

# **Progress Report**

## **Setwang**

### **Propose to**

Assist. Prof. Dr. Thanarat Chalidabhongse

Kasidit	Iamthong	5731005321
Kawin	Liaowongphuthorn	5731004721
Kosate	Limpongsa	5731012721
Parinthorn	Saithong	5730329521
Sirinthra	Chantharaj	5730635521
Thanat	Jatuphattharachai	5730243121

This project is a part of a course 2110332 Systems Analysis and Design

Second Semester, Academic Year 2016

Computer Department, Faculty of Engineering, Chulalongkorn University

## Executive Summary

**Project Name** Setwang

**Problem** 1. Learning resources are scattered and not easily accessible.

**Statement** 2. Students do not know their skill level.

3. There are few comprehensive solutions.

**Goals** To provide a knowledge-sharing community where high school students can simply browse through various easy-to-access problems along with their solutions.

**Methodology** Phased Development and Agile

**System Scope**

1. Feed	7. Virtual Exam
2. Post	8. Search
3. Comment	9. Content Editor
4. Problem Set	10. Flag and Review
5. Voting	11. Login/Register
6. Bookmarking	12. User/Profile

**Duration** 12 January 2017 – 21 May 2017 (130 days)

**Budget** Total cost of 11,221,363.30 with 105% Return on Investment within 5 years and 2.70 years of Break-Event point.

**Benefits**

1. Provide student an access to an examination and exercise problem set easily.
2. Help student access to the comprehensive solution for every question in the system.
3. Make students realize what the level they are at.
4. Provide the competitive environment for students to practice via the Virtual Exam System.
5. Help create educational sharing community for students, teachers and tutors.

## Table of Contents

1. Overview .....	1
2. Progress.....	2
3. Problems & Solutions .....	12
4. Future Development Plan .....	13
5. Change of Definition of Terms and System Requirement .....	13
6. List of Stakeholders and Responsibilities .....	14
7. Method of System Analysis Modeling.....	14
8. Functional Modeling.....	15
9. Structural Modeling.....	21
10. Behavioral Modeling .....	27

## Appendix

A. Definition of Terms .....	36
B. System Requirement.....	37

## List of Figures

Figure 1-1: To-be system's user flow .....	1
Figure 1-2: Technical overview of the to-be system.....	2
Figure 2-1: Updated Gantt Chart (1).....	3
Figure 2-2: Updated Gantt Chart (2).....	4
Figure 2-3: Updated Gantt Chart (3).....	5
Figure 2-4: Problem page's wireframe .....	8
Figure 2-5: Visual style guide.....	9
Figure 2-6: High-fidelity user interface of a problem page.....	10
Figure 2-7: Navigation section .....	11
Figure 2-8: Problem box section .....	11
Figure 2-9: Side section .....	12
Figure 8-1: Use-case diagram.....	15
Figure 8-2: Activity diagram of "Solve a problem" use case .....	16
Figure 8-3: Activity diagram of "Solve a problem set" use case.....	18
Figure 8-4: Activity diagram of "Attend a virtual exam" use case.....	20
Figure 9-1: Class diagram.....	22
Figure 9-2: Object diagram .....	23
Figure 10-1: Sequence diagram of "Solve a problem" use case.....	27
Figure 10-2: Sequence diagram of "Solve a problem set" use case .....	28
Figure 10-3: Sequence diagram of "Attend a virtual exam" use case for virtual exam with 3 problems scenario .....	29
Figure 10-4: Communication diagram of "Solve a problem" use case .....	30
Figure 10-5: Communication diagram of "Solve a problem set" use case .....	30
Figure 10-6: Communication diagram for "Attend a virtual exam" use case for virtual exam with 3 problems scenario .....	31
Figure 10-7: Behavioral state machine for Exam class .....	32
Figure 10-8: Behavioral state machine for Post class .....	33

## List of Tables

Table 6-1: Analysis Model responsibilities.....	14
Table 6-2: Implementation responsibilities .....	14
Table 8-1: Use-case description of "Solve a problem" use case .....	17
Table 8-2: Use-case description of "Solve a problem set" use case.....	19
Table 8-3: Use-case description of "Attend a virtual exam" use case .....	21
Table 10-1: CRUDE matrix .....	35

# 1. Overview

## 1.1. Objective of Progress Report

The objective of this document is to provide a brief progress of Setwang system development. It contains the comparing of progress and the original plan, change of requirements, revised development plan with Gantt's Chart, List of Stakeholders and Responsibilities, Problems and Solutions, followed by system analysis modeling which includes functional models, structural models and behavioral models.

## 1.2. System Overview

We designed the to-be system as represented by Figure 1-1 to create a better environment for the students. There are two main parts of our system.

First, the problems and solutions. Our system is an online platform which aims to be an educational sharing community for the students, teachers, and tutors. Therefore, we provide the system which problem creators can create problems and solutions for problem solvers to comment and discuss about the problem. Furthermore, we also provide set of the problems called problem set so that problem solvers can easily access to the categorized problems.

The other part is virtual examination. This system will provide an exam environment for students so that they can easily participate and know the feedback as well as their level right away. By doing a virtual examination on the system, students can test their knowledge wherever they want. Moreover, there are also detailed statistics and analytics, so students can improve themselves upon this information.

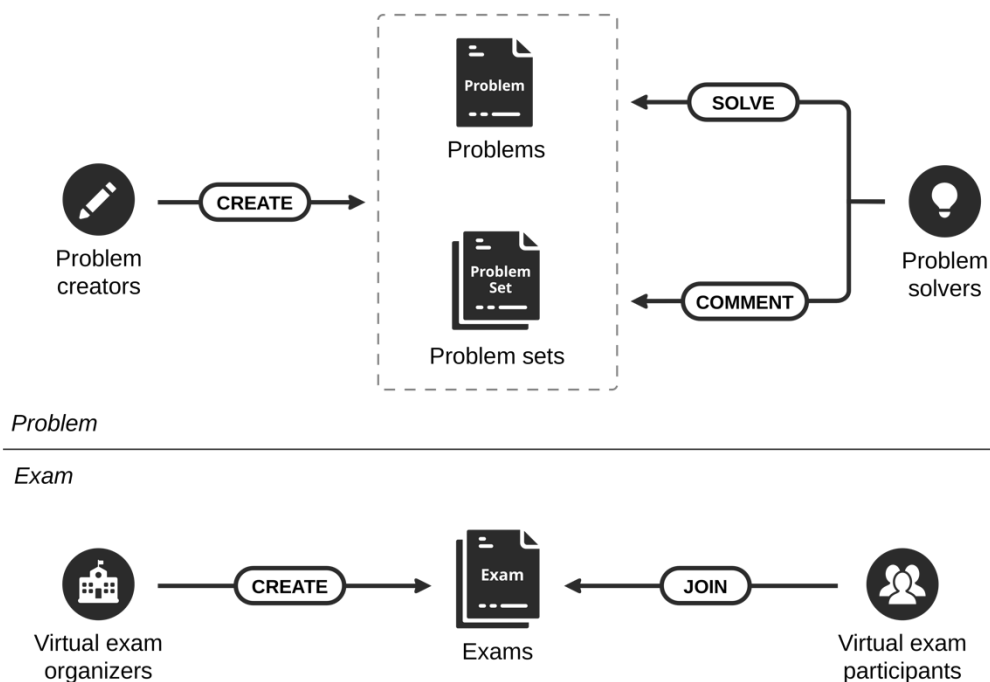


Figure 1-1: To-be system's user flow

When a user requests for a webpage, the client connects to one of available web servers. A web server might connect to any API servers to authenticate the user. It will then serve the user's client with the webpage. The webpage, which is created on Angular framework, will fetch the data from an API server via a REST protocol.

An API server is connected to a data server to do CRUD operations on a database. A data server, which is connected to a database storage, hosts a database engine and listens for a query from an API server. Dynamically uploaded images are hosted on a file storage, which can be requested or updated via any file servers.

Because the system is a user-centric website, all servers are connected and managed by an orchestration tool so that the system is scalable, as it should be able to handle the traffic load of very high number of concurrent users during a large virtual exam. The technical overview of the system can be roughly illustrated by Figure 1-2.

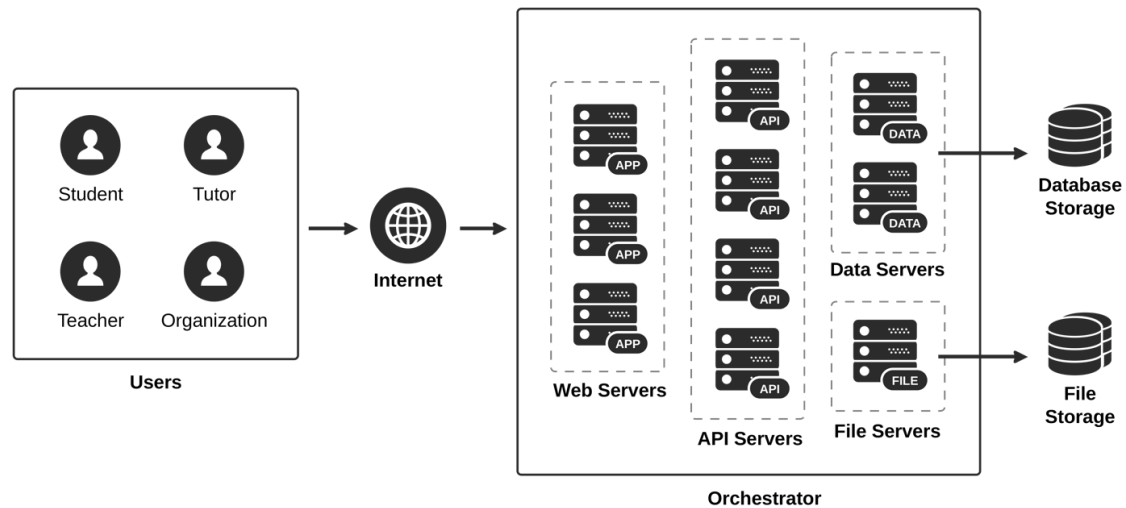


Figure 1-2: Technical overview of the to-be system

## 2. Progress

### 2.1. Change of Gantt's Chart

According to the problem which is described in the next section, our work is delayed and we have to adjust the plan by switching some work order in the project to make the work run smoothly as planned before. All the work in the Planning Phase is done perfectly according to the plan.

However, after we did analysis model during 18 January to 30 January, we found out that there were plenty of analysis models related to functionalities and the use case of the system needed to be done. Therefore, we had to modify the plan from the original plan by extending the analysis phase for another two weeks and reduce the time from the future analysis phase as shown in Gantt's Chart and described in the *Comparing Progress to the Plan* section.

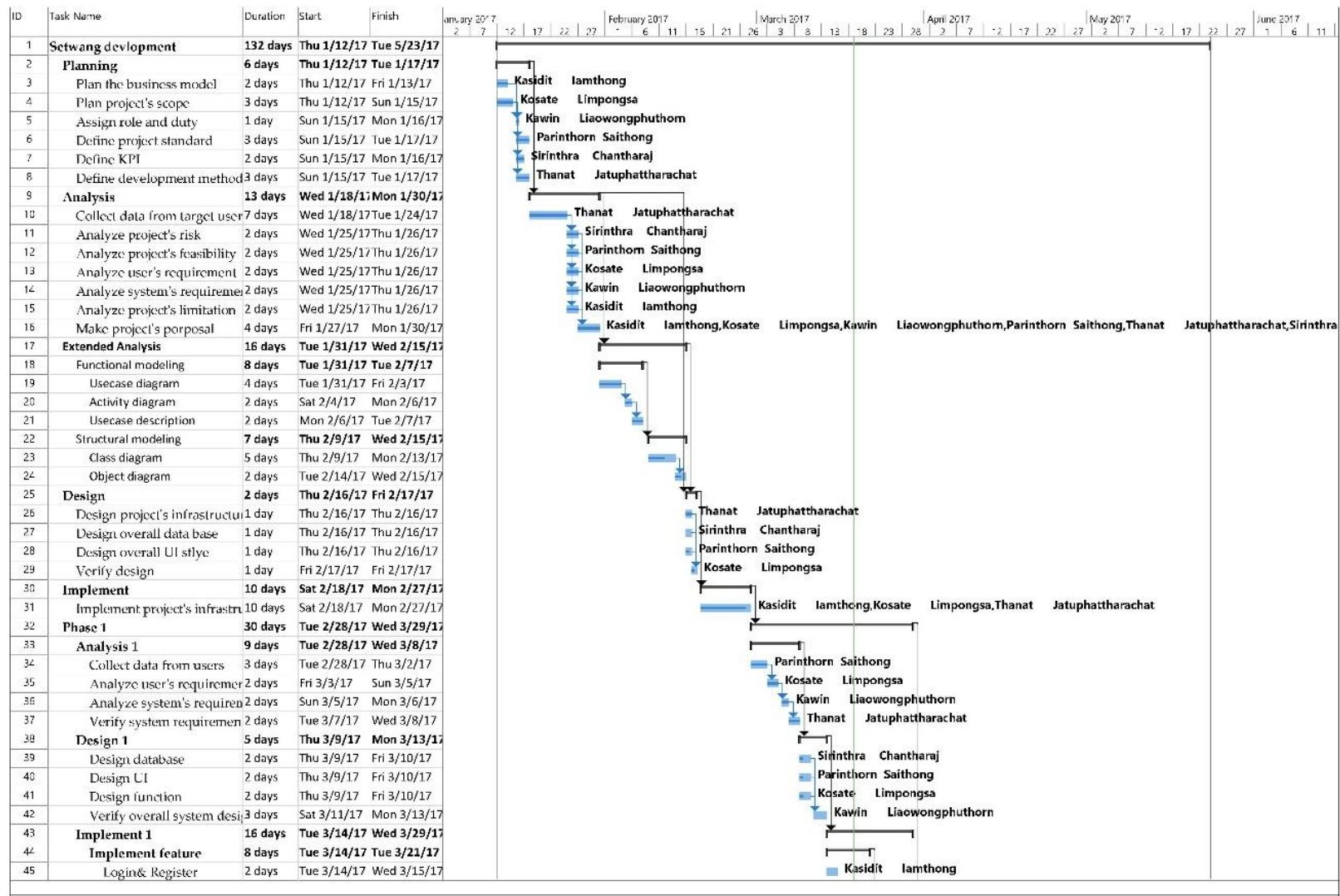


Figure 2-1: Updated Gantt Chart (1)

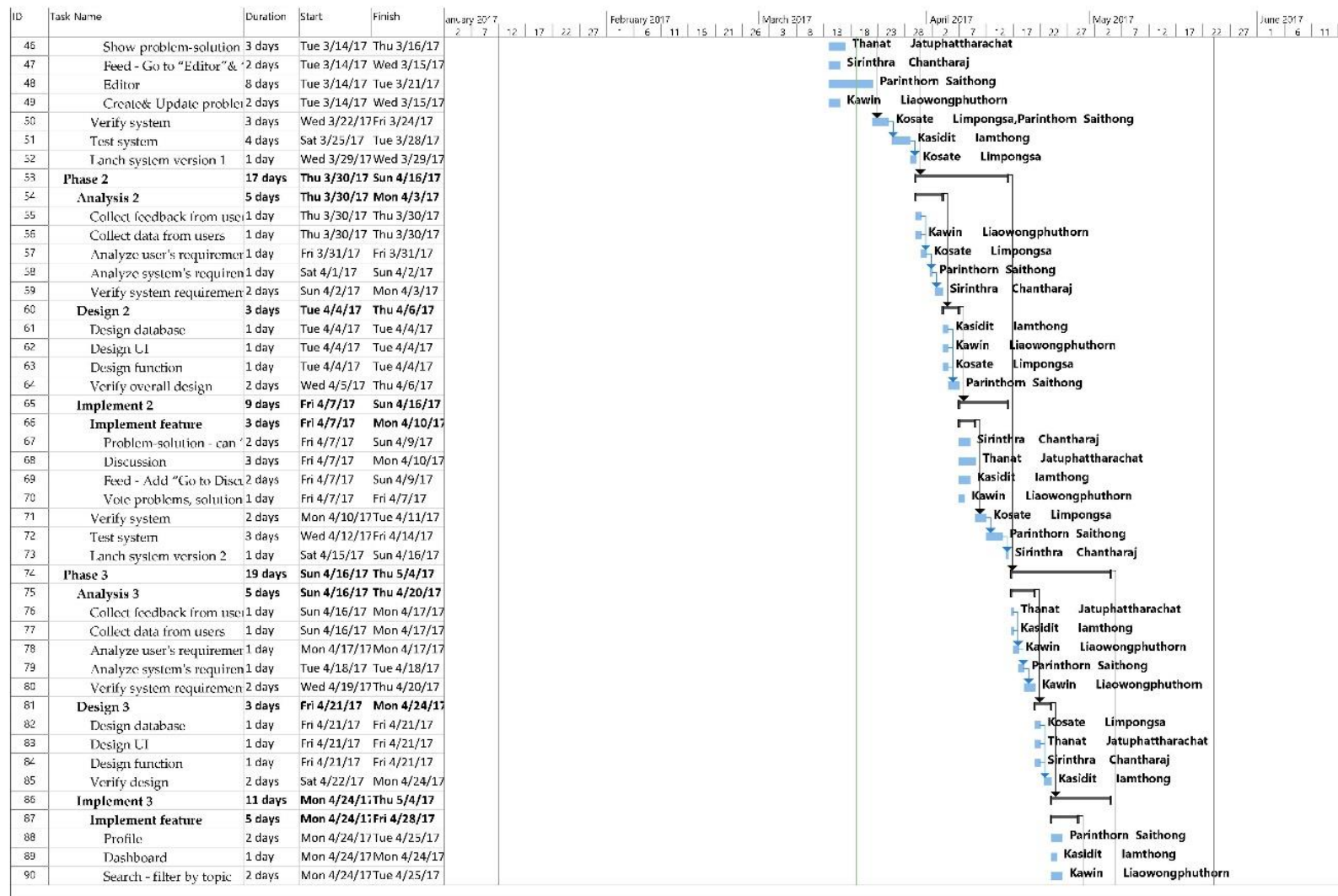


Figure 2-2: Updated Gantt Chart (2)



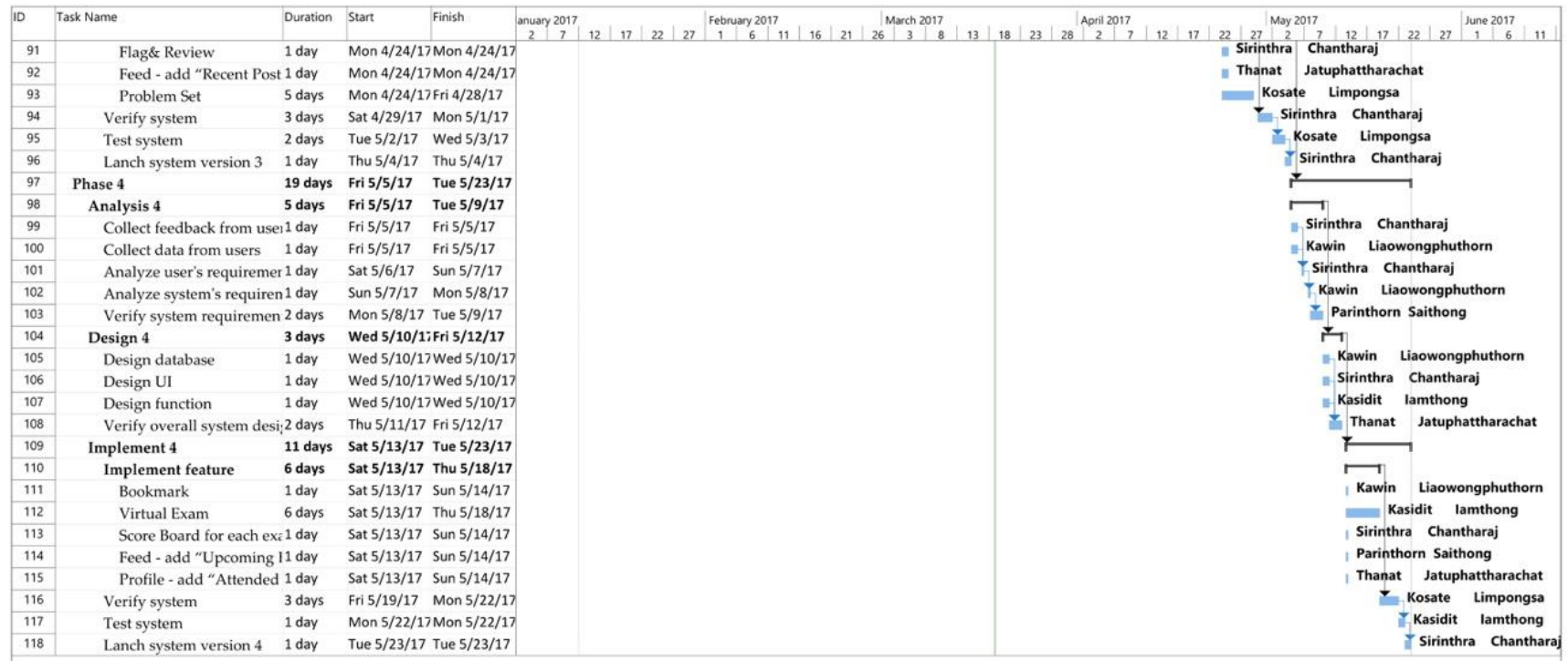


Figure 2-3: Updated Gantt Chart (3)

## 2.2. Comparing Progress to the Plan

### 2.2.1. Planning (12 Jan 2017 – 17 Jan 2017)

The development team worked on fundamental planning and analysis of the system. The team conducted the feasibility analysis for the system and thoroughly clarified business needs and the tangible as well as intangible value of Setwang system. We also defined the scope of the system along with listed the KPI to indicate the success of the system. We also selected an appropriate methodology for developing this system. Everything in this phase went along well with the plan. We finally came up with the details as written in the system request.

### 2.2.2. Analysis (18 Jan 2017 – 30 Jan 2017)

We carried out interviews to every stakeholder to gather data for further analysis of the system. After we carefully considered the collected data, we ran user requirements analysis session to define and categorize user requirements. Besides, the system requirements were listed. We also re-analyzed the feasibility of the system and took possible risks into account. All the progress in this phase was comprehensively detailed in the project proposals.

### 2.2.3. Extended Analysis Phase (30 Jan – 15 Feb 2017)

We extended our analysis phase from the former plan to thoroughly analyze overall of the model. We analyze the project using analysis project consisting of functional models, structural models and behavioral models. In this phase, about 85 percent of the plan was successfully executed. The major tasks done in this phase were the analysis models of the core features of the system. The remaining tasks were analysis model of minor features. All of the progress in this phase is fully documented in the next section of this documentation.

### 2.2.4. Design (16 Feb 2017 – 17 Feb 2017)

We chose the technology and development stack as well as designed overall infrastructure for the system. We planned to support massive scalability of the system in case that the system user grows drastically. We chose to design the system with distributed system and service oriented principle. Half of the tasks were still remained in this phase due to the complexity of the design. The remaining tasks include database design and user interface design. We planned to carry out the remaining works again in each implementation phase.

### 2.2.5. Implement (18 Feb 2017 – 27 Feb 2017)

The team performed the infrastructure implementation according to the design we had planned in the former phase. Over 90 percent of the works was done in this phase. Only deployment on the server was postponed to the end of phase I to save the production cost.

### 2.2.6. Implementation Phase I (28 Feb 2017 – 18 March 2017)

This phase is the first feature implementation phase. The main features for our content generator were developed. User authentication, Problem-Solution page, Content Editor and Feed, which is the home page for the user, were done. The user interface was designed along with technical implementation. At present, 30 percent of work is done. We need an extension period of time for this phase to be completely finished as the delay because of the interrupt from our team members' main jobs or side project work.

The finished tasks and the descriptions are shown in the Development Progress in 2.3 below.

## 2.3. Development Progress

### 2.3.1. Creating a Persona

To develop and design user interface from our little understanding of actual users to meet both system requirements and ease of use, we decided to firstly create a persona which can be used to guide our team in unbiasedly mocking up user interface as well as designing customer journey experience throughout the system from a user perspective. This prevents generating the plethora of features and visual components which are unnecessary and can overwhelm a user at the first glance.

Diving into a persona we created, her name is Fah-Sai. She is 17 years old and lives in Bangkok. As in the technology era, she has a smartphone and desktop computer. Since she is still a high school student, her primary dream is to be eligible for the prominent university. In addition, she is always keen on learning new things by herself, so she tends to find learning resources, such as online/offline exercises, courses, books, etc., in order to practice and improve skills in her free time.

As aforementioned, we used Fah-Sai as a persona to shift our perspective to an actual user one. However, a persona can be changed over the time when we find out and realize that current persona is no longer relevant to our actual user. The change of persona can occur for example after initially rolling out the system.

### 2.3.2. Sketch and Wireframe

After creating a persona, we sketch the visual interface by converting from textual system requirements in order to consolidate them to be more concrete and to facilitate communicating ideas among our team. We intentionally used marker pens to roughly sketch, since we do not want a design in this stage to be too much elaborated, but we want to keep the interface at high level. Then, all sketches from each team member were reviewed one at a time and were merged to construct a wireframe of each page. In a nutshell, a wireframe is de facto a low-fidelity user interface. We created a wireframe to arrange components to best fit in a screen without concerning the corporate identity. Figure 2-4 below illustrates the problem page's wireframe.

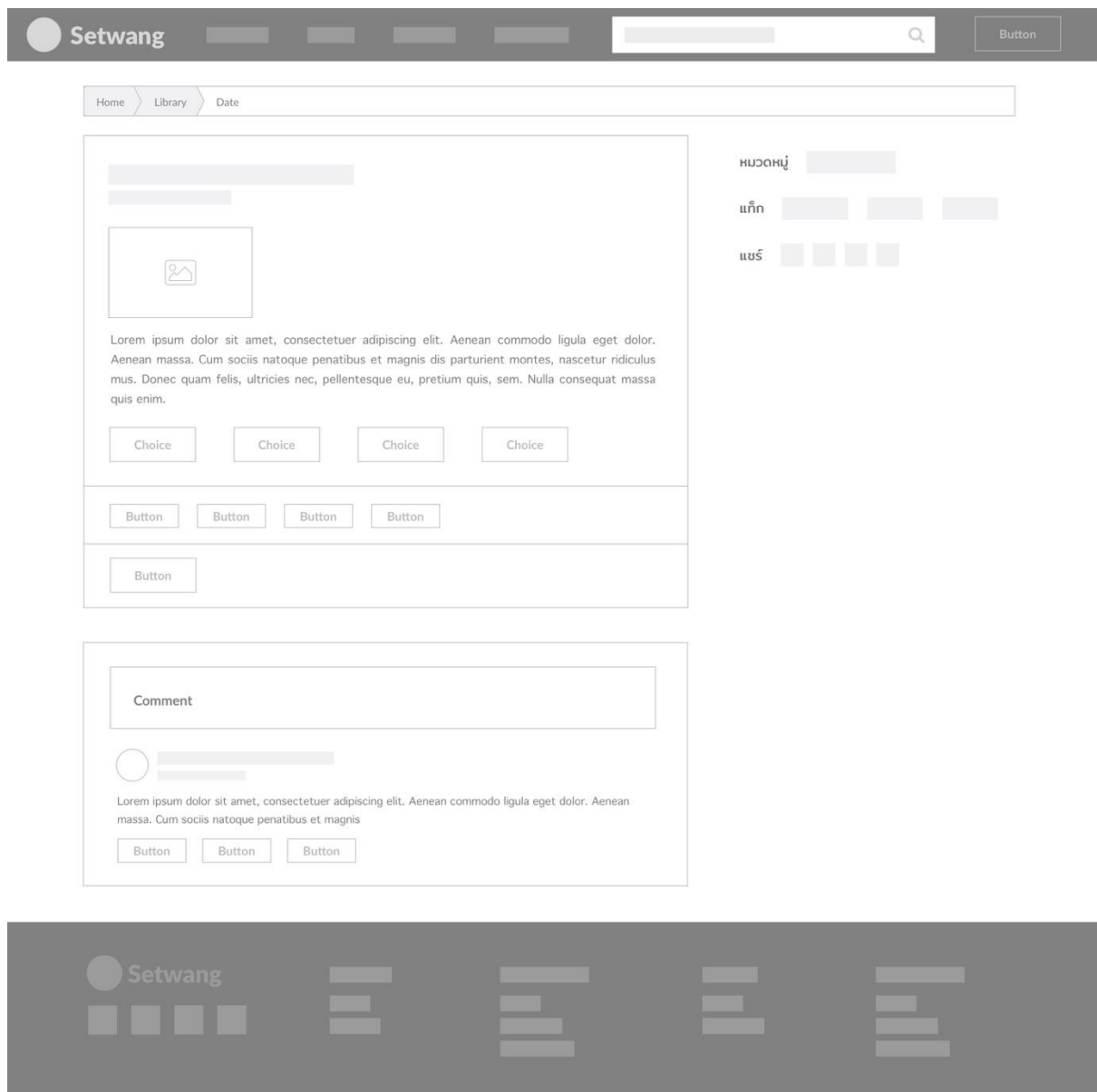


Figure 2-4: Problem page's wireframe

## 2.3.3. High-fidelity User Interface

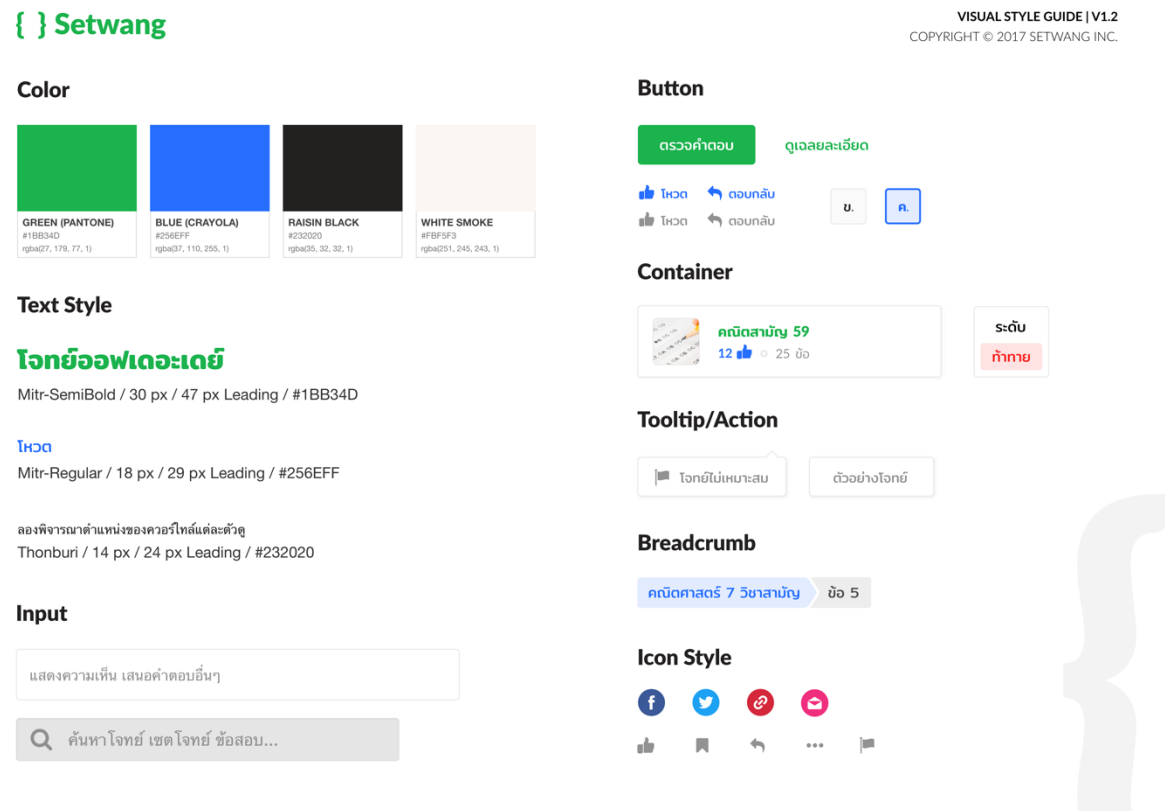


Figure 2-5: Visual style guide

In the next step, each wireframe was streamlined by taking aesthetic into account. A primary product of this step is the high-fidelity user interface which is later fed to front-end development stage. In order to create it, we firstly developed the corporate identity and its byproduct, a visual style guide, which define a guideline for every aesthetic component as shown in Figure 2-5. Their main advantage is to let developers and designers work concurrently while keeping the design congruent and consistent across all pages. After that, the high-fidelity user interface was created by applying such the corporate identity along with the style guide to a wireframe. The Figure 2-6 illustrates the high-fidelity user interface of a problem page deriving from its wireframe as described.

Figure 2-6: High-fidelity user interface of a problem page

Figure 2-6: High-fidelity user interface of a problem page

The breakdowns of this page are discussed in terms of their design background and underlying purpose in the following sections.

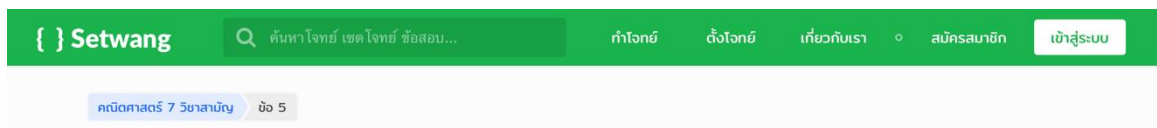


Figure 2-7: Navigation section

As shown in the Figure 2-7 above, the search bar was included right away in a navigation bar. It helps a user browsing through problems, problem sets, and virtual exams in our system. Moreover, we emphasize the login button at the top right corner because we want to encourage a user who come across to the website to engage with the system as a member. Below the navigation bar, we added breadcrumb to aid users easily going back and forth, since it keeps track of their locations within the website.

**คณิตศาสตร์ 59: ค่าเฉลี่ยเลขคณิต**

Parinthorn S. • โพสต์เมื่อ 9 มี.ค. 2560

ระดับ: **ท้าทาย**

Above figure is not directly relevant to this problem. Its main purpose is to represent problem display's variety and to terrify you when you see this problem in the first place. Let's take a deeper look in a "real" problem. A set of numerical data consists of  $x_1, x_2, x_3, x_4$ , and  $x_5$ . If the first, second, and third quartile of it equal 2, 9, and 10 respectively, find the arithmetic mean of this data set.

ภาษาไทยหลังจากนี้เป็นแค่ค่าแปลของโจทย์ปัญหาข้างต้นเพื่อทดสอบทักษะภาษาไทยแบบมีหัวและทำให้โจทย์ดูยาวขึ้น ข้อมูลชุดหนึ่งประกอบด้วย  $x_1, x_2, x_3, x_4, x_5$  ถ้าควอไทล์ที่หนึ่ง ที่สอง และที่สาม เท่ากับ 2, 9 และ 10 ตามลำดับ แล้วค่าเฉลี่ยเลขคณิตของข้อมูลชุดนี้เท่ากับข้อใด?

ก. 6.7

ข. 7.2

**ค. 6.6**

ง. 10.4

12 ใจหวด • ใจหวด ใจหวด ใจหวด

**ตรวจคำตอบ** ดูเฉลยละเอียด

Figure 2-8: Problem box section

The next component is a problem box which contains all problem details as shown in Figure 2-8. Because we want it to be compatible with all problem styles, it is designed to be as simple as possible. To achieve that, we decide to use a combination of white background and black serif font type to assure that problem content is clearly legible. We also emphasize the check answer button over the view

detailed solution one, because we motivate users to initially solve a problem instead of moving toward to view its solution directly.

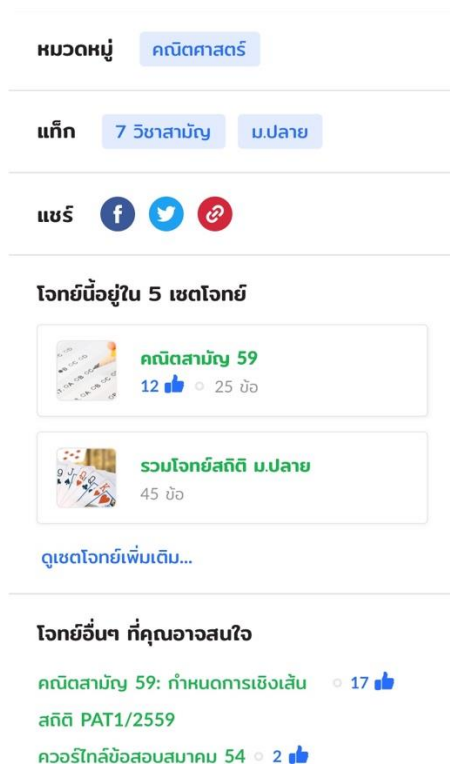


Figure 2-9: Side section

As shown in the Figure 2-9 above, a side section provides the various ways to find other problems to solve after having done with the current one. For example, users can view the remaining problems in the same problem set or reach to a problem in recommendation section. Furthermore, users can also share it to the social media including Facebook, Twitter or share by a permanent link. This helps the system gain more audiences from those channels and keeps the community alive.

### 3. Problems and Solutions

Table 3-1: Problems and solutions

Problems	Solutions
Cannot separate analysis model of each use case into each phase as defined in the proposal because the relationship among each system and the use case.	Shorten the analysis model in phase 1 – 4 and add more time for overall analysis period. We change our plan by adding 'Extended Analysis' period to the original plan.
The former system name (Mongbon) had negative meaning and it sounded unreliable while our system need to be trustworthy.	We decided to change the name by applying keyword brainstorming technique to come up with a new official name with a better meaning.
Analysis model was inconsistent because of parallel work of each small team.	Walkthrough each diagram together to validate and verify the model and to synchronize the work and tune up the model.



Problems	Solutions
A lot of implementation work was postponed because of each team member responsibility for his/her main job or side project.	Adjust the plan by trying to the tasks to the team members by further considering their availability and ability.
Unfamiliarity in some tools in development process.	For the tools that we can substitute with the tools which we are familiar with, we try to use the tools that we are accustomed to and roll back the part which unfamiliar tools were used. For the part which is necessary to the project, we try to go through the official tutorial and tools' documentation.

## 4. Future Development Plan

In order to maximize the time left in the development time frame as well as increase the efficiency of the work plan, we analyze the problem happened during every development phase and come up with the plan to manage the team to finish all tasks.

First, we need to assign the work to each member considering their specialties, skills and abilities so that the task can be fulfilled with the least time possible. In addition, we applied a strategy from scrum development which is using Kanban board to synchronize and track the progress of all the tasks being done by each member and to strictly keep up with the plan in Gantt's Chart. Besides, we will have to always get the feedback from the user to exactly know what does actual users need so that we can adjust our system to satisfy them.

## 5. Change of Definition of Terms and System Requirement

After reviewing the requirements and the need and benefits of the users, we found out that some of the function might not be necessary for the system. Therefore, we change some of the requirements of the system as described later in this section. We have also modified some of the term to make it clearer and more concise. All the rewritten terms and definition are in Appendix A

We cut off the **downvote** function for every votable item because of two main reasons. The first reason is that we want to reduce the alternative to interact with a votable items such as problem, problem set, comment and subcomment. We believe that too much alternatives for interacting with the post will reduce the possibility for interacting with the post. It will give an easier decision for the users of the system to just upvote if they like or think that the content of each votable item is worth sharing and leave it neutral if they think it doesn't worth spreading the content. The second reason is about the mood and tone of the system. We want the system to be a good place for open source. Thus, if there is a downvote system then it will add negative emotion to the system and will also discourage the content provider or generator as well.

We also add a **virtual exam registration system** to imitate the actual exam. This gives the better of having more familiarity and less confusion to the user.

In conclusion, we have a minor change on the functionalities requirements by removing downvote function from our system. All the rewritten system requirements are in Appendix B.

## 6. List of Stakeholders and Responsibilities

We assign the work to each stakeholder considering two main factors: the workload of the work and the specialty of each stakeholders. The responsibilities are shown in Table 6-1 and Table 6-2.

Table 6-1: Analysis Model responsibilities

Analysis Models	Stakeholders
Virtual Exam System	Kasidit Iamthong
Virtual Exam System	Kawin Liaowongphuthorn
Solve Problem System	Kosate Limpongsa
Solve Problem Set System	Parinthorn Saithong
Solve Problem Set System	Sirinthra Chantharaj
Solve Problem System	Thanat Jatuphatthachat

Table 6-2: Implementation responsibilities

Features	Stakeholders
Set up infrastructure, Content Editor, Virtual Exam	Kasidit Iamthong
Discussion Thread, Bookmarking, Voting	Kawin Liaowongphuthorn
Set up infrastructure, Authentication, Virtual Exam	Kosate Limpongsa
Problem Set, Landing Page, Feed	Parinthorn Saithong
Problem Set, Search, Flag	Sirinthra Chantharaj
Problem, Comment System	Thanat Jatuphatthachat

## 7. Method of System Analysis Modeling

We use three analysis models to analyze overall system as listed below:

### 7.1. Functional Model

- 7.1.1. Use-Case Diagram
- 7.1.2. Use-Case Description
- 7.1.3. Activity Diagram

### 7.2. Structural Model

- 7.2.1. Class Diagram
- 7.2.2. CRC Card
- 7.2.3. Object Diagram

### 7.3. Behavioral Model

- 7.3.1. Sequence Diagram
- 7.3.2. Communication Diagram
- 7.3.3. State Diagram
- 7.3.4. CRUDE Matrix

## 8. Functional Modeling

The development team use the functional model to describe the system in the use point of view using three functional models: Use-Case Diagram, Use-Case Description, and Activity Diagram.

### 8.1. Use-Case Diagram

Use-Case Diagram is used to describe overall functionality use of the system together with the related actor of the system. It is also used for validating the functionalities with the system requirements.

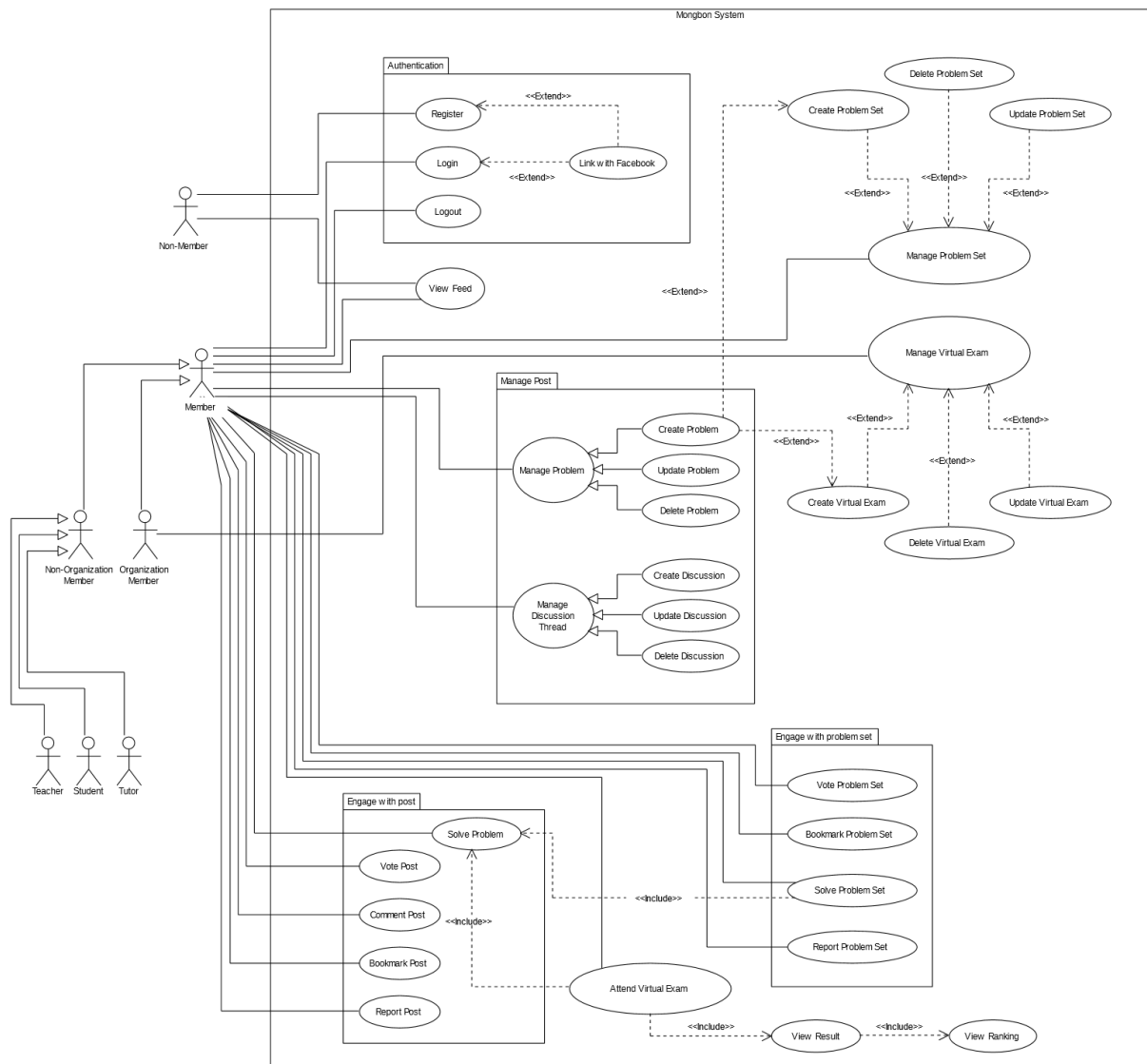


Figure 8-1: Use-case diagram

### 8.2. Activity Diagrams and Use-Case Descriptions

The development team use the activity diagram and to further describe the flow for each Use-Case. The team also use Use-Case Description to identify stakeholders, relationship among the Use-Case and the order of the activities occurring in each Use-Case. Moreover, the Use-Case Description is used for verifying the Activity Diagram and Use-Case Diagram. The Activity

Diagram is verified by the flow listed in the Use-Case Description, while the relationships, primary actors and stakeholders must be consistent with the Use-Case diagram.

### 8.2.1. Use-Case 1: Solve a problem

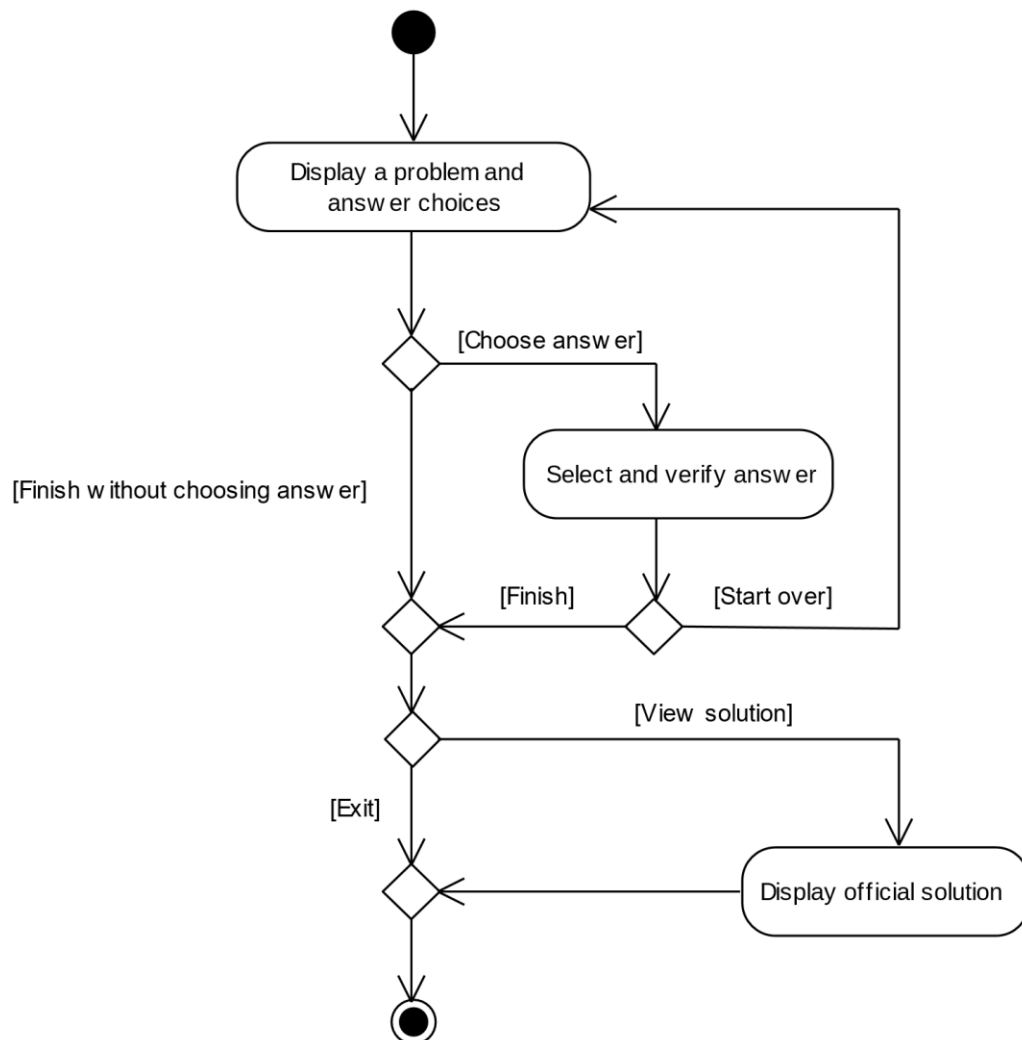


Figure 8-2: Activity diagram of "Solve a problem" use case

Table 8-1: Use-case description of “Solve a problem” use case

<b>Use Case Name:</b> Solve a problem	<b>ID:</b> 1	<b>Important Level:</b> High
<b>Primary Actor:</b> Non-Organization Member	<b>Use Case Type:</b> Detail, Essential	
<b>Stakeholders and Interests:</b> A student – wants to solve a problem to enhance his/her skill		
<b>Brief Description:</b> This use case describes how a member user solve a problem in the system.		
<b>Precondition:</b> User must already register and log in into the system.		
<b>Trigger:</b> Member enters a problem page via a feed, search result, or direct link.		
<b>Type:</b> External		
<b>Relationships:</b> <b>Association:</b> Member <b>Include:</b> <b>Extend:</b> <b>Generalization:</b>		
<b>Normal Flow of Events:</b> <div><div>1. The system displays a problem and answer choices.</div><div>2. A member user solves the problem and then select an answer.</div><div>3. Verify answer and show the result of the answer whether it is correct or not.</div><div>4. The system let user to solve the problem again if the user wants.<div><div>a. If the user wants to solve the problem again, the S-1: Start over is performed.</div></div></div><div>5. The system asks whether the user wish to view the official solution.<div><div>a. If the user wants to see the official solution of the answer, the S-2: Show the solution is performed.</div></div></div></div>		
<b>SubFlows:</b> <div><div>S-1: Start over<div><div>1. The system displays the problem again and resets the selected answer choice.</div><div>2. The system start over by going back to the Normal Flow step 1</div></div></div><div>S-2: Show the solution<div><div>1. The system displays the official solution which is hidden in the first place.</div></div></div></div>		
<b>Alternate/Exceptional Flows:</b>		
<b>Postcondition:</b> The system saves logs to the database.		

## 8.2.2. Use-Case 2: Solve a problem set

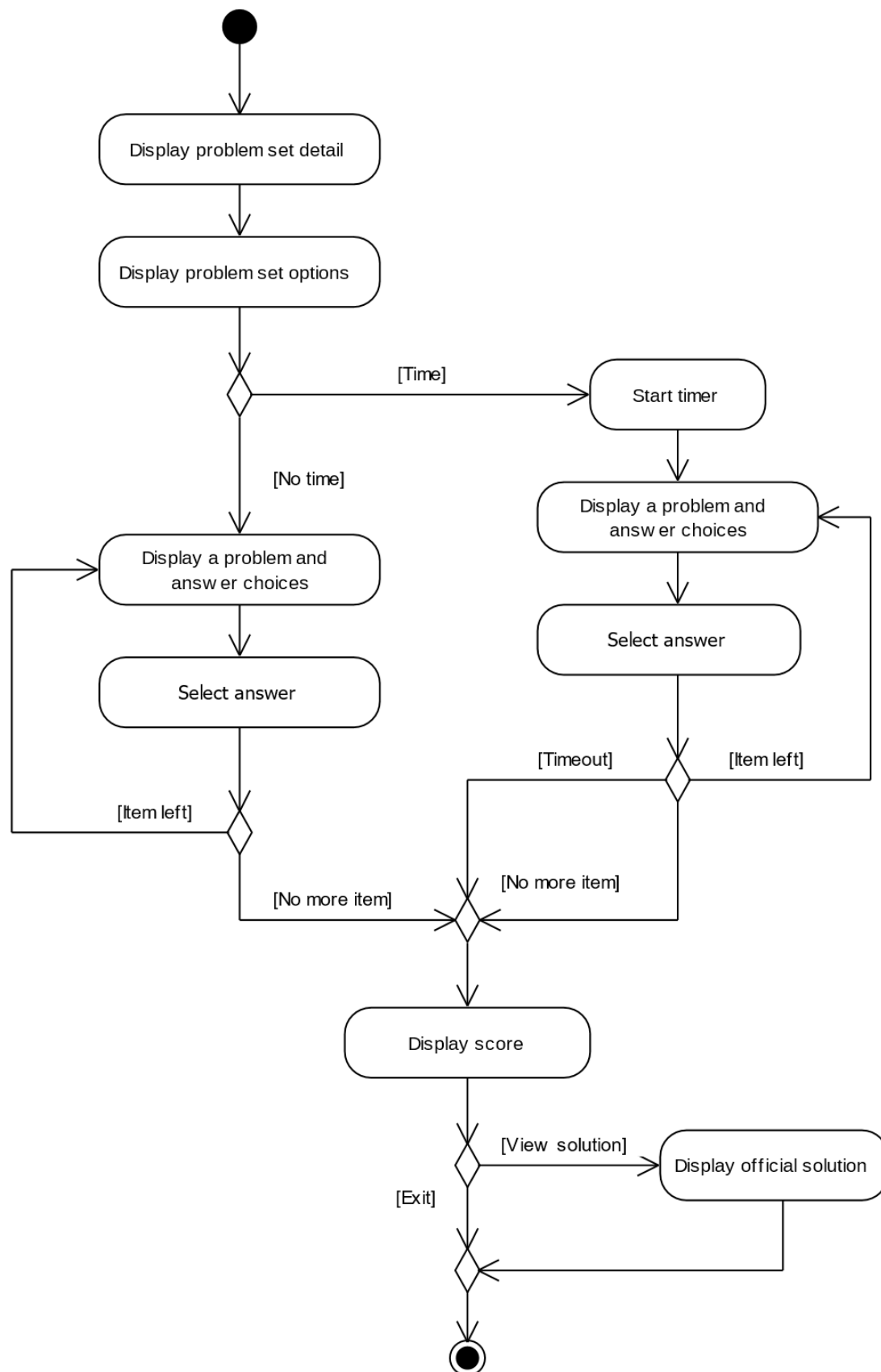


Figure 8-3: Activity diagram of "Solve a problem set" use case

Table 8-2: Use-case description of “Solve a problem set” use case

<b>Use Case Name:</b> Solve a problem set	<b>ID:</b> 2	<b>Important Level:</b> Medium
<b>Primary Actor:</b> Non-Organization Member	<b>Use Case Type:</b> Detail, Essential	
<b>Stakeholders and Interests:</b> A student – wants to solve a problem set to enhance his/her skill		
<b>Brief Description:</b> This use case describes how a member user solve a problem set in the system.		
<b>Precondition:</b> User must already register, log in into the system, and has the right to access the problem set.		
<b>Trigger:</b> Member enters a problem set page via a feed, search result, or direct link. <b>Type:</b> External		
<b>Relationships:</b> <b>Association:</b> Member <b>Include:</b> Solve a problem <b>Extend:</b> <b>Generalization:</b>		
<b>Normal Flow of Events:</b> <div><div>1. The member enters a problem set page to see problem set’s detail and practice options.</div><div>2. The member chooses practice options such as whether to include the timer or not.</div><div>3. The member starts practice session.<div><div>a. If the member wants to include the timer, the S-1: Timed practice is performed.</div><div>b. If the member does not want to include the timer, the S-2: Non-timed practice is performed.</div></div></div><div>4. The member sees the score.</div><div>5. If the member wants to see the official solution, the S-3: Show the solution is performed.</div></div>		
<b>SubFlows:</b> <div><div>S-1: Timed practice<div><div>1. The member sees the countdown timer appeared.</div><div>2. While time is still left and the practice session is not finished,<div><div>a. If the member want to continue solving the problems, the member continues solving the problems.</div><div>b. Otherwise, the member finishes the practice session.</div></div></div></div></div><div>S-2: Non-timed practice<div><div>1. While the member want to continue solving the problems, the member continues solving the problems.</div><div>2. The member finishes the practice session.</div></div></div><div>S-3: Show the solution<div><div>1. The system displays the official solution which is hidden in the first place.</div></div></div></div>		
<b>Alternate/Exceptional Flows:</b> 5a. Unless the member want to see the official solution, exit the practice session.		
<b>Postcondition:</b> The system saves logs to the database.		

## 8.2.3. Use-Case 3: Attend a virtual exam

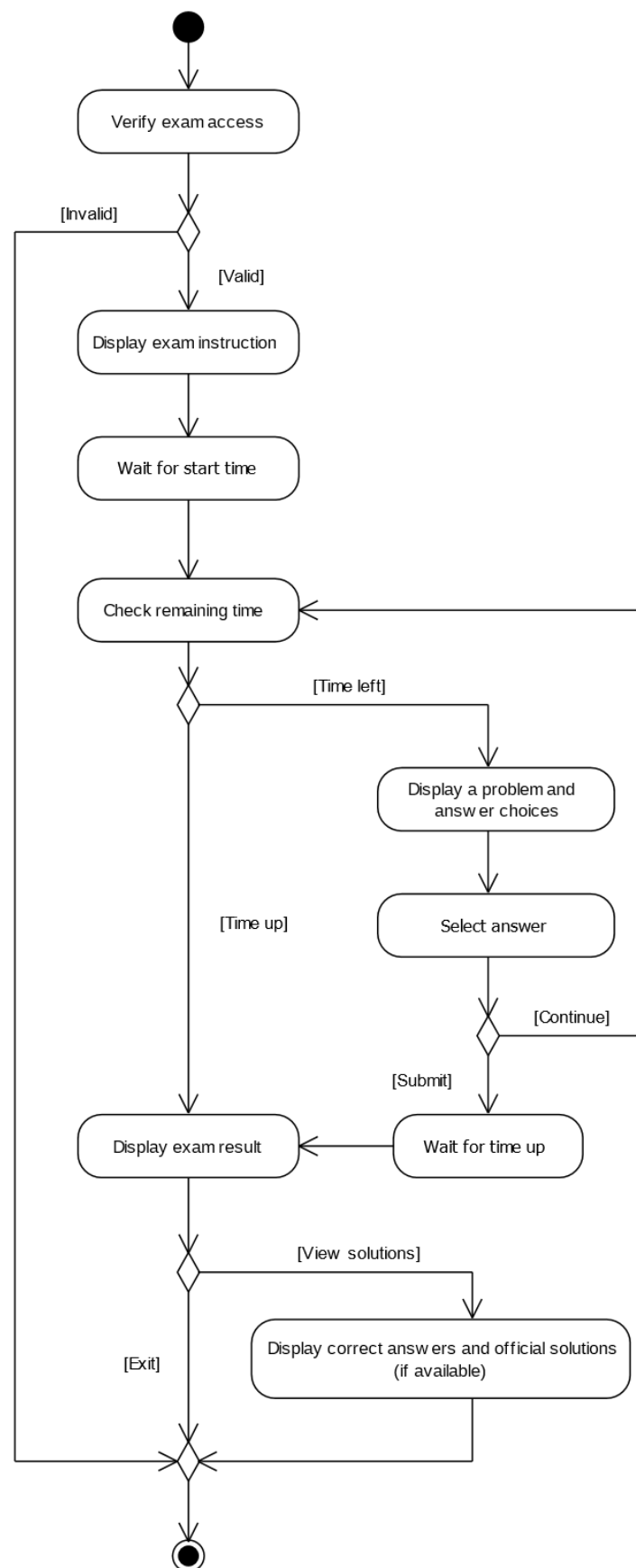


Figure 8-4: Activity diagram of “Attend a virtual exam” use case



Table 8-3: Use-case description of “Attend a virtual exam” use case

<b>Use Case Name:</b> Attend a virtual exam	<b>ID:</b> 3	<b>Important Level:</b> High
<b>Primary Actor:</b> Member	<b>Use Case Type:</b> Detail, Essential	
<b>Stakeholders and Interests:</b> Member – wants to participate in a virtual exam Organization – wants to ensure member has a right to access a virtual exam		
<b>Brief Description:</b> This use case describes how a member engage with the system while doing a virtual exam.		
<b>Precondition:</b> User must already register and log in into the system.		
<b>Trigger:</b> Member requests to attend a virtual exam. <b>Type:</b> External		
<b>Relationships:</b> <b>Association:</b> Member <b>Include:</b> Solve a problem, View exam result <b>Extend:</b> <b>Generalization:</b>		
<b>Normal Flow of Events:</b> <div>1. Verify whether the member has the right to access the virtual exam.</div> <div>2. Display the exam instruction.</div> <div>3. Wait for start time.</div> <div>4. Check remaining time.<div>Unless time is up, the S-1: Display problem is performed.</div></div> <div>5. Display exam result.</div> <div>6. Display correct answers and official solutions (if available).</div>		
<b>SubFlows:</b> S-1: Display problem <div>1. Display a problem and answer choices.</div> <div>2. Save selected answer.</div> <div>3. Execute normal flow 4.</div>		
<b>Alternate/Exceptional Flows:</b> 2a. The member has no right to access the virtual exam, the request is rejected. S-1, 3a. If the member submits the exam, wait for time up to display exam result. 6a. Unless the member want to view official solutions, exit the virtual exam.		
<b>Postcondition:</b> The exam results of each participant are saved to the database.		

## 9. Structural Modeling

### 9.1. Class Diagram

The development team use class diagram to describe high level class for the system. The class diagram includes class names, class attributes, and class operations. Furthermore, the diagram also describes the association between each class. Figure 9-1 shows the class diagram using in modeling the system.

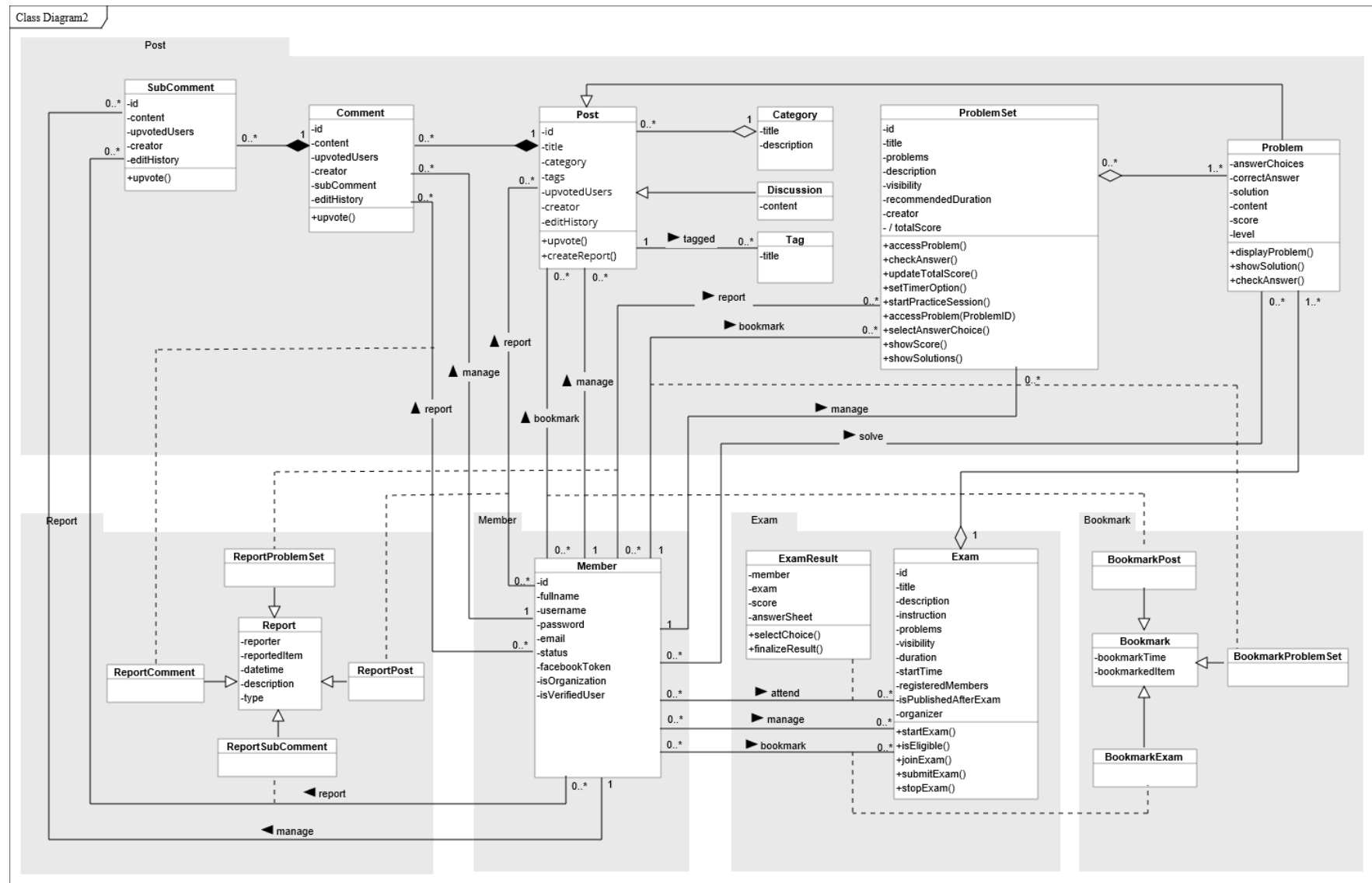


Figure 9-1: Class diagram

## 9.2. Object Diagram

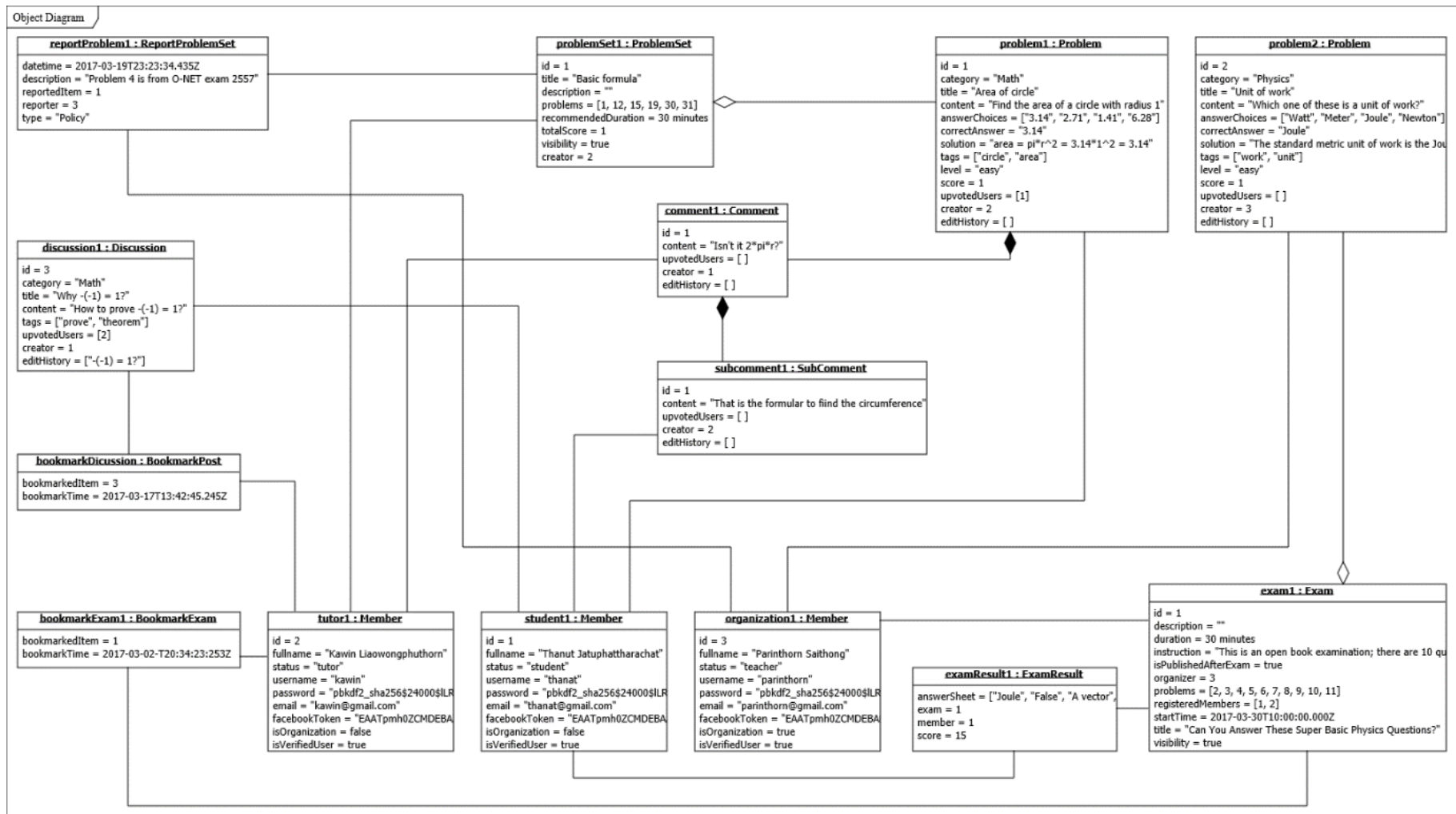


Figure 9-2: Object diagram

### 9.3. Class-responsibility-collaboration (CRC) Card

The CRC card is used to elaborate the detail of the class shown in the class diagram. The CRC card can also be used to verify the correctness of the class diagram. Each CRC card contains the information about the responsibilities of each class which is related to the operation in each class, the attributes of the class and the relationship of each class.

<b>Front:</b>								
<b>Class Name:</b> Exam	<b>ID:</b> CRC-1	<b>Type:</b> Concrete, Domain						
<b>Description:</b> A collection of problems created by an organization		<b>Associated Use Cases:</b> 3						
<b>Responsibilities:</b> <ol style="list-style-type: none"> <li>1a. Start an exam</li> <li>2a. Stop an exam</li> <li>3a. Check member's eligibility</li> <li>4a. Let a user join an exam</li> <li>5a. Let a user submit an exam</li> </ol>		<b>Collaborators:</b> <ol style="list-style-type: none"> <li>1b. -</li> <li>2b. -</li> <li>3b. Member</li> <li>4b. Member</li> <li>5b. Member</li> </ol>						
<b>Back:</b>								
<b>Attributes:</b> <ol style="list-style-type: none"> <li>1. exam id (Number)</li> <li>2. title (String)</li> <li>3. description (String)</li> <li>4. instruction (String)</li> <li>5. problems (Problem [])</li> <li>6. visibility (Boolean)</li> <li>7. duration (TimeSpan)</li> <li>8. start time (DateTime)</li> <li>9. registered member (Member [])</li> <li>10. is published after exam (Boolean)</li> <li>11. organizer (Member)</li> </ol>								
<b>Relationships:</b> <table border="0"> <tr> <td><b>Generalization (a-kind-of):</b></td> <td>None</td> </tr> <tr> <td><b>Aggregation (has-parts):</b></td> <td>Problem</td> </tr> <tr> <td><b>Other Associations:</b></td> <td>Member</td> </tr> </table>			<b>Generalization (a-kind-of):</b>	None	<b>Aggregation (has-parts):</b>	Problem	<b>Other Associations:</b>	Member
<b>Generalization (a-kind-of):</b>	None							
<b>Aggregation (has-parts):</b>	Problem							
<b>Other Associations:</b>	Member							

**Front:**

<b>Class Name:</b> Member	<b>ID:</b> CRC-2	<b>Type:</b> Concrete, Domain
<b>Description:</b> A registered user in the system	<b>Associated Use Cases:</b> 1, 2, 3	
<b>Responsibilities:</b> 1a. Login 2a. Logout	<b>Collaborators:</b> 1b. - 2b. -	

**Back:****Attributes:**

1. member id (Number)
2. fullname (String)
3. username (String)
4. password (String)
5. email (String)
6. member status (String)
7. facebook token (String)
8. is organization (Boolean)
9. is verified user (Boolean)

**Relationships:**

<b>Generalization (a-kind-of):</b>	None
<b>Aggregation (has-parts):</b>	None
<b>Other Associations:</b>	Post, Problem, ProblemSet, Exam, Comment, SubComment

**Front:**

<b>Class Name:</b> Problem	<b>ID:</b> CRC-3	<b>Type:</b> Concrete, Domain
<b>Description:</b> An educational question which members can practice on		<b>Associated Use Cases:</b> 1
<b>Responsibilities:</b> 1a. Display a problem 2a. Show the solution 3a. Check the correctness of the answer 4a. Upvote 5a. Create report	<b>Collaborators:</b> 1b. - 2b. - 3b. - 4b. Member 5b. Member	

**Back:****Attributes:**

1. problem id (Number)
2. title (String)
3. category (Category)
4. tags (Tag [])
5. creator (User)
6. edit history (String [])
7. upvoted users (User [])
8. answer choices (String [])
9. correct answer (String)
10. solution (String)
11. content (String)
12. score (Number)
13. level (String)

**Relationships:**

- Generalization (a-kind-of):** Post
- Aggregation (has-parts):** None
- Other Associations:** Member, ProblemSet, Exam

## 10. Behavioral Modeling

### 10.1. Sequence diagram

To clearly understand, we use sequence diagram to explain a sequence of operation. The following diagrams are parts of the system sequence diagram.

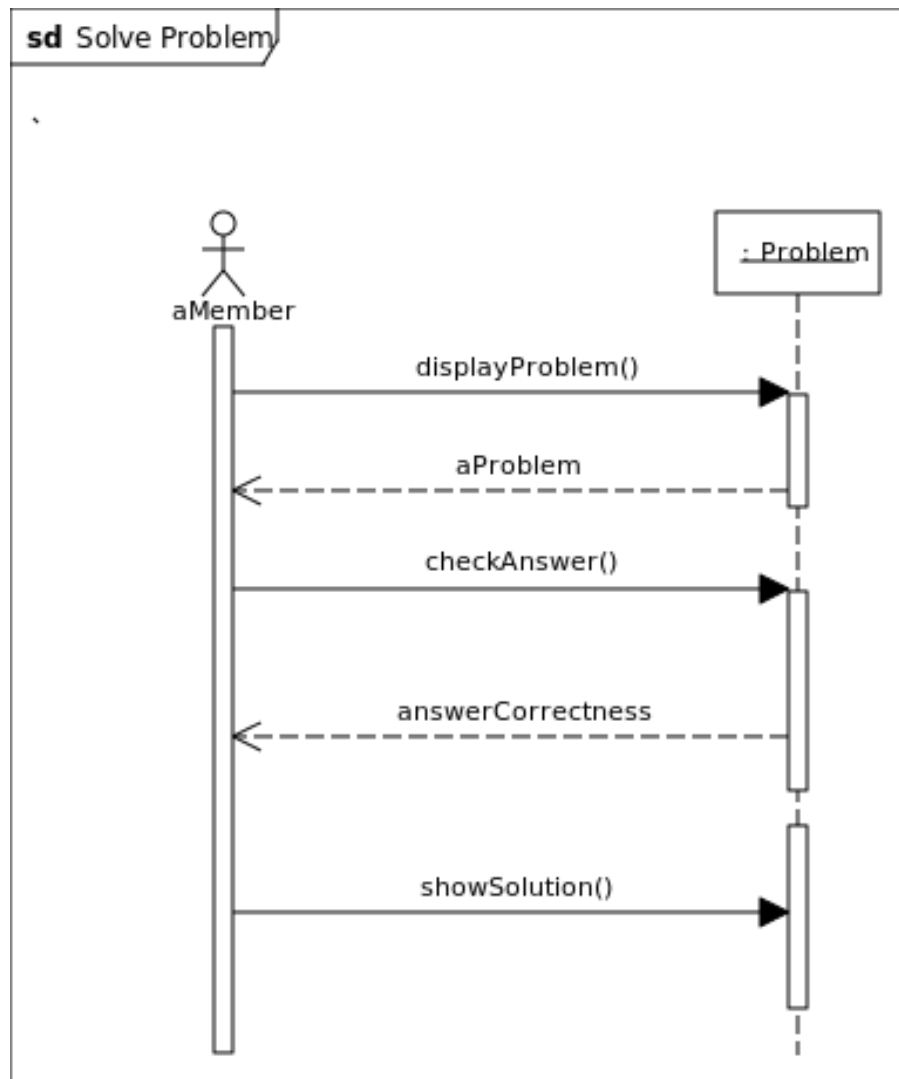


Figure 10-1: Sequence diagram of "Solve a problem" use case

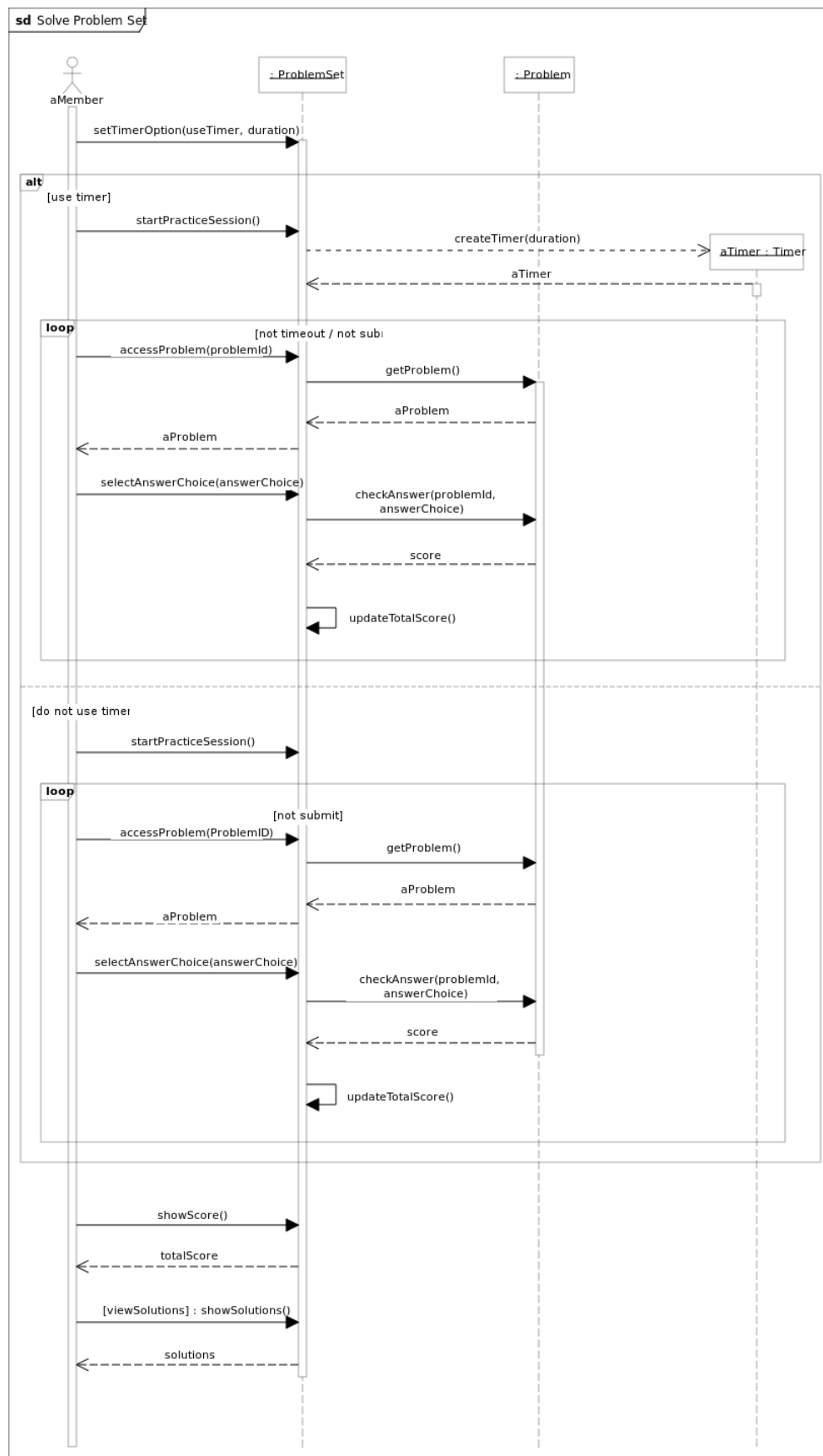


Figure 10-2: Sequence diagram of “Solve a problem set” use case



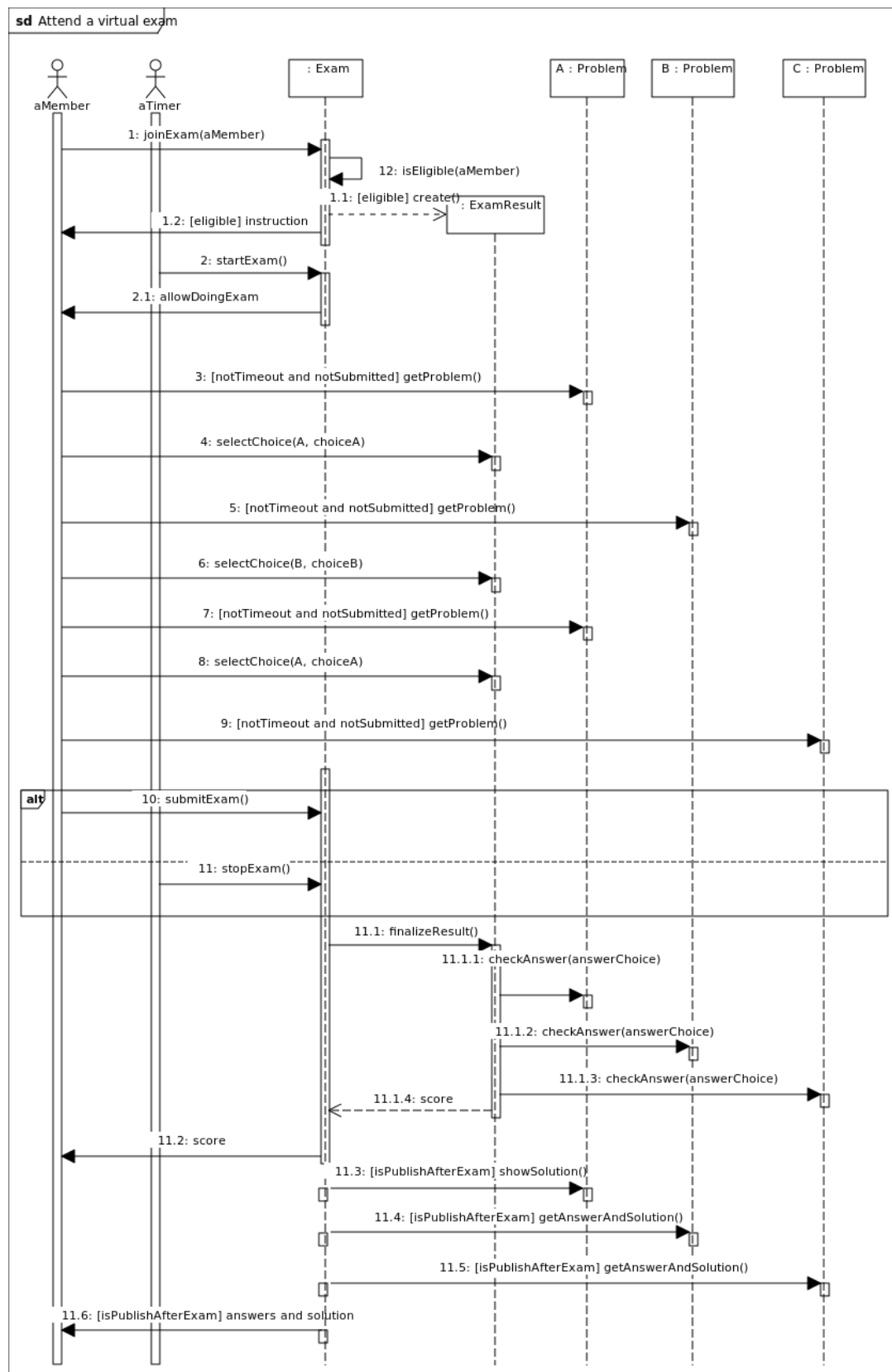


Figure 10-3: Sequence diagram of “Attend a virtual exam” use case for virtual exam with 3 problems scenario

## 10.2. Communication diagram

We use communication diagram to describe communication process for each class and actor. The following diagrams are part of the system communication diagrams:

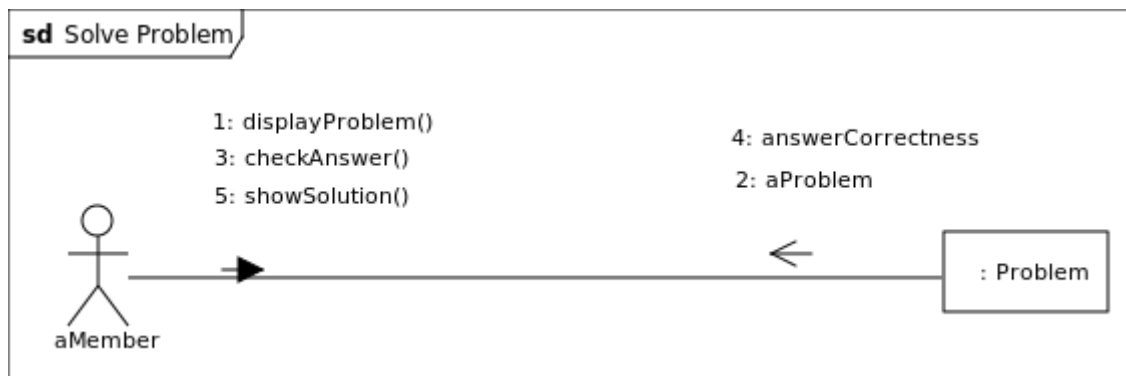


Figure 10-4: Communication diagram of “Solve a problem” use case

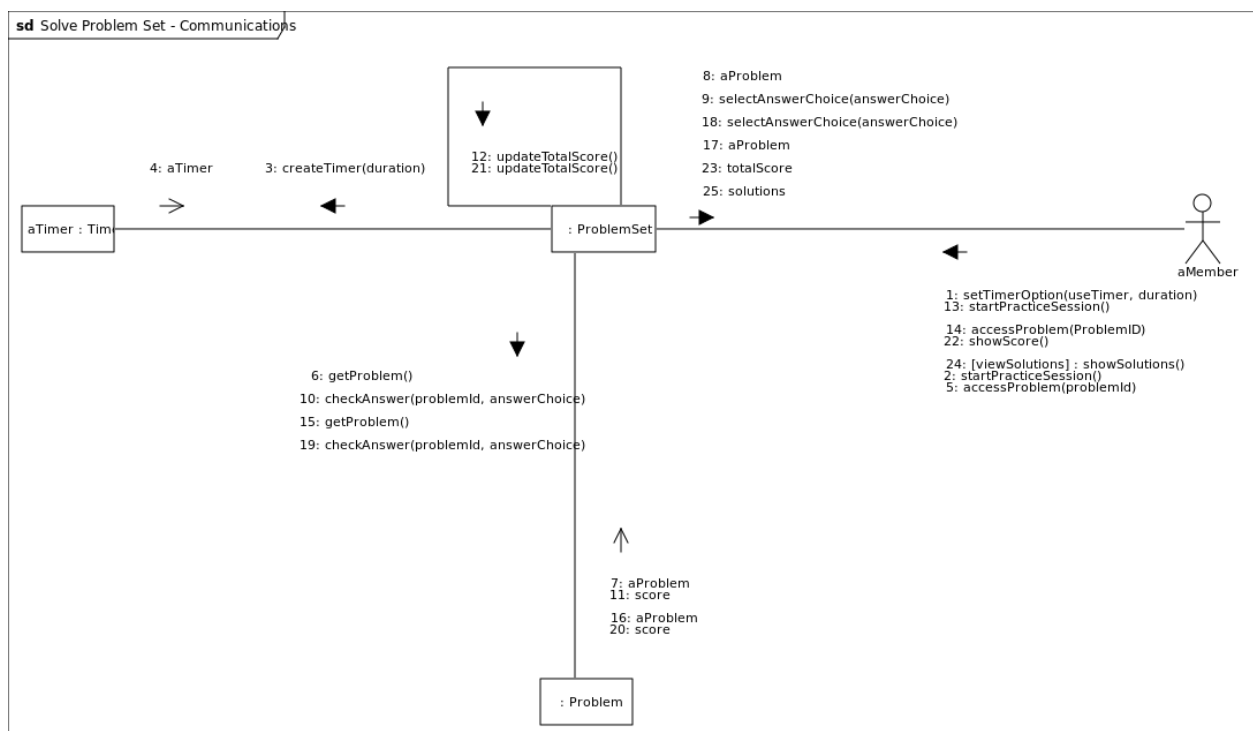


Figure 10-5: Communication diagram of “Solve a problem set” use case

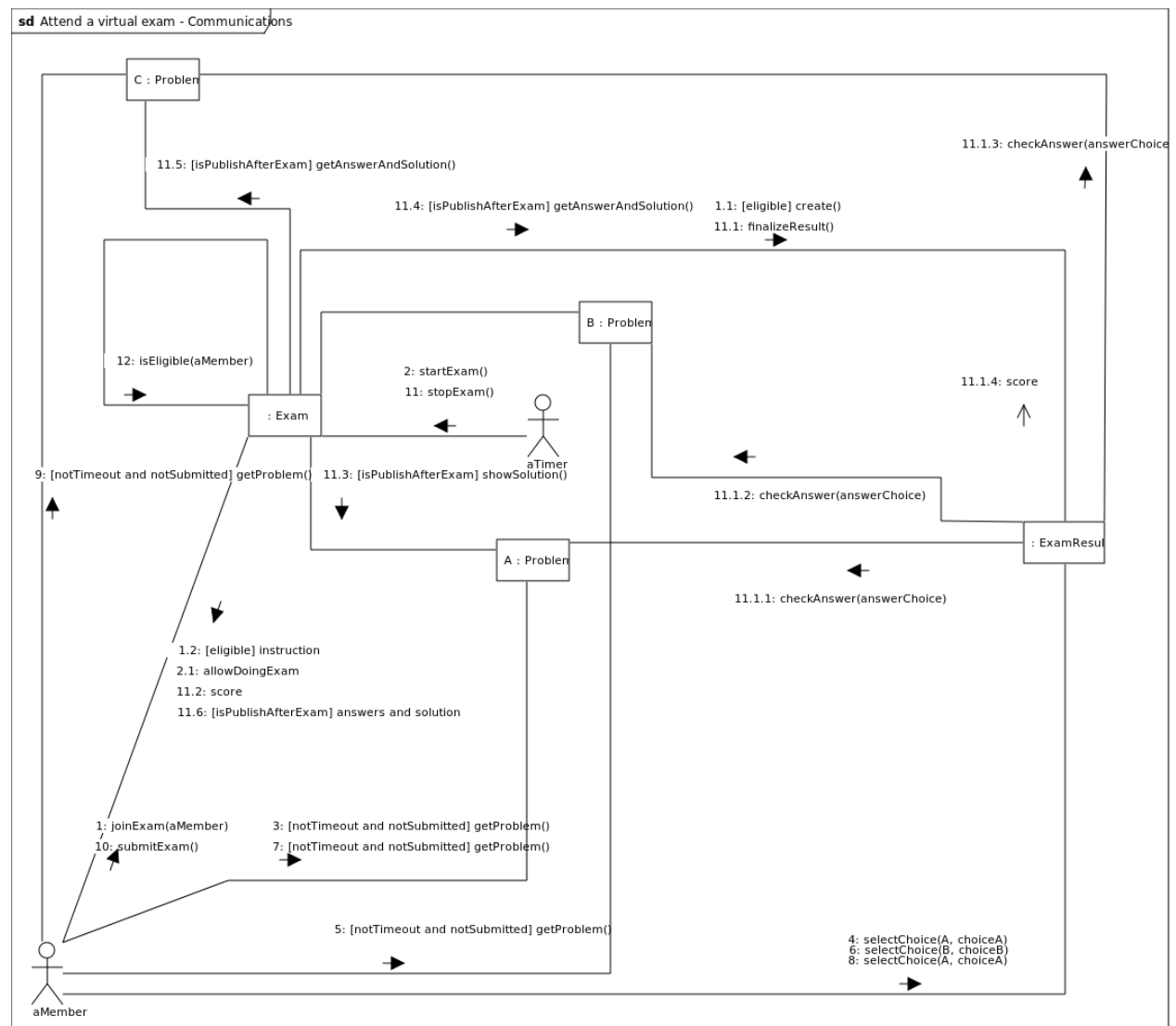


Figure 10-6: Communication diagram for “Attend a virtual exam” use case for virtual exam with 3 problems scenario

### 10.3. Behavioral state machine

We use behavioral state machine to show all possible states and state transition that consist in the system. The following diagrams are post state diagram and exam state diagram.

#### 10.3.1. Exam class

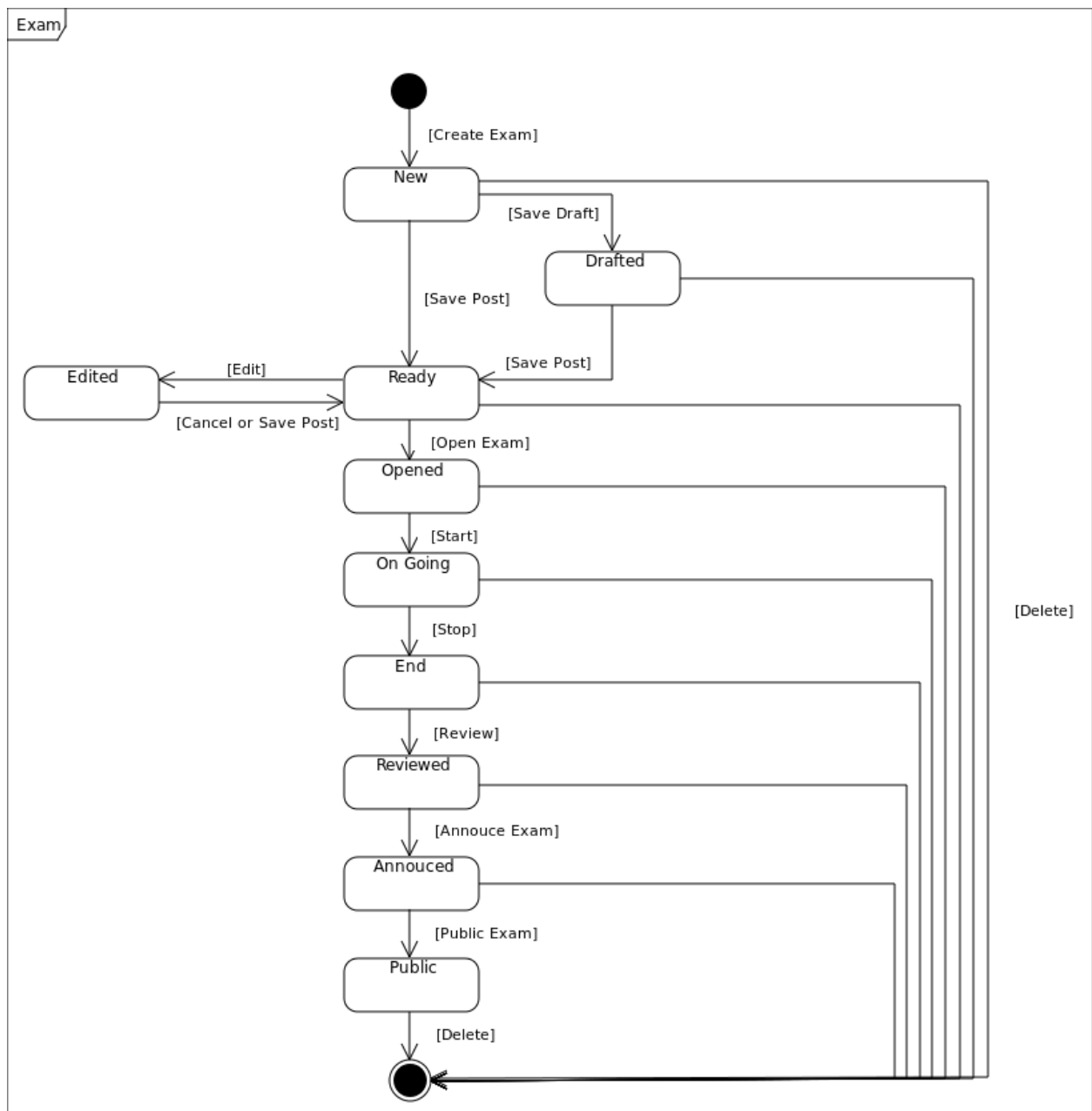


Figure 10-7: Behavioral state machine for Exam class

#### State description

- For all states except *Edited*, if organization member deletes an exam, the state of the exam will change to final state.
- Start: when organization member wants to create an exam, the exam will change to *New* state.
- New state: there 2 options for organization member to do:
  - Save Draft: the exam will change to *Drafted* state.
  - Save Post: the exam will change to *Ready* state that ready to open to member.

- Draft state: if organization member finishes drafting exam and saves post, the exam will change to *Ready* state.
- Ready state: there 2 options for organization member to do:
  1. If organization member wants to edit an exam, the exam will change to *Edited* state.
  2. If organization member wants to open an exam, the exam will change to *Opened* state.
- Edited state: when finish editing or cancel editing, the exam will return to *Ready* state.
- Opened state: member can join an exam. When time is come, the timer will start the exam. Then, the exam will change to *Ongoing* state.
- Ongoing state: member can solve problems in an exam. When time is up, the timer will stop the exam. The exam will then change to *End* state.
- End state: an organization member will check the correctness of both exam and exam results. When finishing review, the exam will change to *Reviewed* state.
- Reviewed state: organization member announces the results and solutions. The exam will change to *Announced* state.
- Announced state: if organization member publishes the exam to ordinary member. The exam will change to *Public* state.
- Public state: allow ordinary member to take the virtual exam as a problem set.

### 10.3.2. Post class

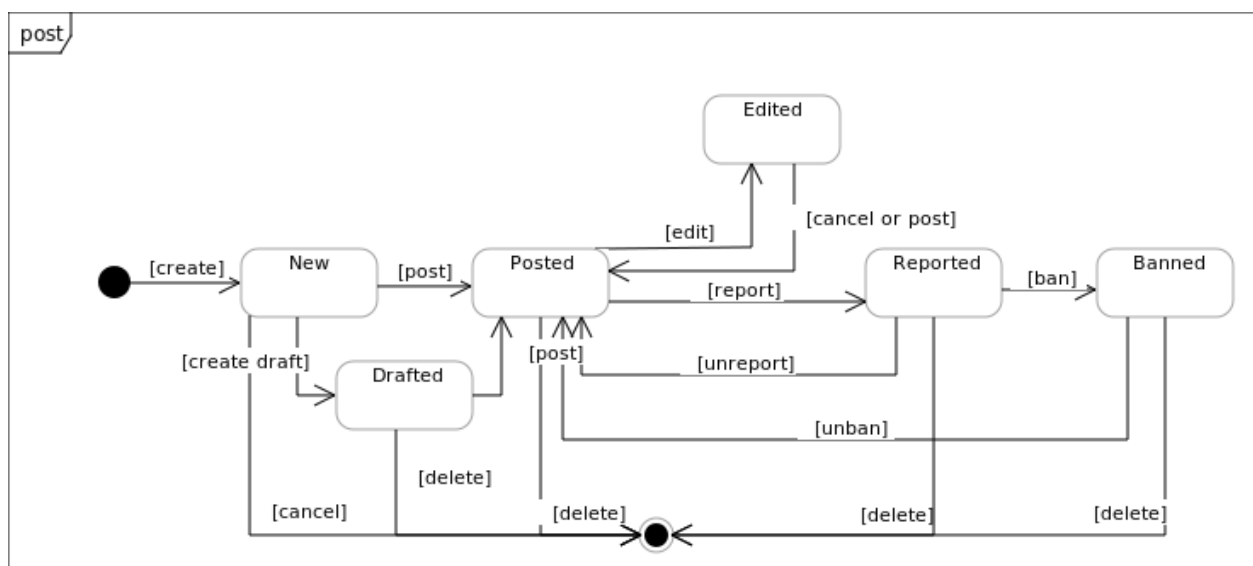


Figure 10-8: Behavioral state machine for Post class

#### State description

- Start: when member create post, the post will change to *New* state.
- New state, there are 3 options for a member:
  1. Post: if a member ready to publish a post, the post will change to *Posted* state.
  2. Create draft: if a member is not yet ready to publish a post, a member can save a draft as a draft instead. The post will then change to *Drafted* state
  3. Cancel: if a member does not want to post anymore. The post will then change to final state.
- Draft state: if a member is ready to publish the exam. The exam will change to *Posted* state. If member don't want to publish anymore and delete the exam instead. The exam will change to final state.
- Posted state, there are 3 options to do:
  1. Edit: if member wants to edit the post, the post will change to *Edited* state.

2. Report: if some member see that a post is violate the policy, he/she can report the post and it will change to *Reported* state.
  3. Delete: if member don't want that post anymore. The post will change to final state.
- Reported state, if admin considers that a post violates the policy. The post will change to *Banned* state. If a member deletes the post. The post will change to final state.
  - Banned state, if admin considers that the post does not violate the policy. The post will change to *Posted* state. If a member deletes the post. The post will change to final state.

#### 10.4. CRUDE matrix

We use CRUDE matrix to show the ability to call function between classes in the system. By verifying with class diagram, CRUDE matrix can be shown as in Table 10-1.

Table 10-1: CRUDE matrix

[illegible]

# Appendix

## A. Definition of Terms

- A.1. **User:** A person who accesses the system via the website interface. It consists of a student, a tutor, a teacher, and an organization.
- A.2. **Member:** A type of user who is registered to the system.
- A.3. **Organization Member:** A user who is either a high school or an education institute in Thailand. This type of user has an ability to host and manage his virtual exam.
- A.4. **Non-Organization Member:** A general member who accessed the system. It can be a student, a tutor, or a teacher.
- A.5. **Student:** A type of non-organization user who is currently studying in a high school in Thailand and is preparing for a university entrance exam.
- A.6. **Tutor:** A type of non-organization user that assists students in preparing for examinations without institutional connection. A teaching can be in a form of direct one-to-one teaching with single student, teaching to a small group, or can be in a form of prerecorded video for opening in a tutorial school.
- A.7. **Teacher:** A type of non-organization user that teaches a student as a primary job in a high school in Thailand.
- A.8. **Administrator:** A person who looks after the system.
- A.9. **Post:** A main, user-generated item on the system which can only be created by a member. It can be either a problem or a discussion thread.
- A.10. **Post Creator:** A member who creates a post on the system.
- A.11. **Problem:** An educational question which members can practice on. It consists of a question, content, answer choices (with correct answer marked), and an official solution. There is also a comment section for a user to discuss on the problem.
- A.12. **Answer Choice:** A choice that is the candidate answer to a problem. Each problem must have at least two answer choices.
- A.13. **Correct Answer:** An answer choice that is the most correct to a problem. Each problem must have exactly one correct answer.
- A.14. **Official Solution:** A text describing the way for a specific problem to come up with the correct answer. An official solution is provided by a problem creator.
- A.15. **Problem Creator:** A user who creates a problem on the system.
- A.16. **Problem Solver:** A person who attempts to solve a problem.
- A.17. **Problem Level:** A measurement that describes how hard the problem is. It should be one of these: Straightforward, Intermediate, and Challenging.
- A.18. **Problem Set:** A set of related problems which are grouped together.
- A.19. **Comment:** A user-generated content that can be added to a post to express their idea.
- A.20. **Sub-comment:** A user-generated content that is similar to a comment, but is added to an existing comment instead.
- A.21. **Voting:** A mechanism where a user can choose to upvote votable item.
- A.22. **Upvote:** An action where user recommends an item.
- A.23. **Category:** A class of post that is used for grouping similar posts. A post must belong to exactly one category.



- A.24. **Tag:** An indexed item that is assigned to a post for helping user searches related posts with the same tag assigned. A post can be assigned with several tags. An assignment of a tag to a post is called tagging.
- A.25. **Virtual Exam:** An event where students can join an online exam to do problems within specified time range.
- A.26. **Virtual Exam Organizer:** An organization member who hosts a virtual exam.
- A.27. **Virtual Exam Problem Set:** A problem set that is used in a virtual exam.
- A.28. **Quota:** A virtual point that represents maximum number of students that can joins a virtual exam within specified time range.
- A.29. **Discussion Thread:** A post user creates to be a place to discuss on very specific topic where people can have an idea exchange.
- A.30. **Feed:** An entrance page containing contents curated for users. Its main purpose is to facilitate browsing through any posts in the system by grouping them into semantic categories.

## B. System Requirement

### B.1. Functional Requirement

#### B.1.1. Feed

- B.1.1.1. It should have a list of recent posts.
- B.1.1.2. It should have a list of popular posts.
- B.1.1.3. It should have a list of recommended categories.
- B.1.1.4. It should have a list of recommended problem set.
- B.1.1.5. It should have a list of upcoming/ongoing virtual exams.

#### B.1.2. Posts

- B.1.2.1. Any user can create a new post, which must be public.
- B.1.2.2. A post can be put into a category.
- B.1.2.3. A post can be tagged up to 5 tags.
- B.1.2.4. A post can be deleted by a post creator or a system administrator.
- B.1.2.5. A post should be commentable using comment mechanism described in B.1.3.
- B.1.2.6. A post should be votable using a voting mechanism described in B.1.5.
- B.1.2.7. A post should be flaggable using reporting mechanism described in B.1.10.
- B.1.2.8. A post should be bookmarkable using bookmark mechanism described in B.1.6.
- B.1.2.9. A post can be divided into 2 types:
  - B.1.2.9.1. Problem
    - B.1.2.9.1.1. Any user can create a problem which requires a title, a content, a level, answer choices (with correct answer marked), and a comprehensive official solution.
    - B.1.2.9.1.2. A problem creator can edit a content, an answer choices' content, a correct choice, and an official solution of a problem using a content editor.
    - B.1.2.9.1.3. A problem creator can also edit other data of a problem, such as problem's title and level.
    - B.1.2.9.1.4. All edits to a problem should be logged into a database and can be displayed as an edit history on the problem post.
    - B.1.2.9.1.5. A problem should be initially shown with a title, a content, a level and answer choices, whereas a correct answer choice together with a comprehensive official solution should be hidden until conditions according to B.1.2.9.1.6 are satisfied.

B.1.2.9.1.6. A correct answer choice and a comprehensive official solution will be shown after:

B.1.2.9.1.6.1. A user checks one of answer choices.

B.1.2.9.1.6.2. A user intentionally chooses to view a solution.

**B.1.2.9.2. Discussion Thread**

B.1.2.9.2.1. A user can create a discussion thread which requires a title and a content.

B.1.2.9.2.2. A discussion thread creator can edit a content using a content editor.

B.1.2.9.2.3. A discussion thread creator can also edit a title.

B.1.2.9.2.4. All edits to a discussion thread should be logged into a database and can be displayed as an edit history on the discussion thread post.

**B.1.3. Comments**

B.1.3.1. Commentable item consists of posts including problems and discussion threads.

B.1.3.2. Any user can reply to a post by commenting to suggest, propose an idea or provide an alternative or more understandable solution.

B.1.3.3. Any user can also reply to an existing comment by sub-commenting.

B.1.3.4. The system provides at most two levels of commenting hierarchy (comment and sub-comment).

B.1.3.5. A comment and sub-comment can be edited by a comment/sub-comment owner only.

B.1.3.6. A comment and sub-comment can be deleted by a comment/sub-comment owner or a system administrator.

B.1.3.7. All edits to a comment should be logged into a database and can be displayed as an edit history on the comment.

B.1.3.8. A content of a comment and sub-comment should be created and edited using a content editor.

B.1.3.9. A long comment and sub-comment should be truncated with Show More button.

B.1.3.10. A comment and sub-comment should be votable and flaggable.

B.1.3.11. All comments should be ranked based on their voting score.

**B.1.4. Problem Set**

B.1.4.1. Any user should be able to create a problem set by specifying a title, description, recommended duration, visibility (can be either public or private), and problems. A problem set must contain at least one problem.

B.1.4.2. A problem set owner should be able to edit any data that can be set upon creation.

B.1.4.3. A problem set owner should be able to delete a problem set.

B.1.4.4. Only user that met one of these criteria should be able to see the problem set:

B.1.4.4.1. A problem set is public.

B.1.4.4.2. A user is the problem set creator himself.

B.1.4.4.3. A user has the direct link to a problem set.

B.1.4.5. A problem in a problem set can be added by:

B.1.4.5.1. Creating a new problem.

B.1.4.5.2. Choosing from an existing problem in the system.

B.1.4.6. A user who met one of the criteria in B.1.4.4 can see a problem set overview, which contains a title, description, recommended duration, visibility, and a list of problems (which is represented by a problem's title). It should also have a difficulty summary, which is a count of difficulty in all problems in a problem set.

B.1.4.7. A problem set should be bookmarkable using bookmark mechanism described in B.1.6.

B.1.4.8. A problem set should be flaggable using flag mechanism described in B.1.10.

B.1.4.9. A registered user can start practicing on a problem set, where user can do an exercise.

B.1.4.9.1. User can also choose whether to include a timer or not, whose duration is initially set to a recommended duration.

B.1.4.9.2. User can see a summary result after finishing practice session. The summary should include a score, as well as a correct answer, official solution, and time usage in each problem.

#### **B.1.5. Voting**

B.1.5.1. Votable item consists of problems, discussion threads, comments, and sub-comments.

B.1.5.2. Votable item can be upvoted by any users except its owner.

#### **B.1.6. Bookmarking**

B.1.6.1. Bookmarkable item can be a post, a problem set, or an exam.

B.1.6.2. User can bookmark a bookmarkable item.

B.1.6.3. User can remove a bookmark.

B.1.6.4. User can see all bookmarks in a bookmark page.

#### **B.1.7. Virtual Exam**

B.1.7.1. An organization can access the virtual exam management system (VEMS).

B.1.7.1.1. VEMS should display an exam creation quotas: private virtual exam quota and public virtual exam quota. Both quotas are given by a system administrator.

B.1.7.1.2. An organization can host an exam by setting up a title, description, duration, visibility, registration period, and an exam problems.

B.1.7.1.3. An exam's visibility can be either public or private.

B.1.7.1.4. An organization can opt whether to publish an exam after it has ended or not.

B.1.7.1.5. All problems in a virtual exam problem should only be brand new problems specific for that exam.

B.1.7.1.6. After any exams, an organization can see a detailed virtual exam result which consists of:

B.1.7.1.6.1. Number of participants.

B.1.7.1.6.2. Statistical data of scores including mean, median, mode, maximum, minimum, total participants, score distribution, rank, and user's percentile.

B.1.7.1.6.3. A selected answer choice distribution of each problem.

B.1.7.2. Any users can participate a virtual exam.

B.1.7.2.1. A user can only see and interact with a virtual exam when one of these criteria are met:

B.1.7.2.1.1. A virtual exam is public.

B.1.7.2.1.2. A user has a direct link to a virtual exam.

B.1.7.2.2. A virtual exam is bookmarkable.

B.1.7.2.3. A user can register for a virtual within registration period.

B.1.7.2.4. A user registered with a virtual exam can join after the start time has passed.

B.1.7.2.5. During a virtual exam, a virtual exam participant can see all the virtual exam's problems along with their answer choices. A user can choose to see and do a problem in any order.

B.1.7.2.6. During a virtual exam, a virtual exam participant can check the answer of a virtual exam's problem as correct. The check status can be changed as many times as the

user want throughout a virtual exam period and should be saved on the server instantly or periodically.

B.1.7.2.7. After exam, all users can see a virtual exam result, which consists of:

B.1.7.2.7.1. Statistical data of scores including mean, median, mode, maximum, minimum, total participants, score distribution, rank, and user's percentile.

B.1.7.2.7.2. Editorial including an official solution to each problem in an exam.

#### **B.1.8. Search**

B.1.8.1. A user can search problems and discussion threads by a category or a tag using a plain text.

B.1.8.2. Search result should be sorted and filtered by customizable configurations and constraints.

#### **B.1.9. Content Editor**

B.1.9.1. It should be What-You-See-Is-What-You-Get (WYSIWYG) editor.

B.1.9.2. It should be able to edit regular text.

B.1.9.3. It should be able to set typeface to be bold, italic, and underline.

B.1.9.4. It should be able to set the bullet, numbering, link, and heading.

B.1.9.5. It should be able to add, edit, and delete an image.

B.1.9.6. It should be able to add, edit, and delete an equation using WYSIWYG equation editor.

#### **B.1.10. Flag and Review**

B.1.10.1. Flaggable item consists of discussion threads, problems, comments, and sub-comments.

B.1.10.2. A user can flag a flaggable item as an inappropriate one and can also provide a reason why it should be banned to facilitate reviewing process by a system administrator.

B.1.10.3. A system administrator can review a flag whether it is false-negative or not.

#### **B.1.11. Login and Register**

B.1.11.1. A user can register an account to the system by one of the following methods:

B.1.11.1.1. Register with an email address.

B.1.11.1.1.1. User must confirm his/her email by clicking on the confirmation link sent to his/her email inbox to activate the account before using any account-related functions.

B.1.11.1.2. Register with Facebook. An email address is acquired from Facebook email account.

B.1.11.2. If a user registers with his email address, he can connect his account with Facebook later.

B.1.11.3. A user can login to the system by one of the following methods:

B.1.11.3.1. Login with an email address.

B.1.11.3.2. Login with Facebook. Connecting an account with Facebook beforehand is required.

B.1.11.4. The system should provide a resetting password mechanism in case a user cannot access his own account.

#### **B.1.12. User Profile**

B.1.12.1. A user can update his personal information: full name, gender, birthdate, occupation, bio, profile picture, display name, email, and password.

B.1.12.2. A user can view past exams' result.

B.1.12.3. A user can see a list of bookmarks.

### **B.2. Nonfunctional Requirement**

#### **B.2.1. Operational Requirements**

B.2.1.1. The system should operate as a web application in the following browsers:

B.2.1.1.1. Chrome: latest version

B.2.1.1.2. Firefox: latest version

B.2.1.1.3. Edge: 13 and all higher versions

B.2.1.1.4. IE: 9 and all higher versions

B.2.1.1.5. Safari: 7 and all higher versions

B.2.1.2. The system should provide an understandable and easy-to-interact interface for target users.

B.2.1.3. It should not be possible to delete a problem data entirely from the system.

#### **B.2.2. Performance Requirements**

B.2.2.1. The system should handle at least 100 concurrent requests per second.

B.2.2.2. The system average response time should be lower than 4 seconds.

#### **B.2.3. Security Requirements**

B.2.3.1. Only registered user can access to problem's solution.

B.2.3.2. Only system administrators can review flags.

B.2.3.3. The private user data must not be accessed without a permission.

#### **B.2.4. Cultural and Political Requirements**

B.2.4.1. All problems in the system should not violate the copyright policy.

B.2.4.2. Copying of other users' problem is not allowed in any case.