bars, More the Risk

Heart_Rate: danger

Systolic Pressure: warning

Diastolic Pressure: warning

Heart Rate:

116.5000

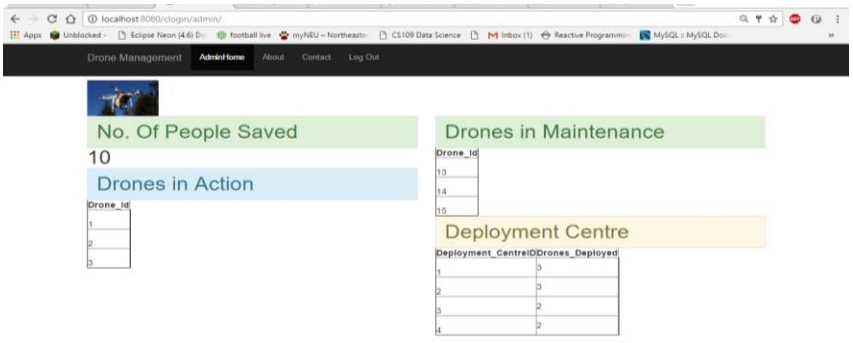
My Location Details

Name	Location
sreevatsan	40 Symphony Road

Emergency Contact:

6178389230

The display page where heart rate, health factor, Pressure, Contact and Location is displayed



No. Of People Saved

10

Drones in Action

Drone_Id
1
2
3

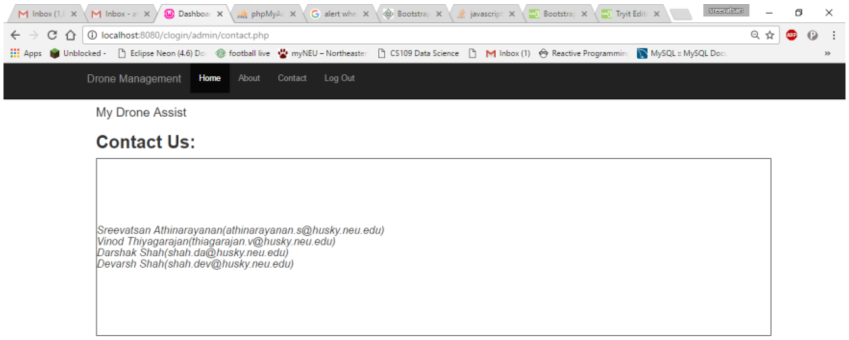
Drones in Maintenance

Drone_Id
13
14
15

Deployment Centre

Deployment_CentreID	Drones_Deployed
1	3
2	3
3	2
4	2

The above picture represents Admin page of system.

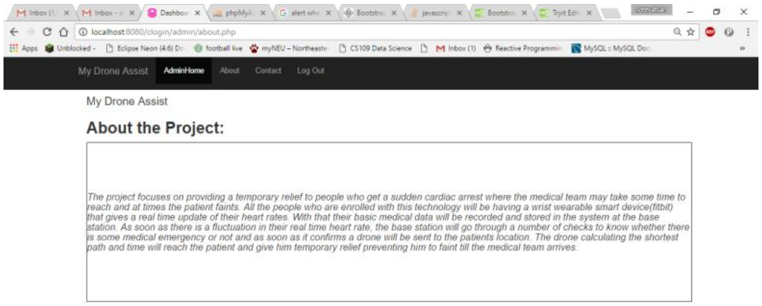


My Drone Assist

Contact Us:

Sreevatsan Athinarayanan(athinarayanan.s@husky.neu.edu)
Vinod Thiyagarajan(thiagarajan.v@husky.neu.edu)
Darshak Shah(shah.da@husky.neu.edu)
Devarsh Shah(shah.dev@husky.neu.edu)

The contact page of the administration is displayed above



My Drone Assist

About the Project:

The project focuses on providing a temporary relief to people who get a sudden cardiac arrest where the medical team may take some time to reach and at times the patient faints. All the people who are enrolled with this technology will be having a wrist wearable smart device(fitbit) that gives a real time update of their heart rates. With that their basic medical data will be recorded and stored in the system at the base station. As soon as there is a fluctuation in their real time heart rate, the base station will go through a number of checks to know whether there is some medical emergency or not and as soon as it confirms a drone will be sent to the patients location. The drone calculating the shortest path and time will reach the patient and give him temporary relief preventing him to faint till the medical team arrives.

About the project page is displayed above it gives salient features of the project

FUTURE SCOPE

1) With the technological innovations and artificial intelligence in the future there can be automatic robotic drones that will not require human assistance for treating a patient.

2) The field can be expanded to treat other medical issues such as diabetes, respiratory problems, traumas etc.

3) With more powerful drones in the future many people can be saved from natural calamities like flood, fire and earthquakes.

4) Issues of network coverage should be addressed for drones.

Conclusion

1) Successful AED usage by lay-persons is currently at 20%. With personalized instructions and communication on the Ambulance Drone, this can be increased to 90%.

2) In short, the Ambulance Drone helps to save lives by extending existing emergency infrastructure with a network of fast and compact UAVs capable of bringing emergency supplies and establishing communication, anywhere.

3) Temporary reviving the heart rate patients in emergency situations and increasing the patient survival rate from 8% to 80%.

4) Fastest air ambulance drone with shortest traversal time and collision avoidance, reaching the patient before he or she faints.

5) Notifying the emergency contacts and people nearby.

6) It can provide the service in rural and remote areas.

REFERENCES

1) F. Techel and B. Zweifel, *Recreational avalanche accidents in Switzerland: Trends and patterns with an emphasis on burial, rescue methods and avalanche danger*. in Proc. of International Snow Science Workshop Grenoble, Chamonix Mont-Blanc, 2013.
2) F. Kendoul, *"Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,"* Journal of Field Robotics, vol. 29, no. 2, pp. 315–378, 2012.
3) A. Bürkle, F. Segor, and M. Kollmann, *"Towards autonomous micro uav swarms,"* Journal of intelligent & robotic systems, vol. 61, no. 1-4, pp. 339–353, 2011.
4) G. Danoy, M. R. Brust, and P. Bouvry, *"Connectivity stability in autonomous multi-level uav swarms for wide area monitoring,"* in ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (ACM DIVANet), pp. 1–8, ACM, 2015.

5) Enabling Drone Communications with WiMAX Technology (Md. Arafatur Rahman)https://www.google.com/patents/US8948935