Bilkent University
Department of Computer Engineering

CS 319 Object-Oriented Software Engineering
Term Project

SNITCH

CS319: Peer Review App: Snitch

Design Report

Group Members:
Ümit Çivi 21703064
Onat Korkmaz 21704028
Abdulkadir Erol 21703049
Erdem Ege Eroğlu 21601636
Muhammed Maruf Şatır 21702908

Instructor: Eray Tüzün
Teaching Assistant(s): Elgun Jabrayilzade, Erdem Tuna, Barış Ardıç

Progress/Design Report IT1
28.02.2021
This report is submitted to the Department of Computer Engineering of Bilkent University.

Table of Contents

1.Introduction

1.1 Purpose of the system

The purpose of the system is to provide an all-in-one system that will operate peer reviews in a group project. The system provides students to scale and comment on peers of the same group. Students will be able to view other groups' artifacts and comment on the artifacts. Hosts and TAs can categorize, arrange or rank the reviews after the process is over. Moreover, in this system, hosts will be able to delete groups, change groups or assign students to groups.

1.2 Design Goals

1.2.1 Functionality

In addition to the basics of the peer review system, we will try to implement additional functionalities. For instance, in the case of a collapsing database, we will send email that will include the information about grades every 4 hours on grading day.

1.2.2 Performance

The peer-reviewing system will have a limited number of users which is about 175 users. There will not be mass throughput so the performance of the system will not be highly affected by the execution of the tasks at the same time. In terms of response time, the peer-reviewing system will have a response time maximum of 0.25 milliseconds.

1.2.3 Dependability

The peer-review system will have great durability as the system will be not affected by the wrong user input since the system will display warning messages. The system will be able to provide service every time. The system will not threaten human lives and will not have negative effects on the environment. Moreover, the reliability of the system will not be an issue, the system will provide reliable information both while storing data and retrieving data from the database for display results. The language decision which is PHP will also assist the reliability of the system.

1.2.4 Cost

There will not be an explicit cost because the time that we spend on this project will be for the course. Moreover, we will use a free database.

1.2.5 Backward Compatibility

As we will not update the system, we will not show great effort to have backward compatibility.

1.2.6 Rapid Development

The aim of our team will be certain to catch up with the demo deadline. As 5 people in our team, we will start coding after the demo presentation. Firstly, we will try to implement the basic functionalities of the peer review system such as creating groups, adding a student to particular groups, and certainly peer grading-commenting. If there will be sufficient time to deadline, we will try to implement additional functionalities to our peer review system.

1.2.7 Readability

We do not have enough clue about others' style of coding. Hence, we will use meaningful names to identifiers and comment on the functionalities of the program in crucial points.

1.2.8 Ease of Learning

In the peer review system, we had decided to make the GUI of the system simple and easy to understand. We will make this aim possible by making the button names reflecting their functionalities, sorting will be indicated by the choosing specific checkbox, commenting and grading will include just the choosing peer, grading and commenting on him/her.

1.2.9 Usability:

As we will develop this as a web based application, a device that has access to enter websites, can use the system as their input devices. System will provide valid output to the interaction of users to the elements of GUI.

1.3 Trade-Offs

In this part all of the design goal trade-offs outlined will be addressed on the basis of our course contents. We will be declaring two opposing design goals in terms of not only importance but also reasons why we have decided to do so.

1.3.1 Functionality and Usability

We are planning to focus on the usability side. This is mainly because we want to implement a peer review system that is quite easy to use and our users are able to complete their work quickly and correctly. We really care about our users' time. Thus, this usability advantage will increase our users' productivity.

1.3.2 Cost and Robustness

Since this project has a large contribution to the letter grade of the course with 4 credits, we do not care about cost. We will be always trying to increase robustness. Rather than cost, we will spend more time on our project.

1.3.3 Efficiency and Portability

We will use PHP as our programming language. Since PHP is platform independent, we do not need to focus on portability. We will build a web application but, platform dependent means same code may not run on all browsers. Thus, that is not a design goal for our. We will be focusing on efficiency.

1.3.4 Rapid Development and Functionality

According to priority we might cut-off some functionalities which are labeled as low priority in order to increase the pace of development and maximize our cost-effectiveness. That is why we are planning to focus on the rapid development side.

1.3.5 Cost and Reusability

We want to implement a peer review system that is usable for a long term. After this semester, coming cs students should use our peer review system to finalize their projects. Therefore, in this trade-off we will focus on reusability.

1.3.6 Backwards Compatibility and Readability

We do not have a previous peer review project; therefore, backward compatibility is not a design goal for our project. That is why we will automatically be optimizing our readability. Since the project requires group work, we need to focus on readability because all group members ought to understand all pieces of code.

1.3.7 Security and Usability

In this trade-off, we will focus on security for the consistency of grades and we do not want people out of the course. Also, grades should be private to anyone. That is why we need to optimize security.

1.4 Definitions, Acronyms, and Abbreviations

In this section acronyms, abbreviations which we will use in our peer review system project is defined.

1.4.1 Personal Computer (PC)

A personal computer in short PC, is a multi-purpose computer that is intended for personal use.

1.4.2 Graphical User Interface (GUI)

Graphical User Interface in short GUI, is any user interface that makes the interaction with the user possible.

1.4.3 Model View Controller (MVC)

Model view controller in short MVC, consist of three parts. Those parts are Model, View, and Controller. Actual entities and objects compose of model parts. View part is composed of objects having to do with any user experience. Control objects compose to control parts.

1.5 Overview

In our peer review project, we have three main design goals that we will be focusing on mostly. Those design goals are reusability, usability and robustness. In terms of trade-offs, we will be optimizing usability as opposed to functionality, robustness as opposed to cost, rapid development as opposed to functionality, reusability as opposed to cost, readability as opposed backwards compatibility, and security as opposed to usability.

2. Current Software Architecture

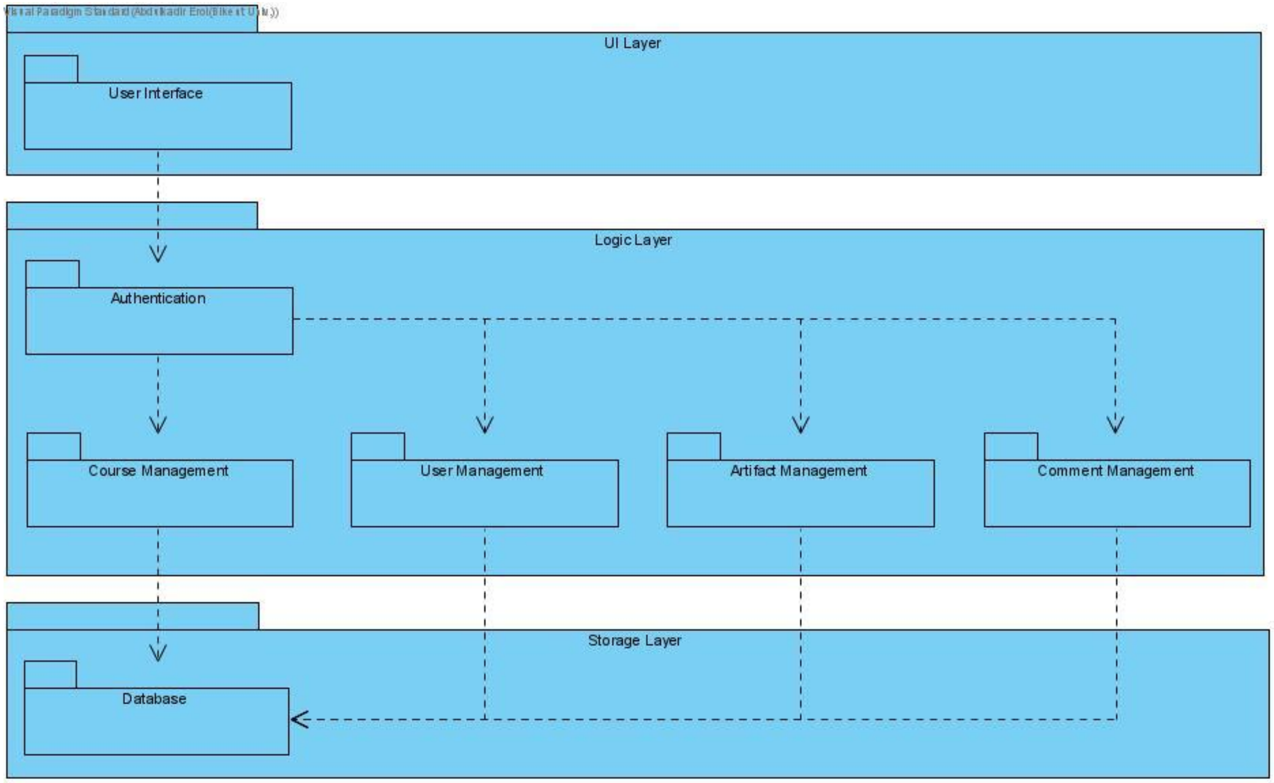
There is no current architecture since we have started our software from the beginning.

3. Proposed Software Architecture

3.1 Overview

Our system is a web application and the MVC (model view controller) architecture will be used for decomposing our system into subsystems. All the changes and related adjustments will be saved in the database. No additional hardware except keyboard and mouse will be needed to run the system. HTML, CSS, and PHP will be used to create the system.

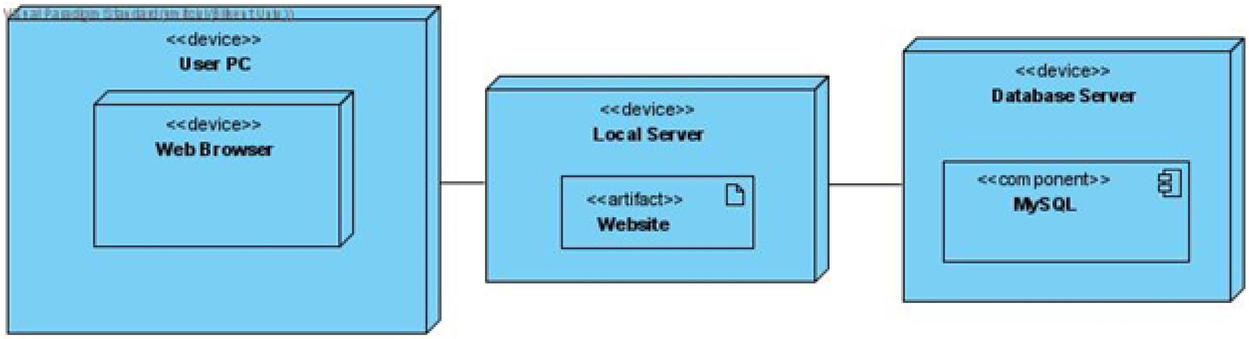
3.2 Subsystem Decomposition



3.3 Hardware/Software Mapping

The project will be implemented with PHP. The user is required to have an internet connection and a web browser to run the website. The project requires a mouse in order to navigate inside the website and a keyboard to enter the comments regarding an artifact or students.

For storage, we will use a database. Therefore, there will not be any hard disk usage in a PC except the local server.



3.4 Persistent Data Management

For persistent data management, MySQL (a relational database) will be used to store and manage the data. Since the group members do not know how to use MySQL, phpMyAdmin, which is an administration tool for MySQL databases, will be used. The database will store the data such as student/host accounts, artifacts, peer reviews, artifact reviews, groups etc. In order to increase the security of the system, only a few people will access the database using authentication.

3.5 Access Control and Security

We will be using network components for our project. That makes our peer review open to attack and difficult to control. Initially, our system allows only bilkent web mail. Therefore, we will not be storing sensitive data and there will be some features to keep the grades unchanged. Initially, our system will accept new users with e-mail configuration, an account of these features, there will be no one out of the course and each student can have a single account in our system. We will be using a remote database, in order to prevent grades from changing, students will have limited time to grade their peers and they start their own grading period on their own thanks to email verification. Every 4 hours of grading day and at the end of the grading time period TA’s get an email about grades. Grades cannot be changed, so If there are differences with e-mail that is sent to TA’s and the current system database. Grades at E-mail will be valid. Also, one student can't see another's comment until the teacher allows.

3.6 Global Software Control

In the peer review system, we had decided to use an event driven control mechanism. Since the peer review system has to interact with one user at a time and events will not be concurrent, the usage of event driven control will assist us to make simpler the interaction between user and the system. Peer review system will react to the events with the order of their arrival time. Moreover, the event handler mechanism will detect the movement of the user such as clicking the mouse and pressing key.

3.7 Boundary Conditions

3.7.1 Initialization

This is a Web application, so for a user to initialize it, s/he is required to have connection to the Internet. After the user is accessed to the app, the first thing that comes up on the interface is to register or log-in. Entering invalid login information will result in unsuccessful initialization of the app. Until the user enters valid login information, the interface will keep asking for email and password. Another scenario is when the user presses the sign-up button, s/he should have a valid Bilkent mail account. For authentication of the student, a mail address with “bilkent.edu.tr” extension should be given with a valid password (minimum 8 characters, uppercase and lowercase letters etc.).

3.7.2 Termination

When the close button is pressed, the user is logged off from his/her account and any unsaved changes will be lost. Also, there will be an exit button to sign off, and there will be a warning message to remind users to save the data.

3.7.3 Failure

Internet connection is required to access the application. Internet connection loss during peer review or artifact review will cause the deletion of unsaved data. Therefore, it is important to save the changes made by the users before exiting the application.

In all cases where data is saved to database or fetched from database for the first time, the connection to the database must be established. In a case in which the connection cannot be established, the user will be notified about the problem. Moreover, the user will be prompted to perform the unsuccessful task again.

4. Subsystem Services

User Interface

The User Interface subsystem is a part of UI Layer. It provides each user with a predefined interface depending on their roles, which are hosts (instructor & teacher assistant) and student. From here the users will have access to the functions according to their roles. Authorization services are required to login, sign-up and display the interface. It provides communication between the users and the rest of the subsystems.

Authentication

Authentication subsystem is the initial subsystem of Logic Layer and it provides an access to other Logic Layer subsystems. Also, handles user registration if the user does not have an account. With this subsystem, a layer of security is established. This subsystem works together with the Database subsystem to authenticate users.

Course Management

The Course Management subsystem is a part of Logic Layer. It provides any functions related to course and the sections.

User Management

The User Management subsystem is also a part of Logic Layer. It provides any user related functions in the system. Any functions done by hosts (instructor and teaching assistant) and students related to changing user specifications are in this system.

Artifact Management

The artifact management subsystem is also a part of Logic Layer. It provides artifact related functions.

Comment Management

The Comment Management subsystem is also a part of Logic Layer. It provides comment related functions. Comments include the reviews about the students or a particular artifact.

Database

The Database subsystem is a part of Storage Layer. It provides storage and retrieval to the whole system. All other subsystems interact with the database subsystem to fetch or store crucial data.