SOFTWARE ENGINEERING

PROJECT REPORT

Project Title
Online LHC room booking system.

Team Members
Manav Gopa l(B19CSE112)
Kartik Narayan (B19CSE110)

Document Type
Software Design and Architecture

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IIT JODHPUR

SOFTWARE DESIGN AND ARCHITECTURE

SOFTWARE DESIGN

What is software design and software design pattern and how we used it in our project

In software engineering, a design pattern is a general reusable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

Used reference for the design of our software- Reference

How we designed our software to focus on quality attributes

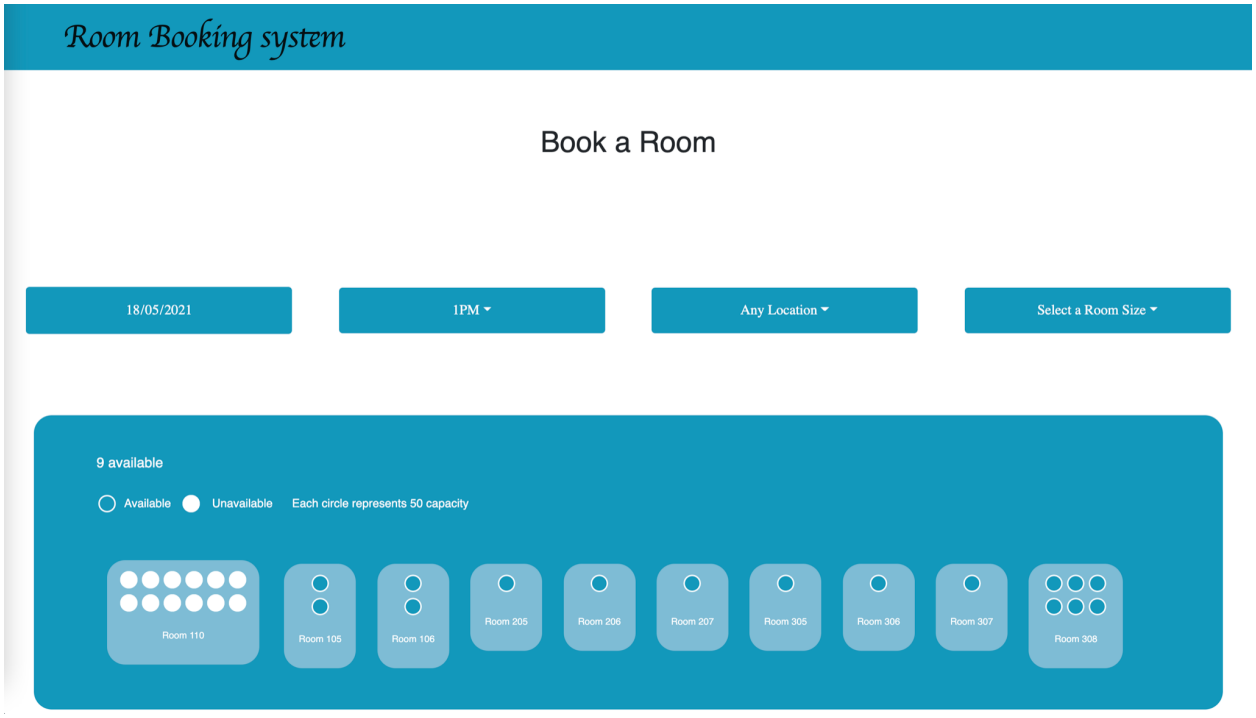
The quality attributes we are focusing on are

- Availability
  We first decided to add an authentication module but later we changed the plan to discard the authentication module because in order to use the functionalities offered by our software the user needs to authenticate themselves and for that the software should be deployed on the internal server of IITJ. In this case, users can authenticate themselves by connecting to IITJ LAN or Wifi by entering their LDAP ID and password. This also implies that only people with IITJ Mail ID can login to the website and book the room for an event. Professors, students, office staff are the only ones who can access the software.

  So to focus on the availability quality attribute we decided to discard the authentication module, this might lead to less security but anyone from anywhere can open the application and use it.

- Reliability
  We made the software reliable by as in case let's say a person has booked a room for a particular time and date slot then for the same time and date the room will be unavailable for the other user.

  Previously there have been problems where two users book the same room lets say 110 for any event by consulting different office staffs at a very short interval then one of them has to drop their consultancies. In this way the user can rely on our software that a booked room will be available for him at that time and there are no discrepancies among users. Here in the figure of our application we can see how a booked room is unavailable and shown by white circles.

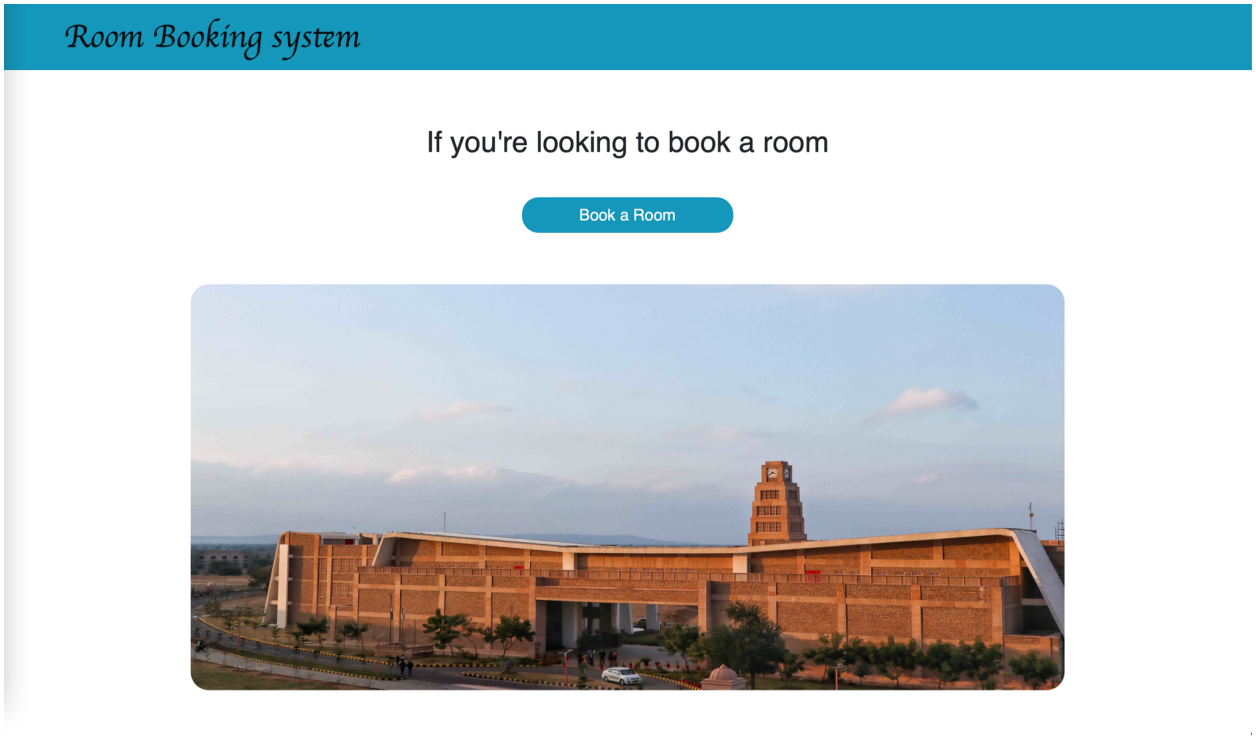


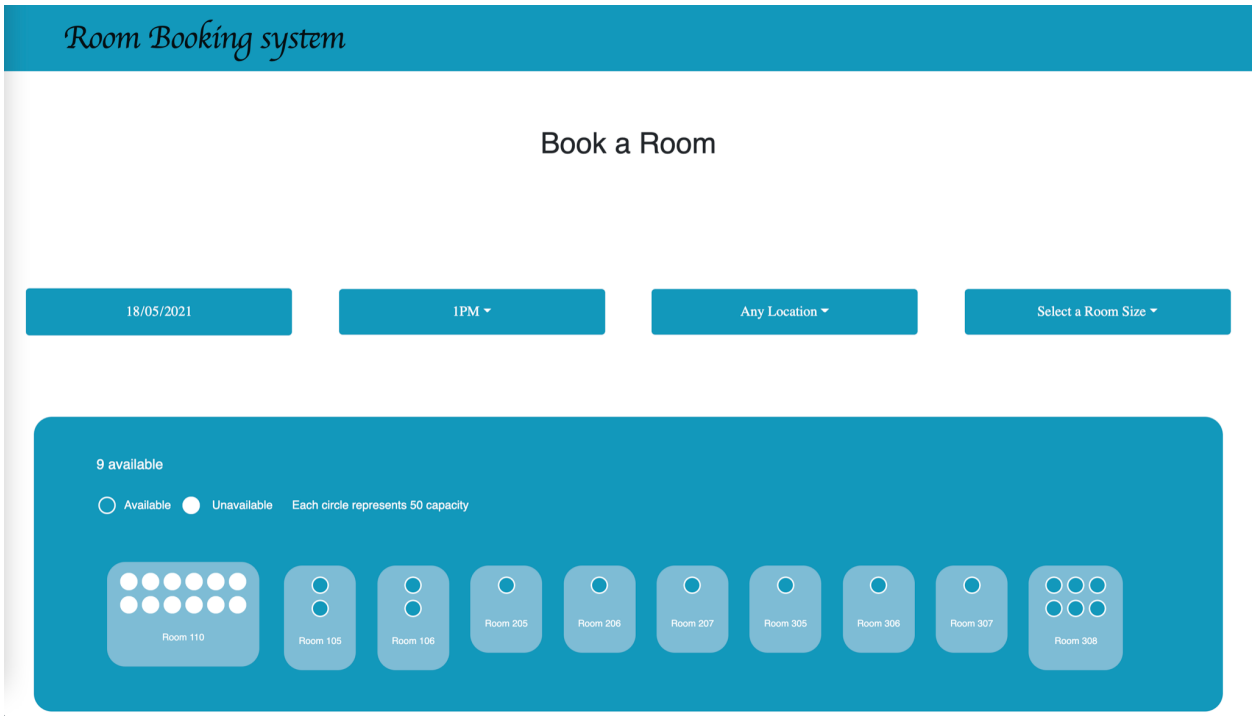
- Scalability
  Nowadays with the increasing number of users in every domain it is necessary for any web application to be scalable in order to meet a large number of user requirements and for this we also tried to scale our app. First of all we had chosen to develop our software in react app module and in javascript because the react app is very much easily scalable and javascript is the widely used language over the globe for all kinds of works.

  The libraries like React dome, react hook, date-fns etc, helps to easily scale any react app to a large code base. React APIs are good for meeting large codebase requirements and in order to increase the no. of users and make the app scalable the code base should be large. All the big software companies like Netflix, Instagram etc use React apps and we all are familiar with how many users they can serve.

- Aesthetics
  We had focused on this quality attribute a lot by designing our software as simple and understandable as possible. We have kept less number of buttons and icons so that any new user can use the software comfortably. The color contrast and icon placements are compatible with the user and UI design is very trivial.

Our software is so user friendly and of easy aesthetics that even a very less knowledgeable person lets say any staff of IITJ can book a room simply. Here is the figure of the frontend of our application which shows how simple aesthetics we have used.





- Usability
  Our software is compatible with all the versions of the web browsers and all the operating systems as we have used a simple react module to develop our app and is nowadays supported by every web browser.

  Our software is compatible with all the mobile phone browsers also as react app is supported by most of the mobile web browsers. We have put the browsers in the package.json file under the development column where other browsers can also be added if needed.

Modularisation

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. We had modularised all the frontend and all the backend components separately and inside each of these also we had followed this modular approach like in backend the data and models are in different modules and in the frontend also source and public folder are separated. The pages, assets and styles are also modularised under the source folder in the frontend.

Why we modularised the components:

- Smaller components are easier to maintain
- Program can be divided based on functional aspects
- Desired abstraction can be brought in the program
- Components with high cohesion can be reused again
- Concurrent execution can be made possible
- Desired from security aspect

Coupling

Coupling is a measure that defines the level of inter-dependability among modules of a program. It tells at what level the modules interfere and interact with each other. The lower the coupling, the better the program.
In our case of the project the coupling is quite high as we used react to develop our webapp. The different components exported and imported in each outer area are used by each other so they are highly interdependent, failure in one component may lead to malfunctioning of the entire frontend application.

- **Data coupling-** Data coupling is when two modules interact with each other by means of passing data, so in our react application most of the modules are highly data coupled.

- **Content coupling-** When a module can directly access or modify or refer to the content of another module, it is called content level coupling. In our case we used this content coupling while modifying the json file, where the availability of the room is changed to false for a certain duration whenever any user books that room for that time slot.

Cohesion

Cohesion is a measure that defines the degree of intra-dependability within elements of a module. The greater the cohesion, the better is the program design.

- In our project we have used the cohesion concept in our modules. We used the sequential cohesion concept where we feed the output of one element to the input of the other elements by the use of react functions.
- We had also used the procedural cohesion in the components of frontend and backend where elements of the module are grouped together, which are executed sequentially in order to perform a task.
- We had used as many functions we can use to execute the elements of any component inorder to ensure the functional cohesion.

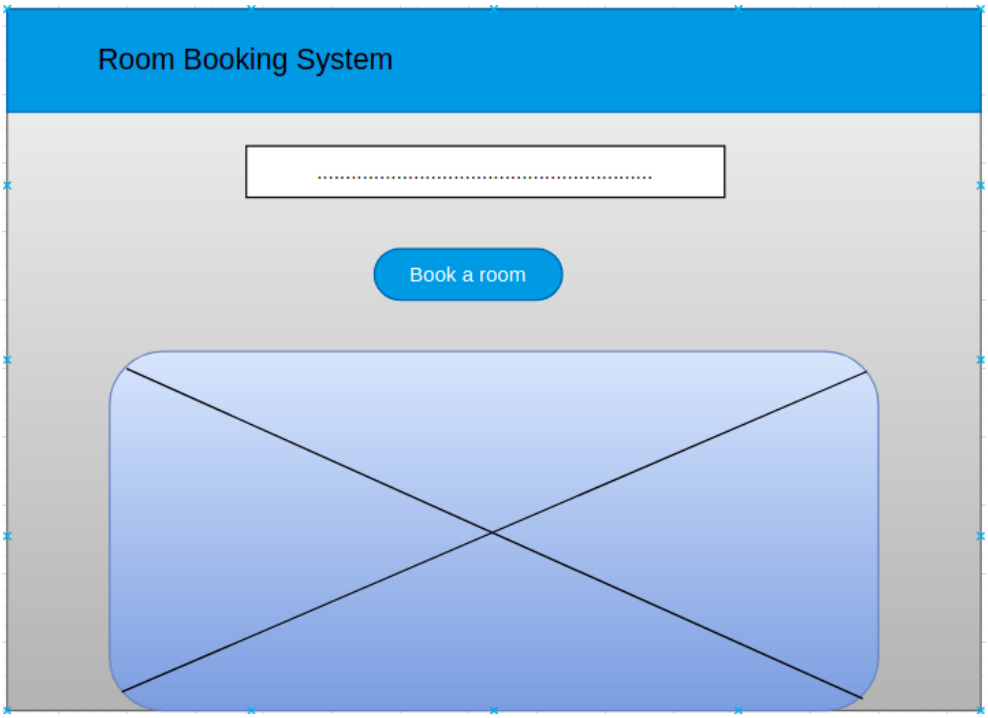
WIREFRAME DESIGN

What is a wireframe

A wireframe is a two dimensional skeletal outline of the web application which also represents the blueprint of the UI/UX. Wireframes provide a clear overview of the page structure, layout, information architecture, user flow, functionality, and intended behaviors. A wireframe usually represents the initial product concept, styling, color, and graphics are kept to a minimum. Wireframes can be drawn by hand or created digitally, depending on how much detail is required.
The process of wireframing allows all the stakeholders to agree on where the information will be placed before the developers build the interface out with the code.

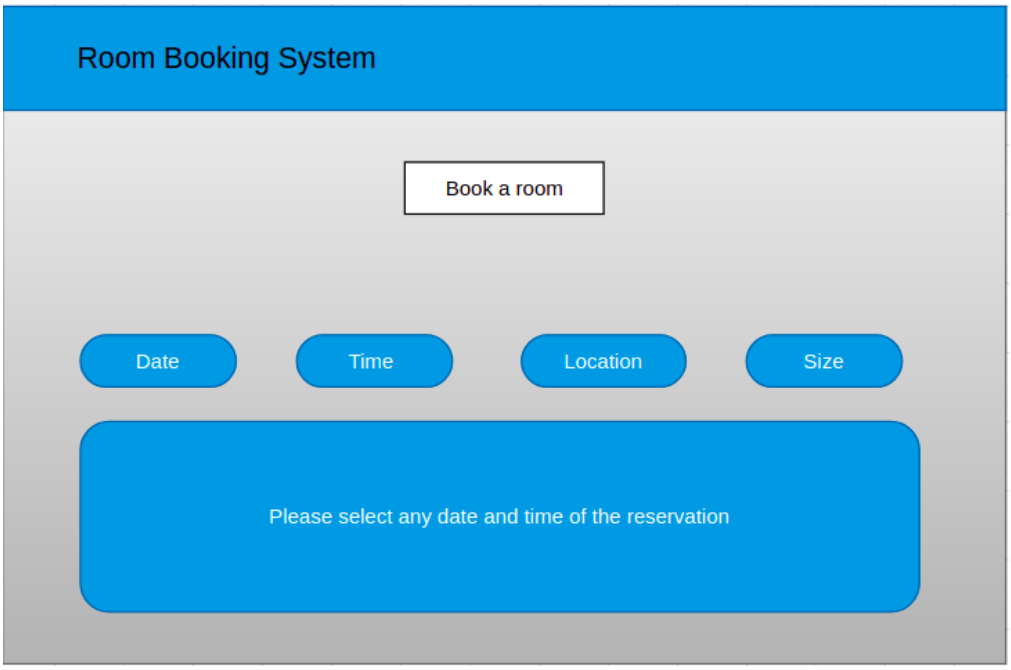
Wireframe design of the developed webapp

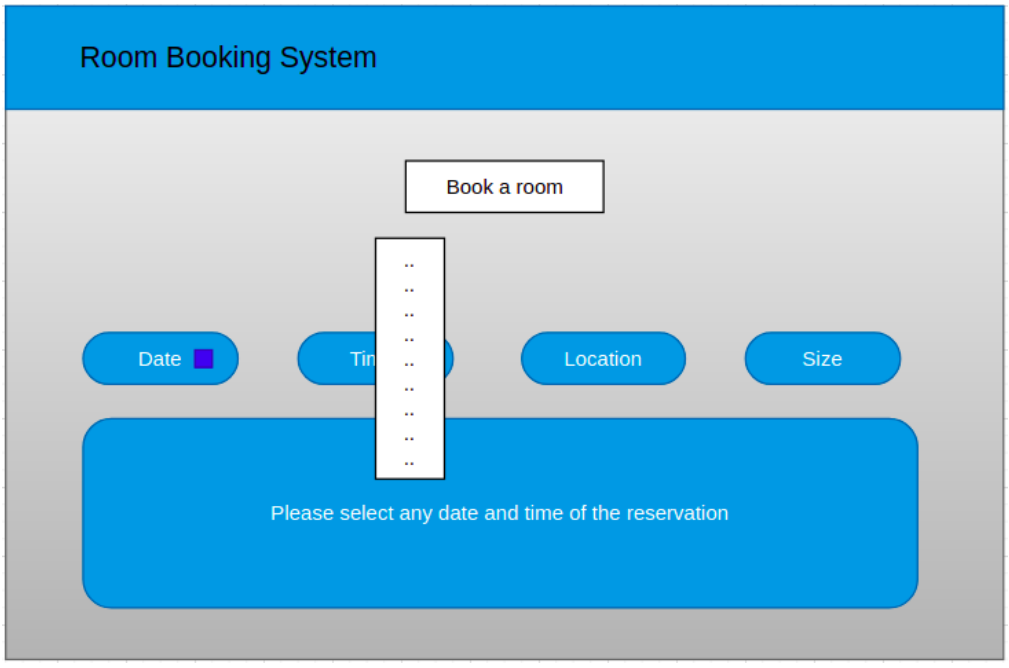
i) The Home page: Consists of a header nav bar, an icon to click to book a room and an image of IITJ.

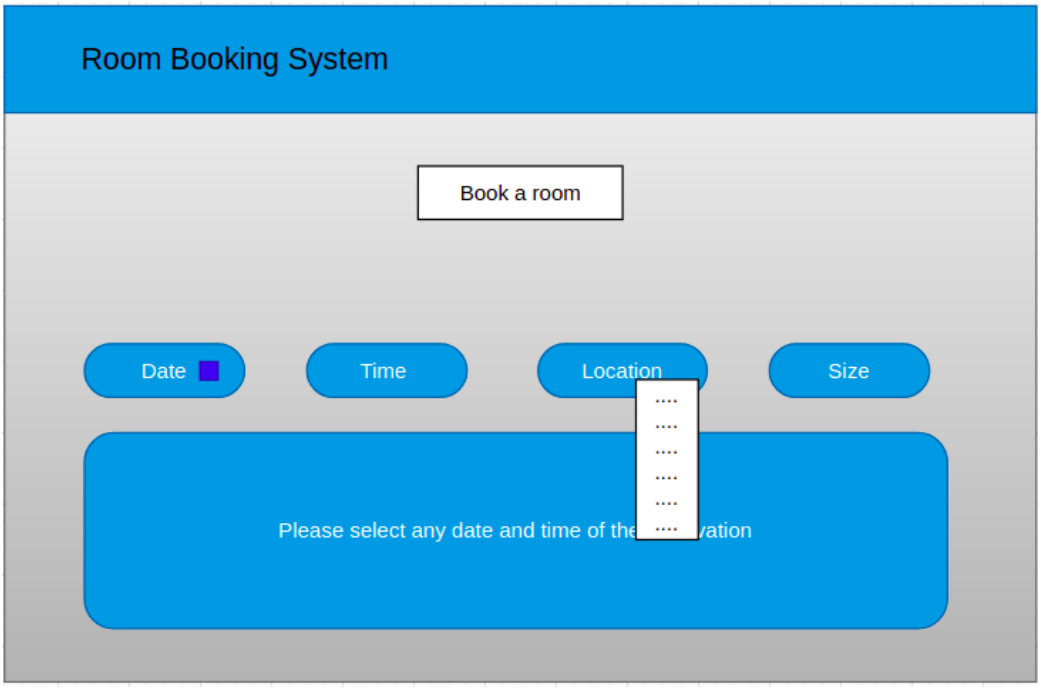


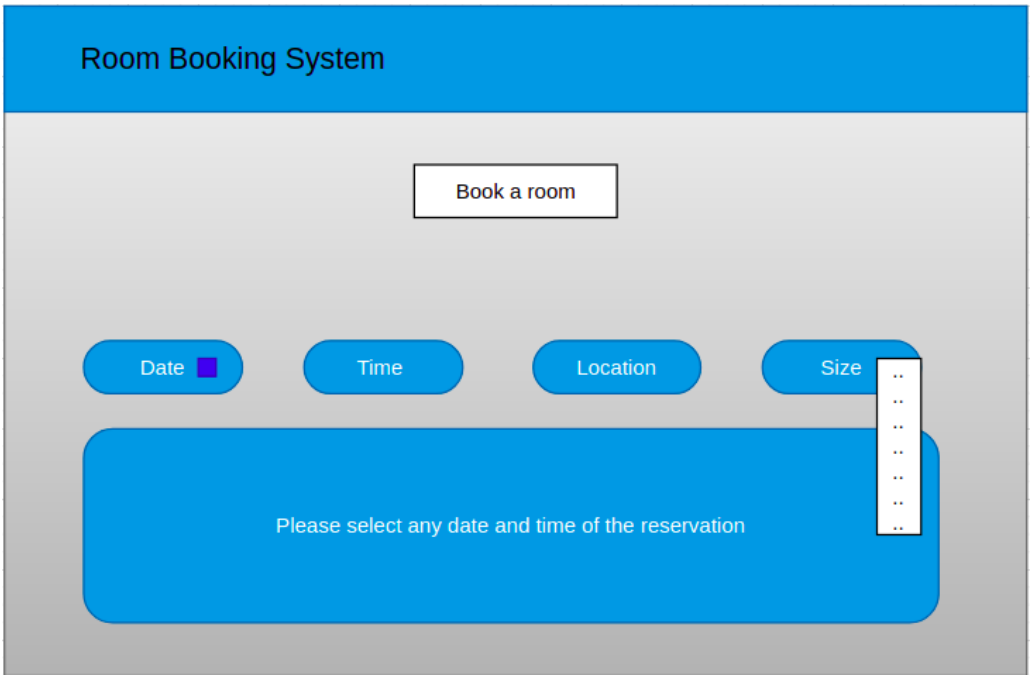
ii) Filtering a room to book on the basis of date, time, location and capacity: Consists of four buttons to select date, time, location and capacity of the rooms and the filtered rooms will be shown accordingly, in the blue space shown below.

Date can be typed and there is a drop down for the time, location and size column.


Room Booking System
Book a room
Date
Time
Location
Size
Please select any date and time of the reservation

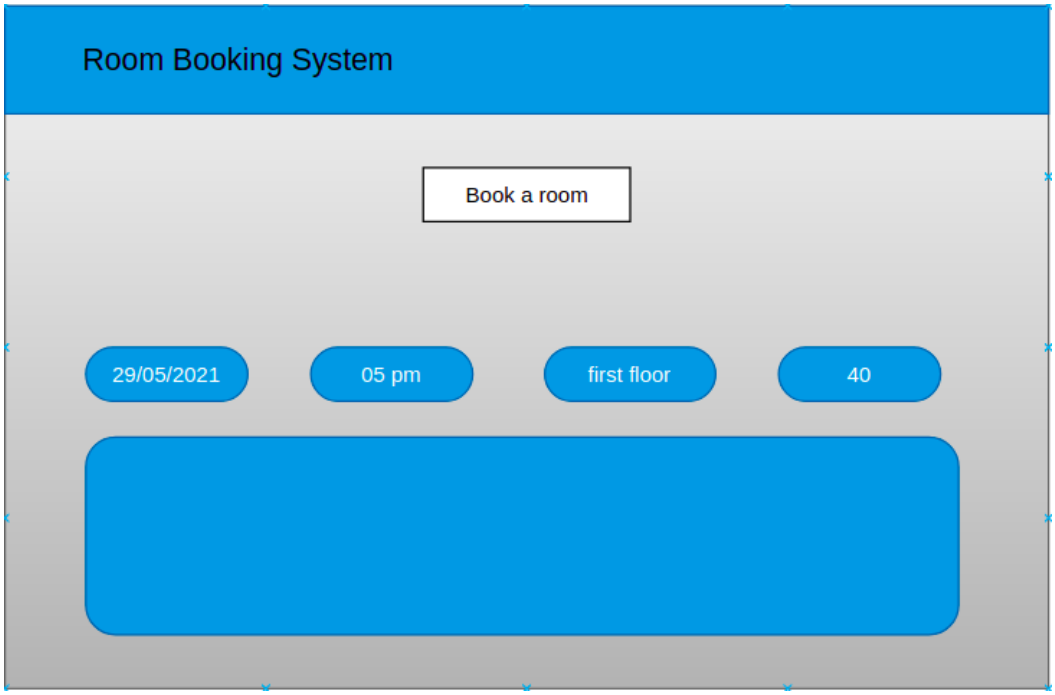

Room Booking System
Book a room
Date
Location
Size
Please select any date and time of the reservation

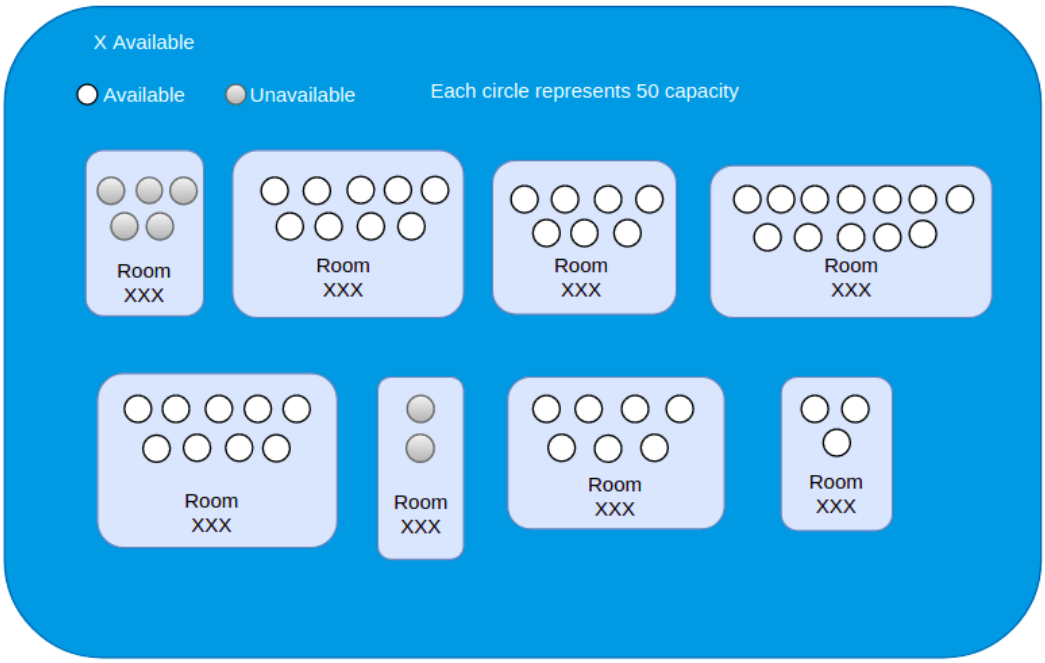




iii) Showing filtered rooms on the basis of the selected filter: Available rooms will have white circles and booked rooms will have grey circles. The circle represents the capacity of the room(1circle = 50 capacity). If no rooms are available for that particular filter then a message with “No Rooms Available” will be shown.

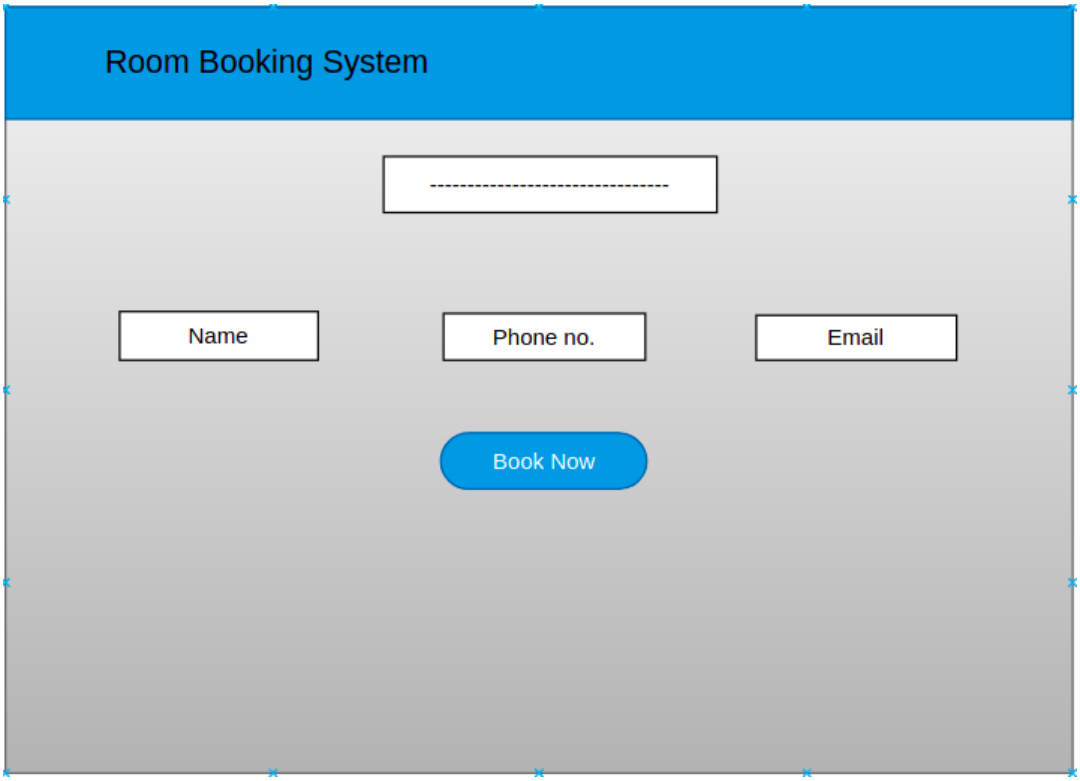
Availability of rooms:



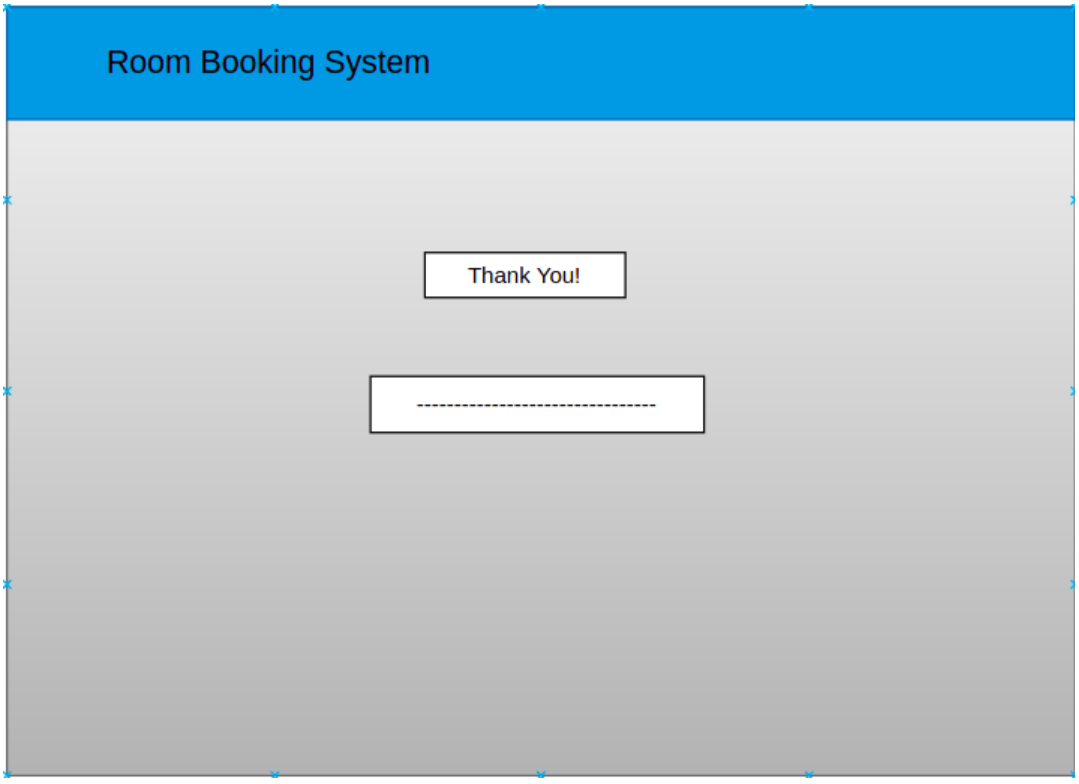


Non Availability of rooms:

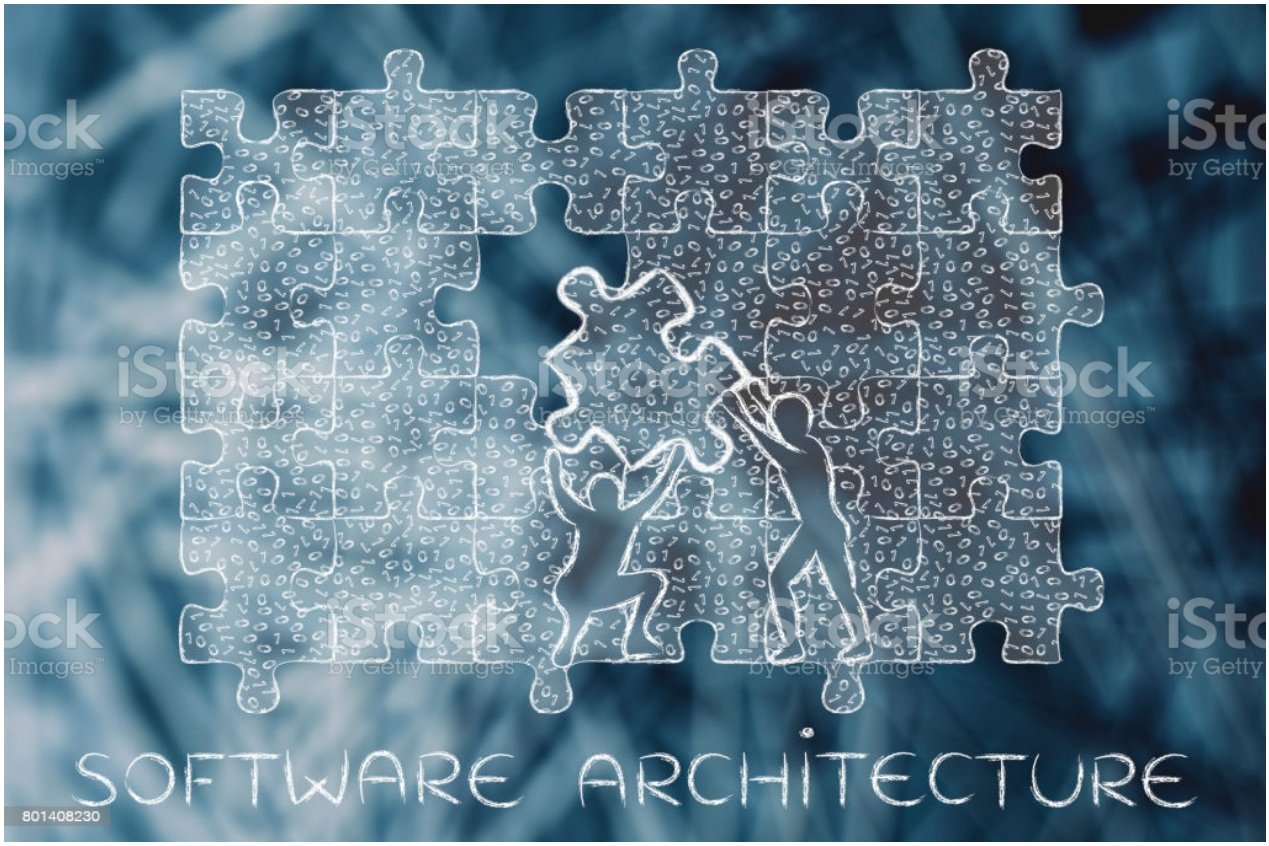
iv) User details for booking: There are three blocks requiring user information for a room to be booked, which are name, email and phone number. Then a book now button is there which will confirm the booking.



v) Booking Confirmation page:

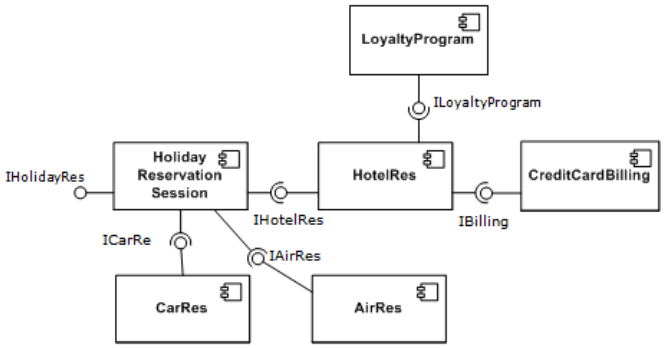


SOFTWARE ARCHITECTURE

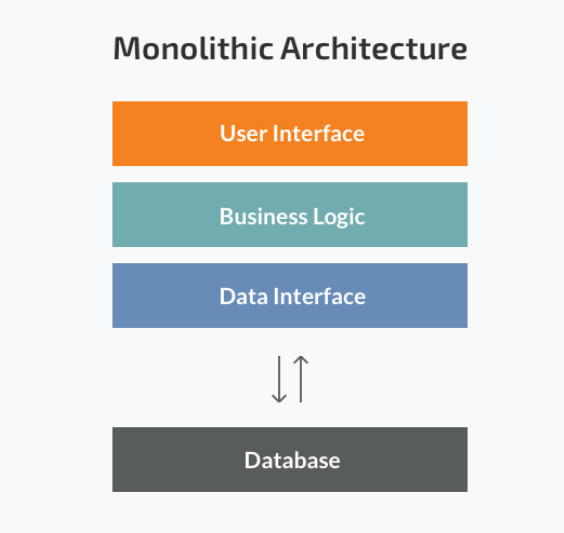


An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context.
There are mainly 4 types of architectural pattern followed by the team of any software developer:

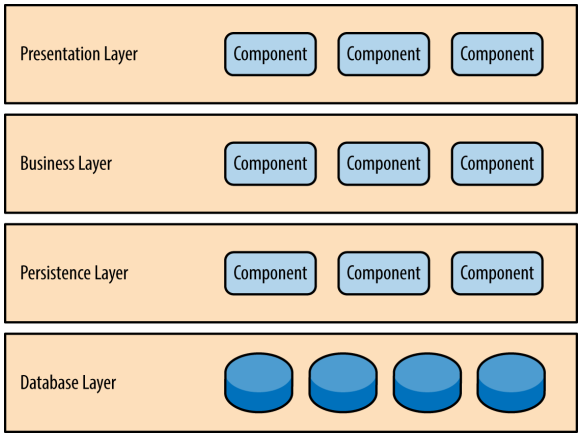
- Component based - Component based architecture focuses on the decomposition of the design into individual functional or logical components that represent well-defined communication interfaces containing methods, events, and properties.The primary objective of component based architecture is to ensure component reusability.



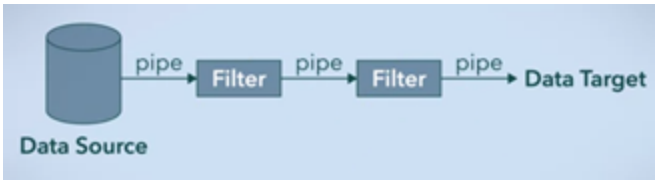
- Monolithic- Monolith means composed all in one piece. The Monolithic application describes a single-tiered software application in which different components combined into a single program from a single platform.



- Layered- This pattern can be used to structure programs that can be decomposed into groups of subtasks, each of which is at a particular level of abstraction. Each layer provides services to the next higher layer.



- Pipe and Filter- Pipe and Filter is another architectural pattern, which has independent entities called filters(components) which perform transformations on data and process the input they receive, and pipes, which serve as connectors for the stream of data being transformed, each connected to the next component in the pipeline.

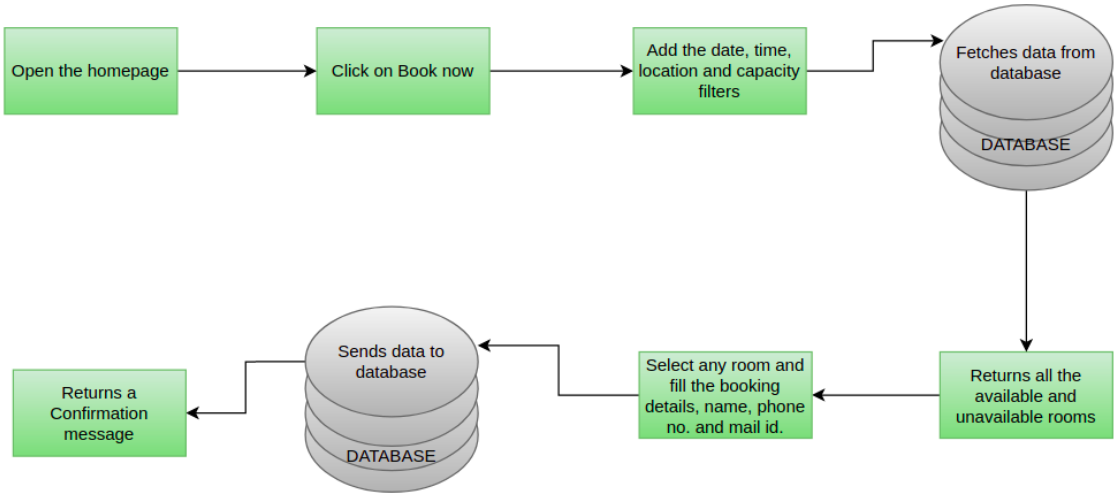


NOTE: We had used the monolithic Architectural pattern in our software design.

Why had we used this monolithic Architectural pattern:

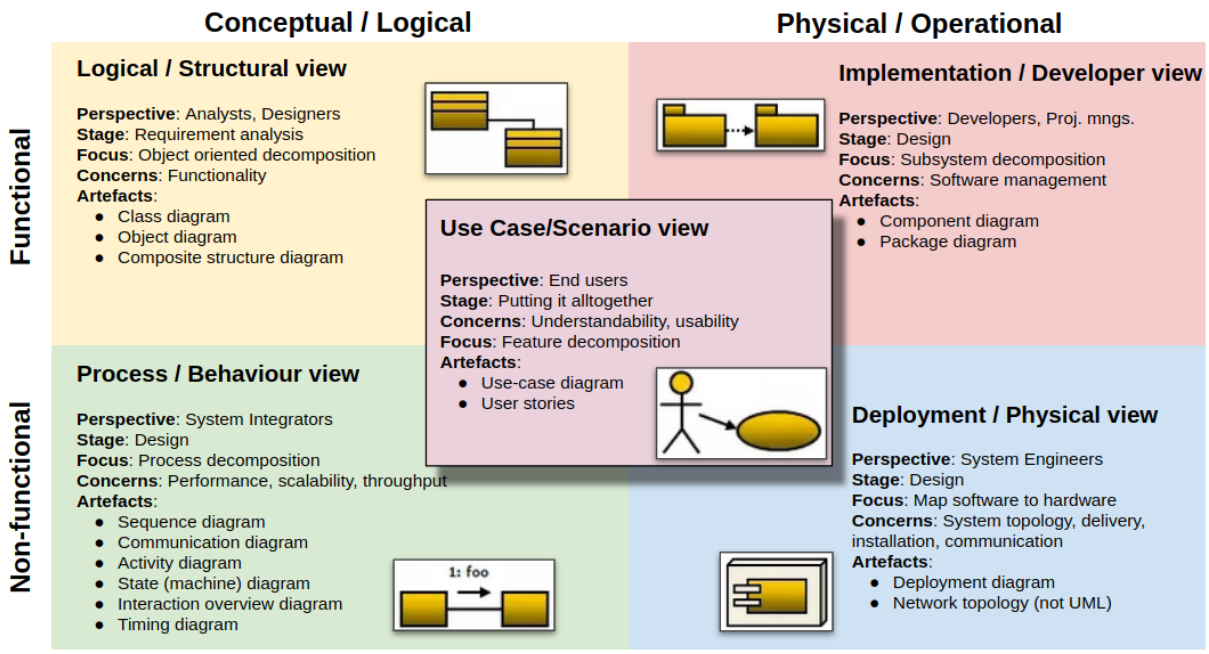
- Simple to develop as all the components are structured in one piece.
- Simple to test as the software application is single tier architecture, we can test all the components in one go and also we can apply end to end testing by simply launching the application.
- Simple to deploy as we just have to copy the packaged application to the cloud server.
- Simple to scale horizontally by running multiple copies behind a load balancer.

The complete Structural pipeline of our web application



4+1 architectural pattern view of our software

What is 4+1 view



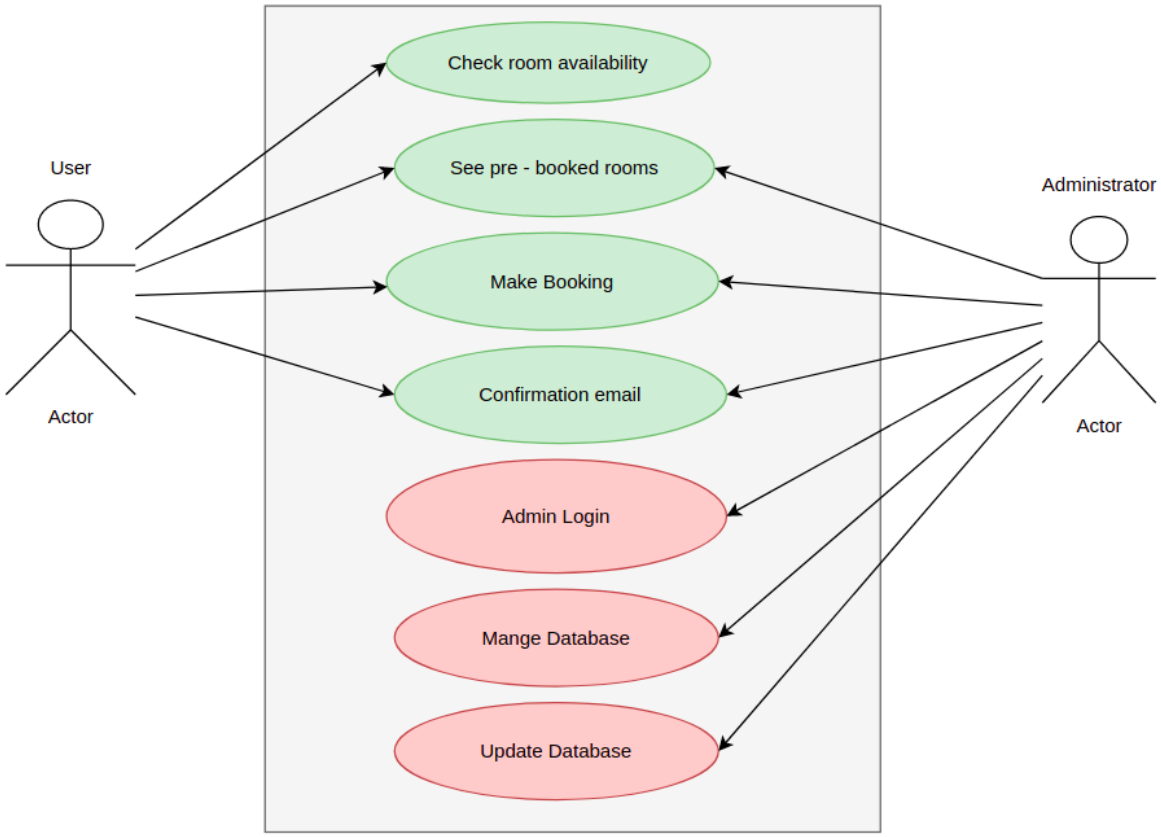
4+1 is a view model based on the use of multiple, concurrent views which is used for "describing the architecture of software systems. According to the viewpoint of different stakeholders, such as end-users, developers, system engineers, and project managers the views are used to describe the system.

The four views of the model are

- logical view
- Development view
- Process view
- physical view

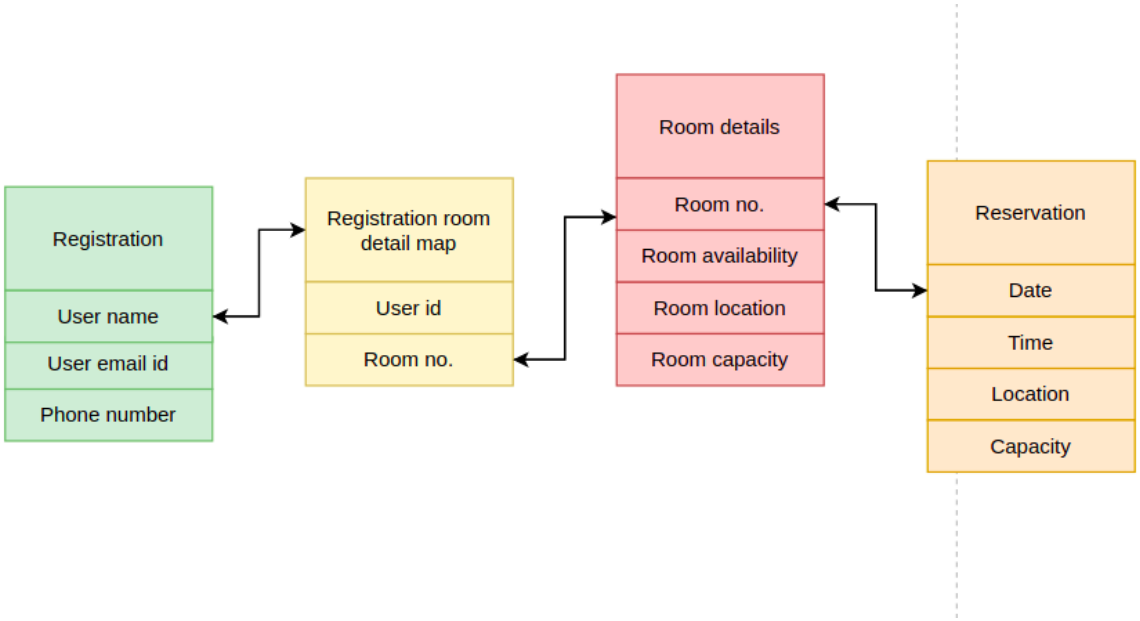
Scenarios

Description: Shows the design is complete by performing validation and illustration and different tasks which users can perform.
Viewer/Stakeholder: All the views and their evaluators.
Consider: System consistency and validity
UML diagram: use case diagram



Logical View

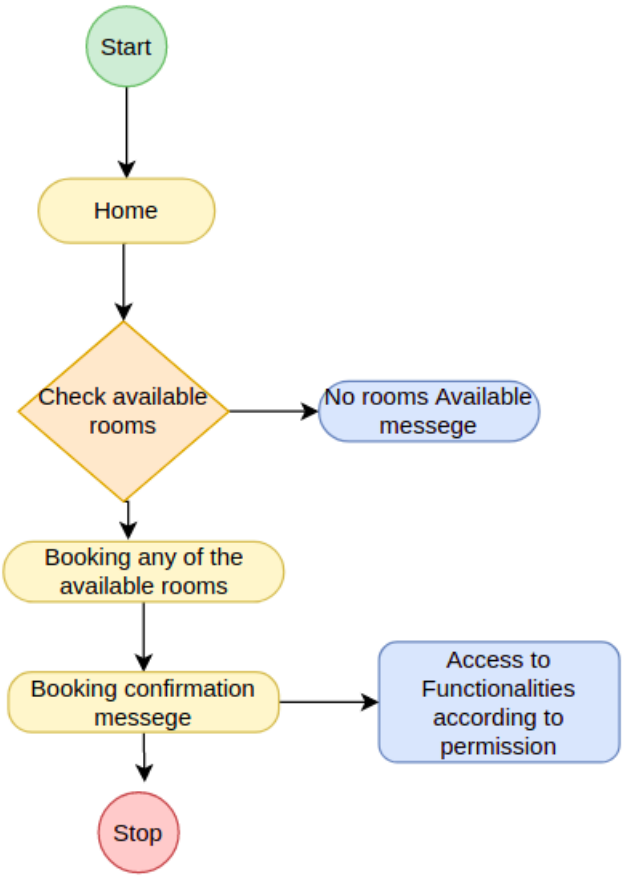
Description: Shows components(objects) as well as their description.
Viewer/Stakeholder: End-user, who are passengers in this case, Analyst and Designer who will design the whole management system.
Consider: Functional requirements
UML Diagram: Class,State,Object,Sequence,Communication Diagram.



The above diagram is a class communication diagram which shows the links between different class attributes used in our software project.
The uppermost rectangle represents the class name and the rest of the Rectangles represents the attributes/parameters of that class. First class is of the registration module where the attributes are user name, email id and phone number.The user name is the key which connects the registration class to the registration room detail map class where the user id is the key which maps the room details via room number after which a room is reserved and linked.

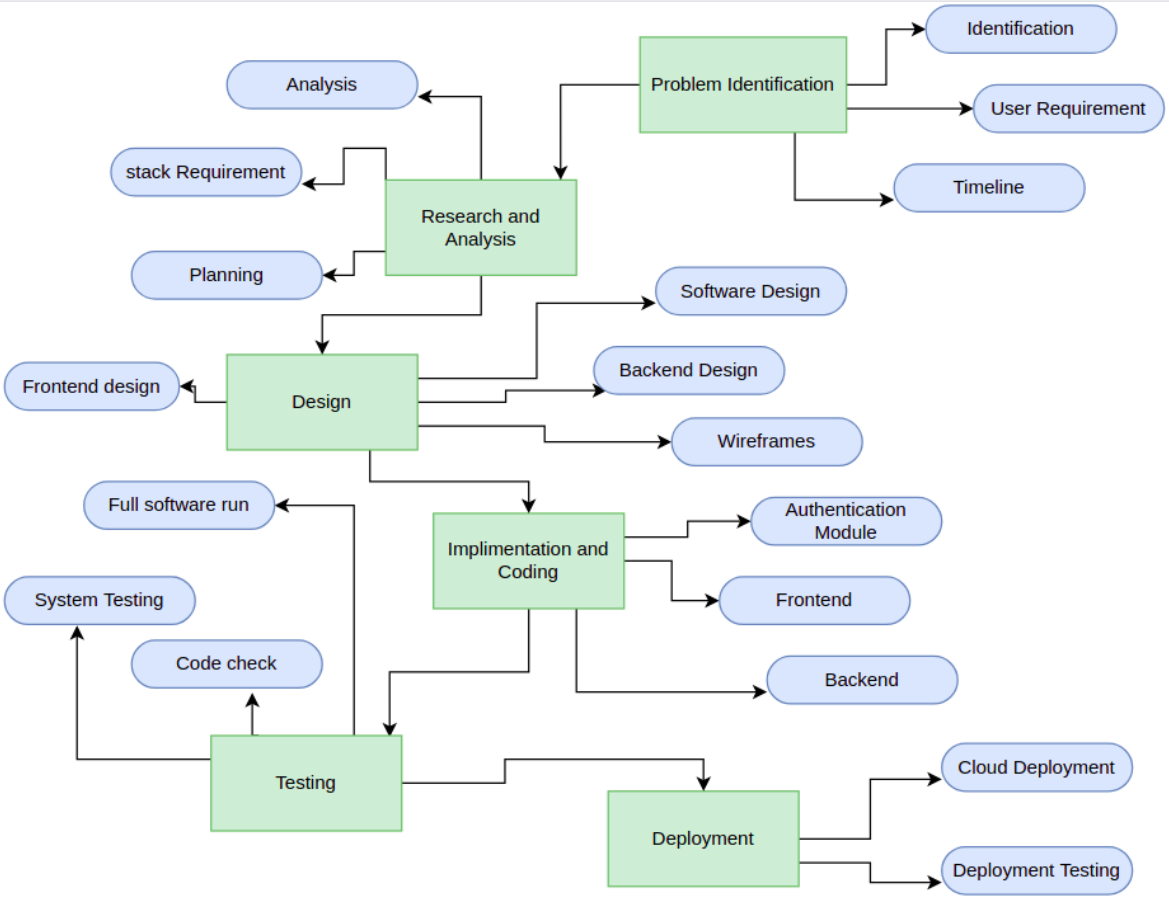
Process View

Description: Shows the processes / Workflow rules of system and how those processes communicate, focuses on dynamic view of the system.
Viewer/Stakeholder: Integrators who integrate all the processes and the Developers of the system.
Consider: Non-Functional requirements
UML Diagram: Activity Diagram



The parallelograms represent the simplified process and the arrows represent the unidirectional or bidirectional instructions/messages transferred between the processes.

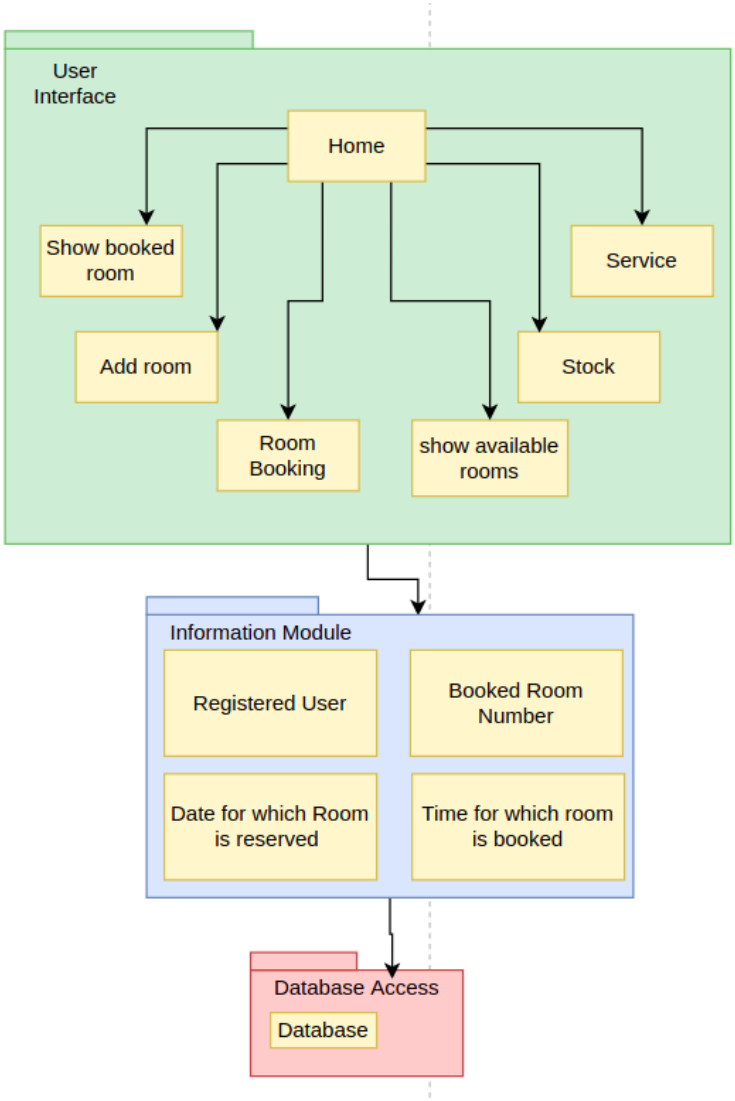
The user has to first open the home page. The user is now logged into the system and now he has to click on book a room option. The software then checks for all the available rooms and shows it to the user after which the user can choose a room and book that, and get a booking confirmation message, after which he can access different functionalities of the software according to the user's permission.



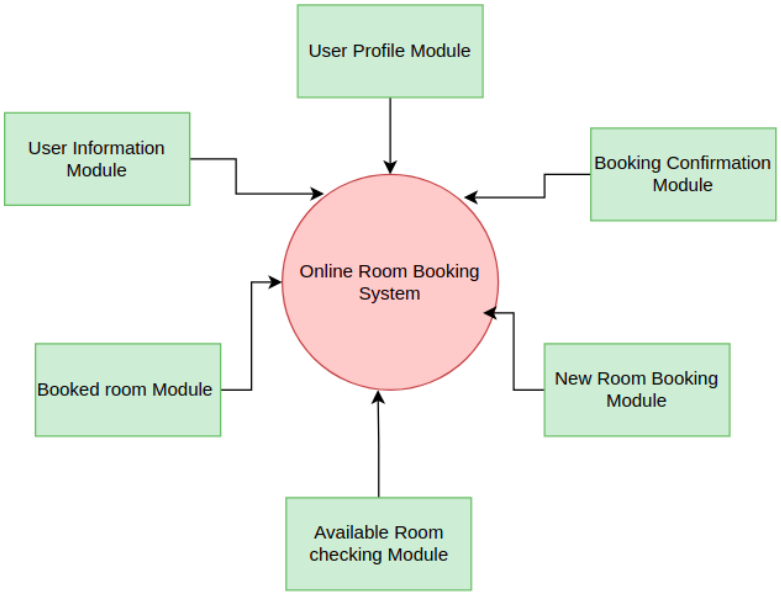
The above diagram shows the process view of the workflow, how different works are connected and proceeded and what are the links through different components and modules at each stage of the workflow.

Development View

Description: Gives building block views of the system and describes static organization of the system modules.
Viewer/Stakeholder: Programmer and Software Project Managers
Consider: Software Module organization (Software management reuse, constraint of tools)
UML Diagram: Component and Package Diagram.



The above shown Diagram is a component Diagram in which the different Layers of components and their related modules are shown. The first component is the user interface component which contains all the accessible modules which a user can access and change. The second component contains all the information fed by the user in order to book a room which is further fed to the database at the backend server where all the information is stored safely and the user cannot access or update it.
This can be accessed and updated only by the administrator.

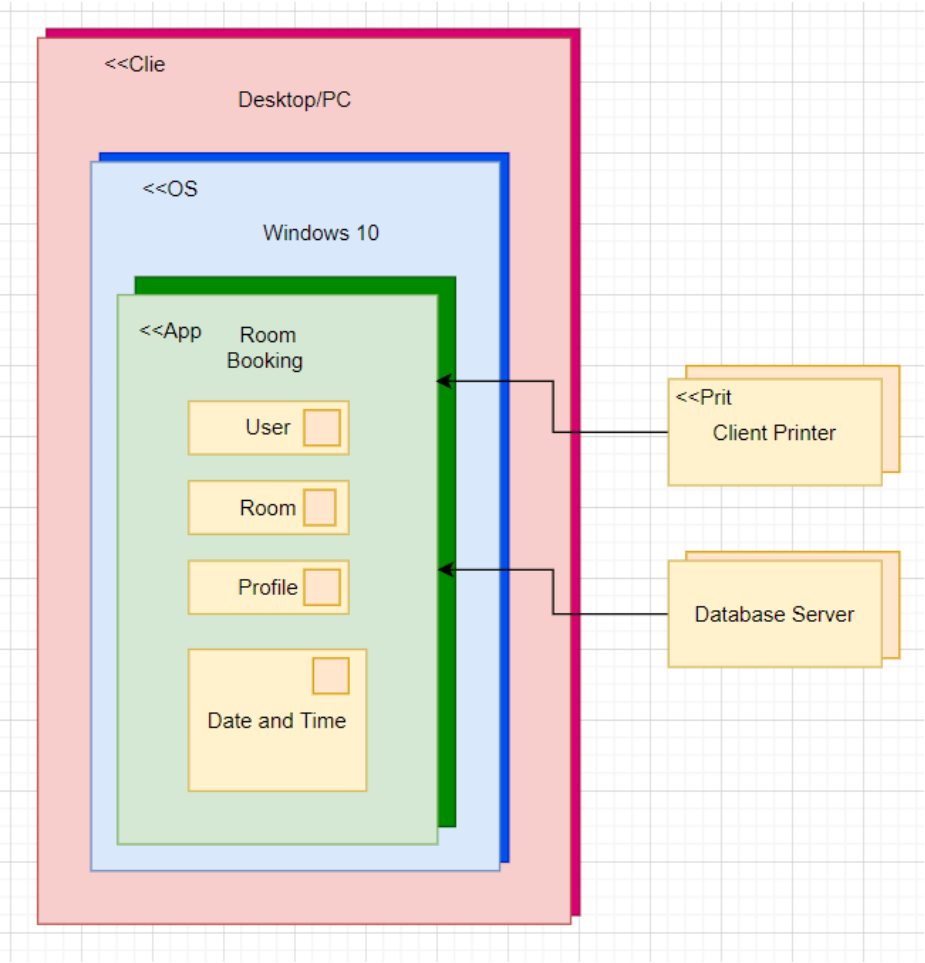


The above diagrams show the different components/modules which a developer will be developing in order to make the software to fulfill the requirements of the user.
In our case the software contains the login module after which the user will login to the user profile where he will be able to search for the available rooms and then book among one of the available rooms.

Physical View

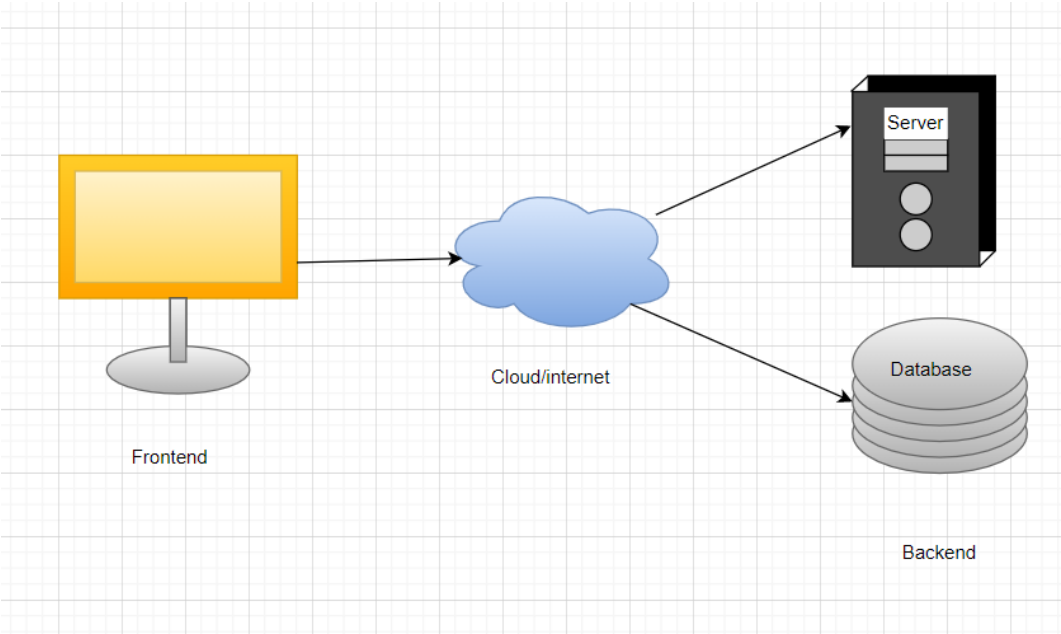
Description: Shows the installation, configuration and deployment of software application
Viewer/Stakeholder: System engineer, operators, system administrators and system installers.
Consider: Non-functional requirements regarding underlying hardware.
UML Diagram: Deployment Diagram.

We expect that several different physical configurations will be used: some for development and testing, others for the deployment of the system for various sites or for different customers. The mapping of the software to the nodes therefore needs to be highly flexible and have a minimal impact on the source code itself.



Here is the Deployment View which shows the run time processing of nodes and components which live in them in a hierarchical way.The outermost layer is the Desktop/Pc which serves as the frontend viewer, then comes the operating system which connects the frontend to the Information module which ultimately links to the backend Database server.

At each hierarchical level there are different types of computer with different storage, capacity and supporting different executables. There are backup nodes at each level which will backup all the data incase of any misfortune.



The software's frontend is developed and then it is uploaded on any of the internet cloud over which the information is passed and stored at the backend in the database.
A server is used to serve, send and receive the information from the user.

Thanking You
Course instructor - DR. Sumit Kalra