2024

Software Architecture Document

DRIESSEN

SANCHEZ, FRANCISCO; THOMAS, GILTON; PESTANA, CRISTIANO; WERNECKROALE, MIGUEL; LE, MINH

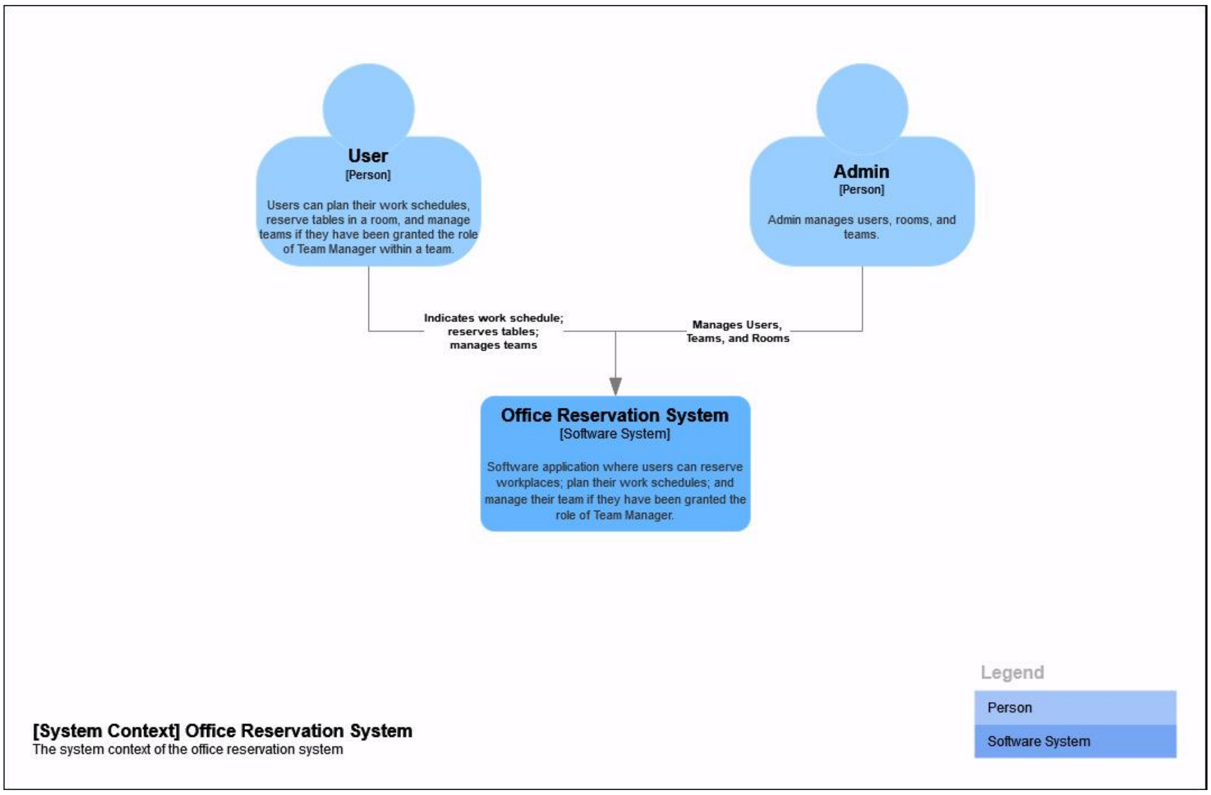
FONTYS UNIVERSITY OF APPLIED SCIENCE | Rachelsmolen 1, 5612 MA Eindhoven

Table of Contents

C4 Model Diagrams

Level 1: System Context Diagram

A System Context diagram provides a starting point, showing how the software system in scope fits into the world around it.


User
[Person]
Users can plan their work schedules, reserve tables in a room, and manage teams if they have been granted the role of Team Manager within a team.
Admin
[Person]
Admin manages users, rooms, and teams.
Indicates work schedule; reserves tables; manages teams
Manages Users, Teams, and Rooms
Office Reservation System
[Software System]
Software application where users can reserve workplaces; plan their work schedules; and manage their team if they have been granted the role of Team Manager.
Legend
Person
Software System
[System Context] Office Reservation System
The system context of the office reservation system

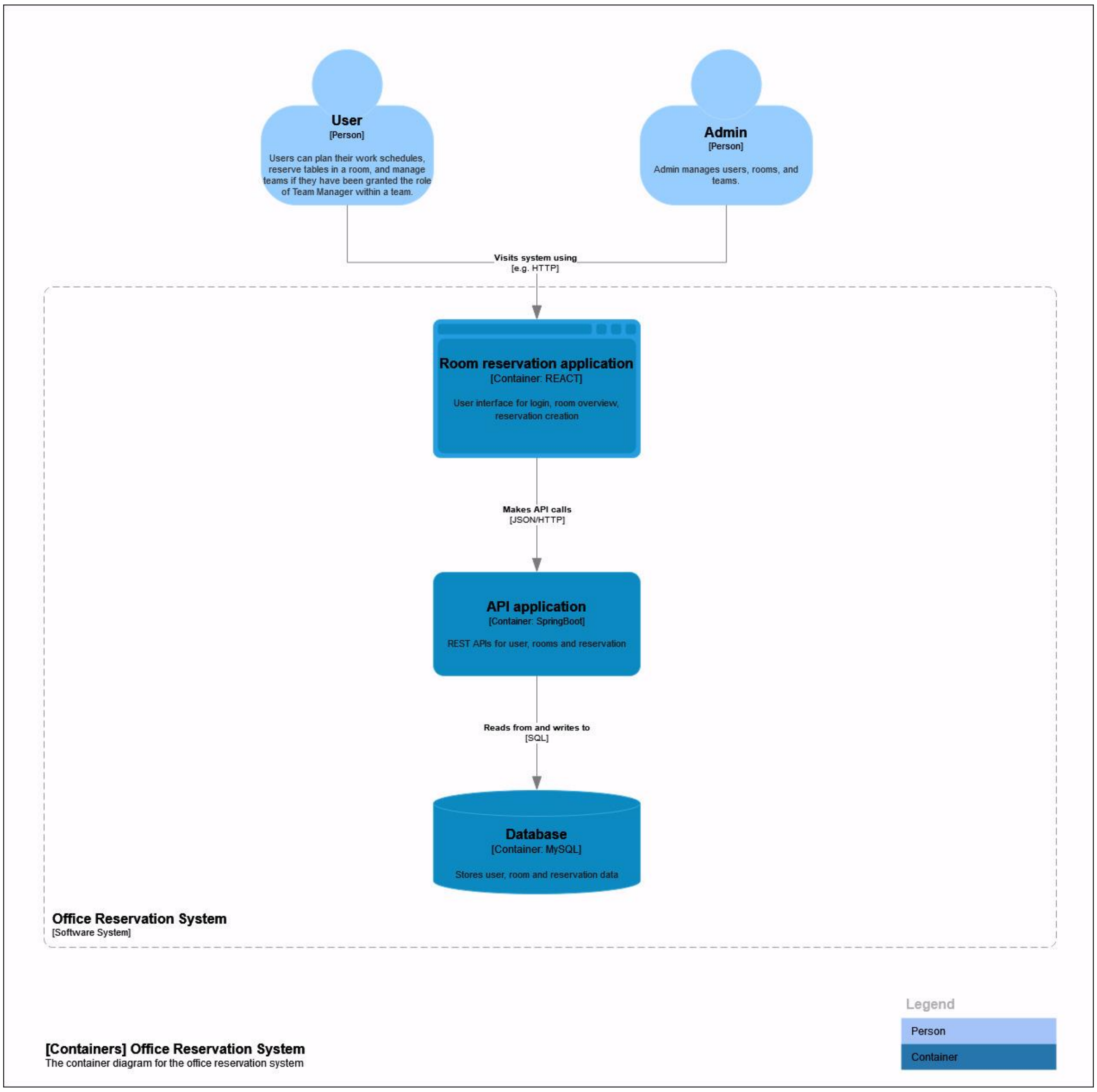
The system context diagram depicts the interactions between the users and the Office Reservation System. The system primarily interacts with two types of users:

1. Users: Regular users can plan their work schedules, reserve tables in rooms, and manage teams if granted the ‘Team Manager’ role within their team.
2. Admins: Admins are responsible for managing users, teams, and rooms.

The Office Reservation System is an application that facilitates these functionalities, serving as the main point of interaction for both user types.

Level 2: Container Diagram

A Container diagram zooms into the software system in scope, showing the high-level technical building blocks.


User
[Person]
Users can plan their work schedules, reserve tables in a room, and manage teams if they have been granted the role of Team Manager within a team.
Admin
[Person]
Admin manages users, rooms, and teams.
Visits system using
[e.g. HTTP]
Room reservation application
[Container: REACT]
User interface for login, room overview, reservation creation
Makes API calls
[JSON/HTTP]
API application
[Container: SpringBoot]
REST APIs for user, rooms and reservation
Reads from and writes to
[SQL]
Database
[Container: MySQL]
Stores user, room and reservation data
Office Reservation System
[Software System]
[Containers] Office Reservation System
The container diagram for the office reservation system
Legend
Person
Container

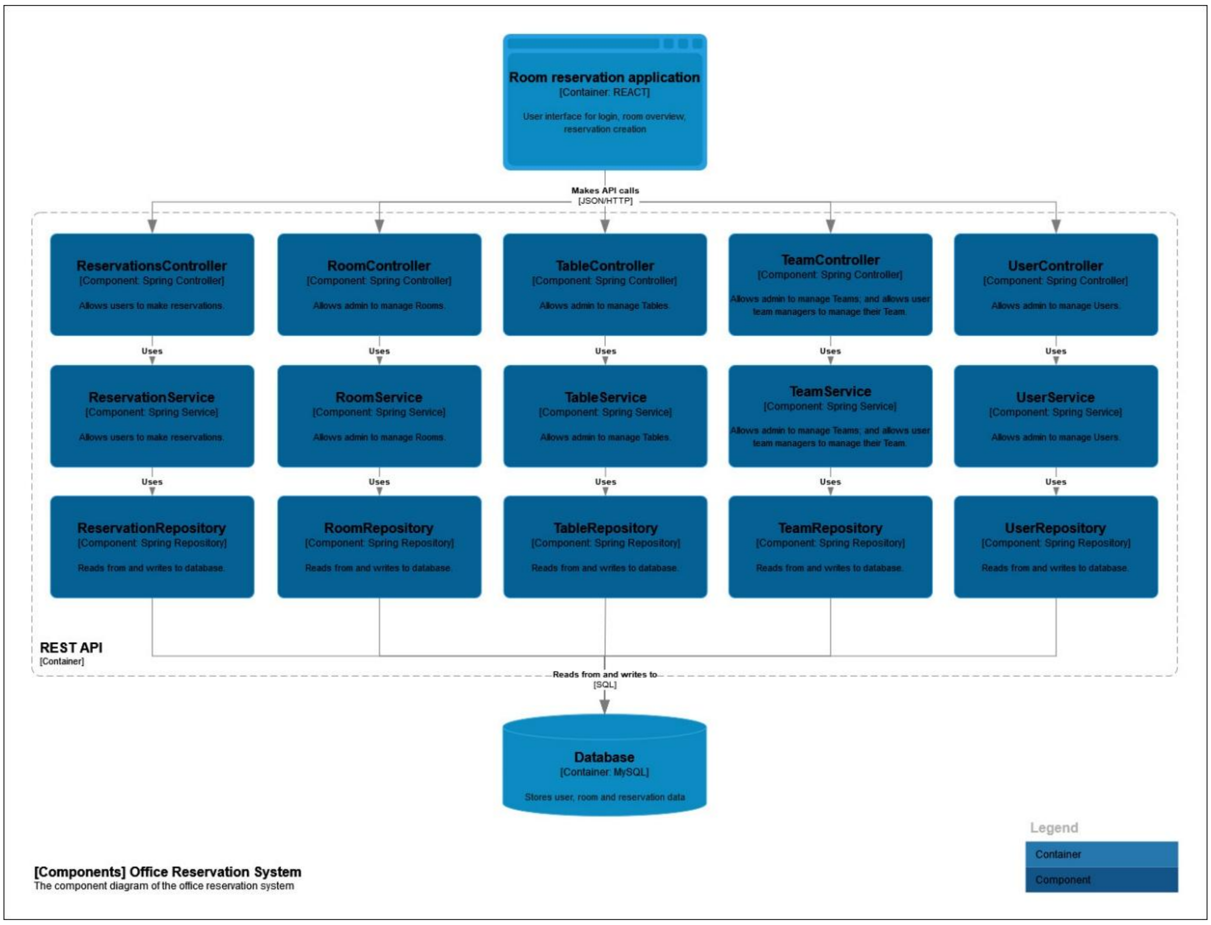
The container diagram illustrates the internal structure of the Office Reservation System, showing how various components interact. Key containers include:

1. Room Reservation Application (React): This is the user interface where users log in, view room availability, and make reservations.
2. API Application (Spring Boot): Serves as the backend, providing RESTful APIs for user, room, and reservation management. It handles requests from the Room Reservation Application.
3. Database (MySQL): Stores critical data, including user profiles, team information, room details, and reservation records.

The flow of data begins with users accessing the system through the React-based interface, which communicates with the API for all operations, with the database storing and retrieving persistent data.

Level 3: Component Diagram

A Component diagram zooms into an individual container, showing the components inside it.


Room reservation application
[Container: REACT]
User interface for login, room overview, reservation creation
Makes API calls
[JSON/HTTP]
ReservationsController
[Component: Spring Controller]
Allows users to make reservations.
RoomController
[Component: Spring Controller]
Allows admin to manage Rooms.
TableController
[Component: Spring Controller]
Allows admin to manage Tables.
TeamController
[Component: Spring Controller]
Allows admin to manage Teams; and allows user team managers to manage their Team.
UserController
[Component: Spring Controller]
Allows admin to manage Users.
Uses
ReservationService
[Component: Spring Service]
Allows users to make reservations.
RoomService
[Component: Spring Service]
Allows admin to manage Rooms.
TableService
[Component: Spring Service]
Allows admin to manage Tables.
TeamService
[Component: Spring Service]
Allows admin to manage Teams; and allows user team managers to manage their Team.
UserService
[Component: Spring Service]
Allows admin to manage Users.
Uses
ReservationRepository
[Component: Spring Repository]
Reads from and writes to database.
RoomRepository
[Component: Spring Repository]
Reads from and writes to database.
TableRepository
[Component: Spring Repository]
Reads from and writes to database.
TeamRepository
[Component: Spring Repository]
Reads from and writes to database.
UserRepository
[Component: Spring Repository]
Reads from and writes to database.
REST API
[Container]
Reads from and writes to
[SQL]
Database
[Container: MySQL]
Stores user, room and reservation data
Legend
Container
Component
[Components] Office Reservation System
The component diagram of the office reservation system

The component diagram breaks down the API application into smaller, focused components, showing their responsibilities and interactions:

1. Controllers: These handle incoming HTTP requests:
    a. ReservationsController: Manages reservations.
    b. RoomController: Manages rooms.
    c. TableController: Manages tables.
    d. TeamController: Manages teams.
    e. UserController: Manages users.
2. Services: These components implement business logic:
    a. ReservationService: Processes reservation-related operations.
    b. RoomService: Manages room-related tasks.
    c. TableService: Handles table-related logic.
    d. TeamService: Manages teams and roles.
    e. UserService: Administers user operations.
3. Repositories: Provide data access to and from the database:

a. ReservationRepository, RoomRepository, TableRepository, TeamRepository, UserRepository.

The database remains the core data store, connected to repositories for seamless data persistence and retrieval. The REST API serves as the backend, implementing business logic and database interactions to provide services for the Room Reservation Application (frontend).