**SWE3999 - Technical Answers to Real World Problems (TARP)**

**ESSENTIAL MEDICAL NEEDS**

**Title**

*By*

17MIS1019 B. KIRAN KUMARREDDY

17MIS1039 S. SUKESH KUMAR

M.Tech Software Engineering

(5 Year Integrated)

*Submitted to*

Dr. A. BHUVANESWARI

**School of Computer Science and Engineering**

****

*November 2020*

*­*

**DECLARATION**

I hereby declare that the report titled “**ESSETIAL MEDICAL NEEDS”** submitted by me to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of Dr. A. Bhuvaneswari, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Signature of the Candidate

**(To be (digital)signed by the student)**

**17MIS1019, 17MIS1039**

**CERTIFICATE**

Certified that this project report entitled “**ESSENTIAL MEDICAL NEEDS”** is a bonafide work of **BODE KIRAN KUMAR REDDY (17MIS1019), SOLLETI SUKESH KUMAR (17MIS1039)** and they carried out the Project work under my supervision and guidance for SWE3999 - Technical Answers to Real World Problems (TARP).

Dr. A. Bhuvaneswari

SCOPE, VIT Chennai

**ACKNOWLEDGEMENT**

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. A. Bhuvaneswari, School of Computer Science and Engineering for his/her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Jagadeesh Kannan**, Dean and **Dr. S. Geetha**, Associate Dean, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending the facilities of the School towards our project and for their unstinting support.

We express our thanks to **Dr. Asnath Victy Phamila,** Head of the Department, M.Tech. Software Engineering (5 year Integrated) for her support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**(To be (digital)signed by the student)**

**BODE KIRAN KUMAR REDDY**

**17MIS1019**

**SOLLETY SUKESH KUMAR**

**17MIS1039**

**ABSTRACT**

Drugs (Medicines) are synthetic substances or mixes used to fix, stop, or forestall malady; ease manifestations; or help in the finding of sicknesses. Advances in meds have empowered specialists to fix numerous illnesses and spare lives.

Now a day’s medicines are playing an important role in human life.

Medicines preventing disease and injury and relieving pain and avoiding premature death and pursuing a peaceful death.

People who are living in the society needs the medicines, in developed areas people having knowledge about buying medicines and consult the specialists, but in developing areas 50% of the people don’t have a knowledge about which medicines they have to use and which type of specialists they have to consult.

Now a day’s medicines scam is also done. So, people are not getting the good quality of medicines from medical shops.

For all the above reasons we are providing a website having different modules to consult doctor, to order quality medicines, and delivery of medicines in less time with less effort and quality of medicines by using scheduling and least cost distance algorithms

iv

**CONTENTS**

Declaration 2

Certificate 3

Acknowledgement 4

Abstract 5

1 Introduction 7

1.1 Objective and goal of the project 8

1.2 Problem Statement 8

1.3 Motivation 8

2 Literature Survey 9

3 Requirements Specification 11

3.1 Hardware Requirements 11

3.2 Software Requirements 11

4 System Design 11

5 Implementation of System 12

6 Results & Discussion 20

7 Conclusion and Future Work 21

8 References 21

1. **Introduction**

Drugs (Medicines) are synthetic substances or mixes used to fix, stop, or forestall malady; ease manifestations; or help in the finding of sicknesses. Advances in meds have empowered specialists to fix numerous illnesses and spare lives.

Basic medications are those that fulfill the need medical care needs of the populace. They are chosen with due respect to general wellbeing importance, proof on adequacy and security, and similar cost-viability. 1975 World Health Assembly (WHA) meeting, report by WHO Director-General with respect to issues confronting nations in region of prescriptions, first notice of basic meds. 1977 first Expert Committee on basic prescriptions issues specialized report and gathering of fundamental drugs. The Alma-Ata statement during the International Conference on Primary Health Care in 1978 reaffirms that wellbeing is a basic liberty and the achievement of the most elevated conceivable degree of wellbeing is a most significant overall social objective. The Alma Ata announcement has sketched out the eight fundamental parts of essential medical services and arrangement of basic drugs is one of them. Alma-Ata Declarations on essential consideration gave hand book on fundamental drugs.1978 Resolution WHA 31.32 passed asking individuals to set up public basic medication records.

**\* Original definition (1977):** Essential drugs of most extreme significance, fundamental, crucial, and vital for the medical care needs of the populace. World Health Organization (WHO) presented the idea of fundamental medications in 1977. Fundamental prescriptions are those that fulfill the need medical services needs of the populace. They are chosen with due respect to general wellbeing importance, proof on adequacy and security, and relative cost-viability.

Basic prescriptions are planned to be accessible inside the setting of working wellbeing Frameworks consistently in satisfactory sums, in the fitting measurement structures, with guaranteed Quality and satisfactory data, and at a value the individual and the network can bear.

The usage of the idea of basic meds is expected to be adaptable and versatile to a wide range of circumstances. It joins the need to routinely refresh meds determinations to reflect new helpful choices and changing remedial needs; the need to guarantee drug quality; and the requirement for proceeded with advancement of better meds, medications for developing sicknesses, and medications to meet changing opposition designs.

Now a day’s medicines are playing an important role in human life. Medicines preventing disease and injury and relieving pain and avoiding premature death and pursuing a peaceful death.

People who are living in the society needs the medicines, in developed areas people having knowledge about buying medicines and consult the specialists, but in developing areas 50% of the people don’t have a knowledge about which medicines they have to use and which type of specialists they have to consult.

Now a day’s medicines scam is also done. So, people are not getting the good quality of medicines from medical shops.

For all the above reasons we are providing a website having different modules to consult doctor , to order quality medicines, and delivery of medicines in less time with less effort and quality of medicines by using scheduling and least cost distance algorithms

* 1. **OBJECTIVE AND GOAL OF THE PROJECT:**

We are providing a web application for doctors, patients and pharmacies also. From this, a patient can discuss about his disease with doctor. Patient can get the prescription from the doctors, if he wants medicines from the same website, he can order those medicines from pharmacies which we provided in the website. The pharmacy members get the list from the patient orders and they can deliver medicines based on the patient’s ordered date and priority. By that pharmacy members can deliver the medicines with in less time to every patient located in different locations. For this we have been used scheduling algorithms (FIFO, PRIORITY) to sort the orders list and also used least cost distance algorithms (KRUSHKALS, DIJKSTRAS) to deliver the orders in less time.

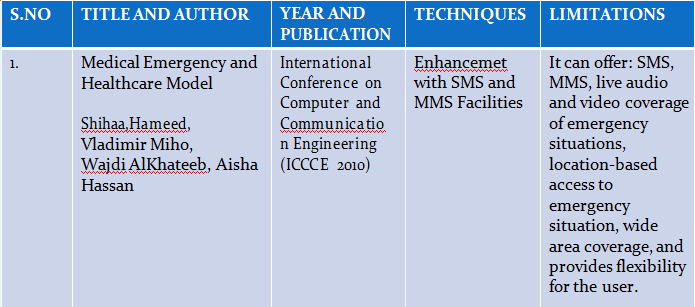
* 1. **PROBLEM STATEMENT**
* **Social issue:**

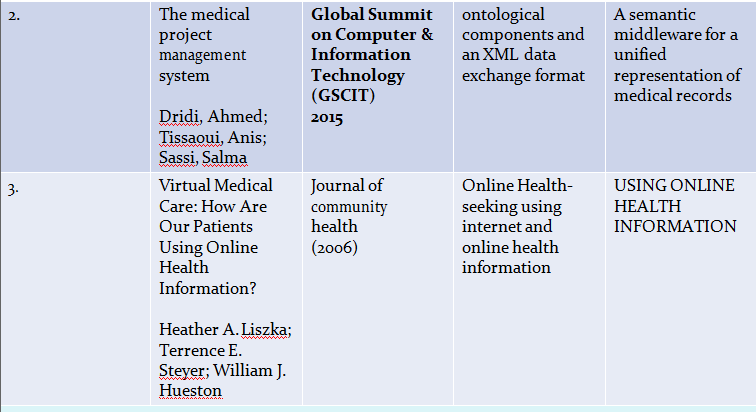
people who are living in the society needs the medicines, in developed areas people having knowledge about buying medicines and consult the specialists, but in developing areas 50% of the people don’t have a knowledge about which medicines they have to use and which type of specialists they have to consult.

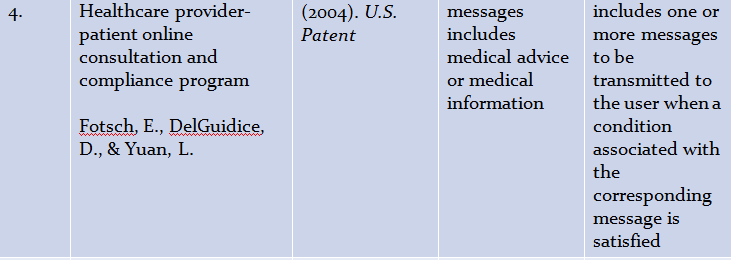
Now a day’s medicines scam is also done. So, people are not getting the good quality of medicines from medical shops.

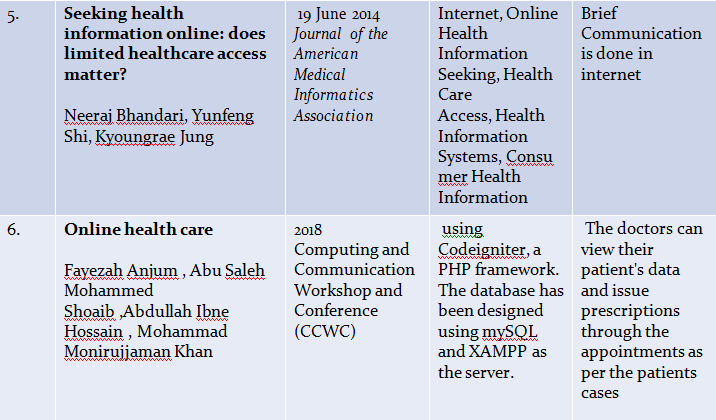
* 🡪Our intension is to lessen this sort of social issue through our web application.

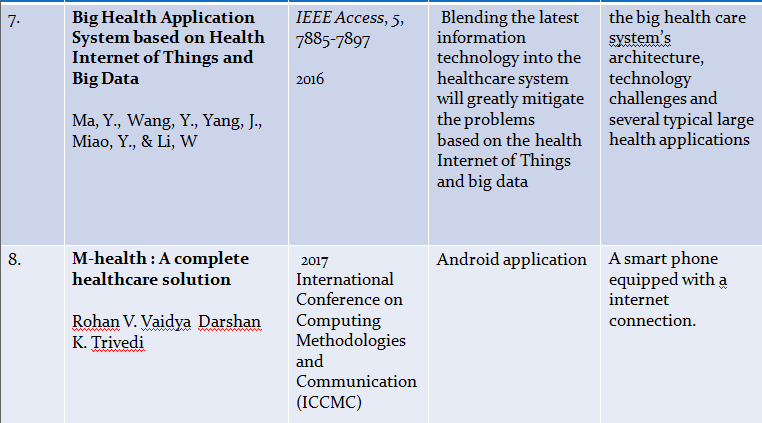
1. **Literature Survey**

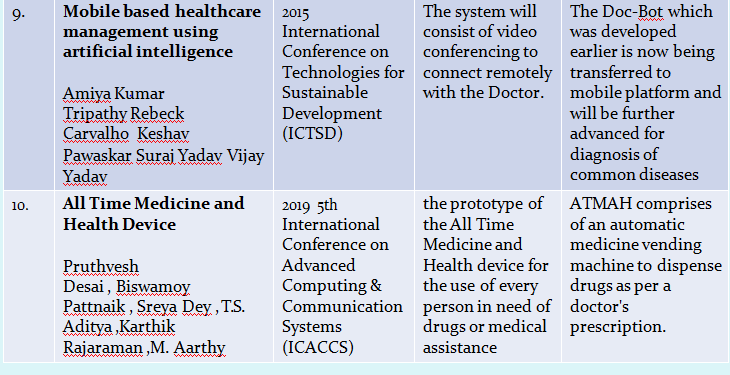


****

****

****

****

****

**3**. **REQUIREMENTS SPECIFICATION:**

3.1 **HARDWARE REQUIREMENTS**:

🡪WINDOWS 10

🡪 4GB RAM / 8GB RAM

3.2 **SOFTWARE REQUIREMENTS:**

🡪COLAB

🡪 XAMPP

🡪JUPYTER (Python notebook)

🡪Languages: HTML, CSS, JAVASCRIPT, PHP, PYTHON

**4 SYSTEM DESIGN:**

**MODULES LIST:**

* Website modules are home page, signup, registration, login, profile, service modules (patient, doctor, pharmacies), clinic, log out, database.
* Backend performing modules are scheduling algorithms (FIFO, PRIORITY), and distance based least cost algorithms (KRUSHKAL’S, DIJKSTRA’S).

**USED ALGORITHMS:**

* SCHEDULING ALGORITHMS:

🡪FIFO (FIRST IN FIRST OUT)

🡪PRIORITY BASED

* LEAST COST DISTANCE ALGORITHMS:

🡪KRUSHKAL

🡪DIJKSTRAS

**5 IMPLEMENTATION OF THE SYSTEM:**

**SCHEDULING ALGORITHMS:**

1. **FIFO:**

First in, first out (FIFO), otherwise called first come, first served (FCFS), is the least complex planning calculation. FIFO essentially lines measures in the request that they show up in the prepared line. This is regularly utilized for an assignment line, for instance as represented in this segment.

🡪Since setting switches just happen upon measure end, and no redesign of the cycle line is required, booking overhead is negligible.

🡪Throughput can be low, in light of the fact that long cycles can be holding the CPU, making the short cycles hang tight for quite a while (known as the caravan impact).

🡪No starvation, in light of the fact that each cycle persuades opportunity to be executed after a distinct time.

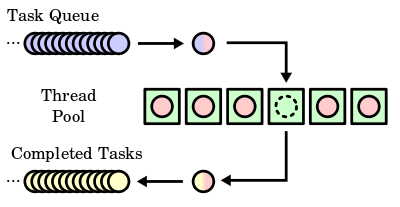
🡪 Turnaround time, holding up time and reaction time rely upon the request for their appearance and can be high for similar reasons above.

🡪 No prioritization happens, along these lines this framework experiences difficulty fulfilling measure time constraints.

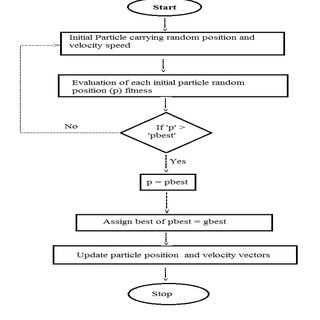
🡪The absence of prioritization implies that as long as each cycle inevitably finishes, there is no starvation. In a climate where a few cycles probably won't finish, there can be starvation.

🡪 It depends on lining.

**ARCHITECTURE DIAGRAM (FIFO):**



**FLOW CHART:**



1. **PRORITY SCHEDULING:**

If there should arise an occurrence of need booking the need isn't constantly set as the backwards of the CPU burst time, rather it very well may be inside or remotely set, however yes the planning is done based on need of the cycle where the cycle which is most pressing is handled first, trailed by the ones with lesser need all together. Cycles with same need are executed in FCFS way.

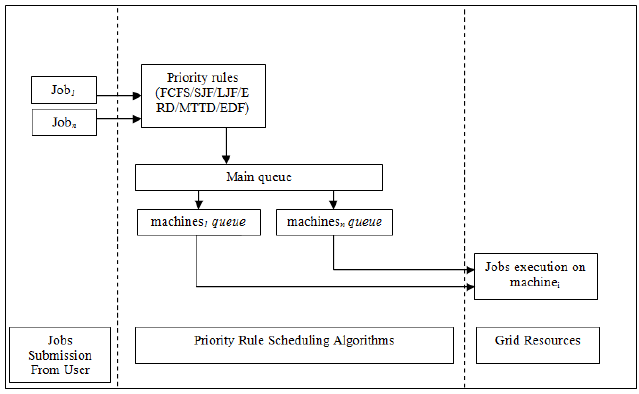
🡪Priority booking is a non-preemptive calculation and one of the most widely recognized planning calculations in bunch frameworks.

🡪Each measure is relegated a need. Cycle with most noteworthy need is to be executed first, etc.

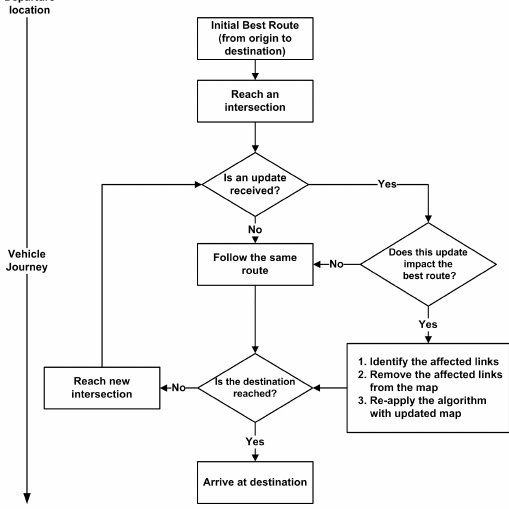
🡪Processes with same need are executed on first start things out served premise.

🡪Priority can be chosen dependent on memory prerequisites, time necessities or some other asset necessity.

**ARCHITECTURE DIAGRAM:**



**FLOW CHART:**



**LEAST COST DISTANCE ALGORITHMS:**

1. **KRUSKAL’S ALGORITHM:**

Kruskal's calculation finds a base spreading over woods of an undirected edge-weighted diagram. On the off chance that the diagram is associated, it finds a base crossing tree. (A base crossing tree of an associated chart is a subset of the edges that frames a tree that incorporates each vertex, where the aggregate of the loads of the apparent multitude of edges in the tree is limited. For a detached diagram, a base spreading over woodland is made out of a base crossing tree for each associated part.) It is an insatiable calculation in chart hypothesis as in each progression it includes the following most reduced weight edge that won't shape a cycle to the base traversing backwoods.

Given an associated and undirected chart, a crossing tree of that diagram is a subgraph that is a tree and interfaces all the vertices together. A solitary chart can have a wide range of traversing trees. A base crossing tree (MST) or least weight traversing tree for a weighted, associated and undirected chart is a spreading over tree with weight not exactly or equivalent to the heaviness of each other spreading over tree. The heaviness of a crossing tree is the aggregate of loads given to each edge of the traversing tree.

What number of edges does a base spreading over tree has?

A base spreading over tree has (V – 1) edges where V is the quantity of vertices in the given diagram.

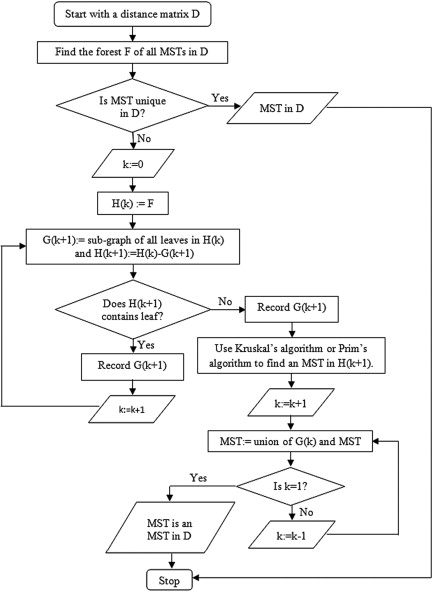
The following are the means for discovering MST utilizing Kruskal's calculation

1. Sort all the edges in non-diminishing request of their weight.

2. Pick the littlest edge. Check on the off chance that it shapes a cycle with the traversing tree framed up until now. On the off chance that cycle isn't framed, incorporate this edge. Else, dispose of it.

3. Rehash step#2 until there are (V-1) edges in the spreading over tree.

**FLOW CHART:**



1. **DIJKSTRA’S ALGORITHM:**

Dijkstra's calculation (or Dijkstra's Shortest Path First calculation, SPF calculation) is a calculation for finding the most limited ways between hubs in a chart, which may speak to, for instance, street organizations. It was brought about by PC researcher Edsger W. Dijkstra in 1956 and distributed three years after the fact.

The calculation exists in numerous variations. Dijkstra's unique calculation found the most brief way between two given hubs, yet a more normal variation fixes a solitary hub as the "source" hub and finds most brief ways from the source to all different hubs in the diagram, creating a most limited way tree.

For a given source hub in the chart, the calculation finds the most brief way between that hub and each other. It can likewise be utilized for finding the most brief ways from a solitary hub to a solitary objective hub by halting the calculation once the most brief way to the objective hub has been resolved. For instance, if the hubs of the diagram speak to urban areas and edge way costs speak to driving separations between sets of urban areas associated by an immediate street (for straightforwardness, overlook red lights, stop signs, expressways and different impediments), Dijkstra's calculation can be utilized to locate the most brief course between one city and every other city. A broadly utilized utilization of briefest way calculation is network directing conventions, most eminently IS-IS (Intermediate System to Intermediate System) and Open Shortest Path First (OSPF). It is additionally utilized as a subroutine in different calculations, for example, Johnson's.

The Dijkstra calculation utilizes marks that are positive numbers or genuine numbers, which are completely requested. It tends to be summed up to utilize any names that are halfway arranged, gave the ensuing marks (a resulting name is delivered while crossing an edge) are monotonically non-diminishing. This speculation is known as the conventional Dijkstra most brief way calculation.

**DESCRIPTION:**

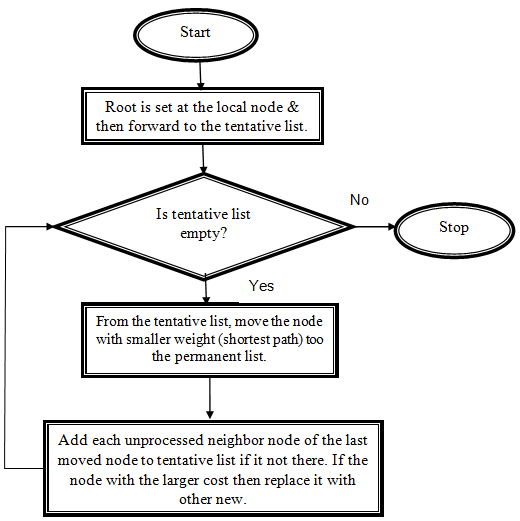
Assume you might want to locate the briefest way between two convergences on a city map: a beginning stage and an objective. Dijkstra's calculation at first denotes the good ways (from the beginning stage) to each other convergence on the guide with endlessness. This is done not to infer that there is an endless separation, however to take note of that those crossing points have not been visited at this point. A few variations of this technique leave the convergences' separations unlabeled. Presently select the current convergence at every emphasis. For the principal emphasis, the current convergence will be the beginning stage, and the separation to it (the crossing point's name) will be zero. For ensuing emphasess (after the primary), the current convergence will be a nearest unvisited crossing point to the beginning stage (this will be anything but difficult to track down).

From the current convergence, update the separation to each unvisited crossing point that is legitimately associated with it. This is finished by deciding the total of the separation between an unvisited crossing point and the estimation of the current convergence and afterward relabeling the unvisited convergence with this worth (the aggregate) on the off chance that it is not exactly the unvisited convergence's present worth. In actuality, the convergence is relabeled if the way to it through the current crossing point is more limited than the recently known ways. To encourage briefest way ID, in pencil, mark the street with a bolt highlighting the relabeled convergence on the off chance that you name/re name it, and delete all others highlighting it. After you have refreshed the separations to each neighboring convergence, mark the current convergence as visited and select an unvisited convergence with insignificant good ways (from the beginning stage) or the most reduced label as the current crossing point. Convergences set apart as visited are marked with the briefest way from the beginning stage to it and won't be returned to or gotten back to.

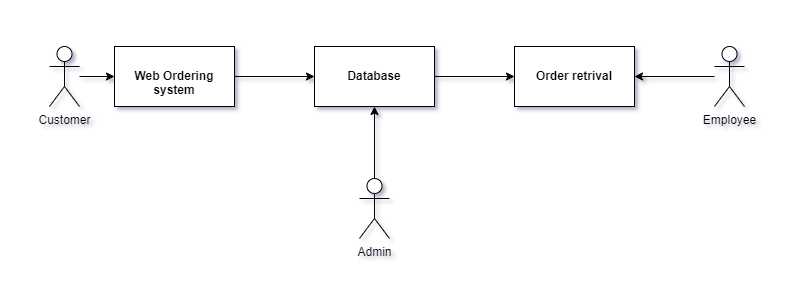
Proceed with this cycle of refreshing the neighboring convergences with the most limited separations, denoting the current crossing point as visited, and moving onto a nearest unvisited convergence until you have denoted the objective as visited. Whenever you have denoted the objective as visited (just like the case with any visited convergence), you have decided the most brief way to it from the beginning stage and can follow your way back after the bolts backward. In the calculation's usage, this is typically done (after the calculation has arrived at the objective hub) by following the hubs' folks from the objective hub up to the beginning hub; that is the reason we additionally monitor every hub's parent.

This calculation makes no endeavor of direct "investigation" towards the objective as one may anticipate. Or maybe, the sole thought in deciding the following "current" convergence is its good ways from the beginning stage. This calculation consequently extends outward from the beginning stage, intelligently considering each hub that is nearer as far as most limited way separation until it arrives at the objective. At the point when perceived thusly, it is clear how the calculation essentially finds the most limited way. In any case, it might likewise uncover one of the calculation's shortcomings: its overall gradualness in certain geographies.

**FLOW CHART:**



**Architecture Diagram:**

****

**6. RESULTS AND DISCUSSION:**

* We developed different modules as all are included in website to get all the inputs and perform the scheduling and least distance (cost) algorithms.
* Modules are home page, signup, registration, login, profile, service modules (patient, doctor, pharmacies), clinic, log out, database.
* Next, we can perform the scheduling algorithms based on the exported csv file of ordered medicines.
* For delivery we have used GEOPY to find out the order location with latitude and longitude.
* And also, we used NETWORKX to perform the distance between the locations, and also implementing the KRUSHKALS and DIJKSTRAS algorithms to find out the least distance between the each and every location mentioned in the orders list
* The result will be the directions that the delivery boy should follow to deliver the products or medicines with less effort and in less time.

**7. CONCLUSION AND FUTURE WORK:**

* From this project we can conclude that using FIFO and priority scheduling algorithms are used to deliver the medicines on first ordered one and priority based. Kruskal’s and Dijkstra’s algorithms are used to deliver the medicines with least cost on distance. The medicines will be delivered accordingly based on the customer choice. The delivery of the products will be faster with less effort and in less time.
* Anyone can solve their medical problems or diseases in this website whether it is minor or major.
* And also, anyone can order the medicines from our website and we can deliver to any locations.
* This project resolves the social issue to deliver the medicines in less time with less effort.

**8**. **REFERENCES**

1) Hameed, S. A., Miho, V., AlKhateeb, W., & Hassan, A. (2010, May). Medical emergency and healthcare model: Enhancemet with SMS and MMS facilities. In *International Conference on Computer and Communication Engineering (ICCCE'10)* (pp. 1-6). IEEE.

2) Dridi, A., Tissaoui, A., & Sassi, S. (2015, June). The medical project management (MPM) system. In *2015 Global Summit on Computer & Information Technology (GSCIT)* (pp. 1-6). IEEE.

3) Liszka, H. A., Steyer, T. E., & Hueston, W. J. (2006). Virtual medical care: how are our patients using online health information?. *Journal of community health*, *31*(5), 368-378.

4) Fotsch, E., DelGuidice, D., & Yuan, L. (2004). *U.S. Patent Application No. 10/641,982*.

5) *Journal of the American Medical Informatics Association*, Volume 21, Issue 6, November 2014, Pages 1113–1117, <https://doi.org/10.1136/amiajnl-2013-002350>

6) F. Anjum, A. S. M. Shoaib, A. I. Hossain and M. M. Khan, "Online health care," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2018, pp. 580-583, doi: 10.1109/CCWC.2018.8301617

7) Ma, Y., Wang, Y., Yang, J., Miao, Y., & Li, W. (2016). Big health application system based on health internet of things and big data. *IEEE Access*, *5*, 7885-7897.

8) R. V. Vaidya and D. K. Trivedi, "M-health : A complete healthcare solution," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 556-561, doi: 10.1109/ICCMC.2017.8282527

9) A. K. Tripathy, R. Carvalho, K. Pawaskar, S. Yadav and V. Yadav, "Mobile based healthcare management using artificial intelligence," 2015 International Conference on Technologies for Sustainable Development (ICTSD), Mumbai, 2015, pp. 1-6, doi: 10.1109/ICTSD.2015.7095895.

10) P. Desai, B. Pattnaik, S. Dey, T. S. Aditya, K. Rajaraman and M. Aarthy, "All Time Medicine and Health Device," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 5-9, doi: 10.1109/ICACCS.2019.8728306.

**APPENDIX**

**SAMPLE CODE:**

import numpy as np

import pandas as pd

df=pd.read\_csv("/content/ordermedic.csv")

df['DATE']=pd.to\_datetime(df['DATE'], infer\_datetime\_format=True)

df.sort\_values(by=['DATE'])

df.sort\_values(by=['PRIORITY'],ascending=False)

df.sort\_values(by=['DATE','PRIORITY'])

!pip install geopandas

!pip install geopy

import pandas as pd

import geopandas as gpd

import geopy

from geopy.geocoders import Nominatim

from geopy.extra.rate\_limiter import RateLimiter

import matplotlib.pyplot as plt

import folium

from folium.plugins import FastMarkerCluster

from geopy.extra.rate\_limiter import RateLimiter

locator = Nominatim(user\_agent="myGeocoder")

geocode = RateLimiter(locator.geocode, min\_delay\_seconds=1)

df['location'] = df['ADDRESS'].apply(geocode)

df['point'] = df['location'].apply(lambda loc: tuple(loc.point) if loc else None)

df.head()

# split point column into latitude, longitude and altitude columns

df[['latitude', 'longitude', 'altitude']] = pd.DataFrame(df['point'].tolist(), index=df.index)

df.head()

df.head()

df.latitude.isnull().sum()

df = df[pd.notnull(df["latitude"])]

map1 = folium.Map(

location=[28.640971,77.044717],

tiles='cartodbpositron',

zoom\_start=12,

)

df.apply(lambda row:folium.CircleMarker(location=[row["latitude"], row["longitude"]]).add\_to(map1), axis=1)

map1

df

df.shape

len(df)

df.to\_csv('orders.csv')

df1=pd.read\_csv('/content/orders.csv')

df1

!pip install Dijkstar

from dijkstar import Graph, find\_path

graph=Graph()

n=df1.shape[0]+1

n

from geopy.distance import geodesic

geodesic((df1['latitude'][0],df1['longitude'][0]),(df1['latitude'][1],df1['longitude'][1])).km

for i in range(n-1):

for j in range(n-1):

graph.add\_edge(i,j,geodesic((df1['latitude'][i],df1['longitude'][i]),(df1['latitude'][j],df1['longitude'][j])).km)

find\_path(graph,0,16)

!pip install networkx

import networkx as nx

g=nx.Graph()

m=[]

for i in range(n-1):

for j in range(n-1):

m.append((i,j,geodesic((df1['latitude'][i],df1['longitude'][i]),(df1['latitude'][j],df1['longitude'][j])).km))

g.add\_weighted\_edges\_from(m)

print(nx.dijkstra\_path(g,1,16))

from collections import defaultdict

class Graph:

def \_\_init\_\_(self, vertices):

self.V = vertices

self.graph = []

def addEdge(self, u, v, w):

self.graph.append([u, v, w])

def find(self, parent, x1):

if parent[x1] == x1:

return x1

return self.find(parent, parent[x1])

def union(self, parent, rank, x, y):

xroot = self.find(parent, x)

yroot = self.find(parent, y)

if rank[xroot] < rank[yroot]:

parent[xroot] = yroot

elif rank[xroot] > rank[yroot]:

parent[yroot] = xroot

else:

parent[yroot] = xroot

rank[xroot] += 1

def KruskalMST(self):

result = []

i = 0

e = 0

self.graph = sorted(self.graph,

key=lambda item: item[2])

parent = []

rank = []

for node in range(self.V):

parent.append(node)

rank.append(0)

while e < self.V - 1:

u, v, w = self.graph[i]

i = i + 1

x = self.find(parent, u)

y = self.find(parent, v)

if x != y:

e = e + 1

result.append([u, v, w])

self.union(parent, rank, x, y)

minimumCost = 0

print ("Edges in the constructed MST")

for u, v, weight in result:

minimumCost += weight

print("%d -- %d == %d" % (u, v, weight))

print("Minimum Distance to deliver" , minimumCost)

g1 = Graph(n-1)

for i in range(n-1):

for j in range(n-1):

g1.addEdge(i,j,geodesic((df1['latitude'][i],df1['longitude'][i]),(df1['latitude'][j],df1['longitude'][j])).km)

g1.KruskalMST()

nx.draw(g)

**SNAPSHOTS (OUTPUT SCREENSHOTS):**

