Ionic

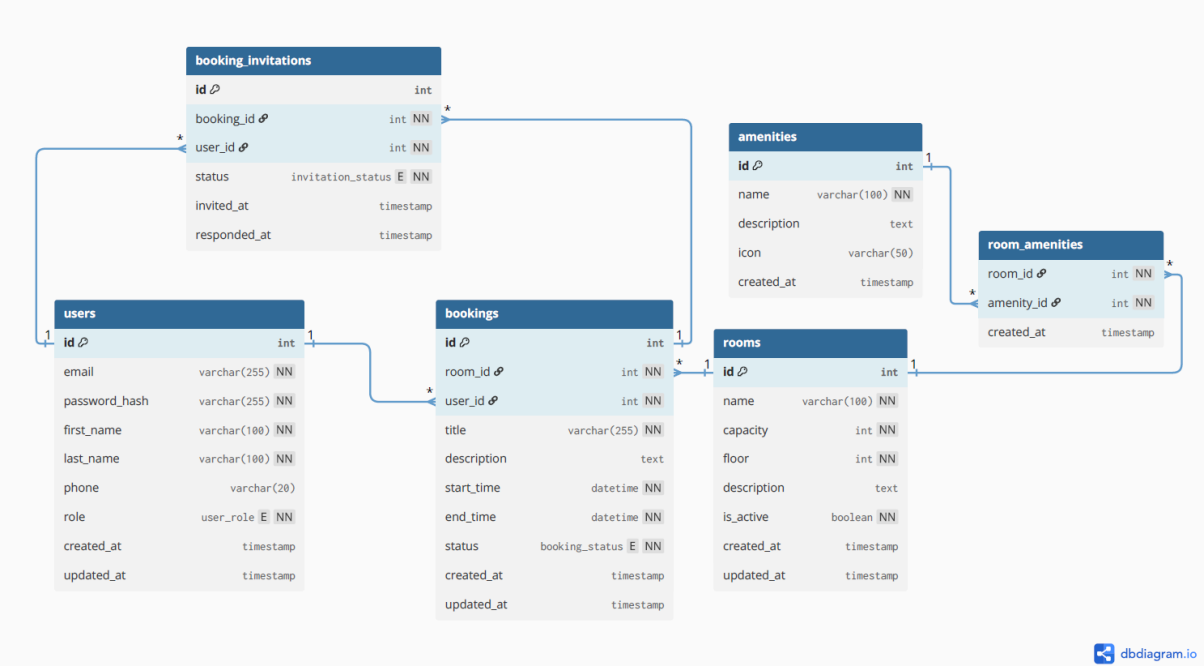
# Technical Assessment

ERD, Design Document and AI Usage Report

Kaylen Cairns
11-16-2025

# Contents

# ERD




```
Enum user_role {
  user
  admin
}


Enum booking_status {
  pending
  confirmed
  cancelled
  completed
}


Enum invitation_status {
  pending
  accepted
  declined
}
```

# Design Documents

## - Explain your database design choices and normalization decisions

Ensured that there was no many to many relationships by introducing room_amenities which helps prevent data duplication.

## - Describe your approach to your logic (database constraints, application logic, locks, etc.)

Added primary keys to prevent duplication, added foreign keys to ensure referential integrity and added Enum types to enforce valid status values.

Preventing double booking by adding triggers on when bookings are created that ensure that business logic passes before an insert can take place.

## - How does your API handle concurrent booking requests?

Next-Key locking is enabled to stop the 2nd transaction from continuing while transaction one is working, when trying to work on the same data.

## - Discuss how your solution would scale (caching strategies, read replicas, partitioning)

Would be able to add read scaling which would allow all select statements to read the data from more than one replica of the primary database.

Could potentially partition the database on time-based partitioning, splitting data between different years, quarters or months, to speed up finding the right data. Older data that might be accessed less could also be on slower or cheaper storage as they won't be accessed as often.

Could add caching for all the rooms that are available along with their amenities as they wont change that often. Room availability could also be cached for future dates, as users will often be checking multiple rooms and this would help speed it up, while reducing database load. Could cache the user's session data, reducing the data that the JWT would need to carry.

## - What would you do differently with more time?

1. Add the business logic that one user can not be in multiple meetings at the same time.
2. Allow one to see which users were invited to the same booking that you were invited to or created.
3. Allow admin users to manage other users' profiles, such as temporary passwords/ password changes.
4. Allow "forgot passwords" for users that forgot their passwords.

# AI usage report

## - How did you use AI tools during this assessment?

I used ChatGPT to help me with designing the database.

I then used copilot to help me during coding within vs code, I would give it a prompt on the backend to update some of my APIs and then I would ask it to give me a prompt that I could give to my frontend team (a separate vs code also using copilot) to fix the way the frontend interacted with the backend, and what recently changed and what the frontend needs to change.

I attempted to use the GitHub copilot through giving it an issue, to create tests for all the APIs that I have. This did not work out as well as I had hoped (would create the tests alongside the APIs next time).

## - What prompts/questions were most effective?

The most effective prompts I had was when I had the frontend and backend teams (two different vs code copilots) communicate with each other where I asked it to create a prompt to tell the frontend or backed team what has been updated and what they need to do to continue working with that API.

## - What did you validate or modify from AI suggestions?

The AI struggled with naming data in a descriptive way for the user and thus needed changes to allow the user to better understand the data they are receiving.

I also had to make sure that the AI wasn't logging any information of the user on the frontend, which it was doing when confirming the information it received from the backend.

## - Did AI help with database design? How did you verify correctness?

AI helped design the database, by verifying my work. I started by designing the database I thought I needed to complete the program, then ChatGPT (the AI I used to verify) gave me some tips and examples on what to change to get to where the database is now.