E-Classroom System

Software Specification

Team 4

(Duo Zhang, Guanqi Guo, Da Lei, Yiming Liu)

1. Introduction

This section provides an overview of all things included in this Software requirement specification document. Also, this document provides the objectives and a list of abbreviations.

1.1 Purpose

The purpose of this document is to give the “E-classroom System” an expatiation to the requirements. The purpose and the integral declaration of the system will be illustrated. Also, the interface, interactions, and system constrains will be explained. After reading this document, customers will have a clear idea about the “E-class System”.

1.2 Scope

The “E-classroom System” is a website based application that helps managers and users manage their class such as choose classes, delete classes, add new class, check information and so on. This application should be easily accessed through website.

The teachers can manage the classes that they will teach. They can add new classes if needed or delete class if they cannot teach this class for any reason. They can provide electronic teaching materials and notations through this system. Also, teachers can let students upload their homework through this system.

Furthermore, this system is a web-based system. All information will be stored in a database, which is web-server located.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
User	Student
Manager	Teacher
SRS	Software requirement specification

1.4 Reference

The Reference of our system can be divided into two parts, the first part is the open source frame that we use in our code, the second part is the design elements that we use in our view layer.

Part1 open source frame:

[1] https://struts.apache.org/

[2] Kurniawan, Budi. *Struts 2 design and programming: a tutorial*. Brainy Software Inc, 2007.

[3] https://spring.io/

[4] Johnson, Rod. "Introduction to the spring framework." *TheServerSide. com*21 (2005): 22.

[5] http://hibernate.org/

[6] Ang, G. A. O., and W. E. I. Wen-xue. "Application of Java data persistence with Hibernate and Struts framework [J]." *Computer Applications* 12.8 (2005): 109-114.

Part2 design Elements:

[1] Html5 frame: "Flaty Mobile Retina" Author: Flaty Group Time: 3/15/2013

[2] Buttons: http://www.zcool.com.cn/ Author: LuoTuo Time: 10/9/2014

[3] ICONS: http://semantic-ui.com/ Author: semantic-ui.com Time: Unknow

[4] ICONS &Buttons: "Flaty Mobile Retina" Author: Flaty Group Time: 3/15/2013

1.5 Overview

The following part of this document includes two sections and appendixes. The second section offers an overview function of the “E-class System” and the interactions between system, users and managers. Further, the system also mentions the constrains and the proposals about the product.

The third section is mainly about the specified requirements in specific terms and descriptions of the different interfaces.

The appendixes are at the end of the SRS.

2. Overall description

This chapter provides an overview of the “E-classroom System”. The context of the system will be explained such as the basic functions of the “E-classroom system”. It will describe how can managers and users use this system. In the end, this part will present the constrains and assumptions of the system.

2.1 Product functions

With “E-classroom System”, users can log in and log out. Beyond that, users can enroll and drop classes. They can view their information and edit their information. Also, users can view classes’ information.

For managers, they have not only the rights that users can do but also some functions that users do not have. Managers can check course information and edit course information. They can also check those students’ who have enrolled in this class information. Managers have the right to add class, update class, and delete class. In addition, managers can publish course materials.

All operations could be done on websites.

2.2 Constraints

Internet connection is required.

2.3 Assumptions and dependencies

Our Assumptions about the “E-classroom system” is that it will always be used while people have the availability to get access to the internet. They can use the system through website.

3. Specific requirements

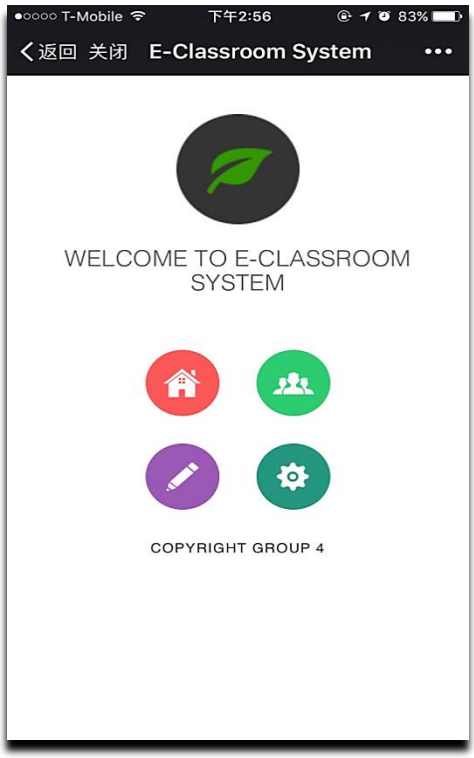
This section illustrates the functions of the “E-classroom system”. It will describe these functions in details.

3.1 External interface Requirements

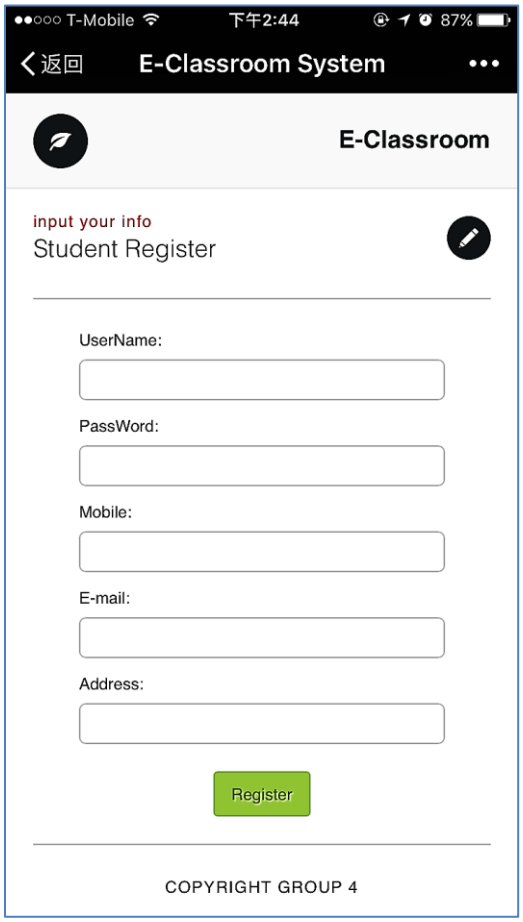
This section shows specific descriptions of inputs and outputs of the system. Hardware interface is described. Some photograph of the user and manager interface are provided.

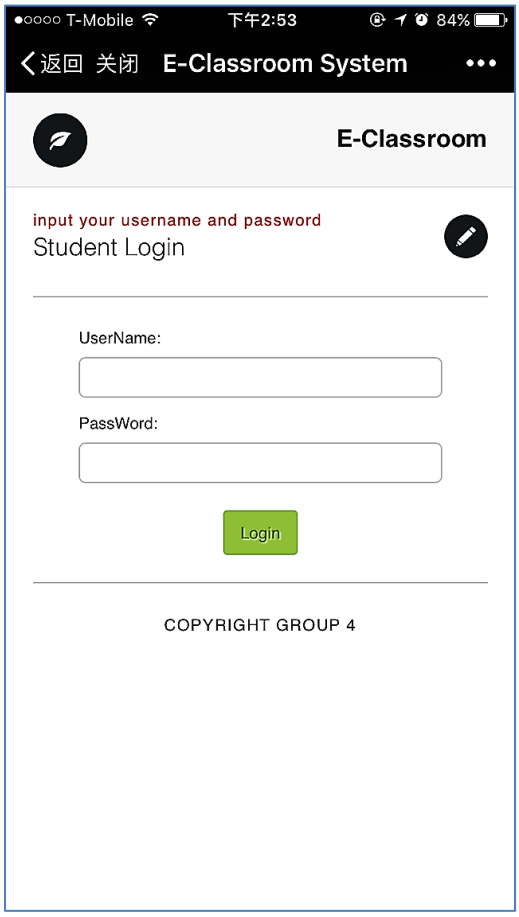
3.1.1 User interfaces

First, if the user is the first time use this system, users can register for the “E-classroom System”.

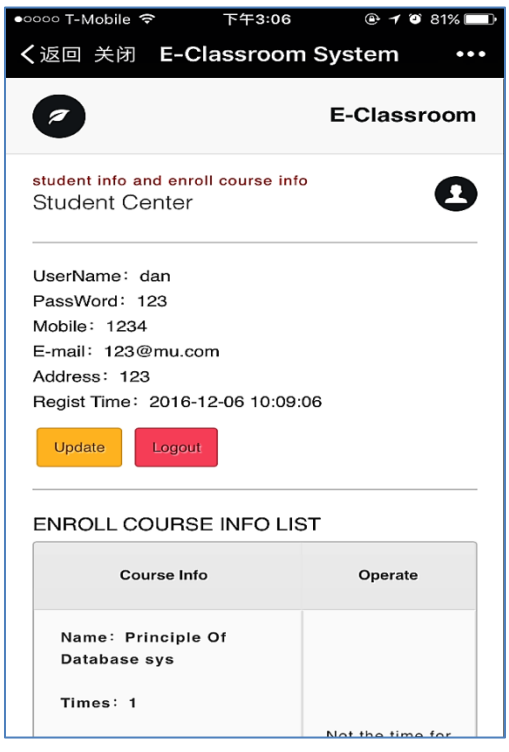


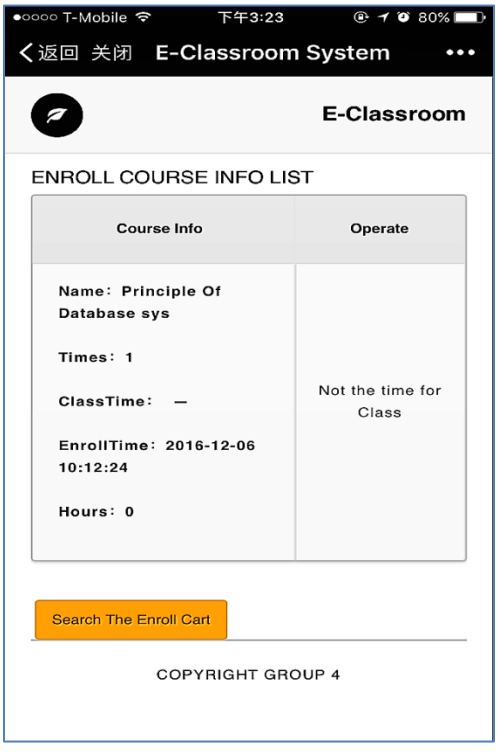
After click the register bottom (pen icon), users need to provide information like this (left), and then user can sign in (right):



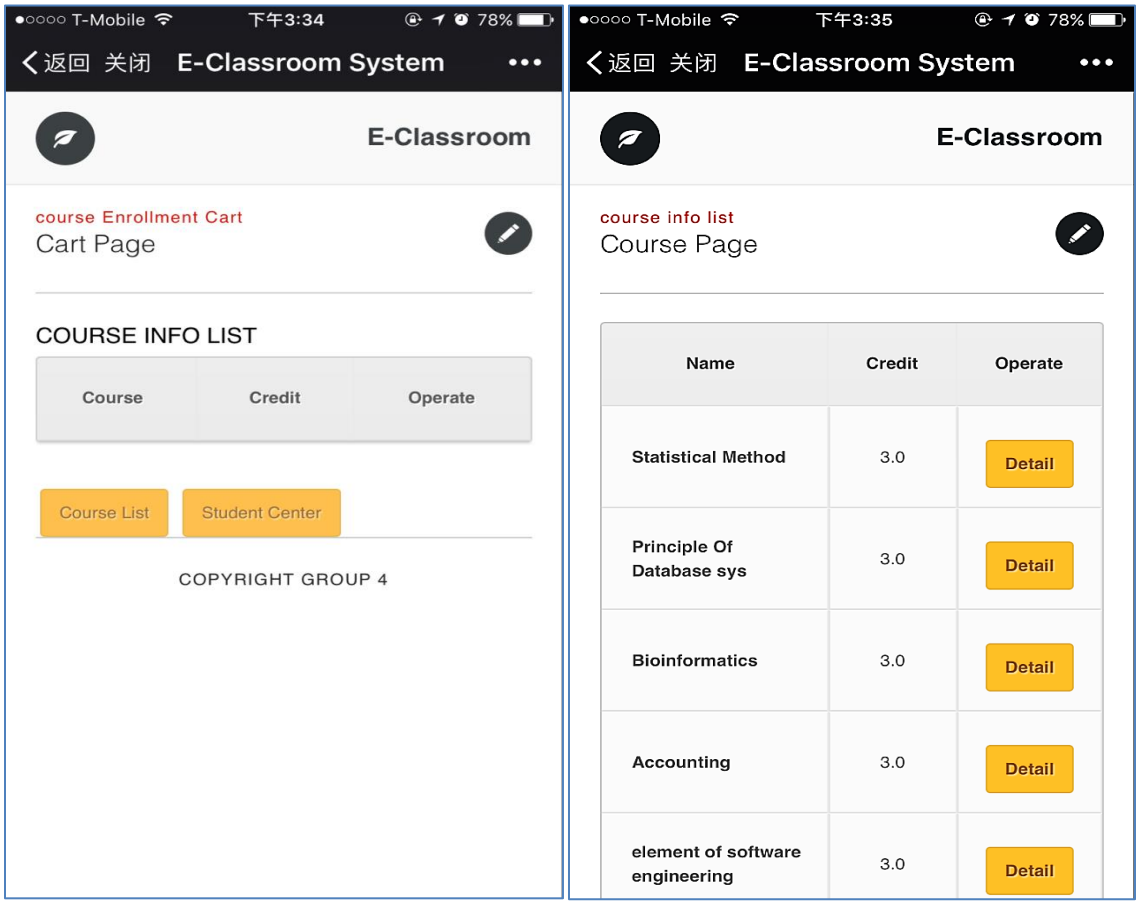


users can check they information after sign in:

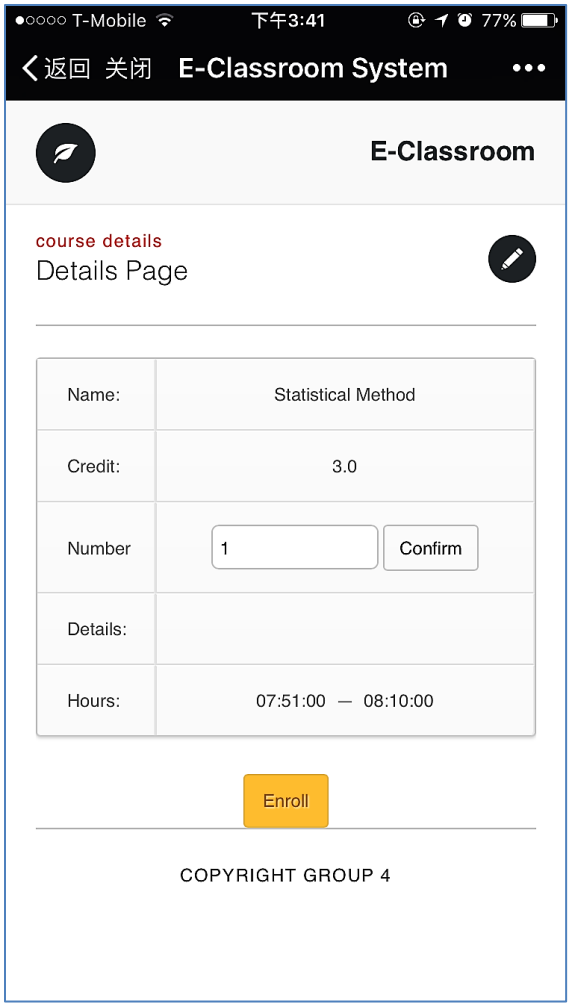


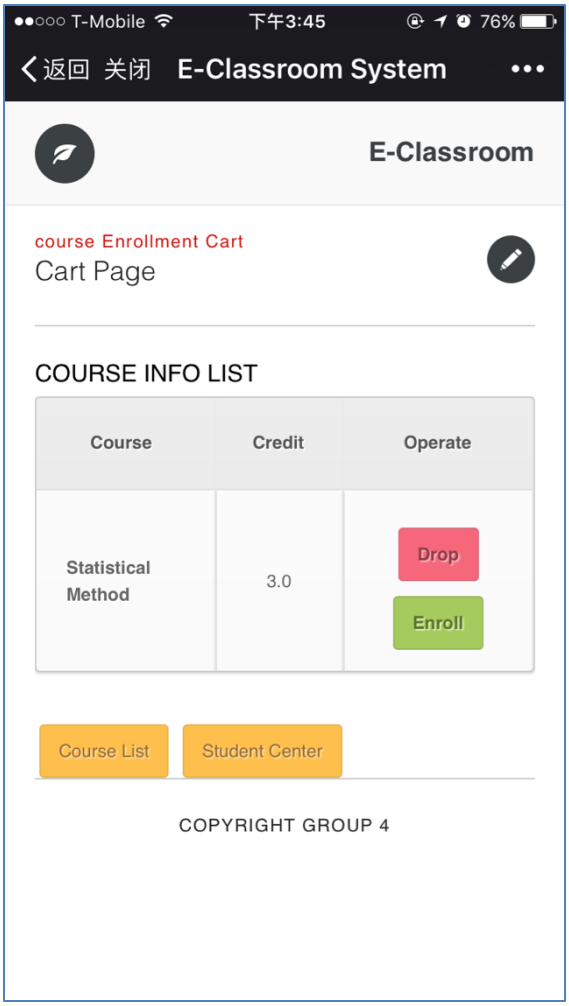


click the search the enroll cart (yellow button at the bottom of the student information interface) and then go to the course list, users can enroll the courses:

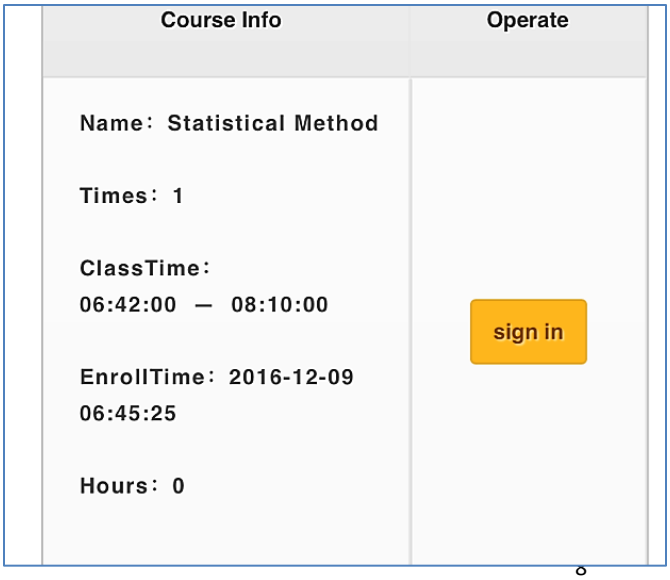


When users enroll classes, classes will not be enrolled immediately. They will see the course information first, they will know how many credits each course and the course description. They can understand what is the prerequisite for the class and the class hour. If users want to take this class, users need to confirm their enrollment. Click the green button enroll and then users will be enrolled in the courses they want. If users want to make sure how many classes and what classes they have enrolled, they just need to go back to the student center.



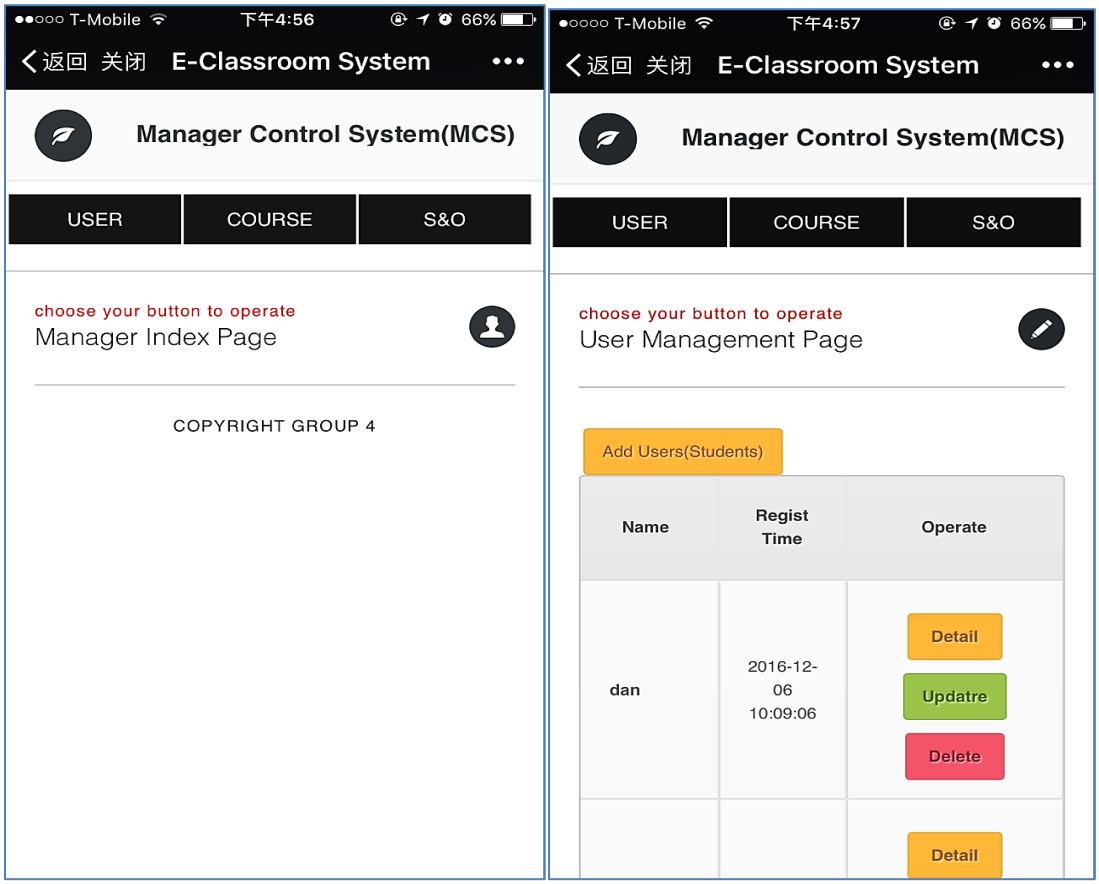


The last function for students is the check in function. When they go to the student center, they can check in at a given time:

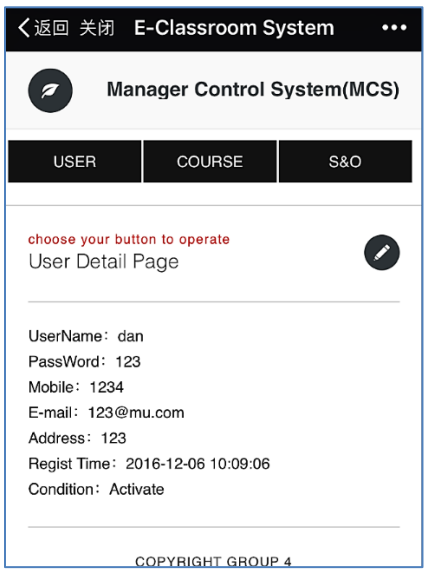


3.1.2 Manager interfaces

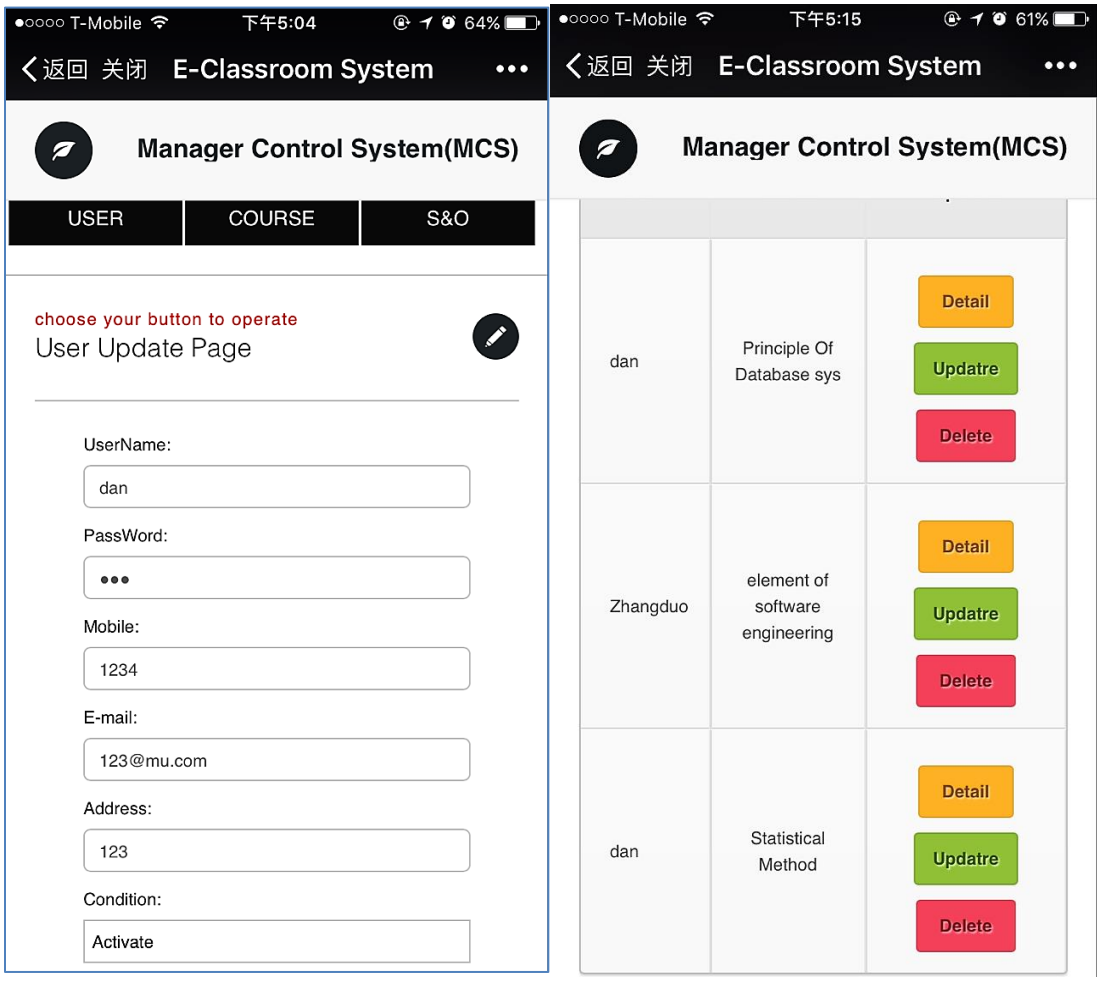
Managers can view students' information:



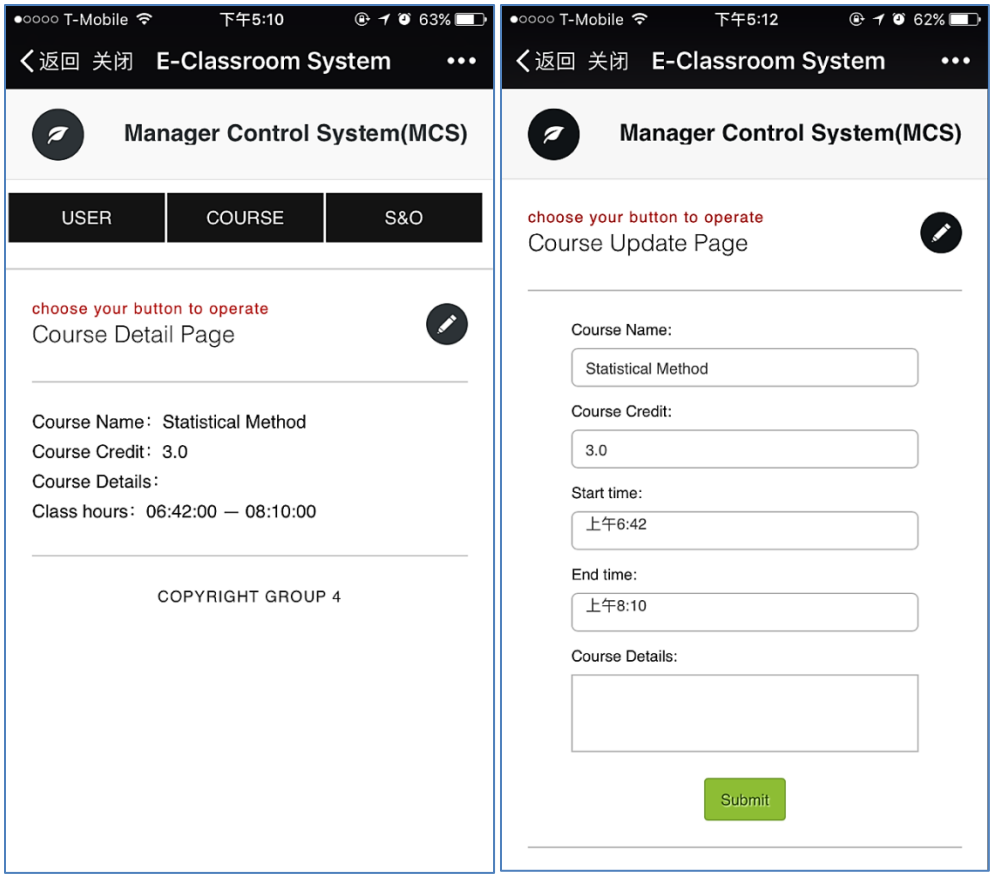
Manager can check the detail information of a student:



Also, Manager can update users' information and delete users:



In addition, managers can manage course information. Manager can check the detail of courses, update the course information, and delete courses. In the update function, managers can change the name of courses, edit the credits of courses, edit the course details, and set the check in start time and end time:



3.1.3 Hardware interfaces

Since this system is web based, there is not any hardware interfaces. All information will be saved on remote side on web server.

3.2 Function requirements

This part details the fundamental actions of the “E-classroom system”.

3.2.1 User Class - The User

3.2.1.1 Functional requirement

ID: FCR 1

Title: Register

Describe: A student should be able to register online. The user must provide user name, password, mobile, email, and address.

Rational: Give students the capability to register to this system.

Dependency: NONE.

ID: FCR 2

Title: Log in

Describe: A user must log in to use the system. Users need to input the user name and password.

Rational: Give users the capability to access to their account.

Dependency: FCR 1.

ID: FCR 3

Title: Check information

Describe: A user could check their own information such as user name, password, mobile, email, and address.

Rational: User should be able to check their information to make sure everything is up to date.

Dependency: FCR 1 & FCR 2

ID: FCR 4

Title: Course list

Describe: Check the courses that users can take

Rational: Uses will be able to take courses.

Dependency: FCR1 & FCR2

ID: FCR 5

Title: Course detail

Describe: students can check the detail of courses they want to know and enroll.

Rational: Users can have better understanding of the courses that they are interested in.

Dependency: FCR1, FCR2 &FCR4

ID: FCR 6

Title: Enroll classes.

Describe: Users can enroll classes by click enroll

Rational: Enroll in classes is one of the most important function of this system. Also, it will be many prerequisites for other functions.

Dependency: FCR 1, FCR2, FCR4 &FCR5

ID: FCR 7

Title: Check in

Describe: Users can use the function to check in.

Rational: Users can use the function to take attendance.

Dependency: FCR1, FCR2 &FCR6

3.2.2 Manager Class - The Manager

3.2.1.2 Functional requirement

ID: FCR 8

Title: View student's information

Describe: The manager can check his/her students' information

Rational: Let the manager and user keep in touch.

Dependency: FCR 1 , FCR2 & FCR 3

ID: FCR 9

Title: Manage student's information

Describe: The manager can check students' information, edit students' information, and delete students.

Rational: Manager will have better capability to control classes.

Dependency: FCR 1 , FCR2 & FCR 3

ID: FCR 10

Title: Manage course information

Describe: Manager can check the details of classes, update classes, delete classes, and set the check in time for students.

Rational: To meet the further changes in classes, manager course information is required.

Dependency: FCR 1 & FCR 2

3.3 Performance requirements

3.3.1 Usage of the link

ID: PFR 1

Title: Usage of the link

Describe: The link should be prominent and it must be a usable link. Using this link should only need one click.

Rational: The link of the system should be convenient.

Dependency: NONE

ID: PFR 2

Title: Response time

Describe: The response time of each function should be no more than 3 seconds.

Rational: Fast response time can give users better use experiences

Dependency: NONE

3.4 Software system attributes

3.4.1 Reliability

ID: SSA 1

Title: System Reliability

Describe: The reliability of the system.

Rational: The system should stable enough for users and managers.

Meter: Measurements obtained from 20 searches during testing.

Requirements: More than 98% of the searches.

Plan: More than 99% of the searches.

Wish: 100% of the searches.

3.4.2 Availability

ID: SSA 2

Title: System Availability

Describe: The Availability of the system when it is operated.

Rational: The system should Available to users and managers.

Meter: Measurements obtained from 100 hours of usage during testing

Requirements: More than 98% of the searches.

Plan: More than 99% of the searches.

Wish: 100% of the searches.

3.5 Test

3.5.1 Introduction of Black box testing

Black box testing, also known as functional testing, it is through the test to detect whether each function can be used normally. In the test, the program will be regarded as a black box which can not be opened, regardless of the internal structure and internal characteristics of the program, the program interface to test, it only checks whether the program function in accordance with the provisions of the specification requirements for normal use, Whether the program can properly receive the input data to produce the correct output information.

Black box testing focused on the external structure of the program, regardless of the internal logical structure, mainly for software interface and software functions to test.

3.5.2 Test cycle arrangements and test methods

Since we have adopted the agile software development model to our project, therefore, we did not stop testing at every stage of the development process. After each iteration, we conducted a centralized test for the new parts and pervious parts. The test procedure are as below:

Iteration 1: Decide the overall features of the system, and make a general view about the logical between the features.

Test:

No specific implementation, so the test link to stay in the logic test.

Iteration2: Implement basic features of users, and managers.

Users:

1. Login/Logout.
2. Update personal information.
3. Register.
4. View Courses and enroll courses.

Manager part1 user information management:

1. Add a user.
2. Delete a user.
3. Update a user's personal information.
4. Check the user's account details.

Manager part2 courses information management:

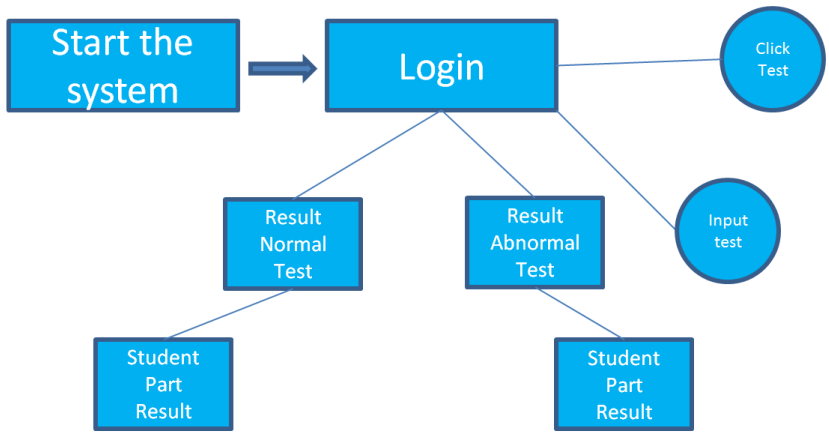
1. Add a course.
2. Delete a course.
3. Update a course's details.
4. Check the course's details.

Manager Part3 Enrollment information management:

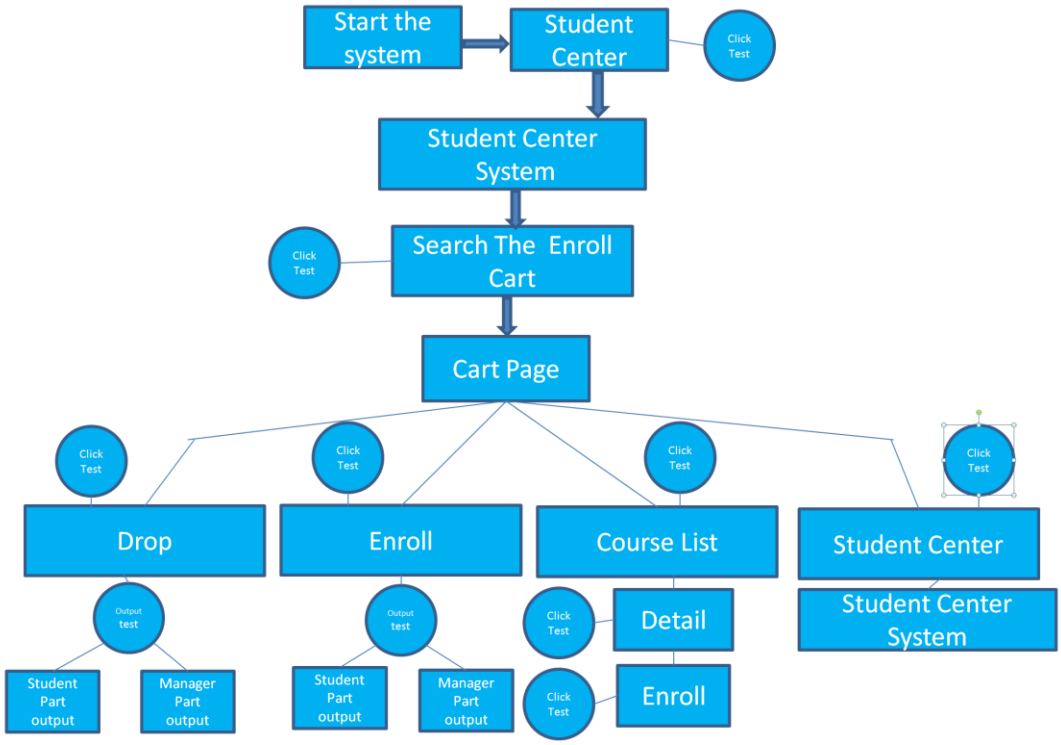
1. Check all the enrollment information divided by user name.
2. Update the enroll time of a user for a specific course.
3. Delete a user's course that have already enrolled.

Test:

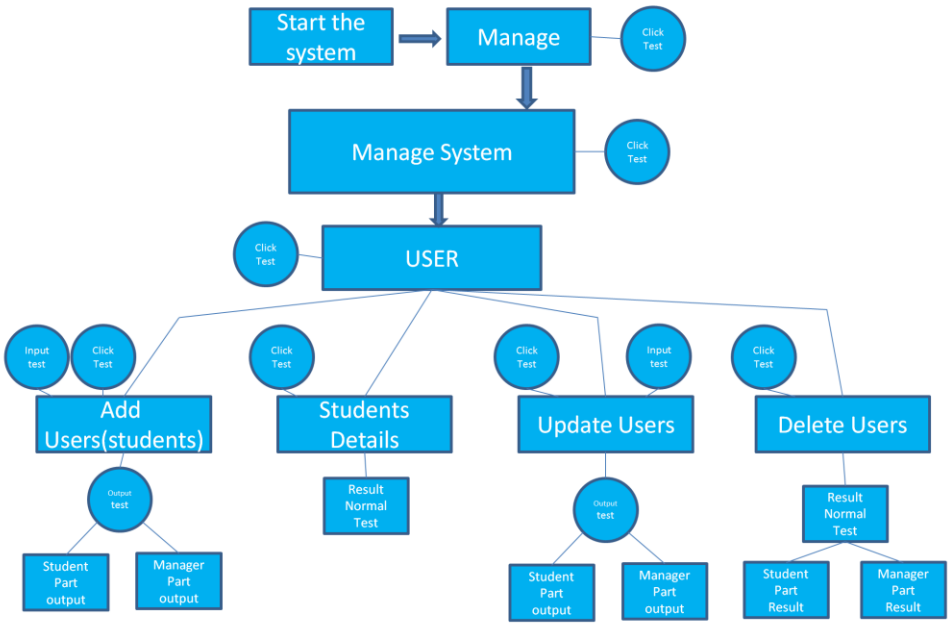
Users part 1 (Test flow chart):



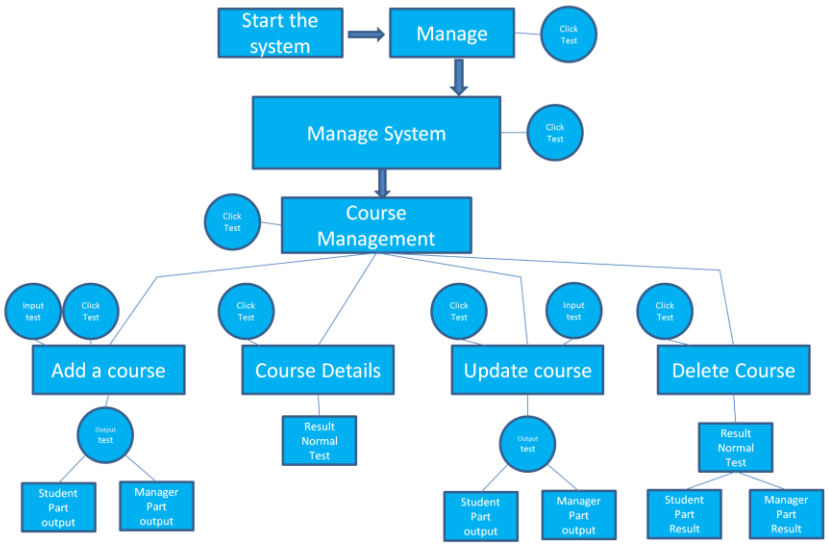
User part 2 to 4 (Test flow chart):



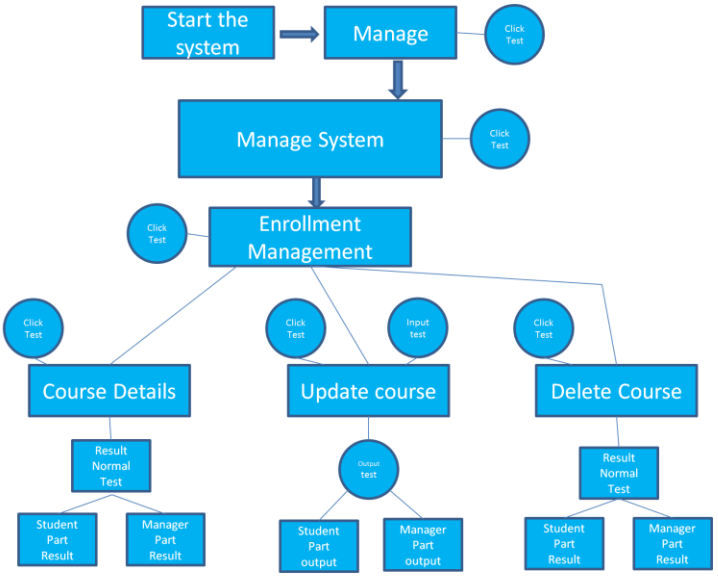
Manager Part1 (Test flow chart):



Manager Part2 (Test flow chart):



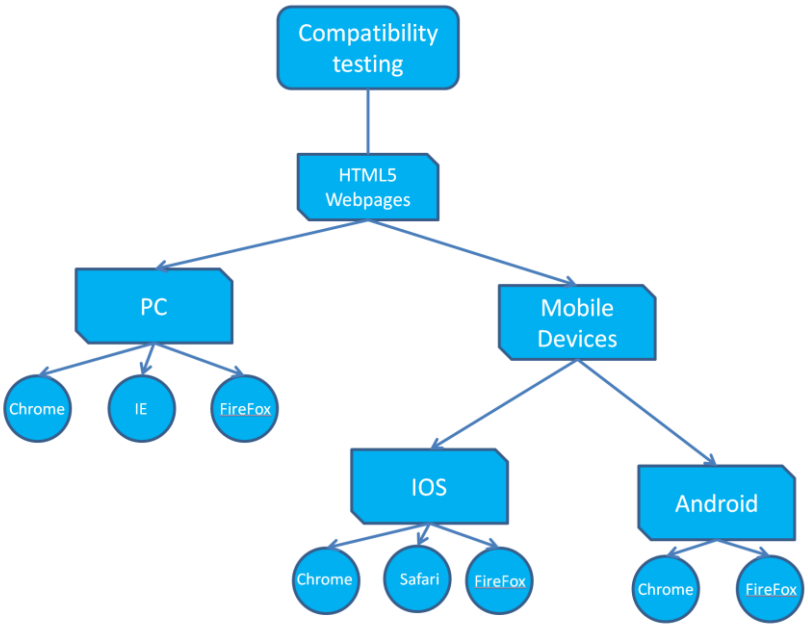
Manager part3 (Test flow chart):



Iteration3: The View layer, HTML5 Web pages will be embedded in the system.

In this iteration, because there is no new logic was added to the system, we only tested compatibility of the web pages for different platforms.

Test:



Iteration4: Sign in part for the system.

User part:

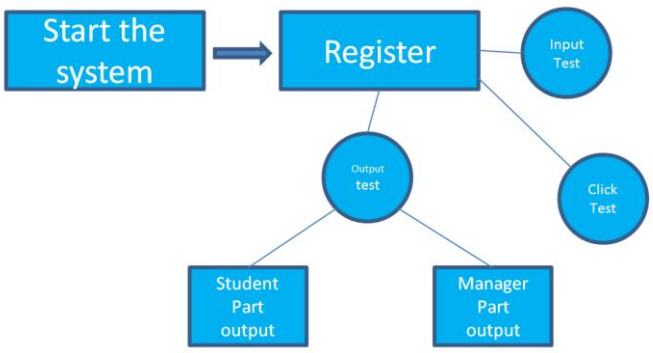
1. Sign-in for a course.

Manager part:

2. Set up a sign-in time zone for a course.

Test:

User part (Test flow chart):



Manager part (Test flow chart):