Student Management System

Core Functionalities

🔐 Authentication

- Admin login (email/password)
- Session management

👩‍🎓 Student Management

- CRUD operations for students/subjects/marks
- Age/name validation

📚 Academic Operations

- Subject creation
- Grade assignment (0–100)
- Student–subject relationships

Architecture

3-Tier Design

🖥️ Presentation Layer (Swing Views)

- LoginView, StudentView, MarkView

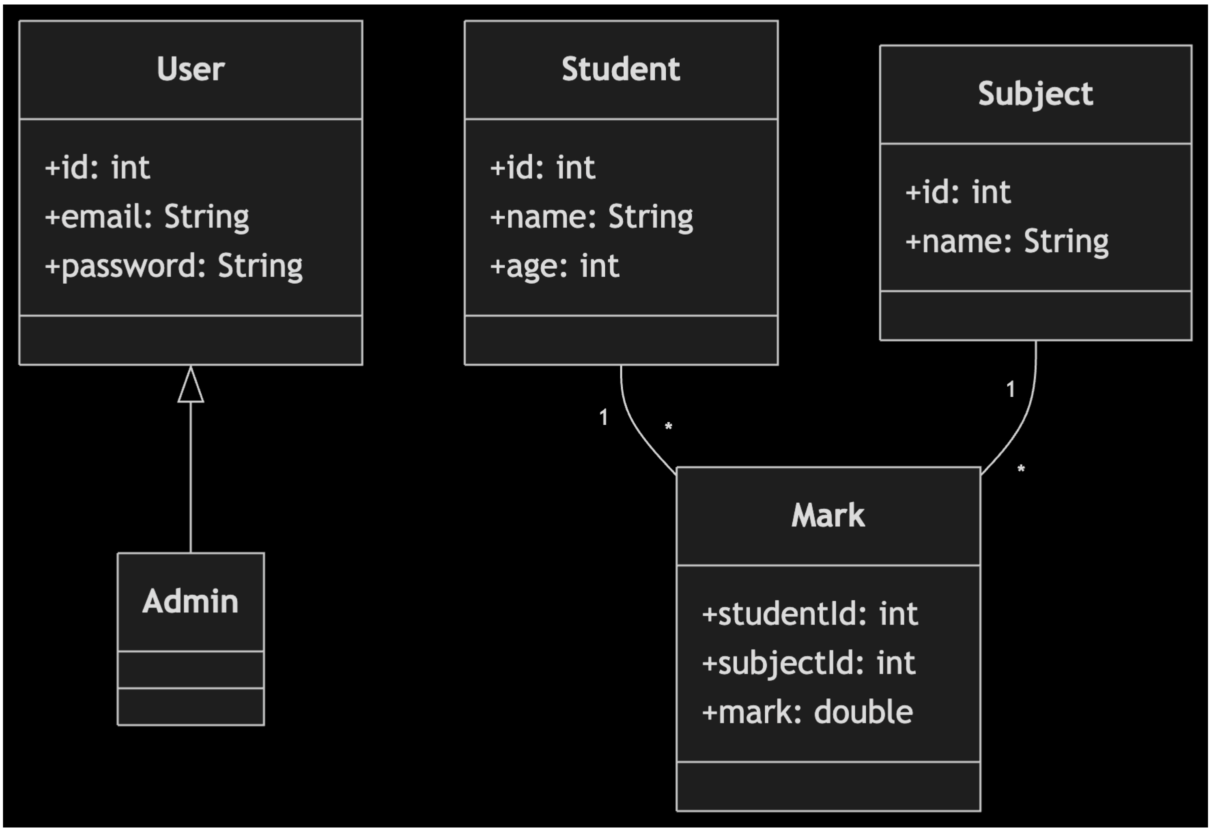
🧠 Business Logic (Controllers)

- AuthController, StudentController
- Input validation logic

🗃️ Data Access (DAO Pattern)

- `DatabaseConnection` (Singleton)
- `UserDAO`, `StudentDAO` interfaces

Class Diagram:



Database Schema

Tables

- `users(id, email, password)`
- `students(id, name, age)`
- `subjects(id, name)`
- `student_subject(student_id, subject_id, mark)`

Relationships

- Many-to-many: Students ↔ Subjects (via `student_subject`)

Design Patterns

✅ Singleton

- For database connection

✅ DAO

- Abstraction layer: `StudentDAO` ↔ `StudentDAOImpl`

✅ MVC

- Flow: View → Controller → Model

✅ Observer

- Swing event listeners for buttons

Slide 8: Demo Highlights

🔑 Login Credentials:

`admin@example.com` / `admin123`

👨‍🎓 Student Management

- Add / Edit / Delete students and subjects. Marks are still buggy

📝 Grade Assignment

- Link students to subjects
- Assign marks (0–100)