Name: Faizullah khan

Student id: 14840

Paper: software engineering

Department: BS (SE)

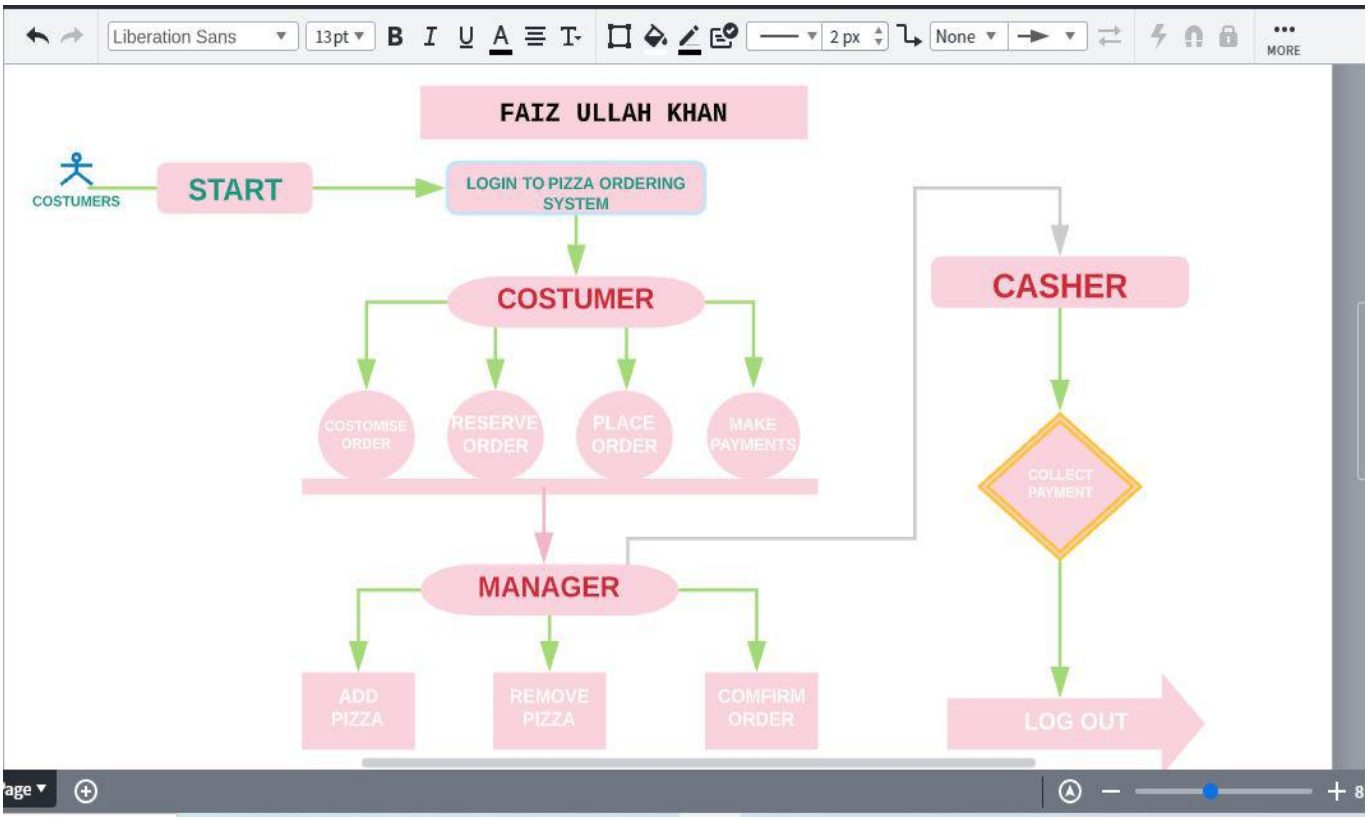
Submitted to: Engr. Ghassan husnain

Dated/18/04/2020

Q1:**The pizza ordering system**

The pizza ordering system allow the users of a web browser to order pizza for home delivery……?

Ans:



Q2: Suggest how an engineer responsible for drawing up a system requirements specification might keep track of the relationships between functional and non-functional requirements?

Functional requirements describe the system what it will do.

Example: inputs and outputs.

Non-functional requirements describe the expectations but it is not concerned with the system.

Example: security.

The official definition of 'a functional requirement' is that it essentially specifies something the system should do.

The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint upon the systems behavior. One could also think of non-functional requirements as quality attributes for of a system.

While drawing up a system requirements specification, an engineer might keep track of the functional and non-functional requirements by ensuring the following:

- The requirements needed to design meets the requirements such as compatibility, portability etc.
- Design the system so that it ensures the safety and security.
- Implementing the system in an efficient manner.
- The cost and time required for the development should not affect the design and implementation of the system

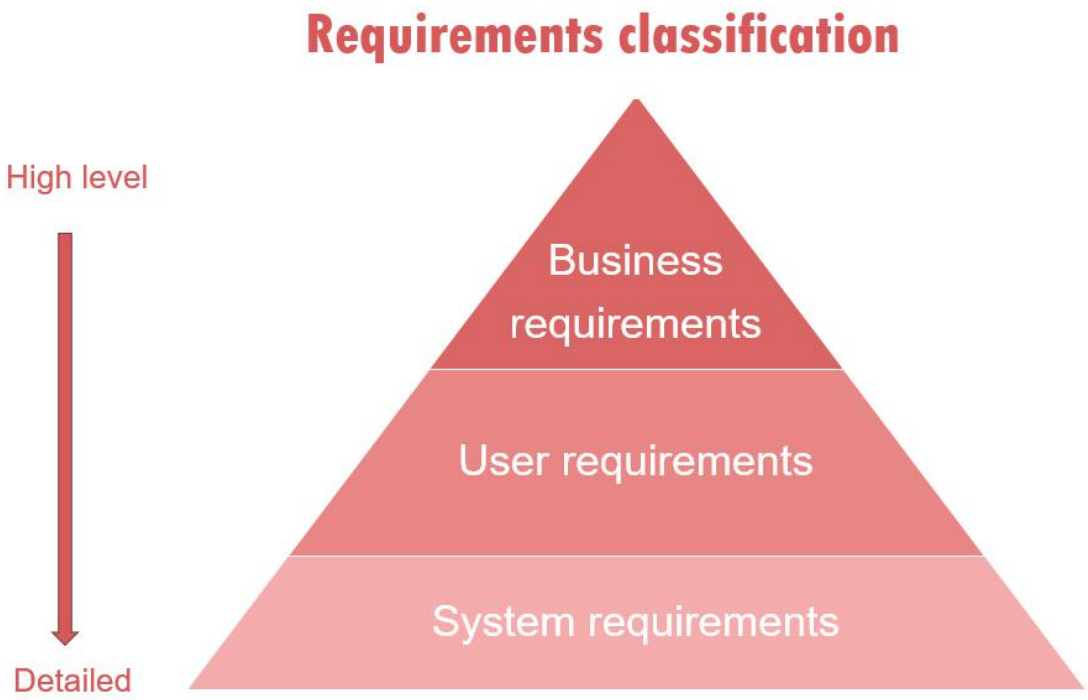
Here, the non-functional requirement defines what the expectations to get out are and the user requirements.

It does not conflict with each other.

The first step is to make the **Systems Requirement Document.**

It is engineer responsible to prepare documents to each functional and non-functional requirement.

- The engineer needs to prepare the document depending on this; Non-functional requirements need the **natural languag**e and functional requirements need the **structured language** to understand better.
- It gives the **matrix** that shows each requirement related to each other.
- It is **very difficult to manage** because the functional and non-functional requirements put efforts with each other on track of relationships.
- Non-functional requirements linked with functional requirements to list, identify the system levels that have related each other.
- The engineer needs to prepare the way to link the functional to non-functional to implement it.
- The functional requirements **enforce** the non-functional requirements that shall be recorded and tracked.



Q3: To reduce costs and environmental impact of commuting, your company decide ti close number of office and to provide support from home…….?

Ans: It is difficult to introduce agile methods into large companies for a number of reasons.

- Project managers who don't have experience of agile methods may be reluctant to accept the risk of a new approach, as they do not know how this will affect their particular projects.
- Large organizations often have quality procedures and standards that all projects are expected to follow because of their bureaucratic nature, these are likely to be incompatible with agile methods. Sometimes, these are supported by software tools (example requirement management tools) and the use of these tools is mandated for all projects.

- Agile methods seem to work best when team members have a relatively high skill level. However within large organizations there are likely to be a wide range of skills and abilities. And people with lower skill levels may not be effective team members in agile processes.

Change management and testing procedures are example of company procedures that may not be compactable with agile methods. Change management is the process of controlling changes to a system so that impact of changes is predictable and costs are controlled. All changes have to be approved in advance before they are made and this conflicts with the notion of refactoring. In XP, any developer can improve any code without getting external approval for large systems there are also testing standards where a system build is handed over to an external testing team. This may conflict with the test first and test often approaches used in XP.

Q4: Discover ambiguities or omissions in the following statement of requirements for part of a ticket issue system…….?

Ans: An automatic ticket-giving system sells rail tickets. Clients select their goal and info a charge card and an individual recognizable proof number. The rail ticket is given and their Visa account charged.

When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a personal identifier. When the credit transaction has been validated, the ticket is issued.

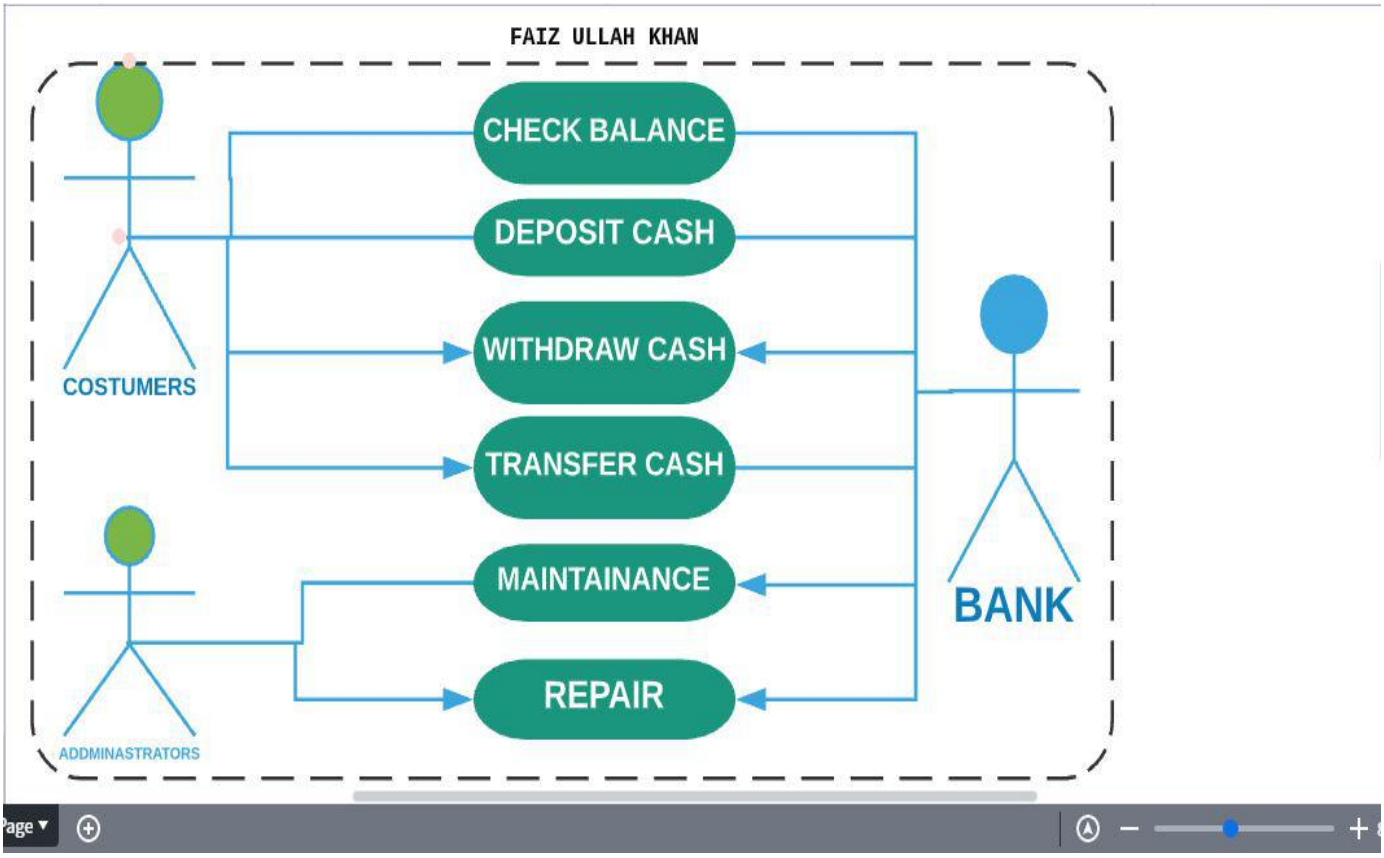
AMBIGUITIES AND OMISSIONS INCLUDE:

- Can a customer buy several tickets for the destinations together or must they be bought one at a time.
- Can customer cancel request if a mistake has been made.
- How a system should is respond if an invalid card is input.
- What happen if costumers try to put their card in before selecting destination (as they would ATM machine).

- Must the press the start button again if they wish to buy another ticket to different destination.
- Should the system only sell tickets between the stations were the machine is situated and direct connections or should it include all possible destinations.

Q5: Using your knowledge of how an ATM is used, develop a set of use cases date could serve as a basic for understanding the requirement for an ATM system?

Ans:



The End

Thank You Sir