

# INFRASTRUCTURE MANAGEMENT

SOFTWARE ENGINEERING PROJECT

*-- For Prof. K. Chandrasekaran*

---

## SUBMITTED BY

### 1. PAVAN VACHHANI

- Roll No.: 16C0151
- Email: [vachhanihpavan@gmail.com](mailto:vachhanihpavan@gmail.com)
- Mobile: 9611592296

### 2. KAUSHIK

- Roll No.: 16C0120
- Email: [kaushik5699@gmail.com](mailto:kaushik5699@gmail.com)
- Mobile: 9482346989

## BRIEF IDEA

Infrastructure Management software can be used for administration of institutions with huge campuses, having various facilities and vast area to manage. The software can be used to effectively handle and facilitate the members of the institute to use the facilities.

Our software has two types of roles, user and admin, where user can request booking of some room or location along with the required assets and admin can approve according to availability. Admin can also keep track of various assets and charge fine in case of late submission.

Suppose we take an example of college like NITK. NITK has many class rooms, seminar halls, labs, discussion rooms, etc. It also has many assets which include computer systems, CNC machines, microscopes, benches, etc. All these things can be booked and tracked by the users like students or teachers according to the authorization. A user can request for a booking and admin can either approve or disapprove accordingly.

## WORK ENVISAGED FOR LAB

We will follow the below stages for our project in the lab:

1. To identify the process model suitable for development.
2. Design domain model, ER diagrams and DFDs.
3. Create SRS and FRS.
4. Design UI/UX.
5. Implement of high level design
6. Reimplementation of the same with proper changes.
7. Testing and Debugging

## PART – I SOFTWARE CRISIS

### CASE – I

---

#### POSSIBLE REASONS OF FAILURE AT CUSTOMERS SITE

1. Number of requests exceeding the limit.

Description: The number of requests of booking are exceeding than the system can handle. This can lead to database crashing.

Solution: This problem can be solved by using more number of data servers that allows more number of users to connect to the database and request for information. Also, we can increase the capacity of the servers to handle to requests by upgrading the firmware of the server.

2. Company's system doesn't meet the requirements

Description: The computer systems and data servers that the customer is using is very primitive of the software, which may lead to incorrect data processing or failure in booking requests.

Solution: The customer must inform us with the exact specifications of the company's systems so that this error can be prevented. Also, the software can be tuned to work in lower level systems by sacrificing performance which is not recommended.

3. Interruption by foreign software causing system crash

Description: Any foreign software interrupts the software's working by deleting any system files or database can lead to crash. Also bugs in the dependency software can lead to crash.

Solution: It is the duty of the customer to provide a safe working environment for the software, so that it is not interrupted by any other competitive software installed previously on the systems. All the dependencies should also be solved properly.

4. Commercial pressure

Description: As a software company, we may have other clients where different software needs to be delivered. Some developers may be transferred to other projects due to urgency or may be removed due to financial crisis. This can lead to improper debugging, which can crash the system.

Solution: Commercial pressure can happen due to tight schedules and poor planning. We as developers need to schedule the work and distribute those work to specified members of the team. Also, we need to set the schedule that can be followed by the developers, rather than trying to rush and end up with failures.

## CASE – II

---

### REASONS FOR DELAY IN DEVELOPMENT

1. Unavailability of needed resources.
  - Lack of resources like computers, management teams, developers, etc. due to financial or other reasons can lead to delay in development.
2. Poor communication among customers, developers, and users.
  - In case there is no communication or very poor communication amongst the customers, developers and users then it would be very difficult to decide the requirements.
3. Development Downtime.
  - It can be due to environmental factors, application bug or error, infrastructure or software failure, virus, hacker, network/hardware failure and operator error.

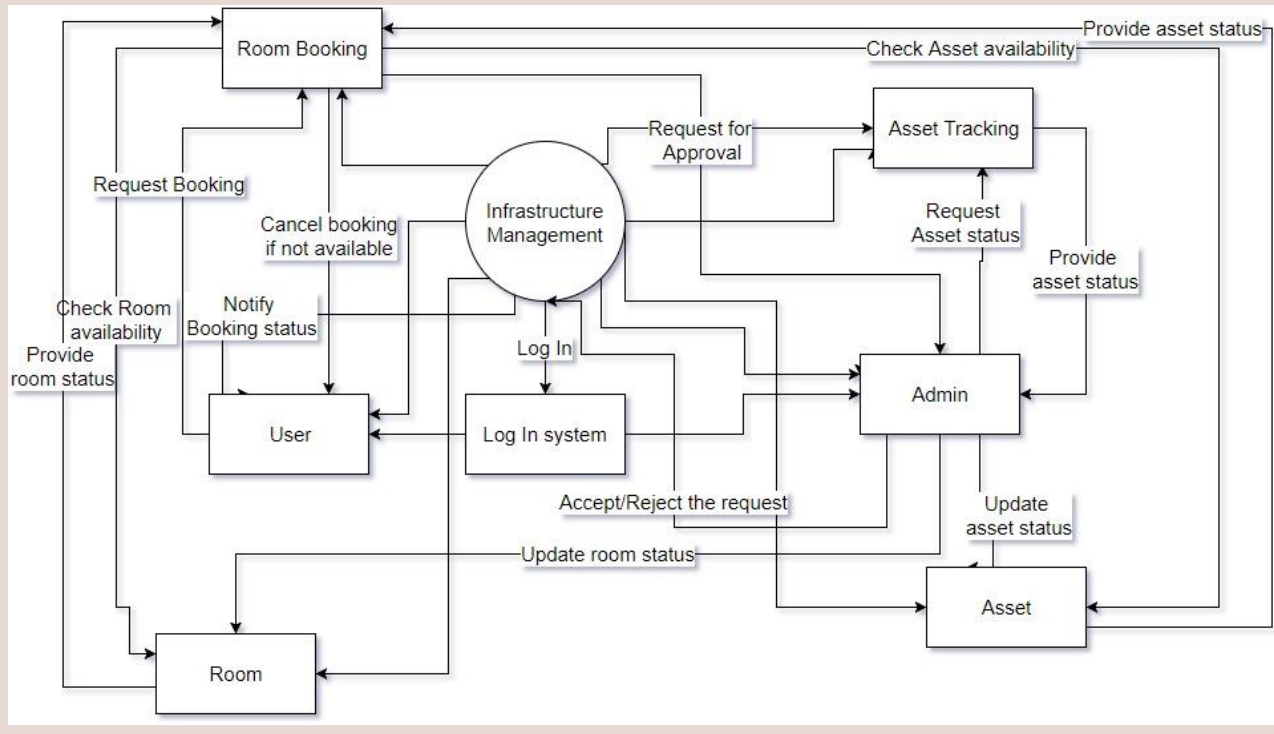
---

### EFFORTS TO CONVINCE THE CUSTOMER

We as developers will try to explain our process model to the customer and would like to communicate more with them. As the fault is not only ours we would communicate to them that we need to have access to more information about your institute. Also we can share the downtime security breaches that occurred during the development process. Other environmental factors are unavoidable so, they should understand the situation.

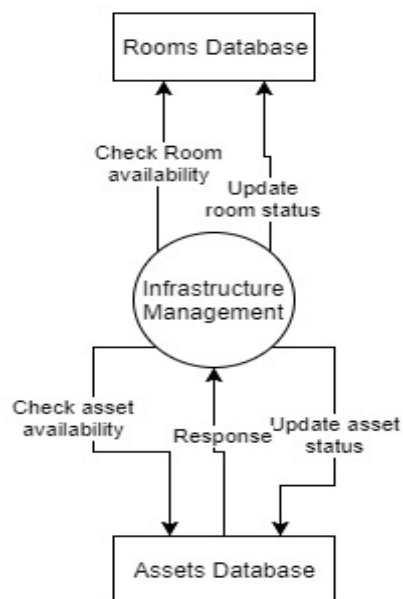
## PART – II CONTEXT DIAGRAM, DATA FLOW DIAGRAM AND ER-DIAGRAM

### QUES – 1 CONTEXT DIAGRAM

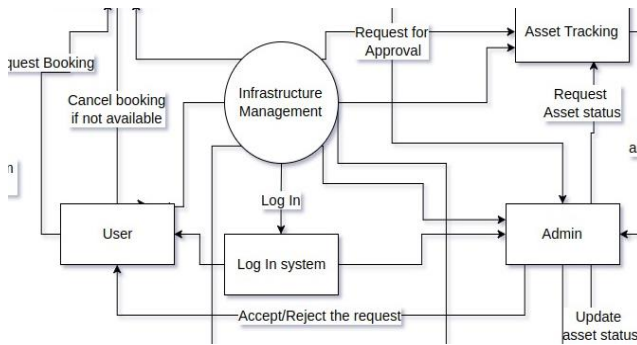


### QUES – 2 ERROR IN DATA-FLOW DIAGRAMS

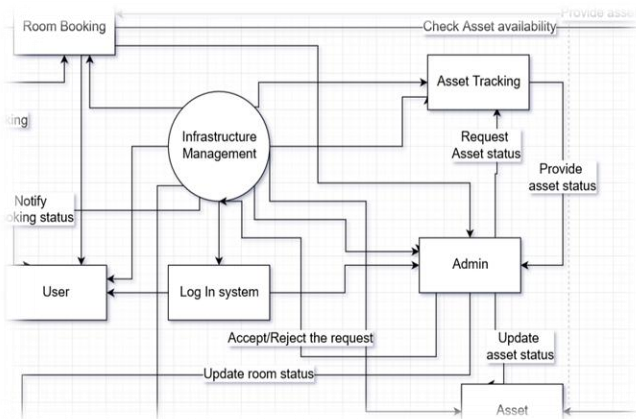
SOME OF THE POSSIBLE ERRORS IN MAKING THE DATA FLOW DIAGRAMS ARE:



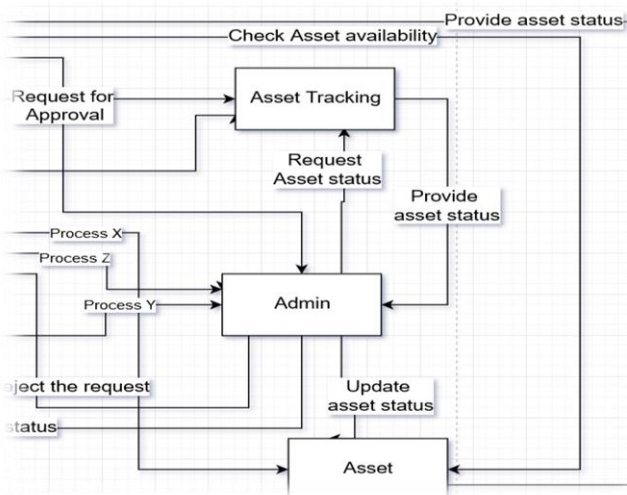
1. Forgot to include a dataflow, or pointing the arrow in wrong direction. This can lead to no response from the corresponding entity. For example, in the diagram Rooms database is not responding to the query of availability, leading to break in dataflow. It should be similar to assets database with proper response.



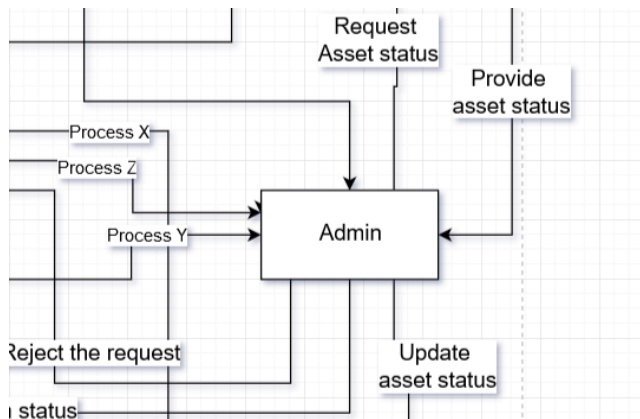
2. Directly connected the external entities without processing by the system. One page or entity cannot communicate with other without passing through the software, due to lack of interface. For example, in the figure, the admin is directly connected to the user, which is not possible. It needs to pass the response via the software.



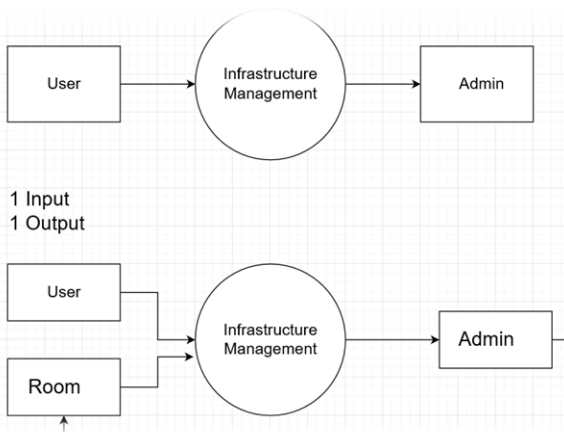
3. Incorrectly labeling processes or data flow. Inspect the data flow diagram to ensure that each object or data flow is properly labeled. A process should indicate the system name or use the verb-adjective-noun format. Each dataflow should be described with a noun. For example, in the figure there are no labelling for the connections that are made between admin and log in system. There has to be definition to what that connection means between the entities.



4. Including more than nine processes on a data flow diagram. Having too many processes creates a cluttered diagram that is confusing to read and hinders rather than enhances communication. If more than nine processes are involved in a system, group some of the processes that work together into a subsystem and place them in a child diagram. For example, there are numerous processes that have been connected to admin with definition. This should be avoided as those processes are not required by the admin for the event to take place.

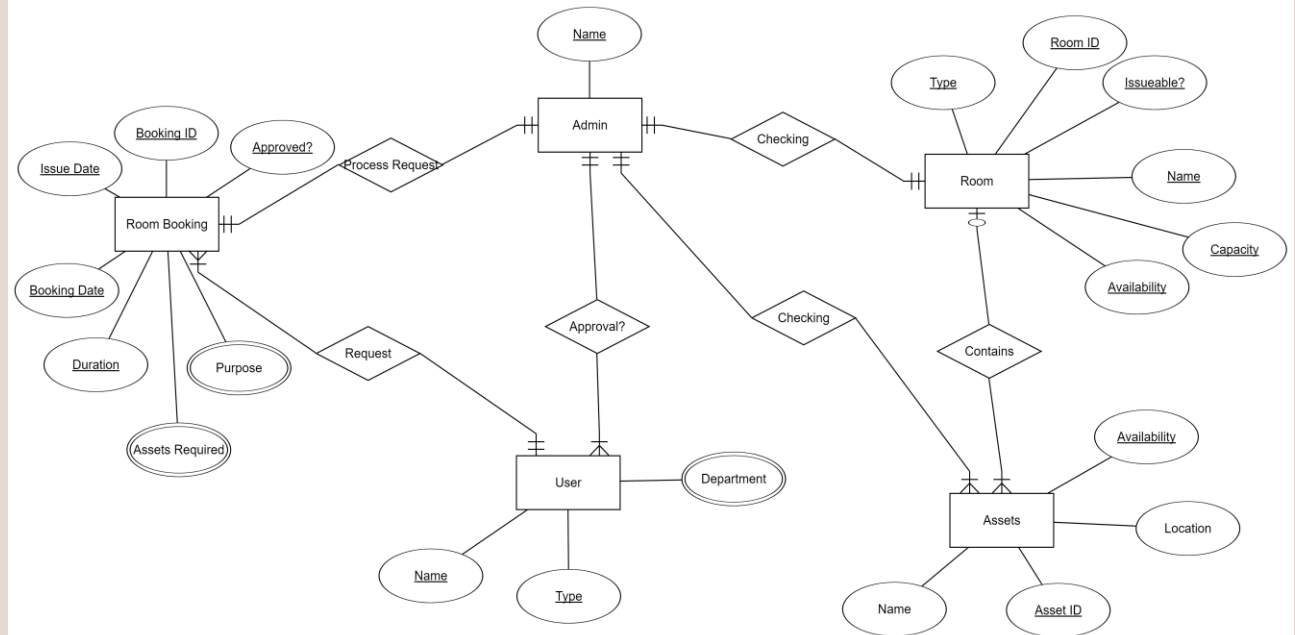


5. Omitting data flow. Examine your diagram for linear flow, that is, data flow in which each process has only one input and one output. Except in the case of very detailed child data flow diagrams, linear data flow is somewhat rare. Its presence usually indicates that the diagram has missing data flow.



6. Creating unbalanced decomposition (or explosion) in child diagrams. Each child diagram should have the same input and output data flow as the parent process. An exception to this rule is minor output, such as error lines, which are included only on the child diagram. For example, in this diagram set contains two diagrams where in the first one which is 0 level Data Flow contains 1 input and 1 output whereas the second diagram which is a level 1 Data Flow contains 2 inputs and 1 output.

### QUES – 3 ER DIAGRAM WITH CHANGES



The above diagram was the ERD according to the previous customer requirements.

The customer then added assets tracking feature to their requirements. It also has features like fine in case of late submission and tracking the issuer of asset.

The new ERD according to the new requirements is as below.

