- [Initiation *Phase*]: Explain? Triggers
    - [Problems]: Explain? Techniques for Identification
    - [Opportunities]: Explain? EG
    - [Directives]: Explain? EG
- [Project scope]: Explain? Benefits

- [Analysis *Phase*]: Explain

- [System requirements]: Explain
    - [Function requirements]: Explain
    - [Non-functional requirements]: Explain

- [Stakeholders]: Explain? Types
    - [Internal stakeholders]: Explain
    - [External stakeholders]: Explain
    - [Operational stakeholders]: Explain
    - [Executive stakeholders]: Explain

- [Information gathering technique]: Explain? Techniques
    - [Interviews]: Explain? Benefits? Disadvantages? Types
    - [Questionnaires]: Explain? Benefits? Disadvantages? Types
    - [Review existing documentation]: Explain Sample vs Blank form
    - [Observation]: Explain? Benefits? Disadvantages
    - [Research vendor solutions]: Explain
    - [Collect user comments and suggestions]: Explain
    - [Prototyping]: Explain? Benefits? Disadvantages

- [User requirements]: Importance
- [Requirement management]: Explain

1. How to identify stakeholders

Firstly, a stakeholder is any person who has an internet in an existing or proposed system. They can be technical or non-technical; internal or external

First step: Read and identify the most obvious stakeholders
Second step: Compare current stakeholder list with

Internal Operational stakeholders:

- Application developer: the interest of the developer in the system is having a working system built. A successful system ultimately means the developer will be fully paid. In addition, the developer may be paid to maintain the working system long term.

- Volunteers: the implementation of the system streamlines the volunteer assignment process. The system provides volunteers with ample number of reminders and offers them the ability to turndown volunteer shifts in advanced or last minute without having to confront the YBCA managing committee member, instead notifying the system which is less daunting. The system also only assigns volunteers to the role they are trained in.

Internal Executive stakeholders:

- Committee of the Youth Bowling Club Association (YBCA): the committee is the main driving force behind the push to implement the Offical-Eze system and is the main beneficiary of the system. The system removes the requirement of having the YBCA committee members having to manually select the appropriate volunteers for dates thus improving the YBCA efficiency in completing their day to day activities. The implementation of the system helps strengthens the relationship with the ABA which may open the door to many other additional benefits such as- increased funding to the YBCA, clubs, and upgrades to the venues, all provided by the ABA. Additionally, being more associated with the ABA potentially improves the YBCA reputation and recognition resulting in an expansion of clubs and club members.

- Bowling clubs: The Offical-Eze system reduces the role the club plays in providing the volunteers. Simply, the club provides volunteers to the ABA committee and from there the system takes over and assigns volunteers to matches. The system minimises the club's role which is means they can focus on more important things.

External Operational stakeholders:

- Children: the system interest them because it is the main system used to officiate and score their matches. A failure in the system in regards to assigning volunteers to their matches would result in delays with the children's scheduled matches. In addition, a failure in the system scoring the children's matches would result in an inaccurate tournament being adjudicated. In contrast, a successful implementation of the system would provide the most accurate method to score and designate officials to children's bowling matches.

External Executive stakeholders:

- Community sponsors: a community sponsor usually provides funding for a club or an organisation for the purpose of philanthropy or to promote their business. Both types of purposes normally expect the organisation receiving the funds to use the funds. The community sponsors for clubs expect their organisation to be promoted regularly. The creation of the system helps facilitate the YBCA bowling competition; so the community sponsors are more consistently promoted.

- ABA: the ABA has various interests in the system. The system facilitates closer association between the YBCA and the ABA. This enables the ABA to monitor YBCA volunteers and venues to help ensure the safety of the children as well as maintaining the integrity of the sport. Secondly, since the system stores both the scores and information about where the competitions are held, the system makes the process of ABA identifying future Australian youth bowling talents easier. Improving the ability to scout talent allows for a more talented National Australia Bowling team. Thirdly, YBCA managing committee requires the ABA to audit the Offical-Eze system; the YBCA provides the ABA a fee to incentivise them. So, the implementation of the system provides economic benefits to the ABA.

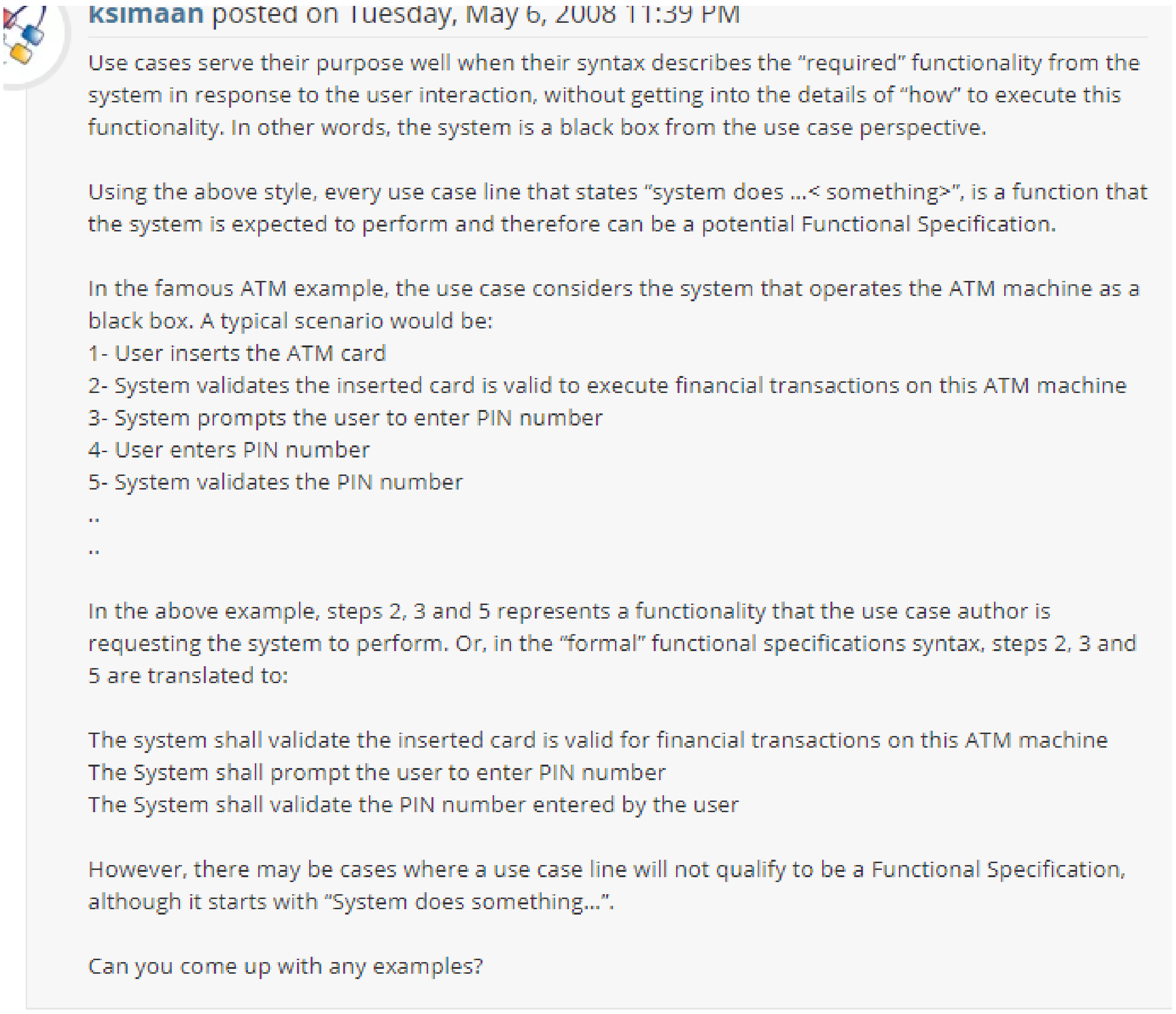
2. How to list functional requirements

Think high level →The simple (though not entirely complete) answer is that functional requirements describe what a system must *do* (higher-level design/functionality), while non-functional requirements describe *how* it should do it (lower-level implementation/architecture).

Firstly think of functional requirements as things the system must do WHAT not HOW eg: send text messages to volunteers. Also the system can be a stakeholder. In addition, maintain information is legit.

First step: Read the passage and think stakeholder/actor is the system and what does it do. A functional requirement should be a summary not word by word exact details of HOW. NOTE → **Different from what the user must do**

Second step: Because in my assignment I included what the student can/must do. Add it at the end. IE → Allow students to upload assignment. But try to avoid it and try to not be too specific



ksimaan posted on Tuesday, May 6, 2008 11:39 PM

Use cases serve their purpose well when their syntax describes the "required" functionality from the system in response to the user interaction, without getting into the details of "how" to execute this functionality. In other words, the system is a black box from the use case perspective.

Using the above style, every use case line that states "system does ...< something>", is a function that the system is expected to perform and therefore can be a potential Functional Specification.

In the famous ATM example, the use case considers the system that operates the ATM machine as a black box. A typical scenario would be:
1- User inserts the ATM card
2- System validates the inserted card is valid to execute financial transactions on this ATM machine
3- System prompts the user to enter PIN number
4- User enters PIN number
5- System validates the PIN number
..
..

In the above example, steps 2, 3 and 5 represents a functionality that the use case author is requesting the system to perform. Or, in the "formal" functional specifications syntax, steps 2, 3 and 5 are translated to:

The system shall validate the inserted card is valid for financial transactions on this ATM machine
The System shall prompt the user to enter PIN number
The System shall validate the PIN number entered by the user

However, there may be cases where a use case line will not qualify to be a Functional Specification, although it starts with "System does something...".

Can you come up with any examples?

Third step: read https://stackoverflow.com/questions/16475979/what-is-the-difference-between-functional-and-non-functional-requirement

3. How to list non-functional requirements

First step: Read PowerPoint for definition
Second step: Look at the examples below and adapt them to your questions. Some of them will be applicable for the question.
Third step: Look back and forth between PowerPoint and examples below

Usability

- Official-EZE system should provide step by step video tutorial on how to use
- Official-EZE system should allow for a built in text to speech reader to allow blind people to easily navigate the system

Reliability

- Official-EZE system should only be unavailable for use for a maximum of five hour per year this includes for maintenance, upgrades and unexpected outages
- Also maximum of one expected system upgrade every two months

Performance

- The Offical-Eze system client should respond in <1 second for any user input
- The Offical-Eze system must support 2000 users' active users at one time

Security

- Employees must change their initial password when they first log in
- Only registered users can access the system through a log in.
- The system will have distributed denial of service protection to keep the system from unexpected outages caused by it

Design constraint

- The Offical-Eze system will be ran on a cloud server specifically Amazon AWS
- Offical-Eze database can be accessed on both PC and smart phones.
- Offical-Eze can be used in any operating system

Implementation

- Offical-Eze system will be programmed In C++ programming language
- The database for the system is created using oracle and adopts oracle database syntax

Interface requirements

- Are irrelevant since our system does not need to send or receive data. Our Official-Eze system stores the data we only need. Additional we do not need other systems.

Physical requirements

- The Amazon AWS cloud server that runs our system contains 500GB storage available
- The Amazon AWS cloud server will have a proper cooling system
- Also, the Amazon AWS server has 16 GB of ram allocated for our system

Supportability requirements

- Automatic updates patching security vulnerabilities must be distributed instantly when the system has security vulnerabilities.
- The Official-Eze system must cost less than $200 a month to run.