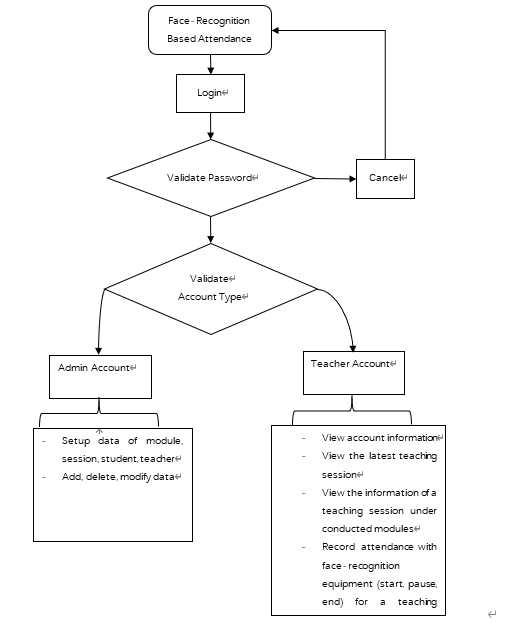
**High level Design**

**Introduction:** The below transfers the system into software structure and data structure based on SRS to build up the logical model of the whole system. It analyzes the division of modules and their connections, and it provides a detailed description of the design of each module as well as data structure.

1. **Background Information**
2. Running environment
3. Development tool
4. **Terminology definition: TBC**
5. **Overall Design**
6. Basic process of the system



1. Structure

1. Manual operation:

* Forget password: user needs to follow instructions shown on help page.

1. **Interface Design**

文档的接口部分应该很简单的，一般分：外部接口和内部接口三个部分；用户接口只要简述用户操作和反馈结果等；外部接口简述硬件输入输出、等；内部接口简述模块间传值、数据传递等即可。

1. User interface

GUI user interface is implemented in this system. After entering the main interface, user can click the corresponding window to enter the corresponding interface or activate other corresponding operations.

1. External interface

* Attendance recording: Computer camera
* Print recording: computer print
* Save recording: computer file system
* Recording: output excel table

1. Internal interface

* Interface between internal system and database: Use database link API from Python package and SQL command to operate on the local database.
* Interface between modules:

1. GUI vs back-end

Implement MVC framework: Each interface page is made up by the extracted public control items and is regarded as View. The control items have their own controller enabling them to perform the specific functions.

Based on required data, models are created and bind with corresponding views. Such as the “Upcoming Event View”.

Meanwhile, each interface page also has its senior controller to manage their component controls, and perform special functions such as page jumping.

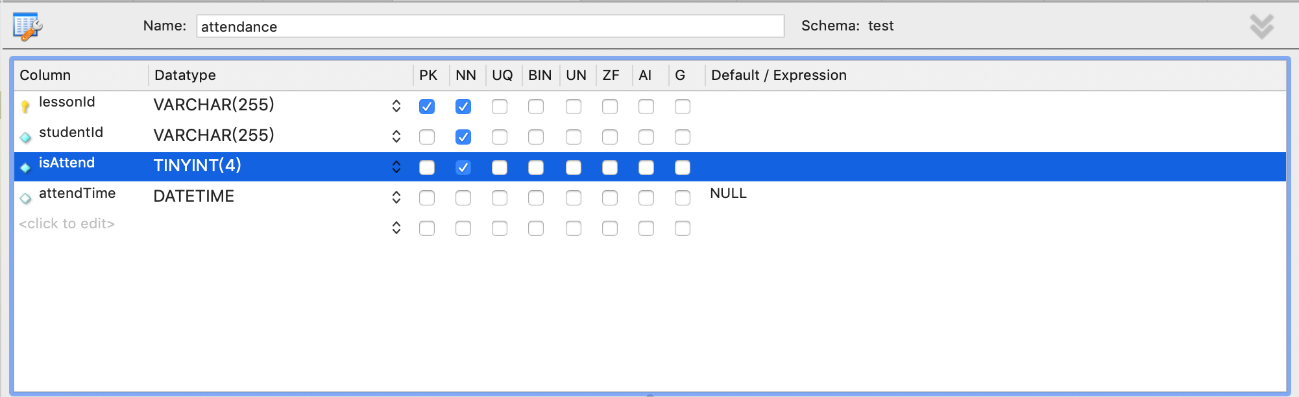
1. Modules

The corporations between modules are realized by the communication between each interface pages. The front-end can send out signals to controllers, and then the controllers will perform required functions.

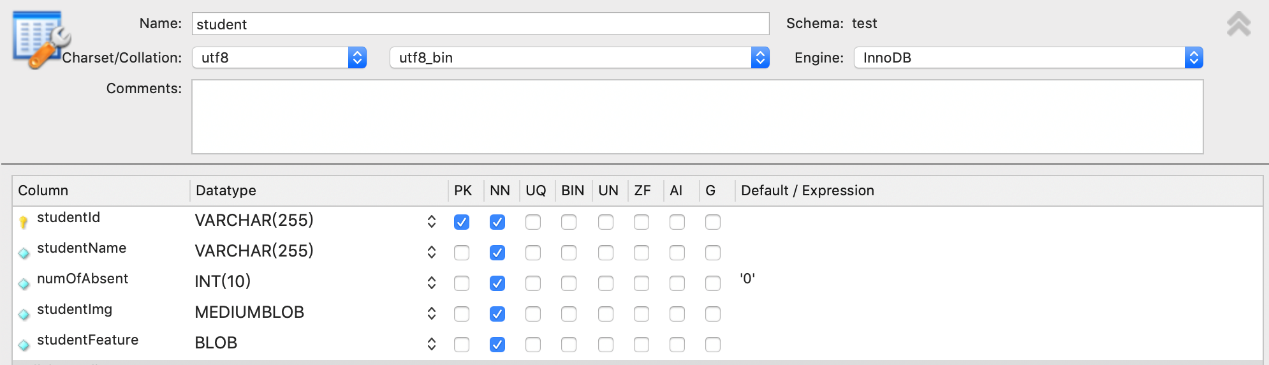
1. **Running Design: TBC**
2. Running composition:
3. Running control:
4. Running time:
5. **System Data Structure Design**
6. Logical structure
7. Attendance data sheet

Attendance (lessonID, stundeID, isAttended, attendTime)

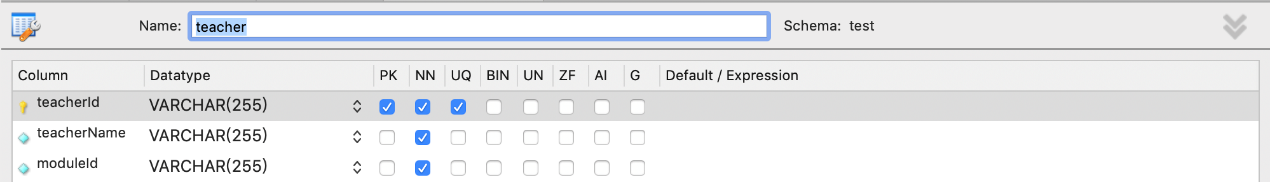
1. Student data sheet
2. Teacher data sheet
3. Module data sheet
4. Lesson (teaching session) data sheet
5. Login data sheet
6. Physical structure
7. Attendance table



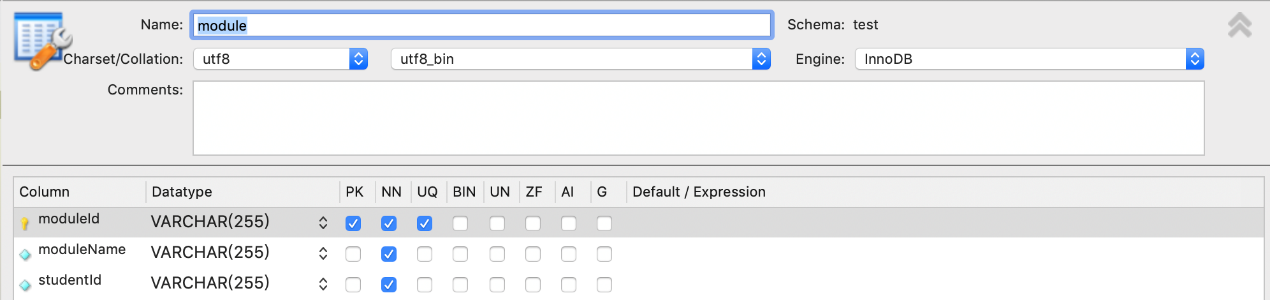
1. Student table



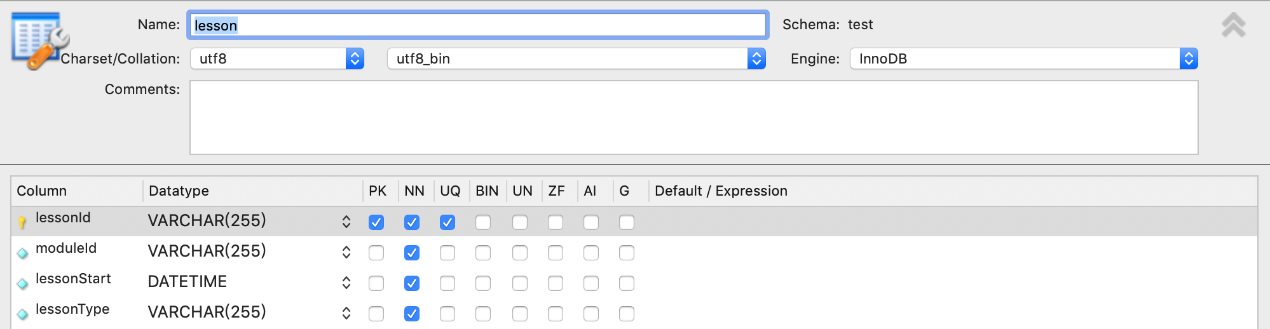
1. Teacher table



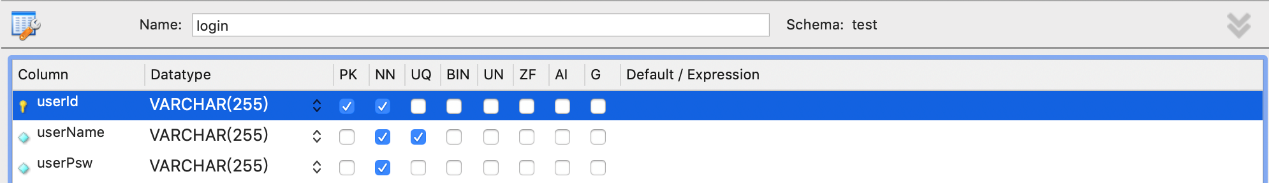
1. Module table



1. Lesson table



1. Login table



1. Data structure and program

The implemented database is relational, hence the standard SQL command is used to operate on the data structures. The system designed a class that is responsible to visit the database, which improves the maintainability and extensibility of system.

1. **Security Design: TBC**
2. **Error Handle Design**
3. Error

|  |  |  |
| --- | --- | --- |
| Error Type | Error Term | Error Cause |
| Database Error | Link error | Connection timeout |
| Database error | Wrong command |
| System custom error | Authority error | Admin authority setting error |
| Input error | Wrong input/ blank inpt |
| Program running error | Program crush | Response timeout |

1. Remedial

|  |  |
| --- | --- |
| Condition | Remedial Operation |
| User Error | Provide error hint windows |
| System Crush | Restart program or hint user to check the operation dialog |

1. System maintenance

* Database: Regularly check database consistency with user operation.
* Set up operation log and regularly check it.

Face-Recognition Based Attendance System

Login

Validate Password

Cancel

Admin Account

Teacher Account

* Setup data of module, session, student, teacher
* Add, delete, modify data

* View account information
* View the latest teaching session
* View the information of a teaching session under conducted modules
* Record attendance with face-recognition equipment (start, pause, end) for a teaching session
* Modify/ delete??
* Logout

Validate

Account Type

