

Wits-Parking-System

Haste Park

Ta Seider | Database Fundamentals | 31 May 2024

**# Wits Parking System Database Report**

**---**

## Introduction

The Wits Parking System application is designed to manage parking spot reservations within the Wits University campus. It includes a login screen, which implies the existence of a user entity containing users' details and metadata about their interactions with the app. The app will prompt users to enter their credentials for authentication and authorization purposes. This report outlines the design and considerations for the database underlying the Wits Parking System.

### User Entity

The user entity stores essential details such as usernames and passwords. However, questions arise about additional information, such as names, surnames, and email addresses, which could help in personalizing user experience and for validation purposes. Google account integration is considered for ease of validation and to ensure the authenticity of users but in this case we are looking for Wits students and/or staff members so using google login in would not be in line with our target audience/customers.

### App Customization

We can try this.

The app does not allow users to customize their preferences after their first login. These preferences must be stored in the database, not on the user’s device, to maintain consistency across multiple devices. The format for storing preferences should be efficient, such as a JSON string or a separate table linking user IDs to their preferences.

## Parking Spot Management

### Abstract Spot Entity

When a user searches for a parking spot, the app interacts with a parking spot entity that stores details about available spots. The database needs to store various attributes of parking spots, including the spot ID, (location,) and availability status.

### Spot Reservation Process

Users can choose to reserve a parking spot. The reservation entity will need to store information such as the user ID, spot ID, reservation time, and duration. Verification of the reservation could be handled through user credentials and real-time updates to the spot's availability.

### Verification and Validation

To ensure the integrity of the system, verification of parking spots and users is crucial. The system should differentiate between verified and unverified spots. Verification could involve checking if the spot is within the university's database and ensuring that only verified users can reserve spots. By verified we mean that only students can park in the students parking lot and only staff members can park in the staff parking lots.

## Database Design

### Entities and Relationships

1. \*\*User Entity\*\*: Stores user details including user ID, username, password, email, and preferences.

2. \*\*Spot Entity\*\*: Stores details about parking spots including spot ID, location, and status.

3. \*\*Reservation Entity\*\*: Stores reservation details including reservation ID, user ID, spot ID, start time, and end time.

### User Entity

The user entity encompasses all users of the application. Attributes include:

- User ID (Primary Key)

- Username

- Password

- Email

- Preferences (stored as a JSON string or in a separate preferences table)

### Spot Entity

The spot entity includes:

- Spot ID (Primary Key)

- Location (building, level, spot number)

- Status (available, reserved, occupied)

### Reservation Entity

The reservation entity includes:

- Reservation ID (Primary Key)

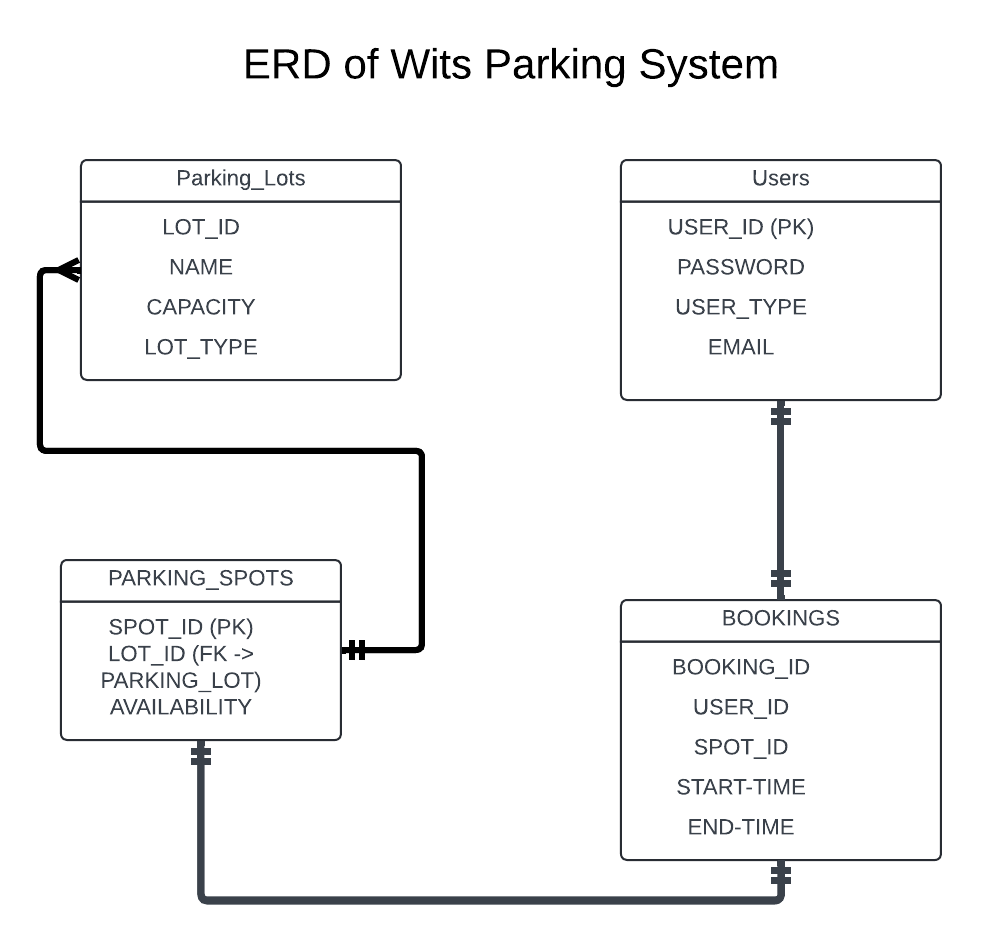
- User ID (Foreign Key to User Entity)

- Spot ID (Foreign Key to Spot Entity)

- Start Time

- End Time

### Example ER Diagram



## Business Rules

1. \*\*User Authentication and Authorization\*\*: Users must provide valid credentials to access the app. Google account integration is not used for validation.

2. \*\*Spot Availability\*\*: Only available spots can be reserved. Once reserved, the spot status changes to "reserved".

3. \*\*Reservation Time Management\*\*: Reservations are time-bound. Users must specify the start and end times for their reservations.

4. \*\*Spot Verification\*\*: Only verified parking spots can be listed in the system. By listed

5. \*\*User Customization\*\*: Users can customize their app settings, which are stored in the database and applied across devices.

### Limitations

- The app cannot verify if a parking transaction (e.g., payment) is successful.

- The app cannot confirm if the actual parking spot matches the database entry due to real-time changes and physical conditions.

## Conclusion

The Wits Parking System database design addresses essential aspects of user management, spot reservation, and data integrity. By ensuring proper entity relationships and adhering to business rules, the system aims to provide a seamless parking experience within the Wits University campus.