Experiment No 8

Identifying System requirements for an Architecture for any specific domain.

39_BECOMP-B_Harsh-Mishra

Learning Objective: Student should be able to understand System requirements for an Architecture for any specific domain.

Theory:

The purpose of a requirements architecture is to structure and organize requirements in such a way that the requirements are stable, usable, adapt to changes, and are elegant. When a requirements architecture is sound, it helps facilitate better design of the system it attempts to describe. When a requirements architecture is faulty, it can cause problems. When the requirements architecture is poor, the following problems result:

i. No one knows why a requirement was changed
ii. Requirements cannot be reused
iii. Traceability is superficial or unused by other teams
iv. Requirements reviews involve irrelevant information
v. Big picture of the system being built and reasons for building it are not well-understood
vi. It is important to keep in mind that the purpose of a good requirements architecture is tobuild working software that meets business objectives.

Software requirements must be testable, unambiguous, and concise, a requirements architecture must also possess certain attributes. The above blueprint provides some general guidelines for how to structure requirements, but keeping in mind the following attributes:

a. **Maintainable**: Whatever choices you make in organizing requirements, ensure that youcreate a structure that can adapt to changes in requirements.
b. **Traceable**: Do you know which requirements any given process flow step is traced to?
c. **Usable**: Consider the stakeholders in the org chart—are the requirements architected in sucha way that you could either produce output for each of them or such that they could navigateto the requirements in the tool and find the requirements objects that are relevant to them? The hierarchies and traces you create should be consistent: Don't create one hierarchy where the FRs are children of the models and another hierarchy for the same project where FRs are not children of the models but are traced to

them. The absolute worst thing to do is to list all requirements objects in a flat list or to manage your requirements in word or excel.

d. **Scalable**: Imagine your requirements architecture with 10 times the number of requirementsit has. Now imagine it with 100 times the number of requirements. Architectures should be able to support the addition of new requirements with minimal overhead.

e. **Elegant**: Are there just enough hierarchies to facilitate use? Are you repeating hierarchiesjust to make traceability easier? Does your architecture contain duplicate models or requirements?

f. **Generalizable**: The architecture approach should be repeatable. You ought to be able to gointo any project and no matter the domain uses the same approach to requirements architecture.

All architectures are tradeoffs – like in software architecture, you may need to sometimes sacrifice aesthetics for robust traceability or reuse. Or you may sacrifice usability for ease of exporting to external formats. Understand the tradeoffs you are making with your requirements architecture.

Domain-specific Software Development Various mechanisms of domain-specific software development are under investigation within the projects. Compositional mechanisms facilitate reuse of existing artifacts, including software. Generative mechanisms are used when needed components are not available.1 Constraint-based reasoning systems and module interconnectionlanguages are critical underlying technologies for software composition. Prototyping technologies underlie generation. The TRW project serves as a technology conduit from the prototyping community into the DSSA program.

Following are the system requirement of Domain-specific Software Development:

- Performance
- Scalability
- Availability
- Reliability
- Security
- Maintainability
- Flexibility
- Configurability
- Personalizabilty
- Usability
- Portability
- Conformance to standard

Software Engineering
Software Requirements Specification (SRS) Document

Harsh Mishra
5/9/2022

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Final Draft	Harsh Mishra	All sections being Filled	5/9/2022

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date
Harsh Mishra			5/9/22

Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date
Ms Drashti Shrimal			5/9/22

Contents

1. Introduction

1.1 Introduction

The project blood bank management system is known to be a pilot project that is designed for the blood bank to gather blood from various sources and distribute it to the needy people who have high Requirements for it. • The Software is designed to handle the daily transaction of the blood bank and Search the details when required. • It also helps to register the details of donors, blood collection details as well as blood issued reports. • The software Application is designed in such a manner that it can suit the needs of all the blood bank requirements in the course of future. • It will help us to find the Blood group with its most efficient time to take care of the blood and it is more easy to hand over the blood to the hospital to help people to get blood on time. • This all thing is been stored and been seen in this blood bank management system. To help more people trying best to do so

1.2 Scope of this Document

This application is built such a way that it suits for all type of blood bank in future.so every effort is taken to implement this project in this blood bank, on successful implementation in this blood bank, we can target other blood banks in the city.

1.3 Overview

Blood Bank Management Software is designed and suitable for several Blood Bank either operating as indiviuals organizations or part of organizations covers all blood banking process from donors recruitment, donor management, mobile session component preparation, screening covering all test, blood stock inventory maintenance, patient registration, cross matching, patient issues etc

1.4 Business Context

This product can be used by Blood Banks across the world and also as a supporting aid in hospitals. The revenue can be obtained from both the parties.

2. General Description

2.1 Product Functions

Here the system admin & the donor are the system users. According to my assumptions the donor who will register to the system from the website easy questions which are in English language & he/she has the ability to realize small instructions & fill the application without any errors & a small knowledge of computers to upload the health condition certificate to the systems. User is very generous to attend the donation with such a small announcement. (Email & SMS Messages).

2.2 Similar System Information

The product has been developed by eraktakosh.in that is government based platform.

2.3 User Characteristics

The users are mostly the patients, blood donors and hospital representatives.

2.4 User Problem Statement

The users' system, currently, is slow and inefficient as it relates to the checkout process. Bidders must wait hours to check out the item they have won. Too many man hours have been needed to enter the wealth of information collected.

2.5 User Objectives

The user wants a database that will store information on a silent auction. The program must faciliatate the speed and ease of input. It also must store the items the IDANRV needs to store.

2.6 General Constraints

Constraints include an easy to use interface for the program through forms, a Windows platform or, at bare minimum, a Mac with Access and Excel for Mac installed. Also, it must be constructed in Access, Excel, or another related program that is easily learnable.

3. Functional Requirements

- Login of admin
- Blood Donor.
    - Change the login password of admin.
    - Register the donor by himself
- . Register the donor by system admin
- . Login of the donor
- .Change the login password of donor.
    - Change personal, contact details by the donors himself.
    - Change personal, contact details by the system admin
- . Withdraw reg. details by the donor.
    - Withdraw reg. details by the admin.
    - Send blood donation details to the relevant donors
- . Send blood testing details.

4. Interface Requirements

The system is basically running on the official website of the govt, blood bank. Mainly there are 2 actors in the system. The system provides some advance features to the system admin than the donor. If the system admin logs in, the system interface provides some main command buttons to the admin.

• Change login password

. • Edit donor profile details.

• Search donors for a exact blood group and send messages.

• Print statements.

• Update the database.

• Send blood testing details.

• Search details from the database.

If the donors log in, the system will provide another different interface with different commands. • Change login passwords.

• Edit personal .contact details

• Details related to contributions to donations.

• Future blood donation details.

• Withdraw name from the system.

4.1User Interfaces

It has been required that every form's interface should e user friendly and simple to use.

4.2 Hardware Interfaces

• 1GHz or High processor

• 512 MB RAM

• 500 MB Hard Disk

4.3 Software Interface

Dept of CEA, GLAU, Mathura Smart blood bank

• Windows

• Internet Explorer, Chrome, Firefox

4.4 Communication Interface

There are:

• Should run on 500GHz, 64MB Machine.

• Should have a proper internet connection.

• The response time for occurs a change will be more than 4 seconds

. • The resp

5. Performance Requirements

The system is interactive and the delays involved are less. When connecting to the server the delays is based editing on the distance of the two System configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds for the sake of good communications

• 1GHz or High processor

• 512 MB RAM

• 500 MB Hard Disk

6. Other non-functional attributes

- **Usability:** The product will be made as simple as possible with respect to usability and this will be achieved by making a lucid UI. The modules namely donate blood and entering of hospital info will be made easy to use.
- **Performability:** The product will have good response time to user queries and the main motive of the product is pertaining to healthcare sector. To aid this, light weight frameworks like MERN stack, MongoDB will be used.
- **Scalability:** During peak time or during surge of requirements for blood, the product will be scalable to handle many requests. The modules namely Donate Blood, Hospital Info entering will be needed to make more robust so as to handle multiple requests at a time.
- **Maintainability:** Constant updates are easy to push into the platform. The Dashboard section where admin will view all of the donors and Patient data will be maintained from time-to-time basis to reduce errors and any breaches,
- **Security:** The most important attribute to ensure that user and hospital data is protected. We will need both physical security of blood. And from software perspective,we need some encryption algorithms to protect our data well. All of our modules will be protected. We will use Hashing Algorithms, MAC and Affine Ciphers to store user passwords and other sensitive data.

7. Operational Scenarios

Scenario A: Donor Login

The Donor will login into the portal and enter basic details and send his approval to donate blood

Scenario B: Patient Login

The patient will login and raise request for blood.

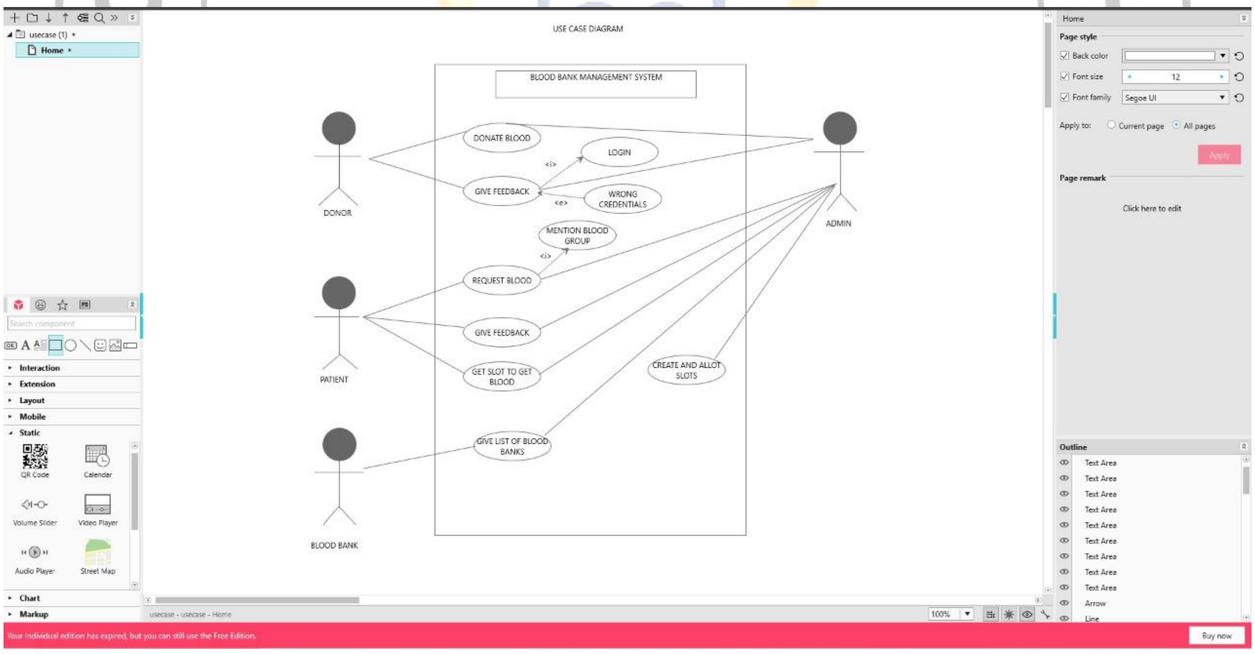
Scenario C: Hospital Staff Login

The hospital staff will se who all want blood and who all are ready to give the same and accordingly allot

8. Preliminary Use Case Models and Sequence Diagrams

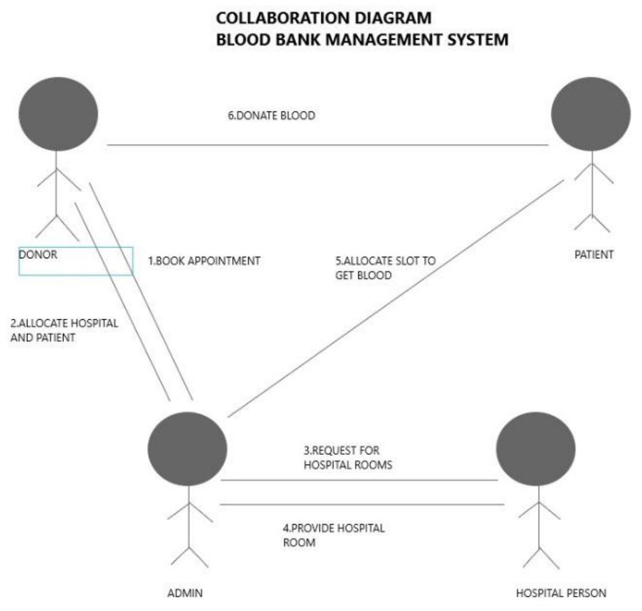
This section presents a list of the fundamental sequence diagrams and use cases that satisfy the system's requirements. The purpose is to provide an alternative, "structural" view of the requirements stated above and how they might be satisfied in the system.

8.1 Use Case Model



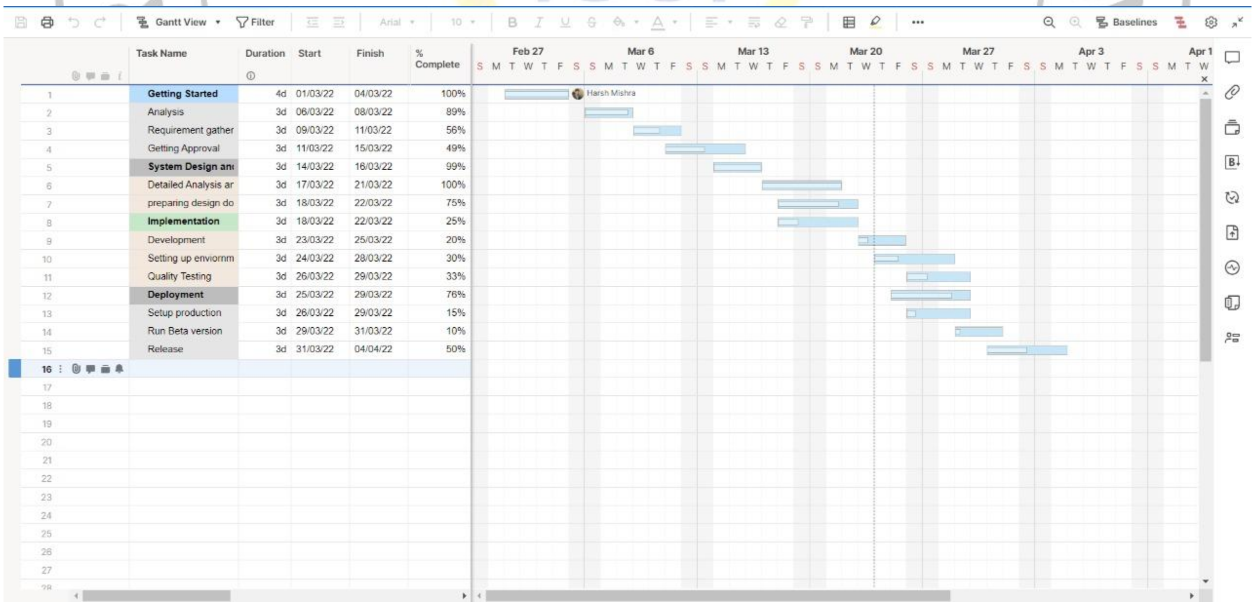
Page style
Back color
Font size 12
Font family Segoe UI
Apply to: Current page [✓ All pages]
Page remark
Click here to edit
usecase - usecase - Home
100%
Your individual edition has expired, but you can still use the Free Edition.

8.2 Collaboration Diagrams


COLLABORATION DIAGRAM
BLOOD BANK MANAGEMENT SYSTEM
6.DONATE BLOOD
DONOR
PATIENT
1.BOOK APPOINTMENT
5.ALLOCATE SLOT TO GET BLOOD
2.ALLOCATE HOSPITAL AND PATIENT
3.REQUEST FOR HOSPITAL ROOMS
4.PROVIDE HOSPITAL ROOM
ADMIN
HOSPITAL PERSON

9. Updated Schedule

The updated PERT/GANTT chart is attached at the end of the document



	Task Name	Duration	Start	Finish	% Complete
1	**Getting Started**	4d	01/03/22	04/03/22	100%
2	Analysis	3d	06/03/22	08/03/22	89%
3	Requirement gather	3d	09/03/22	11/03/22	56%
4	Getting Approval	3d	11/03/22	15/03/22	49%
5	**System Design an**	3d	14/03/22	16/03/22	99%
6	Detailed Analysis ar	3d	17/03/22	21/03/22	100%
7	preparing design do	3d	18/03/22	22/03/22	75%
8	**Implementation**	3d	18/03/22	22/03/22	25%
9	Development	3d	23/03/22	25/03/22	20%
10	Setting up enviornm	3d	24/03/22	28/03/22	30%
11	Quality Testing	3d	26/03/22	29/03/22	33%
12	**Deployment**	3d	25/03/22	29/03/22	76%
13	Setup production	3d	26/03/22	29/03/22	15%
14	Run Beta version	3d	29/03/22	31/03/22	10%
15	Release	3d	31/03/22	04/04/22	50%

10. Updated Budget

An updated budget is attached at the end of this document

11. Appendices

Result and Discussion:

Learning Outcomes: Students should have been able to understand

LO1: Define software System.

LO2: Identify different system requirements of software Architecture.

LO3: Explain system requirement of Domain-specific Software Development.

Course Outcomes: Upon completion of the course students will be able to understand System requirements for an Architecture for any specific domain.

Conclusion:

Viva Questions:

1. Define Software System.
2. Explain different requirements for creating a software Architecture.
3. Explain any five-system requirement.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	