School of Computer Science

University of Nottingham Ningbo China (UNNC)

**COMP2043. GRP Final Group Report**

Smart Class Attendance Taking System

Group 17

Supervisor: Zheng LU

**Name Username**

Boya Wang slybw2

Mingchen Li scyml3

Yiming LI scyyl3

Guohao Yu scygy1

Hongyi Zhu scyhz1

2019.12.12

**Contents**

1. Project Introduction
2. Project Development Plan
   1. Staff allocation
   2. Development plan timetable
3. Project Background Information
   1. Survey of similar product
   2. Market research and user survey
   3. Technical research
   4. Development tool
4. Software Requirement Specification
   1. Diagram
   2. User requirement
   3. System requirement
   4. User interface requirement
5. UI Design
6. System Design
   1. High level design
   2. Database design Specification
   3. Data requirement specification
7. System Implementation
   1. Front-end implementation
   2. Interface implementation
   3. Login System implementation
   4. Teacher System implementation
   5. Admin System implementation
   6. Database implementation
8. Achievement
   1. Common function achievement
   2. UI achievement
   3. face recognition function archievement
   4. Teacher functional achievement
   5. Admin functional achievement
9. System Test
   1. Test plan
   2. Test case
   3. Test record
   4. Test analysis
10. Reflective Summary
    1. System analysis
    2. Design
       1. UI design
       2. Front-end
       3. Interface design
       4. Function design
       5. Database design
    3. Implementation
       1. UI
       2. Front-end
       3. Interface
       4. Function
       5. Database
    4. Test
    5. Management issue
11. References
12. Appendix
    1. Staff Allocation Table
    2. UI design documents
    3. Coding and comment standards
    4. Test records
    5. Formal meeting minutes
    6. User Manual
13. **Project Introduction**

This project is to develop a face-recognition based student attendance recording system. There are two main problems in this project to solve: to implement face recognition and to develop a fully functional system implementation for management.

1. **Project Development Plan**
   1. **Staff allocation**

|  |  |  |
| --- | --- | --- |
| **Group** | **Staff** | **Front-end** |
| **Back-end** | **Mingchen Li** | **Face-recognition** |
| **Yiming Li** | **Teacher functionality development** |
| **Front-end** | **Boya Wang** | **UI building, Set up interfaces between front-end and back end** |
| **Guohao Yu** | **Page jumping function, software test** |
| **Hongyi Zhu** | **QSS, software test** |

* 1. **Development timetable**

**手机屏幕截图

描述已自动生成**

**手机屏幕截图

描述已自动生成**

**手机屏幕截图

描述已自动生成**

1. **Project Background Information**
   1. **Survey of similar products**

There are a significant number of facial recognition attendance taking systems used in companies and universities nowadays. For example, DingTalk, Baidu Brain, Jibble, Integrated…

The following includes two examples of the existing systems. DingTalk represents the facial recognition attendance taking system used domestically. Jibble is a representation in overseas.

1. ***DingTalk:***

*DingTalk*, a professional office application for small and medium-sized enterprises (SMES), is an integration of multiple functionalities. Facial recognition attendance system is applied on *DingTalk* whose version is over 4.0 by using Ding Talk M2 Smart Receptionist and *DingTalk* C1 Smart Communication Center, which can identify whether picture or an alive person and recognize it even under extreme lighting condition. The whole system is able to be set up in 4 steps, downloading *DingTalk*, connecting facility by blue tooth, connecting WIFI and bonding with specific group, which is extremely user friendly. The system can identify multiple people quickly when they expose to the camera. Once the person passes the facial recognition, the attendance sheet will be updated by the system. However, facial recognition attendance system is only a small part of the whole *DingTalk*.

By 2018, *DingTalk* had have more than 100 million users and became one of the world’s largest professional communication and management application in China.(DingTalk, 2019)

1. ***Jibble*:**

*Jibble* is an application mainly for record the attendance with sign in time and sign out time if passed the facial recognition. Then the system offers timesheet views and report to the administrator. By using this application, the administrator can have direct access to employee statistics and improve the productivity. *Jibble* system is based on mobile devices like iPad, so it can only recognize one person at one time using the front camera.

However, *Jibble* doesn’t have multiple functionalities like *DingTalk*. Since the system works on individual device, cannot identify multiple person does not affect a lot. (Jibble, 2019)

* 1. **Market research and user survey**

We did our market research by interviewing a few teachers in our university. The question we asked include the problem of current attendance sheet, the expected features of our system and the time requirement of taking attendance.According to the result, our teacher in university are willing to use a software which can take place of attendance sheet. The problems of current attendance sheet, the expected features of this software and the time requirement are shown below.

1. Problems of current attendance sheet:
2. Inconvenient for teacher
3. Need to print out an attendance sheet and store the attendance sheet into computer manually
4. Student can sign or each other, since there is no verification
5. Expected features:
6. GUI should be user-friendly.
7. Easy and convenient to do the attendance recording.
8. Offer the list of students who has be absent for more than three times
9. Searching by lecture name and show details
10. Comparation of attendance rate of two lectures in same module
11. Comparation of overall attendance rate of two modules
12. Make sure the safety of data
13. Students can only sign in for themselves
14. Time requirement:
15. If record the attendance after class, should make sure it will finish in ten minutes
16. The system should be stable and accurate to record as soon as possible

In conclusion, the system should be easy to use for teachers and reliable enough to reduce the time of recording attendance. In general, by using this system, student cannot sign for each other anymore and the attendance sheet can be stored automatically.

**3.3 Technical research**

1. Platforms:
2. Anaconda is used to finish our project since we were asked to finish it with Python and it is the easiest one to use. In addition, it is free. The version of anaconda is over 3 and it contains 3 bask components, which are Anaconda Navigator, Jupyter Notebook and Spyder.
3. OpenCV packages are imported for picture processing and it is free to use. That is why we choose it.
4. There are 3 suitable packages for UI design. However, the UI made by Tkinter is too simple. PyQt is faster and easier than Kivy. As a result, PyQt is used to design UI.
5. Tools

Camera is the solely tool that the project requires.

1. Technologies

Face recognition system and school management system is needed.

1. Algorithms

Machine learning algorithms are used. The program should learn from the existing student pictures stores in database and recognize whether the student is in the relative department.

**3.4 Development tools**

1. **Programming language: Python**

Python has the largest free third-party library for artificial intelligence implementation, especially for face recognition. Compared to other languages, it can support the development of the key function in this project – student recognition better. Meanwhile, Python is compatible with our chosen programming paradigm: Object-Oriented Programming. Hence the main developing programming language is Python.

1. **Operating system: cross-platform**

For Python is a cross-platform programming language and this project will provide an executable file for software installing, it supports the implementation on different platforms.

1. **Hardware:**
2. **Network:** Campus network. Also, the servers and user desktops running this software should be able to access the campus network to set up service.
3. **User computer:** A computer equipped with a camera, with a minimum of 100MB available space on disk for software implementation.
4. **Remote server:** One server in Artificial Intelligence Lab, which can set up MySQL database. It should provide a minimum of 2GB available space on disk for software implementation, larger space is recommended.
5. **Software:**

All the developing tools support cross-platform developing. Most of tools are open source and can be used for this non-profitable project according to their licenses. If this project will be implemented for business use in the future, more details about license will be checked.

* **Project management:** Microsoft Office Word, Microsoft Office Excel

Word is used for normal text documenting such as meeting minutes. Excel provides easy implementation of project scheduling diagram such as Gant chart.

* **Resource control and document management:** Git and Github

Git supports distributed version control. It enables the group members to share resources and work on a project simultaneously. It takes good control of the work merging and version recording.

Project Team Github url: <https://github.com/Yiming-Li666/GRP-P19>

* **Reference management software:** EndNote

EndNote provides interfaces with the sites of academic resources and document editor. It enables the developing group to manage references and generate documentation with high efficiency.

* **UML drawing software:** Papyrus

Papyrus provides full functions for UML diagram design, which can be used for project analysis.

* **Editor:** Notepad++ (Windows Only)

Easy to convert text from Linux (MacOS) to Windows and vice-versa.

* **Development environment: Anaconda**

Anaconda provides the management of multiple languages developing environments. Here is used for building the environment of python.

* **Integrated Development Environment (IDE): Pycharm, VScode**

Pycharm and VScode provides strong support for development in Python with functions such as refactoring, integrated test and the using of external tools.

* **Database: MySQL**

MySQL is a classic database management system, and our developers have rich experience with it.

* **GUI design: PowerPoint, Photoshop**

PowerPoint provides enough support for the wireframe design of UI. Photoshop enables the designers to achieve the ideal visual effects.

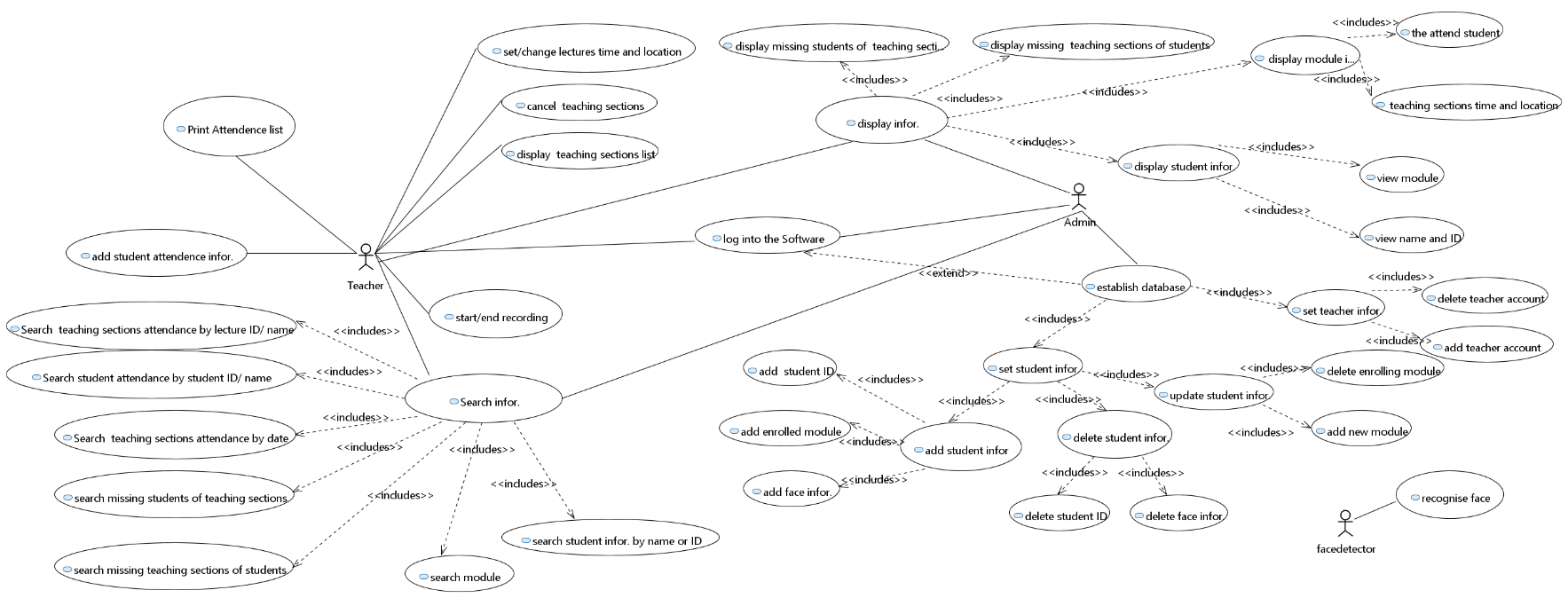
* **GUI implementation: PyQt5, Qt**

Qt is one of the main-stream GUI libraries, it supports cross-platform development, is of rich resources, and is easy to implement. It provides graphic interface design tool – Qt Designer.

PyQt5 enables the developers to use APIs in Qt with Python. It provides collections of modules which can fully support our GUI development. For example, it enables multimedia control, Bluetooth device connect, unit testing and better visual effects.

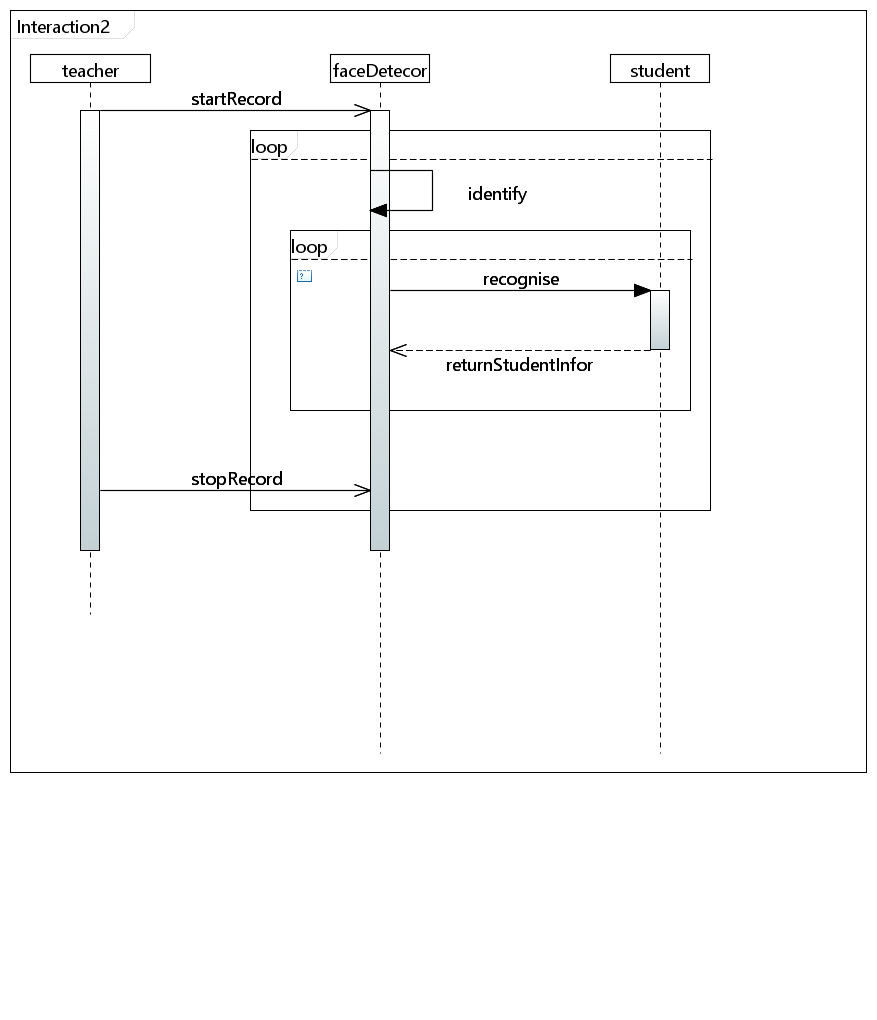
* **Open source library: OpenCV**

OpenCV is a highly optimized library with focus on real-time applications. Cross-Platform C++, Python and Java interfaces support Linux, MacOS, Windows, iOS, and Android.

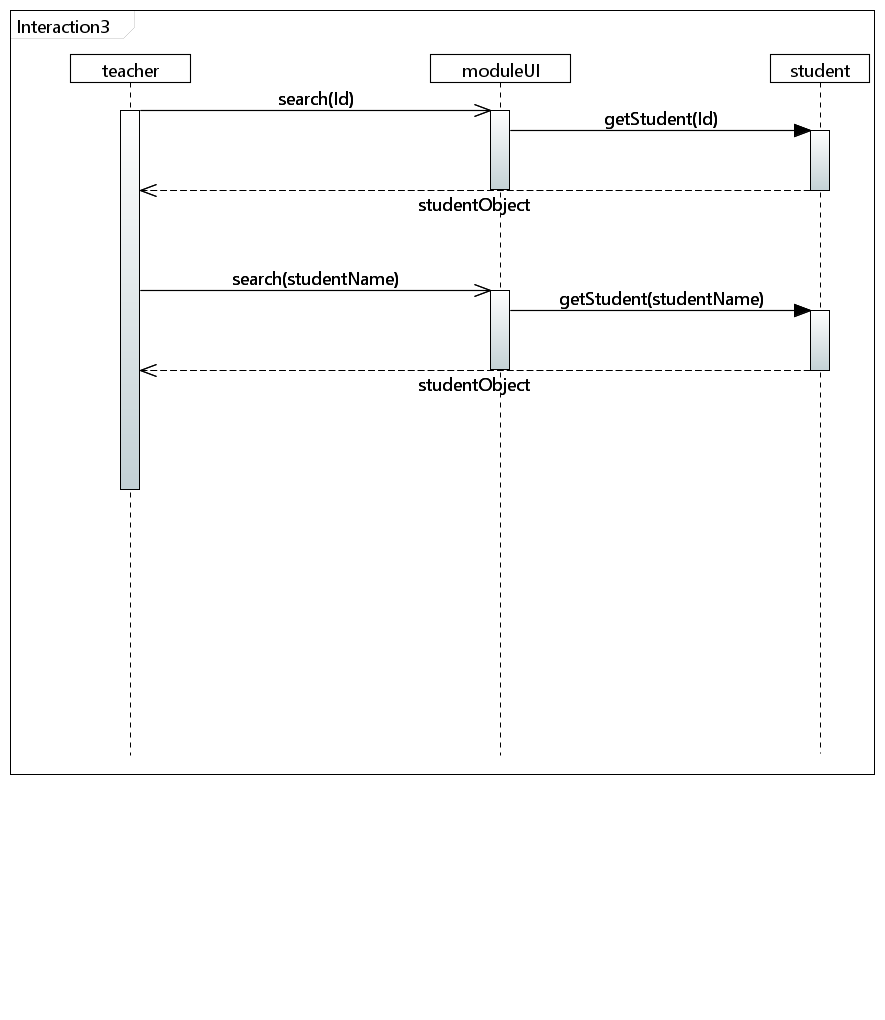
1. **Software Requirement Specification**
   1. **Diagrams**
2. **Use Case Diagram**

The identified users include teacher, administer and face detector. The function of face detector is to recognize face. The common functions of teacher and administer include system log-in, system log-out, searching and viewing attendance information with different categories. For their singular responsibilities, the teacher can edit the information of teaching session, start and stop attendance recording of a teaching session, and edit attendance information. The administer can establish and edit the database.

1. **Sequence Diagram**

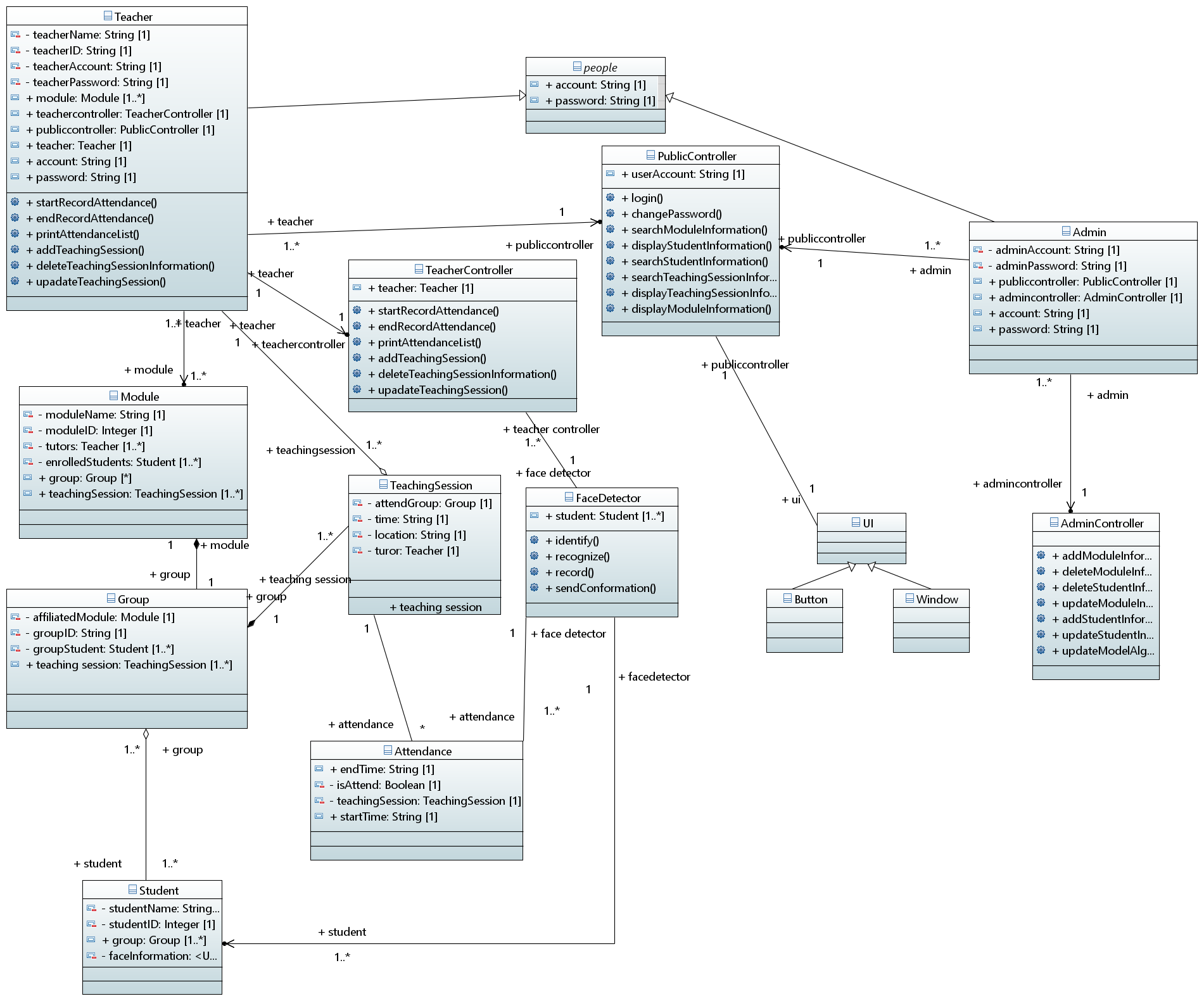
****

This is the sequence diagram of face recognition function of this system. The teacher will start recording firstly. Then the face detector will receive the signal and keep identifying a face until teacher stop recording. After a face is identified, the face detector will go to recognize the face to by conducting a loop of comparing the recognition information and face information of each student. The information of a successfully recognized student or an unfound signal will be returned to the face detector.

****

This is the sequence diagram of searching student function in teacher class. A student can be found by his id or name. The teacher can firstly try to input the student id in module page, and in this page the student with input specified id will be found. The found student object will be returned to module page, and then shown for teacher. Then the teacher may search another student by name, the process will be the same.

1. **Class Diagram**

****

Teacher class and Admin class contain the constructor of all the teacher or admin information. Both of them implements corresponding Controller class and PublicController class to do log in operation, recording operation, modify lecture information operation, modify students’ information operation, searching operation or print operation, because different teachers have different results when doing the same operation.

Teacher class and Admin class implement one same class which is PublicController class. It contains all the methods that teacher and admin can invoke. Meanwhile, TeacherController and AdminController respectively contains methods for only teacher or admin.

Also, Teacher class and Amin class both extend the People class, because both of them have accounts and passwords. The Student class does not have the similarity between these classes, so it does not implement the People class.

1. The Module class contains a constructor including all the module information, a teacher object and a group object.
2. The Group class contains a constructor including all the group information, a module object and a list of student objects.
3. The Student class contains. a constructor including all the student information and a list of group objects.
4. The TeachingSession class contains a constructor including all the lecture or lab information, a group object and a teacher object.

UI classes are not completely included in the class diagram because of the complexity of it will affect the concision and readability. It can be separated into multiple subclasses as different specific buttons and windows which links to FaceDetector class and all the controller classes.

* 1. **User requirement:**

1. **User characteristics:**

The main users include database administer, teachers, students and a face detector.

1. **Functional requirement:**

* The users can interact with the system through a graphic user interface.
* Common functions:

1. User login
2. User logout

* For face detector

1. Recognize student face image
2. Identified student face image
3. Signal identification success or failure

* For administer

1. Add/ update/ delete teacher accounts in user database
2. Import/ update/ delete student information in database
3. Import/ update/ delete information of teaching session in database
4. View database

* For teacher

1. View his own profile
2. Set/update name and time of teaching session
3. Start/continue/end class attendance recording
4. Search information based on teaching session/student name
5. View searched information
6. View and print attendance sheet of a module/teaching session/student
7. View immediate attendance condition for a class
8. Edit attendance information and add remark for it
9. **Un-functional requirement:**
10. Availability: the administer shall check and maintain the database or server, set software maintenance time and make sure the system can run during the work period.
11. Reliability: the system shall recognize and identify students correctly and prompt notice if the student cannot be identified.
12. Safety: the system shall prompt the name of identified student in case of mis-recognizing the student.
13. Security and Privacy: only users holding the role "admin" can access database and server.
14. User friendliness: the system shall provide user-friendly GUI, user-friendly operation button, immediate relative notice to the users if their operation is wrong.
15. Maintainability: make use of the structural software design method, OOP and quality management.
16. **User interface requirement:**
17. Develop Graphical User Interface
18. The UI should be able to support all operations noted in functional requirement
19. The UI should be clear to understand, and easy to use
20. The UI should be responsive in short time
21. The UI should be consistent on all interfacing screens
22. The UI should have a proper appearance which suits its users

**4.3 System Requirement:**

1. **Functional requirement:**

* **Common functions**

1. **User login**

Pre-condition: user logged out

Operation: enters user school ID and password, click confirm button.

Post-condition:

* If information is correct, change user state to log in. Show the main page.
* If not: stay user state, stay in the current page. Prompt a warning window asking the user to enter input again.

1. **User forget password**

Pre-condition: user logged out

Operation: click *forget password* link

Post-condition: Show help page with instruction text on it

1. **User logout**

Pre-condition: user logged in

Operation: click *teacher Info* link, and click logout option

Post-condition: change user state to logout, and show log-in page

* **For face detector**

1. **Recognize student face image**

Pre-condition: face detector is on and the user clicked start recording button

Operation: locate a face, read in and store a shot-cut of the face

Post-condition: a frame of face recognition stream is loaded in and stored in local file

1. **Identify student face image**

Pre-condition: a face has been recognized

Operation: compare face data and identify the corresponding student

Post-condition:

* If a student is identified, add his attendance information
* If the face matches no student, return false value

1. **Signal identification success or failure**

Pre-condition: a face has been processed with identification operation

Operation:

* If recorded successfully, show name and ID of identified student on page
* If recording failed, show error message on page

Post-condition: the user receive feedback message of operation

* **For teacher**

1. **View his own profile**

Precondition: the teacher account must be logged in

Operation: click on his own profile link

Postcondition: show the view of current teacher account’s information.

1. **Set/update lecture name and time**

Precondition: the teacher account must be logged in

Operation:

* choose a specific lecture to edit
* Input lecture name or lecture time
* click confirm button to save changes

Postcondition: lecture name and time is modified.

1. **Start and end class attendance recording**

Precondition: the teacher account must be logged in

Operation:

* choose a specific lecture to record attendance
* click the start button to start recording
* can be stop by clicking the stop button
* can continue recording by clicking the continue button after stop
* click end recording button to finish recording

Postcondition: show the view of attendance information of this lecture and the attendance information will be stored in the database.

1. **Search information based on lecture/student name**

Precondition: the teacher account must be logged in

Operation:

* type in the keyword as lecture or student name
* click search button to start searching

Postcondition: show the view of lecture or student list including the keyword that has been entered.

1. **View searched information**

Precondition: the teacher account must be logged in and has searched for a keyword of lecture/student name

Operation: click on a specific lecture

Postcondition: show the view of attendance information of this lecture.

1. **View attendance sheet of a module/teaching session/student**

Precondition: the teacher account must be logged in

Operation: click on the specific module/teaching session/student

Postcondition: show the view of attendance information of this a module/teaching session/student

1. **View immediate attendance condition for his class**

Precondition: the teacher account must be logged in and has started recording

Operation: /

Postcondition: show the percentage of attendance of current lecture on the right bottom of recording windows.

1. **Edit attendance information and add remark of students**

Precondition: the teacher account must be logged in

Operation:

* choose a specific student
* choose a specific lecture to edit this student’s attendance status
* change the status of current student’s attendance
* add notes to this change
* click confirm button to save changes

Postcondition: The student’s attendance status of this lecture has been changed.

1. **Un-functional requirements:**
2. Availability: the administer shall check and maintain the database or server. When the database or the server loses connection, the system must have a warning alert and try to reconnect.
3. Reliability: The expected number of users that the system can support is around 30,000. To achieve this goal, the database should be large enough to accommodate at least 30,000 students, teachers and lectures records. Furthermore, the system must ensure that it does not crash when using. Also, the system shall prompt notice if the student cannot be identified and it shall allow teachers to edit attendance information manually. When this is done, the reliability of the system can be guaranteed.
4. Safety: the system shall prompt the name of identified student in case of mis-recognizing the student. The system shall recognize and identify students correctly. A confirmation message should be alerted to let students to confirm their information.
5. Security and Privacy: All users’ information should be stored in a separated database on the cloud. Documents stored in a separate database can improve the security of the data, only users holding the role "admin" can access database and server and the privacy of students’ and teachers’ information can also be guaranteed.
6. User friendliness: the system shall provide user-friendly GUI, user-friendly searching operation, immediate alert windows will be notified to the users if their operation is wrong.
7. Maintainability: make use of the structural software design method, OOP and quality management. The administer should be able to add, modify and delete student or teacher or module or lecture information separately or all at once by reading from files. The system must allow administer to update the face identification and recognition algorithm.

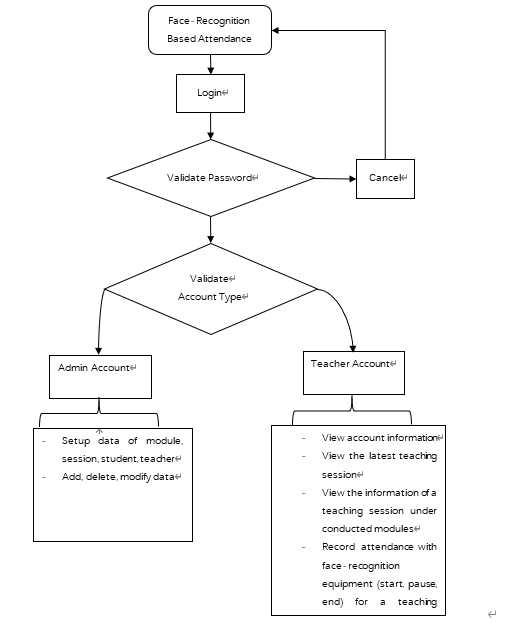
**4.4 User Interface Requirement:**

* **Basic design principle**

1. The sequence of user operations and composing of icons of UI should be consistent in view of the logic of user action.
2. The function composing of UI should be organized. For example, it should be delivered with rules of hierarchy and composition.
3. Implement UI with Qt and PyQT5 for better coupling of system functions and portability.
4. **UI Design**

The UML for initial design has been shown above. The initial design of whole UI system of our project is divided into user part and admin part. For more detail, please check the UI document attachment.

1. **System Design**
   1. **High level design**
2. **Overall Design**
3. Basic process of the system



1. Structure

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.

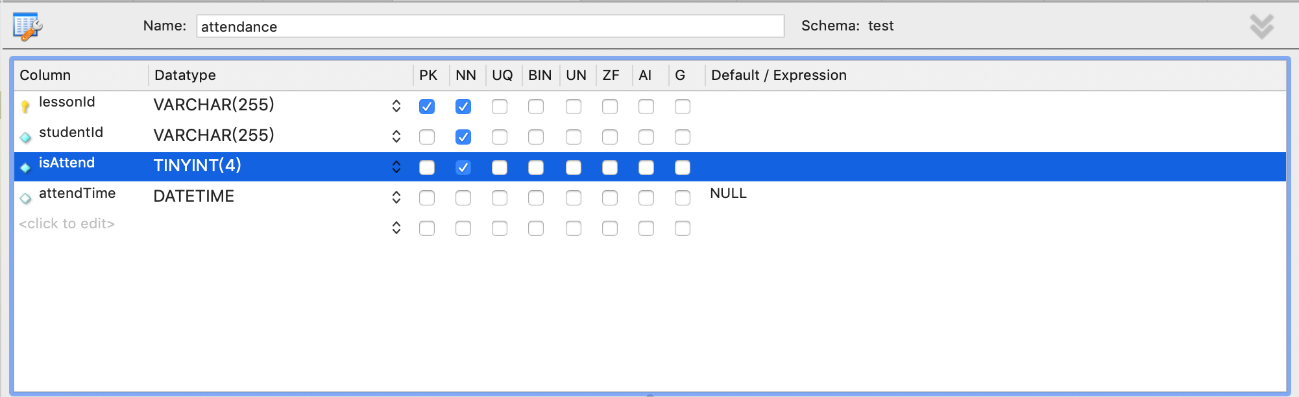
1. Modules

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.

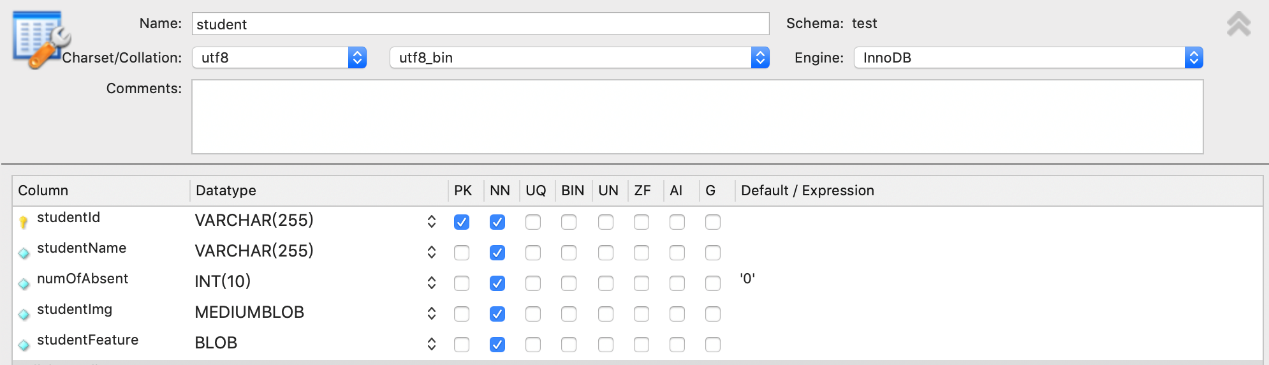
1. **Running Design: TBC**
2. Running composition:
3. Running control:
4. Running time:
5. **System Data Structure Design**
6. Logical structure
7. Attendance data sheet

Attendance (lessonID, stundeID, isAttended, attendTime)

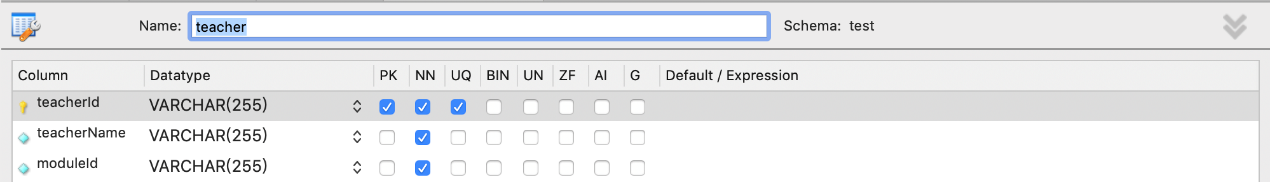
1. Student data sheet
2. Teacher data sheet
3. Module data sheet
4. Lesson (teaching session) data sheet
5. Login data sheet
6. Physical structure
7. Attendance table



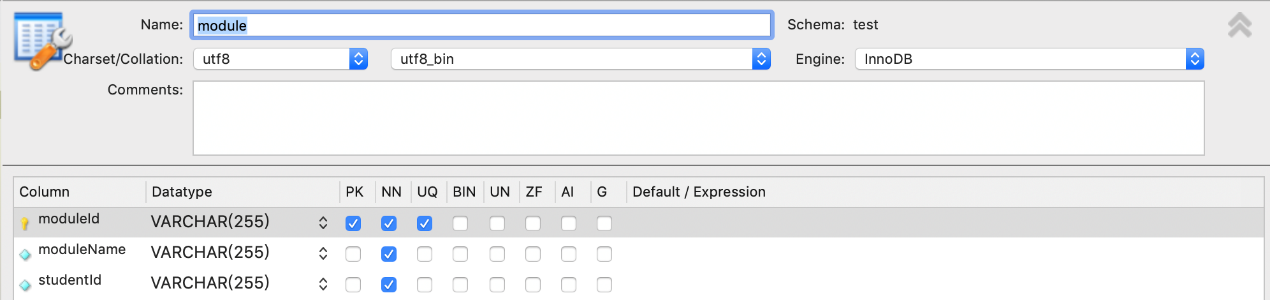
1. Student table



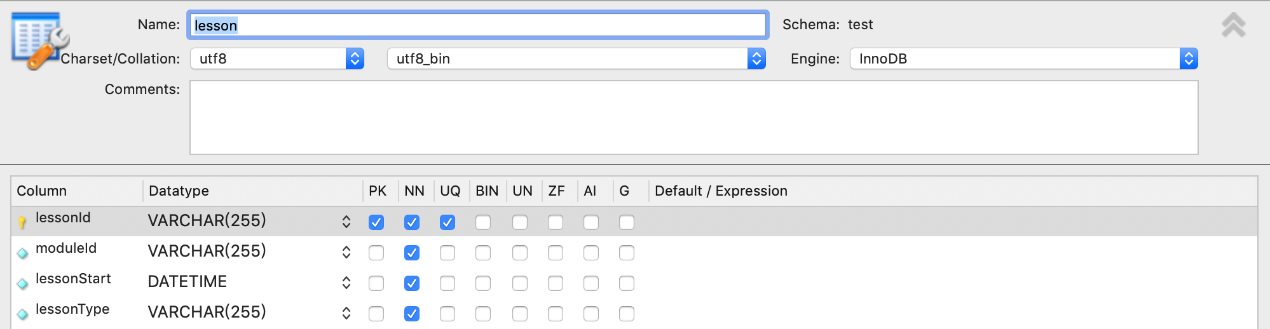
1. Teacher table



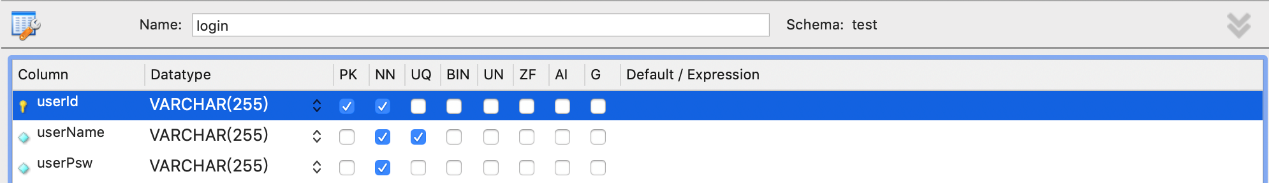
1. Module table



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.
  1. **Database design specification**
* **Login table**

It contains all the user account and password, including both teacher and admin. It will be connected to teacher table by Id.

* **Teacher table**

It contains teacher name, teacher Id and all the module Ids that teacher will teach. It will be connected to module table by module Id.

* **Module table**

It contains module name , module Id, faculty, school and year. It will be connected to lesson table by module Id.

* **Lesson table**

It contains module Id, lesson Id, student Id of whom should attend this session, venue, start time and session type. It is connected to student table using student Id.

* **Student table**

It contains student Id, student Name, student image, student face feature which is studied by algorithm and number of absence. It is connected to attendance table using student Id.

* **Attendance table**

It contains module Id, lesson Id, student Id, attendance status and attend time which can be null if absent. It is connected to student table using student Id and lesson table using lesson Id.

* 1. **Data requirement specification**

1. **System Implementation**
   1. **Front-end implementation**
   2. **Interface implementation**
   3. **Login System implementation**

Login system is a combination of user interface, functionality and back-end database usage. After starting the program, the login page will show up. Two textboxes are used for reading user Id and user password. The “login” button is connected to the login function. After clicking this button or pressing “return” button, the “enterMainPage” function will be invoked.

In the “enterMainPage” function, it will connect to the database, to check if the user Id is in the login table in database. If false is returned, the login requirement will be denied, and there will be an “Invalid username or password” warning. If true is returned, it will countinue to check if the user Id is in teacher table. If true is returned, the teacher operation interface will be shown and the module list fot that teacher will be loaded. If false is returned, the admin operation interface will be shown.

If the user forgets the password, a “forget password” botton is provided. If clicked, a hint text will be shown to contact admin.

The format of the password will be hidden as dots, which is applied by qss.

* 1. **Teacher System implementation**

1. **basic main window**

Basic main window is the base frame which will only be initialized once. The basic main window will remain the same while page changes the inside frames.

“Back to” button and “Home” button are link to page transfer function, which will control the page flow. A textbox is used to search attendance by students. “Search” button or “return” pressed, “searchStudent” function will be invoked.

In “searchStudent” function, database will be searched to return all the students’ name and Id contain the input string. A student search result page will show up and these data will be loaded into a list.

A “teacher information” button is provided for teacher to check his/her own profile. It is connected to “clickTeacherInfo” function to open a dialog, including teacher name, teacher Id and all the modules that the teacher teaches.

1. **module page**

In module page, module list of logged in teacher is loaded with module Ids using “setupModule” function. The listview is connected to “enterSessionPage” function, which will open the session page for chosen module.

1. **session page**

In session page, session list of chosen module is loaded with session Id, module Id, session start time, session venue and session type using “sortSessionList” function. The listview is sorted by session start time using a sort function.

A comboBox is connected to “sort” function to check the session start time and the real time. If the session start time is two hours earlier than real time, the session will be regarded as “past”. If the session start time is between two hours later than real time and one hour late than real time, the session will be regarded as “in progress”. If the session is after one hour than the real time, the session will be regarded as “future”. The session list will be sorted in these categories.

The listview is connected to”goSession” function, which will first compare the session start time with the real time. If the session is under “past” category, a recorded session page will show up. If the session is under “in process” or “future” category, a recording dialog will show up.

1. **recording dialog**

A text browser is loaded, including the chosen module Id and session Id.

A “Start Recording” button is provided to open the recording page.

1. **recorded session page**

In the recorded page, detailed session information will be loaded, including teacher name, module Id, session Id, teaching venue, attendance rate and Absent student names. Module Id and session Id are used as parameter to search database for teaching venue, attendance rate and Absent student names.

1. **recording session page**
2. **general student search**

General student search page will be shown if the teacher using search function on the basic main window. All the students’ name and Id contain the input string will be shown in a list.

The list view is connected to “goToStudent” function, to open the detailed student page.

1. **student information page**

The student information page contains detailed information for selected student. Student name and student Id are used as parameter to search database for the attendance information for that student.

Upper text browser includes student name, student Id, module name and attendance rate, total attendance rate and total absent number.

The list below loaded all the sessions that the student should take, with the attendance status.

Three comboBoxes in the middle have three categories, module name, session type and attendance status. Module name is loaded from database. “Sort” function is used to sort the session list. These categories can work simultaneously. Mainly use database operation to get sorted data, and then load the data to listview. The category of module name is sorted by the algorithm, by picking the module name in the list.

1. **Upcoming event frame**

Upcoming event frame will load upcoming session’s information when teacher chooses one module and enter the session page. On the module page, it will not show anything, because there may be to many upcoming session considering all the module.

Upcomgin events will loaded all the sessions in today and also do not in “past” category. If there is no upcoming session today, it will show that “Today has no upcoming event!”.

The upcoming event listview also connects to the “goSession” function, which allows the teacher to click on the upcoming session to start the session recording.

1. **print page**

Print page will be shown when clicking “print” on the heading control bar. All the session that the teacher has taught will be loaded into the listview.

There are two comboBox at the top to sort the listview by module Id and session type using “sort” function, which allows teacher to find the target session easily. Module Id is loaded from the database.

After clicking on a target session, save button is connect to “toSave” function, which get the module Id and session Id from the selected list. Then, it will connect to the database to get all the attendance information of that session and create a local file, named “moduleId+sessionId.csv”. All the students’ attendance information will be save into this file. If no session is selected, this function will not do anything.

* 1. **Admin System implementation**
  2. **Database implemetaion**

**图片包含 游戏机, 文字, 地图

描述已自动生成**

Six tables are created. Each tables have connection to other tables to form a complete structure. The physical structure of database is listed in the ER diagram.

Class “dbController” provides all the interface for the program to interact with database.

* User log in account correctness is search for the login table to check if the user Id and user password are correct.
* User account check is search for login table and teacher table to fing out if this is an admin account or a teacher account.
* Module Id shown in module page is gotton from teacher table by pass the teacher Id as parameter when logging in.
* Session detailed information is shown in session page is gotton from lesson table by passing the selected module Id as parameter.
* Student search function is search for student table to get all the student names which contains the user input string.
* Module and session attendance information for selected student is gotton from attendance table by passing the student Id as parameter.
* Session attendance information for selected session is gotton from attendance table by passing the module Id and session Id as parameters.
* Real time recording attendance reate is gotton from attendance table to initialise the starting status.
* Student record is passing module Id, lesson Id, student Id, attendance status as parameters to add the record into attendance table.
* If the session recording is over and the stop button is clicked, all the absent students will passed attendance status as false to add into attendance table.
* In the print page, all the sessions that the teacher taught are gotten from teacher table and lesson table by passing teacher Id as parameter.
* In print dialog , comboBox with module Id is gotten from teacher table by passing teacher Id as parameter.
* Sort function in print dialog uses selection operator in session table by passing module Id or session type as parameter.
* In student detailed attendance page, comboBox with module name is gotten from attendance table by passing teacher Id as parameter.
* Sort function in student search page uses selection operator in attendance table by passing module name or session type or attendance status as parameter.
* **Admin TBC**

1. **Achievement**
   1. **UI achievement**
   2. **common function achievement**

* **user requirement:**

1. User log in

Teacher and admin can log in using their own account. If the log in account is inside the login table, but not in the teacher table in database, it is regarded as an admin account. By entering the correct user Id and password, the admin page will be shown. If the account is a teacher account, by entering the correct user Id and password, the teacher page will be shown.

If the account is not correct, a warning will be shown to remind that the account or password is invalid, please enter again.

1. Forget password

If the user forgets the password, this program will not provide password changing service, since this functionality is with low priority. However, a warning will be sent to inform teacher to contact admin for help with admin’s phone number provided.

1. User log out

Teacher can log out by entering teacher’s profile page and clicking the log out button. The current window will be closed.

* 1. **face detector function achievement**
  2. **Teacher function achievement**
* **user requirement:**

1. View his own profile

A button is added at the right top of the main frame. After logged in, teacher can press this button. A dialog will be popped out to show his/her own name, Id, and all the module Id that he/she teaches.

1. Start/continue/end class attendance recording

In recording page, teachers can start recording attendance by clicking start button. The camera will work and record attendance. After starting recording, the start button will be converted to a pause button. After clicking the pause button, the camera will be frozen, the recording will be paused, and the button will be converted back to start. Teachers can click start button to continue recording. After the recording finishes, teacher can click stop button to stop recording. A page with this teaching session’s detailed information will be shown.

1. Search information based on student name

A search textbox is set on the top of the main frame. Teachers can use student name and student Id to search for particular student. If input is part of the student name or student Id, all the satisfied students, including student Id and student name, will be shown in the list for teacher to choose.

1. View searched information

After choosing one particular student in the student list of searching result, a page with detailed student information will be shown, including student name, student Id, all the modules’ attendance rate and total absent times.

1. View attendance sheet of a student

All the teaching session detail will also be listed on the page of one particular student. Teacher can use combo boxes to sort the attendance list by module Id, teaching session type or attendance state. Each record will contain module Id, teaching session Id, student Id, and attend time for that teaching session.

1. View attendance sheet of a teaching session

If the teaching session is passed, when click into this teaching session, a recorded page will be shown to demonstrate the detail information of that teaching session, including attendance rate and all the students’ names who didn’t attended the teaching session.

1. View immediate attendance rate for a class

A bar is added in the recording page to show the real time attendance rate. The accurate number will also be shown next to it.

If the lecture is recorded, a recorded page will be shown to demonstrate the detail information of that teaching session, including attendance rate and all the students’ names who didn’t attended the teaching session.

* **additional achievement:**

1. View all the module of this teacher account

All the module of this teacher account will be loaded after logged in. Teachers can select one of the modules to enter teaching session page.

1. View all the teaching session of selected module

All the sessions of selected module will be loaded after choosing one module. Teachers can select one of the sessions to enter recorded session page or recording page. Whether to enter recorded session page or recording page depends on the time. If the teaching session is passed, a recorded page will be shown will attendance detail information. If the teaching session is still in process or does not start yet, a recording dialog will be shown to allow teacher to start record.

1. View upcoming event

When entering session page, upcoming event list will be loaded. All the sessions that have been started less than two hours and all the sessions of today that have not started yet will be loaded. If there are no more teaching sessions today, it will show that today has no upcoming event.

1. Detailed student information display  
   After searching students, the student’s detailed information will be displayed by clicking on particular students. Detailed student information includes student name, student Id, every module that student takes and their attendance rate, total absence and a list of attendance status for every session.
2. Detailed recorded session display

After clicking on recorded sessions, the session’s detailed information will be displayed. Detailed session information includes module Id, session name, session start time, session venue, session attendance rate and the list of all the absent students’ name.

1. Sort teaching session list

When teacher entering teaching session page, all the teaching sessions of the selected module will be loaded and sorted by the time sequence. There is also a combo box to sort the session by future, in process or past status.

If “future” is selected, the teaching sessions that have been started less than two hours and the teaching sessions of today that have not started yet will be loaded.

If “in process” is selected, the teaching sessions that have been started less than two hours or will start in next one hour, will be loaded.

If “past” is selected, the teaching sessions that have been ended will be loaded. Teacher can click these sessions to see detailed information of attendance.

1. Sort student attendance list

After searching for one particular student, his/her attendance records will be shown in a list. Since this list may contain a lot of teaching sessions for different modules, sort function is added. There are three combo boxes to sort the results by module Id, session type and attendance status. These selection conditions will take effect simultaneously. With this function, teachers can easily figure out which sessions did this student miss.

1. Sort session attendance list

After entering the print page, all the sessions will be loaded in a list. Teachers can choose the session to save its attendance information to local file. Since a teacher will have more than one module and each module will have many teaching sessions, sort function is added. There are two combo boxes to sort the results by module Id and session type. These selection conditions will take effect simultaneously. With this function, teachers can easily find the target teaching session to save attendance information to local file.

1. Save session attendance information to local file

Teachers can save his/her teaching session attendance sheet to local files. The save option is added to the option bar on the top of the program. After clicked print, a page will be shown for teacher to choose which teaching session to save. After clicking save button, a local csv file will be created to store the attendance information of this teaching session. Including module Id, lesson Id, all the students’ Id, attendance state, and attended time if attended. The name of the file will be automatically set to ‘module name + session name.csv’.

* 1. **Admin functionality achievement**

1. **System Test**

**9.1 Test plan**

**9.2 Test case**

**9.3 Test record**

**9.4 Test analysis**

1. **Reflective Summary**
   1. **System analysis**

When we design our app, OO Design Principles is not been used.

There are no teacher, student and module classes. The app will always link to database. However, the cost is too much and it is hard for later maintenance. Hence, teacher, student and module classes are added and database is linked inside these classes.

* 1. **Function design**

1. Search function

There are 2 major part of our UI, one for teacher, the other for Admin. In teacher part, there is a search method that students can be searched by their name and id. It confused us where the function should be put. As a result, a general search function is designed, and users can choose what kind of data they want to search.

图片包含 屏幕截图

描述已自动生成

However, this general search is not needed because teachers can search what they need in the sub-pages. For example, when users attached to some specific module, there is a search function that they can search lectures.

图片包含 屏幕截图

描述已自动生成

As a result, the general search is canceled, and student search function is been added at the top of UI. Teachers can search students whenever they want.

1. Students can only be found by their ID when first time we design this function. However, this is unreasonable because students should be able to be found by their names as well. This error is already modified.
2. There should be a warning when teacher there are no searching result.
3. Attendance recording function

The teacher should be able to stop and restart the record function. A restart button is added. The related function is changed as well.

1. Reused functions

Although there are different users, the search function and view function they used is actually the same. In the end a public controller class is designed, and it stores all the functions that can be both used by teachers and admins.

* 1. **UI design**

1. There are 2 plans for Admin UI. The second one is too hard for us to achieve although it is more convenience to use. As a result, the first one is used.
2. UI location of function

When one specific student is gotten by teacher, teacher can modify some information of this student such as attendance information. A new page is added to show all student information and teacher can change them and add comments there. However, this is inconvenience because there is one additional function in the new page. So, this function is added at original page.

图片包含 屏幕截图

描述已自动生成

* 1. **Management issue**

1. Delay

Time plan is set to distribute the whole program into two semesters with a proper working load every week. However, there exists delay of the work, which means the work has not been finished within the decided time.

In this case, deadline is abided and there will be a punishment if any group member exceeded his/her time to finish the task.

1. Argument

Arguments always exists while making decisions. There has been one case happened when deciding which third party to use.

In this case, communication is needed within the group members and team leader should do this work. A rule is set that if anyone has any disagreements, he/she needs to list the advantage of his idea or the disadvantage of the current one.

3) Assignment

UI and code implementation are two main parts in this project. As the development tools of them have difference, we decided to separate our team into two groups, one for UI and one for functionality implementation. In this way, we can do both UI and functionality implementation simultaneously, which is more efficiency.

1. **Reference**
2. DingTalk, <https://www.dingtalk.com/en> , Available at:2019/12/1
3. Jibble, <http://www.jibble.io/> , Available at: 2019/12/11
4. Documentation principle for test, <https://max.book118.com/html/2017/1204/142742110.shtm>, Available at: 2020/4/20
5. State diagram, <https://plantuml.com/zh/state-diagram>, Available at: 2020/4/20
6. Open and save file with pyqt, <https://www.cnblogs.com/jmlovepython/p/7399166.html>, Available at: 2020/4/20
7. Get index of listview in pyqt5, <https://ask.csdn.net/questions/156657>, Available at: 2020/4/20
8. Apply qss in pyqt, <https://blog.csdn.net/qq_28119741/article/details/80282695>, Available at: 2020/4/20
9. Using python3, pyqt5 and pycharm to open the camera in user interface, <https://blog.csdn.net/u014453898/article/details/88083173>, Available at: 2020/4/20
10. Install headless opencv for python, <https://blog.csdn.net/qq_41429036/article/details/104883231?depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1&utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-1>, Available at: 2020/4/20
11. Connection between signal and slot, <https://blog.csdn.net/u012552296/article/details/50786200>, Available at: 2020/4/20
12. Datetime format in python, <https://blog.csdn.net/u013862444/article/details/89947381>, Available at: 2020/4/20
13. Return value of datebase select operation using python, <https://blog.csdn.net/u011208984/article/details/86065012>, Available at: 2020/4/20

1. **Appendix**

**12.1 staff allocation table**

**12.2 UI design documents**

**12.3 Coding and comment standard**

**12.4 Test records**

**12.5 Formal meeting minutes**

**12.6 User manual**