IQRA NATIONAL UNIVERSITY

INU

DEFINING POSSIBILITIES, CREATING OPPORTUNITIES

Name: Inzamam
ID#12998
Subject: Software Requirement and Specification
Program: BS(SE)

Question No. 1:

Ans:

Requirements are a set of prioritized needs from all the involved stakeholders that form the base for the functionalities or features to be included as a part of the solution.

In product development, a requirement is a singular documented physical or functional need that a particular design, product or process aims to satisfy. It is commonly used in a formal sense in engineering design, including for example in systems engineering, software engineering, or enterprise engineering.

System: Home based security system using raspberry pi through email and voice alert:

Methodoly:

This system can protect humans from any case of intrusion in their property. We developed IoT based system to alert the registered members through Email whenever human intrusion is detected. In this system our main components are Raspberry Pi 3, Pi camera, buzzer and constant internet connection. We have two operating modes, first for any person comes in front of the camera, and camera detects it and capture its image. Secondly, when images are being captured, they are compared with defined images. If images match, the system will show (authorized) message. If not; the system will send the information to authorized user's Email.

In other words, The system works as if its sensors detect any unnecessary movement, the system will immediately capture the movement and pictures are then captured. The pictures will be sent to the specific email, through which the owner will get updates immediately. The hardware that is required to fulfill this project is: Raspberry Pi, Pi camera and PIR sensor. This system is based on the hardware, human interaction is minimal in it.

Hardware Features:

Raspberry Pi:

Raspberry Pi used in this system is Raspberry Pi model 3B. The specifications of Raspberry Pi 3B are:

- Microprocessor comes with a 64bit Quad Core Processor.
- 40 GPIO pins.
- 1GB of RAM.

- Comes with a built-in Wi-Fi and Bluetooth connectivity.
- Four sockets of 2.0 USB ports.
- HDMI port.
- Micro SD card slot
- Ethernet port.

Pi Camera:

This camera is an official product of Raspberry Pi. It is a V2 camera which supports:

- Its resolution is 8 Mega Pixels.
- It supports 1080p video streaming at 30 FPS, also 720p at 60 FPS and 480p at 90 FPS.
- It weighs about 3g.
- The sensor used in this camera is Sony IMX219.
- Sensor resolution is 3280 × 2464 pixels.

Other features working:

- Camera which is used, is an 8MP V2 Pi camera. It is used to detect person and with the help of OpenCV tool, it will recognize the person in its range.
- Raspberry Pi fan is used to keep the main board cool and it consumes 5V of power.
- Buzzer is used to generate a voice alert whenever an unauthorized person is detected.
- Power supply is to power up the Raspberry Pi.

Constraints:

One of the constraint is basically the response of internet. Response is basically defined as the time elapsed between the initiation of the command and command completion. That response time most of the time depends upon the speed of internet. The more better is the speed of internet, the more will be the quick response of our relays to enable or disable the load. It does not depend upon the distance upon which the response time depends. The other factor upon which is the amount of traffic on the website. The more is the traffic, the less will be the response time of our device and vice versa.

Assumptions:

We assumed that each of the features defined could be implemented programmatically in python and using different tools such as OpenCV, Thonny Raspberry Pi. And implement the hardware as per its need.

Question No. 2

Ans:

Functional Requirements:

. Describe functionality or system services.

. Depend on the type of software, expected users and the type of system

where the software is used.

. Functional user requirements may be high-level statements of what the system should do; functional system requirements should describe the system services in detail.

Examples

- The user shall be able to search either all of the initial set of databases or select a subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store.
- Every order shall be allocated a unique identifier (ORDER ID) which the user shall be able to copy to the account's permanent storage area.

To understand functional requirements properly, let us consider the following example of an online banking system.

- The user of the bank should be able to search the desired services from the available ones.
- There should be appropriate documents' for users to read. This implies that when a user wants to open an account in the bank, the forms must be available so that the user can open an account.
- After registration, the user should be provided with a unique acknowledgement number so that he can later be given an account number.

The functional requirements should be complete and concise. Complete means that all the user requirements should be defined. Concise means that all requirements are specified clearly without any contradictory definition.

Non-functional requirements

Types:

Product requirements
Requirements which identify that the delivered product must behave in a particular way, e.g. execution speed, reliability etc.

Organisational requirements
Requirements which are a consequence of organizational policies and procedures, e.g. process standards used, implementation requirements etc.

External requirements
Requirements which emerge from factors which are external to the system and its development process, e.g. interoperability requirements, legislative requirements etc.

Product requirements:

These requirements specify how software product performs. Product requirements comprise the following.

Efficiency requirements: Describe the extent to which the software makes most appropriate use of resources, the speed with which the system executes, and the memory it consumes for its operation. For example, the system should be able to operate at least three times faster than the existing system.

Reliability requirements:

Describe the acceptable failure rate of the software. For example, the software should be able to operate even if a hazard occurs.

Portability requirements:

Describe the ease with which the software can be transferred from one platform to another. For example, it should be easy to port the software to a different operating system without the need to redesign the entire software.

Usability requirements:

Describe the ease with which users are able to operate the software. For example, the software should be able to provide access to functionality with fewer keystrokes and mouse clicks.

Implementation requirements:

Describe requirements such as programming language and design method.

Standards requirements:

Describe the process standards to be used during software development. For example, the software should be developed using standards specified by the ISO and IEEE standards.

Question No. 3:

Ans:

Software Requirement Engineering:

Software requirements engineering is the process of understanding and defining what services are required and identifying the constraints on these services.
Requirements engineering processes ensures your software will meet the user expectations, and ending up with a high quality software.
It's a analytical stage of the software process as errors at this stage will reflect later on the next stages, which will must cause you a tremendous costs.
At the end of this stage, a requirements document that specifies the requirements will be produced and validated with the stockholders.

There are four main activities of requirements engineering:

Feasibility study: An approximate is made of whether the identified can be achieved using the current software and hardware technologies, under the current budget, etc. The feasibility study should be cheap and quick; it should inform the decision of whether or not to go ahead with the project.

Requirements elicitation and analysis: This is the process of acquiring the system requirements through the observation of existing systems, discussions and communications with stakeholders, etc. This may involve the development of one or more system models and prototypes that can help us understanding the system to be identified.

Requirements specification: It is the activity of recording the data accumulated during the elicitation and investigation action into an archive that characterizes a lot of requirements. Two sorts of requirements might be present in this report; user and system requirements.

Requirements validation: It is the way towards checking the requirements for realism, consistency and fulfillment. During this process, our objective is to find out errors in the requirements document. Whenever errors are found, it must be improve to correct these problems.

System Requirement Engineering:

In System requirements Engineering we look into all those requirements at the system level that defines the functions which the system as a whole should manage to satisfy the stakeholder needs and requirements, and are expressed in an correct combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

System requirements play major roles in systems engineering, include:

- Form the basis of system architecture and design activities.
- Form the basis of system integration and verification activities.
- Act as reference for validation and stakeholder confirmation.
- Provide a means of communication between the various technical staff that interact throughout the project.

Elicitation of stakeholder requirements starts in Concept Definition and will be initially developed through interview and mission examination. System requirements are regarded in detail during System Definition. Neither can be considered complete until consistency between the two has been achieved, as demonstrated by traceability, for which a number of iterations may be needed.

Question No. 4:
Ans:

It’s difficult to get one customer to decide which of his requirements are most important. Gaining agreement among multiple customers with diverse expectations is even more challenging. We need to negotiate requirements.

Requirement Negotiation:

Negotiation is a decision-making process that involves interaction and interdependency and implies two parties with conflicts.

Conflicts are inevitable since stakeholders have mismatching goals.

User: many features, high level of service, early availability.

Customer: budget/schedule constraints, cost effectiveness, compliance with standards.

Developer: flexible contracts, stable requirements

Requirements negotiation is an iterative process through which stakeholders make tradeoffs between requested system functions, the capabilities of existing or envisioned technology, the delivery schedule and the cost.

Reasons why requirements negotiation is needed in software engineering:

- **Requirements discussion:**

Requirements that are highlighted as problematical are mentioned and also the stakeholders concerned gift their views concerning the necessities.

- **Requirements prioritization:**

Disputed necessities are prioritized to spot important necessities and to assist the choice creating method.

- **Requirements agreement:**

Solutions to the necessities issues are known and a compromise set of requirements ar in agreement. Generally, this can involve creating changes to a number of the necessities.

- During negotiation, the programmer reconciles the conflicts between what the client desires and what is achieved given restricted business resources

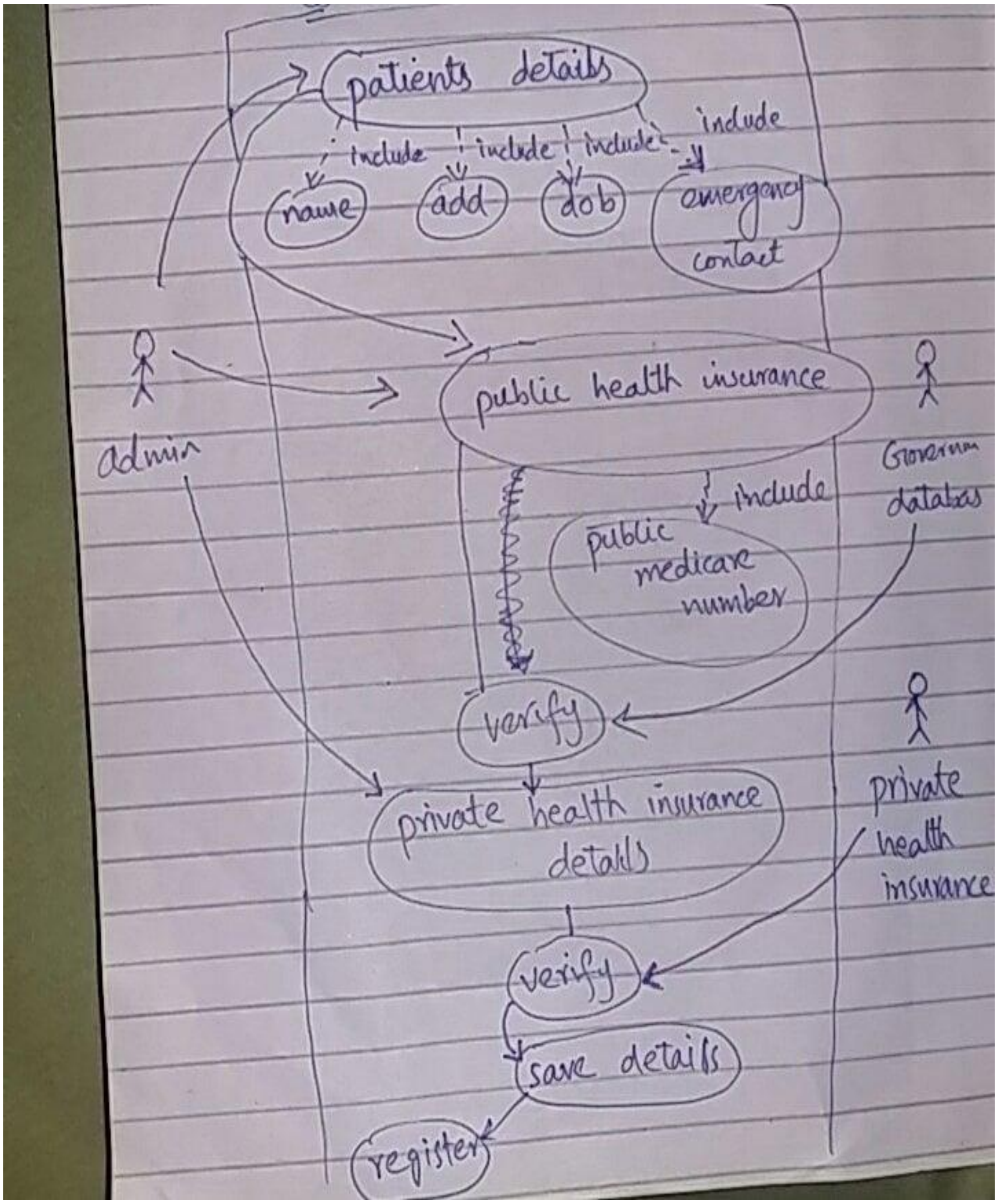
- Requirements are hierarchic (i.e., prioritized) by the shoppers, users, and alternative stakeholders
- Risks related to every demand are known and analyzed
- Rough guesses of development effort are created and wont to assess the impact of every demand on project value and delivery time
- Using an iterative approach, necessities are eliminated, combined and/or changed in order that every party achieves some live of satisfaction

Question No. 5:
Ans:
Actors:

Administrator,
Government Health Database,
Private Health Insurance System

Objects:
Patient
Administrator
Address
EmergencyContact
PublicHealthInsurance
PrivateHealthInsurance
Registration

Use case diagram of hospital management system:


patients details
include
include
include
include
name
add
dob
emergency contact
admin
public health insurance
include
public medicare number
Governm database
verify
private health insurance details
private health insurance
verify
save details
register