
Software Requirements Specification

for

CLASSROOM

Version 1.0 approved

Prepared by Abdullah Zameek,
Bokang Jia,
Michael Liu,
Branden Kang

Tethical Engineering

April 7th 2020

Table of Contents

Introduction	1
Purpose	1
Document Conventions	1
Intended Audience and Reading Suggestions	1
Product Scope	1
References	2
Overall Description	2
Product Perspective	2
Product Functions	2
User Classes and Characteristics	3
Operating Environment	3
Design and Implementation Constraints	3
User Documentation	3
Assumptions and Dependencies	3
External Interface Requirements	4
User Interfaces	4
Hardware Interfaces	4
Software Interfaces	4
Communications Interfaces	5
System Features	5
Course material	5
Course review	7
User credit	9
Course recommendation	10
Other Nonfunctional Requirements	11
Performance Requirements	11
Safety Requirements	11
Security Requirements	11
Software Quality Attributes	11
Business Rules	11
Other Requirements	12

Revision History

Name	Date	Reason For Changes	Version
Initial	01	Initial document	Version 1

1. Introduction

1.1 Purpose

This document provides a detailed outline of the requirements specifications of “Classroom - A platform for sharing course materials among peers”. Classroom is document platform that hosts course materials, initially targeted towards the NYU community. It will include features such as course reviews, syllabus, past assignments, and exams.

In this SRS document we will showcase the functional and nonfunctional requirements of the system, as well as the system boundaries and constraints. This document will serve as the primary form of specifications for the first version of Classroom (v1.0) for both the development team and end users. It will cover the full system, including all relevant sub-systems.

1.2 Document Conventions

The terms “product” and “platform” herein refers to the Classroom platform.

1.3 Intended Audience and Reading Suggestions

The intended audience of this document is primarily the development team of the project so that a clear understanding of the scope of the system can be obtained. For instance, development project leaders are able to gain a quick overview of the specifications to be implemented by reading section §4 which describes the system features. UI developers, for instance, can get a better understanding of the interface demands by analyzing section 3 that describes the UI needs. Furthermore, if any personnel were to take up or join the project at a later date, this document would be the primary resource of consultation to understand the system.

This SRS document contains a brief overview of the product, followed detailed system specifications broken down into sections. The suggested sequence for the document depends on the particular stakeholder that is reading the document. However, for most readers, it would be advisable to read the overview, and then depending on the task of the reader, jump to the pertinent sections.

1.4 Product Scope

Classroom is a platform that allows students to share course resources amongst themselves. The objective of this system is to act as a central repository for students to post course reviews, syllabuses, past assignment prompts and exams. The goals of this platform are to minimize data redundancy in these aforementioned resources being scattered across multiple platforms and to minimize multiple requests for access to these resources. Students would greatly benefit from this since they would all have equal opportunity to access these resources and would also allow for a further sense of community building.

This directly relates to the corporate goals of universities, which is to further the education of students and build a sense of community among peers. Classrook would achieve the first goal by enhancing the grades of students through increased access to academic resources, and it would achieve the second goal by eliminating unfair access to resources and improving

1.5 References

The user interfaces will be designed through the use of material-ui style guides. Material-ui has a number of design principles that define react components for faster and easier web development. Information about these standards can be found on <https://material-ui.com/>

In developing this SRS we gathered requirements from a number of places. One such place can be found in the Requirement Gathering document that describes the use cases we investigated.

2. Overall Description

2.1 Product Perspective

This project will constitute its own self contained product that will amalgamate features of several other platforms that could be considered as its rivals in the competitive landscape. This product is unique in that the primary end-user (i.e students) would only need to consult one platform for their course related needs, rather than needing to consult professors via email, or other social platforms and/or forums. The primary difference between this product and others is that it will be tailored to allow access to personnel of one particular institution only, i.e if the platform is installed for University X, then you need to be part of the University X's community to access it.

[INSERT ER OR WHATEVER DIAGRAM OVER HERE]

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

The primary functions of the system have been listed below:

- User registration to facilitate logging in and out.
- Customize user profiles.
- Allow for different access rights/privileges depending on the user class.
- Submit course reviews.
- Submit course material.
- Browse course material/reviews by course name/code or semester.

- Award credit to users for uploading course reviews/material.
- Allow downloading of material if the user has sufficient credits.

2.3 User Classes and Characteristics

There are two primary end users for this system:

- 1) Students
- 2) Professors

The back end facing user would be the system administrators (i.e the development team) who will deal with the maintenance related matters on the system.

These two user classes will both belong to the same institution, but will have different privileges on the platform. Students will only be able to upload and access course material depending on the amount of credits that they have. However, professors will be able to access all reviews and courses (for only the classes that they teach). If they wish for material to be removed, they may submit a request to the system administrators. These requests will be reviewed by the system administrators and depending on the nature and context of the request, action will be taken.

2.4 Operating Environment

The platform will be a progressive web application so that it is as cross-platform compatible as possible, and accessible from different web browsers.

2.5 Design and Implementation Constraints

The team only has one month to complete the product, so essential functionalities will be prioritized.

Admin will constantly monitor the legality of uploaded documents and reviews, and take down content when necessary.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

A user documentation will be provided in markdown format that provides instructions as well as screenshots on how to access the key functionalities of our website, including registering and logging in, searching for courses, uploading documents and posting reviews.

2.7 Assumptions and Dependencies

We assume that users are affiliated with at least one educational institution that issues educational email addresses which end with “.edu” domain.

3. External Interface Requirements

3.1 User Interfaces

A navigation bar is always displayed on top of all pages. The navigation bar consists of a thumbnail icon of the user profile image, which links to the profile page of that user, and a search interface. If the user is not logged in, a default user profile image will be displayed as a placeholder, clicking on which would allow the user to either login or create a new account. The search interface would allow users to search for the title of any courses listed within that affiliation.

The user profile page presents the profile picture, user name, university affiliation, and the current credits that user owns.

The home page will display the user's favourite courses, and the most popular courses, and allow the user to visit the information page of each course by clicking on the title of each course.

The information page of each course includes the detailed information of that course, the review section where most popular reviews are displayed, and a list of available documents. Information of a course includes: course number, course title, instructor, semesters when offered. Users will be able to set the course as favourite, post new reviews, upvote an existing review, upload new documents for that course, or download a document.

When uploading a new document, the user can set how many credits the document "worths"; when downloading a document, the user will be presented with an error message if he/she does not have enough credit to proceed.

Any request to non-existent or unauthorized webpages will be redirected to a 404 page.

3.2 Hardware Interfaces

Our website will support all the mainstream browsers. Web pages and the components displayed whereupon will automatically adjust according to the size of the physical device (phone, tablet, or PC) that is visiting our site.

3.3 Software Interfaces

Our system consists of a server node, and a database node. User requests are sent to the server, which records session information, with cookies to verify that the user is authorized to view requested information. The server will send requests to the database if needed, and return the information to users rendered as HTML pages.

The types of message that comes in into the system, the response, and their purpose are:

Messages in	Responses	Purposes
Username and password	Homepage	To login or to register account
Course ID	The information page of the corresponding page or 404	To view the review and uploaded materials of a course.
User ID	The user profile page or 404	To visit one's own profile page
File (in binary) and course ID	If uploaded success or not	To upload a document to a course
Review (string) and course ID	If the review is posted or not	To post a review to a course
File ID	File (in binary)	To download a file
Review ID	Updated upvote count	To upvote a review

The website will be hosted on one or more computing nodes on the Google Cloud depending on traffic. An additional node will host the database.

3.4 Communications Interfaces

Front-end framework will communicate with the backend framework using HTTP protocol with GET and POST Requests

4. System Features

4.1 Course material

4.1.1 Description and Priority

Users will be allowed to upload and download course materials on the information page of each course. Course materials are stored in PDF documents. Users will specify how many credits each document worths when uploading the documents.

When other users attempt to download a certain document, they will be asked to pay the price associated with the document in order to proceed. The credits paid will be rewarded to the original uploader. The backend will perform a sanity check to make sure that the same file is not being uploaded repetitively.

Component	Rating
Benefit	9
Cost	5
Risk	8

Overall Priority	High
-------------------------	------

4.1.2 Stimulus/Response Sequences

Use case 1: Uploading course material

User requests	System responses
1. Upload a document to a course.	1.1 Verify that the course ID is correct, and that the course exists.
	1.2 Receive the byte stream of the document and store the document on filesystem/database.
	1.3 Ask the user to set the price for this document.
2. User sets the price of this document.	2.1 Records the price of this document.
	2.2 Publish the document.

Use case 2: Downloading course material

User requests	System responses
1. Download a course material.	1.1 Verify that the user has enough credits to download this document.
	1.2 Transmit the requested file to the user.
	1.3 Deduct corresponding credits associated with the document from the downloader's account.
	1.4 Award the credits to the uploader.

Use case 3: Viewing all available course material

User requests	System responses
1. Requests the information page of a course.	1.1 Verify that the course ID is correct, and that the course exists.
	1.2 Get all files of this course from the database, as well as the metadata of each file.
	1.3 Send the list of document names and links to download to the front-end.
	1.4 Front-end only displays the 5 most downloaded documents, and a button to show more.

2. Requests to see all	2.1 Front-end displays all documents with the option to sort by metadata.
3. Sort	3.1 Sort the documents in descending order by the metadata specified by the user.

4.1.3 Functional Requirements

REQ-1: **File system/Database**. The uploaded documents should be stored in a file system or database, and organized by the course that those documents are associated with. Each document has a file name, and a unique identifier, which is formatted as courseID-fileID. The documents shall be backed up regularly.

REQ-2: **Cost**. Each document has an associated cost that is represented as number of credits in integer.

REQ-2: **Tag**. Each document has an associated tag (e.g. midterm exam, class note etc.) The tag is either user-defined or set by the system admin (TBD).

REQ-3: **Popularity**. The system records the number of downloads each document has accumulated.

REQ-4: **Sort by metadata**. The default metadata by which documents are sorted is popularity. Alternatively, documents can also be sorted by file size or file name.

4.2 Course review

4.2.1 Description and Priority

Users can post a new course review, edit an old course review posted by him-/her-self, or upvote reviews posted by other users. Users will be rewarded with credits when posting a new course review. Reviews will be displayed on the information page of each course. Reviews are sorted and displayed in descending order by number of upvotes by default.

Component	Rating
Benefit	9
Cost	2
Risk	7
Overall Priority	High

4.2.2 Stimulus/Response Sequences

User case 1: Display all course review

User requests	System responses
1. Requests the information page of a course.	1.1 Verify that the course ID is correct, and that the course exists.
	1.2 Get all reviews on this course.
	1.3 Send the list of reviews to the front-end.
	1.4 Front-end only displays the 5 most popular reviews, and a button to show more.
2. Requests to see all reviews	2.1 Front-end displays all reviews sorted by popularity with the option to sort by other metrics.
3. Sort	3.1 Sort the reviews in descending order by most popular, latest created, latest edited, or using a hybrid metric.

Use case 2: Post a course review

User requests	System responses
1. Post a review to a course.	1.1 Verify that the course ID is correct, and that the course exists.
	1.2 Store the course review to the database.
	1.3 Send the updated list of reviews to the front-end.
	1.4 Award user with credits.

User case 3: Edit a course review

User requests	System responses
1. Edit an existing review by providing the unique identifier for the review and the edit.	1.1 Verify that the review ID is correct.
	1.2 Update the course review in the database.
	1.3 Send the updated list of reviews to the front-end.

User case 4: Upvote a course view

User requests	System responses
---------------	------------------

1. Upvote a course review by providing the review ID.	1.1 Verify that the review ID is correct.
	1.2 Update the upvote count of this course review in the database.
	1.3 Send the updated count of upvotes to the front-end.

4.2.3 Functional Requirements

REQ-1: **Differentiate by review ownership.** Users cannot upvote the reviews they themselves posted, and users can only edit the reviews posted by themselves.

REQ-2: **Upvotes.** Each review has an upvote count.

REQ-2: **Sort by hybrid metric.** Newer reviews likely have less upvotes. So we need to design a metric that prioritizes both reviews with highest upvotes and latest reviews.

4.3 User credit

4.3.1 Description and Priority

Each user will have a credit balance recorded on his/her profile. Users start with an initial balance of 10 credits, and they can earn credits by posting course reviews and upload course materials. Downloading course materials costs credits.

Component	Rating
Benefit	9
Cost	2
Risk	2
Overall Priority	High

4.3.2 Stimulus/Response Sequences

Use case 1: Balance inquiry

User requests	System responses
1. Requests the user profile page of oneself.	1.1 Verify that the user ID is correct.
	1.2 Get the credits the user owns and send to the front-end.
	Alt 1.2: if the user ID is invalid, respond with

	code 404. If the user is not authorized to view the requested profile page, respond with authorization error.
--	---

Use case 2: Update balance

User requests	System responses
1. (System requests that) the balance of a user needs to be updated.	1.1 Verify that the user has enough remaining credits if the user wants to spend credits. If yes, proceed; if not, respond with error code.
	1.2 Update the updated balance in the database, and get the updated balance.
	1.3 Send the updated balance to the front-end.

4.3.3 Functional Requirements

REQ-1: **atomicity of balance update.** Balance ledger should be updated atomically.

4.4 Course recommendation**4.4.1 Description and Priority**

There are two types of course recommendations powered by machine learning algorithms: recommendations based on past courses taken (or reviewed), and recommendations by keywords. The first type of recommendation provides the user with a list of courses that the user might be potentially interested in based on the records of courses that the user has taken before. Until we allow users to explicitly record the courses they have taken in the past, the list of courses they have reviewed or have provided materials for will be a makeshift. The second type of recommendation is based on NLP analysis of all the reviews associated with a course. Users can assess this type of recommendation by querying keywords (for example, “useful skills”) or questions (for example, “Which courses give easy A’s?”)

Component	Rating
Benefit	3
Cost	9
Risk	2
Overall Priority	Low

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should allow multiple files being uploaded and downloaded in parallel.

The system should maintain at least 99% software availability throughout the year, and 100% software availability during the peak weeks. The system should also be flexible in capacity since we expect to see a surge in traffic happens around the time when course registration opens, and during the final week; accordingly, we expect few traffic during the school break. The system will scale horizontally prior to the expected peak weeks, and downscale afterwards to achieve maximum economic efficiency.

5.2 Safety Requirements

We encourage users to report any illegal content or remarks posted on the website, including copyrighted documents or books, libels, pornographic content, etc. The admin of this website should policy the uploaded content and reviews regularly, and take necessary measures to make sure the legality of anything posted on the website. They will take down the illegal material when found, and block accounts when needed.

5.3 Security Requirements

All passwords will be stored as salted hashes to minimize the likelihood of accounts being compromised. Furthermore, all data will be stored with confidentiality in line with the FERPA regulations that govern how educational institutions maintain software.

5.4 Software Quality Attributes

Although the platform is being developed for NYU Abu Dhabi, it will be easily adaptable to any college of a similar nature.

5.5 Business Rules

Verified professor accounts can perform unlimited download without being constrained by the amount of credits they have.

The credit system is designed to encourage users to contribute valuable resources to the website. Therefore, the value of credits, and ways in which one can earn credits should be made clear to users when they use the website for the first time.

6. Other Requirements

TBD

Appendix A: Glossary

Horizontal scaling means that you scale by adding more machines into your pool of resources

Appendix B: Analysis Models

TBD

Appendix C: To Be Determined List

- 4.1.3 REQ-2
- Use case of 4.4 and detailed functional requirements
- Other requirements
- Appendix B