File - /Users/notre_ally/CS/PerScholas_IDEA/maintenancerecords/src/main/java/com/ryanmuehe/maintenancerecords/model/Item.java

```java
package com.ryanmuehe.maintenancerecords.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import java.time.LocalDateTime;

@Entity // DB table of the same name
@EntityListeners(AuditingEntityListener.class) // Enables automatic DateTime updates
@Getter // lombok package defines all getter methods
@Setter // lombok package defines all setter methods

// Item model defines the model fields
// that JPA/Hibernate generates into an item DB table
public class Item {

    @Id // specifies the primary key
    @Column(nullable = false, updatable = false) // specifies a required, final field
    @GeneratedValue(strategy = GenerationType.IDENTITY) // auto-increment
    private Long id;

    @Column (nullable = false)
    private String name;

    @Column(columnDefinition = "longtext") // allows for longblob size in DB
    private String description;

    @ManyToOne(fetch = FetchType.LAZY) // does not retrieve associated records automatically
    private User owner;

    @CreatedDate // stores the time created
    @Column(nullable = false, updatable = false)
    private LocalDateTime dateCreated;

    @LastModifiedDate // stores the last updated time
    @Column(nullable = false)
    private LocalDateTime lastUpdated;

}
```

File - /Users/notre_ally/CS/PerScholas_IDEA/maintenancerecords/src/main/java/com/ryanmuehe/maintenancerecords/model/ItemUsage.java

```java
package com.ryanmuehe.maintenancerecords.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;
import java.time.LocalDateTime;

@Entity // DB table of the same name
@EntityListeners(AuditingEntityListener.class) // Enables automatic DateTime updates
@Getter // lombok package defines all getter methods
@Setter // lombok package defines all setter methods

// User model defines the model fields
// that JPA/Hibernate generates into a User DB table
public class ItemUsage {

    @Id // specifies the primary key
    @Column(nullable = false, updatable = false) // specifies a required, final field
    @GeneratedValue(strategy = GenerationType.IDENTITY) // auto-increment
    private Long id;

    @Column
    private String usedBy; // name of person or organization using the item

    @Column
    private LocalDateTime usageStart; // when the usage begins

    @Column
    private LocalDateTime usageEnd; // when the usage ends

    // Many ItemUsages can be associated with one item
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "item_id", nullable = false) // associated by item id
    private Item item; // the item being used

    @CreatedDate // stores the time created
    @Column(nullable = false, updatable = false) // required final field
    private LocalDateTime dateCreated;

    @LastModifiedDate // stores the last updated time
    @Column(nullable = false)
    private LocalDateTime lastUpdated;
}
```

File - /Users/notre_ally/CS/PerScholas_IDEA/maintenancerecords/src/main/java/com/ryanmuehe/maintenancerecords/model/MaintenanceRecord.java

```java
package com.ryanmuehe.maintenancerecords.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;
import java.time.LocalDateTime;

@Entity // DB table of the same name
@EntityListeners(AuditingEntityListener.class) // Enables automatic DateTime updates
@Getter // lombok package defines all getter methods
@Setter // lombok package defines all setter methods

// User model defines the model fields
// that JPA/Hibernate generates into a User DB table
public class MaintenanceRecord {

    @Id // specifies the primary key
    @Column(nullable = false, updatable = false) // specifies a required, final field
    @GeneratedValue(strategy = GenerationType.IDENTITY) // auto-increment
    private Long id;

    @Column(columnDefinition = "longtext") // allows for longblob size in DB
    private String description; // record of the state of the item

    // Many Maintenance Records can be associated with one item
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "item_id", nullable = false) // associated by item id
    private Item item; // the item that gets the maintenance record

    @CreatedDate // stores the time created
    @Column(nullable = false, updatable = false) // required final field
    private LocalDateTime dateCreated;

    @LastModifiedDate // stores the last updated time
    @Column(nullable = false)
    private LocalDateTime lastUpdated;
}
```

```java
package com.ryanmuehe.maintenancerecords.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
import java.util.HashSet;
import java.util.Set;

@Entity // DB table of the same name
@Getter // lombok package defines all getter methods
@Setter // lombok package defines all setter methods

// Role model defines the model fields
// that JPA/Hibernate generates into a Role DB table
public class Role {

    @Id // specifies the primary key
    @Column(nullable = false, updatable = false) // specifies a required, final field
    @GeneratedValue(strategy = GenerationType.IDENTITY) // auto-increment
    private Long id;


    @Column (nullable = false, updatable = false)
    private String name; // role type such as ROLE_USER, ROLE_ADMIN


    @OneToMany(mappedBy = "role") // relationship to a 'role' source in RoleAssignment
    private Set<RoleAssignment> roleAssignments = new HashSet<>(); // Each role may have many
roleAssignments
}
```

```java
package com.ryanmuehe.maintenancerecords.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;
import java.time.LocalDateTime;


@Entity // DB table of the same name
@EntityListeners(AuditingEntityListener.class) // Enables automatic DateTime updates
@Getter // lombok package defines all getter methods
@Setter // lombok package defines all setter methods

// RoleAssignment model defines the model fields
// that JPA/Hibernate generates into a RoleAssignment DB table
public class RoleAssignment {

    @Id // specifies the primary key
    @Column(nullable = false, updatable = false) // specifies a required, final field
    @GeneratedValue(strategy = GenerationType.IDENTITY) // auto-increment
    private Long id;

    // Many RoleAssignments can be associated with one user
    @ManyToOne(fetch = FetchType.LAZY) // does not retrieve associated records automatically
    @JoinColumn(name = "user_id", nullable = false) // must have an associated user id
    private User user; // the user assigned to the role

    // Many RoleAssignments can be associated with one role
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "role_id", nullable = false) // must have an associated role id
    private Role role; // the role assigned to the user

    // Many RoleAssignments can be associated with one item
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "item_id") // associated item id may be null for global roles
    private Item item; // the item assigned to the role and user

    @Column
    private LocalDateTime startTime; // when the RoleAssignment begins

    @Column
    private LocalDateTime endTime; // when the RoleAssignment ends

    @CreatedDate // stores the time created
    @Column(nullable = false, updatable = false) // required final field
    private LocalDateTime dateCreated;

    @LastModifiedDate // stores the last updated time
    @Column(nullable = false)
    private LocalDateTime lastUpdated;

}
```

File - /Users/notre_ally/CS/PerScholas_IDEA/maintenancerecords/src/main/java/com/ryanmuehe/maintenancerecords/model/User.java

```java
package com.ryanmuehe.maintenancerecords.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import java.time.LocalDateTime;
import java.time.OffsetDateTime;
import java.util.HashSet;
import java.util.Set;

@Entity // DB table of the same name
@Table(name = "\"user\"")
@EntityListeners(AuditingEntityListener.class) // Enables automatic DateTime updates
@Getter // lombok package defines all getter methods
@Setter // lombok package defines all setter methods

// User model defines the model fields
// that JPA/Hibernate generates into a User DB table
public class User {

    @Id // specifies the primary key
    @Column(nullable = false, updatable = false) // specifies a required, final field
    @GeneratedValue(strategy = GenerationType.IDENTITY) // auto-increment
    private Long id;

    @Column(nullable = false, unique = true) // required, unique field
    private String email;

    @Column(nullable = false)
    private String username;

    @Column(nullable = false)
    private String password;

    @CreatedDate // stores the time created
    @Column(nullable = false, updatable = false) // required final field
    // ToDO: to change to OffsetDateTime need to customize DateTimeProvider
    //    private OffsetDateTime dateCreated;
    // Update time types in Item, ItemUsage, MaintenanceRecord, RoleAssignment, User
    private LocalDateTime dateCreated;


    @LastModifiedDate // stores the last updated time
    @Column(nullable = false)
    private LocalDateTime lastUpdated;

    @OneToMany(mappedBy = "user") // relationship to a 'user' source in RoleAssignment
    private Set<RoleAssignment> roleAssignments = new HashSet<>(); // each user may have many
roleAssignments

}
```