Intelligent Floor Plan Management System for a Seamless Workspace Experience

*A Case Study by **Sai Chaitanya***

Objectives

To cultivate a workplace characterized by enhanced dynamism, collaboration, and efficiency, it is imperative to acknowledge the ever-evolving nature of workspaces and the diverse requirements of the workforce.

To address this, leveraging technology becomes crucial for optimizing floor plan management. The overarching objective is to design a workplace that seamlessly adjusts to the shifting needs of employees while maximizing the utility of existing resources. This transformative approach entails organized implementations, ensuring zero risk of conflicts or unintended changes, and is designed to be both reliable and scalable.

Embracing technology in floor plan management facilitates a streamlined, adaptable environment that not only meets the current demands of the workforce but also lays the foundation for sustained adaptability in the future.

Tasks

* Meeting Room Optimization (Meeting Room Suggestions and Booking)

* Admin's Floor Plan Management (Onboarding/Modifying the floor plans)

* Ofline Mechanism for Admins (For the UI person)

Schema

Tables:

->user: user_id,user_name,password,role,coordinates((x,y,z)wrt building)

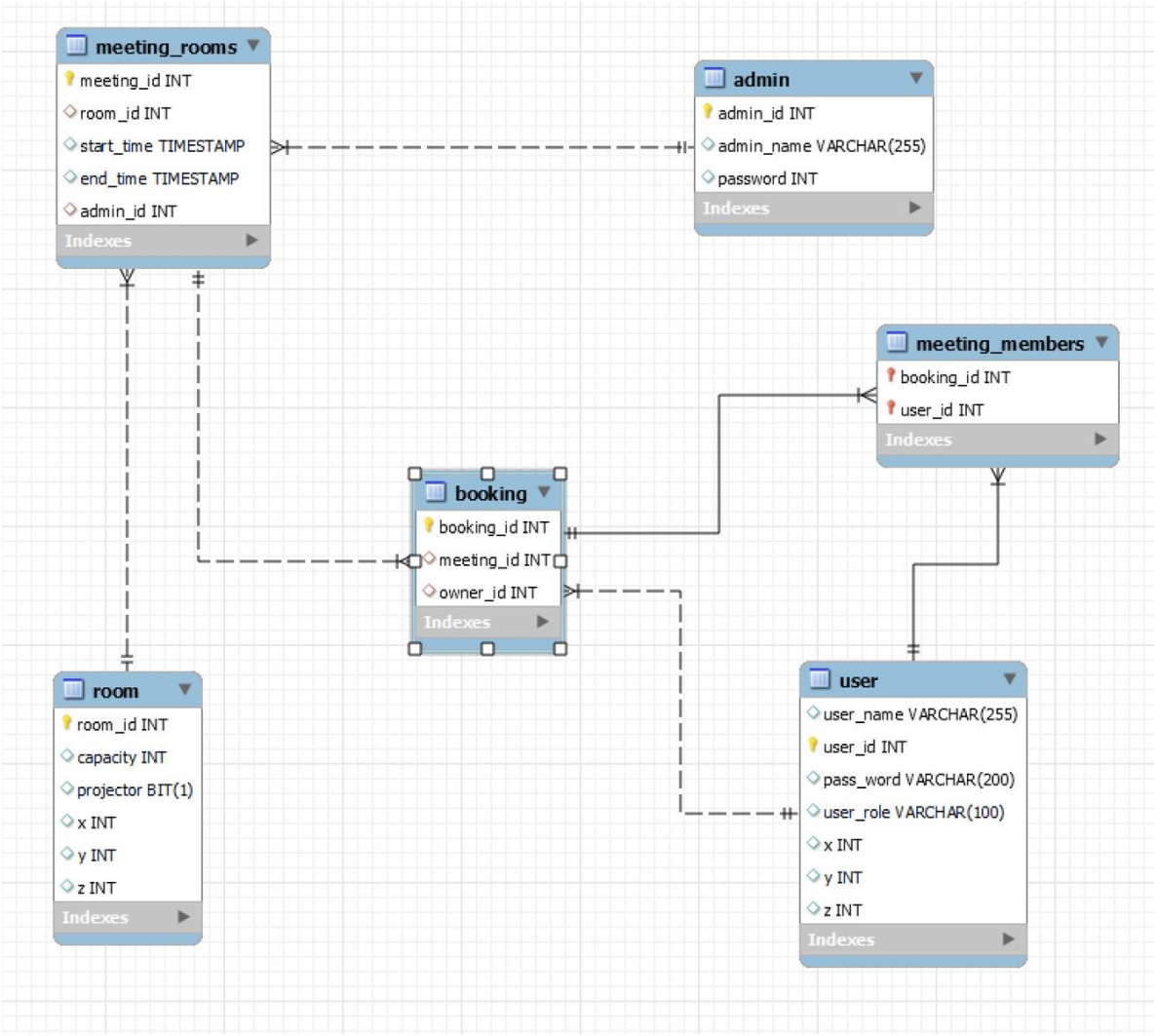
->room: room_id,capacity***,*** attributes like contains projector,coordinates.

->meeting_rooms: id,room_id,start time,endtime,admin.

->booking: id,meeting_id,owner_id.

->meeting_members: booking_id,user_id.

->admin: admin_id,name,password.



Floor plan management

Authorized administrators have access to a dedicated portal for uploading floor plans, where the floor plans are meticulously analyzed and transformed into numerical data for enhanced manageability. Extracting information from CAD models or floor plan images, we convert them into room coordinates. Each building is conceptualized as a 3D model, and individual rooms are assigned coordinates. These coordinates, corresponding to specific room or seat types, serve to streamline identification and categorization. This systematic approach extends to meeting rooms and pathways, ensuring a comprehensive and standardized representation of the workspace.

- ➔ Since it is difficult to measure the distances with a tape for coordinates,we can divide the z -axis by unit size which is the floor no. and we divide each floor into square shapped cells ,all the objects in a cell will be assigned the coordinates of nearest vertex of the cell.(by marking coordinates on the desk).
- ➔ User can upload/update the coordinates in his profile. Admin will upload/update the coordinates of rooms(actually the coordinates room's door)
- ➔ So this process helps us to find the nearest room with respective all the meeting participants.

Meeting room bookings:

In order to enhance user experience and streamline the room booking process, the system has been designed to provide each user with access to specific rooms as authorized by the administrator. Rooms not granted access will be visually distinguished by graying them out, ensuring users only interact with the relevant and permitted options. The core data structures include a bookings object, capturing comprehensive information such as User ID, Room ID, booking timestamp, and meeting time range. Additionally, a room object is maintained, housing details like its ID, type, attendees, and location. When a user initiates a booking, they are prompted to input the names of all invited members. This serves a dual purpose: determining the minimum room size based on expected attendees and mapping the locations of invitees' desks. Leveraging this information, the system employs a Best-fit algorithm to suggest

an available meeting room that is equidistant from all attendees and of optimal size to accommodate the group. Furthermore, functionality can be implemented to automatically send email invitations and reminders to the invited members, utilizing the initial data collected by the host. This comprehensive approach ensures a seamless and efficient room booking process with added convenience for both administrators and users.

Room suggestion algorithm:

The room selection process involves a systematic evaluation to meet user-specific requirements, encompassing criteria such as capacity and availability of amenities like projectors.

Initially, all rooms meeting these prerequisites are identified by using **BINARY_SEARCH**. Subsequently, a two-step sorting algorithm is employed to optimize the ordering of available rooms.

The first sorting criterion is based on capacity, arranging rooms in ascending order to facilitate selection based on group size.

Additionally, a secondary sorting factor is proximity, ensuring that rooms with similar capacities are further ranked by their distance from the median location of the participants.

The algorithm introduces a nuanced approach where, if the difference in capacity between two rooms is less than 25%, priority is given to the room that is closer to the calculated median of participant locations.

This refined ordering system aims to provide users with a curated list of rooms that not only accommodate their specified criteria but also consider spatial proximity, enhancing the overall efficiency of the room selection process.

The median coordinate of all participants can be determined by calculating the arithmetic mean of their respective x, y, and z coordinates, providing a central point that represents the spatial midpoint of the participant locations.

Time and Space complexity

The time and space complexity of the proposed algorithms exhibit an efficiency of O(N) per user, with N representing the number of rooms per floor.

The linear time complexity, denoted as O(N), signifies that the algorithm's execution time scales proportionally with the number of rooms per floor. This scalability ensures that even in scenarios with a substantial user load, the system remains responsive and agile.

Analysis

Evaluating system performance is crucial for optimizing workflow and efficiency in office operations. Using metrics, we can calculate room utilization by analyzing bookings data within a specified time range. Visualizing this information through color-coded regions based on utilization percentages, such as red for 0%-50% and green for 90%-100%, offers an intuitive representation. This approach not only quantifies room usage but also provides a quick visual guide to identify underutilized or highly utilized areas. The combination of quantitative metrics and visual representation facilitates informed decision-making for workspace optimization, ensuring a more effective and responsive office environment.

Conflict Resolution:

Utilizing the geographical coordinates of conflict regions, the system has the capability to generate a Floor Plan layout that strategically highlights these conflict areas, employing a distinctive color such as red to visually denote the presence of conflicts. The resolution of conflicts is governed by a set of prioritization mechanisms, listed in decreasing order of precedence:

- ➔ **Company Priority:** Defined by company-specific parameters, this criterion represents the importance of tasks. Tasks with higher business priority are accorded precedence over others, ensuring strategic alignment.

- ➔ **User Role:** Reflecting the principle of authority, users with higher hierarchical positions possess the ability to override existing plans and

submit their own. Lower-priority tasks can be seamlessly relocated to similar room or seat configurations.

➔ **Timestamp:** Considering the chronological aspect, plans are prioritized based on the time of upload. Earlier submissions take precedence over later ones, allowing for a chronological hierarchy in conflict resolution.

If multiple users modify the same section of a document simultaneously, System detects these conflicts by comparing the order in which the server received them. The operation is applied directly to the document if there's no conflict. When conflicts occur, System resolves these conflicts by transforming the operations to maintain the document's integrity and ensure consistency to accommodate both changes without altering the intended meaning.

This conflict resolution process can be executed either automatically or manually, depending on the organizational needs and preferences. Users affected by lower-priority relocations are promptly notified of the changes and given the flexibility to modify their bookings if necessary, fostering transparency and adaptability within the system.

Authentication:

The authentication process within the system leverages the OAuth 2.0 framework, facilitating user authentication through the accounts associated with their respective organizations. OAuth APIs tailored for platforms like Google or Outlook are seamlessly integrated, enabling users to authenticate using their existing organizational credentials. The authenticated information, inclusive of role hierarchy within the organization, is then stored securely. This comprehensive user profile data becomes instrumental in implementing a refined access control system, dictating which areas of the office each user is authorized to access. Additionally, the role hierarchy information proves invaluable in governing privileges, ensuring that users can book rooms in accordance with their designated roles and responsibilities within the organizational structure. This integration not only enhances security but also streamlines the room booking process by aligning it with the organizational hierarchy and access permissions.

Dynamic updates:

To get real-time updates in case the other users are editing/uploading the floor plans, one can either use Polling or Web Sockets. However, polling suffers from the problem of increasing unnecessary network traffic, which could slow down the system and cost the company dearly. One should go for Web Sockets. They allow for a full duplex connection between the client and server efficiently and in real-time.

Versioning:

For efficient versioning of floor plans, the utilization of a Time Series Database (TSDB) presents a viable solution, with InfluxDB being a noteworthy choice in this regard. However, it's worth noting that employing a TSDB may lead to unnecessary storage expansion, especially when saving identical plans with minor edits. Furthermore, tracking changes would necessitate a full plan comparison, introducing potential inefficiencies.

An alternative approach involves leveraging a Merkle Tree for version control. This method involves generating a hash for each chunk of information, storing these hashes as leaf nodes. Subsequently, the combination of children nodes' hashes is calculated to generate the hash for the parent node, accomplished with an O(n) time complexity. This hierarchical structure allows for efficient tracking of changes across the tree with an O(log n) time complexity. Not only does this approach significantly reduce storage requirements, but it also enhances the speed of change tracking between different plan versions. The Merkle Tree methodology thus proves to be a robust and resource-efficient solution for implementing version control in floor plans.

Offline Mechanism for Admins

Local storage serves as an effective browser storage mechanism, facilitating the retention of local copies that users can access when ofline. When accessing documents online, the browser strategically caches them, enabling seamless

ofline access to previously viewed content. Real-time synchronization ensures that any modifications made to documents while online are instantly mirrored on the servers. Simultaneously, these alterations are locally stored within the browser, guaranteeing the availability of the most up-to-date version even when ofline.

The integration of service workers further enhances ofline accessibility by caching both assets and documents. This allows the service workers to serve locally cached content to users when they are ofline, ensuring a continuous and uninterrupted browsing experience. Upon reconnecting to the internet, the device initiates a synchronization process, uploading locally saved changes back to the server. Operation transformers play a crucial role in this synchronization, seamlessly incorporating changes made ofline into the overall dataset. This comprehensive approach not only ensures data consistency across online and ofline modes but also optimizes the user experience by allowing continuous access to the latest document versions.

System failures

In the event of system failures, the adoption of distributed storage systems such as Google Cloud Storage or Amazon S3 is essential to mitigate risks associated with a single point of failure. The implementation of the Gossip protocol emerges as a reliable strategy for sharing floor plans and upholding consistency across distributed nodes, contributing to fault tolerance and system resilience.

To enhance scalability, the integration of sharding or partitioning techniques proves beneficial, distributing the workload across multiple nodes. Replication mechanisms and regular backups are vital components of a robust system architecture, ensuring data durability and offering a layer of protection against unforeseen failures. Efficient retrieval of document metadata is achieved through the strategic use of databases and indexing, optimizing the performance of data queries.

Moreover, to bolster data security, a proactive approach involves encrypting the stored data, providing a safeguard against potential cyber attacks. By incorporating encryption methodologies, the system ensures that sensitive information remains protected and integral, even in the face of security threats. This comprehensive set of strategies contributes to the establishment of a resilient and secure infrastructure, addressing potential points of failure and fortifying the overall reliability of the system.

System Monitoring

System monitoring is a critical aspect of maintaining the health and performance of a system. Elastic Search is a powerful tool often utilized for logging and querying data, making it an excellent choice for system monitoring purposes.

- ➔ **Logging Data:** Elastic Search excels at efficiently storing large volumes of log data. In the context of system monitoring, various components of the system, such as applications, servers, or services, can generate log entries. These logs capture important information about the system's behavior, performance metrics, errors, and other relevant events. By utilizing Elastic Search, these logs can be structured and indexed for easy retrieval and analysis.
- ➔ **Querying Logs:** Elastic Search provides a robust querying mechanism that allows users to search and analyze log data quickly. This is particularly valuable during system monitoring as it enables administrators and developers to identify issues, troubleshoot problems, and gain insights into the overall health of the system. Queries can be tailored to filter specific logs based on criteria such as time range, severity, or source.
- ➔ **Fault Detection and Troubleshooting:** In case of faults or issues within the system, administrators can leverage Elastic Search to pinpoint the root cause by querying the logs. For example, they can search for error messages, anomalies, or patterns indicative of performance degradation. This facilitates proactive monitoring and swift identification of issues before they escalate.

- ➔ **Real-time Monitoring:** Elastic Search supports real-time indexing and querying, allowing for continuous monitoring of the system. This real-time capability is crucial for promptly detecting and responding to any deviations from normal system behavior.
- ➔ **Scalability:** Elastic Search is designed to scale horizontally, making it well-suited for handling large and growing datasets. This scalability is advantageous for systems with varying workloads and data volumes.

In summary, Elastic Search serves as an integral component of system monitoring by efficiently logging diverse data types and offering powerful querying capabilities. Its flexibility, scalability, and real-time features make it an ideal tool for administrators and developers seeking to maintain the reliability and performance of their systems.

Thank You