Software Engineering Assignment (Extra)

*For Professor K. Chandrashekharan*

Submitted By:

Pavan Vachhani (16CO151)

Kaushik I (16CO120)

COUPLING AND COHESION

Modularization is a technique to divide a software system into multiple discrete and independent modules, which are expected to be capable of carrying out tasks independently. These modules may work as basic constructs like classes for the entire software. Developers use modularization to make maintenance, testing and debugging easier.

When a software is modularized, its tasks are divided into several modules based on some characteristics. But for proper functioning of the software, these modules need to interact with each other. There are measures by which the quality of a design of the modules and their interaction among them can be measured. These measures are called coupling and cohesion.

# COUPLING:

## WHAT?

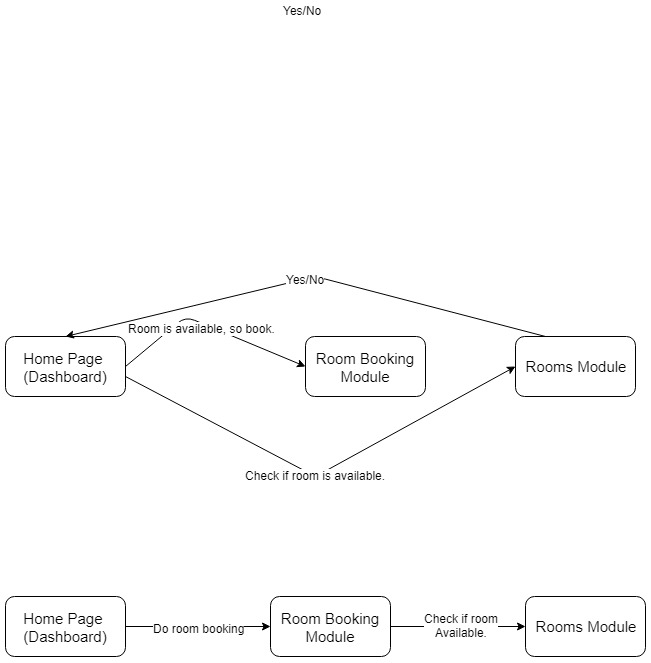
The degree of interdependence between two modules in a software system is called coupling. It is the measure that defines the level of inter-dependability among modules of the program. It tells at what level the modules interfere and interact with each other.

In our software of Infrastructure Management, Asset Tracking is a separate module where all assets can be listed from all rooms and locations. Assets is a module which handles database of the assets, adding new assets, editing, deleting them, etc. These modules are interlinked because the tracking module will send the query to assets module for fetching data from database and same data will be returned. This is called coupling between the modules.

## WHY?

Coupling is very necessary because zero coupling or modules with no dependency are not very usable. There can be considered as programs and not software. But high coupling is also not advantageous as it makes reuse of the code harder. It make understand-ability of the code harder, and forces global changes while debugging which is time consuming. Hence low coupling is maintained by developers, this striking balance is an art.

## HOW?

Low coupling can be achieved by having less classes linking to one another. The best way to reduce coupling is by providing an API (interface). OOP languages like Java, C++ offer mechanisms like public classes, public methods, friend classes, interfaces, etc. to interact between different classes and modules. In our project we have reduced coupling to perfect balance by reducing links among the modules. Suppose for booking a room, user needs to check if that room is available from Room module and then give the request for booking it. But we included availability checking in the Room booking module only. This reduces a link as shown in the figure.

Thus, we reduced coupling to some extent, so that changes at later stages can be made easier.

# COHESION:

## WHAT?

Cohesion is a measure that defines the degree of inter-dependability within elements of the module. The measure of strength of functional relatedness of elements within module is cohesion. It is natural extension of the information hiding concept. A cohesive module performs single task within a software procedure, requiring little interaction with procedures being performed in other parts of a program.

## WHY?

Cohesion helps in limiting the task performed by one module to just one thing. High cohesion is thrived for while software development. It makes it easier to understand what a class/method does, use descriptive names and reuse classes/methods.

## HOW?

Cohesion can be achieved by dividing the modules in to multiple completely unrelated (ideally) methods or functions. It can also be achieved by performing series of related actions one after other, such that return of one is parameter for other.

In our software, cohesion can be seen is Rooms module. It has various methods to create new room, edit room, delete room or view details. Each of these view, new, edit and delete actions are unrelated but in a single class. Also in room booking module, cohesion is followed. When user wants to book, it checks room availability, then asset availability and books the room. Thus it follows the sequence where each method uses return value of previous one.

## DIFFERENCE BETWEEN COUPLING AND COHESION

* Cohesion is the indication of the relationship **within** module while coupling is the indication of the relationships **between** modules.
* Cohesion shows the module’s relative **functional** strength but coupling shows relative **independence** among the modules.
* Cohesion is a degree to which component focusses on the **single** thing. Coupling is a degree to which a component/modules connect to **other** modules.
* **High** cohesion and **low** coupling is desired in a software.

CONCEPTUAL DESIGN AND TECHNICAL DESIGN

Design is the creative process of transforming the problem into a solution. It is the description of the solution.

# CONCEPTUAL DESIGN

## WHAT?

Conceptual design tell the customer exactly what the system will do. It is designed specifically for the customers to refer.

It is an early phase of the design process, in which the broad outlines of function and form of something are articulated. It includes the design of interactions, experiences, processes and strategies. It involves an understanding of people's needs - and how to meet them with products, services, & processes. Common artifacts of conceptual design are concept sketches and models.

## WHY?

Conceptual design is made for deciding if customer needs are fulfilled. The development team must also consider the practical feasibility of developing different concepts. They compare the estimated costs of developing, manufacturing and marketing the product with its potential value to customers and the price point it could achieve. This design also helps in estimating the cost of the project.

## HOW?

Conceptual design is made by designers with few points in mind like:

* Written in customer’s language
* Explaining the observable external characteristics of the system
* Contains no technical jargon
* Describes the functions of the system
* Is independent of implementation
* Is linked to the requirements documents

In our project, conceptual design will include the purpose of the project, functions which are to be implemented. It should contain details where physical interaction is required. For doing the fine payment transaction, admin should take the fine money and manually enter the transaction ID from the register. These are included in this design.

# TECHNICAL DESIGN

## WHAT?

Technical design is the kind of design that is agreed between software architects and software developers. It describes how the system will be built to meet the functional design. It contains detail and terminology not appropriate to business customers but needed by developers. It might describe screen layouts, it might describe database table and column names, it might discuss communication protocols and file formats, it might discuss server and operating system versions and dependencies. It should be detailed enough to enable unit tests, code, configurations, and technical documentation work to begin.

## WHY?

Technical is very necessary for the developers to understand the requirements and start coding the software. Also it helps and enables unit tests, configurations and technical documentation. It is also referred by customer’s engineers while maintenance, developers while working on next version, etc. Thus, it is very important document.

## HOW?

Technical design is made with following points in:

* It is used by software developers.
* It should contain proper technical terms.
* It should show hierarchy and functions of the software components.
* Define data structures and data flow to follow.
* Show interfaces.

The designs of the dashboards of various types of users should be included here. Database to be used like MySQL, columns to be added for each room/asset needs to be specified here.

EXCEPTION HANDLING AND FAULT TOLERANCE

# EXCEPTION HANDLING

## WHAT?

To throw an exception is to signal that the condition it represents has occurred. To catch that exception means to transfer the control to an exception handler, which defines the response that the program takes when the exception occurs. Exception occurs when an object cannot fulfil a responsibility for some reason.

## WHY?

Exception handling is done in the software to avoid following:

* Not allow users from entering wrong information or within particular time.
* Avoid dropped communications.
* Prevent bad data such as corrupt data, inconsistent data, missing files, etc.
* Avoid coding or logic data.
* Stop invalid requests with malformed arguments.

## HOW?

Exception handling can be done by focussing on the following points:

* Objects that interface to the user and the rest of the system.
* Objects within the system and objects that interface with external systems.
* Objects at different abstraction levels
* Objects designed by different developers.

In Infrastructure Management, we need exception handling for deleting the rooms/assets part. It can happen that super-user tries to delete the room which is allotted to a user which can lead to exception.

# FAULT TOLERANCE

## WHAT?

Software fault tolerance is the ability for software to detect and recover from a fault that is happening or has already happened in either the software or hardware in the system in which the software is running in order to provide service in accordance with the specification. Software fault tolerance is a necessary component in order to construct the next generation of highly available and reliable computing systems from embedded systems to data warehouse systems. Software fault tolerance is not a solution unto itself however, and it is important to realize that software fault tolerance is just one piece necessary to create the next generation of systems.

## WHY?

In software systems like safety-critical systems, life-handling systems, etc. faults in the middle of the operation needs to be avoided as far as possible. For example in software for X-ray machine, if machine doesn’t control intensity of the rays, it can kill the person. Thus such systems need to be made fool-proof as much as possible. Also to make system highly available, it is used.

No system is 100% perfect but we need to make it nearly 100%. Fault tolerance helps to avoid inconveniences due to minor problems in some component of the software. But this field is not properly developed and needs proper tools.

## HOW?

Some of the techniques to provide fault tolerance are:

* Recovery Blocks
* N-version Software
* Self-Checking Software

In Infrastructure Management there is a probability of occurrence of fault in the system of databases, these faults are automatically detected by the SQL system and server s and are corrected immediately by the administrators. Sometimes these faults are minor and can be tolerated by the system and are not notified to the administrators.