

HOMEWORK 3- CHAPTER 4

Q1. Under what circumstances is it appropriate to represent an SRS using informal techniques only?

Ans. There are mainly 3 approaches for SRS representation techniques- formal, informal, semi-formal. But most preferable is formal techniques as it is well organized and managed as compared to other two techniques. Certain factors should be considered before selecting which method is best suited for representation, which are: complexity of technique, social issues, technical issue etc.

Informal techniques do not have any negative effective in the representation of the SRS document. These techniques include natural languages, flowcharts etc. It is much better to use the informal technique when we approach a scenario or workshop based techniques as it is much easier to represent data in flowchart format which is very easily understood by the large group in the workshops.

Q2. What can the behavioral specification provide that a requirements document cannot?

Ans. The behavioral specification is generated when the requirement engineers are asked to reverse the requirement for an existing system where the requirements don't exist or incomplete or out of date. **Requirement documents** are the business specifications such as calculations, business rules and process flow. **Behavioral specifications** are the specifications of user interactions with a system. They represent the perspective of the requirement engineer. The requirement document explains the 'what' basis, whereas behavioral specification explains the 'how' basics.

Q3. If the customer requests that future growth and enhancement ideas be kept, where can these ideas be placed?

Ans. Requirement documents is the most essential part of the project even if it a small or a big project. All the requests which the customers demand should be documented in this. It is preferred that all the additional requests are represented in the same document.

As the customer specifies the requirements, they should be documented and should be reviewed by the customer itself to make sure that the developers have understood clearly as to what they demand. After reviewing the documents, if customer requests that future growth and enhancement ideas be kept, then these ideas should be kept in any section of the SRS documents as it is easily accessible to both the requirement engineers and the customers.

Q4 What are some items to be included under "data retention" in the SRS?

Ans. Data retention, also called records retention, is the continued storage of an organization's data for compliance or business reasons. It basically explains the length of the time that data must be retained in the SRS. Some of the items that should be included under data retention are:

- Storage capacity
- Retention policy
- Data size
- Data retention history

- Data security etc

Q5. Here are some more examples of vague and ambiguous requirements that have appeared in real requirements specifications. Discuss why they are vague, incomplete, or ambiguous. Provide improved versions of these requirements (make necessary assumptions)

- i) The tool will allow for expedited data entry of important data fields needed for subsequent reports.**
- ii) The system will provide an effective means for identifying and eliminating undesirable failure modes and/or performance degradation below acceptable limits.**
- iii) The database creates an automated incident report that immediately alerts the necessary individuals.**
- iv) The engineer shall manage the system activity, including the database.**
- v) The report will consist of data, in sufficient quantity and detail, to meet the requirements.**
- vi) The data provided will allow for a sound determination of the events and conditions that precipitated the failure.**
- vii) The documented analysis report will include, as appropriate, investigation findings, engineering analysis, and laboratory analysis.**

Ans. The requirements above are not considered standard requirement as they are inconsistent. Some of the requirement even sound vague. For example, the engineer shall manage the system activity, including database. But this doesn't explain any other system activities other than database. Another example is, the data provided will allow for a sound determination of the events. Here 'sound determination' doesn't fit requirement category.

- The data entry tool used will allow for expedited data entry of important data fields needed for subsequent reports.
- The testing system will provide effective means for identifying and eliminating undesirable failure modes and/or performance degradation below acceptable limits.
- The database creates an automated incident report about the unauthorized access that immediately alerts the necessary individuals.
- The software engineer shall manage all the system activities, including the database.
- The report will consist of requirements data from the customer, in sufficient quantity and detail, to meet the requirements.
- The data provided will allow for a valid determination of the events and conditions that could precipitate the failure.
- The documented analysis report will include, as appropriate, investigation findings, engineering analysis, and laboratory analysis of the given subject mentioned by the customer.

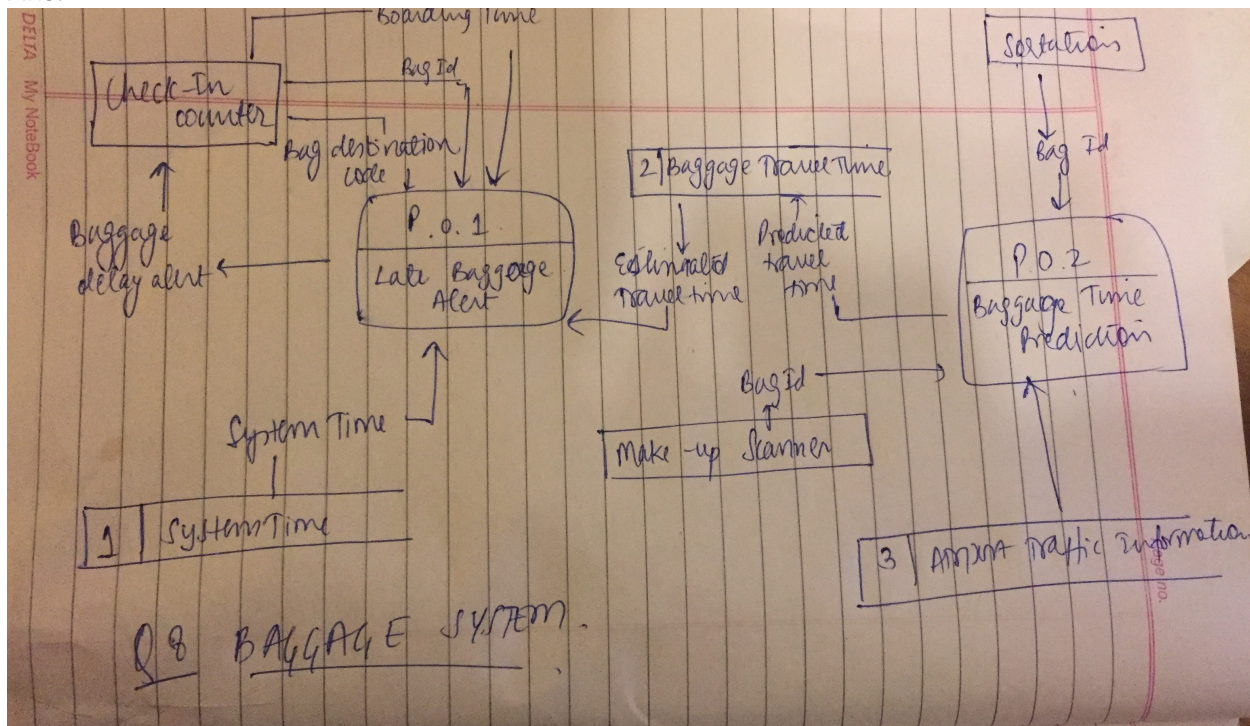
Q6. In section 9.4 of SRS of Appendix A, which requirements are suitable for representation using the "measurable targets" in the format shown in Figure 4.6?

Ans.

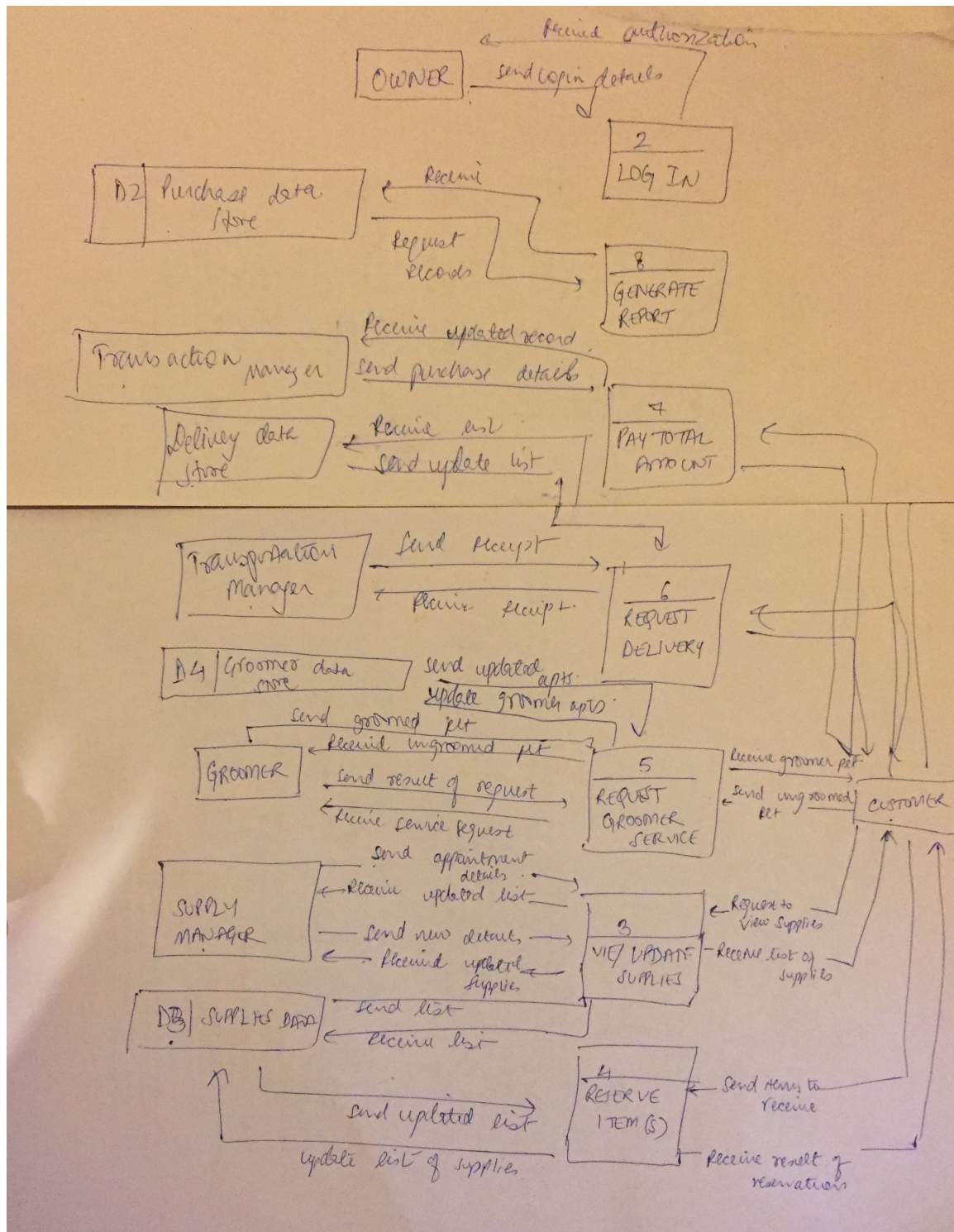
- 9.4.1- The system shall control any number of indoor irrigation access points.
- 9.4.7 - System shall maintain indoor watering information for no less than 45 days
- 9.4.9- System shall send notifications if moisture levels drop below user-defined floors for more than 4 hours.

Q8. Draw a context diagram for baggage handling system, make whatever assumptions you feel like.

Ans.



Q9. Draw a context diagram for pet store POS, make whatever assumptions you feel like.



Ans.

Q10. Research different techniques for creating context diagrams and prepare a report highlighting the strengths and weakness of each.

Ans. A context diagram is a graphic design that clarifies the interfaces and boundaries of the project or process at hand. It not only shows the process or project in its context, it also shows the project's interactions with other systems and users.

A context diagram will fall into one of two categories of rigor:

- The first lacks any formal structure; an object is simply placed in its context, showing its interaction with external entities from a high level. This may also be used in informal settings even by context diagram experts.
- The second type drawing from the same rules, syntax, and symbols established for data flow diagrams. In this instance, the context diagram is a subset of a data flow diagram with the context diagrams being the simplest form of data flow diagrams.

A project can use multiple context diagrams. A context diagram will also reveal omissions and errors in a business plan or business requirements so that any necessary corrections can be brought to light and addressed before a project is deployed. The goal is to get feedback from a project's stakeholders and identify any missing pieces while the project is still in the discovery stage.

The main parts of a context diagram are:

- **The *process***, represented as a rounded rectangle, which shows a given process or activity at its highest level. A process must react in a preplanned way, and indicates where data is transformed, stored, or distributed.
- The ***external entity*** may be an actor (person or thing) that either triggers the process or receives output from the process.
- ***Data flows***, represented as arrows, are the connectors between the main process and the various external entities and show data flow among them.

We can create context diagrams by:

1: Using a white board or other flexible writing tool, draw a context diagram for the highest-level process at hand (known as level 0). Once this is completed, that high-level process may be further decomposed into sub-processes. If the sub-processes are independent of each other, they may each be made into a separate context diagrams (not on level 0) with their own external entities and data flows.

2: For each distinct high-level process draw the process that acts upon the input. Place the process in the center of your white board. Label each process with a unique numeric identifier that will enable easy reference and revision in your requirements. Use a verb-noun structure to label the process.

3: Next, you will *identify and document all external entities* that are sources of data to the process you just listed. List all the external entities you can think of on the margin of the document.

4: Next, *capture the interactions* between this first listed source and the process. Determine what input the source provides into the process. Draw the arrow and label it accordingly. Determine what output the process returns to the source, and draw it accordingly.

5: Now *document the additional sources you've already listed and their data flows*. Determine for each of the remaining sources if it does something different from the source already placed on your diagram. If a source initiates the same input into the process as a previous source,

group them. Otherwise, place it into its own box and draw the data flow. Repeat until all sources are off your list.

6: *Identify and document additional external entities* and don't forget about entities which need data from the process being studied. Draw their inputs and outputs.

7: *Identify and document high-level events*. For each context diagram, brainstorm these by asking, "How could a source interact with this process?" Document these events on the margin of your context diagram. High-level events will be used as inputs.

8: *Capture additional requirements*. If you happen to discover a requirement during the creation of a context diagram, be sure to note it either in your requirements document or in a separate requirements repository designed specifically for requirements unearthed from the creation of context diagrams.

Citation: https://en.wikipedia.org/wiki/System_context_diagram