



# SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

## LAB PROJECT 3

DUE 7 February 2016      BY MIDNIGHT

### OVERVIEW

This document provides an outline of the specifications of what students are expected to submit as part of the Software Requirements Specification (SRS) which is worth 80 points. *Requirement Document is about describing required data, function, and behavior.* How detailed should the Requirements Document be? It should be detailed enough to define 100% of the scope of your software project, leaving no known requirement detail out. It should also have all the information needed to create the Design Document.

### GUIDELINES

Here are some key guidelines to writing a Software Requirements Specification (SRS):

1. Requirement Modeling asks, “What”, not “How”. What user interactions are involved, what objects are manipulated, what functions must be performed, what behaviors does system have, what interfaces, and what constraints apply?
2. Keep in mind the 3 goals of the SRS:
  - a. Describe 100% of what the customer requires in the product
  - b. Create enough information for the next step: software design. This information may include sample user documents currently used such as Excel sheets, printed reports, etc. shown with the requirements description or placed in Appendix section.
  - c. Define requirements that can be validated as “done” once design is completed and again when software is built.
3. Pretty diagrams can help but can also be confusing. Use more description in as text (called narratives), table, and charts along with simple diagrams. Key point is to have clear requirements, not just pretty diagrams.
4. Make sure you spend time to detail even the complicated requirements. Otherwise, unclear information will come back to bite you during design and coding phases.
5. Don’t make it too long. Watch out for over-documenting those functions that are already well understood.
6. Keep the SRS up to date as you make changes during later phases such as design and coding phases.
7. Approximately 20% of the project time should be allocated to requirements definition.
8. A good requirements document will have enough detail to create the software design document and the test plan.

### REQUIREMENTS SPECIFICATION TEMPLATE (SPECIFICATIONS: 80 POINTS)

#### 1. *10 points.* INTRODUCTION

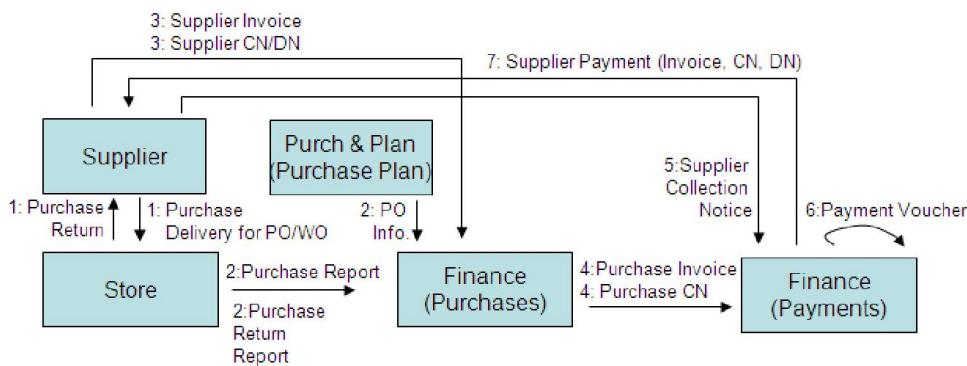
- a. PURPOSE OF THE PROJECT. Why do this project? Goals of this project for ... improved service, increased profit, a better world, etc.
- b. DOCUMENT CONVENTIONS. Definition of all terms including acronyms used in project.

- c. **PROJECT SCOPE OF WORK.** General description of what the software will do. Indicate a summary of what will be done (modules), by whom, and under what circumstances (context). For example, “Procurement will be done by Purchasing Department when Materials are needed”.

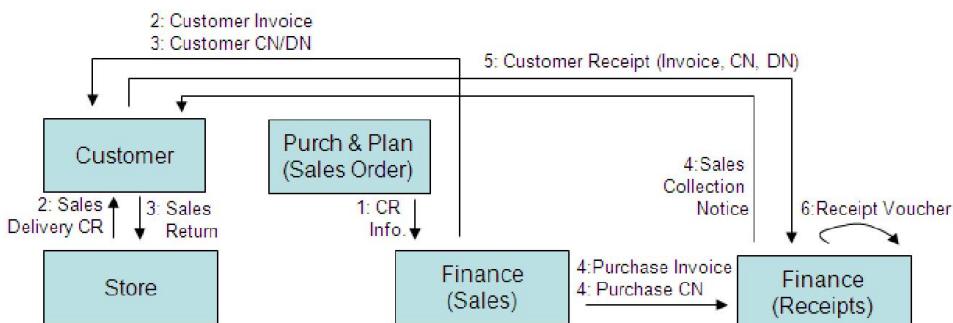
## 2. FUNCTIONAL REQUIREMENTS

- 10 points. PRODUCT PERSPECTIVE.** High-Level Use Case Diagram showing all actors and main use cases. Keep it at simple level such as in [http://en.wikipedia.org/wiki/Use\\_case\\_diagram](http://en.wikipedia.org/wiki/Use_case_diagram).
- 15 points. COLLABORATION OR OTHER DIAGRAM.** You can use either: collaboration, sequence, activity, or swim-lane diagram, whichever is appropriate, to show EVERY scenario. For example, below we show 2 scenarios for Finance Module: One for Accounts Payable (A/P) and One for Accounts Receivable (A/R). This will show the sequence of actions done by various entities (actors) to complete a task loop such as accounts receivable, room reservation, etc.

Finance A/P Scenario



Finance A/R Scenario



- 25 points. COMPLETE FUNCTIONAL REQUIREMENTS LIST.** The functional requirements are those related to the actual work or business your software is trying to model such as reserve room, check-in, change password, etc. The non-functional requirements are those related to the environment of the software such as interface requirement, platform / architecture requirements, performance requirements, and so on. Here you must list all the functional requirements in tabular form grouped by related functions such as each use case or each activity. *You must be very detailed to include ALL known requirements.* Each requirement will be validated during design phase and then again in product delivery phase. For example, just the login aspect must be fully detailed to include rules on acceptable password, disconnect account if incorrect password for 5 attempts, what to do if someone forgot password (email one-time password), role based security to access each form/control/report, creating new user (with option to make inactive when resign from company etc.), changing password by user, changing password by admin, password expiry after 60 days, etc.

## 3. 20 points. NON-FUNCTIONAL REQUIREMENTS

- a. OPERATING ENVIRONMENT. Where will this work? On a client PC using Windows 7 and above, for example.
- b. DESIGN AND IMPLEMENTATION CONSTRAINTS. List the design and programming constraints (requirements) the user may have, such as must be coded in Java and designed using UML etc.
- c. USER DOCUMENTATION (MANUAL) REQUIREMENTS. Type of training and documentation required by user. Do they need video? Online manual linked to special keys in each form? User wants what content in manual?
- d. SYSTEM FEATURES. System features 1, 2, 3... The system features are additional to the software since it may include the manual processes not done by software and other desired behavior of the work system as a whole.
- e. EXTERNAL INTERFACE REQUIREMENTS
  - i. USER INTERFACES. Description of how the user will interact with the software. For web applications, this will be through the web browser, so can just show a sample “look and feel” of the desired interface. For i-Phone applications or games, the user interface can be fancier. This may include appearance requirements, style requirements, and ease of use requirements given to us by the user. For example, “All foreign key fields should have list of value button, where the code, name, and other relevant information appears, so that users need not memorize codes”.
  - ii. HARDWARE INTERFACES. How will the system need to interact with hardware such as POS, barcode reader, game controller, fingerprint scanner, etc.
  - iii. SOFTWARE INTERFACES. How will the software interact with other software in defining the complete system.
  - iv. COMMUNICATION INTERFACES. How the communication will take place in various interfaces such as message passing, data importing, trigger, etc.
- f. OTHER NON-FUNCTIONAL REQUIREMENTS
  - i. PERFORMANCE REQUIREMENTS. May include response time (Speed and Latency Requirements), data capacity, number of transactions to be processed per day, etc.
  - ii. SAFETY REQUIREMENTS. Critical for applications such as airline software.
  - iii. SECURITY REQUIREMENTS. How is it safe from hacking and intrusion? Audit requirements (log file), privacy requirements, access requirements (role based, etc.).
  - iv. SOFTWARE QUALITY ATTRIBUTES. What is the expected quality of the software measured in various aspects such as meeting requirements, timely, downtime, fault-tolerance, Precision or Accuracy Requirements, etc.
  - v. MAINTENANCE AND SUPPORT REQUIREMENTS. maintenance wait time, etc.
  - vi. OTHER REQUIREMENTS. Other requirements that may be relevant to your special software type. This may include support of multiple languages, personalization options per user (such as choice of currency), etc.

## ADDITIONAL NOTES

1. Points may be taken off for poor formatting, writing, and grammar.
2. Bonus up to 10 points will be given for creativity and originality showing you put in some extra effort.

# **Example**

# **Software Engineering 2555**

## **Software Requirements Specification Document Template**

---

**KMUTT Learning Environment**

**Beta Ver1.3**

**(The first step towards better student engagement and monitoring)**

**User Requirement Specification version 1.3**

<b>Version</b>	<b>Date</b>	<b>Descriptions</b>	<b>Name</b>
1.0	6 Jan 13	First Draft	KJ
1.3	18 Jan 13	Included 'learn' and 'teach' widgets	TA

## **1. INTRODUCTION**

---

It is KMUTT's mission to establish a high quality system aided learning in order to foster intellectual mobilization and collaborative lifelong learning culture. Next Generation learning Environment (KMUTT Lrn-Env) project is the initiative created in response to such a drive to establish a rich and diverse learning experience for KMUTT learning community.

Proposed as part of e-kmutt project, KMUTT Lrn-Env is a learning system specially designed to provide a comprehensive online learning environment that augments the traditional learning context. Once completed, the system will offer a unified point of access to a complete suite of web-based learning resources, materials, activities, tools and other related support services.

Developed during Phase#1 of KMUTT Learning Environment Project, LE Beta Version is aimed to provide a small and efficient set of learning tools for KMUTT learning community. The main concept of this Beta version is to provide a small student portal (a small-scale Learning Environment) with LMS and additional learning and organizing tools/services focused on evaluation and monitoring functionality.

From our user perspective, the tools will provide necessary functions of a traditional LMS with a very good student progress monitoring tool allowing social discussion-- among teachers about each particular student-- linked to a well-structured the student portfolio.

We are planning to offer smart and simple GUI with clean and functional main page, share section, drawer concept, and so on.

### **Access:**

- Option1: [www.kmutt.ac.th](http://www.kmutt.ac.th) then login to learning environment
- Option2: my.kmutt.ac.th -> directly go to learning environment

### **Access Validation:**

- User will be using CAS login (LDAP)
- They will need to provide a valid email account before they can use it.
- They need to identify their roles: e.g., teacher, TA, Students, etc.

## 2. Functional Requirements

### 2.1 Product Perspective

#### 2.1.1 Use Case Diagram

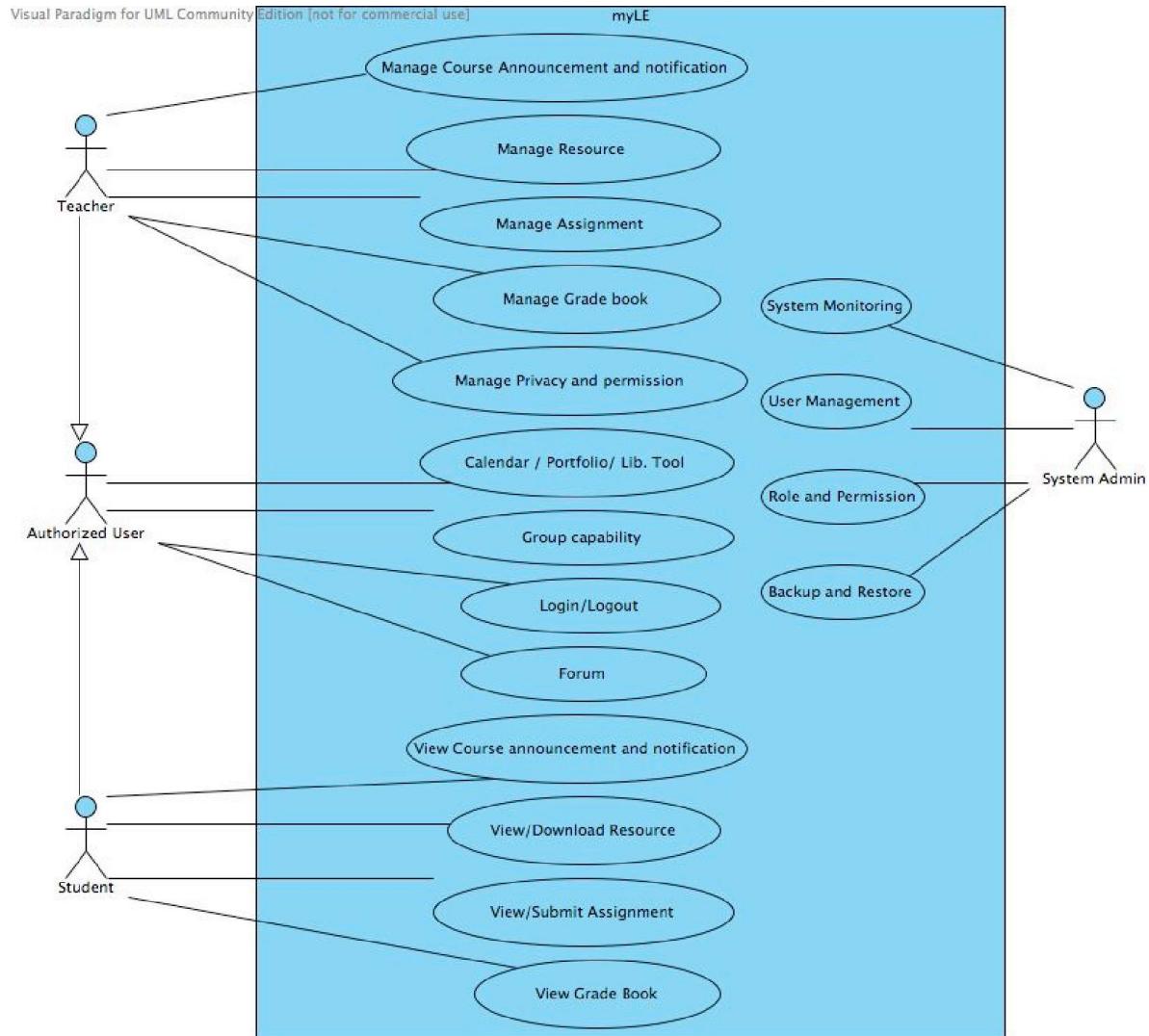


Figure 1. Use Case Diagram

#### 2.1.2 Description

MyLE is a system that provides the facilities to teachers and students. There are 3 types of users: teacher, student, and system administrator. Teachers can manage all the modules in specific courses, which include course announcement, assignment, grade book, and course resources. Students can only view the information in the course page, download/submit files from/to the course page and view their grade book. Notice that students have no right to edit/delete any information that is created by teachers. The system administrator is responsible for preserving the stability, reliability and security of the system. He/she can monitor the

system, manage users, manage the permission of each role, and backup/restore the data.

## 2.2 Complete Functional Requirements List

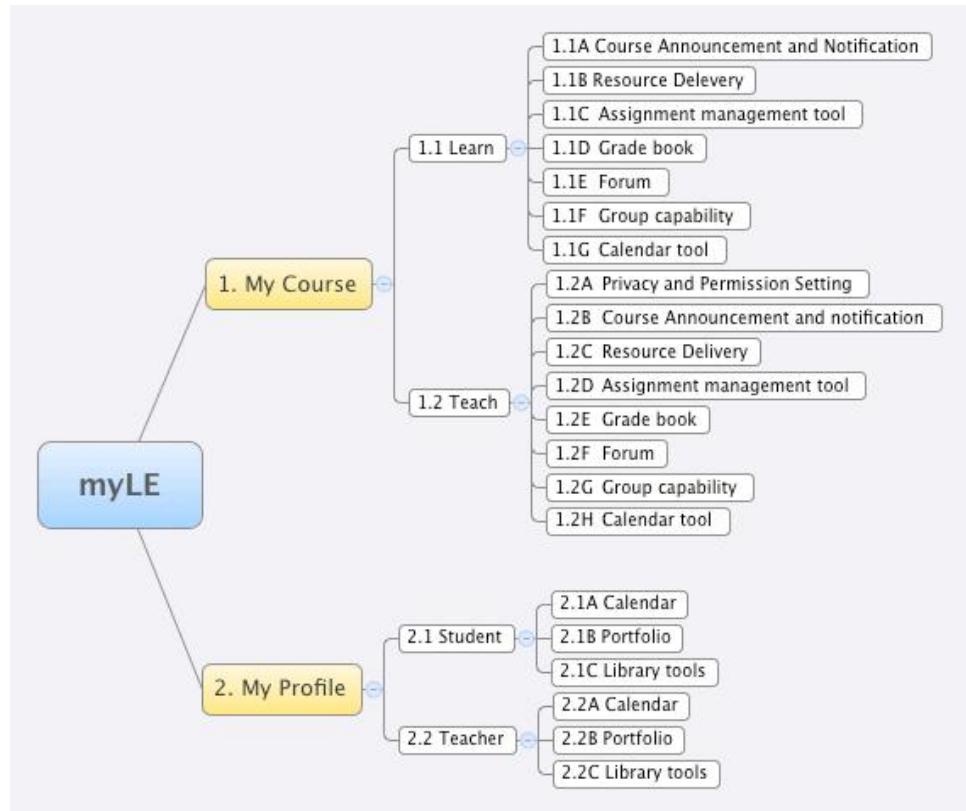


Figure 2. Overview of System

In regards to the figure 1, the system could be illustrated as 2 parts; *My Course* and *My Profile* which are explained in details as below.

### 2.2.1 My Course

My Course includes a collection of traditional and non-traditional learning and teaching tools related to a course. My course listing should list 2 views such as Student View, namely **Learn** and Teacher View, namely **Teach**. This section is a list of the course that the student is participating on (either by registration or by personal request to participate).

1. Learn (Student view) – this section a list of subjects enrolled, and tabs representing all related learning features of the tool.

Tab lists of Student view – The student view should include tools as listed below.

#### 1.1A Course Announcement and notification

- Facebook Synchronization (post to Facebook) is required for Course Announcement and notification message.
- Course Announcement message should also appear on the mouse over message at the desktop of LE (before clicking into the subject )
- Features can be filtered by course or display in overall view.
- Overall view is the landing page for Learn.
- Announcement categorizes its displayed contents as: course announcements, system announcements, coming assignments, and several latest sharing contents.
- Contents in the Announcement can be sorted by updated date (descending), and level of importance.
- An announcement with high level of importance is displayed more noticeable.
- Full list of system announcements and course announcements can be viewed in external display panels.
- Full list of announcements in external display panels can be sorted by date (ascending, descending), and name (ascending, descending).
- Can be searched and filtered by keywords.

#### 1.1B Resource Delivery – Course material uploading – content keeper.

- Features can be filtered by course or display in overall view
- The Lecture Note lists available materials (e.g. PDF documents, media files, You Tube videos) for view and download.
- Materials can be sorted by date (ascending, descending), name (ascending, descending), and type.
- Can be searched and filtered by keywords.
- Able to select views, i.e. Materials in each category, materials of each course, all materials in each semester, all materials, and etc. The widget should allow the students to sort materials by title, date, and category.

#### 1.1C Assignment management tool (assignment submission and organization)

- Features can be filtered by course or display in overall view.

- The Assignment lists assignments with detail. The detail could be in form of an article-like content and/or external files (PDF, media files, etc.)
- Each assignment provides an upload form for handing in.
- The Assignment displays score for submitted assignments after they have been graded.
- Assignments can be sorted by due date, updated date (ascending, descending), and name (ascending, descending).
- The Sharing Tools provides functions to create collaborative sharing contents, articles, and comments with others.
- Can be searched and filtered by keywords.
- The widget should calculates simple statistics of each assignment and present it to the students.

1.1D Grade book – Score announcement

- A collection of all score and comments for a student, Gradebook can be accessed, viewed by a student as part of the student portfolio.
- Features can be filtered by course or display in overall view.
- Can be searched and filtered by keywords.
- Shows score list of assignments, quizzes, and exams.
- Score list can be sorted by category (assignment, exam, etc.), name, updated date.

1.1E Forum or Wiki or any types of collaborative space for project work, knowledge sharing, and online collaboration activities.

- Can be searched and filtered by keywords.
- Contents on the Sharing Tools can be sorted by category, tag, updated date (ascending, descending), and name (ascending, descending).
- Contents from the Sharing Tools can be categorized and tagged for searching and displaying purpose.
- The Sharing Tools provides functions to create collaborative sharing contents, articles, and comments with others.

1.1F Group capability (we need this to support Gen-Ed subjects)

- This feature is provided to allow 'sections' within a subject for all group related learning activities. For example, some of the general education subjects are divided into different Sections (for International Students, group based, etc). Each section will be managed separately with different teacher, different assignment, allocated project and so on.

1.1G Calendar tool – provide a function to add course events into my calendar.

2. Teach (teacher view) – this section is a list of instructive subjects which the teacher is teaching.

Tab lists of teacher view - The teacher view should include tabs representing all related teaching features of the tool.

1.2A Privacy and Permission Setting (management tool for each course)

- Regular course admin stuff (permission, sections, rooms, lecture time and assignment schedule)

1.2B Course Announcement and notification

- Facebook Synchronization (post to Facebook) is required for Course Announcement and notification message.
- Teacher can post an announcement about the course and the announcement can be forwarded automatically to students' emails.
- Course Announcement message should also appear on the mouse over message at the desktop of LE (before clicking into the subject )
- Teacher can set a level of importance for each announcement so that the important ones will catch student's attention when they see it online or receive it through email.
- Announcement can be posted to some or all students (by class, by group, individual).
- Announcement should have display dateline. All announcements can be displayed and sorted by date, level of importance, view by course and view all.

- From the student view, the Announcement provides additional functions to manage (create, edit, delete) announcements.

1.2C Resource Delivery – Course material uploading – content keeper.

- Able to upload materials for students to download. Materials include e-book, lecture notes, papers, presentation slides, and all other supplemental readings. Most of the popular media files should be supported.
- Able to select views, i.e. Materials in each category, materials of each course, all materials in each semester, all materials, and etc. The widget should allow the students to sort materials by title, date, and category.
- A teacher can see which student download which material (not a must-have).
- Able to set up an assignment item (with attached file) for student. Student should be able to upload the assignment file.
- The Lecture Note gives functions to upload, download, delete, and distribute materials to all or selective target (group of students, individual student, etc.) in a course.
- The Lecture Note supports major file types (e.g. MS Office file, PDF, EPUB, video file) and YouTube video URL.

1.2D Assignment management tool (assignment submission and organization)

- The widget should calculates simple statistics of each assignment and present it to the teacher.
- Score announcement for paper-based exams or quizzes. Some exams or assignments cannot be given online, so the score tab should allow a teacher to enter the score through a GUI form as well as import the score from an excel file.
- The scores of all assignments and exams can be exported into an excel file in the format that can be easily adapted
- For all assignments and learning activity, my course will be equipped with a **Social Student Evaluation tool – in response to TQF and KMUTTQF** (linked to portfolio module in [2]) used to provide evaluation for each student both on

subject related assignment/task and on the overall skills received as an outcome of the course. \*

1.2E Grade book – Score announcement

- A collection of all score and comments for a student, Gradebook can be accessed, viewed by a student as part of the student portfolio.
- Can grade the assignment and post the score online. Assignment can be an individual or a group assignment. Teacher can give comments that are linked to student's portfolio (as discussed in the last meeting). Also the 'like' feature should be available.

1.2F Forum or Wiki or any types of collaborative space for project work, knowledge sharing, and online collaboration activities.

1.2G Group capability (we need this to support Gen-Ed subjects)

- This feature is provided to allow 'sections' within a subject for all group related learning activities. For example, some of the general education subjects are divided into different Sections (for International Students, group based, etc). Each section will be managed separately with different teacher, different assignment, allocated project and so on.

1.2H Calendar tool – provide a function to add course events into my calendar.

**Note:** \* These tools though shall not be included in traditional LMS features, they are subject specific and thus are listed under 'My course' for ease of understanding.

2.2.2. My Profiles – This section includes a set of facility tool such as Calendar, Portfolio and Library tool. Once again, these tools are divided into 2 views; Student View and Teacher View.

1. Student View

2.1A Calendar

2.1B Student Portfolio – available for public view

- Provide basic student information: name, Picture, ID, subject enrolled
- Learning progress status (Gradebook – Student View)

- A place for them to plan their IDP (Individual Development plan) for their future career.
- Showcase - They are able to upload their research work, publication, project work, training experience, media files. – A collection of their project experience, activities, etc.
- Award/point collection tool – a place where they are able to collect their award/points received from their subject specific and non-subject specific work.

### 2.1C Library tools

1. Basic Search Service from LE,
2. Text Book status update (subject specific )
3. List of loan items

## 2. Teacher View

### 2.2A Calendar

### 2.2B Teacher Portfolio

- Provide basic teacher information: name, Picture, ID, subject enrolled
- Teaching progress status (Gradebook – Teacher View)
- A place for them to plan their IDP (Individual Development plan) for their future career.
- Showcase - They are able to upload their research work, publication, project work, training experience, media files. – A collection of their project experience, activities, etc.
- Award/point collection tool – a place where they are able to collect their award/points received from their subject specific and non-subject specific work.
- Provide confidential feedback and notes about learning progress and issues need special attentions from previous subjects. (Teacher to Teacher note)

- Red Flag: warning system option for students required urgent attention. (In the future, ‘Red-Flag’ students should be identified and grouped separately under ‘My Course’.

## 2.2C Library tools

1. Basic Search Service from LE,
2. Text Book status update (subject specific )
3. List of loan items

### **3. Non-Functional Requirements**

---

#### **3.1 Operating Environment**

This system is designed to be compatible with the latest versions of web browsers as listed below.

- Firefox 12 and above
- Safari 5 and above
- Internet Explorer 8 and above

#### **3.2 Design and Implementation Constraints**

The system will be implemented using PHP on Drupal framework, which is the content management system. The requirements of system deployment are as follows:

- Windows or Linux operating system
- Apache or Microsoft IIS web server
- MySQL database
- PHP 5.3 or higher

#### **3.3 User Documentation Requirements**

Once myLE is completed and ready to use, it will be delivered with documents as listed below. Note that the documents will be provided in a hard copy format and will be available online.

- Help Module – This module is the list of frequently asked questions.
- User Manual – This document is a guideline for software usage, GUIs will be labeled and explained in details.
- Technical Manual – This document is a report of technical details including the maintenance guideline. It will be provided for further development and maintenance.

#### **3.4 Maintenance, configuration, Administration**

*Administration and Infrastructure*

- Authentication
- Scalability
- Basic Infrastructure
- Role & Permission
- Access Authorization
- Registration
- Stale Data Cleaning
- Backup and Restore

### 3.5 External Data Link

- Retrieve, transfer, exchange data between LrnEnv and other systems
- CC Service Synchronization
  - For *My Course [1]*, subject enrolled Subject name, enrollment status, profiles, etc.
  - For *Calendar tool [1.1G, 1.2H]*, subject enrolled, date/time/room for each subject
- Library Management Synchronization
  - Synchronize with library data such as searching, borrowed books, textbook status and etc.
- Facebook and other web tools

### 3.6 Teaching and Learning Assessment and Monitoring tools

Teaching and learning Assessment (Captured from system activities)– How the tool monitors user's behavior, provide behavior analysis from all system activities. (Please refer to A.songrit document for more technical details)

- User login / logout logging: Keep all the records of log in and log out activities
- User active status: Determine if user is currently using the system
- User login / logout monitoring: Monitor as well as manage login / logout activities
- Module usage logging: Keep all the records of module accesses.
- Assessment data analysis model: Model the analysis type for assessment data

- Assessment data analyser: Analyse assessment data using predefined model

### 3.7 Look and Feel of the Software: Example GUI

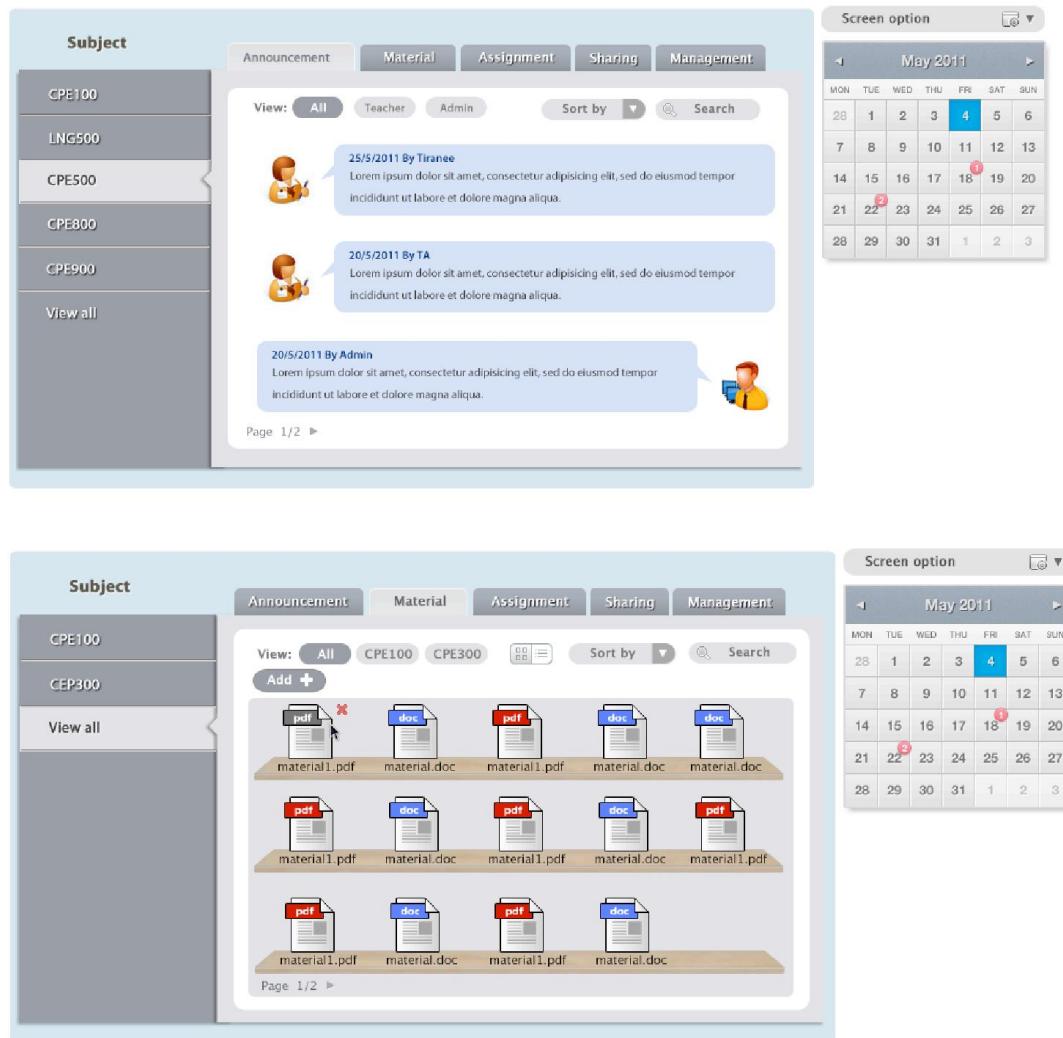


Figure 3. Example GUI