**COURSE PROJECT**

**Instructor:** Truong Ninh Thuan

**Project team:**

* Tran Quang Cuong Student.No: 13020049 K58CA
* Nguyen Thanh Trung Student.No: 13020463 K58CA
* Nguyen Vu Hoang Vuong Student.No: 13020763 K58CA

**Project requirement:** Analyze and design a documentation for any software project.

Table of contents

1. Online Course Reservation System Requirements
   1. Problem statement
   2. Requirements
   3. Glossary
   4. Supplementary Specification
   5. Use-Case Model
      1. Login
      2. Select college and course
      3. Submit grades
      4. Edit professor profile
      5. Edit student profile
      6. Close registration

http://www.slideshare.net/the10dar/document-33319174

**Social network**

**Problem statement**

Online social media nowadays are extremely popular, attracts many users of all age, especially young ones. We are tasked with to developing a simple social network where people can making friends as well as stay connected and interact with each other even if they are distance apart.

**Requirement**

The social network to be developed is a free website where people can access via a computer or a smartphone with Internet connection.

Anyone can register an account. An account has a profile which contains provided user’s information. Some are private, such as email, mobile number; password, and the rest are public to other users. Changing profile is available.

Users can share their story, their thought by writing a post. Unlike Facebook or Twitter, all posts are public. Any user can see what you have written, vote (like/dislike) and make commnents. Posts are displayed in which we call a “wall” by newest post to oldest post. When one user enter to another user’s account, that user’s wall should be appeared.

Users can chat with each other by a chat window. The chat window keep conversations include old ones between users. When a message came, users must be received a notification about it.

**Glossary**

* user: a person who use our social network
* post: a user’s writing
* comment: a writing describe user’s though about a post of some user.
* wall: which contains a posts of a user.
* chat: texting through Internet

1. Online Course Reservation System Requirements
   1. **Problem statement**

Today, technology has made a very big change in the education world. We are tasked with to developing a new online course reservation system which will provided convenience to students for registering a course. With such a system, students could easily stay at home and reserver courses they are interested in by a PC or by a smart phone with Internet connection.

* 1. **Requirements**

The system will have form of a web-based application.

Stakeholders are provided an account to log in the system.

Each student has a profile on the system which contains an unique ID, full name, date of birth and some others basic information. For fee courses which students took part in, their account must be connected to a bank account. The payment must be deal to be legally enroll courses. The fee can be pay back if the student decide to cancel courses soon. After a fixed amount of time since the courses has started, students are no longer be pay the money back. List of courses that students reserved and associated fee must be available to them. After a course ended, students with good result must get a printable certificate.

Each lecturer also has a profile similar to students’s. Besides, their faculty, degree, teaching exprience and achivements must be included. After course registation time, lecturers will received a list of course that they will teach that semester. Each of them must contains a list of enrolled students and schedule. Lecturers have right and responsibility to update progress of student as well as their final result.

Every course has its own ID and schedule. No two different courses would conflict with each other, that is their schedule must not identical or overlap. Necessary information such as course outline, textbook, instructor’s name must also be mentioned. Courses may have some restrictions such as minimum/maximum number of students must/could enroll, prerequisite courses or specific type of students, etc.

*Performance:*

The system must be able to serve 2000 students concurrently on register time.

Payment through Internet must be security.

Each query must be processed within 5 seconds.

* 1. **Glossary**
* Course: class for a particular subject
* Student: people who enrolled the course
* Professor: people who teach at college
* Applicant: people who wish to register the Course.
* Instructor/lecturer: people who give lessons.
* Course offering: registering a course.
  1. **Supplementary Specification**

**Objectives**

The purpose of this document is to define requirements of the Online Course Reservation System. This Supplementary Specification lists the requirements that are not readily captured in the use cases of the use-case model. The Dupplementary Specification and the use-case model together capture a complete set of reuirements on the system.

**Scope**

This Supplementary Specification applies to the Online Course Reservation System, which will be developed by the OOAD students.

This specification defines the non-functional requirements of the system; such as reliability, usability, performance and supportability, as well as functional requirements that are common across a number of use cases. (The functional requirements are defined in the Use Case Specification).

**References**

IBM Rational Software Documentation (Version 2004)

**Functionality**

Multiple users must be able to perform their work concurrently.

If a course offering becomes full while a student is building a schedule including that offering, the student must be notified.

**Usability**

The desktop user-interface shall be clear and easy to use by newbies within 15 minutes.

**Reliability**

The system shall be available 24 hours a day 7 days a week, with no more than 10% down time.

**Performance**

The system shall provide access to the legacy course catalog database with no more than a 10 second latency.

The system must be able to complete 85% of all transactions with 3 minutes.

The GUI transitions must be smooth.

**Supportability**

None

**Security**

The system must prevent people are not manager from changing any course catalog or schedules of students.

Almost changes of the system databases can only be done by the manager. Students can only change their own schedule than others. Require confirm password before submit any changes.

**Design Constraints**

The system shall integrate with an existing legacy system, the Course Catalog System, which is an RDBMS database.

The system shall provide a Windows-based desktop interface.

* 1. **Use-Case Model**

Professor

Manager

Registrar

* + 1. **Login**

**Brief Description**

This use case describes how a user logs into the ONLINE COURSE RESERVATION SYSTEM.

**Flow of events**

*Basic flow*

This use case starts when the actor wishes to Login to ONLINE COURSE RESERVATION SYSTEM.

* + - 1. Actor enters his/her name and password
      2. The system validates the entered name and password and logs the actor into the system.

*Alternative Flows*

**Invalid Name/ Password**

If, in the Basic Flow, the actor entered an invalid name and/ or password, the system displays an error message. The actor can choose to either return to the beginning of the Basic Flow or cancel the login, at which point the use case ends.

**Special Requirements**

None

**Pre-Conditions**

The system has the login screen displayed.

**Post-Conditions**

If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.

**Extension Points**

None

* + 1. **Select college and course**

**Brief Description**

This use case lists out the various courses offered by the institution.

**Flow of events**

*Basic flow*

This use case starts when the actor wishes to select college courses after Login to system.

* + - 1. Actor(he/she) send request for information of the college
      2. The system provides details for the college
      3. Actor request for hostel facilities
      4. System check if availability hostel facilities
      5. Accept the hostel facilities if available
      6. Actor join the course

*Alternative Flows*

**Missing/ Not available – college or course information**

If, in Basic Flow, information of the college or sub-information of hostle facilities is not available, actor can Return to start of Basic flow or cancel.

**Special Requirements**

None

**Pre-Conditions**

The system has the login screen displayed.

**Post-Conditions**

If the use case was successful, the actor is now join the course. If not, the system returns Basic flow or Logged screen depends on actor’s directives.

**Extension Points**

Logged in screen.

* + 1. **Submit grades**

**Brief Description**

This user case given the marks scored by the system.

**Flow of events**

*Basic flow*

This usercase start at when an actor wants to get his/her marks scored by system.

* + - 1. Actor(he/she) send request for completed course.
      2. The system provides the courses that actor had completed.
      3. Actor request for mark of that course.
      4. System return the marks.

*Alternative Flows*

None

**Special Requirements**

None

**Pre-Conditions**

The system has the login screen displayed.

Actor must had done the course for getting the submitted marks.

**Post-Conditions**

If the use case was successful, the actor will by return marks.

**Extension Points**

None

* + 1. **Edit professor profile**

**Brief Description**

This usecase maintain the information about professor in the system.

**Flow of events**

*Basic flow*

This usercase start at when an actor wants to maintain the information about professor in the system.

* + - 1. The system retrieves and displays the detail information of the actor.
      2. The actor makes the desired changes.
      3. The system retrieves the changes then validates inputs and then updates the actor’s information in database.

*Alternative Flows*

**Invalid Input**

If in Basic flow, user entered invalid type if inputs, the system displays an errors message and highlights the position of error. The actor can decide to come back to the beginning of the Basic flow or cancel the use case.

**Special Requirements**

If the actor is instructor but not manager, the system only displays to the actor some type of information. Some other such as working days, number of hours work per day, salary are not displayed to the actor.

**Pre-Conditions**

The system has the login screen displayed.

Make a request to add/edit information.

**Post-Conditions**

If the use case was successful, the actor’s information is updated. If not, the system is unchanged.

**Extension Points**

None

* + 1. **Edit student profile**

**Brief Description**

This usecase maintain the information about student in the system.

**Flow of events**

*Basic flow*

This usercase start at when an actor wants to maintain the information about student in the system.

* + - 1. The system retrieves and displays the detail information of the actor.
      2. The actor makes the desired changes.
      3. The system retrieves the changes then validates inputs and then updates the actor’s information in database.

*Alternative Flows*

**Invalid Input**

If in Basic flow, user entered invalid type if inputs, the system displays an errors message and highlights the position of error. The actor can decide to come back to the beginning of the Basic flow or cancel the use case.

**Special Requirements**

If the actor is student but not manager, the system only displays to the actor some type of information. Some other such as working days, number of hours work per day, salary are not displayed to the actor.

**Pre-Conditions**

The system has the login screen displayed.

Make a request to add/edit information.

**Post-Conditions**

If the use case was successful, the actor’s information is updated. If not, the system is unchanged.

**Extension Points**

None

* + 1. **Close registration**

**Brief Description**

This usecase describes that a course can be cancelled if not enough students or not have suitable professor.

**Flow of events**

*Basic flow*

This usercase start when actor registered a course that not enough student or not have suitable professor.

* + - 1. Actor registerred a course.
      2. System confirm if the course has atleast 5 student minimum and suitable professor or not.
      3. If not, allow save course information else alert a message or send an email and a link redirect to selecting course page.

*Alternative Flows*

**Invalid Input**

No

**Special Requirements**

No

**Pre-Conditions**

Registered a course.

**Post-Conditions**

If the use case was successful, that actor’s course will be deleted from actor’s course.

If not, actor’s course will be unchanged.

**Extension Points**

None