#### Honours Individual Project Dissertation



DEVELOPING A PEER CODE REVIEW SYSTEM

**Prateek Wadhwa**

June 25, 2021

Abstract

Education Use Consent

I hereby grant my permission for this project to be stored, distributed, and shown to other University of Glasgow students and staff for educational purposes. **Please note that you are under no obligation to sign this declaration but doing so would help future students.**

Signature: Prateek Wadhwa Date: 02 September 2022

Table of Contents

**Chapter 1: Introduction**

It is believed that the concept of peer review originated and was used for the evaluation of work in ancient Greece [6]. It has been described in detail in the book “Ethics of the Physician”, written by the physician Ali al Rahwi from Syria, explaining the peer review process's role in medicine [6]. It was only evolved to a more standardized structure since the 2nd World War [7]. The main grounds behind this evolution were attributed to the increase in scientific research [7]. In the modern world, the peer review process has been evolved to be standard and widely accepted form of assessment used in various disciplines such as Scientific Research, Education, and IT Industry for more than 50 years [8].

The application and role of peer review system varies with the discipline. In software industry, the peer review has been an effective way for over last 45 years to ensure the code is in line with the industry standard and remove any bugs, vulnerabilities [9][10]. In the field of research, peer review has been an industry norm to validate the research, maintain the quality of the work to meet the journal standard, etc. In the Education sector,

Needs some addition

In the field of education, the peer review process begins when a student submits his or her novel work for a particular assignment assigned by the course instructor (for instance a computer code, a presentation, or an essay). After the submission deadline expires, the submissions are distributed in a way, that each student is assigned one or more submissions as part of the peer review process. The students submit their reviews on the assigned submissions based on some criteria set by the course coordinator. The last step of the process is students submit their feedback on the peer reviews based on the level of satisfaction. In case of group submissions, a single student from each group submits the assignment on behalf of the group. Each group is assigned one or more group submissions in a way any group does not get their own assignment for review. A single student submits the peer review for the assigned group submissions.

The peer review process in education plays a vital role as a collaborative learning exercise that helps students develop the ability to critique other students' work and envision different solutions that develops creativity in them. It has been seen that including students as part of the peer review process helps develop social and communication skills by working in a collaborative environment through discussions with the other students [5]. With this process, the number of reviewers reviewing work increases, giving the student a better picture of the quality of their submitted work. It also reduces the workload of the course coordinators who must publish reviews for many students associated with a particular course within a tight deadline.

**1.2 Motivation**

**1.3 Aims**

This dissertation aims to design, develop, and document a peer code review system. The system should support two types of users, i.e., Professors and Students. The software system should allow the Professors to create courses and assignments and run peer reviews for the associated assignments. Using the peer review system, the students should submit the assignment with the supported format in a group or individually, view the assignment for review, and submit the peer review in a group or individually.

**1.4 Report Structure**

2 Background

Literature Review

The project will start with a review of the literature on peer review in education, particularly in programming, followed by a review of similar technologies. It will then use appropriate tools to develop the application and test and evaluate the final product

This section talks about in detail about the existing systems present in the industry.

2.1 PeerMark aka Turnitin

Peermark is a peer review tool provided by **Turnitin**. It is a well-known platform used as a plagiarism-checking service. Turnitin, an American company was founded in 1998 which provides software as a service (SaaS). The core functionality of Turnitin is to process the uploaded document by the author and compare it with the existing set of documents available in the database and mark it on a scale of 0-100 based on the plagiarism.

The key features of the PeerMark tool are as follows: -

* As it is a tool from Turnitin, there is an inbuild integration with the Turnitin system. The students can get a plagiarism report on their assignment submission.
* The tool supports formative feedback as part of peer review.
* It allows reviews on an assignment from peers, the course coordinator as well as self-review. This allows the reviews to be more fair and less bias.

2.3 Aropa

Aropa is a Maori term meaning to approach in a friendly way. The freely available online peer review system was developed by Dr. Purchase and Dr. Hamer in 2009. Over the last few years, it has been used by more than 30 institutions across the globe with reach its reach in countries like Australia, Canada, Malaysia, US, and Spain. This system has made constructive feedback possible in a large group of students.

Some of the key features of Aropa are :-

* Aropa allows for both group-based and individual peer review activities
* allows monitoring submissions
* monitoring reviewing
* manage student extensions
* reset passwords
* instructors can impersonate students.

Aropa has seen an iterative development process after receiving constructive feedback from individual instructors and students. It has updated the system in multiple ways in regard to the concerns about plagiarism, concerns of the students about the getting the right review and working on them to get an improved version each year of its existence.

2.4 Praze

Praze founded in 2007 is a web-based peer review application developed by Mulder

From University of Melbourne. The software system also known as APRES (Anonymous Peer Review and Evaluation system) originally supported the submissions of scientific work. The reviews to the submission were provided by 2 other students and their supervisor. The questions are pre-structured by the supervisor so that the reviewers can provide yes/no answers.

The key features of the software system are as follows:-

* It supports the group submissions.
* It allows a lot of customizable options to the course coordinators. For instance, 1) Add or remove students from the course or even the system, 2) Option to participate in the peer review process
* It uses a functionality called ‘Distribution’ link. After all the rules are defined on a particular assignment, it provides a ‘Distribution’ button that gives the option to the coordinator to run simulation on the distribution of allocation of students and their assigned peer review. This is visible to the coordinator in the form of a table, if the coordinator is satisfied with the configuration, they can choose to proceed further.

A case study was conducted on 14 student groups by University of Melbourne. The students were asked to build a web application and submit their respective prototypes designs for review on Praze. Each of the student got another design for a review. Based on the review, it was found that students were able to make changes in the UI design and solve the bugs before the final submission. The feedback from the students about the peer review process shows that 56% students found it ‘quite good, 14% students found it ‘great’, 9% students found it ‘ok’ and 23% of students found it not useful.

Overall, majority of the students found their experience using Praze useful.

# Requirements

The foundation of any functional software system comes down to its concise requirements that would enable the development of its component in an efficient manner. For this dissertation, the MoSCow prioritization technique was used.

## MoSCoW prioritization Technique

For the development of Peer Review System, The MoSCow Prioritization Technique was used which divides the requirements based on different criteria such as impact, budget and time constraint into 4 categories which are:-

* Must Have: - These requirements are linked with the core functionalities of the project without it the project would not work.
* Could Have: - These requirements are not part of the core functionalities but are still vital. Without it, the software system would be able to function and can be deployed to the next stage. But, these are desirable requirements that would add significant value to the overall scheme of things.
* Would like to have: - These are the types of requirements that can be classified as “nice to haves”, which won’t significantly contribute value towards. If there is availability of time and resources after the development of the must have, could have requirements, these requirements can be picked up.
* Won’t have: - This category defines the expectation from the project or software system by defining the scope of the project and stating the requirements that would not be covered for now.

## Functional Requirements

## The requirements can be classified into Functional and non-functional requirements. The functional requirements comprise of all the features or utilities the software system is expected to deliver.

## For Professor

##### Should Have:

* Create course with relevant credits.
* Create individual/group assignment with associated deadline.
* Create Questionnaire in a form of template for Peer Review.
* Export results for an assignment in csv format.
* Extend deadline for an assignment when necessary.
* Add students to a course.
* Form random groups for group submission.
* Delete courses
* Delete assignments
* Login
* Create a new group without causing issues to the existing groups. (Spreadsheet/UI)
* Time zone support for students who are studying remotely in different parts of the world.

##### Could Have:

* Add/remove a student in an existing group.

##### Would Like to Have:

##### Drag and Drop feature to import csv file to include students in a specific module.

##### Manually assign students in a specific group.

##### Manually move students from one group to another.

**For Students:**

##### Should Have:

* Login
* Submit Assignment with a particular extension before the deadline
* Able to submit peer review for an individual or group assignment.
* View Enrolled Courses

**Could have:**

* Send emails to alert students about deadline.
* Edit submissions before the deadline.

##### Would Like to Have:

##### Input support for different types of coding languages.

Software System

##### Should Have:

* Form groups based on an algorithm.
* Secure user data.
* Time zone support
* Assign submissions as part of peer review to students/groups randomly.

**Could have :**

* Manual table-based group formation algorithm.

##### Would Like to Have:

##### Cloud support.

##### Login via one time password.

## Non-Functional Requirements

The non-functional requirements consist of the traits of a software system it should have for smooth functioning and good user experience. For instance, responsiveness is one of the crucial non-functional requirements. If the web application takes a lot of time to respond, the user can shift to alternative service providers.

* Usability
* Security- Data Security
* Aesthetic appeal of the user interface
* Scalability- The application can take load without compromising the performance

# Design

## System Architecture

## Before designing a low-level UML diagram of the PRS, it is crucial to visualize a high-level architecture diagram showing the flow of the software system. Before digging deep into it, it is essential to know about each component.

## Web Browser: - It is a tool for communication of the users to the server. It runs on the user's web browser, which can be anything, for instance, Google Chrome or Mozilla Firefox, on any computer machine, be it macOS or windows.

## Server: - It contains the business logic of the software system. It is solely responsible for processing and validating user requests and blocking malicious, invalid requests. After processing, it fetches the relevant data from the database and returns it to the web browser, where all the information is visible to the user.

## Database: - It is the information storage unit that allows storing, updating, deletion and fetching of the data.

# Diagram Description automatically generated

Figure 1: Schematic representation of system architecture explaining the interaction between the different component of the Peer Review System.

## 

## As shown in figure 1,

## The system comes to live, when the user sends a HTTP request from its web browser through the internet towards the server which returns the data after fetching from the database. Since, the server is the heart of the software system, it is important to protect it from the malicious requests. The architecture is designed in a way to always support all the functional and nonfunctional requirements. For instance, the number of users accessing the portal or sever may increase during a certain period, the Heroku server launches more servers(nodes) to accommodate the users.

## 

## 

## 

Sequence Diagram: -

The Sequence diagram as shown in Figure 1 depicts the interaction between actors the **Student** and **Professor**, and objects, that are the **Server** and the **Database**. The interactions between the actor and the object are a depiction of the functionality of the respective actors. The order of exchange of messages between the actors and the object shows the typical flow of the software system.

The typical interaction between the actor and the object is described below: -

1. The Professor would be able to create a module through the user interface of the PRS running on the client side (Web browser). A subsequent POST Request would be directed to the server where it is processed. After it is validated, the server interacts with the database to execute the SQL Query containing all the necessary parameters of the module supplied by the Professor. The entry gets added to the database, and the result set is returned to the server. The server sends back to the client side the filtered result set with the status code 200, meaning it was a successful operation.

The high-level diagram captures the functionalities of the Professor or a student allowed in the PRS and the stages of these functionalities ordered sequentially. The functionalities are in line with the functional requirements.

Table

Description automatically generated with medium confidence

Figure 2: Schematic representation of sequence diagram explaining the interaction between the students, Professors with the server and the database

Class Diagram: -

The data flow model diagram captures the list of tables, its respective attributes, the attribute types, and the relationship between these tables. The database design has been designed in accordance with the rules of normalization to reduce redundancy and ensure data integrity. All the tables and their functionalities are described below :-

User:- It stores the details of both Student, and the Professor including the username and password. During the login, the details are fetched from this table to verify the details of the user.

Groupwork: - This tables stores the information related to the groups formed for the submission of the assignment as well as the submission of the Peer Review.

Assignment: - The tables stores all the necessary information of the Assignment that is associated to a particular module created by the Professor.

Module: - The table is responsible for storing and displaying information about all the modules that would be created by the Professor.

Submission: - All the student submissions be it individual or groups are stored in this table .

PeerReview: - This table contains the submission and the reviews against that particular submission for both Individual and Group Assignments.

Questionare: - It stores all the information related to the questions associated with the peer review with sequence.

QuestionareTemplate: - This table is the label of all the Questions associated with a peer review. The professor would be able to select the template during the creation of the Assignment, and all the questions associated with the template would automatically be part of the peer review.

StudentCourse: - The table contains the data about the enrolled courses for each student. When the

1. Implementation

This chapter would review and explain in detail about the different implementation strategies, technologies, practices, and libraries that were used to develop the final deployed version of the PRS.

* 1. Front End (User Interface)

The user interface was developed using Angular 13 that is originally based on JavaScript and is worldwide used for development of mobile and web applications due to its cross-platform support, modular code, and lighter applications. Angular supports various packages that need to be included in the package.json file so that it can be included in the project.

To maintain a common design pattern across the web application, Angular material package was used which provides consistent look and feel of the application. It provides matching CSS design for different types of inputs, tables, and other type of HTML tags.

Graphical user interface, application

Description automatically generated

Figure 3: Login view of the Peer Review System

The Angular Material handles the static end of the web application. For the web application to dynamic and interactive, it must interact with the backend which in turn would interact with the database for storing or fetching necessary information. Angular uses HTTPClient package that enables the interaction with the backend and makes the web application.

For figure 1, when the user enters the User ID and password in the input box and clicks on log in button, these details are attached with the http request towards backend endpoint that processes the information and takes an appropriate action of allowing or denying the user based on the credentials entered.

Text

Description automatically generated

Figure 4: Snapshot of the HTTP Login POST Request

For the population of modules or templates, an appropriate http request is sent to the endpoint URL handled by the backend that returns the relevant data in json data. Angular uses inbuilt ngfor functionality to replicate html tags based on the data received.

Database SQLAlchmey

Over the years, the databases have evolved from the tradional relational databases to noSql and GraphQL. Based on the use case and requirements, it is vey crucial to select an appropriate database that fits the software system architecture. For PRS, relational database was more suitable as it deals with organized,structured data with different relationships with other data points. It was crucial, that same information is displayed to all the users, hence choosing strongly consistent database was suitable. SQLAlchemy, a python library that is able to built tables and databases using python code was used for PRS. It converts the different function calls to sql queries automatically by using ORM(Object relational mapper), making it extremly flexible and uself. In addition to this, with backend handled with the flask framework in python, SQLalchemy also a python based library is compatible with the flask.

Text

Description automatically generated

Figure 5: the snapshot depicts how a table is created in python SqlAlchemy

Figure 5, shows a python class that uses the SQLAlchemy library to create a table named “course” as assigned using the attribute \_\_tablename\_\_. The function updateDetails is automatically converted by the object relational mapper to an update query which gets executed when the function gets called.

**Backend logic**

The backend is the brains of any software system that processess the incoming requests and takes the appropriate actions . For PRS, the backend is handled using python flask. Flask is a python framework that gives the developer wide variety of tools and libraries that are extremly useful to build web application. Flask is corely based on Jinja template engine and the Werkzeug WSGI toolkit[1].

As flask is a python based framework, it supports the use of pip which is a package manager that is extensively used to download third party libraries. For instance, PRS supports the functionality of reset password through email using the library flask\_mail that can be easily installed using the pip command by entering “pip install flask\_email” on the command line, making the installation of third party libraries quite convinient.

As shown in figure 1, the login http request sent is handled by the flask backend logic by assigning a function associated with a particular endpoint URL. Whenever a user attempts to login in the PRS, a POST request is directed towards this endpoint URL with the username and password part of the request body. The data is then validated in the flask login function, which is checked against the Database. If the username and password is matched, the user is allowed to login otherwise appropriate error message with the status code is returned to the end user.

Text

Description automatically generated

Figure 6: The code snapshot depicts how an login post request in handled in python flask

**Creation of Peer Review Questionare**

The functionality to create a questionnaire as part of Peer Review in an assignment is complicated multistep implementation. It can be broken down into a smaller stream of tasks. Firstly, depending on the type of user logs in, the headers visible to students and professors are different depending on the functionalities. The creation of the template is only allowed to the Professors user type, and students cannot access this functionality in any way. After the user would sign in through the portal, by clicking on the Peer Review Templates present on the Menu screen, The user would land on the Template component.

On clicking, the Add template Angular Material Button, an input Dialogue box Component is flashed on the screen. It is built using the Mat Dialog component from the Angular Material package. Instead of using the standard dialogue box provided by JavaScript which cannot be customized, This Dialogue box allows the developer to provide complete customization in Angular. The developer is expected to design the dialogue box through the HTML and CSS files in the new Angular Component. The customized Dialogue box expects the user to enter the template name, brief description, and the type of assessment it is being created for. After entering the details of the template, the save button triggers an HTTP POST Request in which the template details are part of the request body. Python Flask handles the post request at a particular endpoint, by validating the inputs and saving the results to the database.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 7, The snapshot shows the Angular material dialogue box

By clicking on the save button, the template gets created, and the control shifts from the Dialogue box component back to the template component. The user gets notified about the status of the creation of the template by Angular SnackBar from the Angular Material package. The library offers customized positions and notifications that can be configured very easily. The duration of the notification is set to 5 seconds which provides enough time for the user to notice. After the duration, the notification gets disappeared automatically. The user would now be able to see the newly created template and its detail in a table. The Angular Material package provides a separate library for the creation of a table. In the table, as part of each row, the user has been given the functionality to either update or delete the template if needed so.

The component depicts the instructions needed to add the Questions under this template, which can be done simply by clicking on the Template Name. The user is routed to the Questionnaire template. Similar to the previous component, the user can click on the Add question button, which would trigger a Dialogue box asking the user to enter the Question. By entering the question, the question is saved under the template in the database through an HTTP Post Request.

Table

Description automatically generated

Figure 8, The snapshot shows the questions in sequence for a template.

As shown in the figure 8, the sequence of the questions is maintained by the system itself. The PRS offers the user to delete or update the question, and the order of questions would be automatically adjusted for the user.

GitHub: -

For the development of the Peer review system, GitHub was used as the central repository for both backend and frontend code. It allows seamless development, rollbacks in case of failure, history of all the commits for reference and many more benefits. Different branches were used for the development of PRS. The main branch was used for the production environment, for every other functionality a unique number was assigned.

For the development of functionalities in the Peer Review System, every independent functionality was developed under a single branch name, while the dependent functionalities, were developed under a single branch name. In the end, a merge request was raised from each of these branches to the master branch. After reviewing all the changes, the branches were merged. This process ensured that the code present in the master branch is refined and bug-free. This further reduces the issues present at the production level, as the code reaches the master branch after some level of testing.

## 

References

[1]

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.214.9676&rep=rep1&type=pdf

[2] B. Cashell, W.D. Jackson, M. Jickling, and B. Web, “CRS Report for Congress: The Economic Impact of Cyber-Attacks,” Congressional Research Service, Apr. 2004.

[3] Nancy Falchikov. 2013. Improving Assessment through Student Involvement: Practical Solutions for Aiding Learning in Higher and Further Education. Routledge.

[4] John Hamer, Quintin Cutts, Jana Jackova, Andrew Luxton-Reilly, Robert McCartney, Helen Purchase, Charles Riedesel, Mara Saeli, Kate Sanders, and Judithe Sheard. 2008. Contributing student pedagogy. SIGCSE Bull. 40, 4 (Nov. 2008), 194–212. DOI:https://doi.org/10.1145/1473195.1473242

[5] Edward F. Gehringer, Donald D. Chinn, Manuel A. Pérez-Quiñones, and Mark A. Ardis. 2005. Using peer review in teaching computing. In Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE’05). ACM, New York, NY, 321–322. DOI:https://doi.org/10.1145/1047344.1047455

[9] M. Fagan. Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal, 15(3):182–211, 1976.

[10] M. Fagan. A history of software inspections. Software pioneers: contributions to software engineering, Springer-Verlag, Inc., pages 562–573, 2002.