Bilkent University

Department of Computer Engineering

Senior Design Project

The Game – Code Review

Analysis Report

Burak Erkılıç	21501035
Mert Sezer	21400246
Okan Şen	21202377
Berk Yıldız	21502040

Supervisor: Halil Altay Güvenir
Jury Members: Shervin Rahimzadeh Arashloo, Can Alkan

Progress/Final Report
Nov 11, 2019 (Last Update: December 31, 2019)

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction

Code review is an important necessity for software development however is seen as a bothering task for most of the software developers. One of the reasons for that is, usually, code review is not taught as a requirement in the development process to students or if it taught as a requirement, the practical approach is missing toward the code review. To change this approach, we are providing a code review game that will gamify the code review especially for the students who are learning software development. Our goal is to adopt the habit of code reviewing to the coders of the future starting from the first steps of education.

Writing code or reviewing one can be quite tedious and energy-draining at times and this tiresome process can result in revisions or codes to be faulty and even inefficient. The time these actions require grows exponentially into tremendous amounts as the complexity of the work increases. In order to soften the burden of these works, The Game comes up with several innovative ideas. The classic, repetitive methods can be gruesome and cause disinterest within coders which we want to avoid. The Game focuses on creating a more appealing environment to learn about reviewing codes using game modes such as; challenges within classes or in a single form, defect detection. There will be teacher and student user types. Teachers will create and post new challenges thorough pathways and codes to be reviewed, while the students will have to work their way around these challenges and find the defects in these faulty codes. The Game can work as a social/professional platform bringing many likeminded people together, helping them to find friends to discuss code related topics or to find group members for their projects.

In this report, the proposed system of the program will be projected by indicating and analyzing functional requirements, non-functional requirements and pseudo requirements. In addition to these system models of the program will be provided by diagrams and screen mockups. Finally, analysis elements like possible risks, project plan, ethics & professional responsibilities, and learning strategies will be examined in the report.

2 Proposed System

2.1 Overview

The Game – Code review is a web application that allows teachers to provide code review exercises for their students. The aim of the program is to make code review a natural member of the education process by using gamification methods.

In the gameplay, the teacher creates a challenge by uploading a code script to the system by indicating the name of programming language and specifying the defects on the code. Then he/she set a due date and a timer for the challenge. Students can enter the challenge individually. At the end of the challenge, scores of the participants are seen on a table.

Especially for freshmen and fresh graduates, code review can be seen as unnecessary and extra; however, there is a need for code review for every software project for bug reducing and clean coding. So, freshmen will adopt the habit of code reviewing by our application at the beginning of their education.

2.2 Functional Requirements

- The Game will include two actors:
  - **Teacher –** the one who posts the challenges
  - **Student –** the one who solves the challenges
- All actors should login to the game by their saved e-mail addresses and passwords. The accounts which are not logged in will not be allowed to use facilities of the game
- The Game will have three main features:
  - **Pinpoint Defects (Student)**

    The goal is to mark defects and to do that the player has to follow some steps. The player has to be sure that there is some code selected in the code area. The player can click the "Pin" button to mark the excerpt as a defect. In the code, the excerpt will be highlighted and the defect will be added to the list in the sidebar, having a border with a certain color according to the selected severity and the description equal to the selected type.
  - **Pick & Unpick Defects (Student)**

    A player simply picks or unpicks a defect according to the available option for the defect that depends on its current state. If the defect is picked, it will be colored on the list and marked on the code in a bright color, with the available option being to unpick. If it’s unpicked, it will be gray on the list and marked with a darker color on the code, while the available option is to pick the defect.

- **Create Pathway (Teacher)**

  Teacher creates pathways which include several challenges in different levels. Teacher will assign created pathways to classrooms. Students won't be able skip a level without solving the current one.

- The teacher will upload the code script or create pathway by indicating its programming language and enter the defects into the system.
- The teacher will set a timer for each challenge he/she created.
- At the end of each challenge, rankings and statistics will be shown in a table and winner of the challenge will be stated. Class results will be visible to all participants of classroom
- Badges will be available for students. The badges won by the respective student will be displayed.
- All the users will be able to preview and edit their profiles. Also teacher will be able to review pathways and code scripts, students will be able to preview previous code review results and class rankings.

2.3 Non-functional Requirements

- **Portability:** The Game will be a web program and going to work in all browsers with any operating system on desktop.
- **Usability:** The Game will have a simple user-friendly interface. A user will be able to use all facilities in his/her first meet with the program by our basic navigational path.
- **Privacy:** All users should login to system for using the facilities. All personal data, e-mail addresses and password will be protected and will never share with any third parties.
- **Availability:** There is no need of extra installation to run program. Program is accessible from all type of internet browsers.
- **Scalability:** System will be able to respond several requests because especially in class work, several requests will arrive concurrently.
- **Performance:** System will respond to user in acceptable amount of time (less than a second) for any sort of request.

2.4 Pseudo Requirements

- The Game will be a desktop web program. Internet will be needed to access.
- Frontend will be implemented with HTML5, CSS, Bootstrap and AngularJS.
- Backend will be implemented with Spring Java.
- Embedded Spring Hibernate integration will be used for database.
- GitHub will be used for source code management.

2.5 System Models

2.5.1 Scenarios

Scenario 1

Use Case Name: Login

Participating Actor: Student

Entry Condition: Student is on Login Screen

Exit Condition: Student is on Home Screen

Main Flow of Events:

1. Student enters his/her name.
2. Student enters his/her password.
3. System directs user to Home Screen.

Alternative Flow of Events:

Student enters his/her password wrong in step 1 or 2.

3. System asks the user his/her name and password again.

Step 1 and Step 2 from the main flow of events need to be repeated in this case.

Scenario 2

Use Case Name: Register

Participating Actor: Student

Entry Condition: Student does not have an account.

Exit Condition: Student has an account.

Main Flow of Events:

1. Student enters his/her name.
2. Student enters his/her password.
3. Student clicks "Register".

Scenario 3

Use Case Name: Preview Profile

Participating Actor: Student

Entry Condition: Student is on Home Screen.

Exit Condition: Student goes back to Home Screen.

Main Flow of Events:

1. Student clicks to Preview Profile.
2. Student preview stats and do one or more of the following:

- Join or view class
- Set cosmetics
- Edit Profile Information

3. Student clicks "Save".

Alternative Flow of Events:

1. Student clicks to Preview Profile.

2. Student does not preview stats and do one or more of the following:

- Join or view class
- Set cosmetics
- Edit Profile Information

3. Student clicks "Save".

Alternative Flow of Events:

1. Student clicks to Preview Profile.

2. Student does not set cosmetics, edit profile information and
join class. (Student previews stats or not and student views a class or not).

3. Student goes back.

<u>**Scenario 4**</u>

Use Case Name: Preview Pathway

Participating Actor: Student

Entry Condition: Student is on Home Screen.

Exit Condition: Student goes back to Home Screen.

Main Flow of Events:

1. Student clicks to Preview Pathway.

2. Student does one or more of the following:

- Preview class ranking
- Preview personal attributes.

3. Student goes back to Home Screen.

Alternative Flow of Events:

1. Student clicks to Preview Pathway.

2. Student does none of the following:

- Preview class ranking
- Preview personal attributes.

3. Student goes back to Home Screen.

<u>**Scenario 5**</u>

Use Case Name: Solve Code Review

Participating Actor: Student

Entry Condition: Student is on Home Screen.

Exit Condition: Student goes back to Home Screen.

Main Flow of Events:

1. Student clicks to solve a code review problem that is posted by teacher.

Scenario 6

Use Case Name: Preview Code Review Solution

Participating Actor: Student

Entry Condition: Student is on Home Screen.

Exit Condition: Student goes back to Home Screen.

Main Flow of Events:

1. Student clicks to preview a code review solution.
2. Student goes back to Home Screen.

Scenario 7

Use Case Name: Prepare Review Bank

Participating Actor: Teacher

Entry Condition: Teacher is on Home Screen.

Exit Condition: Teacher goes back to Home Screen.

Main Flow of Events:

1. Teacher clicks to Prepare Review Bank.
2. Teacher does one or more of the following:

- clicking to set a Java review code
- clicking to set a C++ review code.

3. Teacher goes back to Home Screen.

Alternative Flow of Events:

1. Teacher clicks to Prepare Review Bank.
2. Teacher does none of the following:

- clicking to set a Java review code
- clicking to set a C++ review code.

3. Teacher goes back to Home Screen.

Scenario 8

Use Case Name: Prepare Pathway

Participating Actor: Teacher

Entry Condition: Teacher is on Home Screen.

Exit Condition: Teacher goes back to Home Screen.

Main Flow of Events:

1. Teacher clicks to prepare pathway.
2. Teacher does one or more of the following:

- clicking to prepare a Java pathway
- clicking to prepare a C++ pathway

3. Teacher goes back to Home Screen.

Alternative Flow of Events:

1. Teacher clicks to prepare pathway.

2. Teacher does none of the following:

- clicking to prepare a Java pathway
- clicking to prepare a C++ pathway

3. Teacher goes back to Home Screen.

Scenario 9

Use Case Name: Manage Classes

Participating Actor: Teacher

Entry Condition: Teacher is on Home Screen.

Exit Condition: Teacher goes back to Home Screen.

Main Flow of Events:

1. Teacher clicks to manage classes.

2. Teacher does one or more of the following:

- clicking to create a class
- clicking to preview a class
- clicking to edit a class
- clicking to assign a pathway

3. Teacher goes back to Home Screen.

Alternative Flow of Events:

1. Teacher clicks to manage classes.

2. Teacher does none of the following:

- clicking to create a class
- clicking to preview a class
- clicking to edit a class
- clicking to assign a pathway

3. Teacher goes back to Home Screen.

2.5.2 Use-Case Model

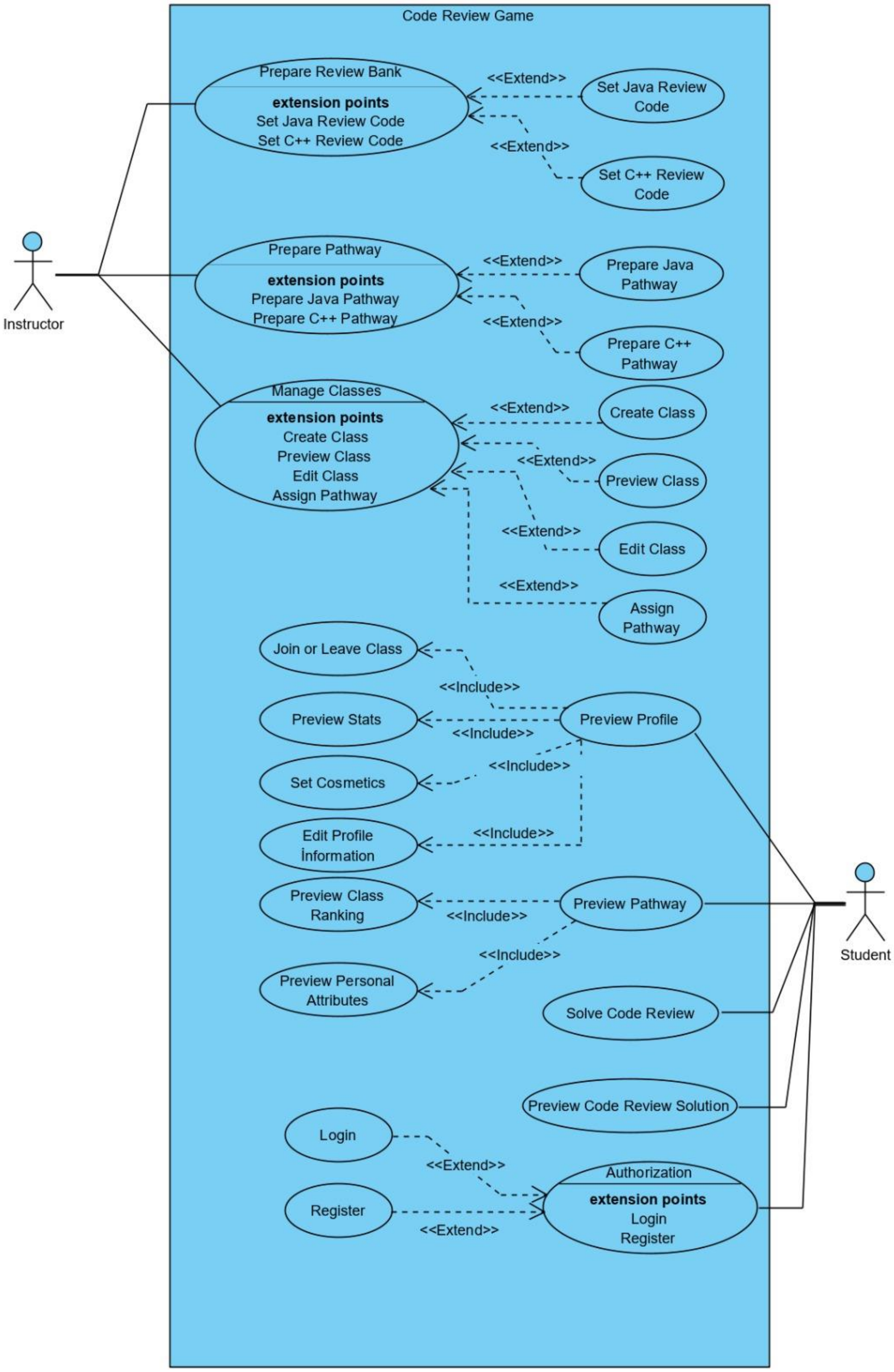


Figure 1: Use Case Model

2.5.3 Object and Class Model

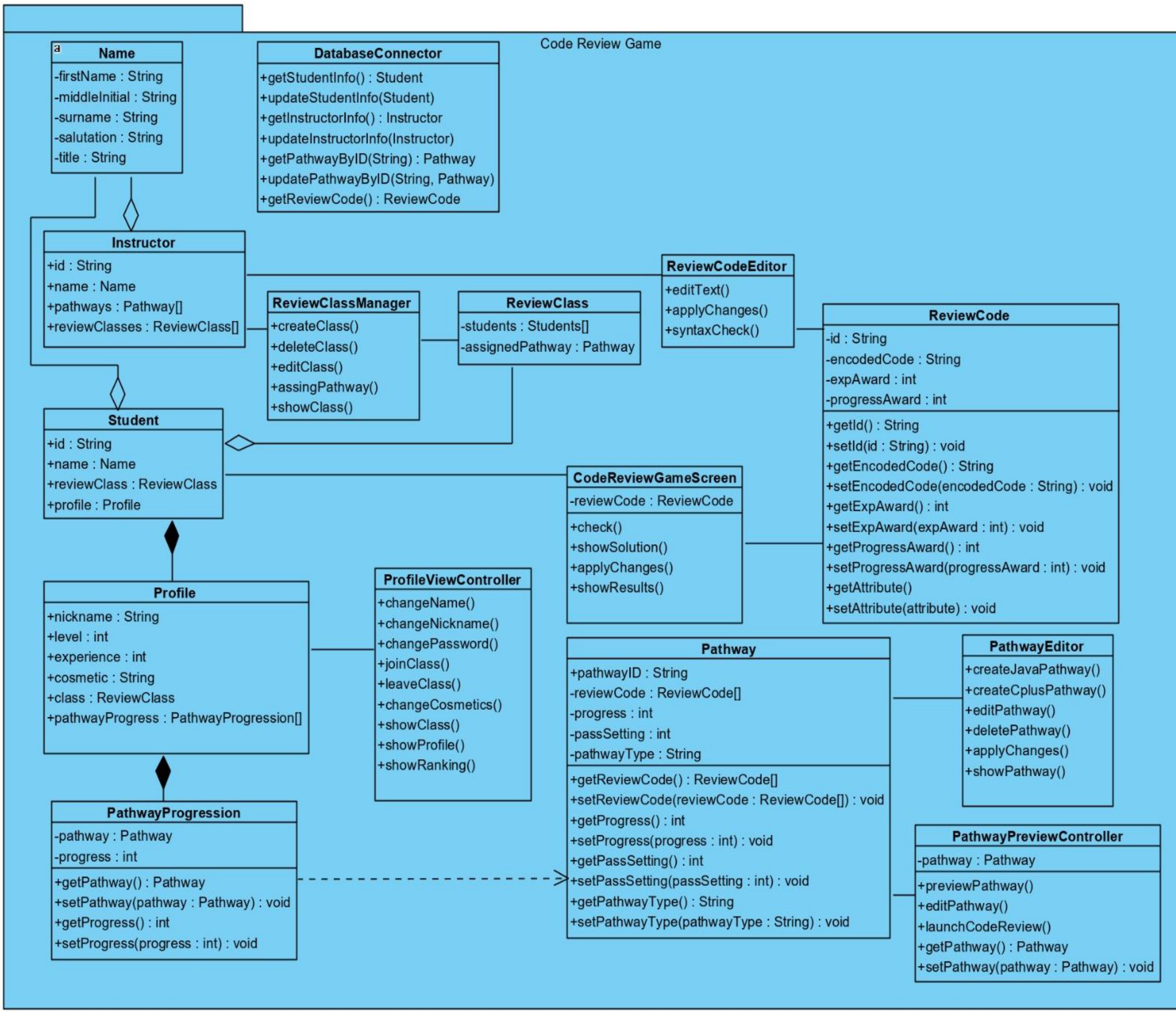


Figure 2: Class Diagram

2.5.4 Dynamic Models

2.5.4.1 Activity Diagram

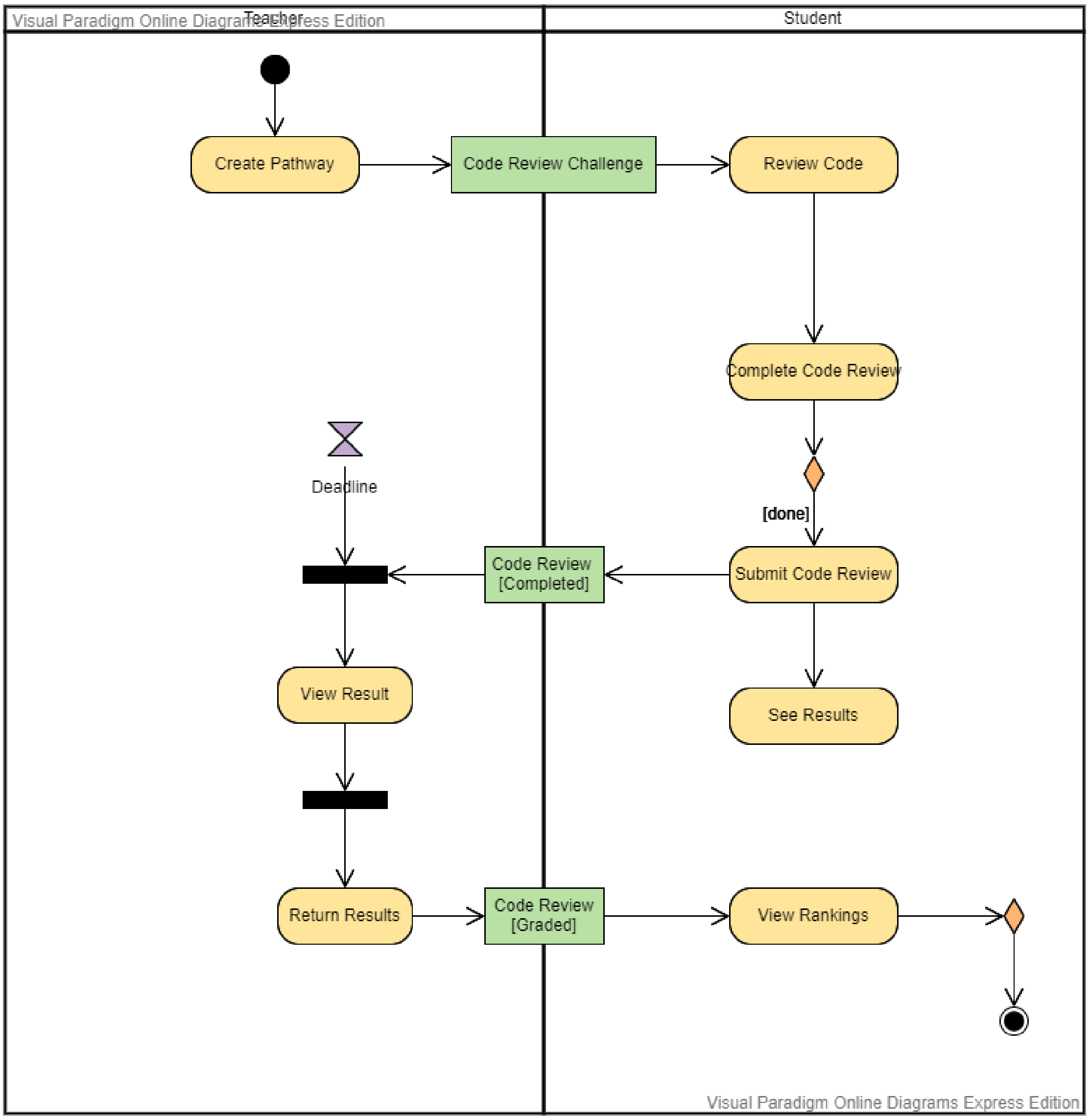


Figure 3: Activity Diagram

A regular activity in the application starts with the creation of a pathway by teacher. Teacher assigns pathway to student. Student solves the challenge and completes code review. Student submits his/her solution and sees the result. Teacher receives the solution and display the result. After collecting data from the whole class, teacher returns the results and student displays his/her rankings.

2.5.4.2 Sequence Diagrams

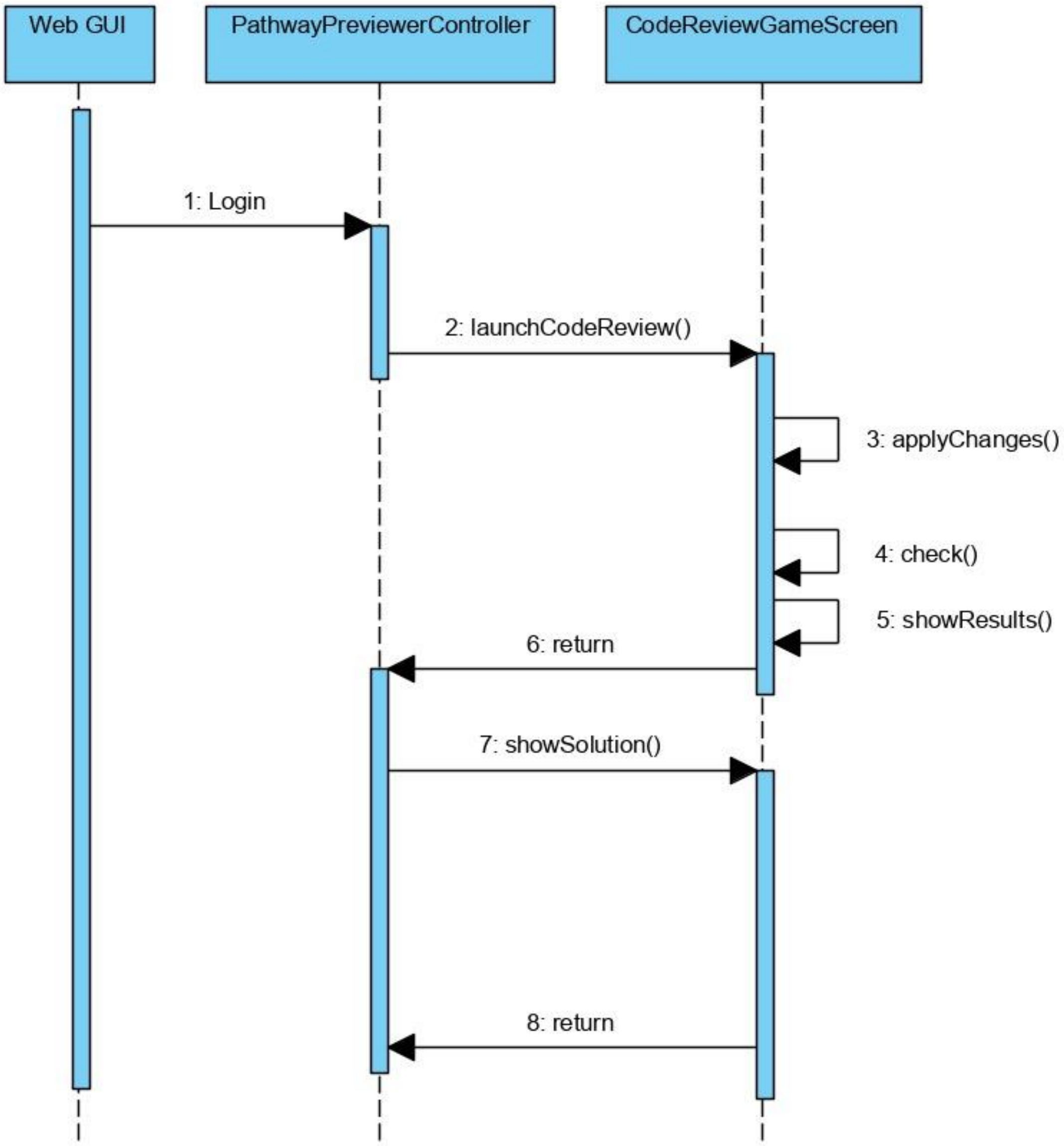


Figure 4: Sequence Diagram of Code Review Game Play

In the sequence diagram above (Figure 4), student logins. Check the assigned pathways for his/her class and selects the respected pathway. Code Review launches and appears in the main game screen. Student solves the challenge and submits it. The solution of student checked and result returned to student.

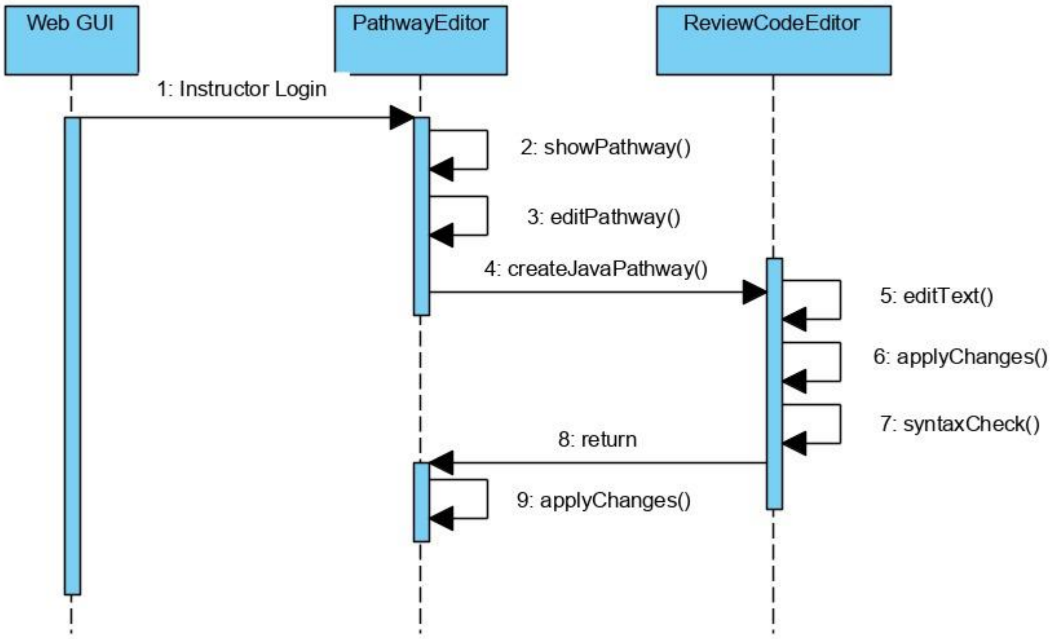


Figure 5: Sequence Diagram of Creating a Pathway by Instructor

In the sequence diagram above (Figure 5), instructor logins. He/she displays the pathways and create a new pathway. Then, edit the text manually or upload a code script. ReviewCodeEditor applies the changes, makes the syntax check and returns it to PathwayEditor. PathwayEditor applies changes in to pathway.

2.5.5 User Interface

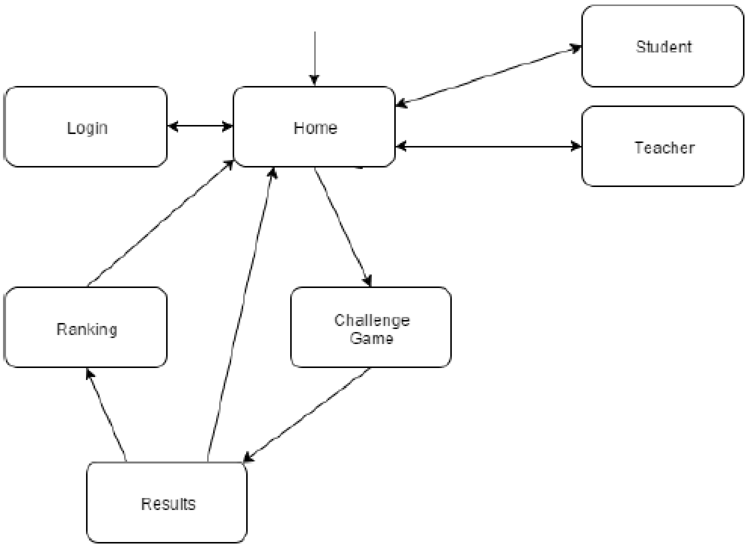


Figure 6: Navigational Path

2.5.5.1 Login View

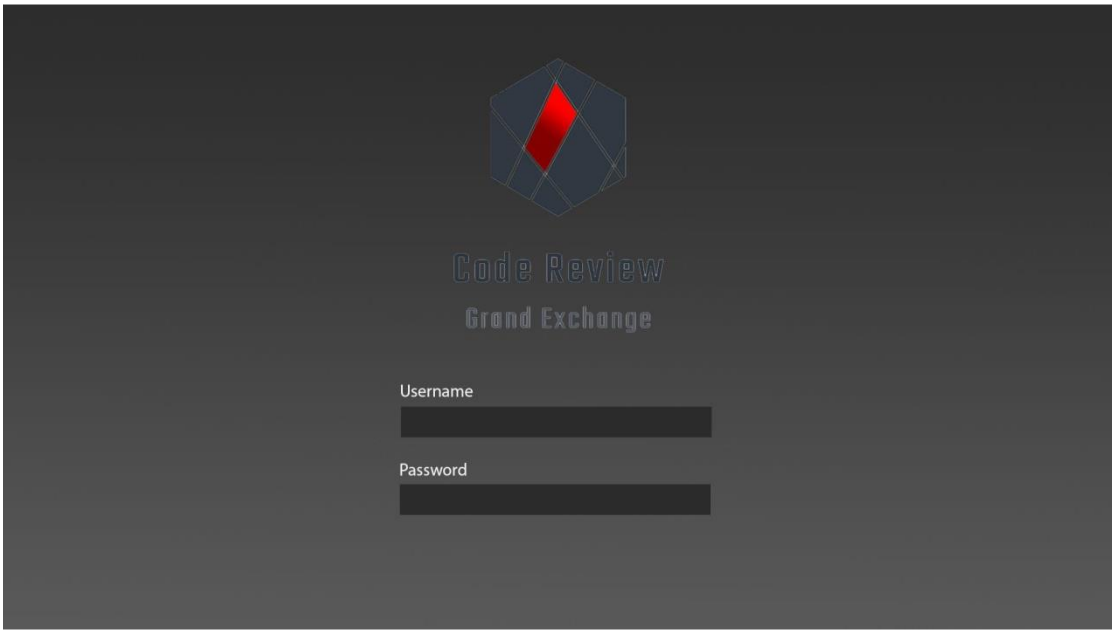
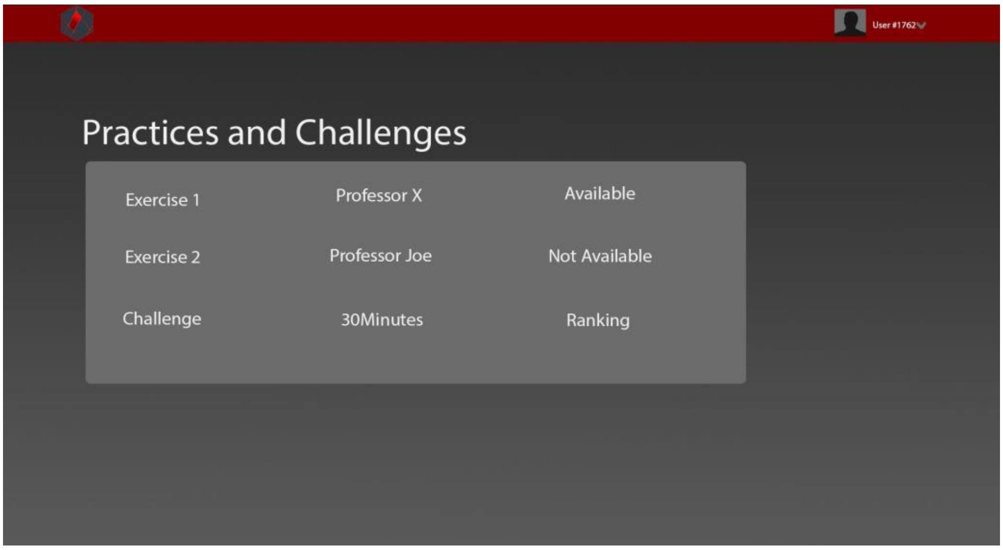


Figure 6: Login view

This view welcomes users without a logged in session, no matter what page they try to access. It consists of a form where users insert their usernames and passwords in order to gain access to the application if their input matches a user registered in the database. The login and logout feature utilizes the local storage of the browser to save the application's session information for the current user. In the case of the login, the credentials inserted are used as arguments in a query to the database to check if there's a register with those credentials. In the affirmative case, it returns the user's identifier, to be stored in local storage and dealt with as previously described.

2.5.5.2 Home View



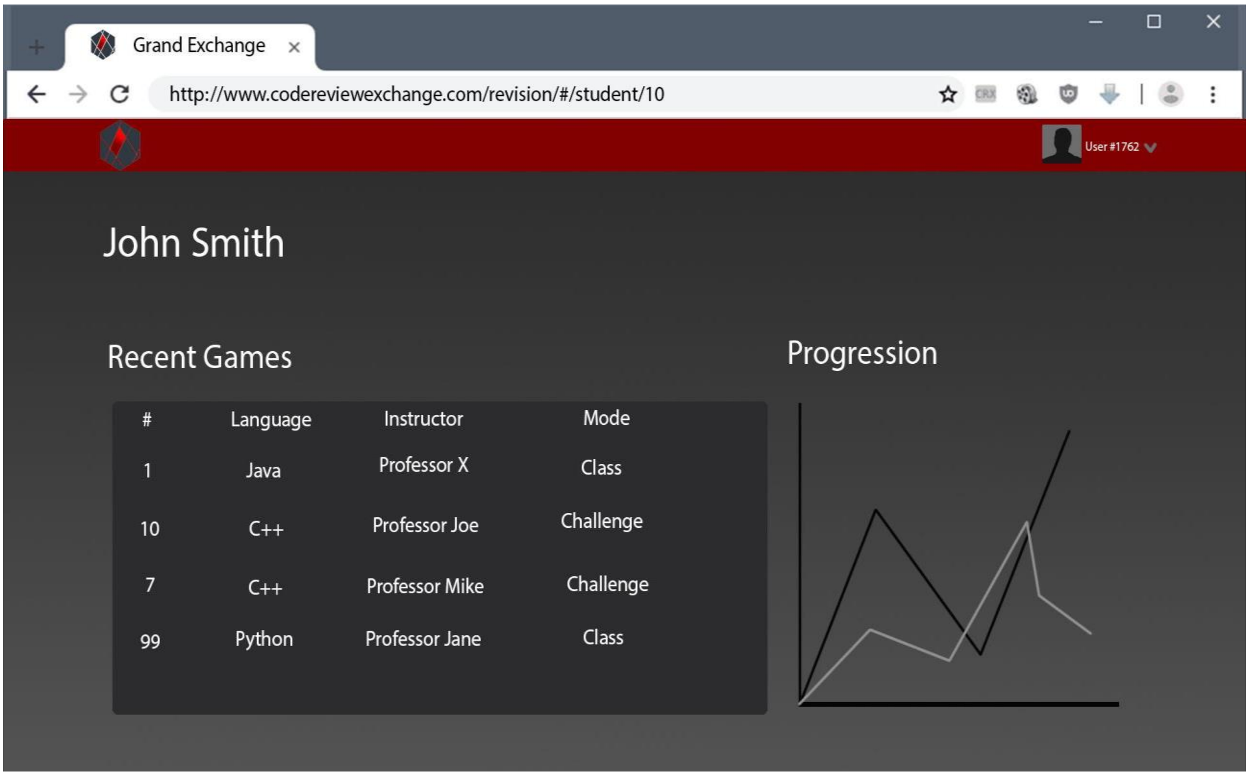
Practices and Challenges

Exercise 1	Professor X	Available
Exercise 2	Professor Joe	Not Available
Challenge	30Minutes	Ranking

Figure 7: Home view

This is the main view for logged users. It lists the available exercises and their information, as well as links for the player's page and to logout of the application. The exercise list has comprised a series of tables that succeed vertically, in which each table corresponds to an exercise. Each table will have 2 or 3 rows of information, depending on whether both game modes will be available for that exercise. Each game mode row lists the game mode it corresponds to, the time limit and a link to its respective ranking. The first row of the table describes the game number, the teacher that posted it and the availability of the exercise.

2.5.5.3 Student View



John Smith

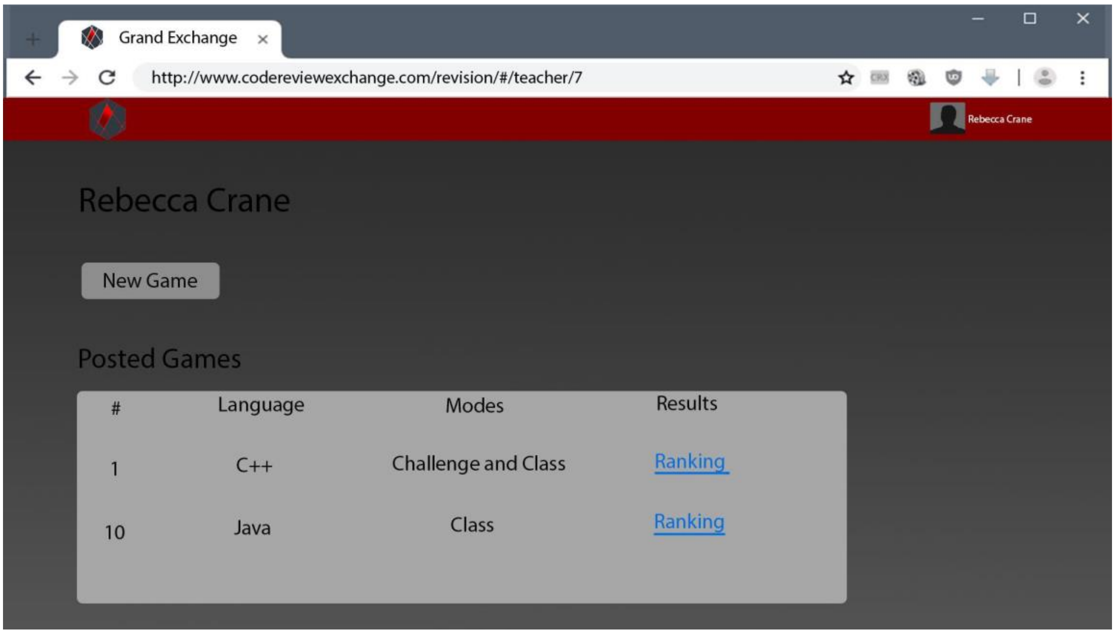
Recent Games

#	Language	Instructor	Mode
1	Java	Professor X	Class
10	C++	Professor Joe	Challenge
7	C++	Professor Mike	Challenge
99	Python	Professor Jane	Class

Figure 8: Student view

The information related to a specific student and the attempts that student made in any exercise are displayed here. It holds some visual information on the player's evolution as well as the badges won.

2.5.5.4 Teacher View



Rebecca Crane

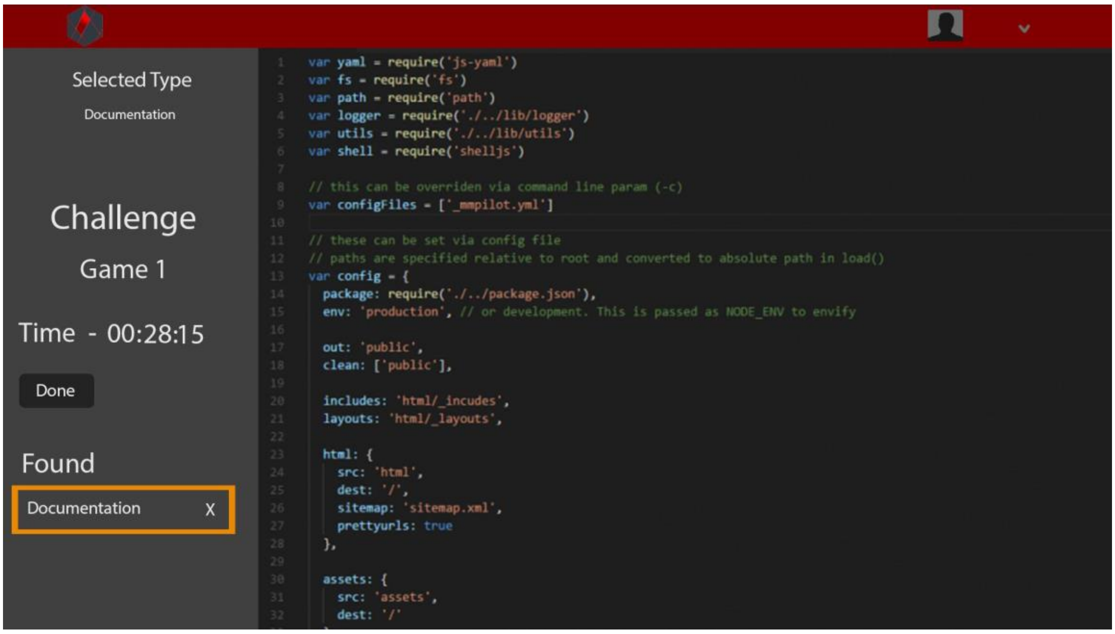
Posted Games

#	Language	Modes	Results
1	C++	Challenge and Class	Ranking
10	Java	Class	Ranking

Figure 9: Teacher view

This page lists the exercises posted by the teacher, as well as links to the respective games and rankings.

2.5.5.5 Game View

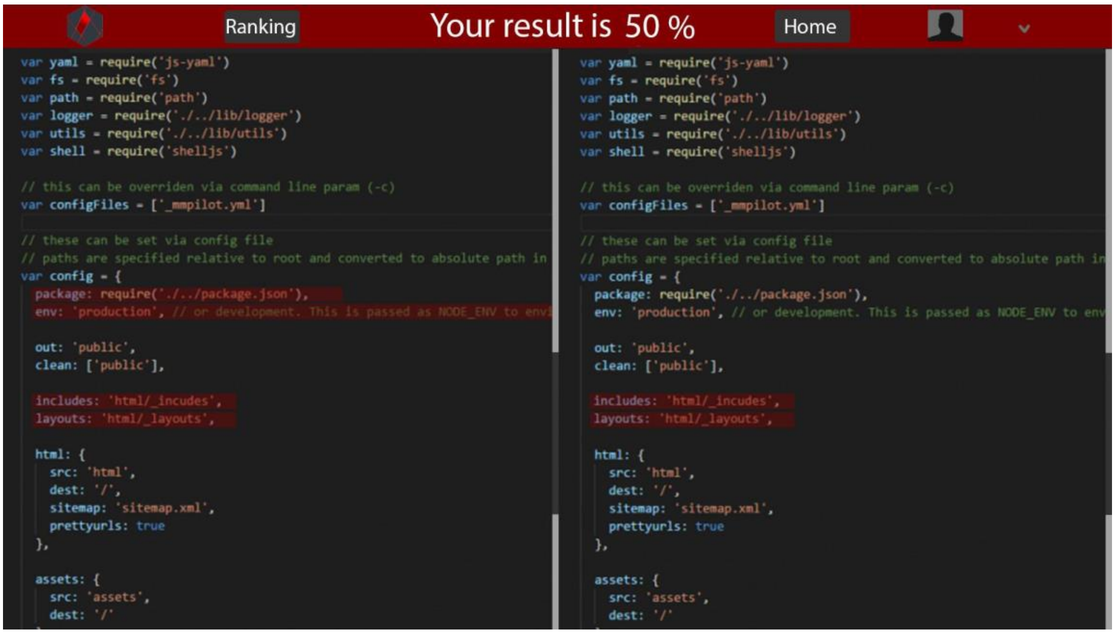


```
var yaml = require('js-yaml')
var fs = require('fs')
var path = require('path')
var logger = require('./../lib/logger')
var utils = require('./../lib/utils')
var shell = require('shelljs')

// this can be overriden via command line param (-c)
var configFiles = ['_mmpilot.yml']

// these can be set via config file
// paths are specified relative to root and converted to absolute path in load()
var config = {
  package: require('./../package.json'),
  env: 'production', // or development. This is passed as NODE_ENV to envify

  out: 'public',
  clean: ['public'],

  includes: 'html/_incudes',
  layouts: 'html/_layouts',

  html: {
    src: 'html',
    dest: '/',
    sitemap: 'sitemap.xml',
    prettyurls: true
  },

  assets: {
    src: 'assets',
    dest: '/'
```

Figure 10: Game view

In this view, the revision process of the inspection takes place. It’s a common screen for both game modes and it holds all the features related to marking defects.

2.5.5.6 Result View



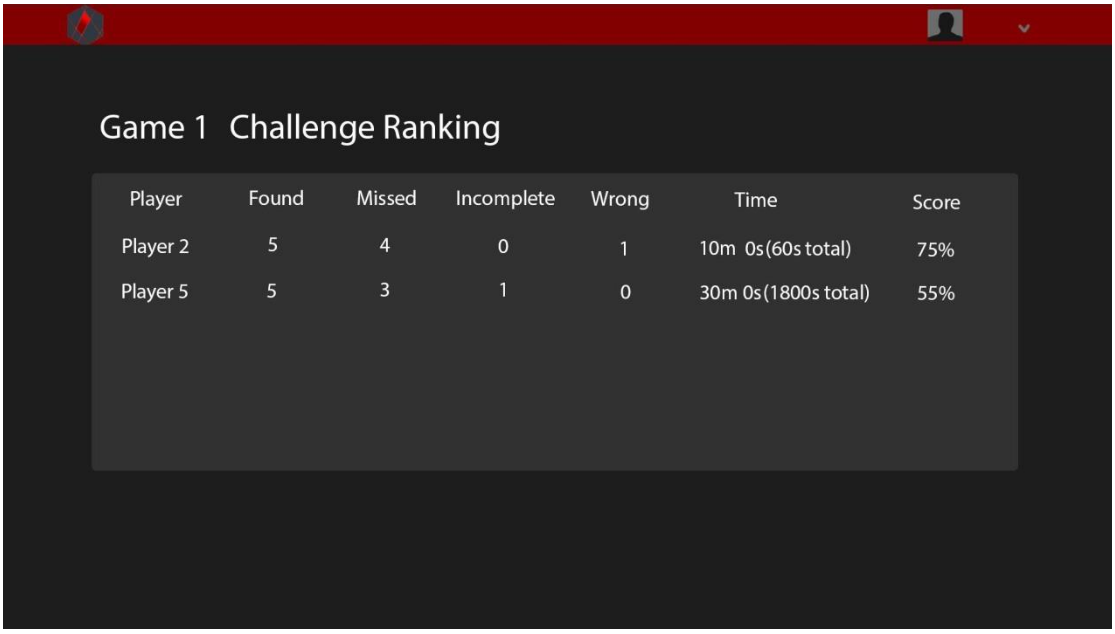
```
var yaml = require('js-yaml')
var fs = require('fs')
var path = require('path')
var logger = require('./../lib/logger')
var utils = require('./../lib/utils')
var shell = require('shelljs')

// this can be overriden via command line param (-c)
var configFiles = ['_mmpilot.yml']

// these can be set via config file
// paths are specified relative to root and converted to absolute path in
var config = {
  package: require('./../package.json'),
  env: 'production', // or development. This is passed as NODE_ENV to env

  out: 'public',
  clean: ['public'],

  includes: 'html/_incudes',
  layouts: 'html/_layouts',

  html: {
    src: 'html',
    dest: '/',
    sitemap: 'sitemap.xml',
    prettyurls: true
  },

  assets: {
    src: 'assets',
    dest: '/'
```

```
var yaml = require('js-yaml')
var fs = require('fs')
var path = require('path')
var logger = require('./../lib/logger')
var utils = require('./../lib/utils')
var shell = require('shelljs')

// this can be overriden via command line param (-c)
var configFiles = ['_mmpilot.yml']

// these can be set via config file
// paths are specified relative to root and converted to absolute path in
var config = {
  package: require('./../package.json'),
  env: 'production', // or development. This is passed as NODE_ENV to env

  out: 'public',
  clean: ['public'],

  includes: 'html/_incudes',
  layouts: 'html/_layouts',

  html: {
    src: 'html',
    dest: '/',
    sitemap: 'sitemap.xml',
    prettyurls: true
  },

  assets: {
    src: 'assets',
    dest: '/'
```

Figure 11: Result view

This is another view that it consists in the screen that shows the rating of the solution attempt and a visual comparison to the solution of the respective exercise.

2.5.5.7 Challenge Ranking View



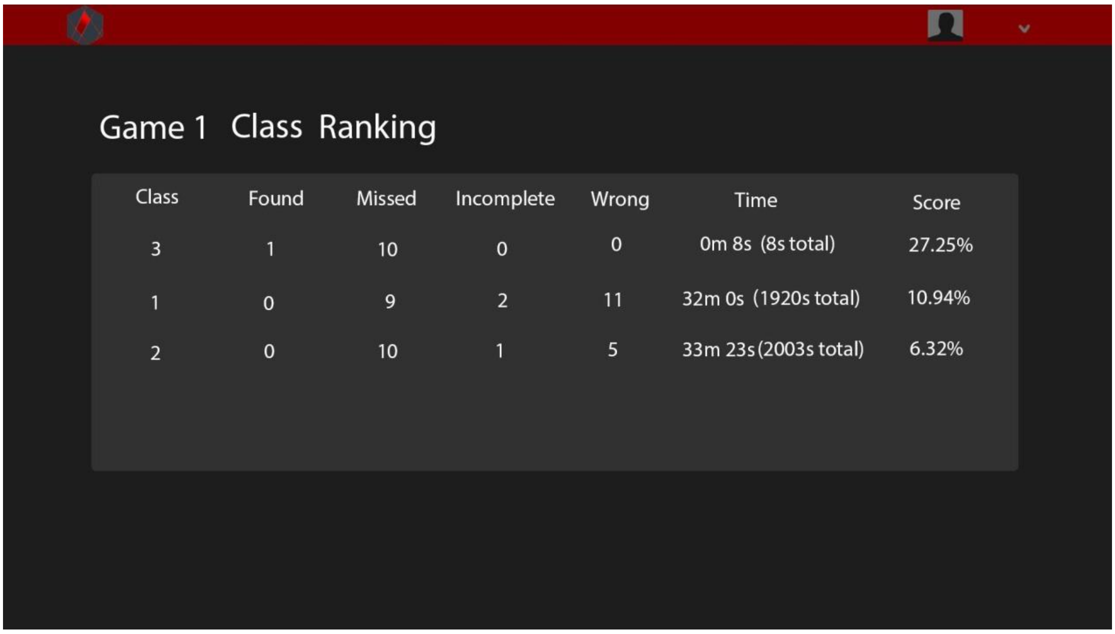
Game 1 Challenge Ranking

Player	Found	Missed	Incomplete	Wrong	Time	Score
Player 2	5	4	0	1	10m 0s(60s total)	75%
Player 5	5	3	1	0	30m 0s(1800s total)	55%

Figure 12: Challenge ranking view

The view that lists the completed attempts for a given exercise, in decreasing order of rating. Along with the rating, it also lists the main metrics of the respective attempts, which consist of the time spent and the defects found, partially found, missed and the number of wrong guesses.

2.5.5.8 Class Ranking View



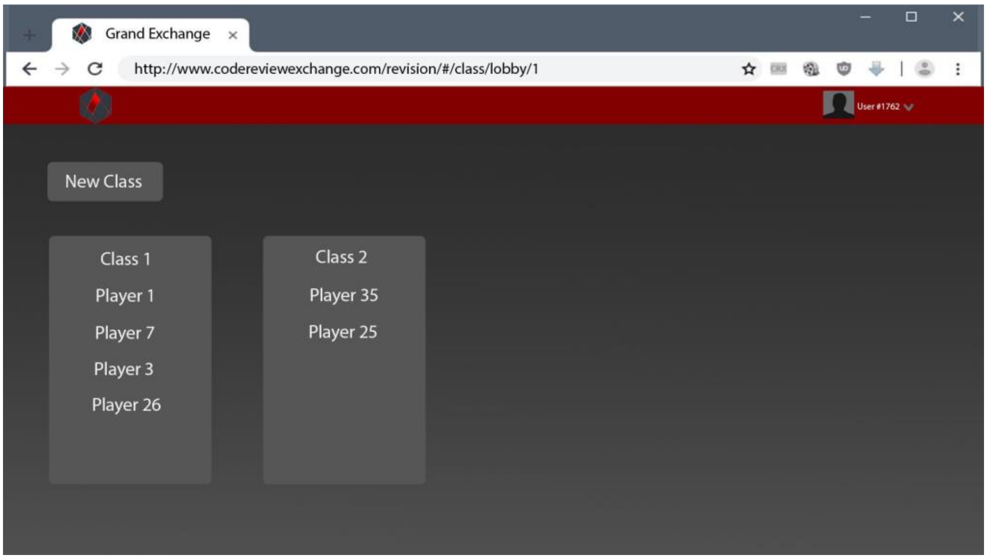
Game 1 Class Ranking

Class	Found	Missed	Incomplete	Wrong	Time	Score
3	1	10	0	0	0m 8s (8s total)	27.25%
1	0	9	2	11	32m 0s (1920s total)	10.94%
2	0	10	1	5	33m 23s (2003s total)	6.32%

Figure 13: Class ranking view

This view is similar to the Challenge ranking view, with the exception that instead of the solution attempt being connected to a player, it's connected to a class identifier, which refers to the players in that class.

2.5.5.10 Class Lobby View



http://www.codereviewexchange.com/revision/#/class/lobby/1

Class 1
Player 1
Player 7
Player 3
Player 26

Class 2
Player 35
Player 25

Figure 14: Class lobby view

This is the first view players interact with after clicking to start game.

3 Other Analysis Elements

3.1 Consideration of Various Factors

The impact of software products increases on a daily routine. Day by day people are seeing technology more intensely in their daily life thorough software products. As a result of this, some external factors like public health, public welfare or social factors start to affect the analysis phase of software products. In this case, the responsibility of the software designers and developers increase according to the share of technology in people’s life.

The Game – Code Review is a product which aims to adopt code review habit for the students who learn software development. The target audience of the product is very limited so that it does not have a comprehensive effect for public health.

Everyday billions of people enter personal data to software applications and the developers of the products are responsible for protecting the private data from third parties. In our case, we are designing the code review game by considering the importance of privacy. E-Mail addresses, passwords or any other personal data will not be shared with any different application. On the other hand, code review is an important phase for revealing bugs in the written code before a release and some of the bugs can cause serious security problems for the software products [2]. By the impact of our product, code review will be adopted by the freshman developers and this will increase the rate of finding security bugs in a program before a release. In one sense, secure software product means secure public.

The Game will give professional culture to freshman developers in terms of code review aspects. Usually, code review is not seen as a regular phase in the development process and it is not a settled culture especially for the less experienced developers [1]. To provide a code review as a regular phase of development will change the professional culture of these developers.

The application gives chance for users to compete as a whole class. Reviewing a code script with classmates in a competitive way would increase the sense of togetherness with adopting a code review habits. The ability to work in a team is a very important earning for a developer because most of the software projects are developed by teamwork. So, our design will have positive effects on social factors in terms of interoperability and compete with respect. On the other hand, our game carries educational purposes. Education is totally related to social factors and our analysis is done by considering the educational purposes. We aimed to teach code review to freshman developers with all aspects in an enjoyable way by keeping the balance between education and entertainment.

The Game – Code Review has a very specific target audience and this situation isolates our solutions from the global, environmental and economic factors. Already, The Game is for the usage of freshman students and general concepts like global, environmental and economic factors are not deeply related to our target audience and product design.

Table 1: Factors that can affect analysis and design.

	Effect level	Effect
Public health	0	Not a concrete affect
Public safety	5	Code review will reduce bugs and more secure products will be released with fewer bugs. It is important for data privacy.
Public welfare	0	Not a concrete affect
Global factors	0	Not a concrete affect because of the target audience
Cultural factors	4	Provide professional culture in software developments manners. Code review will be adopted as a regular phase of a development process.
Social factors	6	Improving teamwork and educate freshmen for the code review

3.2 Risks and Alternatives

Every software development project has some risks that can affect several factors like schedule, labor force, budget, etc... To minimize the possible negative effects of risks, a risk management plan should be prepared. The risk management plan should analyze the possible risks by scaling its likelihood; discover effects on the projects and propose alternative solutions.

A common risk for all projects is staff turnover. We also have a risk of losing one or more contributors because of personal reasons. Some of us can decide to leave the course. This sort of situation would directly affect our scheduling because of the loss of workforce. Our alternative solution for this risk is setting a meeting immediately with the continuing group members and reconsidering the prioritization of the features. Also increasing the working hours on the project would be plus for schedule management.

The game will be implemented with Spring Hibernate and this will be a new technology for contributors. So some misunderstandings about Spring Hibernate can cause design issues and fixing the problems would cause a delay in the schedule. The B plan for this sort of situation is implementing a different type of database like MongoDB which is used before by some of the contributors.

Another possible risk is the wrong time scheduling for the project. Time scheduling is done in the analysis phase however some workforce could be estimated wrong. This risk can cause a delay in the schedule. The alternative solution for wrong time scheduling is reconsidering the prioritizations and rescheduling. Also increasing the working hours on the project would help us to meet up the deadlines.

We can have some sort of productivity or qualification problems while developing the product. Some reasons for this possible risk can be mistakes in the distribution of work and disability of ensuring proper team-work. If we feel that something goes wrong in terms of productivity and qualification, we will immediately get in contact with each other and share the work again according to our knowledge and capabilities.

Wrong cost estimation is the final possible risk of the project. According to our current analysis, we are not planning to spend money and we do not have allocated budget. However, there can be a need for license of software in our development process that we can’t estimate now. As a solution to this problem, we can pay the price together with the team members.

The summarized information about the risks and decided alternative solutions for the risks are shown in Table 2.

Table 2: Risks

	Likelihood	**Effect on the project**	**B Plan Summary**
Stuff Turnover	Low	Delay on schedule, increase in individual work	Reconsider prioritization, more working hours individually
Misunderstandings in Database	Low	Schedule delay, quality problems	Implement different database like MongoDB
Wrong time scheduling	High	Schedule delay	Reconsider prioritization, more working hours individually
Productivity or qualification problems	Medium	Dysfunctional product	Reconsider distribution of work and change the responsibilities of contributors

Wrong cost estimation	Low	Not a concrete effect	Pay any cost together with teammates

3.3 Project Plan

The objective of the project is to develop an application that is turning code inspection into more effective and amusing process. This application will be used by instructors and students and students will be able to solve exercises posted by teachers through the application at the end.

Important milestones of the project:

- Project Specifications
- Analysis Report
- High-Level Design Report
- Low-Level Design Report
- Final Report
- Presentations & Demonstrations

Communication for Project Plan:

Although we haven't determined when we will do formal communications during the project development, it is obvious that we will take advices from our supervisor Halil Altay Güvenir and from our innovation expert Ahmet Eren Başak about the project stages for checking whether we are on the right track and whether we can complete the project within the given time interval or not and whether our product will be at the desired level or not.

<table>
<tr><td colspan="4"><b>Design User Stories</b></td></tr>
<tr><td colspan="4"><b>Start date:</b> <i>October 8, 2019</i> <b>End date:</b> October 14, 2019</td></tr>
<tr><td><b>Leader :</b></td><td><i>&lt;write the name of the work package leader&gt;</i></td><td><b>Members involved:</b></td><td><i>Burak Erkılıç 21501035</i>
<i>Mert Sezer 21400246</i>
<i>Okan Şen 21202377</i>
<i>Berk Yıldız 21502040</i></td></tr>
<tr><td colspan="4"><b>Objectives:</b> <i>User stories are divided in 3 types according to the actor: teacher, student and available to both</i></td></tr>
<tr><td colspan="4"><b>Tasks:</b>
<i>- Define Login whose actors are Teacher and Student</i>
<i>- Define solving code review whose actor is student</i>
<i>…</i></td></tr>
<tr><td colspan="4"><b>Deliverables</b>
<b><i>Internal Deliverable is the strategic plan that is considering the customer needs and is adopted by team members and was delivered to our innovation expert who gave the team members feedbacks about whether we are doing the project right or not, whether we are doing it in an innovative way or not.</i></b>
<b><i>External Deliverable is the Analysis Report and the use cases are mentioned in the analysis report after designing user stories.</i></b></td></tr>
</table>

<table>
<tr><td colspan="4"><b>Design Reviews</b></td></tr>
<tr><td colspan="4"><b>Start date:</b> <i>November 12, 2019</i> <b>End date:</b> <i>November 30, 2019</i></td></tr>
<tr><td><b>Leader :</b></td><td><i>&lt;write the name of the work package leader&gt;</i></td><td><b>Members involved:</b></td><td><i>Burak Erkılıç 21501035</i>
<i>Mert Sezer 21400246</i>
<i>Okan Şen 21202377</i>
<i>Berk Yıldız 21502040</i></td></tr>
<tr><td colspan="4"><b>Objectives:</b> <i>Objective of designing reviews are determining the identification of specific errors and defects in the software artifact and identification of problematic components in the software that needs improvement for the overall quality and conformance to organizational standards. Because our application will be code review game, we will first work on designing reviews.</i></td></tr>
<tr><td colspan="4"><b>Tasks:</b>
<b><i>- Determine the type of review: formal, informal, technical, managerial</i></b>
<b><i>- Define size of items being reviewed</i></b>
<b><i>- Define number of defects found</i></b>
<b><i>- Define number of code or pages of document that is reviewed per hour, or Review Rate</i></b>
<b><i>- Define number of defects found per hour of reviews time</i></b>
<b><i>- Define number of defects found per page or per line of code</i></b>
<b><i>- Define preconditions for the review</i></b>
<b><i>- Define review steps</i></b>
<b><i>- Define time requirements</i></b><i>…</i></td></tr>
<tr><td colspan="4"><b>Deliverables</b>
<i>Internal Deliverable is the working software to the team members and team members will determine what improvements will be done and after that we will start working on what will be the strategic plan for making these reviews a game.</i></td></tr>
</table>

Gamification Design Framework			
Start date: December 1, 2019 **End date:** *December 20, 2019*			
Leader :	*<write the name of the work package leader>*	**Members involved:**	*Burak Erkılıç* *21501035* *Mert Sezer* *21400246* *Okan Şen* *21202377* *Berk Yıldız* *21502040*
Objectives: the objective is to achieve developing code review as a game concept in later phases			
Tasks: ***-Define business objectives: justifying objective of the game*** ***-Describe players*** ***-Delineate target behaviours: this will allow players to receive feedback on their attempts and this keeps players engaged in the intended behaviour*** ***-Make it fun:*** ***-Design activity loops: The plans to motivate players using progression loops*** ***Cheating is a possible risk behind gamification*** ***-Rewards: players should take reward by completion of a task****…*			
Deliverables *Internal deliverable is the strategic plan to the team members and team members will apply this strategic plan throughout the development of code review game.*			

Define Solution Concepts			
Start date: *December 21, 2019* **End date:** *December 30, 2019*			
Leader :	*<write the name of the work package leader>*	**Members involved:**	*Burak Erkılıç* *21501035* *Mert Sezer* *21400246* *Okan Şen* *21202377* *Berk Yıldız* *21502040*
Objectives: *Note that the goal is to mark the defects and to do that the player has to follow some steps and rating system will be used for determining the result of the game play.*			
Tasks: ***- Pinpoint defects: The player has to be sure that there is some code selected in the code area, that there is a selected type and one of the two severity options is also selected. Once all those conditions are met, the player can click the "Pin" button to mark the excerpt as a defect with the intended severity and type.*** ***- Define Rating Calculation: Each item is compared to each guess through a Jaccard index formula that measures the similarity between strings. If the solution defect found no match, it's counted as miss and that means the player failed to find the defect.***			
Deliverables *External Deliverable is the High Level Design Report and in this report, new knowledge acquired and learning strategies used for achieving the solution concepts are mentioned.* *Internal Deliverable is the strategy that all team members must adopt during the development of code review game.*			

WP 5: Achieve Application Adaptation			
Start date: *January 3, 2020* **End date:** *February 15, 2020*			
Leader :	*<write the name of the work package leader>*	**Members involved:**	*Burak Erkılıç 21501035 Mert Sezer 21400246 Okan Şen 21202377 Berk Yıldız 21502040*
Objectives: *To find whether the application is a valid adaptation of the code inspection process and whether it can hold its' own as a serious game*			
Tasks: ***- Achieve modularity(The main program where the user interacts with the application and An API for database communication3, as well as a MySQL database, hosted at OpenShift by Red Hat4, an online platform that allows the NodeJS and database hosting necessary for this experiment)****…*			
Deliverables *- External Deliverable is the Low-Level Design Report and in it, packages, class interfaces, engineering standards that will adopted until working on the final report will be determined.* *- Internal Deliverable is the working software that convinces all team members that the design of product is right.*			

WP 6: Achieve Implementation			
Start date: *February 18, 2020* **End date:** *May 5, 2020*			
Leader :	*<write the name of the work package leader>*	**Members involved:**	*Burak Erkılıç 21501035 Mert Sezer 21400246 Okan Şen 21202377 Berk Yıldız 21502040*
Objectives: *achieving a high quality, working software that is ready to be used by stakeholders.*			
Tasks: ***- Achieve Architectural and Technical Specifications:*** ***Two proposed solution has a client device and server node that has user interface and database interface. The first is a simple web-site hosting server while the second can be any device with a web browser compatible with the presented technologies.*** ***- Database Design*** ***- Website Design****…*			
Deliverables *- External Deliverable is Project Demo.* *- Internal Deliverable is the working software and all team members can start working on the Final Report after implementation will be achieved.*			

3.4 Ensuring Proper Team-Work

The most important thing for achieving team work is that everybody in the team understands the team goals and is committed to attaining the mission and the purpose of the team; in other words, all team members share a clear understanding of the stages of project development and the required things to move the team successfully through the stages for CS491/CS492 courses. To achieve this, time estimations and task lists are created for each phase of the project and future improvements task lists are created after some progress is done by any team member. To ensure proper team work, we are setting clear goals that are separated parts of the overall goal of the specific phase and responsibilities to every individual and they should be adopted by every individual in the team. It is crucial that every team member must respect the division of labor and commit to their workshare during the project phases. These separate goals that achieve division of labor are for contributing to the overall team performance positively and for ensuring that each individual in the team remains dedicated to the overall effort. Also we are arranging team meetings for addressing overall team issues. Open discussions related to team issues helps the team members to take positive and helpful recommendations and at the end of the discussions and providing a concrete plan for correcting these issues. Although team members should have a strong sense of belonging to group, disagreements are expected and encouraged. We know that each team member can have a different background, different experiences, different skills and therefore there is diverseness in the team that brings improvement of different processes, skills to solve different problems and creation of new and exciting things. In other words, we do realize that individual differences regarding to the efforts, talents and strategies should move the group forward and help the group to achieve progress. For this reason there must be communication between team members that needs to be open, honest and respectful. Lack of communication that is the result of fear of conflict may prevent may hinder sharing of knowledge and experience for realizing common tasks. Therefore it is important that each team members should be comfortable in taking reasonable risks, for instance one team member can start doing the use case diagram before the meeting that aims to discuss the goals of the analysis report. Through that, the team's workload can decrease or the team can have a preconception about what should be done in the following phase to deliver it at the desired level and before the given deadline. As a result of our findings we are going to use Github and Asala to keep track of our progress.

3.5 Ethics and Professional Responsibilities

The ethical and professional responsibilities of software engineers increase as the share of technology in daily life increases. This situation brings the necessity of making analysis in ethical manners with professional approaches. To meet these requirements, data privacy, information security, and copyright issues are prioritized in our product.

There is a risk of a teacher to upload a copyrighted piece of code to the program and this case will cause an ethical issue. So before any sort of upload to the program, the uploader should agree on a text that he/she takes the responsibility of the uploaded code.

The program will store personal data of users and it will contain the game results of contributors which will assign their rankings. As the developers of the program, we won't share any sort of personal data with third parties and we will not allow sharing any sort of data sharing with third parties without the permission of the particular contributor. Also, we won't use any sort of data for our own purposes.

At the beginning of the program we will clearly state all ethical rules to users for avoiding possible professional and ethical issues. In terms of profession, we will work for producing high standard software with qualified features in it.

The impact of our engineering will mostly be related to the societal context because our product carries educational purposes. In this manner, we are responsible for creating a useful and functional learning environment and our decisions will be taken regarding our purposes.

3.6 New Knowledge and Learning Strategies

For this project, each group member is aiming to improve himself in technological manner and documentation. Background of each other is totally different but we are willing to contribute equally in each phase of the project. To achieve this, we chose trend technologies which can be useful for us in future careers. We will implement the frontend in AngularJS which is a popular JavaScript framework. AngularJS is totally new for all of us and we are going to learn AngularJS together. We will implement the backend by Spring Java. Already we all know Java but Spring framework will be new to some of us. Those members will cover Springboot by themselves and ask for help to other members when needed.

We will mostly use internet sources to learn all of these technologies. Already official documentations are published for each of them and there are dozens of tutorials we can follow. The other important option for learning is asking each other. We will state our understandings and missing regularly to each other to check progress for learning. We will directly help each other when needed.

4 References

[1] Wind, D. K. (2019, May 3). Teaching code review to university students. Retrieved from https://towardsdatascience.com/teaching-code-review-in-university-courses-using-peer-feedback-5625fe039f2a.

[2] Thompson, C., & Wagner, D. (2017). A Large-Scale Study of Modern Code Review and Security in Open Source Projects. Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering - PROMISE. doi: 10.1145/3127005.3127014

** Used technologies changed in the last update.