Design report for assignment management system (AMS)

T5 group project

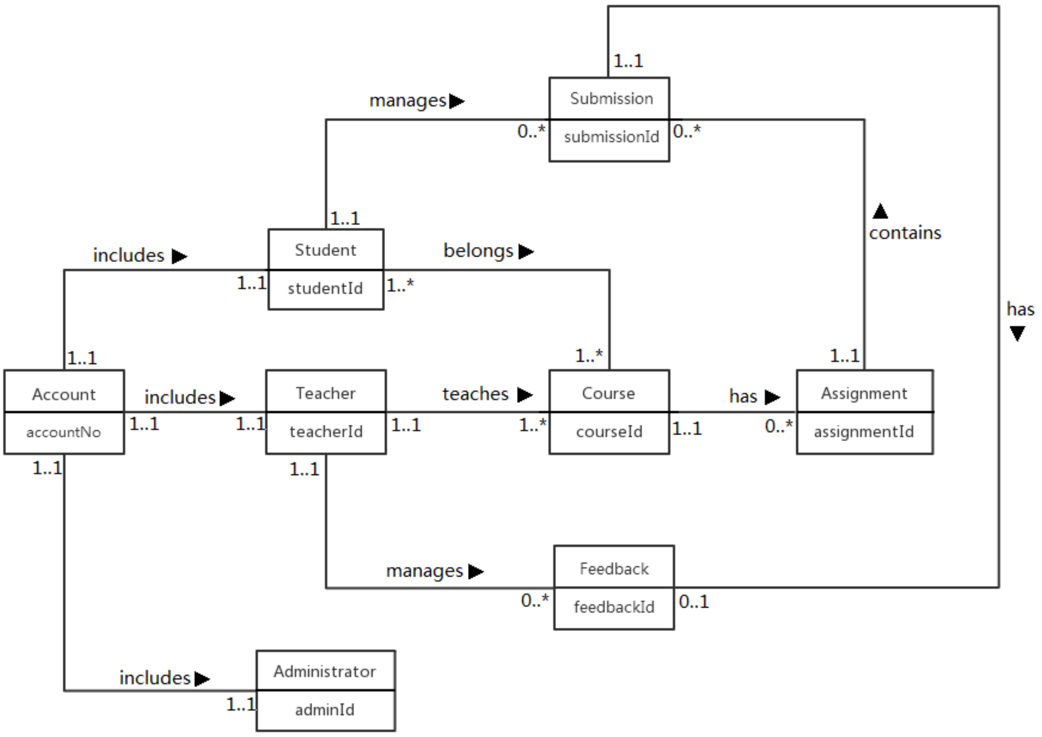
Xiangpeng zhou, Yaotian Li, Zhiyang Li, Chengcheng Jiang, Haochen Wang

The AMS is a typical database program based on webpages, so there are to main parts of design phase, which are database design and webpage design (process design and interface design).

1. Database design

● Global Logical Data Model(ER Diagram)

As we did in our requirement analysis before, the AMS has 3 kinds of users, student teacher and administrator. A teacher teaches one or more courses, and he/she could set assignments for courses taught by him/her. Students belong to courses, so they would link to assignments by courses, and they could manage their submissions for those assignments. In addition, the teacher could add feedback for those submissions. Above all, ER diagram is designed shown below.



● Data Dictionary

ER diagrams shows 8 entities, and there are some basic information about them in tables.

Entity name	Description	Aliases	Occurrence
Account	Account used to login system	User account	Each member of user has a unique account
Student	General term describing all students belong to *Courses*.	None	More than one students are studying at the university.

Teacher	Person who has a course to teach.	Professor and doctor.	More than one teacher works at the university.
Administrator	Person maintaining database	None	One or two is enough.
Submission	Documents submitted by *student*	Files and documents	Each member of submission belongs to a particular assignment submitted by a particular student.
Course	Course set by university.	Lecture	More than one courses are provided by the university.
Assignment	General term describing homework or task that student needs to complete.	Homework and task	Each member of assignment belongs to a particular course set by a particular teacher.
Feedback	Marks and some documents teacher gave to student.		Each submission has 0 or 1 feedback.

The relationship between entities are shown below

Entity	Multiplicity	Relationship	Multiplicity	Entity
Account	1..1 1..1 1..1	Includes Includes Includes	1..1 1..1 1..1	Student Teacher Administrator
Student	1..* 1..1	BelongsTo Manages	1..* 0..*	Course Submission
Teacher	1..1 1..1	Teaches Manages	1..* 0..*	Course Feedback
Course	1..1	Has	0..*	Assignment
Assignment	1..1 1..1	Contains Has	1..* 0..1	Submission Feedback

And the data used in entities is defined below based on requirement. The length of data is a default value and could be changed in further implementation.

Entity	Attributes	Description	Data type and length	Nulls
Account	AccountNo accountName	Uniquely identifies an account Name of account	5 fixed characters 30 characters	No No

	password	Password of account	15 characters	No
	email	Email address	320 characters	No
	type	Account type(1,2,0 stand for teacher student and administrator)	1 fixed integer	No
Student	studentId	Uniquely identifies a student	9 fixed characters	No
	name	Name of student	30 characters	No
	gender	Gender of student.	6 characters	No
	academicYear	The academic year of student	20 characters	No
Teacher	teacherId	Uniquely identifies a teacher	9 fixed characters	No
	name	Name of teacher	30 characters	No
	gender	Gender of teacher	6 characters	No
	office	Office address of teacher	100 characters	No
Administrator	adminId	Uniquely identifies an administrator	9 fixed characters	No
	administratorName	Name of administrator	30 characters	No
Submission	submitId	Uniquely identifies a submission	9 fixed characters	No
	file	Path links to submitted file	1000 characters	No
	submitDate	The submit date of submission	datetime	No
Course	courseId	Uniquely identifies a course	7 fixed characters	No
	name	Name of courses	150 characters	No
Assignment	assignmentId	Uniquely identifies an assignment.	9 fixed characters	No
	name	Name of assignment	150 characters	No
	description	Description of assignment	10000 characters	Yes
	deadline	Deadline of assignment	datetime	No
	file	Path links to assignment document.	1000 characters	No
	fileType	Type of files specified by teacher	10 characters	No
Feedback	feedbackId	Uniquely identifies a feedback.	9 fixed characters	No
	Mark	Mark of submission.	5 characters	No
	File	Path of a feedback document.	1000 characters	Yes

- **Logical Table Structures**

  Since the definition and domain of each entities is clearly displayed above, the relations for those entities still need further explanations, especially for keys of the tables. In order to make database more efficient, each entity is considered to be a table, and the links between adjacent entities are through foreign keys. The detail is shown as following.

Administrator （adminId , administratorName） **Primary Key** adminId	**Account** (accountNo , teacherId , administratorId , studentId, accountName, password, email, type) **Primary Key** accountNo **Foreign Key** teacherId **reference** Teacher(teacherId) **Foreign Key** adminId **reference** Administrator(adminId) **Foreign Key** studentId **reference** Student(StudentId)
Teacher （teacherId, teacherName, gender, office） **Primary Key** teacherId	**Course** (courseId , courseName , teacherId) **Primary Key** courseId **Alternate Key** courseName **Foreign Key** teacherId **reference** Teacher(teacherId)
Student （ studentId , studentName , courseId, gender, academicYear） **Primary Key** studentId	**Assignment** (assignmentId , submissionId, courseId, name, description, file, fileType, deadline) **Primary Key** assignmentId **Foreign Key** submissionId **reference** Sbumission(submissionId) **Foreign Key** courseId **reference** Course(courseId)
Submission (submissionId , submissionDate, file) **Primary Key** submissionId	StudentBelongsToCourse (studentId. courseId) **Primary Key** studentId **Primary Key** courseId **Foreign Key** studentId **reference** Student(studentId) **Foreign Key** courseId **reference** Course(couseId)
Feedback (feedbackId, teacherId, feedbackId, mark, file) **Primary Key** feedbackId **Foreign Key** teacherId **reference** teacher(teacherId) **Foreign Key** feedbackId **reference** feedback(feedbackId)	

- **Physical Table Structures**

  With data dictionary and logical table structures, the physical table structures can be easily defined.

Account table

```
domain Account_Numbers        fixed length character string length 5
domain Account_Names          variable length character string maximum length 30
domain Passwords              variable length character string maximum length 30
domain Emails                 variable length character string maximum length 30
domain Types                  fixed length character integer length 1

Account( accountNo            Account_Numbers     NOT NULL,
         accountName          Account_Names       NOT NULL,
         password             Passwords           NOT NULL,
         email                Emails              NOT NULL,
         type                 Types               NOT NULL)
         Primary Key accountNo
         Foreign Key teacherId reference Teacher(teacherId)
         Foreign Key administratorId reference Administrator （administratorId）
         Foreign Key studentId reference Student(StudentId)
```

Student table

```
domain Student_Ids            fixed length character string length 9
domain Names                  variable length character string maximum length 30
domain Genders                variable length character string maximum length 6
domain Academic_Years         variable length character string maximumlength 20

Student( studentId            Student_Ids         NOT NULL,
         name                 Names               NOT NULL,
         gender               Genders             NOT NULL
         academic year        Academic_Years      NOT NULL)
         Primary Key studentId
```

Teacher table

```
domain Teacher_Ids            fixed length character string length 9
domain Names                  variable length character string maximum length 30
domain Genders                variable length character string maximum length 6
domain Office_Address         variable length character string maximum length 100

    Teacher( teacherId        Teacher_Ids         NOT NULL,
             name             Names               NOT NULL,
             gender           Genders             NOT NULL
             office           Office_Address      NOT NULL)
             Primary Key teacherId
```

Administrator table

```
Domain Admin_Ids                   fixed length character string length 9
Domain Admin_Names                 variable length character string maximum length 30

Administrator( adminId                Admin_Ids             NOT NULL
               administratorName      Admin_Names           NOT NULL)
               Primary Key adminId
```

Submission table

```
domain Submit_Ids                   fixed length character string length 9
domain Files                       variable length character string maximum length 3000
domain submitDate                  datetime

Submission(    submitId               Submit_Ids            NOT NULL,
               file                   Files                 NOT NULL,
               submitDate           Submit_Dates            NOT NULL)
               Primary Key submitId
```

Course table

```
domain Course_Ids                  fixed length character string length 7
domain Names                       variable length character string maximum length 150

Course(   courseId                 Course_Ids            NOT NULL,
          Name                     Names                 NOT NULL)
          Primary Key courseId
          Alternate Key courseName
          Foreign Key teacherIdreference Teacher(teacherId)
```

Assignment table

```
domain Assignment_Ids           fixed length character string length 9
domain Course_Ids               fixed length character string length 9
domain Submission_Ids           fixed length character string length 9
domain Names                    variable length character string maximum length 150
domain Descriptions             variable length character string maximum length 10000
domain Deadlines                datetime
domain Files                    variable length character string maximum length 1000
domain FileTypes                variable length character string maximum length 10

Assignment(  assignmentId       Assignment_Ids     NOT NULL,
             crouseId           Crouse_Ids         NOT NULL,
             submissionId       Submission_Ids     NOT NULL,
             name               Names              NOT NULL,
             description        Descriptions,
             deadline           Deadlines          NOT NULL.
             file               Files              NOT NULL.
             fileType           FileTypes          NOT NULL    )
             Primary Key assignmentId
             Foreign Key submissionIdreference Sbumission(submissionId)
             Foreign Key courseId reference Course(courseId)
```

Feedback table

```
domain Feedback_Ids             fixed length character string length 9
domain Teacher_Ids              fixed length character string length 9
domain Submission_Ids           fixed length character string length 9
domain Marks                    variable length character string maximum length 5
domain Files                    variable length character string maximum length 1000

Feedback (   feedbacktId        Feedback_Ids       NOT NULL,
             teacherId          Teacher_Ids        NOT NULL,
             submissionId       Submission_Ids     NOT NULL,
             mark               Marks              NOT NULL,
             file               Files              NOT NULL. )
             Primary Key feedbackId
             Foreign Key teacherId reference teacher(teacherId)
             Foreign Key feedbackId reference feedback(feedbackId)
```

StudentBlongsToCrouse table

```
domain Student_Ids                   fixed length character string length 9
domain Course_Ids                    fixed length character string length 9

StudentBelongsToCrouse
                ( studentId              Student_Ids       NOT NULL,
                  courseId               Crouse_Ids        NOT NULL )
                  Primary Key studentId
                  Primary Key courseId
                  Foreign Key studentId reference Student(studentId)
                  Foreign Key courseId references Course(couseId)
```

- **Business Rules**

  In practice, some of data has constraints:

  gender is either male or female.
  email is a valid email format
  type of account is one of 0,1,2
  academicYear is one of year1, year2, year3, foundationYear.
  mark is a percentage number between 0% and 100%

- **Transaction/Table Matrix**

  The following information shows the basic privilege for entities.

  Transaction (rm): Enter details of new staff and students registering at a branch.
  Transaction (dm): Update/delete the details of given students and staff.
  Transaction (sf): Submit files to database.
  Transaction (ds): Update/delete the details of submitted files.
  Transaction (rc): Enter details of courses of related teachers.
  Transaction (dc): Update/delete the details of given courses' list.
  Transaction (ra): Enter assignment details of related teacher.
  Transaction(da): Update/delete the details of given assignment.
  Transaction(rf): Submit feedback files to database.
  Transaction(df): Update/delete the details of given feedback.

Transaction	(rm)				(dm)				(sf)				(ds)				(rc)				(dc)				(ra)				(da)				(rf)				(df)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Account	X																																							
Student									X					X	X	X						X								X								X		
Teacher										X												X	X		X					X	X	X	X					X	X	X
Adminstrator	X					X	X	X									X							X																
Submission									X																															
Course																	X																							
Assignment																									X															

I = Insert; R = Read; U = Update; D = Delate;

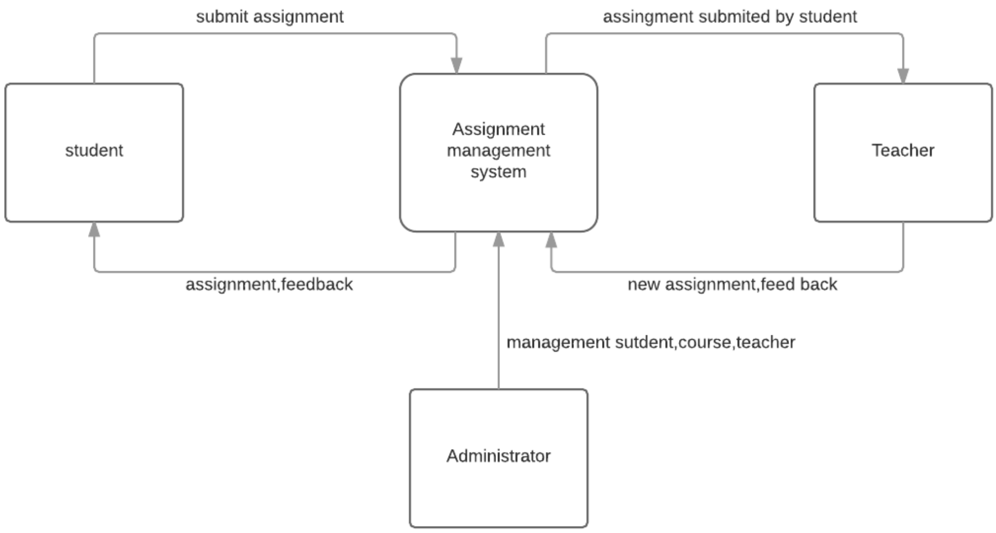
- **Additions**

  In order to improve security and protect personal privacy, HASH encryption is used in our AMS to encrypt some data like password.

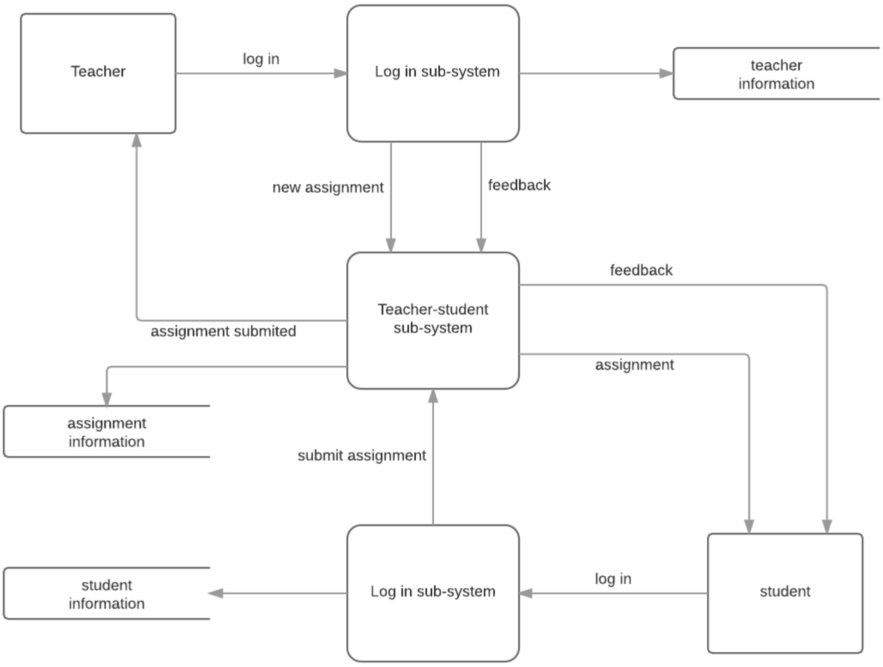
2. Webpage design (process design and interface design)

- **Data flow diagrams**

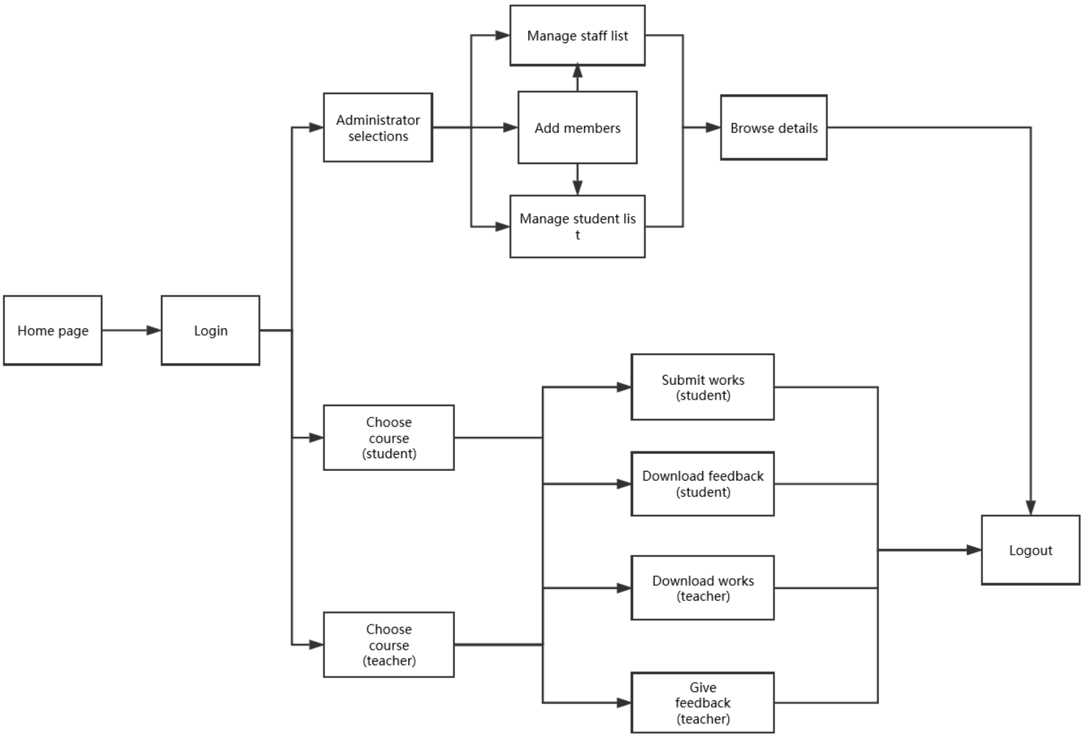
  Data flows in system is a way that 3 kinds of users interact with AMS.


submit assignment
assingment submited by student
student
Assignment management system
Teacher
assignment,feedback
new assignment,feed back
management sutdent,course,teacher
Administrator

In more details, teachers log in first through login sub-system, then they manage them assignment and feedback through teacher-student subsystem. Students are almost the same. They log in through login sub-system and manage their submission through teacher-student subsystem.

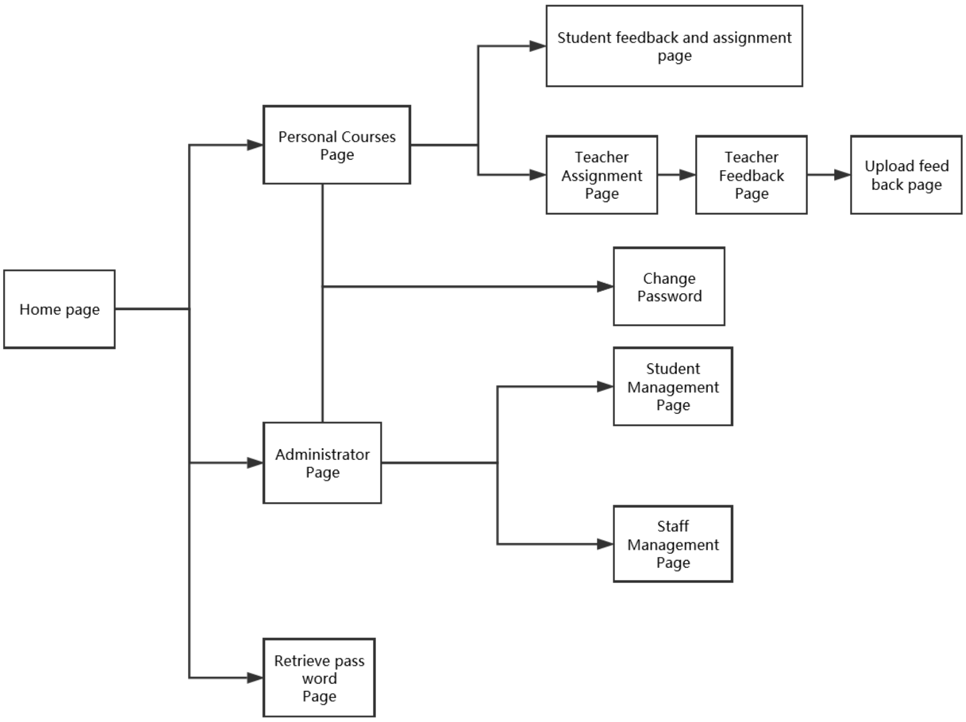


- **A process map**



If we focus on processing, in homepage, user needs login, and then according to their account information, it will turn to 3 kinds of parts. In this case, for administrator, they could choose to manage student, manage teacher or add new members to them, and see details after they did something. For teacher, they need choose a course first. Then they could download students' submission, and give feedback. For student, they could need choose a course first too. Then they could submit works and download feedback. All users could logout at the end.

- **Navigation structure charts**



There are pages design for AMS.

Home page is used for login subsystem.

Personal courses page is used for both teacher and student to choose a particular course to process further actions.

Student feedback and assignment page allows them to manage their submission and download feedback.

Teacher assignment page is used to set and manage teacher’s assignment.

Teacher feedback page is used to give marks for submissions.

Upload feedback page is used to upload files feedback for submissions if they want.

Student management page is used to modify basic information for student.

Stuff management page is used to modify basic information for teacher

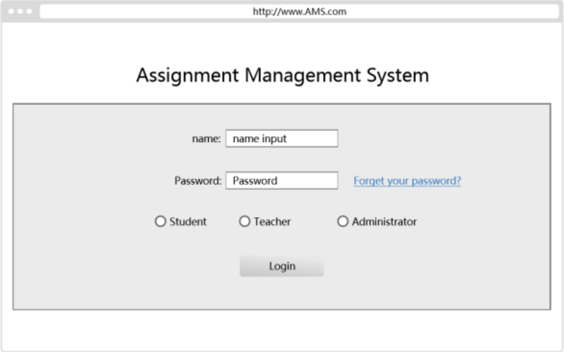
Retrieve password page is used to retrieve password for users.

Change password page is used to change password for who have logged in.

- **User Interface**

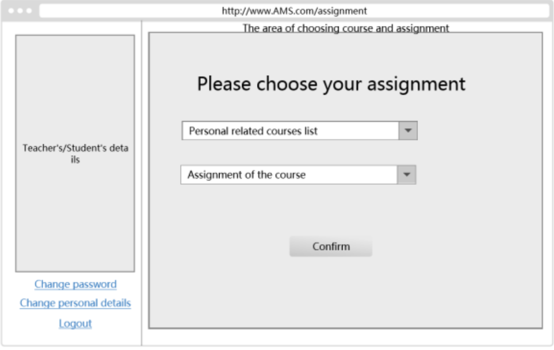
According to the pages defined before, we design a sequence of initiatory UI.

Home page:



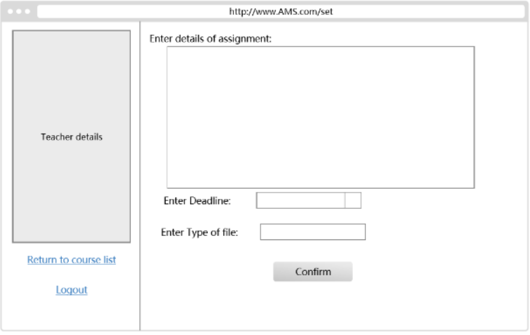
There are 3 kinds of user and user needs enter account name and password to login. If password is forgot, that hyperlinks could lead to retrieve password page.

Personal courses page:


http://www.AMS.com/assignment
The area of choosing course and assignment
Teacher's/Student's deta ils
Change password
Change personal details
Logout
Please choose your assignment
Personal related courses list
Assignment of the course
Confirm

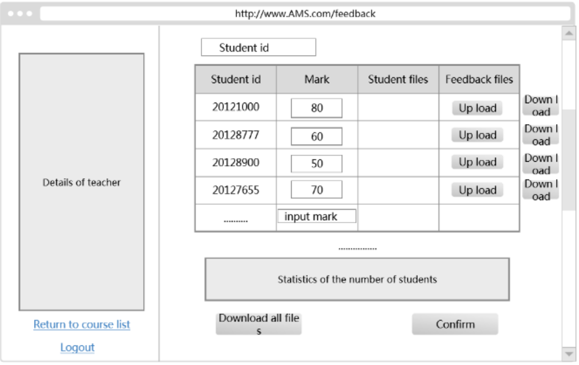
After choosing a course, the details about assignments will show. If user is teacher, add/manage assignment can be chosen. If user is student, submit page can be entered.

Teacher assignment page:



This is the basic layout for adding a new assignment. "confirm" will create a new assignment project if input acceptable.

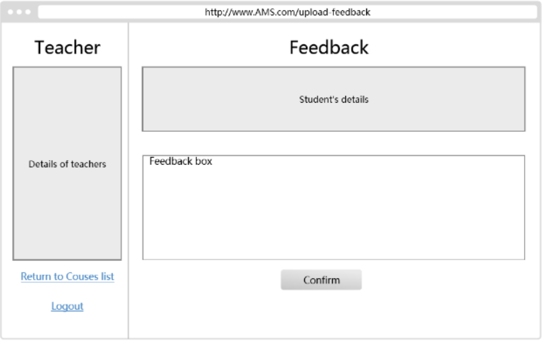
Teacher feedback page:



Student id	Mark	Student files	Feedback files	
20121000	80		Up load	Down load
20128777	60		Up load	Down load
20128900	50		Up load	Down load
20127655	70		Up load	Down load
....	input mark			

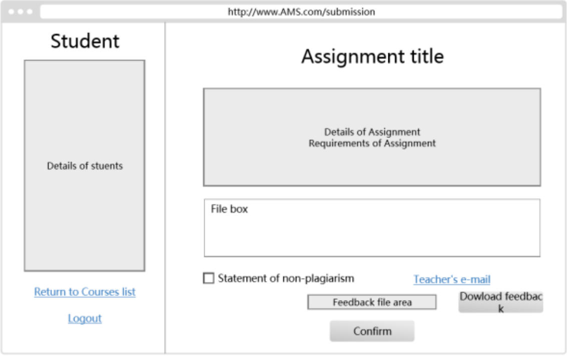
For each line, teacher could "click upload" to enter upload feedback page, "download" to download a specific submission, "download all files" to download all submissions and "confirm" to submit marks.

Upload feedback page:



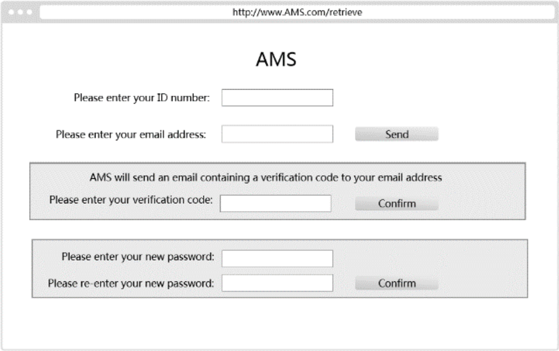
This is the basic layout for adding a feedback file.

Student feedback and assignment page:



Student could download assignment document, submit works, query result and download feedback in this page.

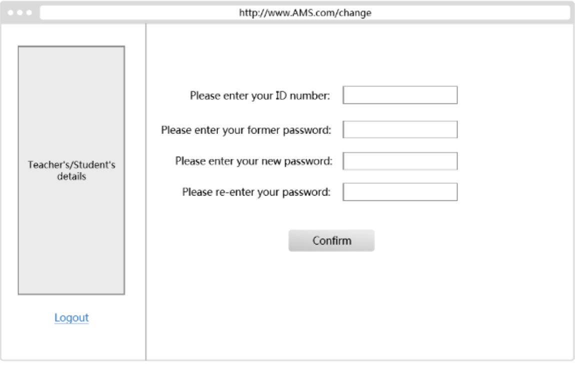
Retrieve password page:


http://www.AMS.com/retrieve
AMS
Please enter your ID number:
Please enter your email address:
Send
Please enter your verification code:
Confirm
Please enter your new password:
Please re-enter your new password:
Confirm

AMS will send an email containing a verification code to your email address

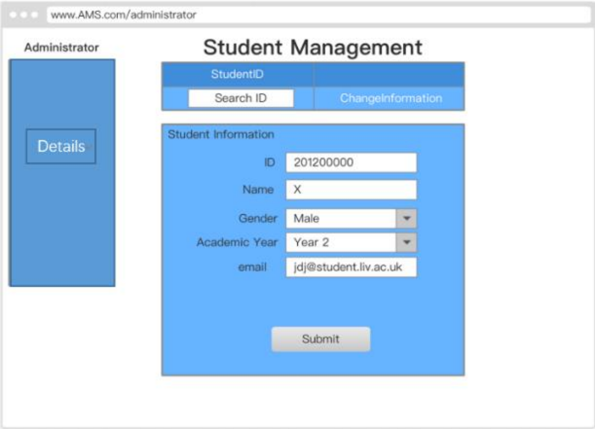
This page provides a mechanism to retrieve password. If ID and email address matched, AMS will be sent. After verification code confirmed, new password could be set.

Change password page:


http://www.AMS.com/change
Teacher's/Student's details
Logout
Please enter your ID number:
Please enter your former password:
Please enter your new password:
Please re-enter your password:
Confirm

This page can be accessed after login. If ID and old password matched, new password will be accepted.

Student management page:



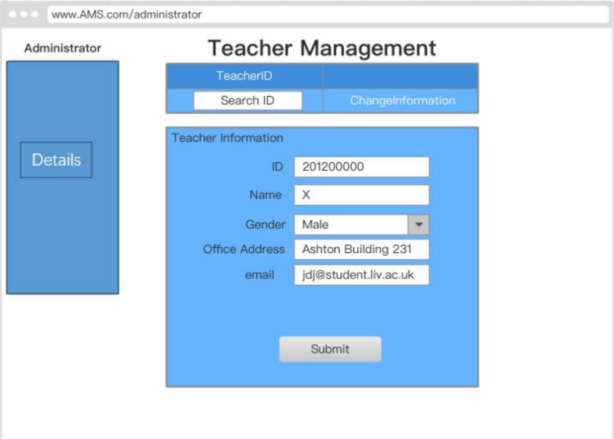
Student Management

Student Information

ID 201200000
Name X
Gender Male
Academic Year Year 2
email jdj@student.liv.ac.uk

Administrator could assess a particular student by ID and modify information. “submit” confirms changes.

Stuff management page:



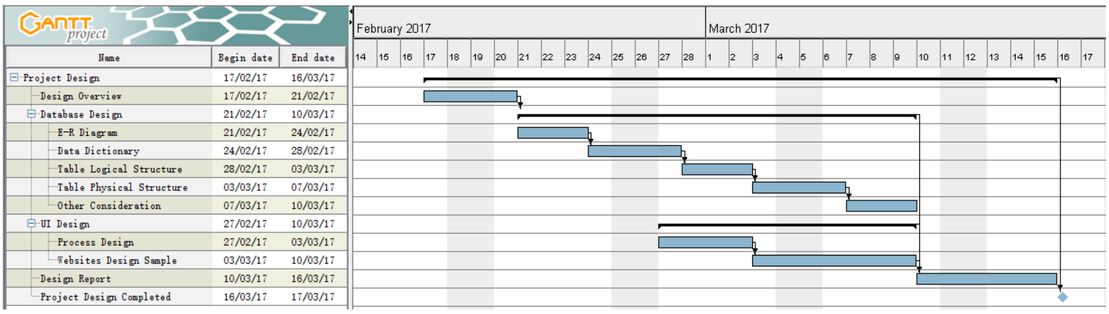
Teacher Management

Teacher Information

ID 201200000
Name X
Gender Male
Office Address Ashton Building 231
email jdj@student.liv.ac.uk

The design of layout for this page is same as student. “submit” confirms changes.

3. Gantt diagram



Name	Begin date	End date
Project Design	17/02/17	16/03/17
Design Overview	17/02/17	21/02/17
Database Design	21/02/17	10/03/17
E-R Diagram	21/02/17	24/02/17
Data Dictionary	24/02/17	28/02/17
Table Logical Structure	28/02/17	03/03/17
Table Physical Structure	03/03/17	07/03/17
Other Consideration	07/03/17	10/03/17
UI Design	27/02/17	10/03/17
Process Design	27/02/17	03/03/17
Websites Design Sample	03/03/17	10/03/17
Design Report	10/03/17	16/03/17
Project Design Completed	16/03/17	17/03/17

Xiangpeng zhou: data dictionary, report writing and coordinating.
Yaotian Li: E-R diagram and Gantt diagram
Haochen wang: physical table structure and data flow diagram
Zhiyang Li and Chengcheng Jiang: UI design.