

# **A Cloud-based Software System for Online Multimedia Examinations**

Congwu Tao

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Master of Science  
in  
Computer Science and Applications

Osman Balci, Chair  
James D. Arthur  
Anderson H. Norton III

February 15, 2016

Blacksburg, Virginia

*Keywords and phrases:* Cloud-based software, software engineering, client-server architecture, multimedia examination, examination system, online examination

# A Cloud-based Software System for Online Multimedia Examinations

Congwu Tao

## ABSTRACT

With the advancement in information technology, online assessments are getting more attention and online examinations are regarded as important parts of online learning. Online examinations can be easily taken by remote students, help the students get exam results quickly and save their time; online examinations also aid instructors in collecting students' exam answers and generating the exam reports effectively. In addition, online examinations can help reduce cost and save trees for our world.

Multimedia elements like images, graphics, video and audio have been widely integrated into online learning environments. They not only help instructors design more engaging online learning content, but also help provide more interactive and pleasant learning experience for learners. However, integrating multimedia elements into online examination systems is rarely reported. Multimedia elements generally consume amounts of computing resources in a separated software system running on a single computer. "Software-as-a-Service (SaaS)" has become a new software paradigm and cloud-based software systems are becoming more attractive due to their dynamic scalability and effective usage of computing resources. Yet, how to effectively integrate multimedia elements into a cloud-based software system for online examinations is not significantly investigated. Although a variety of online-assessment tools have been developed, few of them adopt the "Software-as-a-Service (SaaS)" paradigm and most of them focus on the assessment in a specific domain or an application area with short of multimedia elements. There is a lack of a comprehensive software solution for online multimedia examinations.

This thesis tries to utilize the "Software-as-a-Service (SaaS)" paradigm, design and develop a cloud-based software system for online multimedia examination (ARENAS), and explore a comprehensive software solution for the online assessment field. ARENAS employs a multi-tiered client-server architecture and includes five subsystem modules: user module, question repository module, exam module, exam report module and configuration module. The developed cloud-based software system can present online questions with multimedia elements, and also support a myriad of question types, flexible accounts to the exam-takers, randomized question order in an online exam, flexible grading mechanisms, and analytical exam reports. For instructors, the developed system can help design more engaging online questions; for exam-takers, the developed system can help provide more user-friendly experience; for other educators and researchers, the design and development processes of ARENAS can be taken as a reference to designing and developing other large-scale cloud-based educational software systems.

# **A Cloud-based Software System for Online Multimedia Examinations**

Congwu Tao

## **GENERAL AUDIENCE ABSTRACT**

Online examinations can be easily taken by remote students, help the students get exam results quickly and save their time; online examinations also aid instructors in collecting students' exam answers and generating the exam reports effectively. In addition, online examinations can help reduce cost and save trees for our world.

Multimedia elements like images, graphics, video and audio have been widely integrated into online learning environments. They not only help instructors design more engaging online learning content, but also help provide more interactive and pleasant learning experience for learners. However, integrating multimedia elements into online examination systems is rarely reported. It is clear that few studies investigated online multimedia examination systems based on a review of the literature and there is a lack of a comprehensive software solution for online multimedia examinations. This thesis focused on designing and developing a software system (ARENAS) for multimedia examinations.

The developed system (ARENAS) can help instructors effectively use multimedia elements like images, graphics, video and audio to design more engaging online assessment content; ARENAS can also help students get more interactive and pleasurable online learning and assessment experience. In addition, the design and development processes of ARENAS can be taken as a reference to designing and developing other educational software systems.

## **ACKNOWLEDGMENTS**

First and foremost, I would like to thank my advisor Dr. Osman Balci for his professional and invaluable guidance. Without his help, I could not have completed my study at Computer Science Department, Virginia Tech. Dr. Balci risked being my adviser, even though I had double majors at VT and worked full-time for the past half year. Thank you, Dr. Balci. I cannot find any other words to express my gratitude.

I am also very grateful to my committee members Dr. James D. Arthur and Dr. Anderson Norton for their professional guidance in my thesis.

Lastly, I am deeply thankful to my family members: Weixiao Tao, Dongxiang Cai, Chunjiao Tao, Xiaoming Liu and Xiaoyuan Zhao. Thank you for your love, support, encouragement, understanding and sacrifices. Your love and encouragement have been the most valuable thing in my life.

# TABLE OF CONTENTS

<b>ABSTRACT</b> .....	<b>ii</b>
<b>GENERAL AUDIENCE ABSTRACT</b> .....	<b>iii</b>
<b>ACKNOWLEDGMENTS</b> .....	<b>iv</b>
<b>LIST OF FIGURES</b> .....	<b>vii</b>
<b>LIST OF TABLES</b> .....	<b>viii</b>
<b>LIST OF ACRONYMS</b> .....	<b>ix</b>
<b>CHAPTER 1:Introduction</b> .....	<b>1</b>
1.1 <b>RELATED WORK</b> .....	1
1.2 <b>STATEMENT OF THE PROBLEM</b> .....	3
1.3 <b>STATEMENT OF OBJECTIVES</b> .....	3
1.4 <b>OVERVIEW OF THESIS</b> .....	3
<b>CHAPTER 2: ARENAS Requirements Specification</b> .....	<b>4</b>
2.1 <b>USE CASE DIAGRAMS</b> .....	4
2.2 <b>USE CASES AND FUNCTIONAL REQUIREMENTS</b> .....	14
2.2.1 <i>Use Case 1: Create an Account</i> .....	14
2.2.1.1    Use Case 1 Documentation .....	14
2.2.1.2    Functional Requirements Associated with Use Case 1 .....	15
2.2.2 <i>Use Case 2: Create a Question Repository</i> .....	15
2.2.2.1    Use Case 2 Documentation .....	15
2.2.2.2    Functional Requirements Associated with Use Case 2 .....	17
2.2.3 <i>Use Case 3: Create an Exam</i> .....	18
2.2.3.1    Use Case 3 Documentation .....	18
2.2.3.2    Functional Requirements Associated with Use Case 3 .....	19
2.2.4 <i>Use Case 4: Publish an Exam</i> .....	19
2.2.4.1    Use Case 4 Documentation .....	19
2.2.4.2    Functional Requirements Associated with Use Case 4 .....	20
2.2.5 <i>Use Case 5: Generate an Exam Report</i> .....	21
2.2.5.1    Use Case 5 Documentation .....	21
2.2.5.2    Functional Requirements Associated with Use Case 5 .....	22
2.2.6 <i>Use Case 6: Take an Exam</i> .....	22
2.2.6.1    Use Case 6 Documentation .....	22
2.2.6.2    Functional Requirements Associated with Use Case 6 .....	23
2.2.7 <i>Use Case 7: View Exam Report</i> .....	24
2.2.7.1    Use Case 7 Documentation .....	24
2.2.7.2    Functional Requirements Associated with Use Case 7 .....	25
2.2.8 <i>Use Case 8: Administrator Dashboard</i> .....	25
2.2.8.1    Use Case 8 Documentation .....	25
2.2.8.2    Functional Requirements Associated with Use Case 8 .....	26
2.3 <b>NON-FUNCTIONAL REQUIREMENTS</b> .....	27
<b>CHAPTER 3: ARENAS Architecture Specification</b> .....	<b>28</b>
3.1 <b>CLIENT-SERVER ARCHITECTURE</b> .....	28
3.2 <b>PEER-TO-PEER ARCHITECTURE</b> .....	29

3.3	SERVICE-ORIENTED ARCHITECTURE .....	30
3.4	JAVA EE-BASED CLIENT-SERVER ARCHITECTURE .....	31
3.5	ARENAS ARCHITECTURE MODEL BASED ON DODAF .....	32
3.5.1	<i>OV-2: Operational Resource Flow Description</i> .....	32
3.5.2	<i>SV-4: Systems Functionality Description</i> .....	33
<b>CHAPTER 4: ARENAS Design Specification .....</b>		<b>35</b>
4.1	CLASS DIAGRAMS .....	35
4.2	PACKAGE DIAGRAMS.....	36
4.3	COMPONENT DIAGRAMS .....	37
4.4	STATECHART DIAGRAMS.....	38
4.4.1	<i>Instructor Statechart Diagram</i> .....	38
4.4.2	<i>Student Statechart Diagram</i> .....	39
4.5	ACTIVITY DIAGRAMS.....	40
4.6	SEQUENCE DIAGRAMS .....	41
4.6.1	<i>Exam Creation Sequence Diagram</i> .....	41
4.6.2	<i>Exam Taking Sequence Diagram</i> .....	41
4.7	COLLABORATION DIAGRAMS.....	42
4.7.1	<i>Exam Creation Collaboration Diagram</i> .....	42
4.7.2	<i>Exam Taking Collaboration Diagram</i> .....	43
4.8	DEPLOYMENT DIAGRAMS.....	43
<b>CHAPTER 5: ARENAS Functionality .....</b>		<b>45</b>
5.1	USER REGISTRATION .....	45
5.2	QUESTION REPOSITORY .....	45
5.3	QUESTION MANAGEMENT .....	46
5.4	EXAM CREATION.....	47
5.5	EXAM QUESTION MANAGEMENT .....	48
5.6	PUBLISH EXAM .....	49
5.7	REPORT EXAM .....	51
5.8	TAKE EXAM.....	52
5.9	VIEW EXAM REPORT .....	53
5.10	ADMINISTRATOR DASHBOARD.....	54
<b>CHAPTER 6: Self-Evaluation of ARENAS.....</b>		<b>55</b>
6.1	FUNCTIONALITY .....	55
6.2	USABILITY.....	55
6.3	SCALABILITY .....	55
6.4	RELIABILITY .....	56
6.5	PERFORMANCE.....	56
6.6	REUSABILITY .....	56
6.7	SUPPORTABILITY.....	57
<b>CHAPTER 7: Conclusions and Future Research .....</b>		<b>58</b>
7.1	CONCLUSIONS.....	58
7.2	CONTRIBUTIONS.....	58
7.3	FUTURE DEVELOPMENT AND RESEARCH.....	58
<b>REFERENCES .....</b>		<b>60</b>

## LIST OF FIGURES

FIGURE 1. ARENAS USE CASE DIAGRAM.....	5
FIGURE 2. ACCOUNT USE CASE DIAGRAM.....	6
FIGURE 3. QUESTION REPOSITORY USE CASE DIAGRAM.....	7
FIGURE 4. EXAM CREATION USE CASE DIAGRAM.....	8
FIGURE 5. EXAM PUBLISHING USE CASE DIAGRAM.....	9
FIGURE 6. EXAM REPORT GENERATION USE CASE DIAGRAM.....	10
FIGURE 7. TAKING AN EXAM USE CASE DIAGRAM.....	11
FIGURE 8. VIEWING EXAM REPORT USE CASE DIAGRAM.....	12
FIGURE 9. ADMINISTRATOR DASHBOARD USE CASE DIAGRAM.....	13
FIGURE 10. A TWO-TIER CLIENT-SERVER ARCHITECTURE [PRESSMAN 2010].....	28
FIGURE 11. A MULTI-TIERED CLIENT-SERVER ARCHITECTURE [PRESSMAN 2010].....	29
FIGURE 12. A PEER-TO-PEER ARCHITECTURE [KWOK 2012].....	30
FIGURE 13. A SERVICE-ORIENTED ARCHITECTURE [IBM 2007].....	31
FIGURE 14. A JAVA EE-BASED CLIENT-SERVER ARCHITECTURE [BALCI 2015].....	32
FIGURE 15. OV-2: OPERATIONAL RESOURCE FLOW DESCRIPTION.....	33
FIGURE 16. SV-4: SYSTEMS FUNCTIONALITY DESCRIPTION.....	34
FIGURE 17. ARENAS CLASS DIAGRAM.....	36
FIGURE 18. USER-EXAM-DATABASE PACKAGE DIAGRAM.....	37
FIGURE 19. EXAM-QUESTION COMPONENT DIAGRAM.....	38
FIGURE 20. INSTRUCTOR STATCHART DIAGRAM.....	39
FIGURE 21. STUDENT STATCHART DIAGRAM.....	39
FIGURE 22. INSTRUCTOR AND STUDENT ACTIVITY DIAGRAM.....	40
FIGURE 23. EXAM CREATION SEQUENCE DIAGRAM.....	41
FIGURE 24. EXAM TAKING SEQUENCE DIAGRAM.....	42
FIGURE 25. EXAM CREATION COLLABORATION DIAGRAM.....	42
FIGURE 26. EXAM TAKING COLLABORATION DIAGRAM.....	43
FIGURE 27. ARENAS DEPLOYMENT DIAGRAM.....	44
FIGURE 28. USER REGISTRATION INTERFACE.....	45
FIGURE 29. QUESTION REPOSITORY INTERFACE.....	46
FIGURE 30. QUESTION MANAGEMENT INTERFACE.....	47
FIGURE 31. EXAM CREATION INTERFACE.....	48
FIGURE 32. EXAM QUESTION MANAGEMENT INTERFACE.....	49
FIGURE 33. PUBLISH EXAM INTERFACE.....	50
FIGURE 34. REPORT EXAM INTERFACE.....	51
FIGURE 35. TAKE EXAM INTERFACE.....	52
FIGURE 36. VIEW EXAM REPORT INTERFACE.....	53
FIGURE 37. ADMINISTRATOR DASHBOARD INTERFACE.....	54

## LIST OF TABLES

TABLE 1. USE CASE 1 DOCUMENTATION .....	14
TABLE 2. USE CASE 2 DOCUMENTATION .....	15
TABLE 3. USE CASE 3 DOCUMENTATION .....	18
TABLE 4. USE CASE 4 DOCUMENTATION .....	19
TABLE 5. USE CASE 5 DOCUMENTATION .....	21
TABLE 6. USE CASE 6 DOCUMENTATION .....	22
TABLE 7. USE CASE 7 DOCUMENTATION .....	24
TABLE 8. USE CASE 8 DOCUMENTATION .....	25

## LIST OF ACRONYMS

API	Application Programming Interface
ARENAS	A Cloud-Based Software System for Online Multimedia Examinations
CSA	Client-Server Architecture
CSS	Cascading Style Sheets
DCOM	Distributed Component Object Model
DOM	Document Object Model
EJB	Enterprise JavaBean
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JDK	JAVA Development Kit
JMS	Java Message Service
JSF	JavaServer Faces
JSON	JavaScript Object Notation
JSP	JavaServer Pages
JRE	JAVA Runtime Environment
P2P	Peer-to-Peer
RDBMS	Relational Database Management System
RMI	Remote Method Invocation
SDK	Software Development Kit
SOA	Service-Oriented Architecture
SQL	Structured Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
XHTML	Extensible Hypertext Markup Language
XML	eXtensible Markup Language

# CHAPTER 1: Introduction

An examination is an assessment of someone's knowledge, skill, or aptitude in a particular subject matter [Ikweze 2014]. In the educational field, examinations play a significant role in helping instructors understand to what extent their students grasp the knowledge taught, so as to help the instructors tweak their teaching and further improve their students' learning.

Paper-based examinations are inconvenient for instructors to implement for online courses and grading paper-based exams by hand is time-consuming and inefficient. Online examinations can be conducted for remote students and help these students get exam results quickly, so as to save their time from long waiting which is the normal thing in traditional paper-based exams; online examinations are also beneficial for instructors to collect students' exam answers easily, as well as are helpful for reducing cost and saving trees for our world.

However, in online examinations, there exist some challenges to academic integrity. It is hard to invigilate an online exam and stop cheating in the environment of online examinations [Rodchua, Yiadom-Boakye and Woolsey 2011]. Some institutions ask proctors to supervise online exams. Yet, proctoring in online exams can be costly. For instance, the costs may include the salaries of the people who administrate the test center, the wages for the proctors, the time to find a qualified proctor and the efforts of scheduling an exam in the test center [Cluskey, Ehlen and Raiborn 2011]. In addition, proctoring in online exams is not practical for all remote students. Therefore, anti-cheating is a crucial aspect that needs to be considered carefully when implementing an online examination system.

Multimedia elements like images, graphics, video and audio have been widely integrated into online learning environments and these multimedia elements have helped provide more interactive and enjoyable learning experience [Wang 2008]. Online assessment is a part of online learning experience and online examinations can be more interactive and pleasurable when incorporating with multimedia elements [Hao 2010]. In fact, online assessment is inseparable from online teaching and learning. Hence, creating online examinations that align instructions and also improve students' learning experience should be an important goal for educators and researchers.

However, based on a review of the literature, it is clear that few studies investigated online multimedia examination systems. Most of the existing online examination systems focus on copying the styles of paper-based exams and it is hard to improve the user experience when using those systems.

The software paradigm has been shifting during the last decade from "Software-as-a-Product (SaaS)" to "Software-as-a-Service (SaaS)" provided "in the cloud" [Balci 2014]. Cloud computing is becoming a more attractive technology due to its dynamic scalability and effective usage of resources, yet how to use cloud computing technology to develop an online examination system is not significantly reported.

This thesis tries to integrate multimedia elements and cloud computing technology into online examinations, design and develop a cloud-based software system for online multimedia examinations (ASENRS), and make a further research on exam systems in the online assessment field.

## 1.1 Related Work

Several online examination systems have been designed and developed. For example, the first version of WebAssign was developed at North Carolina State University in 1998 for online assignment and test. The latest version of WebAssign has become a commercial online instructional system for helping instructors

deploy assignments and instantly assess individual student performance. [McGough et al. \[2001\]](#) proposed a web-based test system with special support for advanced mathematics typesetting and equation parsing. [Yuan et al. \[2003\]](#) developed a web-based online examination system with automatic grading for objective questions. This system was successfully deployed to assess the basic knowledge in computer science courses for the students in the local universities. [Sung et al. \[2005\]](#) designed and developed a web-based self- and peer-assessment system for facilitating instructors to arrange various self- and peer-assessment procedures. [He \[2006\]](#) developed a web-based educational assessment system to evaluate students' learning outcomes after they had learned curriculum content. The performance of this web-based system was encouraging when used in science and mathematics courses at two local high schools. [Emary and Al Sondos \[2006\]](#) developed an online system for tutoring and e-examination of economics courses; this system included two major modules: the first module was an online website which was used to review all related materials of economics courses; the second one was an online examination with a large question bank. [Hua, Shu and Bian \[2008\]](#) proposed an online training and exam system based on web-service technologies, but the researchers did not discuss the question types. [Hlaing \[2009\]](#) presented a framework to allow students to carry out online assessments through a mobile user agent. [Fluck, Pullen and Harper \[2009\]](#) explored a computer-based examination system with open source live CD based on Ubuntu in a case study.

Unfortunately, the above mentioned systems more or less focus on one specific domain through using the online examination system for assessing objective questions. Some researchers also explored developing online examination systems for general assessment purpose. For instance, [Rashad et al. \[2010\]](#) developed an Arabic web-based exam management system using open source technologies such as PHP and MYSQL. This system included automatic grading function, exam administration function and exam report function; it also supported several question types like multiple-choice questions and essay questions. This system was used by more than 250 engineering students in Mansoura University and the evaluation result was encouraging. However, the system utilized limited multimedia elements to present the online questions. [Hang \[2011\]](#) presented a preliminary campus-based online examination system which included instructor module, student module and administrator module, but the researcher did not discuss the detail of question types and the evaluation of the system. [Temitayo, Adebisi and Alice \[2013\]](#) developed a computer-based examination system for Nigerian Universities using Visual Studio 2012. This system was successfully employed by several local institutions, but the system just supported limited question types. [Lu et al. \[2013\]](#) designed and developed an intelligent assessment management system for the summative assessment of college courses, but the system had a hybrid structure for integrating online assessments and in-classroom examinations. [Yağci and Ünal \[2014\]](#) developed an adaptive online examination system based on measuring student competence level using PHP programming language; this system was successfully applied at Ahi Evran University in Turkey, but the question types and multimedia elements in presenting questions were limited.

There are also some commercial online examination systems such as [\[ClassMarker 2015; Exambuilder 2015; ExamSoft 2015\]](#) and so on. Examsoft emphasizes exam delivery and management and it cooperates with institutions rather than provides service to instructors and students directly. ClassMarker is a web-based test system for business and education; it is a quiz maker and allows users to create an online exam with setting time limits, public or private test access, randomization of question order and instant feedback in an exam. Exambuilder provides online test authoring and delivery; it can be easily integrated into learning management systems (LMSs) and helps generate real-time scoring results. However, these commercial online exam systems do not pay much attention to the multimedia elements in an online exam and have limited functionalities for instructors and students to use for free.

Although a variety of online-assessment tools have been developed, they just focus on the assessment in some specific domains or application areas with limited multimedia elements. In addition, few of them employ the "Software-as-a-Service (SaaS)" software paradigm. There is a lack of a cloud-based software solution to support multimedia elements in an online examination with a myriad of question types, flexible

accounts to the exam-takers, randomized order in an online exam, flexible grading mechanisms, and analytical exam reports.

## **1.2 Statement of the Problem**

Multimedia elements have been integrated into online teaching and learning for several years, however, fully integrating multimedia elements into online examinations is not reported much. A lot of existing examination systems present questions with monotonous styles and support limited question types. In addition, few existing online examination systems support flexible accounts to the exam-takers, randomized question order in an online exam, flexible grading mechanisms, and analytical exam reports. Most of the existing online-assessment systems do not utilize the “Software-as-a-Service (SaaS)” paradigm and are short of flexibility, robustness and scalability for a large scale exam-takers who take the exam at the same time.

By utilizing a multi-tiered client-server architecture (CSA) and related cloud computing technologies, we develop a cloud-based software solution for multimedia online examinations (ARENAS). This software solution can help instructors effectively design engaging online questions with multimedia elements and provide much friendly user experience for instructors and students.

## **1.3 Statement of Objectives**

The objectives of the research include the following aspects:

1. Investigating what functionalities are necessary for ARENAS.
2. Exploring what kind of architecture could be the most appropriate for ARENAS.
3. Designing and developing the system for instructors, students and administrators.
4. Enabling instructors to effectively create an engaging online exam with multimedia elements and generate exam reports quickly.
5. Engaging students in taking an online exam and getting the exam results rapidly with friendly user experience anywhere, anytime.
6. Providing comprehensive exam reports to administrators for helping evaluate the teaching and learning process.

## **1.4 Overview of Thesis**

This thesis is organized as follows: Chapter 1 presents the introduction and need of this study; Chapter 2 discusses the AREANS requirements specification; Chapter 3 describes the AREANS architecture specification; Chapter 4 discusses the AREANS design specification; Chapter 5 describes the developed AREANS functionalities; Chapter 6 presents the self-evaluation of AREANS; Chapter 7 states the concluding remarks, contributions and future work.

## CHAPTER 2: ARENAS Requirements Specification

The requirements for a software system are the descriptions of what the system shall do, the services that the system offers and the constraints on its operation [Sommerville 2009]. Requirements are often categorized as functional requirements and non-functional requirements. Functional requirements emphasize the service that a software system should provide, how the system should react to particular inputs and how the system should behave in particular situations. Non-functional requirements highlight the system quality attributes such as usability, performance and so on [Pressman 2010].

Use case-based requirements engineering is considered as a useful practice for creating software requirements specification due to these reasons: 1) a use case states the behavior of a system and describes the sequent actions for a small amount of work that the software system is required to perform; 2) a use case describes an interaction, and based on that description, the real functional requirements can be more successfully identified and associated with that use case; 3) use cases can help decompose a complex software system functionality and enable the modularizations to overcome the complexity; and 4) use cases can help turn the different requirements into classes in an object-oriented design and considerably facilitate such transitions [Balci 2014].

This chapter focuses on ARENA requirements specification through use case-based requirements engineering.

### 2.1 Use Case Diagrams

A use case diagram is a graphic depiction of the interactions among the elements of a software system [Balci 2014]. ARENAS contains nine use case diagrams as follows.

*Figure 1* indicates the overall use case diagram for ARENAS. There are three types of user roles: instructor, student and administrator in ARENAS. For an instructor, there are four major functions: 1) creating a question repository; 2) creating an exam; 3) publishing an exam; and 4) reporting an exam. Every major function contains several sub-functions. For example, an instructor can add, delete, update or view a question in a question repository after he or she has created that question repository; the instructor can also select a question from the question repository to add into an exam; the instructor can assign the exam to a bunch of registering student accounts or non-registering student accounts; and the instructor can review or change a grade in student exam reports. For a student in ARENAS, he or she can take an exam, view the exam report, download the exam report and export the exam report. For an administrator in ARENAS, he or she can manage the user accounts, view statistical exam information and view statistical exam report information.

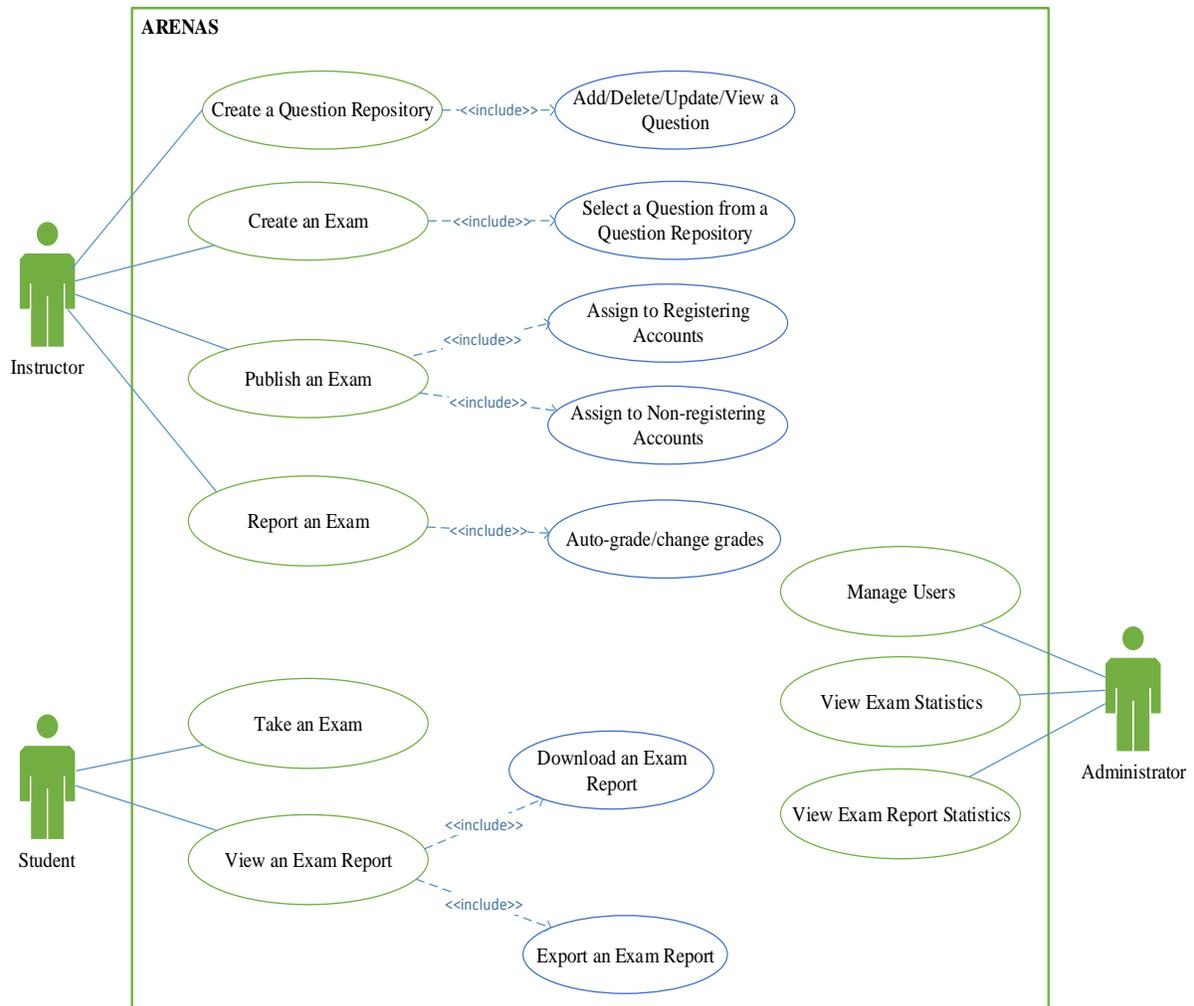


Figure 1. ARENAS Use Case Diagram

Figure 2 displays the account use case. A student or an instructor can create an account, delete an account, update an account, view an account and reset a password for the account.

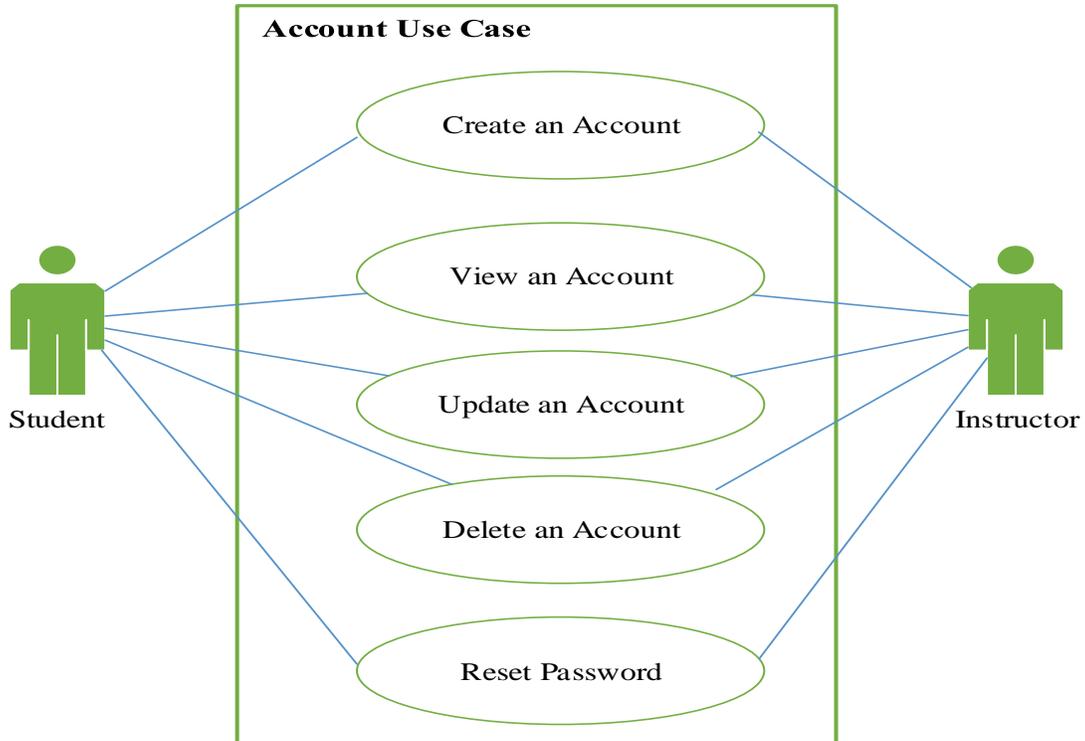


Figure 2. Account Use Case Diagram

Figure 3 displays the question repository use case. An instructor can create a question repository and name it, add a question into the question repository, delete a question from the question repository, update a question in the question repository, and view a question in the question repository. For each question in the question repository, the instructor can set a question type for the question and assign points to the question. When the instructor wants to look for some questions in the question repository, he or she can query the question repository with key words.

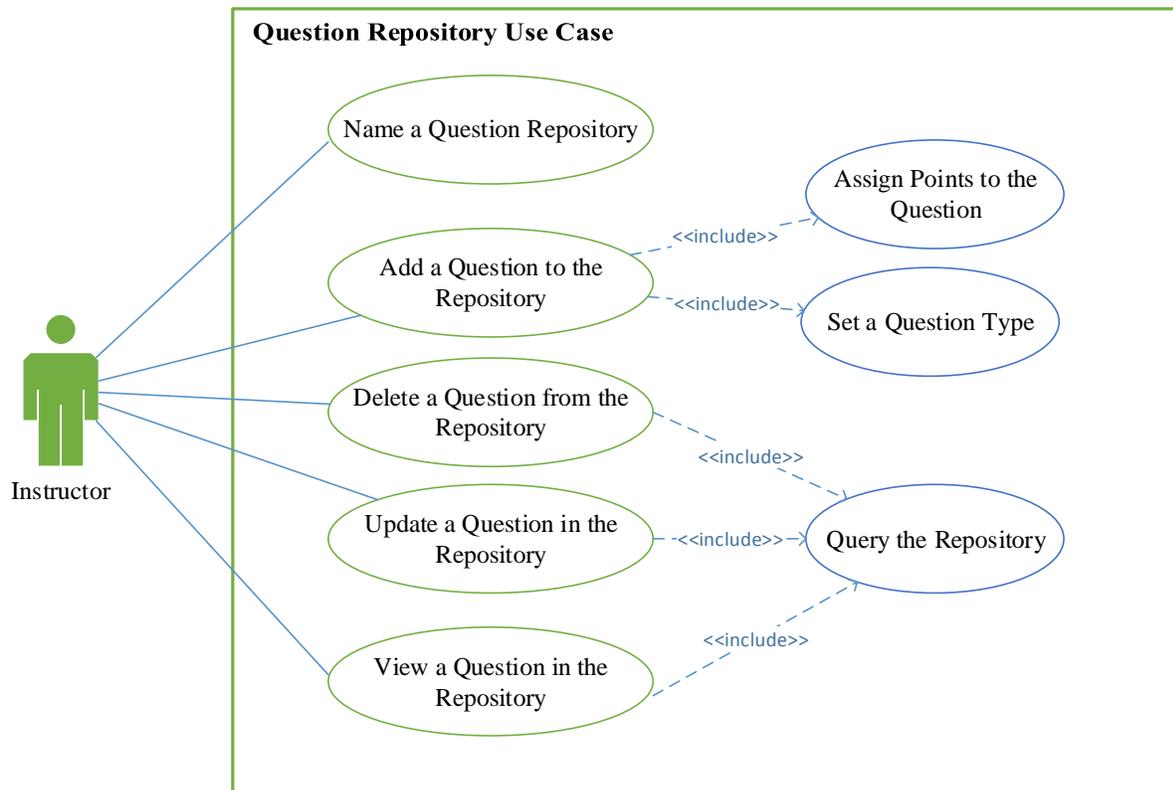


Figure 3. Question Repository Use Case Diagram

*Figure 4* shows the exam creation use case. An instructor can create an exam and name it, add a question into the exam through selecting a question from the question repository which he or she has created earlier, update a question in the exam, delete a question from the exam, view a question in the exam, and change points for each question in the exam.

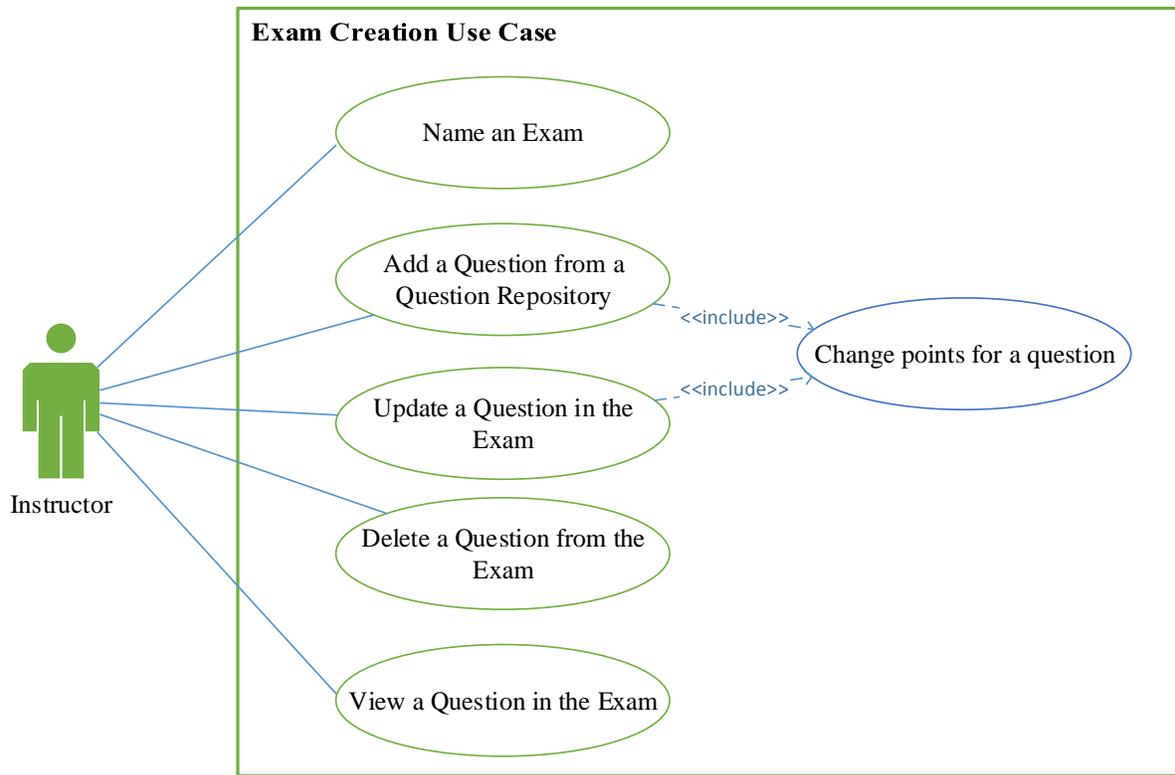


Figure 4. Exam Creation Use Case Diagram

*Figure 5* displays the exam publishing use case. When an instructor wants to publish an exam, he or she can set some attributes for the exam. For example, the instructor can set exam date, randomization question order and immediate feedback for that exam. The instructor can assign the exam to the registering student accounts and email these students. The instructor can also generate a bunch of temporary student accounts and assign the exam to these temporary student accounts.

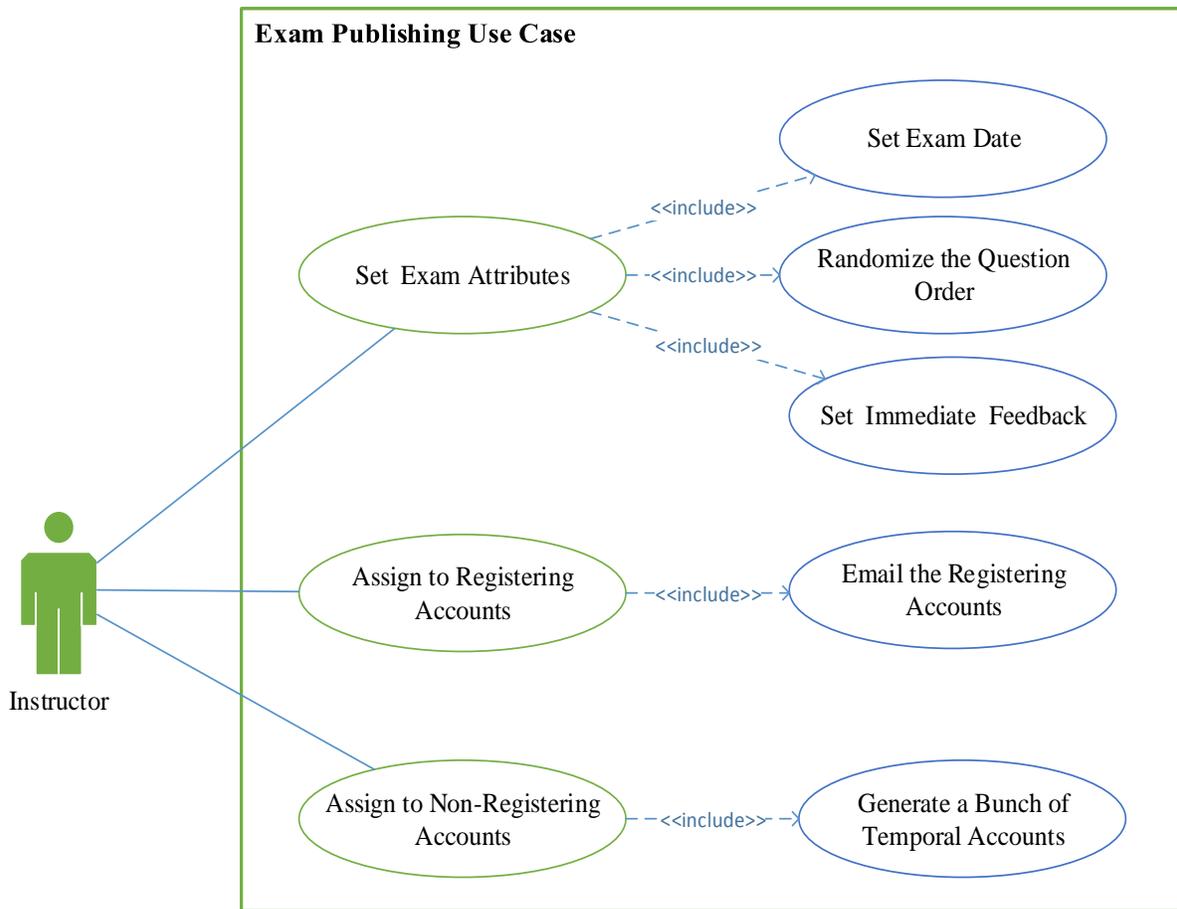


Figure 5. Exam Publishing Use Case Diagram

Figure 6 shows the exam report generation use case. An instructor can check or review an exam report which is auto-graded by the system. For the subjective question like short answer or essay question, the instructor needs to grade it by hand. The instructor can check the statistical information about the student in that exam and write a brief summary in the student exam report. The instructor can also choose to publish the exam report in the system or send the exam report to the students by email.

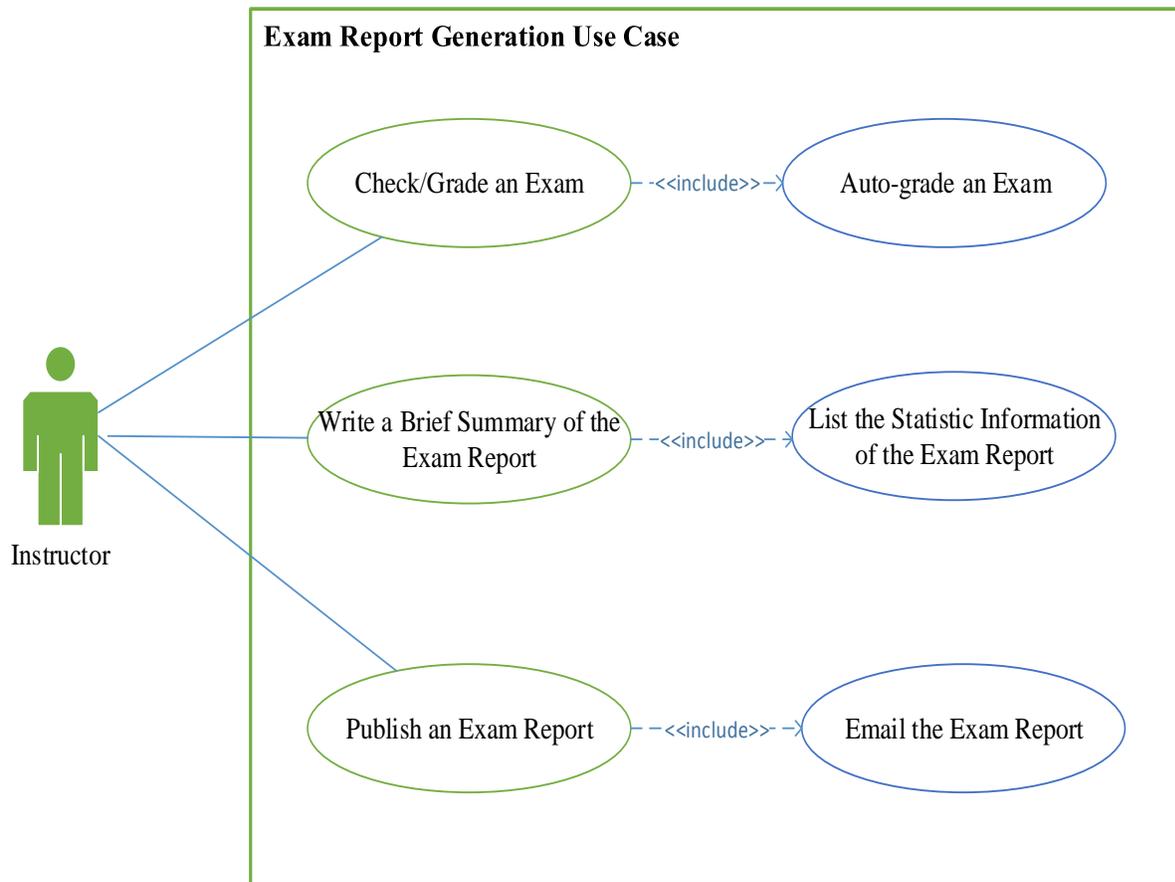


Figure 6. Exam Report Generation Use Case Diagram

*Figure 7* displays the taking an exam use case. A student can resume an incomplete exam which he or she began before, or start a new exam which has been assigned to him or her. The precondition is that the exam period is still valid. A student can also hide or show a timer on the webpage when he or she is taking an exam.

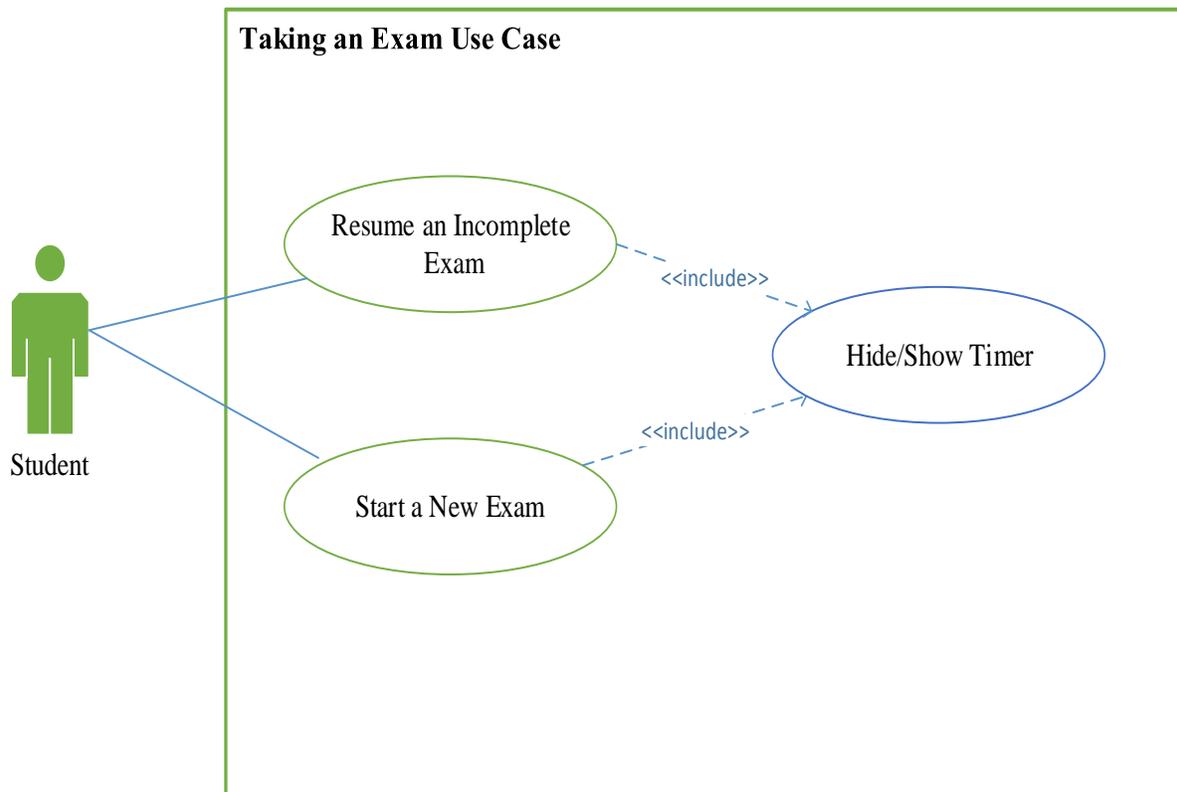


Figure 7. Taking an Exam Use Case Diagram

Figure 8 shows the exam report view use case. After an exam report is published, a student can view the exam report online. The student can also download the exam report or export the exam report.

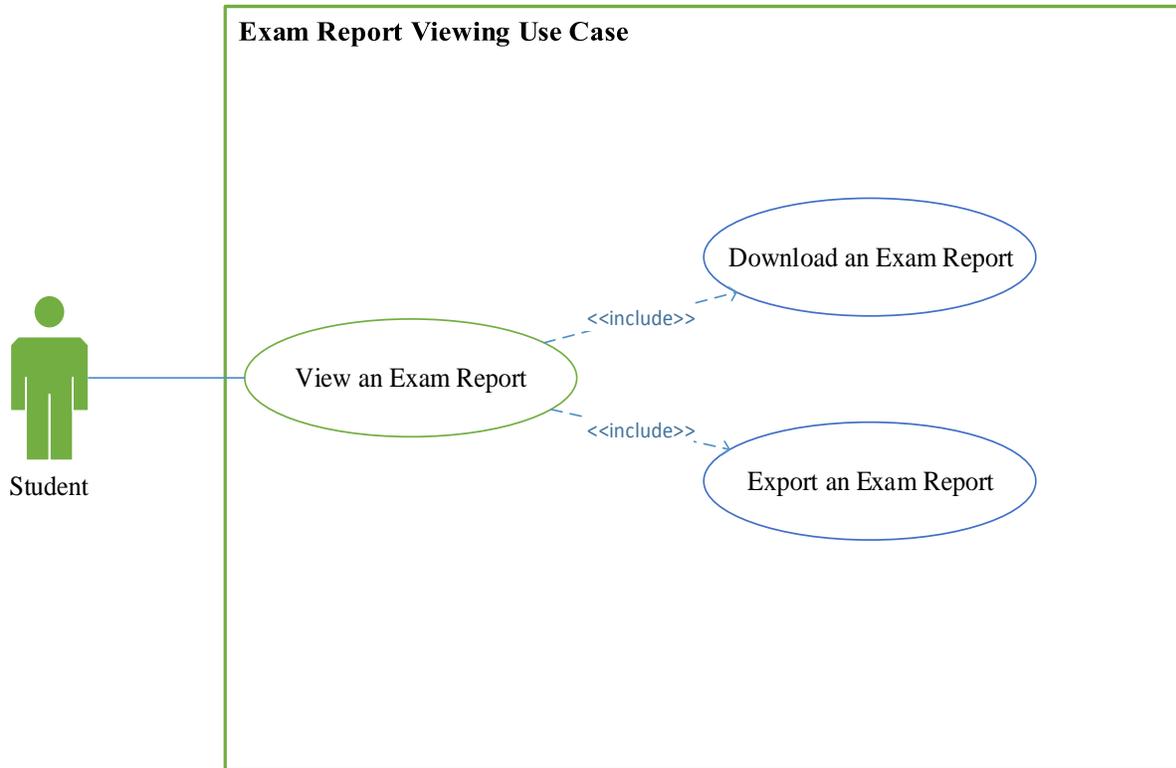


Figure 8. Viewing Exam Report Use Case Diagram

*Figure 9* displays the administrator dashboard use case. An administrator can manage user accounts. For example, an administrator can add, delete, update or view an instructor account or a student account. An administrator can also view statistical exam information or statistical exam report information.

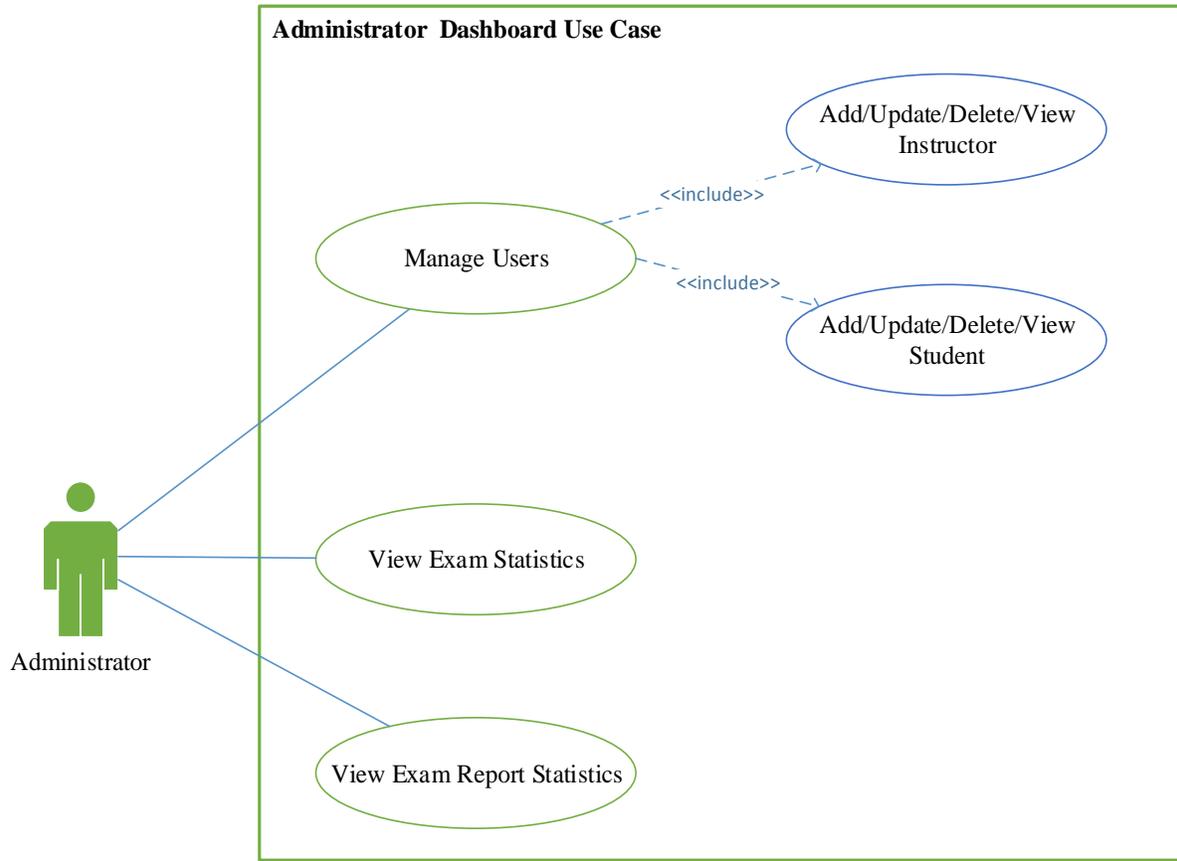


Figure 9. Administrator Dashboard Use Case Diagram

## 2.2 Use Cases and Functional Requirements

This section includes eight use case documentations, which describe the use cases in terms of these aspects: actors, preconditions, flow of events of the primary scenarios, flow of the events of the alternative scenarios, flow of the events of the exception scenarios, extension points and post conditions. For each use case, the associated functional requirements are also identified.

### 2.2.1 Use Case 1: Create an Account

This use case (**Table 1**) describes how an instructor or a student creates an account in ARENAS.

#### 2.2.1.1 Use Case 1 Documentation

Table 1. Use Case 1 Documentation

Use Case 1 Documentation	
<b>Use Case ID:</b>	1
<b>Use Case Name:</b>	<b>Create an Account</b> This use case describes the interaction between a user (instructor or student) and ARENAS about how the user creates an account.
<b>Actors:</b>	
Instructor or Student	
<b>Preconditions:</b>	
The <i>Create an Account</i> interface is displayed	
<b>Flow of Events of the Primary Scenario:</b>	
<ol style="list-style-type: none"> <li>1. The use case starts when the user accesses the <i>Create an Account</i> interface.</li> <li>2. The user inputs a user name, password, confirmed password, email address and major field.</li> <li>3. The user selects the user type: instructor or student.</li> <li>4. The user clicks <b>Create</b>.</li> <li>5. The system verifies the input information:               <ul style="list-style-type: none"> <li>If the user name is not valid, The system will give an alert message.</li> <li>If the user name already exists in the system, The system will give an alert message.</li> <li>If the major field is empty or exceeds 50 characters, The system will give an alert message.</li> <li>If the email address is not valid, The system will give an alert message.</li> <li>If the email address already exists in the system, The system will give an alert message.</li> <li>If the password or the confirmed password is not valid, The system will give an alert message.</li> <li>If the password does not match the confirmed password, The system will give an alert message.</li> </ul> </li> <li>6. The system stores the information into the database.</li> <li>7. The system displays <b>Create Account Successfully</b>.</li> <li>8. The use case ends.</li> </ol>	

<b>Flow of Events of the Alternative Scenarios:</b>
None
<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The interface of <i>Create an Account</i> cannot display.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Create an Account</i> interface.
<b>Postconditions:</b>
The user account is created and the corresponding notification is displayed to the valid user.

### 2.2.1.2 Functional Requirements Associated with Use Case 1

1. ARENAS shall enable the user name to consist of minimum 4 and maximum 8 alphanumeric characters.
2. ARENAS shall enable the major field to consist of minimum 2 and maximum 50 alphanumeric characters.
3. ARENAS shall enable the password or confirmed password to contain minimum 6 and maximum 12 characters of which at least 1 must be uppercase letter, at least 1 must be lowercase letter, at least 1 must be number, and at least 1 must be special character.
4. ARENAS shall enable the user to select a user type: instructor or student.
5. ARENAS shall enable the user to input a valid email address.

### 2.2.2 Use Case 2: Create a Question Repository

This use case (**Table 2**) describes how an instructor creates a question repository in ARENAS.

#### 2.2.2.1 Use Case 2 Documentation

Table 2. Use Case 2 Documentation

Use Case 2 Documentation	
<b>Use Case ID:</b>	2
<b>Use Case Name:</b>	<b>Create a Question Repository</b> This use case describes the interaction between an instructor and ARENAS about how an instructor creates a question repository.
<b>Actors:</b>	
Instructor	
<b>Preconditions:</b>	

- The user has a valid account.
- The user has logged into the system.
- Five question types: multi-choice question, single-choice question, true-false question, matching question, short answer/ essay question, have been predefined and set in the system.

#### Flow of Events of the Primary Scenario:

1. The use case starts when the user accesses the *Create a Question Repository* interface.
2. The user inputs a question repository name and the brief description.
3. The user clicks **Save**.
4. The name of the question repository is created.
5. If the user wants to add a question to the repository:
  - The user selects **Add Question**, then switches into the adding question interface.
  - The user selects the question type from the Question Type List.
    - If the question type selected belongs to objective question ( multi-choice question, single-choice question, true-false question and matching question),
      - The user inputs question title, question content, question answer options, and sets the referenced answers.
    - If the question type selected belongs to subjective question (short answer/essay),
      - The user inputs the question title and question content.
  - If the user wants to add pictures and graphics to the question content,
    - The user selects the corresponding function of the PrimeFaces Editor Widget (<http://www.primefaces.org/>), which will be tweaked and integrated into ARENAS.
  - If the user wants to add audio to the question content,
    - The user uploads a sound file (.mp3, .mp4 or .wav), which is less than 100M.
  - If the user wants to add video to the question content,
    - If the video is less than 100M,
      - The user can upload the video file (.mp4) to the Server.
    - If the video is more than 100M,
      - The user can upload the video to Youtube website (<http://www.youtube.com>).
      - The user inputs the embed address which comes from the Youtube website.
  - The user sets points to the question.
  - The user sets whether to publish the question or not.
  - The user selects **Save**.
6. If the user wants to delete a question from the repository,
  - The user inputs the key words to query the question repository,

<p>The user selects the question which he or she wants to delete, The user clicks <i>Delete</i>.</p> <p>7. If the user wants to update a question in the repository, The user inputs the key words to query the question repository, The user selects the question which he or she wants to update, The user edits the question information, The user clicks <i>Update</i>.</p> <p>8. If the user wants to view a question in the repository, The user inputs the key words to query the question repository, The user selects the question which he or she wants to view, The user clicks <i>View</i>.</p> <p>9. The use case ends.</p>
<b>Flow of Events of the Alternative Scenarios:</b>
None
<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The user cannot login.</li> <li>• The interface of <i>Create a Question Repository</i> cannot display.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Create a Question Repository</i> interface.
<b>Postconditions:</b>
The question repository is created and the corresponding notification is displayed to the valid user.

#### 2.2.2.2 Functional Requirements Associated with Use Case 2

1. ARENAS shall enable the user to name a question repository and add a brief description for that repository.
2. ARENAS shall enable the user to create many question repositories.
3. ARENAS shall list how many questions in each question repository.
4. ARENAS shall provide the question type as an attribute for each question (multi-choice question, single choice question, true-false question, matching question, short answer/essay question).
5. ARENAS shall be able to provide a corresponding interface for the different question types when the user adds that type of question into the question repository.
6. ARENAS shall enable the user to add/ update/ delete/ view a question in the question repository.
7. ARENAS shall enable the user to query the question repository with key words.
8. ARENAS shall be able to tweak and integrate PrimeFaces Editor Widget for the managing multimedia element (pictures, graphics, audio and video) in each question content.
9. ARENAS shall enable the user to upload a sound file (.mp3, .mp4 or .wav) or a video file (.mp4) for question content, which is less than 100M.
10. ARENAS shall enable the user to input an embed address from Youtube for adding a video to question content.

11. ARENAS shall enable the user to set the correct answers to the objective questions (multi-choice question, single choice question, matching question, true-false question).
12. ARENAS shall enable the user to set the assessment rubrics for the subject question (essay/short answer question).
13. ARENAS shall enable the user to set the points for each question.
14. ARENAS shall be able to allow the user to publish the question or not.

### 2.2.3 Use Case 3: Create an Exam

This use case (**Table 3**) describes how an instructor creates an exam in ARENAS.

#### 2.2.3.1 Use Case 3 Documentation

Table 3. Use Case 3 Documentation

Use Case 3 Documentation	
<b>Use Case ID:</b>	3
<b>Use Case Name:</b>	<b>Create an Exam</b> This use case describes the interaction between an instructor and ARENAS about how an instructor creates an exam.
<b>Actors:</b>	
Instructor	
<b>Preconditions:</b>	
<ul style="list-style-type: none"> <li>• The user has a valid account.</li> <li>• The user has logged into the system.</li> <li>• A <i>Question Repository</i> has been created and there exists at least one question.</li> <li>• The <i>Create an Exam</i> interface is displayed.</li> </ul>	
<b>Flow of Events of the Primary Scenario:</b>	
<ol style="list-style-type: none"> <li>1. The use case starts when the user accesses the <i>Create an Exam</i> interface.</li> <li>2. The user inputs exam name, description, beginning time and ending time for creating an exam.</li> <li>3. The user clicks <b>Save</b>.</li> <li>4. An exam is created.</li> <li>5. If the user wants to add a question into the exam, <ul style="list-style-type: none"> <li>The user chooses a <i>Question Repository</i>,</li> <li>If the questions in the <i>Question Repository</i> do not meet the user's needs, <ul style="list-style-type: none"> <li>The user clicks <i>Add Question</i> in the <i>Question Repository</i> to add questions.</li> </ul> </li> <li>If the questions in the <i>Question Repository</i> meet the user's needs, <ul style="list-style-type: none"> <li>The user selects the questions from the question repository to add into the exam.</li> </ul> </li> <li>The user clicks <b>Add</b>.</li> <li>The selected question is added into the exam.</li> </ul> </li> <li>6. If the user wants to set points for a question, <ul style="list-style-type: none"> <li>The user input points.</li> </ul> </li> <li>7. The user clicks <b>Save</b>.</li> <li>8. The submit request is verified by ARENAS and an updated notification displays.</li> <li>9. The questions will be added into the exam and the new exam shows up in the exam list.</li> <li>10. The use case ends.</li> </ol>	

<b>Flow of Events of the Alternative Scenarios:</b>
None
<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The user cannot log into the system.</li> <li>• The <i>Create an Exam</i> interface does not display.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Create an Exam</i> interface.
<b>Postconditions:</b>
The exam is saved in the system and the corresponding notification is displayed to the valid user.

### 2.2.3.2 Functional Requirements Associated with Use Case 3

1. ARENAS shall enable the user to input the exam name, description, beginning time, ending time and save it.
2. ARENAS shall enable the user to reuse the questions that he or she created earlier.
3. ARENAS shall enable the user to select questions from the question repository.
4. ARENAS shall enable the user to set points to each question in the exam.
5. ARENAS shall enable the user to add, view and delete the question in the exam.

### 2.2.4 Use Case 4: Publish an Exam

This use case (**Table 4**) describes how an instructor publishes an exam in ARENAS.

#### 2.2.4.1 Use Case 4 Documentation

Table 4. Use Case 4 Documentation

Use Case 4 Documentation	
<b>Use Case ID:</b>	4
<b>Use Case Name:</b>	<b>Publish an Exam</b> This use case describes the interaction between an instructor and ARENAS about how an instructor publishes an exam.
<b>Actors:</b>	
Instructor	
<b>Preconditions:</b>	

<ul style="list-style-type: none"> <li>• The user has a valid account.</li> <li>• The user has logged into the system.</li> <li>• An exam has already been created.</li> <li>• The interface of <i>Publish an Exam</i> is displayed.</li> </ul>
<b>Flow of Events of the Primary Scenario:</b>
<ol style="list-style-type: none"> <li>1. The use case starts when the user accesses the <i>Publish an Exam</i> interface.</li> <li>2. If the user wants to assign an exam to the existing student accounts, The user inputs the filtering keywords to query, The user selects the student accounts, The user clicks <i>Assign</i>.</li> <li>3. If the user wants to assign the exam to a bunch of temporary accounts, The user inputs the specific number and valid time for the temporary accounts, The user clicks <i>Generate</i>, The specific number of temporary accounts will be generated, The user clicks <i>Assign</i>.</li> <li>4. If the user wants to display the immediate exam report for the exam-takers, The user select the <i>Checkbox of Display Immediate Report</i>.</li> <li>5. The user clicks <i>Submit</i>.</li> <li>6. The submitted request is verified by ARENAS and an updated notification will display.</li> <li>7. The use case ends.</li> </ol>
<b>Flow of Events of the Alternative Scenarios:</b>
None
<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The user cannot login.</li> <li>• The interface of <i>Publish an Exam</i> cannot display.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Publish an Exam</i> interface.
<b>Postconditions:</b>
The corresponding completion notification is displayed to the valid user.

#### 2.2.4.2 Functional Requirements Associated with Use Case 4

1. ARENAS shall enable the user to assign an exam to a bunch of student accounts which are already created in the system.
2. ARENAS shall enable the user to send an email to those student accounts to inform an exam is ready for them.

3. ARENAS shall enable the user to generate a bunch of temporary accounts and assign an exam to these temporary accounts.
4. ARENAS shall enable the user to set whether to display the immediate exam report for the exam-takers or not.

### 2.2.5 Use Case 5: Generate an Exam Report

This use case (**Table 5**) describes how an instructor generates an exam report in ARENAS.

#### 2.2.5.1 Use Case 5 Documentation

Table 5. Use Case 5 Documentation

Use Case 5 Documentation	
<b>Use Case ID:</b>	5
<b>Use Case Name:</b>	<b>Generate an Exam Report</b> This use case describes the interaction between an instructor and ARENAS about how an instructor generates an exam report.
<b>Actors:</b>	
Instructor	
<b>Preconditions:</b>	
<ul style="list-style-type: none"> <li>• The user has a valid account.</li> <li>• The user has logged into the system.</li> <li>• An exam has already been completed by a student.</li> <li>• The interface of <i>Generate an Exam Report</i> is displayed.</li> </ul>	
<b>Flow of Events of the Primary Scenario:</b>	
<ol style="list-style-type: none"> <li>1. The use case starts when the user enters into the <i>Generate an Exam Report</i> interface.</li> <li>2. If there are some subjective questions in the exam, The user scores the subjective questions and input a numerical value, The user clicks <b>Confirm</b>.</li> <li>3. If the user wants to change the points of some questions, The user inputs a numerical value for that question, The user clicks <b>Confirm</b>.</li> <li>4. The user inputs a brief report summary in the exam report field.</li> <li>5. If the user wants to email an exam report to a student, The user inputs the student account, The user clicks <b>Send</b>.</li> <li>6. If the user wants to publish the exam report in the system, The user selects the <i>Checkbox of Publish Exam Report</i>.</li> <li>7. The user clicks <b>Submit</b>.</li> <li>8. The submitted request is verified by ARENAS and an updated notification displays.</li> <li>9. The completion of reporting an exam information displays in the interface.</li> <li>10. The use case ends.</li> </ol>	
<b>Flow of Events of the Alternative Scenarios:</b>	
None	

<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The user cannot login.</li> <li>• The interface of <i>Generate an Exam Report</i> cannot display.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Generate an Exam Report</i> interface.
<b>Postconditions:</b>
The corresponding completion notification is displayed to the valid user.

2.2.5.2 *Functional Requirements Associated with Use Case 5*

1. ARENAS shall enable the user to score the *subjective* questions in the exam.
2. ARENAS shall enable the user to tweak the points for the questions in the exam.
3. ARENAS shall be able to automatically sum up all of the points after an instructor completes the points change.
4. ARENAS shall enable the user to input a brief summary in the exam report.
5. ARENAS shall enable the user to email the exam report to the exam-takers.
6. ARENAS shall enable the user to publish the exam report or not in the system.

2.2.6 *Use Case 6: Take an Exam*

This use case (**Table 6**) describes how a student takes an exam in ARENAS.

2.2.6.1 *Use Case 6 Documentation*

Table 6. Use Case 6 Documentation

Use Case 6 Documentation	
<b>Use Case ID:</b>	6
<b>Use Case Name:</b>	<b>Take an Exam</b> This use case describes the interaction between an instructor and ARENAS about how a student takes an exam.
<b>Actors:</b>	
Student	
<b>Preconditions:</b>	

<ul style="list-style-type: none"> <li>• The user has a valid account.</li> <li>• The user has logged into the system.</li> <li>• An exam has already been assigned to the student.</li> <li>• The exam period is still valid.</li> <li>• The interface of <i>Take an Exam</i> is displayed.</li> </ul>
<b>Flow of Events of the Primary Scenario:</b>
<ol style="list-style-type: none"> <li>1. The use case starts after the user accesses the <i>Take an Exam</i> interface.</li> <li>2. The user reads the Exam Description and confirms understanding the <i>Terms of the Exam</i>.</li> <li>3. If the user takes the same exam which he or she does not complete before, The user can resume the uncompleted exam.</li> <li>4. If the user wants to start a new exam, The user clicks <b>Start</b> to begin the exam.</li> <li>5. If the user wants to hide or show the timer for the exam, The user clicks the <i>Hide or Show of Timer</i>.</li> <li>6. The user answers the questions in the exam.</li> <li>7. The user clicks <b>Submit</b>.</li> <li>8. The submitted request is verified by ARENAS and an updated notification displays.</li> <li>9. The completion of taking an exam information display in the interface.</li> <li>10. The use case ends.</li> </ol>
<b>Flow of Events of the Alternative Scenarios:</b>
None
<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The user cannot login.</li> <li>• The interface of <i>Take an Exam</i> cannot displays.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Take an Exam</i> interface.
<b>Postconditions:</b>
The corresponding notification is displayed to the valid user.

### 2.2.6.2 Functional Requirements Associated with Use Case 6

1. ARENAS shall enable the user to confirm understanding the *Terms of the Exam* and test his or her computer (audio, video, mouse, and monitor) and network connection before the exam.
2. ARENAS shall enable the user to show or hide the timer when he or she is taking an exam.
3. ARENAS shall store the user’s answer to each question into the database.
4. ARENAS shall allow the user to resume his or her uncompleted exams in the valid exam period.
5. ARENAS shall give a 5-minute-left alert in an exam.
6. ARENAS shall close an exam at the ending time.

## 2.2.7 Use Case 7: View Exam Report

This use case (**Table 7**) describes how a student views an exam report in ARENAS.

### 2.2.7.1 Use Case 7 Documentation

Table 7. Use Case 7 Documentation

Use Case 7 Documentation	
<b>Use Case ID:</b>	7
<b>Use Case Name:</b>	<b>View Exam Report</b> This use case describes the interaction between a student and ARENAS about how a student views an exam report.
<b>Actors:</b>	
Student	
<b>Preconditions:</b>	
<ul style="list-style-type: none"> <li>• The user has a valid account.</li> <li>• The user has logged into the system.</li> <li>• An exam report has already been published.</li> <li>• The interface of <i>View Exam Report</i> is displayed.</li> </ul>	
<b>Flow of Events of the Primary Scenario:</b>	
<ol style="list-style-type: none"> <li>1. The use case starts when the user enters into the <i>View Exam Report</i> interface.</li> <li>2. The user clicks <i>View Exam Report</i> to see the detail report.</li> <li>3. If the user wants to download the exam report, The user clicks the <i>Download Exam Report</i>.</li> <li>4. If the user wants to export the exam report, The user clicks the <b>Export</b>, The user selects the file format, The user clicks <b>Confirm</b>.</li> <li>5. The user clicks <b>Close</b>.</li> <li>6. The use case ends.</li> </ol>	
<b>Flow of Events of the Alternative Scenarios:</b>	
None	
<b>Flow of Events of the Exception Scenarios:</b>	
<ul style="list-style-type: none"> <li>• The user cannot login.</li> <li>• The interface of <i>View Exam Report</i> cannot display.</li> <li>• The network connectivity is lost.</li> </ul>	
<b>Extension Points:</b>	
None	
<b>“Used” Use Cases:</b>	
Display the <i>View Exam Report</i> interface.	
<b>Postconditions:</b>	

The corresponding notification is displayed to the valid user.

### 2.2.7.2 Functional Requirements Associated with Use Case 7

1. ARENAS shall enable the user to view the exam report online.
2. ARENAS shall enable the user to download the exam report.
3. ARENAS shall enable the user to export the exam report (PDF).

### 2.2.8 Use Case 8: Administrator Dashboard

This use case (**Table 8**) describes how an administrator manages user accounts and views the statistical information in ARENAS.

#### 2.2.8.1 Use Case 8 Documentation

Table 8. Use Case 8 Documentation

Use Case 8 Documentation	
<b>Use Case ID:</b>	8
<b>Use Case Name:</b>	<b>Administration Dashboard</b> This use case describes the process of how a user such as an administrator manages user accounts and views the statistical report.
<b>Actors:</b>	
Administrator	
<b>Preconditions:</b>	
<ul style="list-style-type: none"><li>• The user has a valid account.</li><li>• The user has logged into the system.</li><li>• The <i>Administration Dashboard</i> interface is displayed.</li></ul>	
<b>Flow of Events of the Primary Scenario:</b>	

<ol style="list-style-type: none"> <li>1. The use case starts when the user enters into the <i>Administration Dashboard</i> interface.</li> <li>2. The user click <i>View User Account</i> to view the account information.</li> <li>3. If the user wants to add a user account,            The user inputs a user name,            The system checks whether the user name exists,            If the user name does not exist, the user inputs a temporary password and clicks <b>Add</b>.            If the user name does exist, the user needs to input a new user name.</li> <li>4. If the user wants to delete a user account,            The user clicks <b>Delete</b>, the system checks whether it relates to other records,            If the user account relates to other records, the system gives an alerted message;            If not, the system deletes the user account.</li> <li>5. If the user wants to view the statistical exam information,            The user clicks <i>View Exam</i>.</li> <li>6. If the user wants to view the statistical exam report information,            The user clicks <i>View Exam Report</i>.</li> <li>7. If the user wants to export the statistical exam and report information,            The user clicks <b>Export</b>.</li> <li>8. The user clicks <b>Close</b>.</li> <li>9. The use case ends.</li> </ol>
<b>Flow of Events of the Alternative Scenarios:</b>
None
<b>Flow of Events of the Exception Scenarios:</b>
<ul style="list-style-type: none"> <li>• The user cannot login.</li> <li>• The <i>Administration Dashboard</i> interface cannot display.</li> <li>• The network connectivity is lost.</li> </ul>
<b>Extension Points:</b>
None
<b>“Used” Use Cases:</b>
Display the <i>Administration Dashboard</i> interface.
<b>Postconditions:</b>
The corresponding notification is displayed to the valid user.

#### 2.2.8.2 Functional Requirements Associated with Use Case 8

1. ARENAS shall enable the user to add, edit, delete or view a teacher account or a student account.
2. ARENAS shall check whether a user account exists or not when the account is added.
3. ARENAS shall alert a message when deleting a user account which has related records.
4. ARENAS shall enable the user to view the statistical examination information.
5. ARENAS shall enable the user to view the statistical examination report information.
6. ARENAS shall enable the user to export the statistical user account information, the statistical examination information and the statistical examination report information.

## 2.3 Non-Functional Requirements

The non-functional requirements of ARENAS are listed as below.

1. ARENAS shall be usable with the following web browsers on a network-connected desktop or laptop computer or a tablet mobile device:
  - a. Google Chrome version 38 or higher,
  - b. Microsoft Internet Explorer version 6.0 or higher,
  - c. Mozilla Firefox version 32 or higher, and
  - d. Safari version 8 or higher.
2. ARENAS shall be developed and delivered by using the latest versions of the following royalty-free software:
  - a. CentOS Unix operating system,
  - b. NetBeans IDE,
  - c. GlassFish Java EE Application Server, and
  - d. MySQL Relational Database.
3. ARENAS shall not be specific to an educational institution and anybody shall be able to create and administer an exam as well as use it.
4. ARENAS shall enable the creation of multimedia exams containing videos, audios, images, and texts.
5. ARENAS shall be usable with the LockDown Browser and Respondus Monitor tools available from <https://www.respondus.com/>
6. ARENAS shall enable the play of a YouTube video as embedded within a web page.
7. ARENAS shall enable the web content creation through PrimeFaces Editor Widget available from <http://www.primefaces.org/>.
8. ARENAS shall enable the questions to be reusable by different exams.
9. ARENAS shall handle these privacy issues:
  - a. Ensure the user's personal information cannot be shared with any other users or a third party.
  - b. Ensure an instructor's intellectual property rights cannot be shared with any other user or a third party without the instructors' agreement.
  - c. Enable to notify the user any change about their user information.
10. ARENAS shall handle these security issues:
  - a. Identify all of its client applications before allowing them to use the system's capabilities.
  - b. Ensure that all of the users need to change their passwords every 6 months.
  - c. Enable all operations in the system to be recorded in its log.
  - d. Enable all data to be backed-up daily.
  - e. Enable the user's information to be encrypted with the RSA algorithm.
11. ARENAS shall handle these usability issues:
  - a. Enable any graphical user interface to use an adequate font size which is usable even by persons with limited visual acuity.
  - b. Ensure the layouts on every web page are consistent.
  - c. Enable the *Help* text to be provided in *HTML* format.
  - d. Ensure all data can be exported in *XML* format.
  - e. Enable any instructor to create an online exam consisting one question in 2 minutes.

## CHAPTER 3: ARENAS Architecture Specification

Software architecture is the necessary structure of a system and embodies the components and the relationships among these components [Jen and Lee 2000]. Software architecture is a high-level design of a system; it breaks the system into subsystems or components and describes how those subsystems or components interact with each other [McGovern 2004].

Software architecture is useful when structuring complex software systems and it provides a blueprint which is the foundation for later software engineering activities [Aleti et al. 2013]. This is because that software architecture can help build a bridge between the practical requirements and technical requirements.

A software architecture usually covers the following four aspects: 1) exposing the structure of the system, but hiding the implementation details; 2) realizing all of the use cases and scenarios; 3) addressing the requirements of various stakeholders; and 4) handling both functional and non-functional requirements. A set of architectural patterns or architectural styles have been created to help software engineers solve common design problems; these architectural patterns include three representatives: client-server architecture (CSA), peer-to-peer architecture (P2P), and service-oriented architecture (SOA) [Pressman 2010].

This chapter firstly explores the architectural patterns and chooses the appropriate one for ARENAS, then provides ARENAS architecture description focusing on two models based on Department of Defense Architecture Framework (DODAF), from its operational viewpoint and system's viewpoint [2010a, b, c].

### 3.1 Client-Server Architecture

A *Client-Server Architecture* (CSA) is a distributed computing architecture which allocates tasks between servers and clients. The clients send various requests to the servers through the network and the servers respond the appropriate information to the clients [Pressman 2010]. There exist several different types of CSA; the most widely used are two-tier CSA and multi-tiered CSA (*Figure 10 and Figure 11*). The key difference between the different tiered CSA is that the server in two-tier CSA is capable of directly responding all of the clients' requests, while the servers in multi-tiered CSA can specialize in a certain task (i.e., web server, application server, or database server).

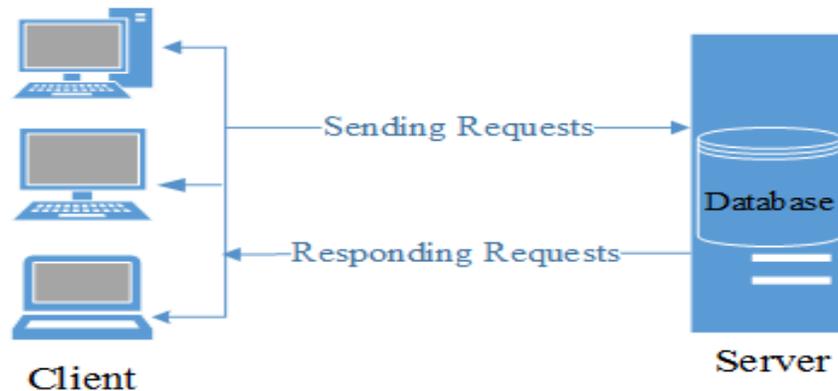


Figure 10. A two-tier Client-Server Architecture [Pressman 2010]

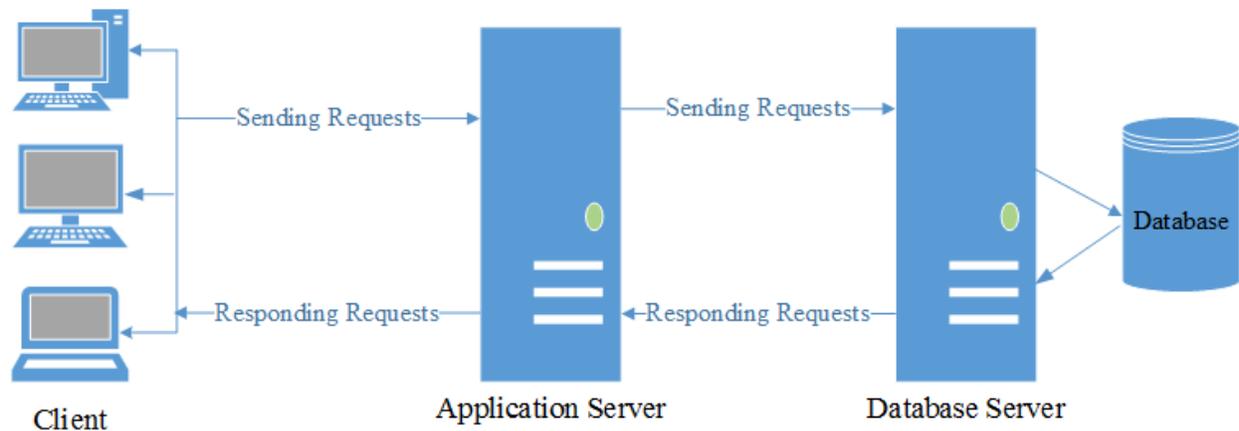


Figure 11. A multi-tiered Client-Server Architecture [Pressman 2010]

Compared to a two-tier CSA, a multi-tiered CSA splits the user interface, the data logic and the database server into different separated tiers, so that the business logic part and data logic part run as independent processes on server computers.

The advantages of CSA include: 1) task specificity and independence, which means that the diverse tasks can be executed on the most appropriate server computer; 2) increase of performance and security; and 3) a greater degree of flexibility. The distinction of the tasks in the proper tier is the core of a successful CSA application [Bibinagar and Kim 2013].

### 3.2 Peer-to-Peer Architecture

A Peer-to-Peer (P2P) architecture is a distributed application architecture which allocates tasks between workstations and servers [Kwok 2012]. These workstations or servers can be seen as peers. Each peer has the same responsibilities and each peer can launch a session with other peer (**Figure 12**). P2P can be decentralized, centralized or hybrid structure; each peer of P2P can be connected to many others [Vu, Lupu and Ooi 2010].

In a P2P application, each peer may share a part of their own computing resources and each peer is both resource provider and resource requestor. The advantages of a P2P application include: 1) unnecessary specialized application and database servers and 2) good scalability and reliability. The shortcoming of a P2P application is poor security since it allows to easily share and download copyrighted resources.

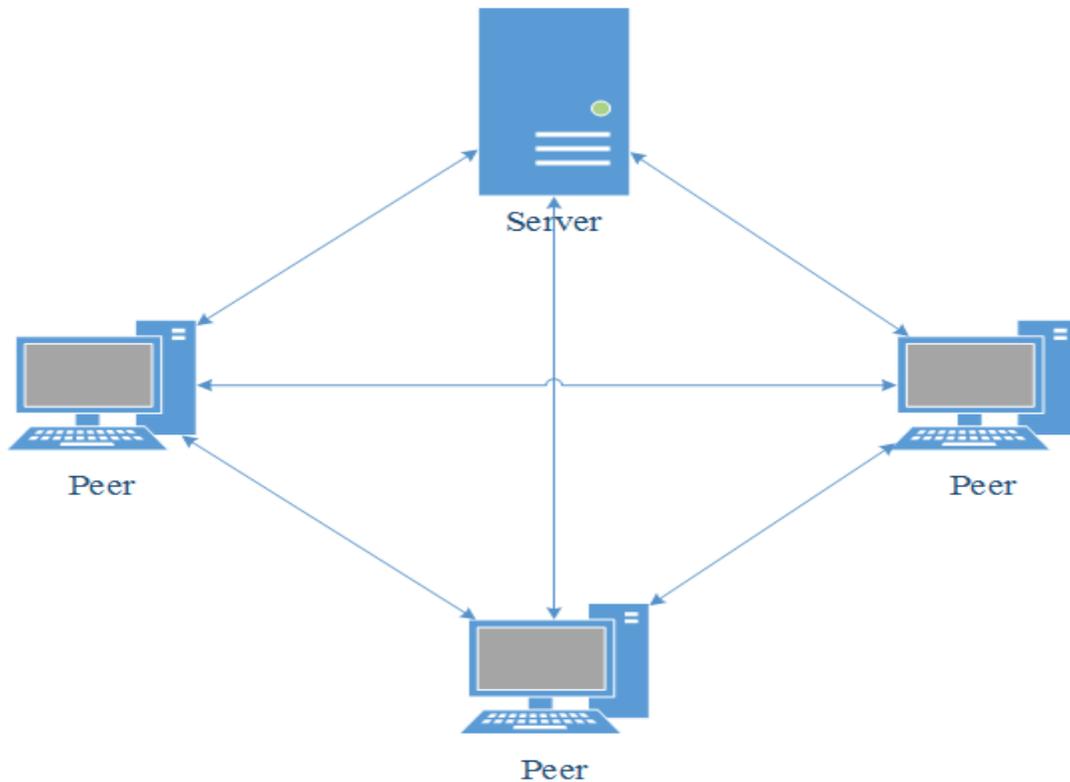


Figure 12. A Peer-to-Peer Architecture [Kwok 2012]

### 3.3 Service-Oriented Architecture

Service-oriented architecture (SOA) is a loosely-coupled architecture for designing and developing distributed systems [Li, Muthusamy and Jacobsen 2010]. SOA is based on the concept of service and any service can be seen a software component. SOA proposes that one computer application's needs might be met by the capabilities offered by another computer application offered by a different owner. The principle of service-orientation is independent of any technology, product or vendor [IBM 2007].

Web services are the basis for integrating different applications in SOA. Any SOA solution is designed for allowing integration regardless of diverse programming languages, protocols or hardware; any SOA solution can be designed to have the capabilities of flexible incorporation with legacy system applications, streamlined business logics and processes, and adaptable applications to the contextual environments [Rosen et al. 2012].

A SOA generally includes service provider, service broker and service consumer (**Figure 13**). A SOA architecture is very useful when there are a bunch of legacy system applications needed to be integrated. Take a university as an example, several disparate departments within the university may develop and deploy SOA services in different implementation languages, but their users may benefit from a well-defined interface to access them [Pressman 2010].

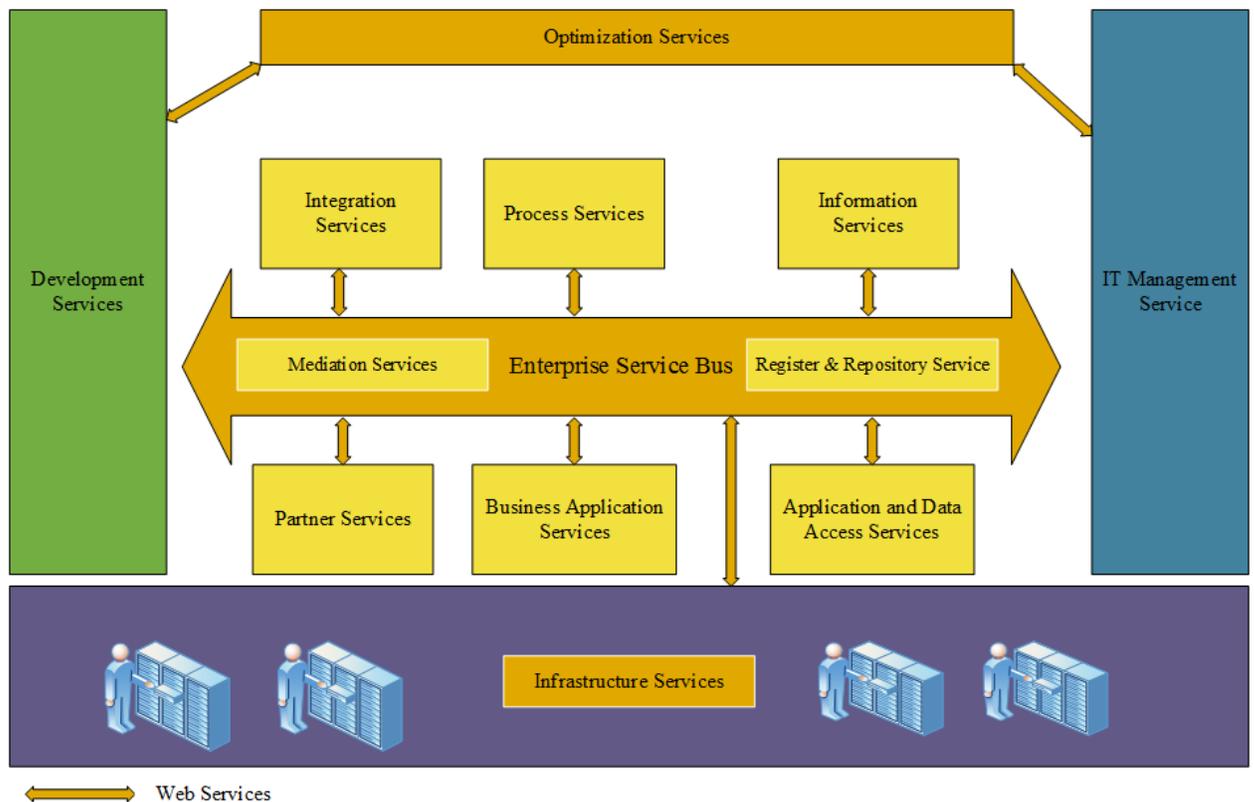


Figure 13. A Service-Oriented Architecture [IBM 2007]

### 3.4 Java EE-based Client-Server Architecture

Compared to P2P and SOA, the multi-tiered CSA is the most appropriate for ARENAS due to: 1) one purpose of ARENAS is to enable anybody to create and administer an exam as well as use it; students can just take an exam with a Brower, therefore, there is no need to use P2P; 2) there is no legacy systems which need to be integrated into ARENAS and ARENAS is independent; and 3) multi-tiered CSA can provide a very useful model for the design and development of ARENAS.

Two major industry standards can be employed to build a system with CSA: Java Platform, Enterprise Edition (Java EE) [Oracle 2015] and Microsoft .NET Framework [Microsoft 2014]. They both aim to simplify the development of applications by providing a set of standardized, modular components and services, and offer different tools and methods to achieve the same goal of enterprise level application development with similar computing technologies. But they have some differences and there are not necessary advantages or disadvantages for one platform over the other like: Windows vs. Multi-Platform, Proprietary vs. Open Source. It is hard to say which one is better. However, considering the cost of building, deploying, and maintaining ARENAS, as well as our own situation and the needs of the application, we would like to choose Java EE Platform for ARENAS.

A Client-Server Architecture (CSA) based on the Java platform, Enterprise Edition (Java EE) [Oracle 2015], is shown as follows (**Figure 14**) with five tiers: 1) client tier, 2) web tier, 3) business tier, 4) data mapping tier, and 5) data source tier [Balci 2015].

Layer 1. *Client Presentation Layer* (e.g., Asynchronous JavaScript and XML (AJAX), Cascading Style Sheets (CSS), Document Object Model (DOM), Extensible

HTML (XHTML), Extensible Markup Language (XML), Extensible Stylesheet Language Transformation (XSLT), HyperText Markup Language (HTML))

- Layer 2. *Web Container Layer* (consisting of controller/mediator and server-side presentation preparation components)
- Layer 3. *Business or Application Logic Layer*
- Layer 4. *Data Mapping Layer* (e.g., Entity Enterprise Java Beans, ActiveX Data Objects)
- Layer 5. *Data Source Layer* (e.g., relational database management systems)

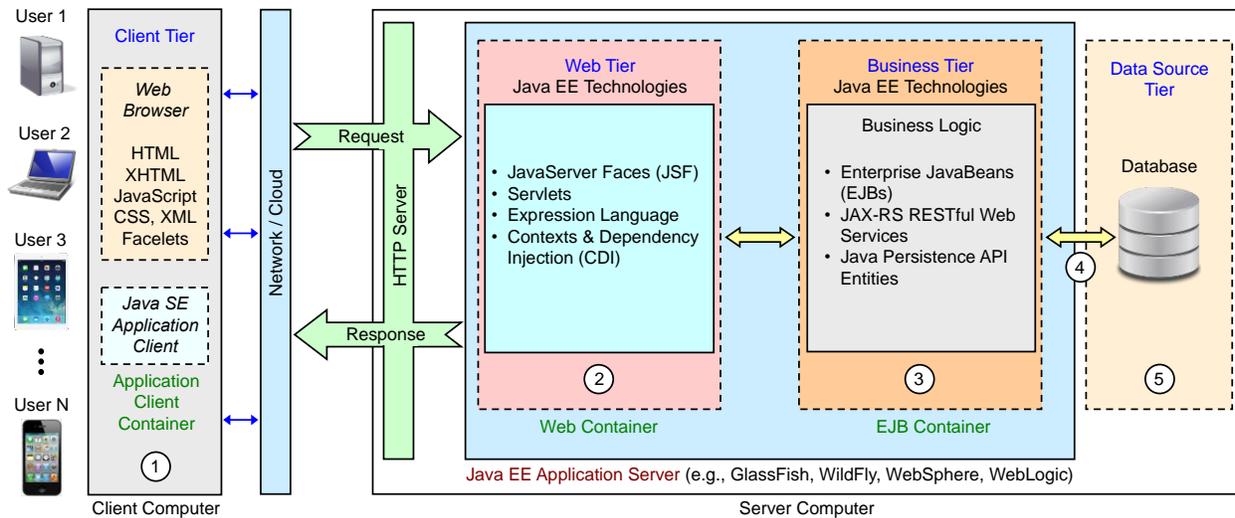


Figure 14. A Java EE-based Client-Server Architecture [Balci 2015]

### 3.5 ARENAS Architecture Model Based on DODAF

The ARENAS architecture model is designed based on the DODAF [2010a, b, c] and follows the business rule of the process of architecting for software engineering. We provide two models for ARENAS based on DODAF from its operational viewpoint and systems viewpoint.

#### 3.5.1 OV-2: Operational Resource Flow Description

In the following Figure (Figure 15), we describe how the resources flow in ARENAS. Once an instructor creates a question repository, he or she can add questions into the question repository, select questions from the question repository for creating an exam. The students can take an exam and submit the answers to the exam questions, then an exam report is generated; the instructor can check and tweak the exam report, especially for scoring the student answers to the subjective questions. The students can view, download or export the exam report. An administrator can view the statistical exam or exam report information as well as manage instructor accounts and student accounts.

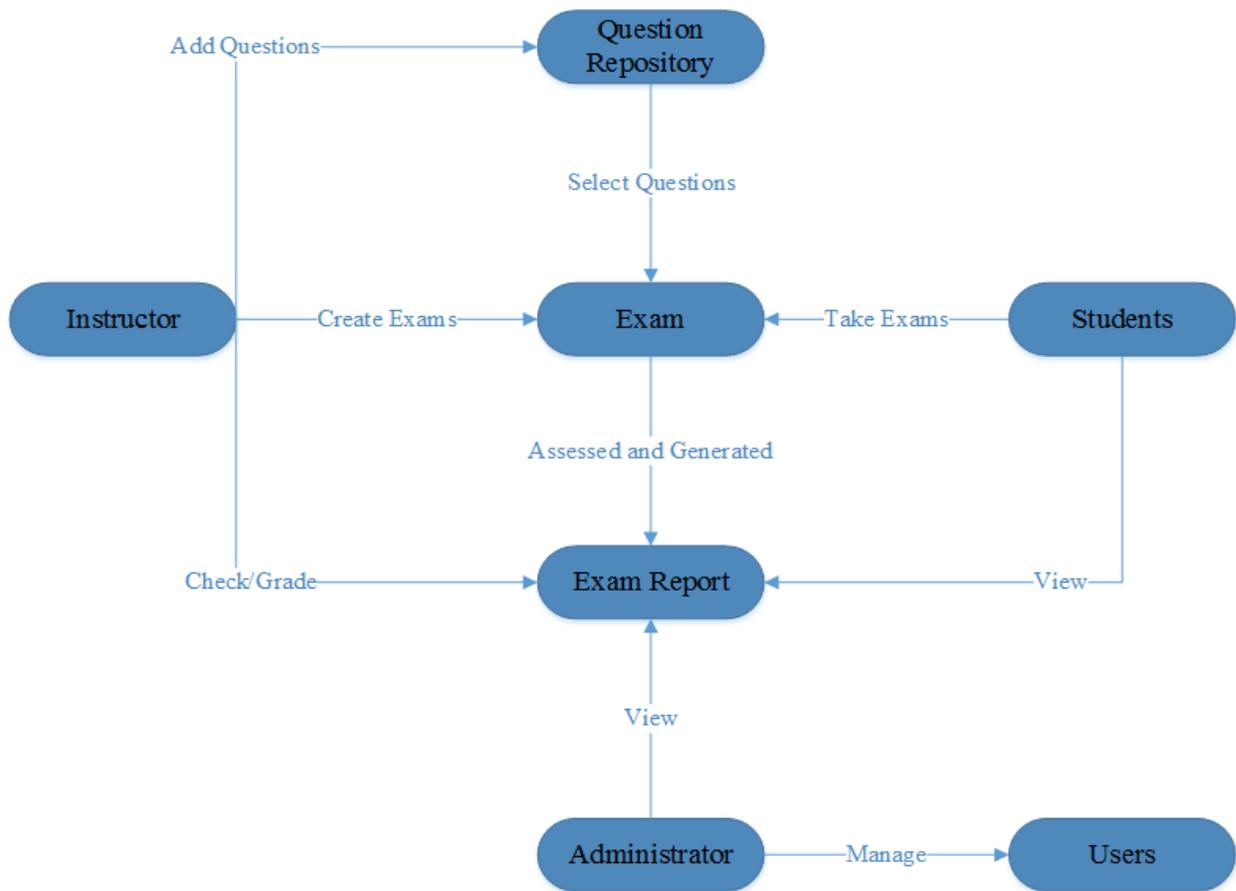


Figure 15. OV-2: Operational Resource Flow Description

### 3.5.2 SV-4: Systems Functionality Description

The following Figure (**Figure 16**) describes the functions performed by ARENAS with a tree system structure. According to the figure (**Figure 16**), ARENAS includes five modules: 1) User Module, 2) Question Repository Module, 3) Exam Module, 4) Exam Report Module, and 5) Configuration Module. Each module contains several sub-functions which are associated with particular capabilities.

To be specific, the User Module includes these sub-functions: registering a user account, authenticating a user account, creating a user account, deleting a user account, updating a user account, viewing a user account, resetting user passwords, exporting user accounts, and importing user accounts. The Question Repository Module contains these sub-functions: creating a question repository, updating a question repository, deleting a question repository, viewing a question repository, adding a question into a question repository, updating a question in a question repository, deleting a question from a question repository, viewing a question in a question repository, and setting points to a question. The Exam Module includes these sub-functions: creating an exam, updating an exam, deleting an exam, viewing an exam, adding a question into an exam, deleting a question from an exam, updating a question in an exam, viewing a question in an exam, publishing an exam and taking an exam. The Exam Report Module contains these following sub-functions: generating an exam report, checking and tweaking an exam report, adding a brief summary in an exam report, updating a brief summary in an exam report, deleting a brief summary from an exam report, viewing a brief summary in an exam report, publishing an exam report, emailing an exam report, viewing an exam report and exporting an exam report. The Configuration Module contains these

sub-functions: setting question type, setting exam policy, setting browser, backup data, software update, and API.



Figure 16. SV-4: Systems Functionality Description

## CHAPTER 4: ARENAS Design Specification

As a design and development method, the object-oriented (OO) paradigm is on the basis of the concept that systems should be built on reusable components which are called objects. The objects in object-oriented paradigm include both data and functionality, instead of isolating data and functionality as do in the structured paradigm [Roy and Das 2012]. For ARENAS, the objected-oriented design is very useful for managing the complexity of the system through reusing components [Tegarden, Dennis and Wixom 2012].

UML (Unified Modeling Language) is created as the modelling language for designing and developing a large-scale software system; it provides various notations for representing different aspects of software system development [OMG UML 2011; Weikens 2011].

This chapter focuses on describing several design aspects for ARENAS based on the objected-oriented design with UML, including 1 class diagram, 1 package diagram, 1 component diagram, 2 statechart diagrams, 1 activity diagram, 2 sequence diagrams, 2 collaboration diagrams, and 1 deployment diagram.

### 4.1 Class Diagrams

A *class diagram* describes the basic information about the classes, their data fields and methods, as well as their relationships in an object-oriented system design [OMG UML 2011].

The class diagram (*Figure 17*) shows 10 classes and their relationships for ARENAS. Each class has its data fields and methods, as well as relationships with other classes. To be specific, the class “Instructor” and “Student” inherit the class “Users”; the class “Address” is aggregated to the class “Users”. The class “Users” is associated with the class “TakeExams”, which is also associated with the class “Exams”. The class “QuestionType” is aggregated to the class “Questions”; the class “Questions” is aggregated to the class “Exams”. The class “Exams” is associated with the class “AssessExam”, which is also associated with the class “ExamReports”.

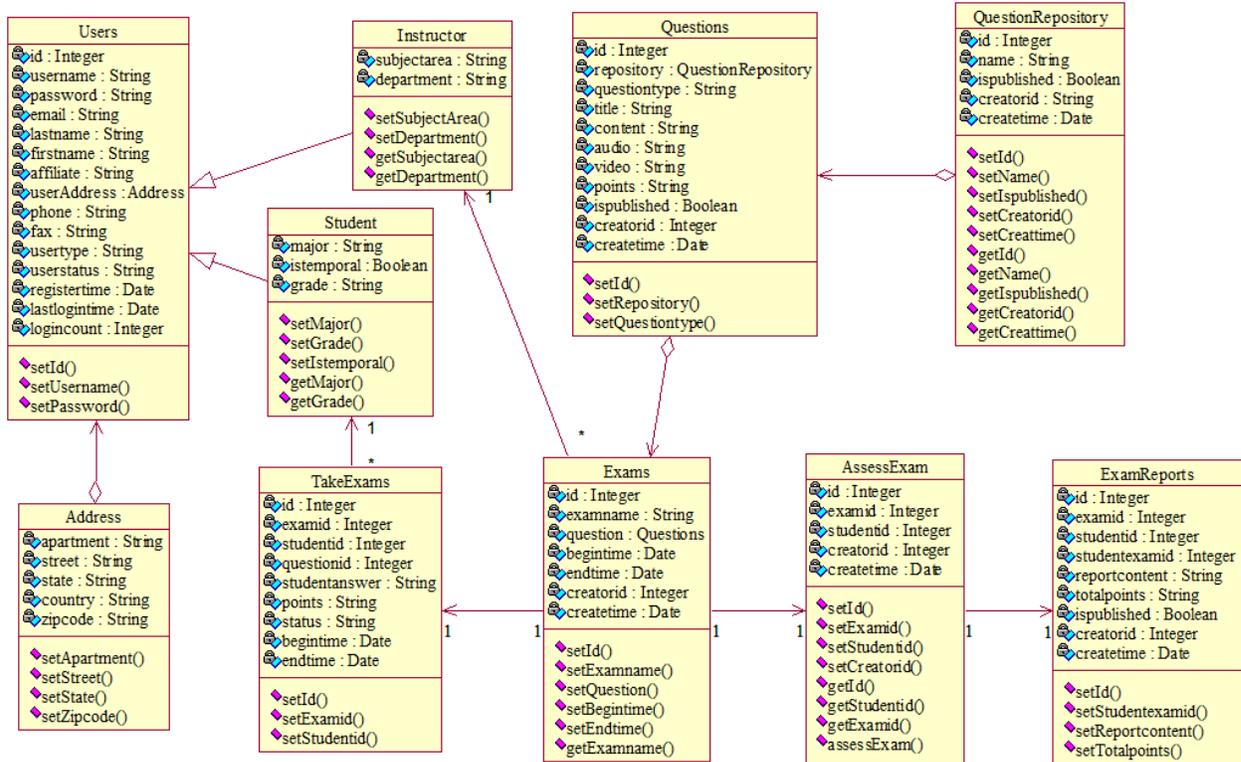


Figure 17. ARENAS Class Diagram

## 4.2 Package Diagrams

A *Package diagram* is a subset of class diagrams, which organizes elements of a system into related groups to minimize the amount of dependencies. Package diagrams can be seen as a high-level view of a system when compared to class diagrams. One package can contain other packages in the diagram, displaying the degree of organization in the system [OMG UML 2011].

The package diagram (*Figure 18*) shows four packages and their structures and relationships. The exam package is dependent on the user package; the exam assessment package is dependent on the user package and exam package; the user package, exam package and exam assessment package are dependent on the database package.

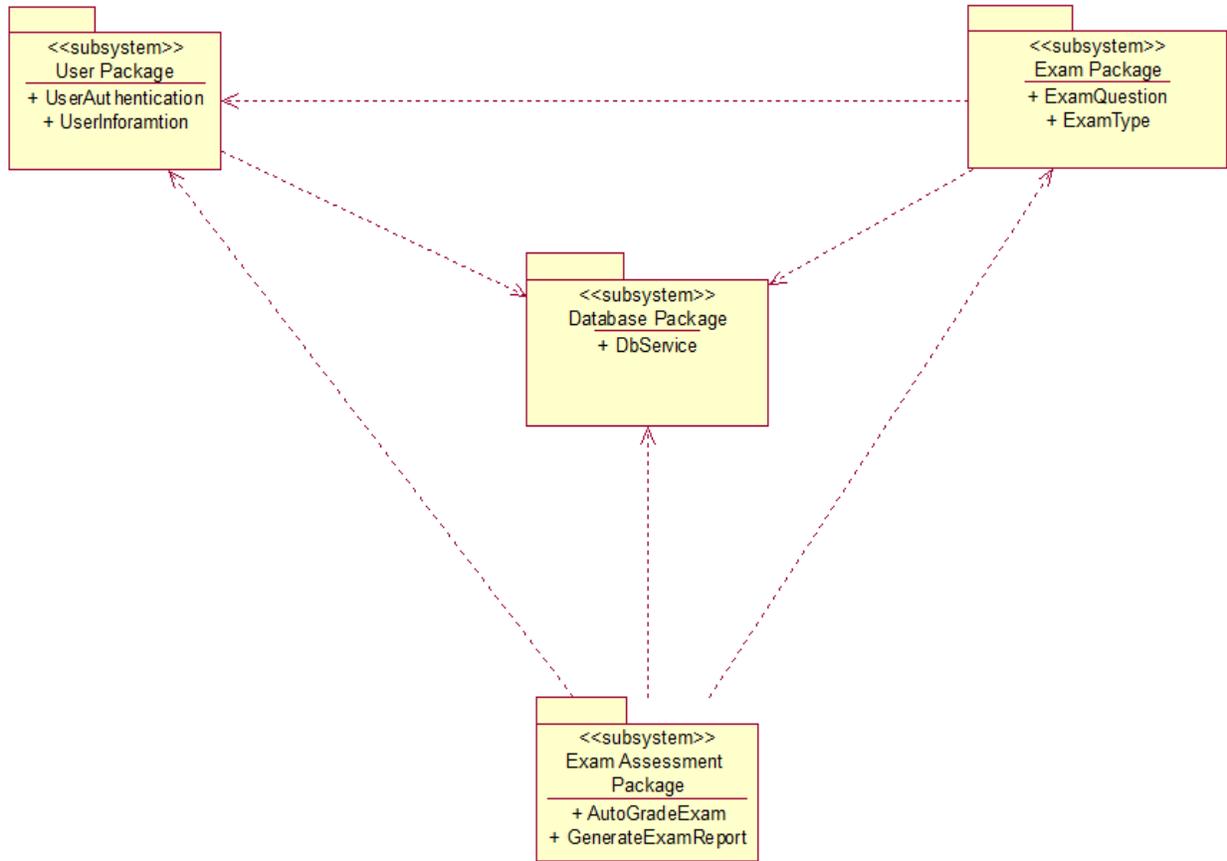


Figure 18. User-Exam-Database Package Diagram

### 4.3 Component Diagrams

A *component diagram* shows the organizations and dependencies among software components, including source code components, binary code components, and executable components. Component diagrams also demonstrate the interactive behaviors through exposing the interfaces of the components [OMG UML 2011].

The component diagram (*Figure 19*) shows the dependencies among four components that work together in ARENAS. The question repository component and exam component are dependent on the authorization component; they are also dependent on the question component. The exam report component is dependent on the exam component.

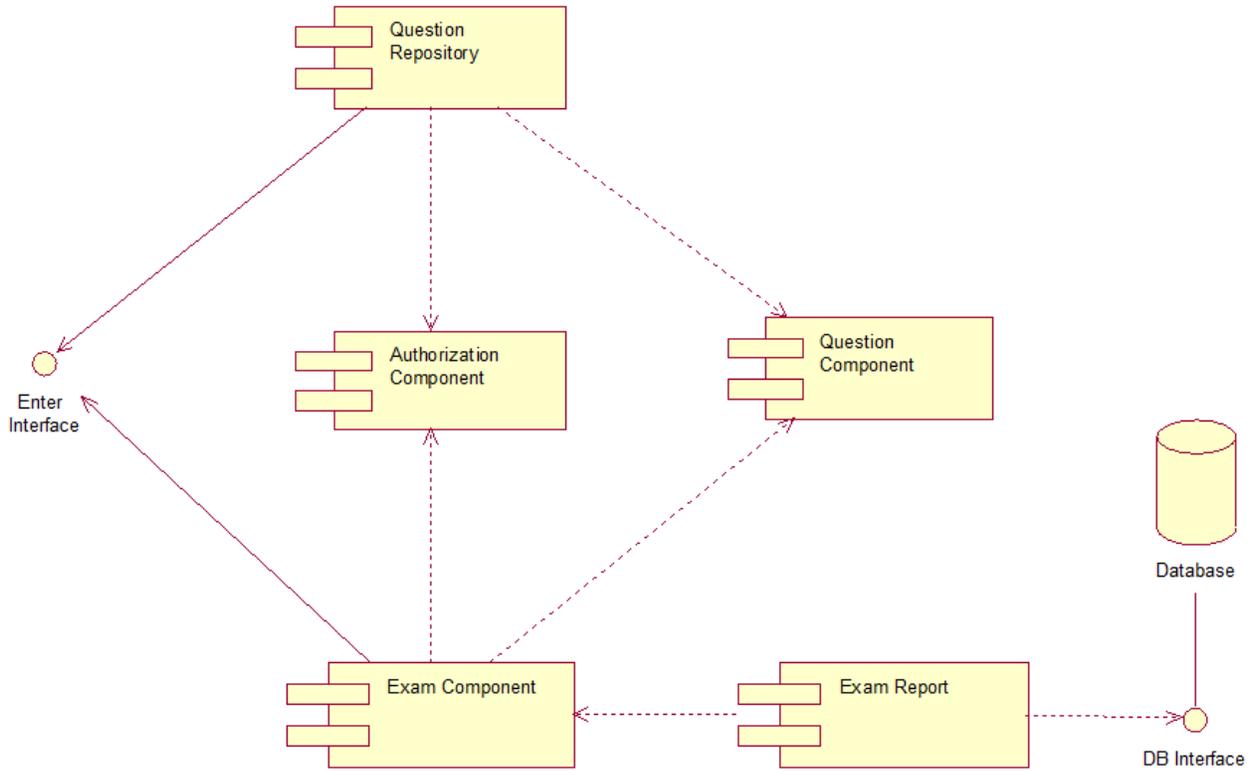


Figure 19. Exam-Question Component Diagram

#### 4.4 Statechart Diagrams

*Statechart diagrams or state diagrams* depict the behavior of a system through demonstrating all of the possible states of an object when events occur [OMG UML 2011].

##### 4.4.1 Instructor Statechart Diagram

The statechart diagram (**Figure 20**) shows the different states when an instructor logs into ARENAS. Beginning with the initiate state, if an instructor account is verified to be incorrect, the instructor cannot log into the system successfully; if the instructor account is verified to be correct, the instructor logs into the system successfully; the instructor can create question repository, add questions into the question repository, create an exam and publish an exam, then the state ends.

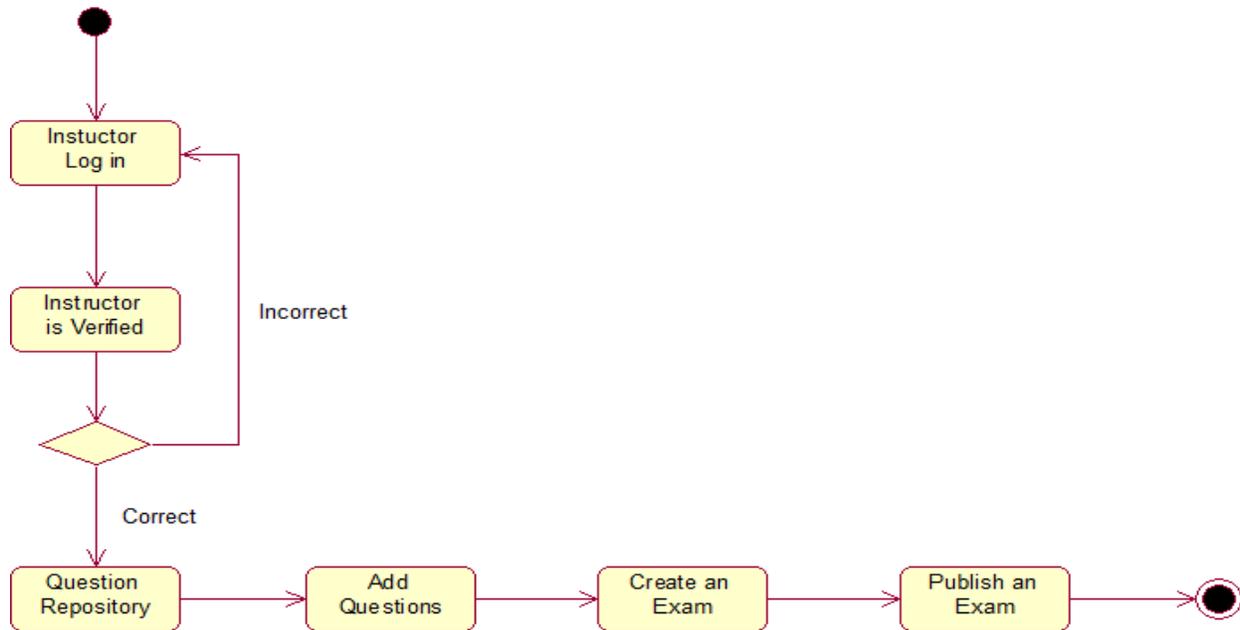


Figure 20. Instructor Statechart Diagram

#### 4.4.2 Student Statechart Diagram

The statechart diagram (**Figure 21**) shows the different states when a student logs into ARENAS. Beginning with the initiate state, if a student account is verified to be incorrect, the student cannot log into the system successfully; if the student account is verified to be correct, the student logs into the system successfully; the student can choose an exam, take an exam, and submit an exam, then the state ends.

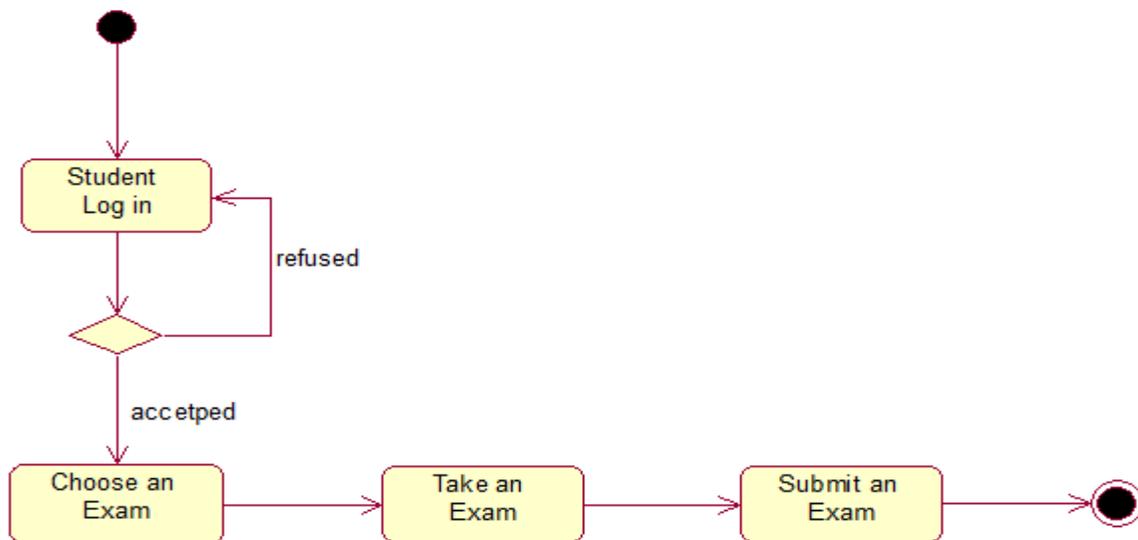


Figure 21. Student Statechart Diagram

## 4.5 Activity Diagrams

Activity diagrams depict the workflow behavior of a system. The diagrams show the states of activities through presenting the sequences of activities performed. Activity diagrams can display activities that are conditional or parallel; this diagram provides a way to model the workflow of a process [OMG UML 2011].

The activity diagram (*Figure 22*) shows the sequent activities related to exams. Beginning with the initiate state, one instructor creates a question repository, adds questions into the question repository, selects questions from the question repository into an exam created by him or her, and publishes the exam. Then a student takes the exam, submits the answers; the system generates the exam report; the instructor tweaks the report or not; the student views the exam report, then the activity ends.

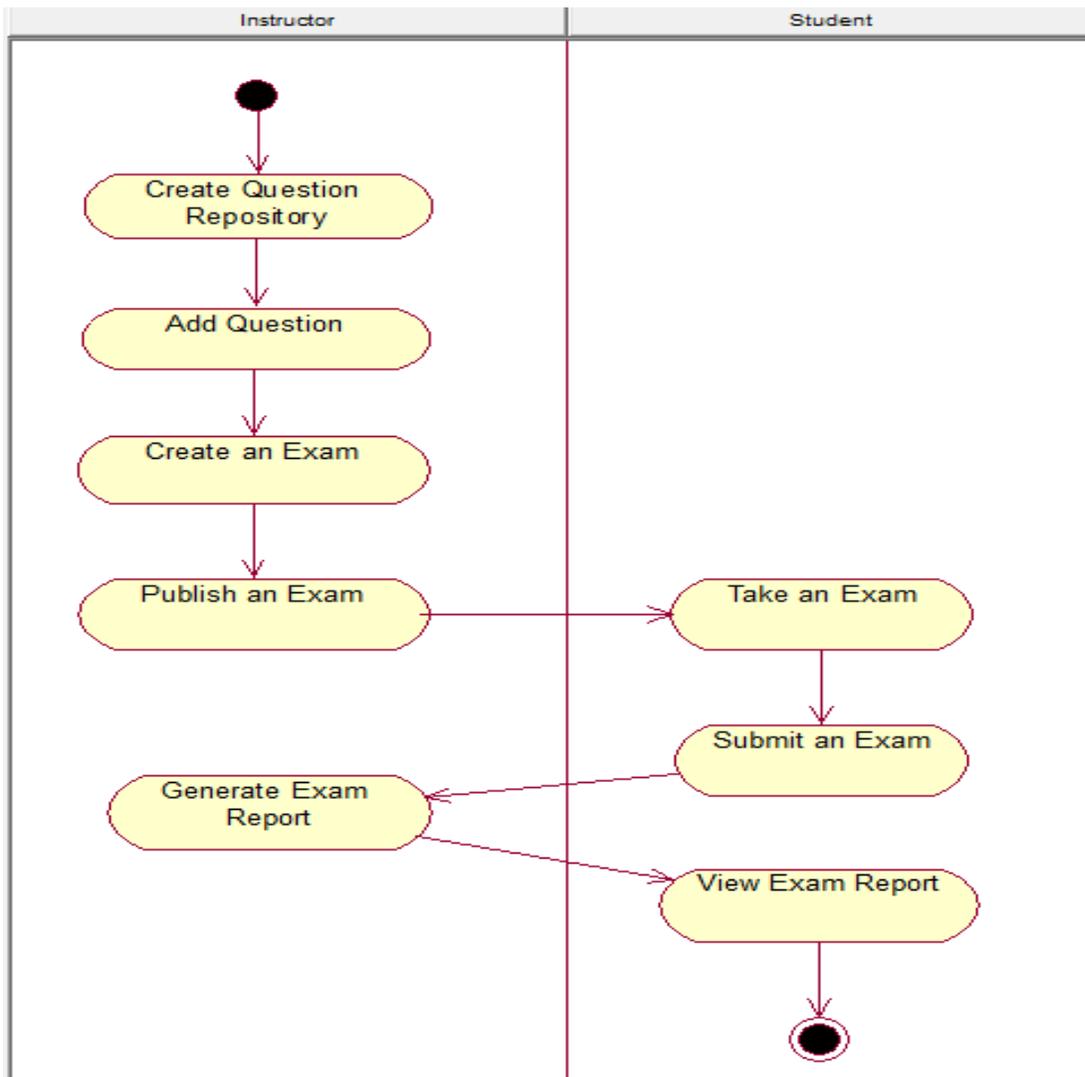


Figure 22. Instructor and Student Activity Diagram

## 4.6 Sequence Diagrams

A *sequence diagram* is a graphical view of a scenario that shows the interactions in a time-based sequence. Sequence diagrams display the sequences of events that occur [OMG UML 2011].

### 4.6.1 Exam Creation Sequence Diagram

The sequence diagram (**Figure 23**) illustrates the time-based sequent interaction for creating an exam by an instructor. An instructor creates a question repository, adds questions into the question repository, then the instructor selects questions from the question repository and adds them to an exam, so as to create an exam.

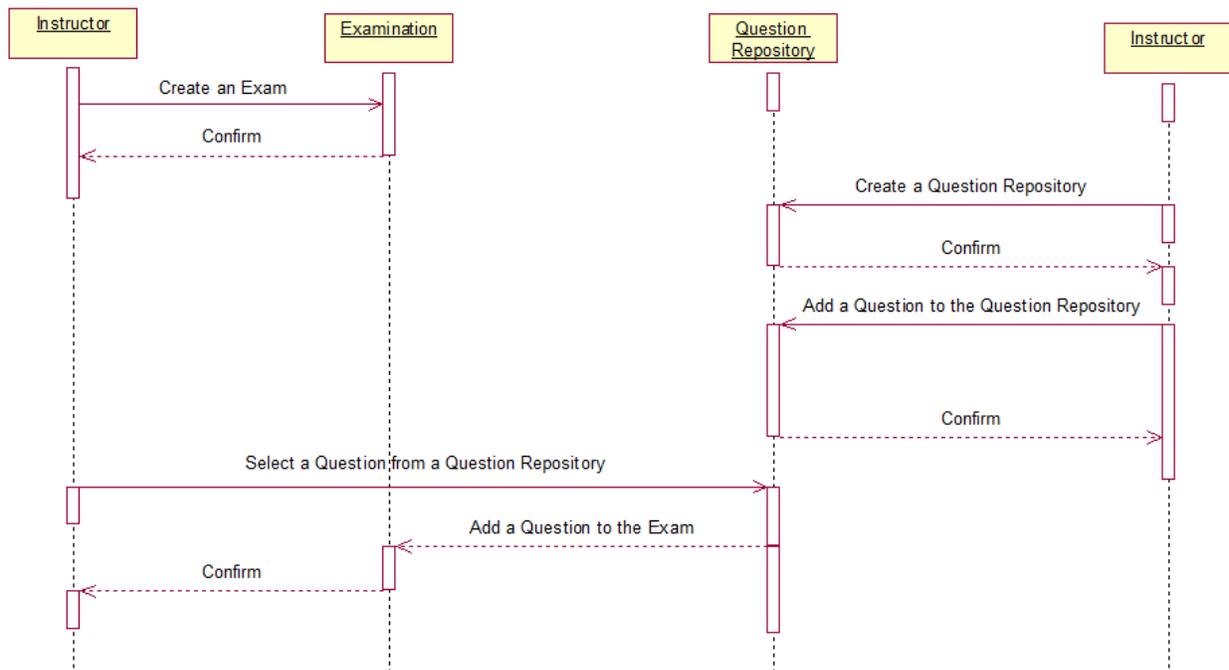


Figure 23. Exam Creation Sequence Diagram

### 4.6.2 Exam Taking Sequence Diagram

The following sequence diagram (**Figure 24**) displays the time-based sequent interaction for taking an exam by a student. A student takes an exam, then submits the answers to the exam; the student's answers are assessed by the system and an exam report is generated; an instructor checks or tweaks the exam report and publishes the exam report; then the student receives the exam report.

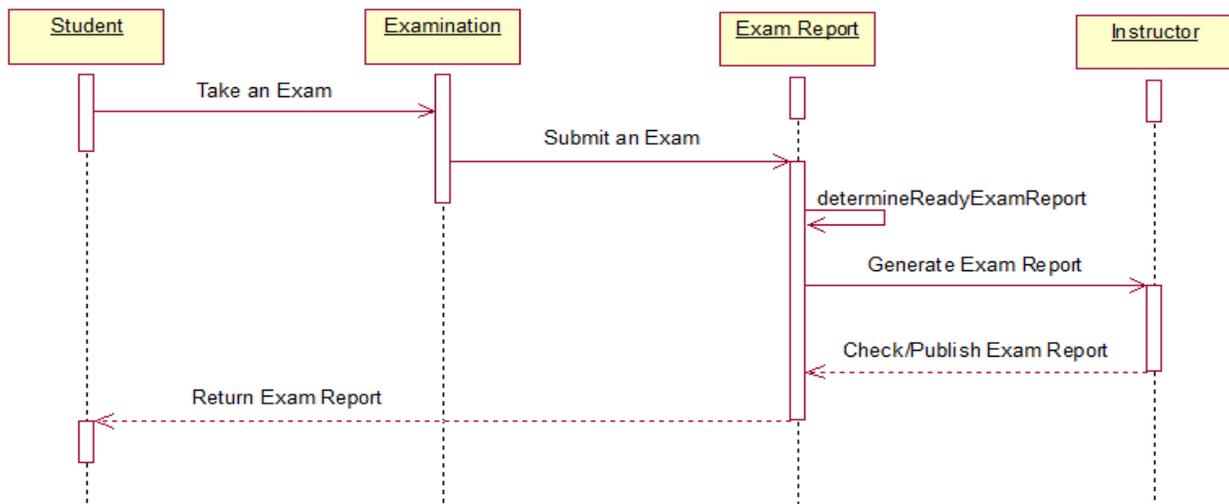


Figure 24. Exam Taking Sequence Diagram

## 4.7 Collaboration Diagrams

A *collaboration diagram* shows the order of messages when implementing an operation or a transaction. Collaboration diagrams often display objects, their links and their messages [OMG UML 2011].

### 4.7.1 Exam Creation Collaboration Diagram

The exam creation diagram (**Figure 25**) shows the order of the messages when an instructor creates an exam. When an instructor plans to create an exam, he or she firstly selects a question from a question repository, then the question repository returns the question and the instructor adds the question into an exam.

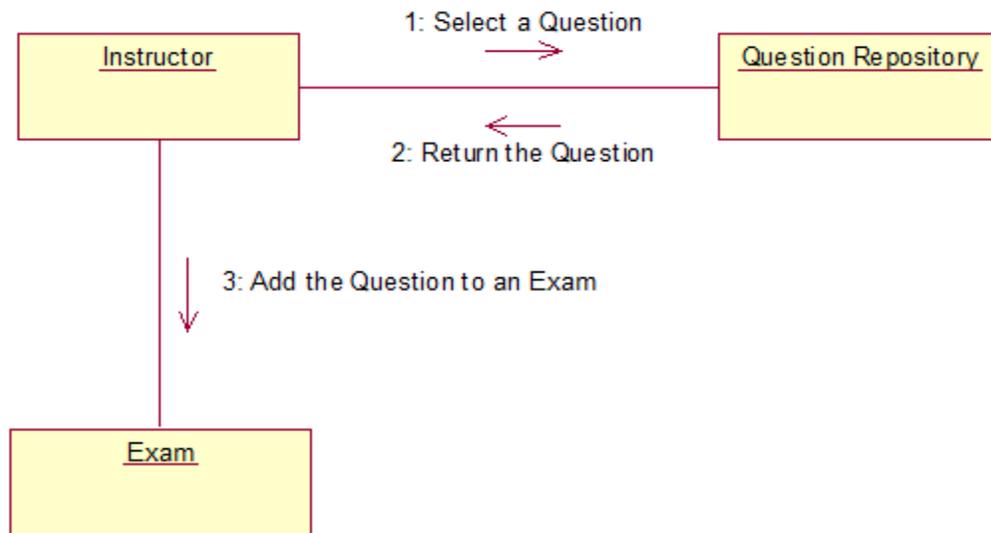


Figure 25. Exam Creation Collaboration Diagram

#### 4.7.2 Exam Taking Collaboration Diagram

The exam creation diagram (**Figure 26**) shows the order of the messages when a student takes an exam. When a student chooses an exam to take, he or she gets the confirmation information, after the student completing the exam, he or she submits the answers to the exam; the exam assessment module returns the exam report.

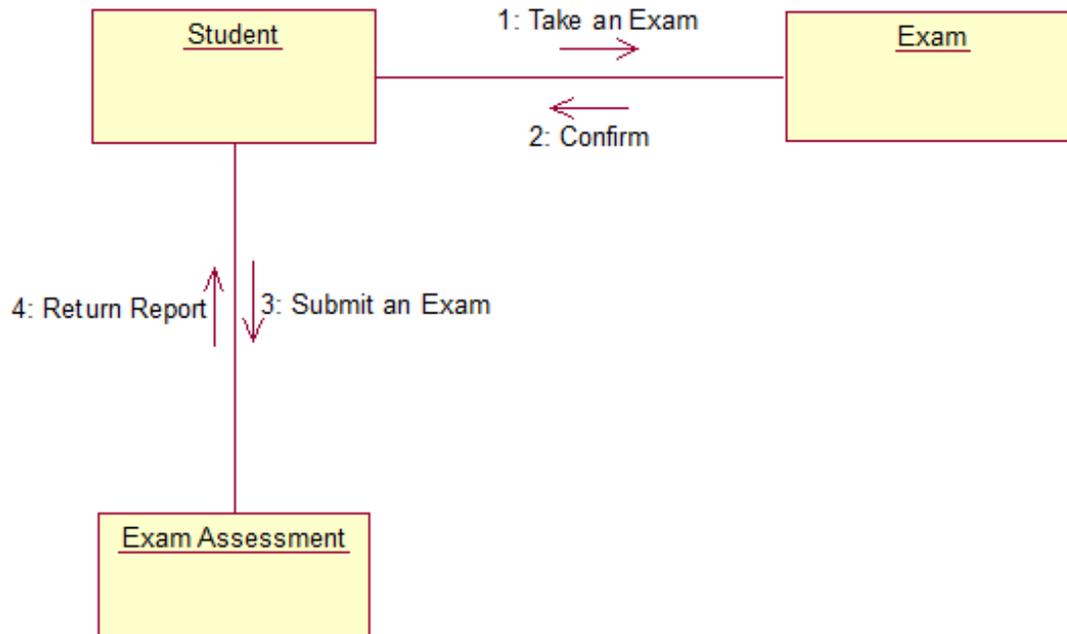


Figure 26. Exam Taking Collaboration Diagram

#### 4.8 Deployment Diagrams

A *deployment diagram* shows the run-time configuration of processing nodes and the components that run on those nodes. Deployment diagrams display processors, devices, and connections in terms of hardware perspective for a system [OMG UML 2011].

The deployment diagram (**Figure 27**) shows the physical deployment of the hardware components or the nodes in ARENAS. The Database Server is MySQL; the Multiple Virtual Storage Mainframe like IBM MVS can be added seamlessly for data storage and backup. The application Server is Glassfish and the Firewall is Cisco products. The protocol between web clients and Firewall is Http/Https; the protocol for other different processors is TCP/IP.

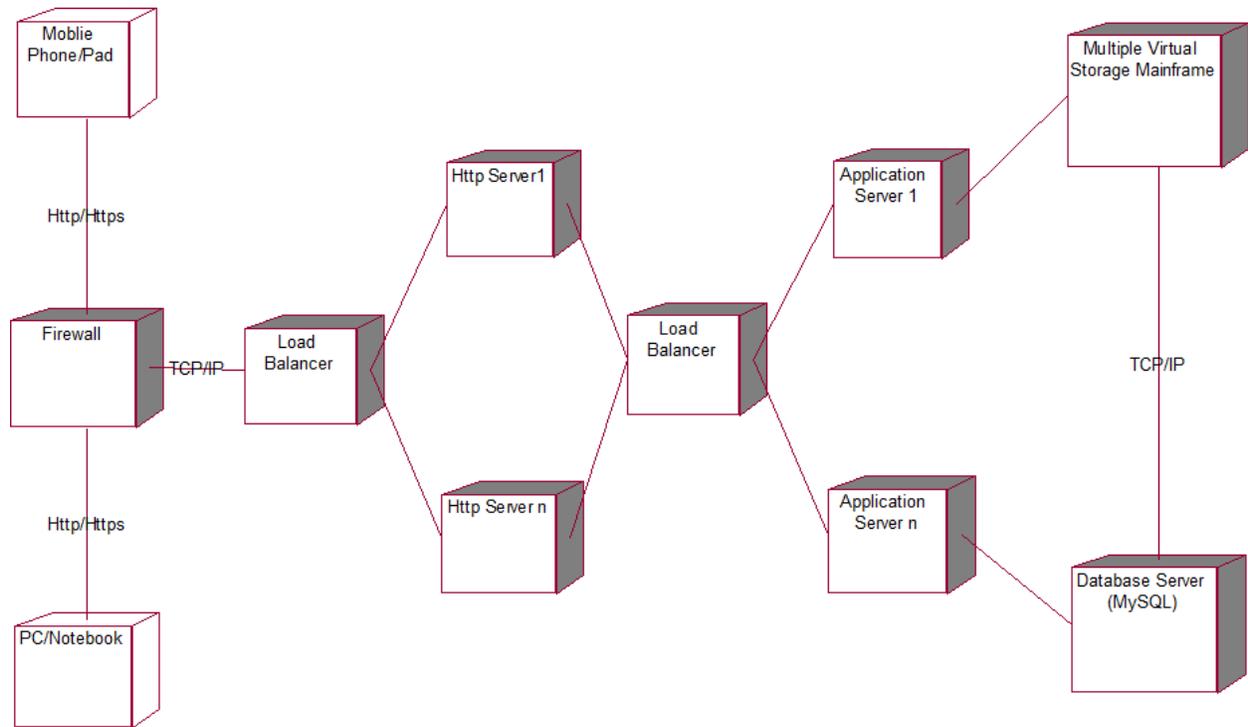


Figure 27. ARENAS Deployment Diagram

# CHAPTER 5: ARENAS Functionality

This chapter describes the developed functionalities of ARENAS. There are three user roles in this system: instructor, student and administrator in ARENAS. Each user role performs different functionalities. The following section presents the developed functionality for these use roles.

## 5.1 User Registration

A regular user can register to be an instructor or a student. A user needs to provide the following required information: user name, user type, email and major field. The admin account is set by default in the system (*Figure 28*).

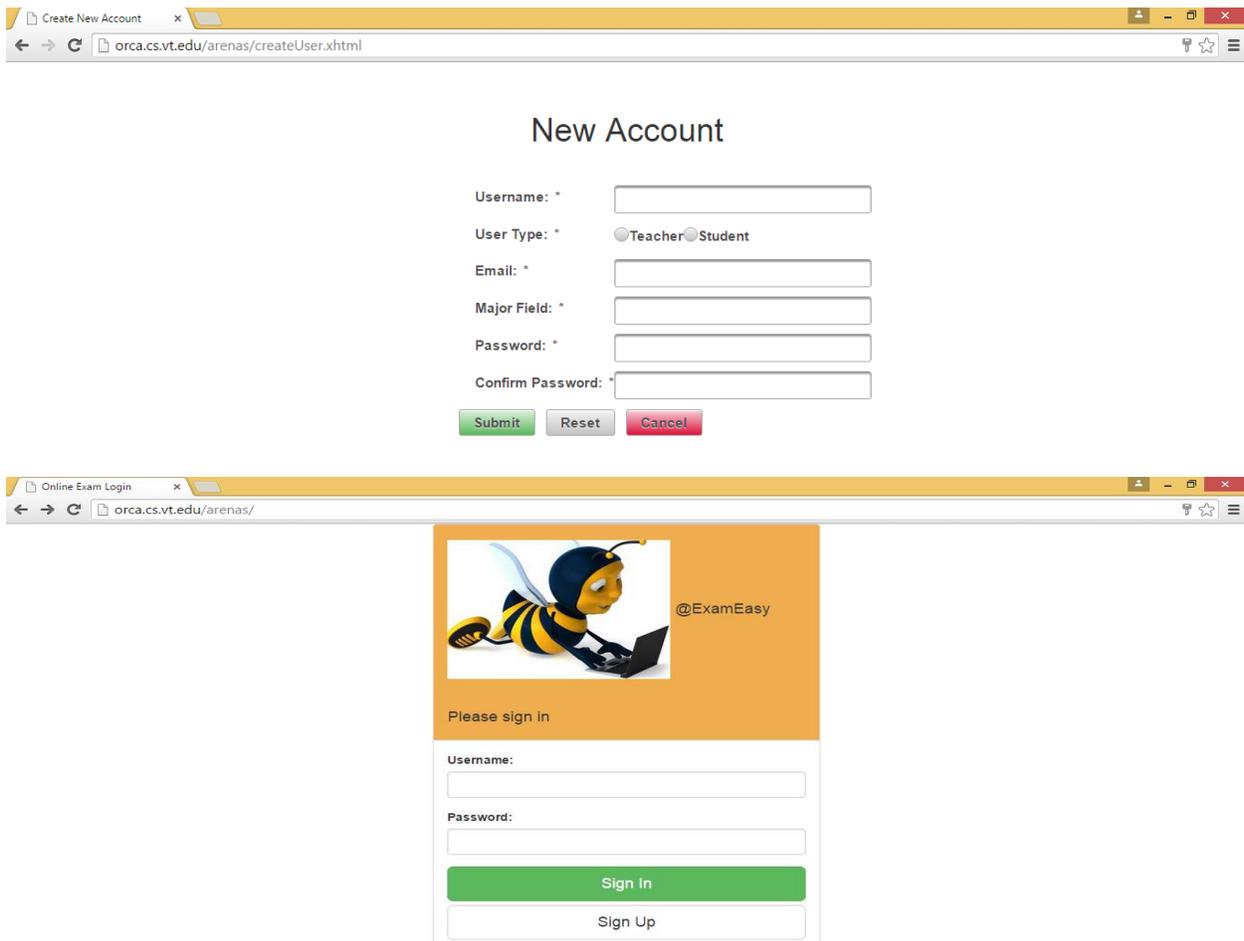


Figure 28. User Registration Interface

## 5.2 Question Repository

An instructor can create, delete, edit or view a question repository. When creating a question repository, the following information are required: repository name, description, and whether published or not (*Figure 29*).

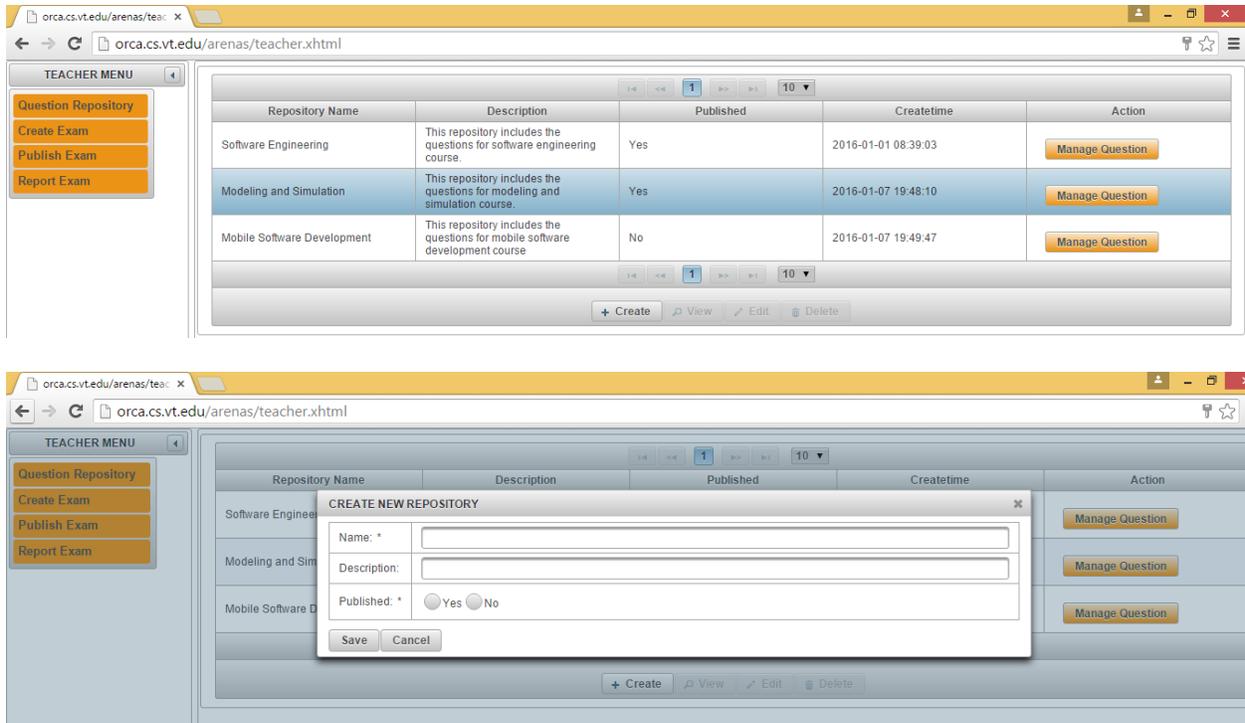


Figure 29. Question Repository Interface

### 5.3 Question Management

After a question repository is created, an instructor can add a question into the repository, delete a question from the repository, edit or view a question in the repository. Currently, three types of questions: single-choice question, multiple-choice question and essay/short answer question can be added into a question repository; other types of question will be added in the future. When creating a question, the following information are required: question type, question title, question content, question options (only for single-choice question or multiple-choice question), points and whether published or not.

When a question is created, an instructor can choose to add multimedia elements such as images, graphics, video or audio; these multimedia elements can be uploaded into the Server in which ARENAS is running if the instructor selects the “upload” option. If the instructor selects “embedded” option, he or she can copy a URL which directs a video in Youtube website and clicks “Submit URL” button, the video will be automatically embedded into the question content.

When creating a single-choice question or multiple-choice question, an instructor also needs to mark the correct answers, which will be stored into the database for auto-grading and generating an exam report (*Figure 30*).

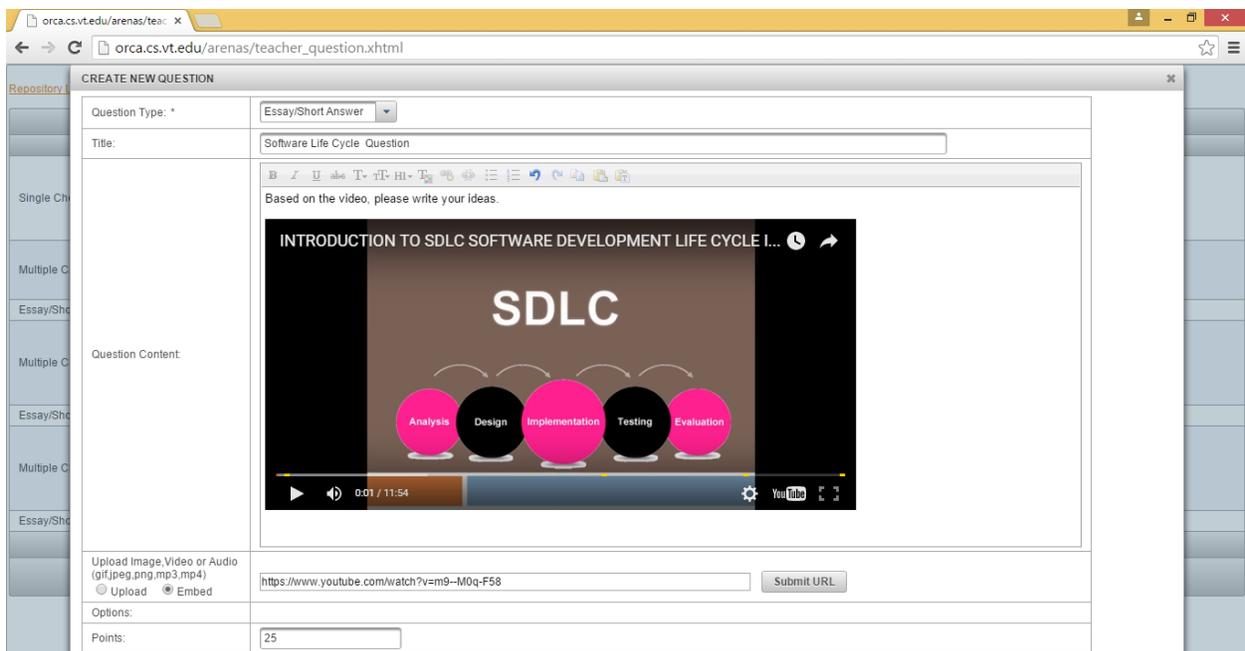
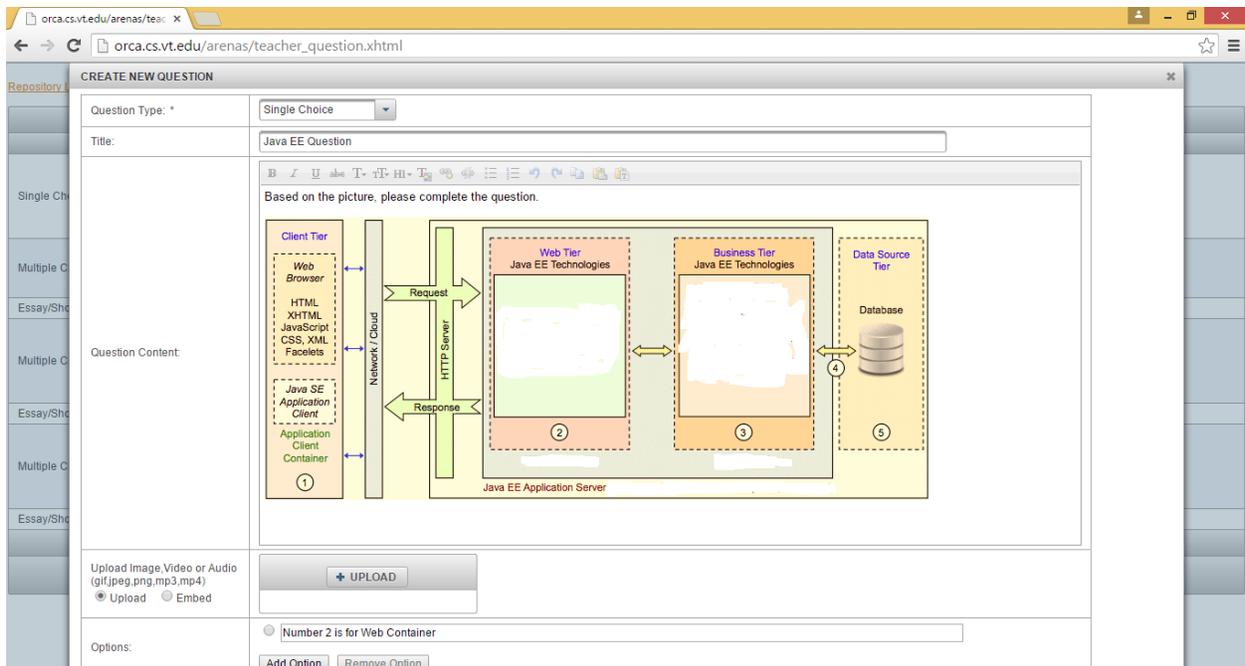


Figure 30. Question Management Interface

## 5.4 Exam Creation

An instructor can create, edit, delete or view an exam anytime. The following information are required when creating an exam: exam name, exam description, beginning time and ending time. When setting the exam time, the exam time information will be formatted into YYYY-MM-DD HH:MM:SS at the backend (*Figure 31*).

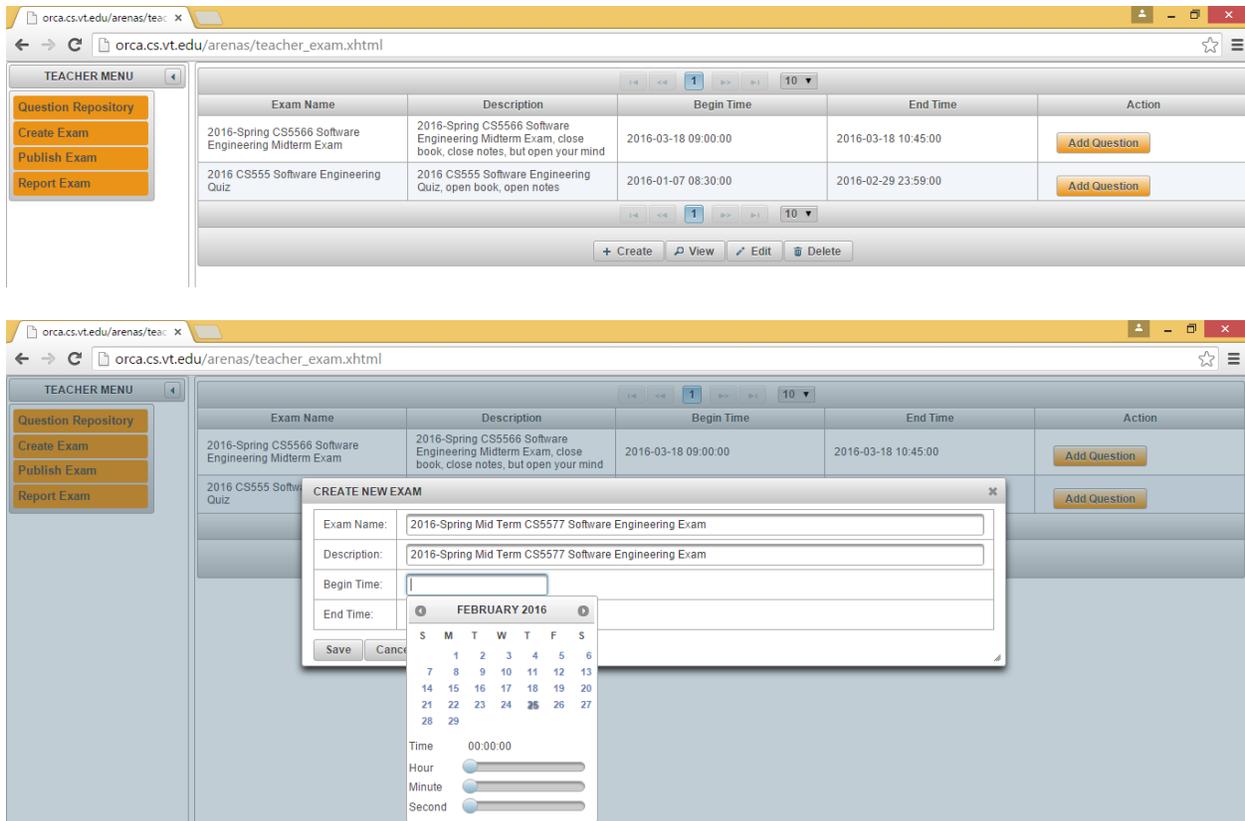


Figure 31. Exam Creation Interface

## 5.5 Exam Question Management

After an exam is created, an instructor needs to select questions from the question repositories, so as to add these questions into the exam. Only the published question repositories will be opened for the instructor to choose from. If a question repository is published, but one question in the question repository is not published, the unpublished question cannot be selected. After a question is added into an exam, the instructor can edit, delete or view that question. In addition, the instructor can choose to edit the points for that question which is already added into the exam; otherwise, the default points for that question will be used (*Figure 32*).

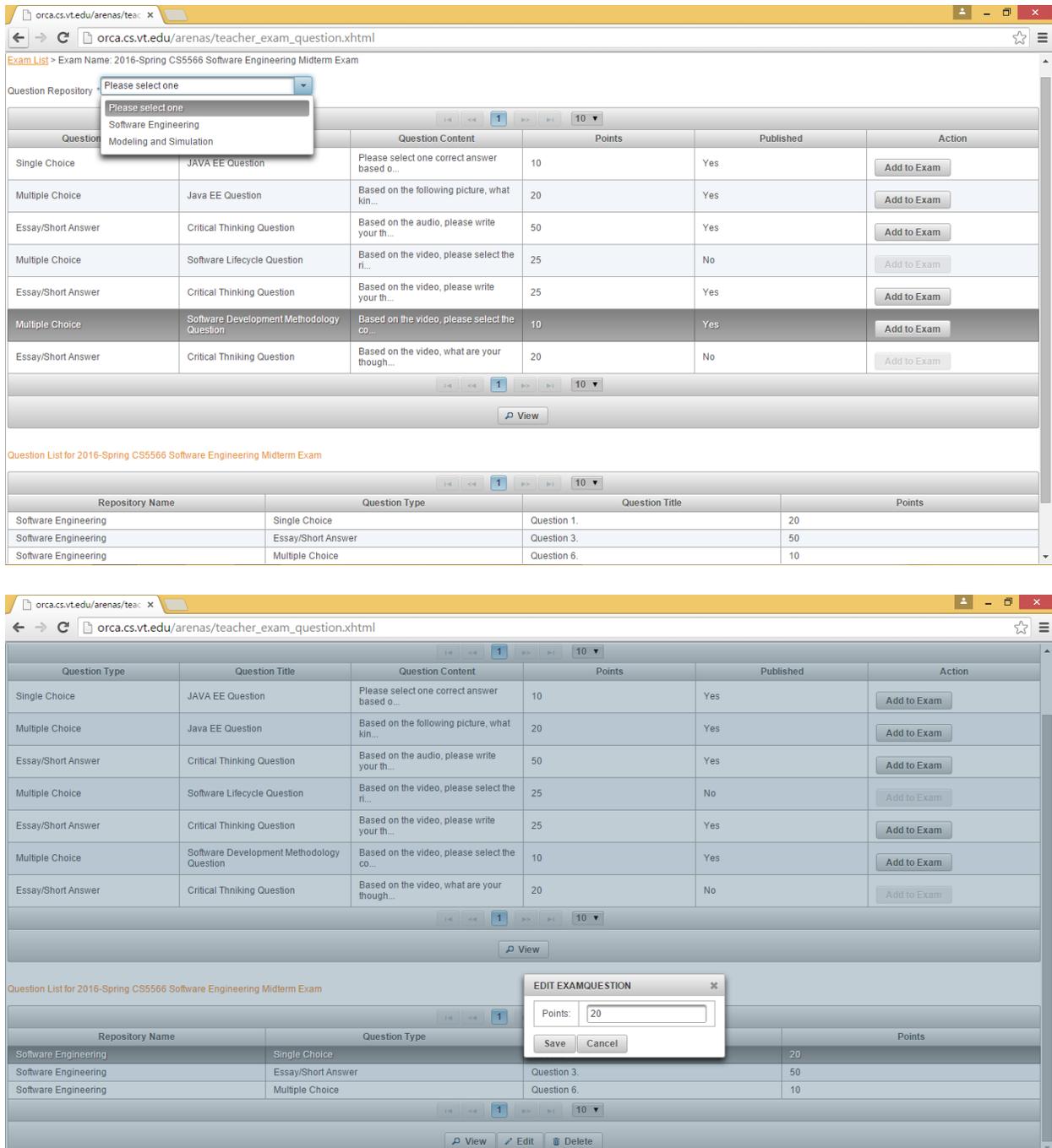


Figure 32. Exam Question Management Interface

## 5.6 Publish Exam

After an exam is created by an instructor, the instructor can choose to publish the exam to the existing student accounts or the randomly generated student accounts. When the instructor publishes an exam to the randomly generated student accounts, he or she can set the number and valid time for the generated student accounts. A student can use a generated account to log into the system to take an exam directly during the

valid time period without registering the system. A generated account will be locked beyond the valid time period (*Figure 33*).

The figure consists of three screenshots from a web browser showing the 'Publish Exam' interface. The browser address bar shows 'orca.cs.vt.edu/arenas/teacher\_exam\_publish.xhtml'.

**Top Screenshot: Exam List**  
 The interface shows a 'TEACHER MENU' on the left with options: Question Repository, Create Exam, Publish Exam, and Report Exam. The main area displays a table of exams:

Exam Name	Description	Begin Time	End Time	Action
2016-Spring CS5566 Software Engineering Midterm Exam	2016-Spring CS5566 Software Engineering Midterm Exam, close book, close notes, but open your mind	Fri Mar 18 2016 09:00:00 GMT-0500 (Central Daylight Time)	Fri Mar 18 2016 10:45:00 GMT-0500 (Central Daylight Time)	<a href="#">Publish</a>
2016 CS555 Software Engineering Quiz	2016 CS555 Software Engineering Quiz, open book, open notes	Thu Jan 07 2016 08:30:00 GMT-0600 (Central Standard Time)	Mon Feb 29 2016 23:59:00 GMT-0600 (Central Standard Time)	<a href="#">Publish</a>

**Middle Screenshot: Assign Exam**  
 The 'Exam List' dropdown is set to '2016-Spring CS5566 Software Engineering Midterm Exam'. The 'Exam-Taker Type' is set to 'Existed Exam-Taker Account'. A table lists existing students:

Username	Email	Firstname	Lastname	Majorfield	Generated	Action
student	student@vt.edu	Sun	Zhao	Computer Science	no	<a href="#">Assign to Take Exam</a>
tcongwu	tcongwu@vt.edu			CS	no	<a href="#">Assign to Take Exam</a>
student98	skyely@W126.com			CS	no	<a href="#">Assign to Take Exam</a>
student97	ske@126.com			CS	no	<a href="#">Assign to Take Exam</a>

**Bottom Screenshot: Generate Account**  
 The 'Exam-Taker Type' is set to 'Generate Exam-Taker Account'. The 'Input the number' field contains '2'. The 'Expired Time' is set to 'FEBRUARY 2016'. A calendar widget shows the month of February 2016. Below the calendar, there are sliders for 'Time', 'Hour', 'Minute', and 'Second'. The 'Generated Accounts' table is empty. The 'Generate Account' and 'Assign to Take Exam' buttons are visible.

User Name	Email	By Valid Time
student	student@vt.edu	
8aMkaVpOPA		Sat Mar 19 00:00:00 CDT 2016
student97	ske@126.com	

Figure 33. Publish Exam Interface

## 5.7 Report Exam

After a student user completes an exam, an exam report will be automatically generated in the system. An instructor can tweak the report and write a brief summary feedback in the report and publish it to the student. If the exam includes essay or short answer questions, it requires the instructor to manually score such type of questions (*Figure 34*).

The figure shows three screenshots of the 'Report Exam' interface in a web browser.

**Step 1: Exam List**  
 The browser address bar shows 'orca.cs.vt.edu/arenas/teacher\_exam\_report.html'. A 'TEACHER MENU' on the left includes 'Question Repository', 'Create Exam', 'Publish Exam', and 'Report Exam'. The main table lists exams:

Exam Name	Description	Begin Time	End Time	Action
2016-Spring CS5566 Software Engineering Midterm Exam	2016-Spring CS5566 Software Engineering Midterm Exam, close book, close notes, but open your mind	Fri Mar 18 2016 09:00:00 GMT-0500 (Central Daylight Time)	Fri Mar 18 2016 10:45:00 GMT-0500 (Central Daylight Time)	Report
2016 CS555 Software Engineering Quiz	2016 CS555 Software Engineering Quiz, open book, open notes	Thu Jan 07 2016 08:30:00 GMT-0600 (Central Standard Time)	Mon Feb 29 2016 23:59:00 GMT-0600 (Central Standard Time)	Report

**Step 2: Exam Report List**  
 The browser address bar shows 'orca.cs.vt.edu/arenas/teacher\_exam\_report\_list.html'. The page title is 'Exam List > Exam-Taker List for: 2016 CS555 Software Engineering Quiz'. The table lists exam takers:

User Name	Email	By Valid Time	Action
student	student@vt.edu		View report
X28rsUEICd		2016-02-29	View report
1BbKvLI87M		2016-02-28 23:59:59	View report
student97	ske@126.com		View report

**Step 3: Exam Report Detail**  
 The browser address bar shows 'orca.cs.vt.edu/arenas/teacher\_exam\_report\_list\_detail.html'. The page title is 'Exam Report List > Exam Report'. The report is for '2016 cs555 software engineering quiz' for user 'student'. It shows scores for different question types and a 'Show/Hide Detail' button.

1 (10 points) JAVA EE Question Please select one correct answer based on your understanding of the picture.

The diagram shows a multi-tier architecture:

- Client Tier (1):** Includes a Web Browser (HTML, XHTML, JavaScript, CSS, XML, Facelets) and a Java SE Application Client (Application Client Container).
- Network/Cloud:** Connects the Client Tier to the Server.
- Java EE Application Server:** Contains:
  - Web Tier (2):** Java EE Technologies.
  - Business Tier (3):** Java EE Technologies.
  - Data Source Tier (5):** Includes a Database.

Arrows indicate a 'Request' flow from the Web Tier to the Business Tier, and a 'Response' flow from the Business Tier back to the Web Tier. A bidirectional arrow connects the Business Tier and the Data Source Tier.

Multiple Choice Question Number of Correct Response: 0.0 out of 2.0 Score: 0.0 out of 30.0

Essay/Short Answer Question Score: 55.0 out of 75.0

Legend:

- A. Web Tier often indicates EJB container
- B. Web Tier often indicates Web container
- C. Tier and container are the same
- D. 2-Tier is better than 3-Tier for complicated software applications

Figure 34. Report Exam Interface

## 5.8 Take Exam

A student with a valid account can take an exam when the exam is open to him or her. When a student begins to take an exam, he or she can choose to hide or show the countdown timer. For answering any question in the exam, the student needs to click the “Confirm” button to submit his or her answer to that question. After the student completes all of questions in the exam, he or she needs to check a checkbox at the bottom of the webpage to indicate that he or she consents to submit the answers and the exam will be closed to him or her (*Figure 35*).

The screenshot displays the 'Take Exam' interface in a web browser. The top section shows a 'STUDENT MENU' with 'Exam' and 'Exam Report' options. Below this is a table listing exams:

Exam Name	Exam Description	Begin Time	End Time	Action
2016-Spring CS5566 Software Engineering Midterm Exam	2016-Spring CS5566 Software Engineering Midterm Exam, close book, close notes, but open your mind	2016-03-18 09:00:00	2016-03-18 10:45:00	Take Exam
2016 CS555 Software Engineering Quiz	2016 CS555 Software Engineering Quiz, open book, open notes	2016-01-07 08:30:00	2016-02-29 23:59:00	Take Exam
test	test	2016-01-17 00:00:00	2016-01-18 00:00:00	Submitted
2016 CS555 Software Engineering Quiz	2016 CS555 Software Engineering Quiz, open book, open notes	2016-01-07 08:30:00	2016-02-29 23:59:00	Take Exam

The second screenshot shows the 'Take Exam' page for the '2016 CS555 Software Engineering Quiz'. It includes a countdown timer (03 days 23:01:32) and a question: '1 (10 points). JAVA EE Question Please select one correct answer based on your understanding of the picture.' The question is accompanied by a diagram of a Java EE Application Server architecture:

The diagram illustrates the architecture of a Java EE Application Server. It is divided into four tiers:

- Client Tier (1):** Includes a Web Browser (HTML, XHTML, JavaScript, CSS, XML, Facelets) and a Java SE Application Client (Application Client Container).
- Web Tier (2):** Java EE Technologies.
- Business Tier (3):** Java EE Technologies.
- Data Source Tier (5):** Database.

The diagram shows a Request flow from the Client Tier to the Web Tier, and a Response flow from the Web Tier back to the Client Tier. The Web Tier and Business Tier are connected, and the Business Tier is connected to the Data Source Tier.

Below the diagram are four radio button options:

- A. Web Tier often indicates EJB container
- B. Web Tier often indicates Web container
- C. Tier and container are the same
- D. 2-Tier is better than 3-Tier for complicated software applications

The 'Confirm' button is highlighted. Below the question is another question: '6 (10 points). Software Development Methodology Question Based on the video, please select the correct answer for the question'. The question is accompanied by a video player showing a slide titled 'Agile Software Development - Part 2 - History'. The slide features a cartoon character and text: 'A collection of light weight software' (in a yellow box) and 'evolved' (in a blue box) with 'in the mid-1990s' (in a green box) above it. A red arrow points from the yellow box to the blue box, with the text 'development methods' below it.

Below the video are three radio button options:

- A. Agile development methodology is the best one for any software development.
- B. Agile development methodology has its limitations.
- C. Agile development is good for small-size software development

The 'Confirm' button is highlighted. At the bottom of the page, there is a checkbox for 'I consent to submit my exam' and a 'Submit' button.

Figure 35. Take Exam Interface

## 5.9 View Exam Report

After an exam report is published to a student, a student with a valid account can log into the system and view the exam report. The student can view the brief summary report at the top of the web page; he or she can also click the “Show/Hide Detail” button to see the detail feedback of each question in the report (*Figure 36*).

The screenshot displays the 'STUDENT MENU' on the left with 'Exam' and 'Exam Report' options. The main area shows a table of exams with columns for Exam Name, Exam Description, Begin time, End time, and Action. Below the table, the 'Exam Report > Exam Report' section shows a summary for the '2016 cs555 software engineering quiz'.

Exam Name	Exam Description	Begin time	End time	Action
2016-Spring CS5566 Software Engineering Midterm Exam	2016-Spring CS5566 Software Engineering Midterm Exam, close book, close notes, but open your mind	2016-03-18 09:00:00	2016-03-18 10:45:00	<a href="#">View report</a>
2016 CS555 Software Engineering Quiz	2016 CS555 Software Engineering Quiz, open book, open notes	2016-01-07 08:30:00	2016-02-29 23:59:00	<a href="#">View report</a>
test	test	2016-01-17 00:00:00	2016-01-18 00:00:00	<a href="#">View report</a>
2016 CS555 Software Engineering Quiz	2016 CS555 Software Engineering Quiz, open book, open notes	2016-01-07 08:30:00	2016-02-29 23:59:00	<a href="#">View report</a>

**Exam Report > Exam Report**

report of exam: 2016 cs555 software engineering quiz

User Name : student	Total Score : 55.0 out of 115.0	Submitted Time : 2016-01-16 20:11:38
Single Choice Question Number of Correct Response: 0.0 out of 1.0 Score:0.0 out of 10.0	Multiple Choice Question Number of Correct Response: 0.0 out of 2.0 Score:0.0 out of 30.0	Essay/Shor Answer Question Score:55.0 out of 75.0

[Show/Hide Detail](#)

**Show/Hide Detail**

1 (10 points) JAVA EE Question Please select one correct answer based on your understanding of the picture.

The diagram illustrates a Java EE Application Server architecture. On the left, the Client Tier includes a Web Browser (HTML, XHTML, JavaScript, CSS, XML, Facelets) and a Java SE Application Client (Application Client Container). These connect to the Java EE Application Server via Network/Cloud and HTTP Server. The server is divided into three tiers: Web Tier (Java EE Technologies), Business Tier (Java EE Technologies), and Data Source Tier (Database). Arrows indicate the flow of Request and Response between the client and server, and between the tiers.

- A. Web Tier often indicates EJB container
- B. Web Tier often indicates Web container
- C. Tier and container are the same
- D. 2-Tier is better than 3-Tier for complicated software applications

**Incorrect. The Answer is: B**

2 (10 points) Software Development Methodology Question Based on the video, please select the correct answer for the question

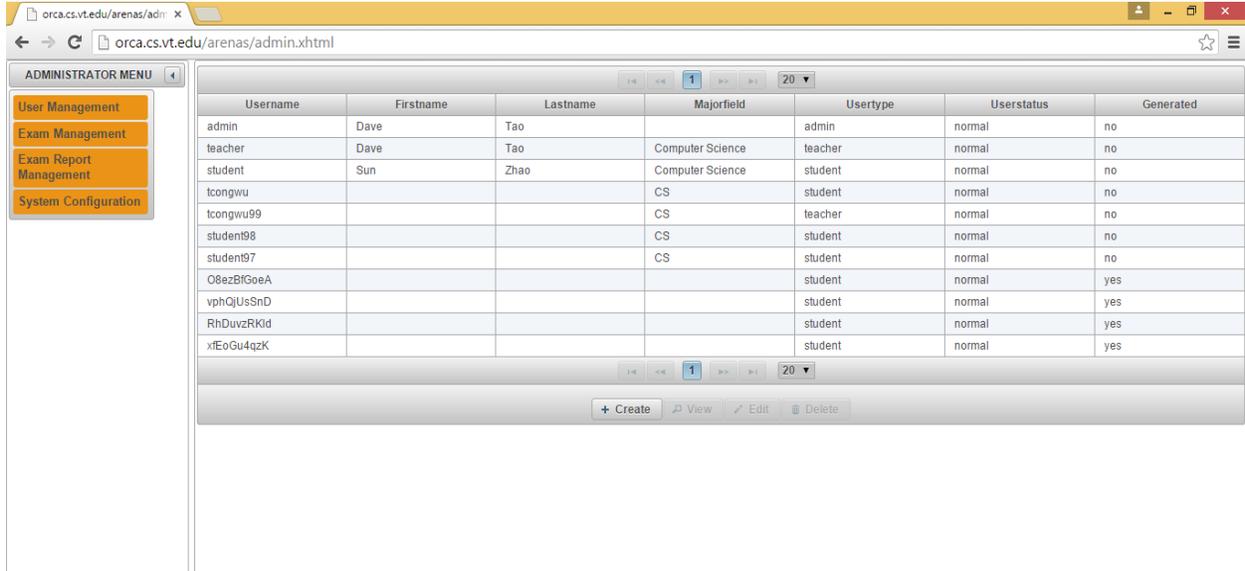
Agile Software Development - Part 2 - History

in the mid-1990s

Figure 36. View Exam Report Interface

## 5.10 Administrator Dashboard

An administrator can log into the system with a valid account and manage all of the user accounts, exams and exam reports in the system. The administrator can also configure the system in the dashboard (*Figure 37*).



The screenshot displays a web browser window with the URL `orca.cs.vt.edu/arenas/admin.xhtml`. On the left, there is an "ADMINISTRATOR MENU" with four options: "User Management", "Exam Management", "Exam Report Management", and "System Configuration". The main area features a table with the following columns: Username, Firstname, Lastname, Majorfield, Usertype, Userstatus, and Generated. The table contains 14 rows of user data. Below the table, there are pagination controls showing page 1 of 20, and action buttons for "+ Create", "View", "Edit", and "Delete".

Username	Firstname	Lastname	Majorfield	Usertype	Userstatus	Generated
admin	Dave	Tao		admin	normal	no
teacher	Dave	Tao	Computer Science	teacher	normal	no
student	Sun	Zhao	Computer Science	student	normal	no
tcongwu			CS	student	normal	no
tcongwu99			CS	teacher	normal	no
student98			CS	student	normal	no
student97			CS	student	normal	no
O8ezBKGoA				student	normal	yes
vphQUsSnD				student	normal	yes
RhDuvzRKId				student	normal	yes
xTEoGu4qzK				student	normal	yes

Figure 37. Administrator DashBoard Interface

## CHAPTER 6: Self-Evaluation of ARENAS

This chapter discusses a self-evaluation of ARENAS in terms of the following quality attributes: functionality, usability, scalability, reliability, performance, reusability and supportability [[Microsoft 2009](#); [Pressman 2010](#)].

### 6.1 Functionality

Functionality defines how well a system's features and capabilities meet users' needs and it can be assessed through evaluating the capabilities of the delivered program [[Pressman 2010](#)].

The capabilities of ARENAS were developed based on the requirements specification and the requirements were elicited from use case-based requirements engineering. Use case-based requirements engineering is considered as the most useful practice for creating a software requirements specification [[Balci 2014](#)]. During the process of use case-based requirements engineering, each use case was discussed and refined before it was identified. Before the development of ARENAS, a HTML-based demo was created to present the functionalities of ARENAS. Then the demo was discussed and refined until it demonstrated all of the required functions. After the demo was confirmed to display all of the required functionalities, the development of ARENAS began. Through such process, the developed functionalities of ARENAS could meet users' needs.

### 6.2 Usability

Usability describes how well the system meets the requirements of end users by being intuitive with an overall user-friendly experience. This attribute can be assessed in terms of aesthetics and user experience [[Pressman 2010](#)].

In order to make ARENAS easy to use, the ultimate UI Framework for Java EE – PrimeFaces (<http://primefaces.org/>) was utilized as the front end framework. PrimeFaces is regarded as an excellent lightweight library for creating rich user interfaces using Java; it also provides a comprehensive list of JSF components emphasizing simplicity, ease of use and performance. In addition, PrimeFaces provides a myriad of themes and layouts which fully support HTML5, CSS3 and cross-browsers.

Every Interface in ARENAS was created with the consistent layout and theme, which can help provide each interface with the same font size, webpage style, and color arrangement. Additionally, PrimeUI (<http://www.primefaces.org/primeui/>) was used in ARENAS for creating highly user interactive interfaces.

Through utilizing these front end techniques, ARENAS can provide end users with pleasant and ease of use experience.

### 6.3 Scalability

Scalability defines how well a system handles the increases of workload without impacting its system performance and how well the system can be enlarged [[Microsoft 2009](#)].

ARENAS was designed as a cloud-based software and was created with Java EE 7.0, running on the application server Glassfish 4.1 in CentOS 7.0 based on multi-tiered architecture. Using these technologies can fundamentally guarantee that this system has the ability to be scalable from a technical perspective.

In addition, open API is also provided in ARENAS. The system was created to have the ability to easily integrate a third-party application like course management system based on web service.

## 6.4 Reliability

Reliability defines how well the system can stay normal operational over time. It can be assessed as the likelihood that a system could perform its intended functions without failure over a specified time interval. In addition, the precise outputs of the system, the predictability of the system and the capability to recover from failure are also the features of reliability [Pressman 2010].

ARENAS was created using Java EE at the back end. Java EE provides Java Message Service (JMS) to help achieve the level of reliability [Antonio Goncalves 2013]. For instance, JMS messages are created to be *Persistent* by default in any transaction in Java EE. A transaction is defined as a unit consisting of a series of operations such as sending messages and receiving messages. If a transaction is not successful, a friendly message would be provided and lead the end users to try other possible actions. The reliability mechanisms in JMS include:

1. *Controlling message acknowledgement* – specifying various levels of control over message acknowledgement.
2. *Setting message priority levels* – various priority levels for messages can be configured.
3. *Allowing messages to expire* – specifying an expiration time for messages.
4. *Creating temporary destination* – setting temporary destinations that only last for the duration of that connection.

These techniques used in the development of ARENAS could guarantee the reliability of ARENAS to some extent.

## 6.5 Performance

Performance defines how well a system responds when executing any action within a given time interval. It can be measured in terms of latency or throughput. Latency is the time taken to respond to any event. Throughput is the number of events that take place within a given amount of time. Therefore, performance can be assessed through taking into account the processing speed, the response time, system resource consumption and efficiency [Pressman 2010].

To test the system performance, we simulate 2000 users to log into the system one time with JMeter (<http://jmeter.apache.org/>), the system runs smoothly. However, the test was held by the researcher and it was hard for the researcher to make unexpected actions happen. In addition, Glassfish4.1, MySQL5.6, ARENAS application and all of the related libraries are on the same CentOS Server; it is not a perfect system environment for testing the performance of ARENAS.

## 6.6 Reusability

Reusability defines how well the components and subsystems can be suitable for use in other applications and in other scenarios [Microsoft 2009].

ARENAS was decomposed and modularized based on the principle of maximizing cohesion

within each module and minimizing dependencies between modules [[Sommerville 2009](#)]. Five subsystem components were designed and developed in ARENAS. They were user component, question repository component, exam component, exam report component and configuration component. These five components were loosely coupled and could be reused by any other related applications or scenarios.

## **6.7 Supportability**

Supportability defines how well a system can be adapted and extended. ARENAS was designed as independent cloud-based software system and it could be easily extended or adapted with API. Also, the decomposed and modularized structure of ARENAS made it easy to be maintained [[Pressman 2010](#)].

In addition, the major programming language in this system is Java, which is an object-oriented and platform-independent programming language. The features of Java programming language contribute to easy adaptation of the system. Furthermore, JavaServer Faces technology was used in the development of the system. JavaServer Faces technology provides a clean separation between behavior and presentation for the system application. Other advantages of Javaserer Facelets benefiting supportability of the system include as follows [[Antonio Goncalves 2013](#)]:

1. Code can be reused and extended for components through the template and composite component features.
2. Helping automatically register the managed bean as a resource using the annotation feature.
3. Providing a rich architecture for managing component state, processing component data, validating user input, and handling events.

# CHAPTER 7: Conclusions and Future Research

## 7.1 Conclusions

This thesis presents the research of a cloud-based software system for online multimedia examinations. Currently, few of existing online assessment systems employ the cloud-based technology to provide a scalable software solution for online multimedia examinations. Our ARENAS utilizes cloud-computing technology based on the multi-tiered client-server architecture to provide a comprehensive software solution for large-scale online multimedia examinations.

The system was created based on the software developed lifecycle [Balci 2014]: problem formulation, requirements engineering, architecture design (high-level design), software components design and development (programming). The whole process was integrated with verification, validation and quality assurance activities.

The quality attributes of the final developed system are acceptable based on the self-evaluation. For instructors, the system is easy for them to create an online examination incorporating with multimedia elements like images, graphics, video and audio; it is also easy for them to assign the exam to their students whether these students have registered accounts in the system or not. For students, they can take the online exam anywhere, as long as they can connect the Internet and the exam period is still valid. The exam reports can be quickly generated by the system; the instructors can tweak the exam reports or add feedback before publishing the exam reports to their students; the students can easily view the exam reports after they are published. For administrators, they can easily manage and monitor the whole process.

## 7.2 Contributions

The contribution of this study is the cloud-based software solution for multimedia online examinations, utilizing the multi-tiered client-server architecture and cloud-based technology. This software solution can help allow instructors to effectively design engaging online questions with multimedia elements, provide a large-scale online assessment, and afford more user-friendly online teaching or learning experience for instructors and students.

In addition, the design and development process of ARENAS can be a reference to designing and developing other large-scale cloud-based educational software systems for educators and researchers.

## 7.3 Future Development and Research

This study concentrated on the design and development of a cloud-based software system for online multimedia examinations. Future development work may include:

1. Providing a function to collect and visualize student data. For example, visualizing how much time a student would spend on each question in an online examination.
2. Providing more question types for instructors to use when they create questions for an online examination.
3. Supporting more file-extensions of multimedia elements for instructors to use when they create questions for an online examination.

4. Improving the interface by providing different themes to allow users to select the most appropriate theme for them.
5. Improving the functionality of creating online questions by allowing instructors to use more third-party video-provider platform like Vimeo (<http://www.vimeo.com>).
6. Improving the functionality of managing assessment criteria through allowing instructors to set flexible assessment criteria like assessment rubrics for short-answer or essay questions.
7. Integrating the third-party tools such as Respondus LockDown Browser and Respondus Monitor from <https://www.respondus.com/> into the system for helping reduce or preclude the probability of student online exam cheating.

In addition to the development work, involving more instructors and students to use ARENAS could provide useful information.

## REFERENCES

- Aleti, A., Buhnova, B., Grunske, L., Koziolok, A., and Meedeniya, I. (2013). Software architecture optimization methods: A systematic literature review. *Software Engineering, IEEE Transactions*, 39(5), 658-683.
- Antonio Goncalves (2013) *Beginning Java EE 7*, Apress, New York, NY
- Balci, O. (2014), "CS5704 Software Engineering Course Website," Department of Computer Science, Virginia Tech, Blacksburg, VA, <http://manta.cs.vt.edu/cs5704>
- Balci, O. (2015), "CS4984 Cloud Software Development Course Website," Department of Computer Science, Virginia Tech, Blacksburg, VA, <http://manta.cs.vt.edu/cs4984/>
- Bibinagar, N., and Kim, W. J. (2013). Switched Ethernet-based real-time networked control system with multiple-client-server architecture. *Mechatronics, IEEE/ASME Transactions* 18(1), 104-112.
- ClassMarker (2015), "ClassMarker website," ClassMarker Pty Ltd, <http://www.classmarker.com>
- Cluskey Jr, G. R., Ehlen, C. R., and Raiborn, M. H. (2011). Thwarting online exam cheating without proctor supervision. *Journal of Academic and Business Ethics*, 4, 1-7.
- DoDAF (2010a), "DoD Architecture Framework Version 2.02 Volume 1: Introduction, Overview, and Concepts," Architecture Framework Working Group, Department of Defense, Washington, D.C.
- DoDAF (2010b), "DoD Architecture Framework Version 2.02 Volume 2: Architectural Data and Models," Architecture Framework Working Group, Department of Defense, Washington, D.C.
- DoDAF (2010c), "DoD Architecture Framework Version 2.02 Volume 3: DoDAF Meta-model Physical Exchange Specification - Developer's Guide," Architecture Framework Working Group, Washington, DC.
- Emary, I. M. E., and Al Sondos, J. A. (2006). An online website for tutoring and e-examination of economic course. *American Journal of Applied Sciences*, 3(2), 1715.
- ExamBuilder (2015), "ExamBuilder website," ExamBuilder Company, <http://www.exambuilder.com>
- ExamSoft (2015), "ExamSoft website," ExamSoft Worldwide, Inc., <http://www.examssoft.com>
- Fluck, A., Pullen, D., and Harper, C. (2009). Case study of a computer based examination system. *Australasian Journal of Educational Technology*, 25(4), 509-523.
- Hang, B. (2011). The design and implementation of on-line examination system. In *Proceedings of 2011 International Symposium on Computer Science and Society* (pp. 227-230). IEEE.
- Hao, Y. (2010). "Does multimedia help students answer test items?" *Computers in Human Behavior*, 26(5), 1149-1157.
- Hlaing, S. Z. (2009). An Authenticated Paradigm for Mobile Agent System in Online Examination. In *Proceedings of International Conference on Computer Engineering and Technology* (Vol. 2, pp. 420-424). IEEE.
- Hua P., Shu Y., and Bian L. (2008), A Web Services Based Online Training and Exam System. In *Proceedings of the IEEE International Conference on Wireless Communications Networking and Mobile Computing*, Dalian, China, pp. 1-4.
- IBM (2007) "Using UML service components to represent the SOA architecture pattern," <http://www.ibm.com/developerworks/architecture/library/ar-logsoa/> Armonk, New York

- Ikwueze, L. (2014) "Do Exam Policies Matter in College?" *Creative Education*, 5, 177-184. doi:10.4236/ce.2014.54027.
- Kwok, Y. R. (2012). *Peer-to-peer computing: Applications, architecture, protocols, and challenges*. Boca Raton, FL: Taylor & Francis.
- Li, G., Muthusamy, V., and Jacobsen, H. A. (2010). A distributed service-oriented architecture for business process execution. *ACM Transactions on the Web (TWEB)*, 4(1), 2
- Lu, Y. C., Yang, Y. S., Chang, P. C., and Yang, C. S. (2013). The design and implementation of intelligent assessment management system. In *Proceedings of 2013 Global Engineering Education Conference (EDUCON)*, (pp. 451-457). IEEE.
- Jen, L. R., and Lee, Y. J. (2000). Working Group. "IEEE recommended practice for architectural description of software-intensive systems." In *2000 IEEE Architecture*.
- McGough, J., Mortensen, J., Johnson, J., and Fadali, S. (2001). A web-based testing system with dynamic question generation. In *Proceedings of 31st Annual Frontiers in Education Conference*. (Vol. 3, pp. S3C-23). IEEE.
- McGovern, J., Ed. (2004). *A practical guide to enterprise architecture*. Prentice Hall Professional.
- Microsoft (2009) "Microsoft Application Architecture Guide," Microsoft Corporation, Redmond, WA <https://msdn.microsoft.com/en-us/library/ff650706.aspx>
- Microsoft (2014) ".NET Technology Guidance," Microsoft Corporation, Redmond, WA <https://blogs.msdn.microsoft.com/dotnet/2014/12/04/introducing-net-core/>
- Object Management Group (OMG, 2011) — "OMG Unified Modeling Language Infrastructure Version 2.4.1" <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>
- Oracle (2015), "Java EE at a Glance," Oracle Corporation, Redwood Shores, CA, <http://www.oracle.com/technetwork/java/javaee/overview/>
- Pressman, R. (2010), *Software Engineering, A Practitioner's Approach, 7th edition*, McGraw- Hill, New York, NY
- Rashad, M. Z., Kandil, M. S., Hassan, A. E., and Zaher, M. A. (2010). An Arabic web-based exam management system. *International Journal of Electrical & Computer Sciences IJECS-IJENS*, 10(01), 48-55.
- Rodchua, S., Yiadom-Boakye, M. G., and Woolsey, R. (2011). Student Verification System for Online Assessments: Bolstering Quality and Integrity of Distance Learning. *Journal of Industrial Technology*, 27(3), 1-8.
- Rosen, M., Lublinsky, B., Smith, K. T., and Balcer, M. J. (2012). *Applied SOA: service-oriented architecture and design strategies*. John Wiley & Sons.
- Roy, U. and Das, S. (2012). Agile Software Development Methodology and its Cost Estimation Technique. *National Journal of System and Information Technology*, 4(2), 209-225.
- Sommerville, I. (2009). *Software engineering*. Boston: Pearson/Addison-Wesley.
- Sung, Y. T., Chang, K. E., Chiou, S. K., and Hou, H. T. (2005). The design and application of a web-based self-and peer-assessment system. *Computers & Education*, 45(2), 187-202.
- Tegarden, D. P., Dennis, A., and Wixom, B. H. (2012). *Systems Analysis and Design with UML*. John Wiley & Sons, Inc.

- Temitayo, M. F., Adebisi, A. A., and Alice, O. O. (2013). Computer-Based Test (CBT) System for University Academic Enterprise Examination. *International Journal of Scientific & Technology Research*, 2(8).
- Vu, Q. H., Lupu, M., and Ooi, B. C. (2010). *Peer-to-peer computing: Principles and applications*. Heidelberg; New York: Springer.
- Wang, L. (2008). Developing and evaluating an interactive multimedia instructional tool: Learning outcomes and user experiences of optometry students. *Journal of Educational Multimedia and Hypermedia*, 17, 43-58.
- Weilkiens, T. (2011). *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Morgan Kaufmann.
- Yağci, M., and Ünal, M. (2014). Designing and Implementing an Adaptive Online Examination System. *Procedia-Social and Behavioral Sciences*, 116, 3079-3083.
- Yuan, Z., Liang, Z., and Guohua, Z. (2003). A novel web-based online examination system for computer science education. In *Proceedings of 2013 IEEE Frontiers in Education Conference (FIE)* (Vol. 3, pp. S3F7-10). IEEE.