Attendance Management System

Models

UserRoleType Enumeration:

```
type UserRoleType string

const (

      Principal UserRoleType = "principal"

      Teacher   UserRoleType = "teacher"

      Student   UserRoleType = "student"

)
```

- **Purpose**: Enumerates different user roles in the system.
- **Constants**:
    - **Principal**: Represents the role of a principal.
    - **Teacher**: Represents the role of a teacher.
    - **Student**: Represents the role of a student.

User Model:

```
type User struct {

      tableName struct{}          `sql:"user"`

      ID        string            `json:"id" pg:",pk"`

      Username  string            `json:"username"`

      Password  string            `json:"password"`

      Role      UserRoleType      `json:"role"` // "principal", "teacher", or
"student"
```

```
	ClassMap  []ClassMappingUser `json:"-" pg:"rel:has-many"`

}
```

- **Purpose**: Represents generic user information.
- **Fields**:
  - **ID**: Unique identifier for each user.
  - **Username**: User's username for authentication.
  - **Password**: User's password for authentication.
  - **Role**: Enumerated type (UserRoleType) representing the role of the user (principal, teacher, or student).
  - **ClassMap**: List of ClassMappingUser records indicating the mapping between the user and multiple classes. The "has-many" relationship denotes that one user can be associated with several class mappings, allowing for the representation of the user's connection to various classes.

Class Model:

```
type Class struct {

	tableName struct{} `sql:"class"`

	ClassID   int      `json:"class_id" pg:",pk"`

	ClassName string   `json:"class_name"`

}
```

- **Purpose**: Represents information about classes.
- **Fields**:
  - **ClassID**: Unique identifier for each class.
  - **ClassName**: Name for the class.

ClassMappingUser Model:

```
type ClassMappingUser struct {

      tableName struct{} `sql:"classmapuser"`

      ID        int      `json:"id" pg:",pk"`

      UserID    string   `json:"user_id" pg:",fk"`

      ClassID   int      `json:"class_id" pg:",fk"`

      Class     Class    `json:"-" pg:"rel:has-one"`

}
```

- **Purpose**: Establishes the relationship between users and classes.
- **Fields**:
  - **ID**: Unique identifier for each mapping record.
  - **UserID**: Foreign key referencing the User model, indicating the user associated with the mapping.
  - **ClassID**: Foreign key referencing the Class model, indicating the class associated with the mapping.
  - **Class**: Represents the related Class information.

Attendance Model:

```
type Attendance struct {

      tableName struct{}   `sql:"attendance"`

      ID    int     `json:"id" pg:",pk"`

      UserID  string    `json:"user_id" pg:",fk"`

      Day    int     `json:"day"`

      Month   int     `json:"month"`

      Year   int     `json:"year"`
      PunchMap []PunchInOut `json:"-" pg:"rel:has-many"`

}
```

- **Purpose**: Represents individual attendance records for users.
- **Fields**:
  - **ID**: Unique identifier for each attendance record.
  - **UserID**: Foreign key referencing the User model, indicating the user associated with the attendance record.
  - **Day, Month, Year**: Date components representing the day of attendance.
  - **PunchMap**: List of PunchInOut records representing punch-in and punch-out instances associated with this attendance.

PunchInOut Model:

```
type PunchInOut struct {

      tableName    struct{}                 `sql:"punchinout"`

      ID           int                      `json:"id" pg:",pk"`

      AttendanceID int                      `json:"attendance_id" pg:",fk"`

      UserID       string                   `json:"user_id"`

      PunchIn      string                   `json:"punch_in" `

      PunchOut     string                   `json:"punch_out" `

      ClassMap     []ClassMappingAttendance `json:"-"`

}
```

- **Purpose**: Represents individual punch-in or punch-out records for attendance.
- **Fields**:

        - **ID:** Unique identifier for each punch-in/out record.
        - **AttendanceID**: Foreign key referencing the Attendance model, indicating the attendance record associated with the punch-in/out.
        - **UserID**: Represents the user associated with the punch-in/out.
        - **PunchIn, PunchOut**: Timestamps indicating the times of punch-in and punch-out.
        - **ClassMap**: List of ClassMappingAttendance records representing the mapping between this punch-in/out event and classes.

ClassMappingAttendance Model:

```
type ClassMappingAttendance struct {

      tableName struct{} `sql:"classmapattendance"`

      ID        int      `json:"id" pg:",pk"`

      PunchID   int      `json:"punch_id" pg:",fk"`

      ClassID   int      `json:"class_id" pg:",fk"`

      Class     Class    `json:"-" pg:"rel:has-one"`

}
```

    - **Purpose**: Establishes the relationship between punch-in/out records and classes.
    - **Fields**:
        - **ID**: Unique identifier for each mapping record.
        - **PunchID**: Foreign key referencing the PunchInOut model, indicating the punch-in/out event associated with the mapping.
        - **ClassID**: Foreign key referencing the Class model, indicating the class associated with the mapping.
        - **Class**: Represents the related Class information.

JWT Model:

```
type JWT struct {

      PrivateKey []byte

      PublicKey  []byte

}
```

- **Purpose**: Represents the structure for handling JSON Web Tokens (JWT) for authentication and authorization.
- **Fields**:
  - **PrivateKey**: Private key used for JWT generation and decoding.
  - **PublicKey**: Public key used for JWT verification.

Routes

Principal Routes:

1. **/principal/addStudent:** POST endpoint to add a new student to the system.
2. **/principal/addTeacher:** POST endpoint to add a new teacher to the system.
3. **/principal/teacherAttendance:** GET endpoint to retrieve attendance records for teachers.
4. **/principal/Students:** GET endpoint to retrieve a list of students managed by the principal.
5. **/principal/Teachers:** GET endpoint to retrieve a list of teachers managed by the principal.

Teacher Routes:

1. **/teacher/punchIn:** POST endpoint for teachers to record a punch-in event.
2. **/teacher/punchOut:** POST endpoint for teachers to record a punch-out event.
3. **/teacher/attendance:** GET endpoint to retrieve attendance records for the teacher.
4. **/teacher/classAttendance:** GET endpoint to retrieve class-wise attendance records for the teacher.

Student Routes:

1. **/student/punchIn:** POST endpoint for students to record a punch-in event.
2. **/student/punchOut:** POST endpoint for students to record a punch-out event.
3. **/student/attendance:** GET endpoint to retrieve attendance records for the student.

Login Routes:

1. **/login:** GET and POST endpoint for user authentication.
2. **/logout:** GET endpoint for user logout.

Dashboard Routes:

1. **/dashboard/principal:** GET endpoint to access the principal's dashboard.
2. **/dashboard/teacher:** GET endpoint to access the teacher's dashboard.
3. **/dashboard/student:** GET endpoint to access the student's dashboard.