**Birla Institute of Technology & Science, Pilani**

**Work-Integrated Learning Programmes Division**

**First Semester 2019-2020**

**Comprehensive Examination**

**(EC-3 Regular)**

Course No. : SS ZG653

Course Title : SOFTWARE ARCHITECTURES

Nature of Exam : Open Book

No. of Pages = 3

# No. of Questions = 6

Weightage : 45%

Duration : 3 Hours

Date of Exam : Saturday, 16/11/2019 (FN)

Note:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

**Answer questions 1 and 2 based on the following case:**

**Better World**

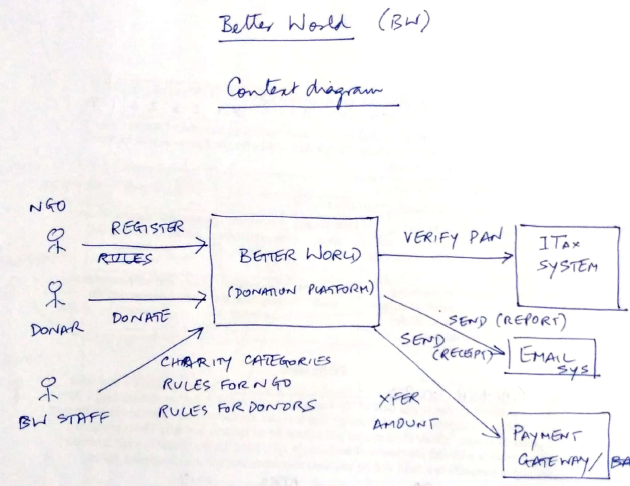
Better World is an online donation platform that helps channelize funds to credible non-governmental organisations (NGOs) across India. As a [web portal](https://en.wikipedia.org/wiki/Web_portal), it helps raise funds from individuals across India and the world and then disburses these donations to credible Indian NGOs.

NGOs register in the portal by providing their details and PAN number. The PAN number is verified against the Income Tax system.

Ten categories of charities are listed, such as charity for children, charity for education, health, etc. Donors can choose an NGO to make a donation. Payments can be made using card. Donors get an instant receipt via email. Later, donors also get a report via email telling them where exactly their donation was used.

One of the challenges is, government regulations for registering NGOs and accepting donations, keep changing. For example, donations greater than Rs. 10,000, should have PAN number of the donor.

1. **Answer the following:**
2. Draw a context diagram showing the systems involved and the key messages exchanged between them [2]



1. List the functional modules in the system [2]

Functional modules:

* Register NGO,
* Interface to Income Tax system (to verify PAN of NGO),
* Maintain categories of charities,
* Donation

Technical modules

* Interface to Payment gateway (to accept donations and transfer amount to NGO),
* Interface to Email system,
* Rules engine for NGOs (to check government regulations regarding NGO),
* Rules engine for donors (to check government rules regarding donations)

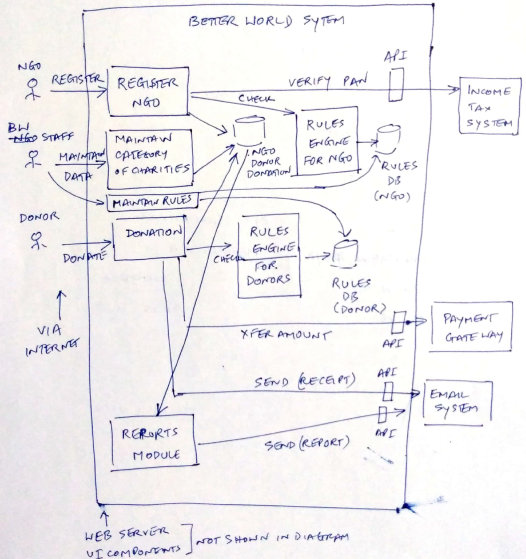
1. Identify 2 most important quality attributes of the system (other than Availability, Performance and Interoperability). Explain tactics to address them. [2]

(Usability, Modifiability.

Tactics for usability: Show the steps to register / make donation and indicate in which step the user is, provide confirmation after each step, do not clutter the screen with too much information.

Tactics for Modifiability, encapsulate the changing aspects such as government regulations in a separate module and provide stable interface to that module, such that even if changes are made to the module, its interface API does not change. Call this module after capturing all the details of the donation to check if all regulations are complied with or not)

1. **Answer the following:**
2. Draw an Architecture diagram showing functional & technical components, their inter connections and showing where the components reside. [5]



1. In the diagram, indicate key messages exchanged between components. Mention the communication methods used to interact with external systems. Justify the method. [3]

APIs of Income tax department providing user id and password of Better World Foundation to Income Tax system. and APIs to interact with Email system

1. Mention 2 architecture patterns (other than Layered pattern), used in the architecture. Mention where in the architecture, these patterns have been used. [2]

(MVC, SoA)

**Answer questions 3 and 4 based on the following case:**

**Sea Buoys**



A sea buoy is a floating equipment in the sea which provides weather and navigation data to passing ships. There are a number of free-floating buoys in the sea.

Here are the features and capabilities of the sea buoy:

Sensor data:

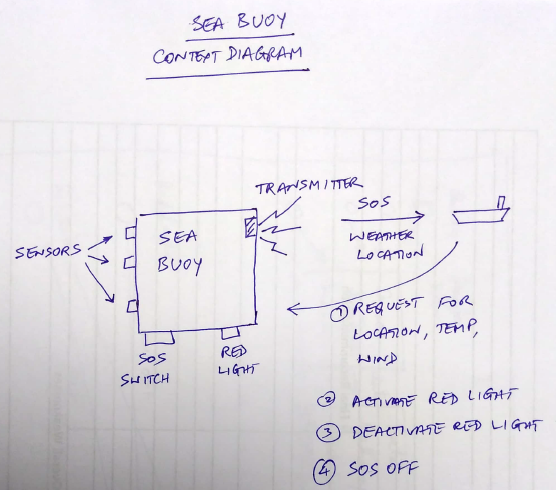
* Sea buoy collects data regarding air and water temperature, wind speed and location through a variety of sensors. Using a radio transmitter it broadcasts weather & location information every minute, so that passing ships can receive it.
* Wind speed readings are taken every 25 seconds, temperature readings every 10 seconds and location every 10 seconds. It stores this data for the last 24 hours. Also it maintains the running average of wind and temperature values.
* Passing ships can send a request to the buoy to get location, temperature and wind information of past 24 hours. This takes priority over the periodic broadcast.

Red light: The buoys are equipped with a red light that can be activated or deactivated by a request from a passing ship. This helps in search operations.

SOS switch: If a sailor is able to reach the buoy, he or she may flip a switch on the side of the buoy to initiate an SOS broadcast. This signal takes priority over all other broadcasts and continues until reset by a passing ship.

It should be possible to add new types of sensors or replace existing ones by more advanced sensors, in the future. In such a case, there should be minimal change to the buoy software.

1. **Answer the following:**
2. Draw a context diagram of the buoy system showing the different hardware components and the external systems. Show the key messages exchanged between external system and the buoy system. [3]



1. What are the top 2 ASRs and how would you address them? Note: do not just mention the quality attribute. Mention the scenario / requirement in detail. [4]

ASRs

* Give priority to requests coming from ships, requests such as SOS and request for data
* Ability to add / replace sensors of new types

Tactics

* Priority: When the request is received from ships, suspend regular broadcasts temporarily
* Modifiability: Have a sensor gateway / interface module that will shield the interface characteristics of the sensors. All modules wanting to interact with the sensors should go via the sensor gateway module. Thus we will be able to limit the modifications to the gateway module when new sensors are introduced.

1. **Answer the following:**
2. List all the functional and technical software modules of the system [2]

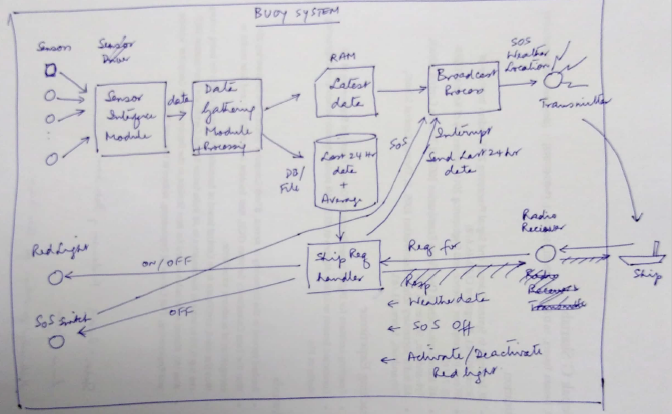
Functional modules:

* SOS Handler process
* Sensor interface module
* Sensor data gathering and processing module
* Ship message request handler
* Broadcast process

Technical modules:

* Sensor drivers
* Red light driver
* Transmitter driver
* Radio receiver driver

1. Draw a detailed architecture diagram of the buoy system, showing different hardware components and software modules to understand the working of the system. [6]



1. In the above diagram, show all the key messages and data being exchanged between various components – hardware and software. [3]
2. **Architecture patterns**
3. **Satellite health monitoring system**

Health monitoring of satellites is an important function for all satellite missions. Data received from the satellite are presented in different ways to understand the overall health of the satellite. Different techniques used to present the data are tabular display, graphical display, pictorial representations, etc.

What is an appropriate architecture pattern for this software? Justify. Draw a high level architecture diagram of the system using the pattern. [3]

MVC

1 mark for pattern and 2 marks for diagram

1. **Game software**

A game development company has developed a game that runs on mobile phones. The game can be played by 2 to 6 players sitting together in a room. When a player performs an action in the game, it is reflected on the screens of all players in a graphical manner. The good part of the software is that it does not use internet to communicate with different players. Rather it uses Bluetooth for communication.

What is an appropriate architecture pattern for this software? Justify. Draw a high level architecture diagram of the system using the pattern. [3]

Peer-to-peer

1 mark for pattern and 2 marks for diagram

1. **Architecting for the cloud**

TeamWork is a software product company that plans to develop a Collaboration software. This software allows employees of an organization to collaborate, brainstorm, discuss and improve the productive of the company. The features of the software include definition of Users, creation of discussion Topics, entering comments and creating discussion threads, etc.

TeamWork conducted a market survey to understand the market potential and found that there is indeed a good market for the product. However the expectations of organizations differ.

Large organizations (5000+ employees), are concerned about Security of information (such as topics & discussions). They expect very low down time (less than 1 day in a year) and a response time of less than 3 seconds for viewing and responding to discussions.

On the other hand Small organizations (less than 5000 employees) are not particular about security. Further, they are fine with a down time of 3 days in a year. Performance expectations are same as that of large organizations.

TeamWork wants to architect the product to cater to the expectations of small as well as large organizations. They want to offer this product on the cloud as a SaaS.

How would you architect & deploy this product? Provide detailed explanations. [5]

Answer:

Deploy one instance for all small organizations put together. This will have common database for all organizations, since security of information is not a concern for them.

Deploy one instance each for each large org with separate database for each org since they are particular about security of information.

To address the uptime expectations:

* Deploy the software on multiple servers. The number of servers needed would be different for small organizations and for large organizations since the uptime expectations are different.
* The number of servers would also be dependent on the reliability of servers and the quality of software (number of defects) because if the defects are more, the software may crash often and we need backup servers to handle the requests.

\*\*\*\*\*\*\*\*\*\*\*