Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	12.10.2023
Team ID	NM2023TMID06074
Project Name	Analysing the Performance and Efficiency of the Radisson Hotels using Data Visualization Techniques using IBM COGNOS
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

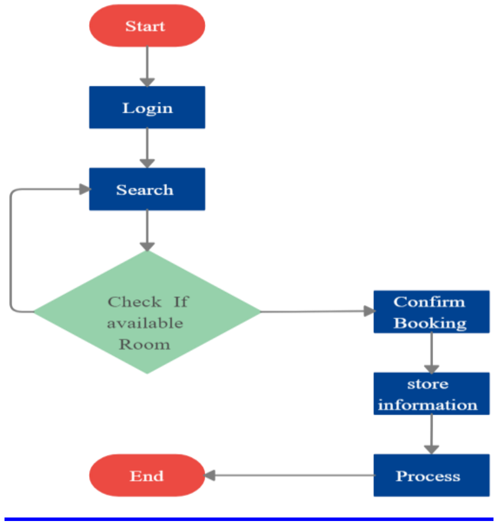

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Develop user-friendly web and mobile applications for guests to make reservations, check-in/check-out, request services, and provide feedback.	React for web applications, React Native for mobile applications
2.	Application Logic-1	Implement business logic to handle reservations, room allocation, pricing, and availability.	Java or Python
3.	Application Logic-2	Develop modules for check-in/check-out, room service requests, and billing.	IBM Watson Speech to Text (STT) service
4.	Application Logic-3	Implement a system to handle reservations, including booking, modification, and cancellation.	IBM Watson Assistant
5.	Database	Set up a relational database to store information about rooms, reservations, guest profiles, and transaction history.	PostgreSQL
6.	Cloud Database	Utilize cloud-based database solutions to enable scalability and redundancy.	IBM Db2 on the cloud
7.	File Storage	Use cloud-based file storage for storing images, documents, and logs.	IBM Cloud Object Storage or similar cloud-based storage solutions
8.	External API-1	Integrate with external APIs for payment processing, geolocation services, and other third-party systems.	Stripe or PayPal for payment processing
9.	External API-2	Ensure secure and reliable communication with external services.	OAuth 2.0 for authentication and authorization
10.	Machine Learning Model	Implement machine learning models to predict room demand and optimize pricing.	Advanced machine learning models and data preprocessing
11.	Infrastructure (Server / Cloud)	Deploy the system on cloud infrastructure for scalability and flexibility. Implement load balancing and auto-scaling to handle varying workloads.	Cloud platform (e.g., IBM Cloud or AWS), Kubernetes for container orchestration

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilize open-source frameworks for building and maintaining the software components. For example, use open-source web development frameworks for creating booking and reservation systems.	Node.js with Express for backend, React for frontend
2.	Security Implementations	Prioritize security in all aspects of the system, from guest data protection to financial transactions.	SHA-256 for data hashing, Strong encryption algorithms, Identity and Access Management (IAM) controls, OWASP for security best practices
3.	Scalable Architecture	Design an architecture that can handle varying levels of demand throughout the year. This is essential for accommodating high occupancy periods.	Cloud-based architecture (e.g., AWS, Azure, Google Cloud), Container orchestration with Kubernetes, Auto-scaling solutions for dynamic resource allocation

S.No	Characteristics	Description	Technology
4.	Availability	Ensure high availability of critical services, such as the reservation system and the guest check-in/check-out process.	**Load Balancers:** Load balancers distribute incoming network traffic across multiple servers to ensure that no single server is overwhelmed, improving the system's availability.
5.	Performance	Optimize system performance for quick response times and a seamless guest experience. Use caching mechanisms to reduce the load on the database and speed up data retrieval.	**Caching Mechanisms:** Utilizing caching systems like Redis or Memcached to store frequently accessed data in memory, reducing the need to fetch the same data from the database repeatedly.