Indian Institute of Information Technology, Allahabad
Software Engineering

Instructors: Dr. Abhishek Vaish,Dr. Sonali Agarwal

SOFTWARE REQUIREMENT SPECIFICATION

Problem Statement- Design a software for Massive E-Health services using Real time data for IoMT(Internet of medical things) Enabled Healthcare Systems.

GROUP MEMBERS-

Aditya Raj - IIB2019007

Hitika Rajesh Kumar - IIB2019021

Harsh Mahajan - IIB2019001

B Chetan Rao - IIB2019018

1. Introduction:

TMinus0 is a real time application that detects the real time drift(concept drift, time series data variation) in medical data and does overall analysis of medical clinical data to provide software-hardware interlinked based IOT-ML integrated services for patients like ‘critical condition. prediction’,’stay time analysis’ etc..

1.1 Purpose:

With the rise in application of technology in the healthcare domain, both expectation and need of real time healthcare analysis and monitoring software has increased as well as a lot of data has been generated over the last few years.Still, real time analysis and service has been difficult in this domain. One of the primary reasons has been the problem of concept drift in medical data.Our purpose is to build software that can solve problems of concept drift and then provide real time analysis and service.

From patient’s point of view:-

- Delay in healthcare services causes risk to lives.
- Lack of real time analysis.
- High mortality rate of critical patients due to time delay in analysis.

From Developer’s point of view:-

- Concept drift becomes hindrance in time-series analysis or split of data.This causes problems because the predictions might become less accurate as the time passes or opportunities to improve the accuracy might be missed
- Due to these overtime change in data distribution, linear or probabilistic models give bad results on performance metrics when used on these data
- Problems like overfitting and poor accuracy can also occur
- Since, concept drift reduces applicative confidence of data that leads to wastage of critical data

- Occurence of human error is more prone while analysis.

From Doctor's point of view:-

- Analyse patients without getting in direct contact in case of communicable disease.
- Multi-patient analysis is possible.
- Dependency of immediate healthcare service will decrease.
- Treatment can be started in case of shortage of doctors.

So, this project is an attempt to overcome these drawbacks in the healthcare domain.

1.2 Scope:

1) Immediate effective results by Tminus0, which states whether ICU transfer is required or not ,helps in decreasing the fast-track requirement of the doctor.
2) Tminus0 if developed into more user-friendly software can have more major scopes as well.
3) Since it's a software ,it is portable i.e can be used from anywhere in the world.
4) It may decrease physical presence in hospitals too.

Out of Scope:

a. Cannot eliminate doctor's dependence.
b. At this level computation error may become possible

1.3 Definitions, Acronyms, and Abbreviations:

Acronyms and Abbreviations:

a. "TMinus0" is a software that provides real time concept drift detection on medical data and patient services like “critical condn. prediction”,”stay time prediction”.

b. SRS: Software Requirement Specification

Definitions:

a. Concept drift :- concept drift means that the statistical properties of the target variable, which the model aims to predict, change over time unexpectedly.

b. Data Sampling:- Reducing the skewness of a particular class or set of classes in a dataset by means like upsampling,downsampling,oversampling etc. .

1.4 References:

IEEE SRS Format

1.5 Overview:

The SRS starts with an introduction of the problem,our opinion and purpose to solve the problem,then, it gives the point of view of the main involvers in the software. After that abbreviations and definitions are covered.

Below is the overall description of software with product perspective,functions and main actors and general constraints of development.

2. Overall Description:

2.1 Product Perspective:

“TMinus0” is aimed to analyze real time drift detection and analysis in massive medical data as well as provide healthcare services .It can do multi-patient analysis and store factual data for future research. This can allow application of supervised models on data and also solve some critical problems like poor performance metrics result on real time medical data.

2.2 Product Functions:

"TMinus0" supports the following use cases:

Use cases	**Description of use cases**
Developer:	
Authorized login	Allows Developer to login.
Add new visualization tools	Developer can add new tools and analyzers to featurize or simplify data.
Add new Model	Developer can add new models and algorithms for testing, api deployment.
Modify Front-end	Developer can make changes to the front-end to make it more user friendly.
Update preprocessing code	Developer can update or change code of data preprocessing.
Add/remove profiles	Developer can add/remove/update doctor’s profile
Maintain security	Developer can hide/protect or public the data or tools as per security need
Doctor:	
Authorized login	Allows Doctor to login
Visualize the data	Doctors can visualize and analyze patient’s data before further operations

critical condition	Doctor can use the software to predict critical condition and handling of data
request change	Doctor can comment or request change in application of software
Analyze and treat patient	Doctor can do treatment and analysis of patient with the help of software

2.3 User Characteristics:

a. The user should be familiar with the operations of computer softwares and analysis.

2.4 Principal Actors:

The three principal actors in "TMinus0" are “Developer” ,“Doctor” and “Patients”.

2.5 General Constraints:

a. Working on "TMinus0" requires Internet connection.

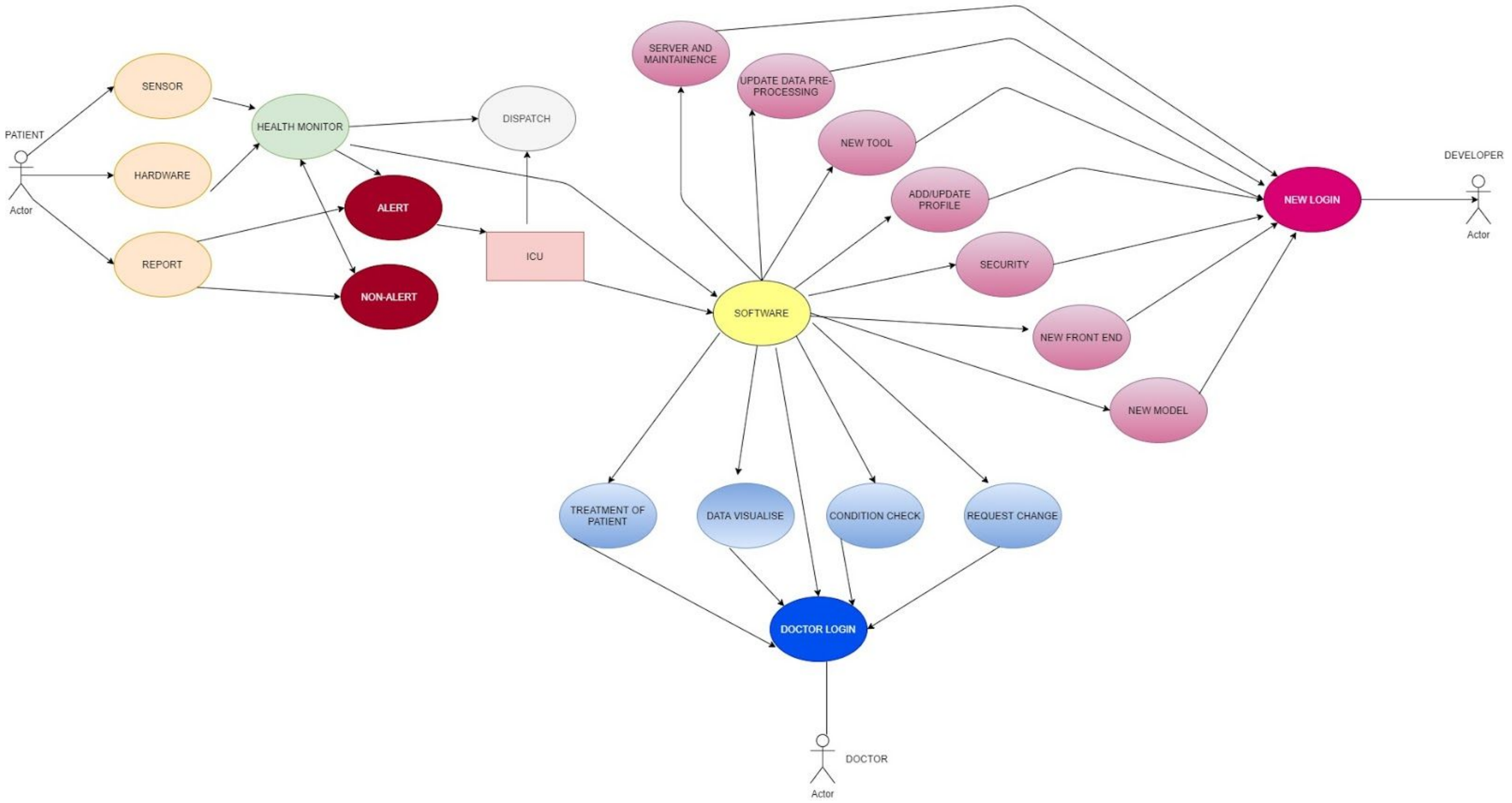
b. "TMinus0" is a single-user Application. Every Doctor must have this application on their computer system.

2.4 Assumptions and Dependencies:

a. Working of "TMinus0" is dependent on the availability of Internet connection.

b. Admin must be active to insert all the required information on the app.

APPENDIX A: USE CASE DIAGRAM



3. Specific Requirements:

3.1 Functional Requirements:

We describe the functional requirements by giving various use cases.

Use Case 1:

Name: Authorized login

Summary: Allows Admin to login.

Actors: Developer/Doctor

Pre-conditions:

- Internet connectivity.
- Only for allowed mail ids.

Main success scenario:

- software checks for the authorization of login.

Extension:

Id or password incorrect. Shows error dialog box.

Post-condition:

Actor can now access all features of the software.

Use Case 2:

Name: Add new visualization tools.

Summary:add new tools and analyzers to featurize or simplify data.

Actors: Developer.

Pre-conditions:

- Internet Connectivity.
- the new tool to be added must be of use.

Main success scenario:

- Added a new visualization tool.

Use Case 3:

Name: Add new Model

Summary: add new models and algorithms for testing, api deployment.

Actors: Developer

Pre-conditions:

- Internet connectivity.
- the new model must be in accordance with the basic model/algorithm framework/skeleton.

Main success scenario:

- successfully added a new model.
- Developer can integrate new tested better performing model with software.

Use Case 4:

Name: Modify Front-end

Summary: make changes to the front-end to make it more user friendly.

Actors: Developer

Pre-conditions:

- Internet connectivity.
- backup of current software.

Main success scenario:

- software becomes more user friendly.
- The front-end bug issue must be solved.

Use Case 5:

Name: Update preprocessing code

Summary: update or change code of data preprocessing.

Actors: Developer.

Pre - Conditions:

- Backup of working code.

Main Success scenario:

- fill in missing values,
- noisy data becomes smooth.
- identify and/or remove outliers.
- resolve inconsistencies

Use Case 6:

Name: Add/remove profiles

Summary: add/remove/update doctor's profile

Actors: Developer

Pre - Conditions:

- information about the Doctor.
- authentication of doctor.

Main Success Scenario:

- successful add/remove/update doctor's profile.

Use Case 7:

Name: Maintain security

Summary: Developer can hide/protect or make public the data or tools as per security needs.

Actors: Developer

Main Success Scenario:

- Security fault is now solved
- Security enhancement.

Use Case 8:

Name: Visualize the Data

Summary: Actors: Doctors can visualize and analyze patient’s data before further operations

Pre-conditions:

- Internet connectivity.

Main success scenario:

- Doctor has analyzed the patient's data and can now provide required treatment.

Use Case 9:

Name: Critical Condition

Summary: Actors: Doctor can use the software to predict critical condition and handling of data.

Pre-conditions:

- Internet connectivity.

Main success scenario:

- Doctor has analyzed the patient's data and checked if the condition of the patient is critical or may soon become critical.

Use Case 10:

Name: Request Change

Summary: Actors: Doctor can comment or request change in application of software

Pre-conditions:

- Internet connectivity.

Main success scenario:

- the request for a change in application of software has been received and the developer can now look into servicing those requests.

Use Case 11:

Name: Analyze and treat patient

Summary: Actors: Doctor can do treatment and analysis of patient with the help of software

Pre-conditions:

- Internet connectivity.
- There must be more than zero patients for doctors to give treatment to any patient.

Main success scenario:

- The software has successfully accomplished its mission of providing patients with proper healthcare treatment at the right time required.
- The doctors' work of physically examining the patients regularly has also been reduced if not eliminated as now the software predicts and alarms doctors of the need of patients.

3.2 Non-Functional Requirements:

All the technical and developmental changes can be done by the developer only.

It is tried to provide end to end software development service from data preparation to user service.

Developer can also change data preparation code or pattern or visualization segment but he also doesn't have access to critical data so security is prompt here.

3.3 Hardware Requirements:

Should run on a computer device with enough memory to pass data and store heavy models and also support visualizing software.

Requires minimum 4gb ram for smooth functionality of the app.

Advance medical computational equipment is must to generate required real time data.

3.4 Software Requirements:

Minimum SDK version: windows 10 or ubuntu 15+ will support the software in a better way.

3.5 Software Design and Development Constraints:

Security: The files in which the information regarding patient and clinical data should be secured against malicious deformations.

Fault Tolerance: Data should not become corrupted in case of system crash or power failure.

API Integration: Developing algorithm/model for drift detection and and real time analysis can be done but it can be only connected to interface with API because the model can be heavy so designing of software must be done with API - integration friendly interface.

Real-Time: Since the request for process input to algorithm or model will be done through request of client via API to server,the software design should be good to make this action fast for real time analysis.

Low-Latency:- One of the major constraint will be deployment of overall better performing low latency software.