IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HOTEL MANAGEMENT SYSTEM USING WEB DEVELOPMENT

Priya Dharmaraj Bhole[1], Prof. A.D. Gujar[2]
[1]Master of Computer Engineering
[1]TSSM's Bhivarabai Sawant College of Engineering and Research, Pune, India

Abstract: The Hotel Management System is a web-based application developed to automate and manage hotel operations efficiently. Traditional systems are manual, time-consuming, and error-prone. This system provides a secure, scalable, and user-friendly platform for managing room allocation, student records, complaints, and leave requests.

To enhance security, modern encryption techniques and authentication mechanisms are incorporated to protect sensitive data. The system uses cloud-based storage for better performance, availability, and scalability. It also ensures data confidentiality and integrity through secure login and access control mechanisms.

The proposed system reduces manual work, improves efficiency, and provides a reliable solution for both administrators and users**.**

Index Terms: Hotel Management, Web Application, Cloud Computing, Security, Automation

I. INTRODUCTION

he Hotel Management System is a web-based application designed to manage various hostel activities efficiently. It helps administrators maintain records of students, rooms, staff, and other operations in a centralized system.

Traditional systems rely on manual processes, which are time-consuming and prone to errors. This system eliminates manual work and provides a digital solution for managing hostel operations.

There are two main users in the system:

Admin: Manages students, rooms, staff, and overall system

User (Student/Staff): Accesses personal data, submits complaints, and applies for leave.

1.1 Goal

- To automate hostel management processes
- To ensure secure and efficient data handling
- To reduce manual errors and improve productivity

1.2 Problem Statement

Traditional hotel management systems are manual and inefficient. Users face difficulties in managing records and operations. Hence, a secure web-based system is required.

1.3 **Outcomes**

- Reduced paperwork
- Centralized data storage
- Improved efficiency

Scope

- Student data management
- Room allocation
- Complaint system
- Leave management

II. LITERATURE SURVEY

Existing systems mainly provide limited functionality and lack security features.
Attribute-Based Encryption (ABE) ensures secure access control.
Common security issues include unauthorized access, data leakage, and attacks.

the above website of hostel 15 is just a frontend website of hostel Management website. they have just given information about hostel 15. this website is divided into different modules such as Home, Gallery, Council, event and contact. at the to of the page homepage the main navigation menu they have keep it simple yet attractive they use white background and black color of font that makes it more readable to the user. in the home page along with navbar they also add images which enhance the user interface and makes website more interactive . in the footer of the website they have given a Links to their YouTube, Twitter, Instagram, Facebook, and LinkedIn. this website does not have a backend technology as it only a static website unlike this project our project have a backend technology that makes our project dynamic and interactive.

2.1 Ciphertext-Policy Attribute-Based Encryption

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is an advanced encryption technique used to provide fine-grained access control over data. In this method, the data owner defines an access policy, and only users whose attributes satisfy this policy can decrypt the data.
In CP-ABE, the encryption is done based on a policy (such as role, department, or designation). For example, a file can be encrypted with a policy like "(Admin AND HostelStaff) OR Warden". Only users who have these attributes in their private keys can access the data.
This approach is highly useful in systems where multiple users need different levels of access, such as cloud storage, healthcare systems, and organizational databases. It ensures data confidentiality and prevents unauthorized access.

*2.*2 Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data.

Attribute-Based Encryption (ABE) is an advanced form of public key encryption that provides fine-grained access control over encrypted data. In ABE, access rights are defined based on user attributes such as role, department, or designation rather than individual identities.

In this approach, data is encrypted with an access policy (e.g., "Admin AND Hostel Staff"). Only users whose attributes satisfy this policy can decrypt the data. This ensures that sensitive information is accessible only to authorized users.

ABE is highly useful in systems like cloud storage, healthcare, and organizational data sharing, where different users require different levels of access. It eliminates the need to encrypt data separately for each user, making the system more efficient and scalable

2.3 Security issues and requirements for Internet-scale publish-subscribe sys-tems Security Issues:

1. Unauthorized Access – Untrusted users may subscribe or publish data without permission.
2. Data Leakage – Sensitive information can be exposed to unintended subscribers.

3. Message Tampering – Attackers may modify messages during transmission.
4. Impersonation Attacks – Fake publishers or subscribers can send/receive data.
5. Lack of Confidentiality – Messages may not be encrypted properly.
6. Scalability Challenges – Security mechanisms become complex at large scale.
7. Key Management Issues – Handling encryption keys for many users is difficult

2.4 Event Guard: A System Architecture for Securing Publish-Subscribe Net-works:

Event Guard is a security architecture designed to protect publish–subscribe systems, where publishers send messages (events) and subscribers receive them based on their interests. These systems are widely used in distributed applications but are vulnerable to threats such as unauthorized access, data tampering, and privacy leakage

III. Application Based

3. 1 A Semantic Overlay for Self- Peer-to-Peer Publish/Subscribe

A semantic overlay in a peer-to-peer (P2P) publish/subscribe system is a network structure where nodes (peers) are organized based on the meaning (semantics) of the data they handle rather than just network location.

In a self-organizing P2P system, nodes automatically form connections with other nodes that share similar interests or data topics. This improves message routing efficiency because messages are delivered only to relevant subscribers, reducing unnecessary network traffic.

The semantic overlay uses metadata and content-based filtering to match publications with subscriptions. For example, if a user subscribes to “hostel room availability,” they will only receive related updates.

3.2 The PADRES Publish/ Subscribe System

The PADRES (Publish/Subscribe Architecture for Distributed Event-based Systems) is a distributed publish/subscribe system designed to support scalable and flexible communication between publishers and subscribers.
allows more precise and efficient data delivery making it suitable for large distributed systems such as cloud applications and real-time data processing systems. It also includes security mechanisms to ensure safe data transmission.

3.3 Hermes: A Scalable Event-Based Middleware

Hermes is a scalable, distributed, event-based middleware designed to support efficient communication in large-scale distributed systems. It follows the publish/subscribe (pub/sub) model, where publishers generate events and subscribers receive only the events they are interested in.

Hermes is designed to overcome limitations of traditional centralized systems by using a decentralized architecture. Instead of relying on a single broker, it distributes event routing across multiple nodes, improving scalability, fault tolerance, and performance.

Key Features of Hermes:
Scalability: Can handle a large number of publishers and subscribers.
Decentralization: No single point of failure, improving reliability.
Efficient Routing: Uses content-based filtering to deliver only relevant messages.
Dynamic Network: Supports nodes joining and leaving without affecting the system.
Low Latency: Faster communication due to distributed design.

Working of Hermes:
Publishers send events into the system.
The middleware routes these events through a network of brokers/nodes.
Subscribers receive only the events that match their interests.

Advantages:
High performance in large-scale systems
Better fault tolerance
Flexible and adaptive communication

Limitations:
Complex implementation
Security challenges in distributed environments.

3.4 Encryption-Enforced Access Control in Dynamic Multi-Domain Publish/Subscribe Networks

In dynamic multi-domain publish/subscribe networks, multiple domains (organizations or systems) communicate and share data. These environments are highly flexible, where publishers and subscribers can join or leave frequently. Ensuring secure access control in such systems is a major challenge.

Encryption-enforced access control is a technique where data is protected using encryption, and only authorized users can decrypt and access it. Instead of relying only on traditional access control mechanisms, security is embedded directly into the data.

3.5 Secure Distribution of Events in Content-Based Publish Subscribe Systems

Content-based publish/subscribe systems allow subscribers to receive messages based on the content of events rather than specific topics. While this increases flexibility, it also raises security concerns such as unauthorized access, data leakage, and privacy risks.

Secure distribution of events ensures that only authorized subscribers can access relevant data. This is achieved using encryption techniques, access control mechanisms, and filtering policies. Publishers encrypt event data, and only subscribers with matching permissions and valid keys can decrypt the information.

Techniques like attribute-based encryption (ABE) and identity-based encryption (IBE) are commonly used to enforce fine-grained access control. These methods ensure that even if messages are intercepted, unauthorized users cannot read them.

3.6 Scalable access control in content-based publish subscribe systems

Content-based publish/subscribe systems deliver messages to subscribers based on the content of the data rather than predefined topics. As the number of users increases, managing access control efficiently becomes challenging.

Scalable access control ensures that only authorized subscribers receive specific content without affecting system performance. This is achieved by using efficient filtering mechanisms and cryptographic techniques such as attribute-based encryption or identity-based encryption.

In this approach, access policies are defined based on user attributes (e.g., role, department, permissions). Messages are encrypted according to these policies, and only users who satisfy the conditions can decrypt and access the data.

3.7 Supporting Publication and Subscription Confidentiality in Pub/Sub Net-works

In publish/subscribe (pub/sub) networks, publishers send messages and subscribers receive them based on their interests. Ensuring confidentiality in such systems is important to protect both the published data and the subscriber's identity or interests.

Publication confidentiality ensures that only authorized subscribers can access the published messages. This is achieved by encrypting the data before it is transmitted, so unauthorized users cannot read it.

Subscription confidentiality focuses on protecting the subscriber's interests or subscription details. Without proper security, attackers may infer sensitive information about users based on their subscriptions. Techniques like encryption, anonymization, and secure matching protocols help hide subscriber identities and interests

IV PROPOSED SYATEM

4.1 Background .

With the rapid growth of distributed systems, cloud computing, and IoT applications, secure communication has become a major concern. Traditional encryption techniques rely on Public Key Infrastructure (PKI), which involves complex certificate management and high computational overhead.

In many modern systems such as broker-less publish/subscribe models, the absence of a central authority makes security even more challenging. Ensuring confidentiality, authentication, and secure key distribution becomes difficult in such decentralized environments.

To address these issues, Identity-Based Encryption (IBE) has emerged as an effective solution. It simplifies the encryption process by using user identities as public keys, eliminating the need for certificates. This makes it highly suitable for dynamic and large-scale systems where ease of use and strong security are essential.

In many modern systems such as broker-less publish/subscribe models, the absence of a central authority makes security even more challenging. Ensuring confidentiality, authentication, and secure key distribution becomes difficult in such decentralized environments.

4.2 **Cloud Computing**

Cloud Computing is a technology that provides computing services such as storage, servers, databases, networking, and software over the internet.

Instead of using local computers or physical servers, users can access these resources from anywhere using cloud platforms.

It works on a pay-as-you-go model, which means users only pay for the resources they use.

Cloud computing is widely used because it reduces infrastructure cost and provides flexibility and scalability.

There are three main service models in cloud computing:

IaaS (Infrastructure as a Service): Provides virtual machines and storage

PaaS (Platform as a Service): Provides development platforms for building applications.

SaaS (Software as a Service): Provides software applications over the internet.

Cloud computing also offers advantages such as high availability, data backup, security, and easy access from multiple devices.

It is used in various fields like education, business, healthcare, and data storage system

4.3 Attacker model: cloud computing

The attacker model in cloud computing defines the possible threats and types of attackers that can compromise the security of cloud systems. It helps in designing secure systems by identifying how attacks can occur.

In cloud computing, attackers can be classified as:

External attackers – Unauthorized users outside the system who try to access cloud data through hacking, phishing, or malware attacks.

Internal attackers – Authorized users (employees or administrators) who misuse their access privileges.

Malicious tenants – In multi-tenant environments, one user may try to access or interfere with another user's data.

- Common attack methods include:
- Data breaches and data leakage
- Denial of Service (DoS) attacks
- Man-in-the-Middle attacks
- Account hijacking
- Malware injection

The attacker model assumes that attackers may have different levels of knowledge, such as system architecture, APIs, or network details. Therefore, cloud systems must implement strong security measures like encryption, authentication, access control, and monitoring.

Understanding the attacker model helps in improving cloud security by preparing defenses against potential threats and ensuring data confidentiality, integrity, and availability.

4.4 TPA

A Third Party Auditor (TPA) is an independent entity used to verify and audit data stored in a system, especially in cloud environments. The main role of the TPA is to ensure that the data stored by users is secure, intact, and has not been modified or corrupted.

In systems involving secure communication or storage, the TPA performs auditing without accessing the actual content of the data, thus maintaining data privacy. It checks data integrity using cryptographic techniques and provides reports to users about the status of their data.

The use of TPA reduces the burden on users, as they do not need to manually verify their data. It also increases trust between service providers and users by ensuring transparency.

However, the TPA must be reliable and secure, as it plays a crucial role in maintaining system integrity.

4.5 Proposed System

The proposed system focuses on developing a secure and efficient communication framework using Identity-Based Encryption (IBE). The system is designed to overcome the limitations of traditional security mechanisms by simplifying key management and enhancing data confidentiality.

In this system, users can securely exchange information without relying on complex certificate-based infrastructures. The sender encrypts the data using the receiver's identity, and only the intended receiver can decrypt it using a private key issued by a trusted authority. This approach ensures secure data transmission in both sender/receiver and broker-less publish/subscribe environments.

3.2.1 System Flow PHASE-I:

4.6 System Flow

describes the overall working process of the Hostel Management System, showing how data moves between users, system modules, and the database. It explains how different operations like login, registration, room allocation, and complaint handling are performed step by step.

The system mainly consists of two users:

Admin

Student/User

4.6 Architecture.

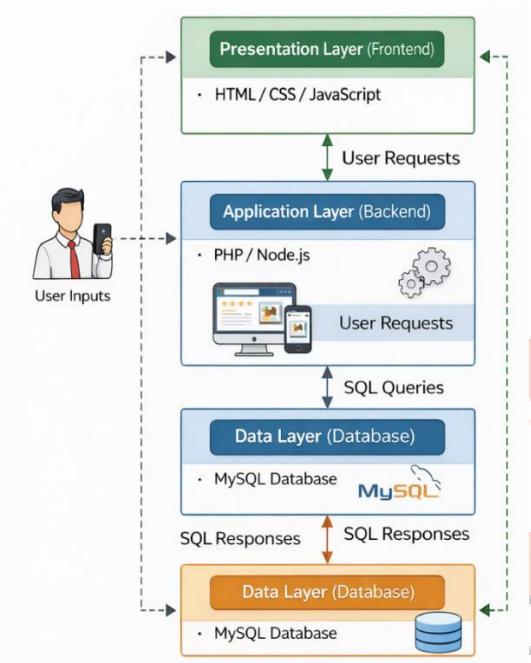


Figure 3.1: System architecture

The flow begins when the user interacts with the system through the web interface and ends with storing or retrieving data from the database.

Phase–I : ensures smooth user authentication, data entry, and initial interaction with the system, forming the foundation for further operations.

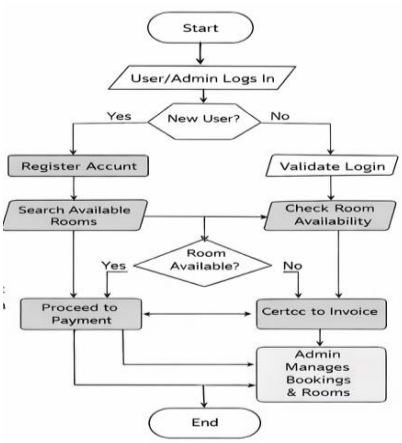
The proposed system follows a three-tier architecture:

A. Presentation Layer
Developed using HTML, CSS, and JavaScript to provide an interactive user interface.

B. Application Layer
Implemented using PHP/Node.js to handle business logic and user requests.

C. Data Layer
MySQL database used for storing and managing system data

4.7 Algorithm



V.Mathematical Model

Set Theory

The mathematical model of the Hostel Management System represents the system in terms of inputs, processes, and outputs.

1. Set Definition

S = {Student, Admin, Staff}
R = {Rooms}
A = {Applications}
C = {Complaints}
L = {Leave Requests}

Input

Student details (Name, ID, Contact, Email)
Room details (Room No., Capacity)
Login credentials (Username, Password)
Application/Complaint/Leave data.

Functions

f1: Register Student → Adds student to database
f2: Login → Authenticates user
f3: Allocate Room → Assigns room to student
f4: Submit Complaint → Stores complaint
f5: Apply Leave → Records leave request
f6: Approve/Reject → Admin decision

Constraints

Only admin can modify/delete data
Valid login required for access
Room allocation based on availability
Unique ID for each student

5.1 External Interface Requirement

User Interfaces

- The User Interface (UI) of the Hostel Management System is designed to be simple, user-friendly, and easy to navigate for both administrators and users (students/staff). The system provides different interfaces based on user roles to ensure efficient interaction.
- Admin Interface
- Secure login page for administrator access
- Dashboard to manage students, staff, and rooms
- Options to add, update, delete student and staff records
- View complaints, applications, and reports
- Room allocation and deallocation features
- User Interface (Student/Staff)
- Registration and login pages
- Profile viewing and updating options
- Application form for leave requests
- The overall UI ensures smooth communication between users and the system while reducing complexity and improving efficiency

Hardware Interfaces

- Hardware interfaces define how the Hostel Management System interacts with physical devices required for its operation. Since the system is a web-based application, the hardware requirements are minimal and commonly available.
- The system requires a computer or laptop with a processor capable of running a web browser efficiently. Input devices such as a keyboard and mouse are used for entering and managing data. A display monitor is required to view the application interface.

- For network communication, a stable internet connection is necessary to access the system, especially if it is hosted on a remote server. The server-side may require a machine with sufficient processing power, memory, and storage to handle multiple users and database operation.

Software Interfaces

- he Hostel Management System interacts with various software components to ensure smooth functioning and user accessibility. These software interfaces define how the system communicates with other applications, databases, and user platforms.
- The system is developed as a web-based application and uses standard web technologies. The frontend interface is built using HTML, CSS, JavaScript, and Bootstrap, which provides an interactive and user-friendly environment for users such as administrators and student

System Features

System Features-I (Functional Requirements)

- Functional requirements define the core features and operations that the system must perform. The “Hostel Management System” includes the following key functional requirements:
- 1. Admin Module
- Admin can add, update, and delete student records.
- Admin can allocate and deallocate hostel rooms.
- Admin can view all registered students.
- Admin can manage staff details.
- Admin can view and respond to student complaints.
- Admin can approve or reject leave applications.
- 2. User (Student) Module
- Students can register and create an account.

- Students can log in securely using credentials.
- Students can view and update their profile.
- Students can apply for leave.
- Students can submit complaints.
- Students can check application status.
- 3. Authentication System
- Secure login system using username and password.
- Only authorized users can access system features.
- 4. Data Management
- System stores and retrieves student, room, and staff data efficiently.
- Allows easy updating and deletion of records.
- 5. Complaint & Leave Management
- Students can submit complaints and leave requests.
- Admin can track and manage all requests.

5.2 System Features-II (Non-Functional Requirement.

Non-functional requirements define the quality, performance, and reliability of the system. The Hostel Management System includes the following:

1. Security
- Only authorized users can access the system using login credentials.
- Admin has full control over data modification (add, update, delete).
- Data is protected from unauthorized access.

2. Performance
- System provides quick response time for user actions.
- Supports multiple users simultaneously without performance issues.
- Efficient data processing and retrieval.

3. Reliability
- System should be available at all times with minimal downtime.
- Ensures accurate and consistent data handling.

4. Maintainability
- Easy to update and modify the system when required.
- Supports regular data backup and recovery.
- Errors are logged and can be tracked easily.

5. Usability
- Simple and user-friendly interface.
- Easy navigation for both admin and students.

5.3 Conclusion

The “Hostel Management System” successfully provides a digital solution to manage hostel activities efficiently. It reduces manual work and improves the accuracy of maintaining student, staff, and room records. The system ensures secure data handling and provides easy access to important.

VI .SYSTEM ANALYSIS

6.1 POJECT PLAN

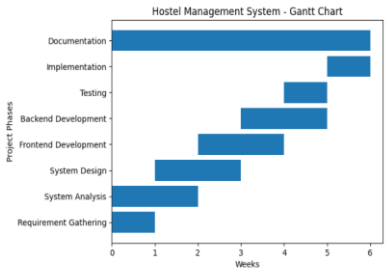
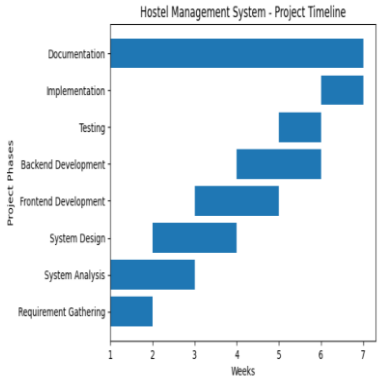


Figure 5.1: Project Plan

6.2 Time Line Of Project



6.3 RISK ANALYSIS

Risk Analysis is the process of identifying potential problems that may affect the development and performance of the system and planning ways to minimize them.

1. Technical Risks

System bugs or errors during development

Compatibility issues with different browsers or devices

Mitigation: Proper coding practices and thorough testing.

2. Security Risks

Unauthorized access to data

Data breaches or loss of sensitive information

Mitigation: Use of login authentication, validation, and secure database handling

3. Performance Risks

Slow system response time

System crash due to high user load

4. Operational Risks

Users may face difficulty in using the system

Lack of proper training

Mitigation: Provide user-friendly interface and basic guidance.

VII SYSTEM DESIGN DIAGRAMS

7.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of how data moves through a system. It shows the flow of information between external entities, processes, and data stores in the “Hostel Management System”.

The main purpose of the DFD is to explain how data is processed, stored, and used within the system.

7.2 DFD Level 0

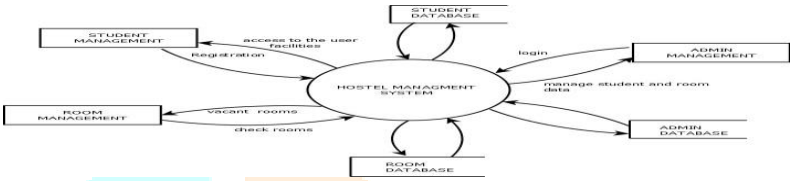


Figure 7.2: DFD Level 0

7.3 ***DFD Level 1***

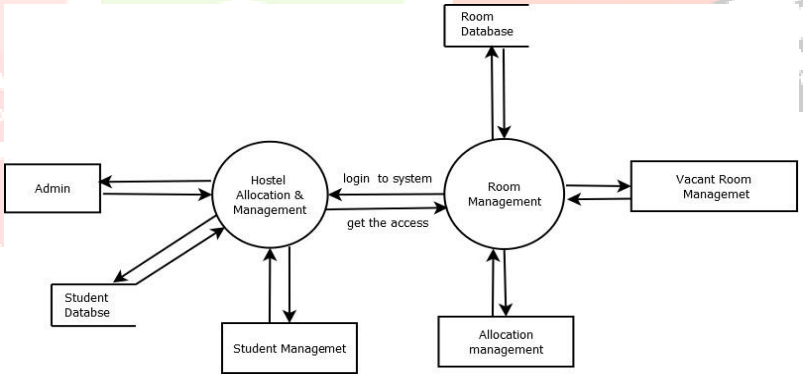


Figure 7.3: DFD Level

7.4 Use Case Diagram

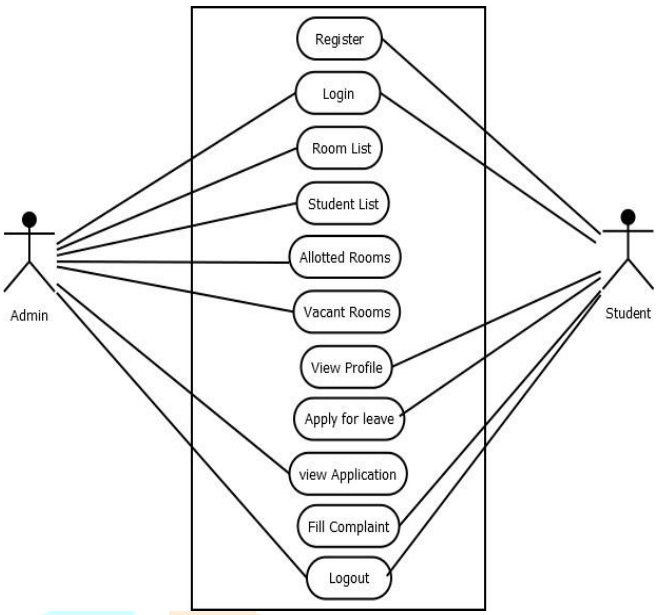


Fig 7.4 Use Case Diagram

7.5 Activity Diagram

Activity Diagram shows the workflow of the system step by step.

It represents the flow of control from one activity to another.

It is used to understand the dynamic behavior of the Hostel Management System.

It clearly describes how admin and student activities are performed.

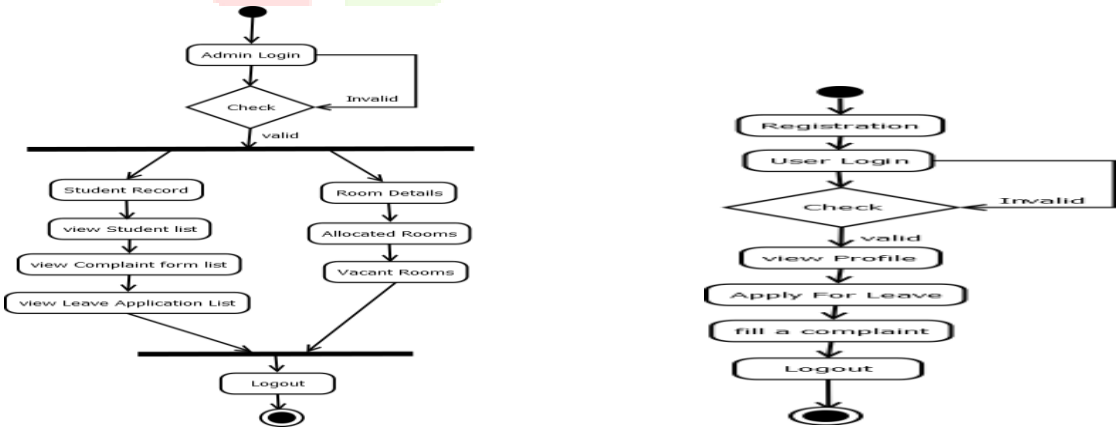


Fig 7.5 Activity Diagram.

REFERENCES

[1] M.A.Tariq, B. Koldehofe, and K. Rothermel, "Securing Broker-Less Pub-lish/Subscribe Systems Using Identity-Based Encryption",IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014.

[2] S. N. Shitole, A D Gujar, "Survey of Identity-Based Encryption for Providing Se-curity in Sender/Receiver Systems", International Journal of Science and Research (IJSR), Volume 4 Issue 1, January 2015.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based En-cryption," Proc. IEEE Symp. Security and Privacy, 2007.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. ACM 13th Conf. Computer and Comm. Security (CCS), 2006).