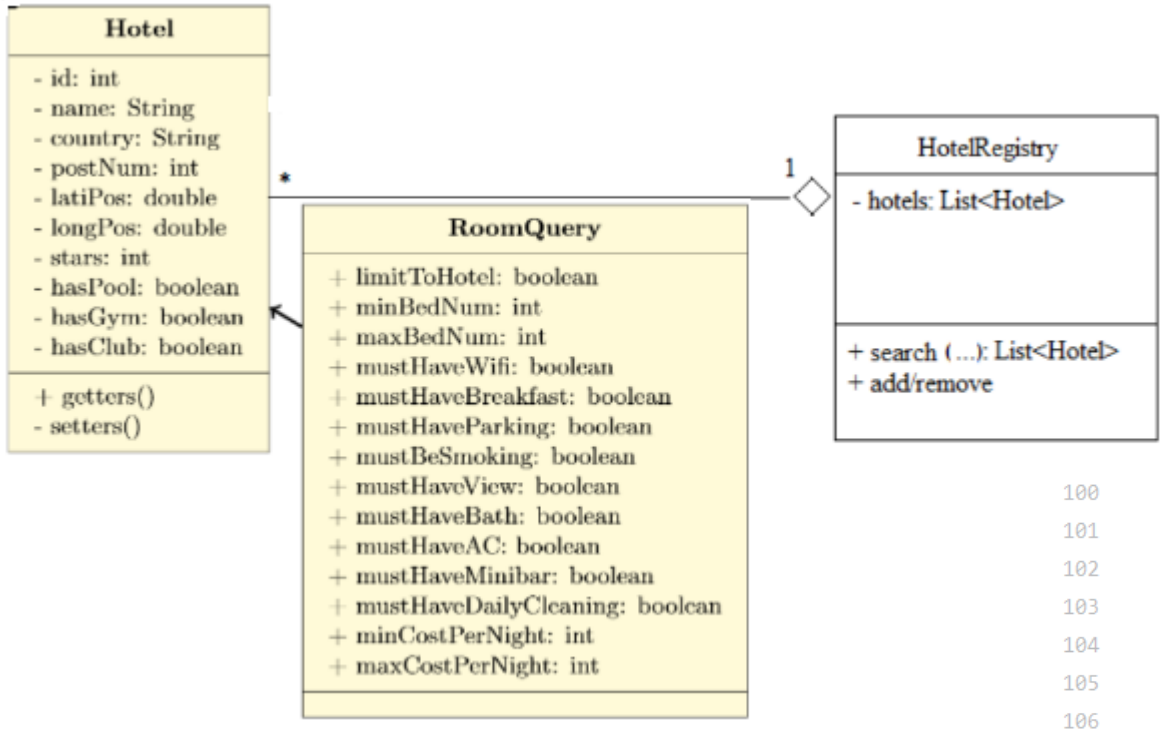
System architecture: Highlights



RoomQuery class that contains the parameters

JDBC code fragment

```
100     public static Hotel getHotel(int id) {
101         try {
102             Class.forName("org.sqlite.JDBC");
103         } catch(ClassNotFoundException e) {
104             return null;
105         }
106         Connection conn = null;
107         try {
108             conn = DriverManager.getConnection("jdbc:sqlite:hotelData.db");
109             PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM HotelRegistry WHERE id = ?");
110             pstmt.setInt(1, id);
111             ResultSet r = pstmt.executeQuery();
112             if(!r.next()) {
113                 return null;
114             }
```

```
124             Hotel result = new Hotel(_id, _name, _postNum, _latiPos, _longPos, _stars,
125                     _hasPool, _hasGym, _hasClub);
```

A look back

- Things that went well:
  - Implementation and coding of classes (UML effective blueprint)
  - Filtering and sorting using SQL

- Things that could have gone smoother:
  - Communication between teams in cluster (bigger challenge than first thought)
    - In the future we would communicate expectations better between groups
  - JDBC/SQLite and JUnit took time because of inexperience