Online Learning Platform (Google Classroom)

Supplementary Specification

Version 1.2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 27/Oct/2021 | 1.0 | Initial version. | Nguyễn Tuấn Nghĩa |
| 06/Dec/2021 | 1.1 | Added content | Nguyễn Tuấn Nghĩa |
| 25/Dec/2021 | 1.2 | Final Revision | Nguyễn Tuấn Nghĩa |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations 5

1.4 Overview 5

2. Functionality 5

2.1 User role (Teacher) 5

2.2 User role (Student) 6

3. Usability 6

3.1 Navigation 6

3.2 Consistency 6

3.3 Accessibility 6

4. Reliability 6

4.1 Availability 6

4.2 Mean Time Between Failures (MTBF) 6

4.3 Mean Time to Repair (MTTR) 7

5. Performance 7

5.1 Response time 7

5.2 Capacity 7

5.3 Resource Usage 7

6. Supportability 7

6.1 Maintenance 7

6.2 Extensibility 7

7. Design Constraints 7

7.1 Software Language 7

7.2 Architecture and Design 7

8. Online User Documentation and Help System Requirements 8

9. Purchased Components 8

10. Interfaces 8

10.1 User Interfaces 8

10.2 Hardware Interfaces 8

10.3 Software Interfaces 8

11. Licensing Requirements 8

12. Legal, Copyright, and Other Notices 8

13. Applicable Standards 8

Supplementary Specification

# Introduction

Google Classroom is a free blended learning platform for educational institutions that aims to simplify creating, distributing, and grading assignments. The primary purpose of Google Classroom is to streamline the process of sharing files between teachers and students.

## Purpose

## This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system’s scope

## Scope

The software architecture document covers all aspects of the system, both static and dynamic. The reference model is the view model, which includes many different views of the system, making the document complete and coherent.

The document examines class diagrams, package diagrams, and other static architecture designs under the system's static behavior. Use case realization and system sequence diagrams are used to expound on the system's dynamic characteristics.

## Definitions, Acronyms, and Abbreviations

DBMS – Database Management System.

MVC – Model view control architecture.

## Overview

The architectural representation of the project, including architectural representation, architectural aims and limitations, and use case realizations, are covered in other parts. The next sections go into the exact details of the four main viewpoints (logical view, process view, deployment view and implementation in further depth.

# Functionality

## User role (Teacher)

### The system shall allow the teacher to create class.

### The system shall allow the teacher to customize class.

### The system shall allow the teacher to create classwork

### The system shall allow the teacher to manage people.

### The system shall allow the teacher to grade student.

### The system shall allow the teacher to view schedule.

### The system shall allow the teacher to manage post.

### The system shall allow the teacher to archive class.

## User role (Student)

### The system shall allow the user to join a class.

### The system shall allow the user to take quiz.

### The system shall allow the user to complete assignment.

### The system shall allow the user to view schedule

# Usability

## Navigation

Use clear, easy-to-understand language, and links to the most important pages. It makes use of ample white space, color changes, or other design techniques to separate itself from the main content clearly.

## Consistency

* Have menus in the same position on each page
* Have same fonts and colors throughout the site
* Have a clear, visual hierarchy to the elements on your page
* Ensure that your business logo appears on each page
* Make sure that your logo links back to the homepage
* Include a search box on each page in the same location

## Accessibility

* [Choose a content management system that supports accessibility](https://webaccess.berkeley.edu/resources/tips/web-accessibility" \l "accessible-CMS)
* [Use headings correctly to organize the structure of your content](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-headings)
* [Include proper alt text for images](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-alt)
* [Give your links unique and descriptive names](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-links)
* [Use color with care](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-color)
* [Design your forms for accessibility](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-forms)
* [Use tables for tabular data, not for layout](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-tables)
* [Ensure that all content can be accessed with the keyboard alone in a logical way](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-content)
* [Use ARIA roles and landmarks (but only when necessary)](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-ARIA)
* [Make dynamic content accessible](https://webaccess.berkeley.edu/resources/tips/web-accessibility#accessible-dynamic)

# Reliability

## Availability

The program and all of its features will be available 99 percent of the time.

On Sunday morning, from 0:00 AM to 4:00 AM GMT+7, scheduled maintenance will take place during a four-hour period.

## Mean Time Between Failures (MTBF)

The system is allowed to get a mean time between failures at least one

months.

## Mean Time to Repair (MTTR)

When there is a system outage, the system support personnel should be alerted right away.

If the server hardware fails, the service should be restored within one hour, and external services should be restored within three hours.

# Performance

## Response time

The program must deliver a repeatable answer to the user, as well as acceptable response times.

## Capacity

More than 10,000 people are allowed to access the system at the same time

## Resource Usage

The software's size has yet to be determined, although it is expected to be between 150 and 250 megabytes.

To access the system, the client PCs must have a web browser installed. Except for the compilation of statistics reports, all functions will be processed in the backend.

# Supportability

## Maintenance

The program and all of its features will be available 99 percent of the time.

On Sunday morning, from 0:00 AM to 4:00 AM GMT+7, scheduled maintenance will take place during a four-hour period.

## Extensibility

The MVC architecture pattern is used in Google Classroom web application. The fundamental rationale for using this style is to layer functions, which improves maintainability and reusability.

# Design Constraints

## Software Language

Java is used to create the system's Models and Controllers, which handle business logic, as well as Views that make calls to the Models.

HTML, CSS, and REACT NATIVE — These are the languages that are used to manage client-side actions and events.

To access relational databases, MySQL – Structured Query Language is utilized. MySQL is used to execute all database server interactions.

## Architecture and Design

Model-View-Controller architecture will be used to develop the system. The overall design will be divided into three layers:

Views: The user-interactive screens and the components that process all of the data and show it to screens are clearly separated. The web server supports and provides material based on the demands of the user.

Models: In this layer, models represent all of the business logic. Models are mentioned, but they do not communicate directly with views. Models are pure Java objects that correspond to a single database entity in the database management system.

Controllers: This layer is in charge of how the Views and Models interact with one other. The controllers in this web application establish routes to handle requests and request models to retrieve suitable resources from views.

# Online User Documentation and Help System Requirements

Not required.

# Purchased Components

Not required.

# Interfaces

## User Interfaces

### 10.1.1 The Online Learning Platform will use a web-based interface for all interaction.

### 10.1.2 To access the Educational Management System, a secure user interface will be employed, which will involve the usage of a specified login name and password.

### 10.1.3 Users will be presented with an error web page indicating the error message if the system performs an unexpected operation.

## Hardware Interfaces

### 10.2.1 The system solely communicates with the web server and database server that are specified. The operating system and any other supporting software systems will handle any further system contact directly.

## Software Interfaces

### For database interactions, the system interface will connect with MySQL.

### Clients will receive HTML content from the system.

# Licensing Requirements

No special licensing requirements.

# Legal, Copyright, and Other Notices

The program policies listed below apply to content when using Google Classroom with a personal Google Account. This includes all Classroom features, such as course names, information shared in the class stream, communications and collaboration between teachers, students, and class materials. The program policies listed below play an important role in maintaining a positive experience for everyone using Google products.

# Applicable Standards

MVC and RUP standards should be followed in all documentation.