Software Architecture

Group-12
Govinda Rohith Y (CS21BTECH11062)
R Bhargava Ram (CS21BTECH11052)
P Pranav (CS21BTECH11063)
Teja (CS21BTECH11059)

Contents

1 Introduction:

1.1 Purpose:

The purpose of this document is to provide a comprehensive overview of the architecture of the Hostel Room Management System, serving as a reference for developers, architects, and stakeholders involved in the project. It aims to elucidate the system's design, functionalities, and interactions to ensure effective development, deployment, and maintenance.

1.2 Scope:

The architecture documentation covers the high-level structure of the Hostel Room Management System, including its key components, user roles, interactions, and functionalities. It delineates the responsibilities and interactions of users, such as students, Hostel Representatives (HR), and Hostel Office (HO), within the system.

We describe two possible architectures for Hostel Room Management System. Thereafter, we will compare the advantages of one over another and choose an architecture that describes our model appropriately. We will also provide the logic behind our selection.

1.3 Definitions and Acronyms:

1. **HRMS:** Hostel Room Management System.
2. **HR:** Hostel Representative (per Block).
3. **HO:** Hostel Office (Admin).

2 Overview:

2.1 System Overview:

The Hostel Room Management System's primary functionality is to allocate rooms for students based on availability and preferences. We have further extended functionalities to:

1. Students can request for room exchange, which can be accepted or rejected by administrators.
2. The system also facilitates announcements to inform students about important updates or events.
3. For check-ins and check-outs, students may need approval from hostel staff, which the system can manage.
4. Additionally, the system provides a platform for students to view working status of facilities, maintenance info.
5. Students can also lodge complaints about room/pod/floor facilities or services, which can be addressed by the hostel management.

Overall, the system enhances the hostel experience by efficiently managing room allocations, facilitating room exchange requests, keeping students informed, and ensuring smooth check-in/check-out processes and complaint resolution.

2.2 System Context:

The stakeholders of the HRMS includes students, HR and HO. The external entities and systems that interact with the software system are house keeping staff/ Security/ Technician's for facility servicing (facilitated by HO) and academic office of IITH for student personal details exchange.

2.3 Stakeholders:

The stakeholders for your hostel room management system are:

- **Students:** Primary users residing in the hostel, interact with the system for room management tasks, announcements, complaints, and facility status.
- **HR:** Use the system to manage hostel operations, resolve complaints, and make announcements.
- **Hostel Office:** Administers the system, manages facilities, resolves escalated complaints, and communicates with house keeping staff/ Security/ Technician's for facility servicing.

These stakeholders are crucial for the successful implementation and operation of HRMS. Understanding their needs and expectations will help ensure that the system meets their requirements effectively.

2.4 Context Diagram:

The users/stakeholders and their interactions with the system can be visually depicted in the context diagram below.

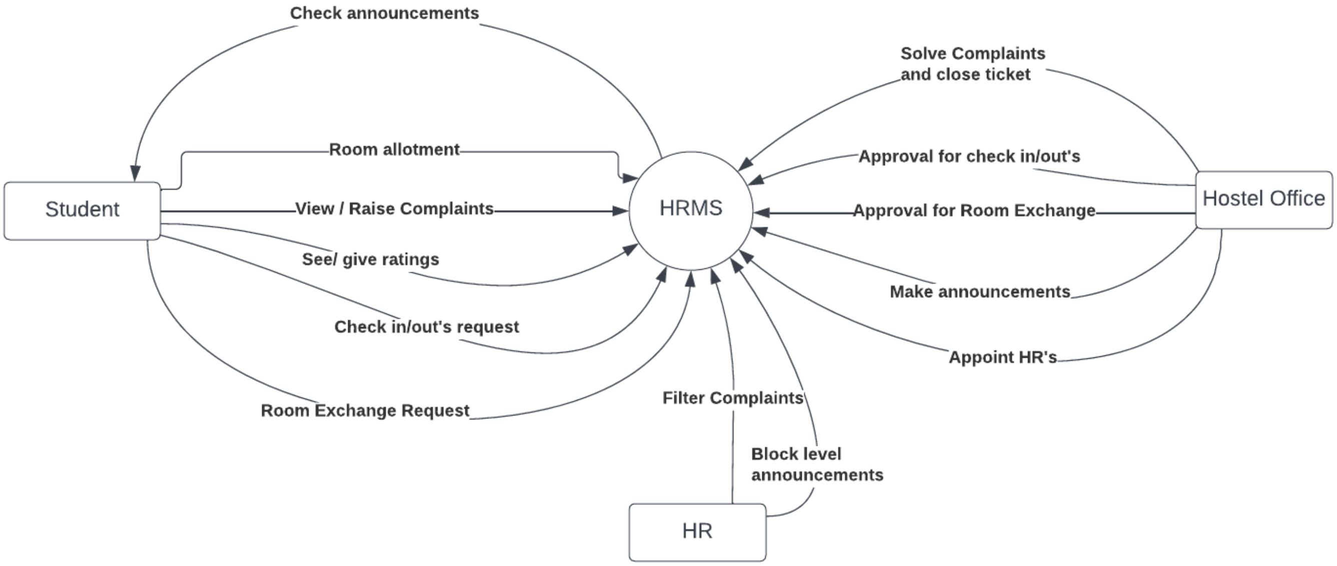


Figure 1: Context Diagram

3 Architecture Design:

3.1 Architecture 1: The Repository Model

Following table describes all the components in the architecture:

S.No	Component	Component Type	Description
1	Data Repository	Database	This module contains information about users, hostel block, announcements, complaints and booking related requests etc.
2	Authentication	Processing(Database access and modification)	This module is responsible for managing user authentication and for updating password in-case of forgetting.
3	Room Allocation	Processing(Computation and Database modification)	This module is responsible for allocating room on FCFS basis for a room.
4	Edit personal Details	Processing(Database modification)	This module updates details in the database accordingly.
5	View pod details and status of facilities	Processing(Database access)	This module displays student personal details, facilities details and their status at pod/floor/block level.
6	Give/View reviews	Processing(Database access and modification)	This module is responsible for giving and displaying reviews.
7	Room exchange request	Processing(Database modification)	This module creates a request for room exchange which is to be approved by HO.
8	Room exchange approval	Processing(Database access and modification)	This module is responsible for displaying, approving the request and updating the database after approval.
9	Check-In/out request	Processing(Database modification)	This module creates a request for check-in/out which is to be approved by HO.
10	Check-In/out approval	Processing(Database access and modification)	This module is responsible for displaying, approving the request and updating the database after approval.
11	Create and view status of complaints	Processing(Database access and modification)	This module is responsible for creating and displaying complaints.
12	Filter/Close Complaints	Processing(Database access and modification)	This module is responsible for filter and passing, closing/deleting solved complaints.
13	Create/Edit/Delete announcement for target user	Processing(Database access and modification)	This module creates, updates and deletes announcements for target users and updates it in the database accordingly.
14	View Announcements	Processing(Database access)	This module displays announcements which the user receives.

Following table describes all the connectors in the architecture:

S.No	Connector	Connector Type	Description
1	Read Only	Database access	Represents reading of data from data repository.
2	Write Only	Database access,modification	Represents writing of data to the data repository.
3	Read/Write	Database access,modification	Read as well as write by modules to the data repository.

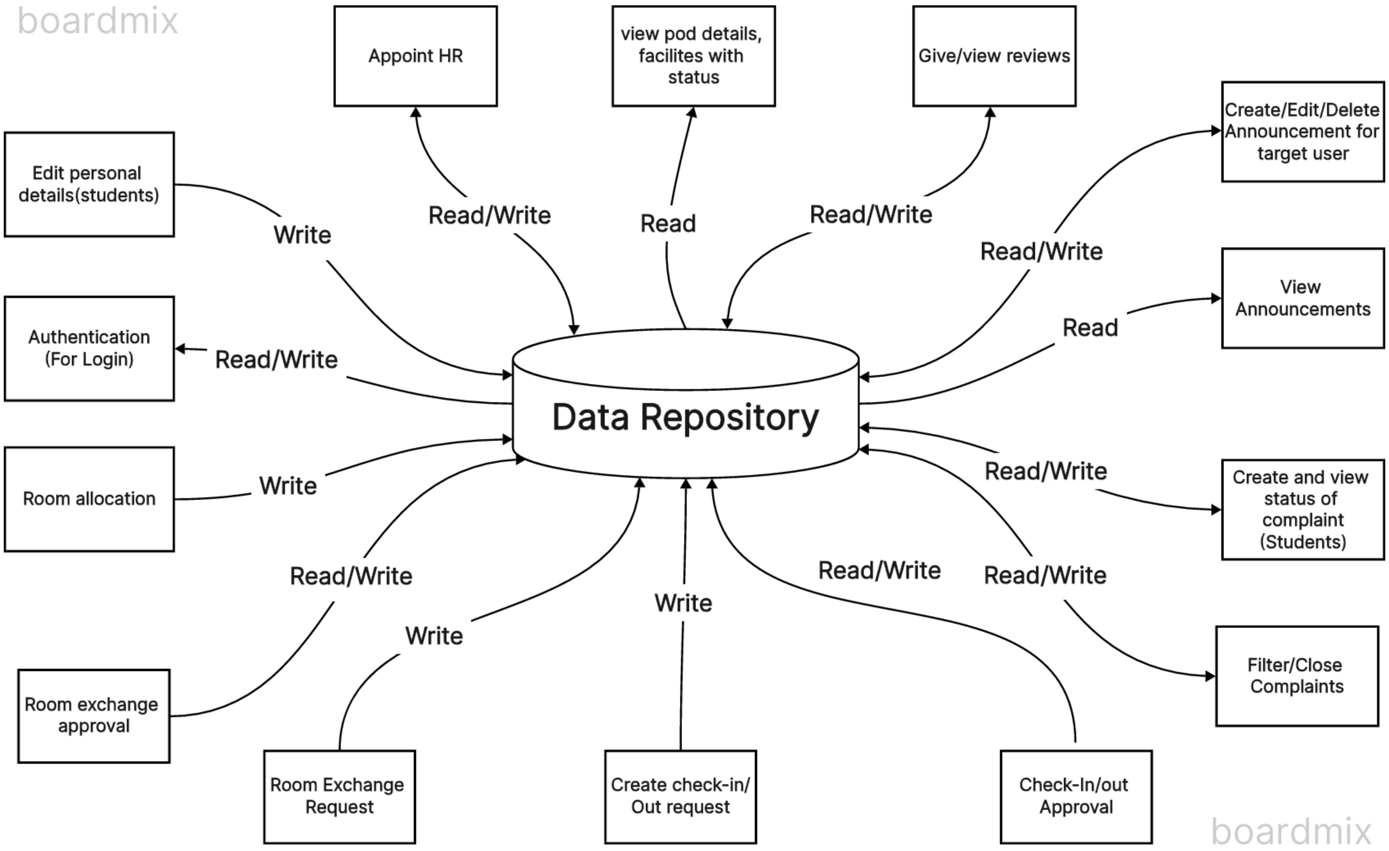
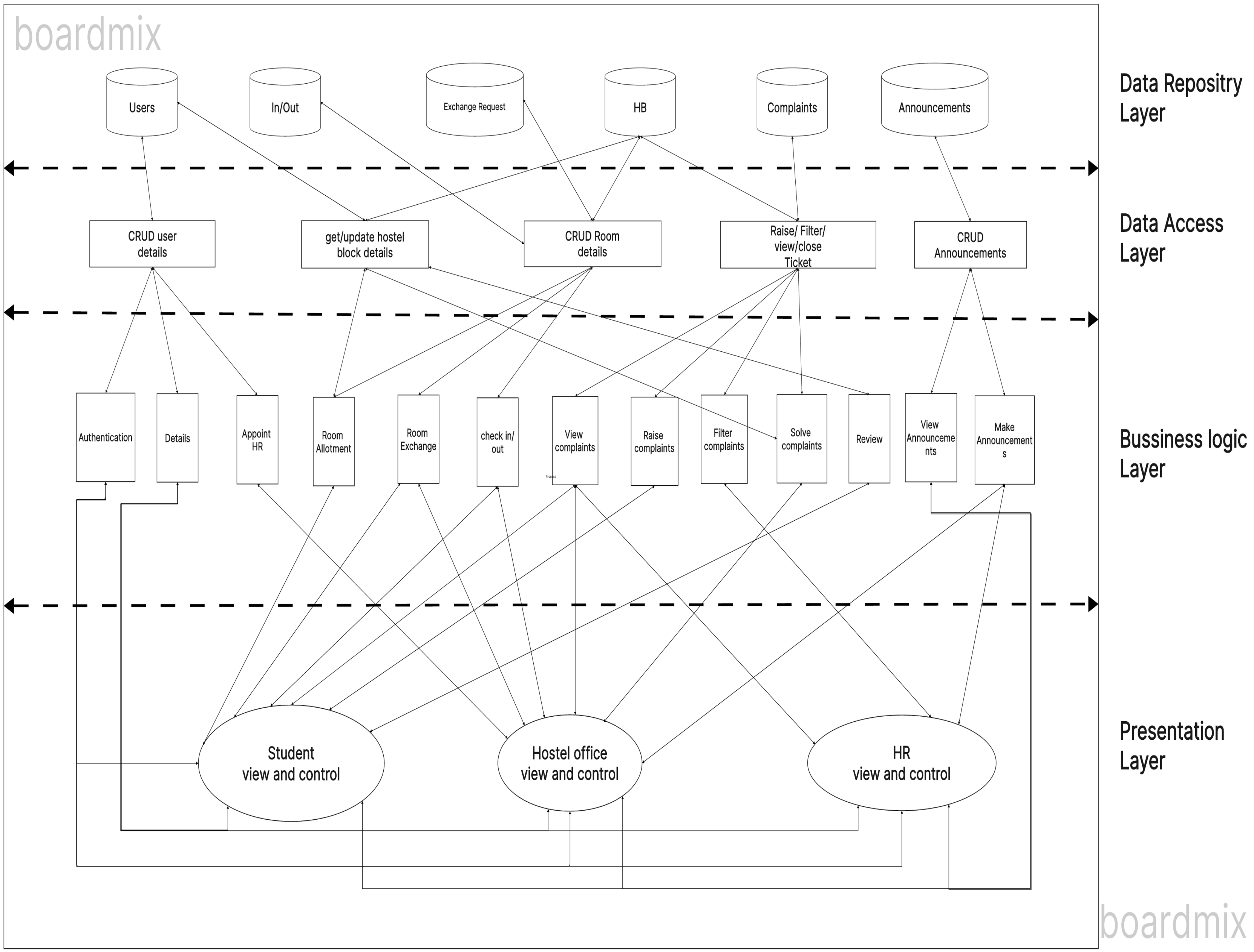


Figure 2: Repository Model

Description:

The repository model is based around the data repository which contains all the Authentication, Room allotment and exchange requests, check-in/out approvals, raise tickets for complaints and other data that needs persistent storage. This data repository is the central point of this architecture and all the other modules are developed around it.

3.2 Architecture 2: The Access Layer Model (4 layered)


Data Repositry Layer
Data Access Layer
Bussiness logic Layer
Presentation Layer
Users
In/Out
Exchange Request
HB
Complaints
Announcements
CRUD user details
get/update hostel block details
CRUD Room details
Raise/ Filter/ view/close Ticket
CRUD Announcements
Authentication
Details
Appoint HR
Room Allotment
Room Exchange
check in/ out
View complaints
Process
Raise complaints
Filter complaints
Solve complaints
Review
View Announceme nts
Make Announcement s
Student view and control
Hostel office view and control
HR view and control

Following table describes all the additional components which are not there in the previous architecture:

S.No	Component	Component Type	Description
1	Student view and control	Processing (Interface module)	This module is responsible for providing students with an intuitive user interface to manage room assignments, report issues, check facility status, and receive announcements within the HRMS.
2	HR view and control		This module provides HR's with a comprehensive interface to manage hostel operations, resolve complaints, make announcements, and communicate with hostel authorities.
3	HO view and control		This module serves as the administrative hub for overseeing hostel management operations, managing facilities, resolving escalated complaints, and communicating with HR's and other stakeholders.
4	Users	Databases	This module contains student/HR/HO personal details
5	In/Out		This module contains check-In/Out requests.
6	Exchange request		This module contains exchange requests.
7	HB		This module contains details of residents, facilities and their status in rooms/floor/block.
8	Complaints		This module contains complaints/ tickets raised by students.
9	Announcements		This module contains announcements created by HR/HO.

Description:

In this architecture, we have a clear separation of concerns, with a data access layer handling data retrieval, a business logic layer implementing core business logic, and a presentation layer containing the master controller and views. The data access layer retrieves data from the data repository, while the business logic layer processes this data to perform various operations required by the system. The presentation layer is responsible for displaying the results to the user's browser, with the master controller coordinating between the business logic layer and the views. This separation allows for easier maintenance, scalability, and flexibility in the system's design.

The primary advantage of separating the data access layer is that if there are any changes to the data repository layer, such as replacing the database with a new one, only the access layer needs to be modified, and not the logic in other layers. The business logic layer operates on data in the form of language-level objects provided by the data access layer. These objects, or models, can be easily converted into a displayable format by the controller using the corresponding views for each model.

4 Analysis (ATAM):

Attribute/Constraint	Architecture-1	Architecture-2
Data security	Less secure since all the modules are accessing the database	More secure since only DAL is allowed to access the database
Response Time	It takes more time as there is only one database and query execution takes more time due to many attributes	It takes less time as it has different databases and query execution takes lesser time
Application server down	When the application server is down no student, hostel office, HR can access the system	Here one user's server doesn't get effected with another user
Reading and understanding of code for developers	Difficult	Relatively Easy
Debugging	It can be difficult to debug as we might not able to identify which particular user has bugs.	Easy as we can test student, HR and Hostel Office functionalities separately and this makes debugging easier.
Switching database	Difficult as we only have one repository and if any problem occurs to that DB, data loss occurs.	It is relatively simpler as we have multiple DB.

After considering all the points mentioned in the table, the Data Access Layer Model appears to be the better choice. Its strength lies in its ability to handle changes more effectively by addressing them in specific layers rather than overhauling the entire service. Additionally, future functionalities and features can be developed more seamlessly as modules within their respective layers, allowing for greater code re usability across different parts of the system. Therefore, we have decided to adopt architecture 2, the four-layered architecture, for the HRMS Software.

Regarding the non functional attributes, we might have server breakdown related issues. In the given two architectures 2nd architecture will function faster because of faster execution of queries (due to many number of databases). In the 1st architecture we have only one database so we have many attributes so relatively slower execution.