UML Class Diagram

Schedule
- Name: String
- scheduleIntro: String
- startDate: Date
- endDate: Date
- sceneOnScedule: ArrayList<Scene>

- Schedule(String, ArrayList<Scene>)
- getCityOnSchedule(): ArrayList<Scene>
- getName(): String
- getScheduleIntro(): String
- getEndDate(): Date
- getStartDate(): Date
- addScene(Scene): void
- sortSceneByTime(): void
- getSceneNum(): int
- getScene(int): Scene
- toString(): String

User
- userID: String
- password: String
- name: String
- age: int
- sex: int
- email: String
- phone: String
- address: String
- country: String

- User(String, String, String, int, int, String, String, String, String)
- getUserID(): String
- getPassword(): String
- getName(): String
- getAge(): int
- getSex(): int
- getEmail(): String
- getPhone(): String
- getAddress(): String
- getCountry(): String
- toString(): String
- print(): void

IdPasswordCombo
- id: String
- password: String

- IdPasswordCombo(String, String)
- getId(): String
- getPassword(): String

Controller
- socket: Socket
- sendout: ObjectOutputStream
- receive: ObjectInputStream

- Controller(Socket)
- close(): void
- run(): void

Listener
- serversocket: ServerSocket

- Controller(Socket)
- close(): void
- run(): void

Scene
- sceneID: int
- name: String
- city: City
- introduction: String
- score: double
- date: int
- time: int

- Scene(int, String, City, String, double, int, int)
- getSceneID(): int
- getName(): String
- getCity(): City
- getIntroduction(): String
- getScore(): double
- getDate(): int
- getTime(): int
- toString(): String

ScheduleUserIdCombo

- ScheduleUserIdCombo(Schedule, String)
- getSchedule(): Schedule
- getUserId(): String

SceneComparator

- SceneComparator()
- compare(Scene, Scene): int

<< Interface >> DBConnection
- conn: Connection
- statement: Statement

- DatabaseAccess()
- insert(Object): boolean
- load(Object): Object
- update(Object): boolean
- close(): void

TableCreator
- conn: Connection
- statement: Statement

- TableCreator()
- link_sql(): void
- createAutoTable(String): void
- insertSample(): void
- closeConn(): void

City
- cityID: int
- name: String
- location: String
- introduction: String
- score: double

- City(int, String, String, String, double)
- getCityID: int
- getName(): String
- getLocation(): String
- getIntroduction(): String
- getScore(): double
- toString(): String
- toView(): String
- print(): void

CityTableAccess
- conn: Connection
- statement: Statement

- CityTableAccess()
- insert(City): boolean
- load(City): City
- update(City): boolean
- close(): void

SceneTableAccess
- conn: Connection
- statement: Statement

- SceneTableAccess()
- insert(Scene): boolean
- load(Scene): Scene
- update(Scene): boolean
- close(): void

ScheduleTableAccess
- conn: Connection
- statement: Statement

- ScheduleTableAccess()
- insert(Schedule): boolean
- load(Schedule): Schedule
- update(Schedule): boolean
- close(): void

UserTableAccess
- conn: Connection
- statement: Statement

- UserTableAccess()
- insert(User): boolean
- load(User): User
- update(User): boolean
- close(): void

Relationships:
- Schedule — Scene : Composition (1...*)
- Scene — City : Composition (1)
- Controller --> User : Use
- Controller --> IdPasswordCombo : Use
- Controller --> DBConnection : Use
- Controller --> TableCreator : Use
- Controller --> ScheduleUserIdCombo : Use
- Listener --> Controller : Use
- Schedule --> ScheduleUserIdCombo : Use
- Scene --> SceneComparator : Use
- TableCreator --> DBConnection : Use
- User --> ScheduleUserIdCombo : Use
- CityTableAccess, SceneTableAccess, ScheduleTableAccess, UserTableAccess --|> DBConnection : Extends