VolunteerCloud: A Cloud-Based Software Application for Volunteer Recruitment

Andres Pico

Report 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

Stephen H. Edwards

Calvin J. Ribbens

May 11, 2018

Blacksburg, Virginia

*Keywords and phrases:* Software engineering, cloud software application, requirements engineering, model-view-controller architecture, client-server architecture, web services, volunteer recruitment, webchat

VolunteerCloud: A Cloud-Based Software Application for Volunteer Recruitment

Andres Pico

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 and get fast exam results, save their time, and also help instructors effectively collect students’ exam answers, get the exam report, as well as reduce cost and save trees for our world.

Multimedia elements like image, graphic, 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 enjoyable learning experiences for learners. However, integrating multimedia elements into online examination systems is rarely reported. In addition, multimedia elements usually 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 system is becoming more attractive due to its 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 and reported.

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 some specific domain or application area with short of multimedia elements. There is a lack of a comprehensive software solution for supporting multimedia elements for presenting the online questions and providing a myriad of question types, flexible accounts to the exam-takers, randomized order in an online exam, flexible grading mechanisms, and analytical exam reports.

This thesis tries to utilize the “Software-as-a-Service (SaaS) paradigm, design and develop a cloud-b**A**sed softwa**R**e syst**E**m for o**N**line multimedi**A** examination**S** (**ARENAS**), and explore a comprehensive software solution for the online assessment field. ARENAS employs a multi-tiered client-server architecture, and it includes five subsystem modules: user module, question repository module, exam module, exam report module and configuration module. The developed cloud-based software system can help instructors design more engaging online questions and help provide more user-friendly experience for the exam-takers; the design and development process of ARENAS can be taken as a reference to designing and developing other large-scale cloud-based educational software systems for educators and researchers.

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 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**

[ABSTRACT ii](#_Toc440833468)

[ACKNOWLEDGMENTS iii](#_Toc440833469)

[LIST OF FIGURES vi](#_Toc440833470)

[LIST OF TABLES vii](#_Toc440833471)

[LIST OF ACRONYMS viii](#_Toc440833472)

[CHAPTER 1: Introduction 1](#_Toc440833473)

[1.1 Related Work 1](#_Toc440833474)

[1.2 Statement of the Problem 3](#_Toc440833475)

[1.3 Statement of Objectives 3](#_Toc440833476)

[1.4 Overview of Thesis 3](#_Toc440833477)

[CHAPTER 2: ARENAS Requirements Specification 4](#_Toc440833478)

[2.1 Use Case Diagrams 4](#_Toc440833479)

[2.2 Use Cases and Functional Requirements 14](#_Toc440833480)

[2.2.1 Use Case 1: Create an Account **Error! Bookmark not defined.**](#_Toc440833481)

[2.2.1.1 Use Case 1 Documentation **Error! Bookmark not defined.**](#_Toc440833482)

[2.2.1.2 Functional Requirements Associated with Use Case 1 **Error! Bookmark not defined.**](#_Toc440833483)

[2.2.2 Use Case 2: Create a Question Repository 17](#_Toc440833484)

[2.2.2.1 Use Case 2 Documentation 17](#_Toc440833485)

[2.2.2.2 Functional Requirements Associated with Use Case 2 19](#_Toc440833486)

[2.2.3 Use Case 3: Create an Exam 19](#_Toc440833487)

[2.2.3.1 Use Case 3 Documentation 19](#_Toc440833488)

[2.2.3.2 Functional Requirements Associated with Use Case 3 20](#_Toc440833489)

[2.2.4 Use Case 4: Publish an Exam 20](#_Toc440833490)

[2.2.4.1 Use Case 4 Documentation 20](#_Toc440833491)

[2.2.4.2 Functional Requirements Associated with Use Case 4 21](#_Toc440833492)

[2.2.5 Use Case 5: Generate an Exam Report 22](#_Toc440833493)

[2.2.5.1 Use Case 5 Documentation 22](#_Toc440833494)

[2.2.5.2 Functional Requirements Associated with Use Case 5 23](#_Toc440833495)

[2.2.6 Use Case 6: Take an Exam 24](#_Toc440833496)

[2.2.6.1 Use Case 6 Documentation 24](#_Toc440833497)

[2.2.6.2 Functional Requirements Associated with Use Case 6 25](#_Toc440833498)

[2.2.7 Use Case 7: View Exam Report 25](#_Toc440833499)

[2.2.7.1 Use Case 7 Documentation 25](#_Toc440833500)

[2.2.7.2 Functional Requirements Associated with Use Case 7 26](#_Toc440833501)

[2.2.8 Use Case 8: Administrator Dashboard 26](#_Toc440833502)

[2.2.8.1 Use Case 8 Documentation 26](#_Toc440833503)

[2.2.8.2 Functional Requirements Associated with Use Case 8 27](#_Toc440833504)

[2.3 Non-Functional Requirements 32](#_Toc440833505)

[CHAPTER 3: ARENAS Architecture Specification 34](#_Toc440833506)

[3.1 Client-Server Architecture 34](#_Toc440833507)

[3.2 Peer-to-Peer Architecture 35](#_Toc440833508)

[3.3 Service-Oriented Architecture 36](#_Toc440833509)

[3.4 Java EE-based Client-Server Architecture 37](#_Toc440833510)

[3.5 ARENAS Architecture Model Based on DODAF 38](#_Toc440833511)

[3.5.1 OV-2: Operational Resource Flow Description 38](#_Toc440833512)

[3.5.2 SV-4: Systems Functionality Description 39](#_Toc440833513)

[CHAPTER 4: ARENAS Design Specification 41](#_Toc440833514)

[4.1 Class Diagrams 41](#_Toc440833515)

[4.1.1 ARENAS Class Diagram 41](#_Toc440833516)

[4.2 Package Diagrams 42](#_Toc440833517)

[4.2.1 User-Exam-Database Package Diagram 42](#_Toc440833518)

[4.3 Component Diagrams 43](#_Toc440833519)

[4.3.1 Exam-Question Component Diagram 43](#_Toc440833520)

[4.4 Statechart Diagrams 44](#_Toc440833521)

[4.4.1 Instructor Statechart Diagram 44](#_Toc440833522)

[4.4.2 Student Statechart Diagram 45](#_Toc440833523)

[4.5 Activity Diagrams 46](#_Toc440833524)

[4.5.1 Instructor and Student Activity Diagram 46](#_Toc440833525)

[4.6 Sequence Diagrams 47](#_Toc440833526)

[4.6.1 Exam Creation Sequence Diagram 47](#_Toc440833527)

[4.6.2 Exam Taking Sequence Diagram 47](#_Toc440833528)

[4.7 Collaboration Diagrams 48](#_Toc440833529)

[4.7.1 Exam Creation Collaboration Diagram 48](#_Toc440833530)

[4.7.2 Exam Taking Collaboration Diagram 49](#_Toc440833531)

[4.8 Deployment Diagrams 49](#_Toc440833532)

[4.8.1 ARENAS Deployment Diagram 49](#_Toc440833533)

[CHAPTER 5: ARENAS Functionality 51](#_Toc440833534)

[5.1 User Registration 51](#_Toc440833535)

[5.2 Question Repository 52](#_Toc440833537)

[5.3 Question Management 52](#_Toc440833539)

[5.4 Exam Creation 53](#_Toc440833544)

[5.5 Exam Question Management 54](#_Toc440833545)

[5.6 Publish Exam 55](#_Toc440833546)

[5.7 Report Exam 57](#_Toc440833547)

[5.8 Take Exam 57](#_Toc440833548)

[5.9 View Exam Report 59](#_Toc440833549)

[5.10 Administrator Dashboard 60](#_Toc440833550)

[CHAPTER 6: Self-Evaluation of ARENAS 61](#_Toc440833551)

[6.1 Functionality 61](#_Toc440833552)

[6.2 Usability 61](#_Toc440833553)

[6.3 Scalability 61](#_Toc440833554)

[6.4 Reliability 62](#_Toc440833555)

[6.5 Performance 62](#_Toc440833556)

[6.6 Reusability 62](#_Toc440833557)

[6.7 Supportability 63](#_Toc440833558)

[CHAPTER 7: Conclusions and Future Research 64](#_Toc440833559)

[7.1 Conclusions 64](#_Toc440833560)

[7.2 Contributions 64](#_Toc440833561)

[7.3 Future Development and Research 64](#_Toc440833562)

[REFERENCES 66](#_Toc440833563)

LIST OF FIGURES

[Figure 1. ARENAS Use Case Diagram 5](#_Toc440833620)

[Figure 2. Account Use Case Diagram 6](#_Toc440833621)

[Figure 3. Question Repository Use Case Diagram 7](#_Toc440833622)

[Figure 4. Exam Creation Use Case Diagram 8](#_Toc440833623)

[Figure 5. Exam Publishing Use Case Diagram 9](#_Toc440833624)

[Figure 6. Exam Report Generation Use Case Diagram 10](#_Toc440833625)

[Figure 7. Taking an Exam Use Case Diagram 11](#_Toc440833626)

[Figure 8. Exam Report Viewing Use Case Diagram 12](#_Toc440833627)

[Figure 9. Administrator Dashboard Use Case Diagram 13](#_Toc440833628)

[Figure 10. A two-tier Client-Server Architecture 34](#_Toc440833629)

[Figure 11. A multi-tiered Client-Server Architecture 35](#_Toc440833630)

[Figure 12. A Peer-to-Peer Architecture 36](#_Toc440833631)

[Figure 13. A Service-Oriented Architecture 37](#_Toc440833632)

[Figure 14. A Java EE-based Client-Server Architecture 38](#_Toc440833633)

[Figure 15. OV-2: Operational Resource Flow Description 39](#_Toc440833634)

[Figure 16. SV-4: Systems Functionality Description 40](#_Toc440833635)

[Figure 17. ARENAS Class Diagram 42](#_Toc440833636)

[Figure 18. User-Exam-Database Package Diagram 43](#_Toc440833637)

[Figure 19. Exam-Question Component Diagram 44](#_Toc440833638)

[Figure 20. Instructor Statchart Diagram 45](#_Toc440833639)

[Figure 21. Student Statchart Diagram 45](#_Toc440833640)

[Figure 22. Instructor and Student Activity Diagram 46](#_Toc440833641)

[Figure 23. Exam Creation Sequence Diagram 47](#_Toc440833642)

[Figure 24. Exam Taking Sequence Diagram 48](#_Toc440833643)

[Figure 25. Exam Creation Collaboration Diagram 48](#_Toc440833644)

[Figure 26. Exam Taking Collaboration Diagram 49](#_Toc440833645)

[Figure 27. ARENAS Deployment Diagram 50](#_Toc440833646)

[Figure 28. User Registeration Interface 51](#_Toc440833647)

[Figure 29. Question Repository Interface 52](#_Toc440833648)

[Figure 30. Question Management Interface 53](#_Toc440833649)

[Figure 31. Exam Creation Interface 54](#_Toc440833650)

[Figure 32. Exam Question Management Interface 55](#_Toc440833651)

[Figure 33. Publish Exam Interface 56](#_Toc440833652)

[Figure 34. Report Exam Interface 57](#_Toc440833653)

[Figure 35. Take Exam Interface 58](#_Toc440833654)

[Figure 36. View Exam Report Interface 59](#_Toc440833655)

[Figure 37. Administrator DashBoard Interface 60](#_Toc440833656)

LIST OF TABLES

[Table 1. Use Case 1 Documentation **Error! Bookmark not defined.**](#_Toc440833660)

[Table 2.Use Case 2 Documentation 17](#_Toc440833661)

[Table 3. Use Case 3 Documentation 19](#_Toc440833662)

[Table 4. Use Case 4 Documentation 20](#_Toc440833663)

[Table 5. Use Case 5 Documentation 22](#_Toc440833664)

[Table 6. Use Case 6 Documentation 24](#_Toc440833665)

[Table 7. Use Case 7 Documentation 25](#_Toc440833666)

[Table 8. Use Case 8 Documentation 26](#_Toc440833667)

LIST OF ACRONYMS

API Application Programming Interface

CSA Client-Server Architecture

CSS Cascading Style Sheets

EJB Enterprise JavaBean

GUI Graphical User Interface

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

JDK JAVA Development Kit

JSF JavaServer Faces

JSON JavaScript Object Notation

JSP JavaServer Pages

JRE JAVA Runtime Environment

MVC Model-View-Controller Architecture

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

VC A Cloud-Based Software Application for Volunteer Recruitment

VolunteerCloud A Cloud-Based Software Application for Volunteer Recruitment

VM VolunteerMatch

XHTML Extensible Hypertext Markup Language

XML Extensible Markup Language

# Introduction

This chapter looks at the new and emerging technologies and approaches for building and consuming services and applications that are hosted remotely. These kinds of services and applications are accessed over the Internet and run in what is usually termed the cloud, hence the commonly used description cloud computing. Cloud solution hosters and vendors will generally provide prebuilt service applications with varying levels of configurability and customization. Alternatively, you may prefer to create your own application in-house and host it either internally on your own systems, or externally in the cloud at a hosting provider. [Microsoft 2009]

**Software as a Service (SaaS)**. Applications that perform comprehensive business tasks, or accomplish business services, and allow you to consume them as services with no internal application requirements other than composition and UI. [Microsoft 2009] (Definition of web services provided in chapter 25)

An examination is an assessment of someone’s knowledge, skill, or aptitude in a particular subject matter [[Ikwueze 2014](#Ikwueze)]. 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 are beneficial for the students to get fast exam results, so as to save their time from long waiting which would have been the case on 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 exists a serious challenge for academic integrity. It is hard to invigilate an online exam and minimize cheating in the environment of online examination [[Rodchua, Yiadom-Boakye, and Woolsey 2011](#Rodchua2011)]. Some institutions ask proctors to supervise online exams. Yet, proctoring in online exams can be expensive. For example, the costs may include the salaries of staff for administrating the test center, the wages for the proctors, the time to find a qualified proctor and the effort of arranging a time for an exam in the test center [[Cluskey, Ehlen and Raiborn 2011](#Cluskey)]. In addition, proctoring in online exams is not practical for all remote students. Therefore, minimizing cheating is one aspect which needs to be considered carefully when implementing an online examination system.

Multimedia elements like image, graphic, video and audio have been widely integrated into online learning environments; they help provide more interactive and enjoyable learning experiences [[Wang 2008](#Wang2008)]. Online assessment is a part of online learning experience, and online examinations can be more interactive and fun with incorporating multimedia elements [[Hao 2010](#Hao2010)]. In fact, online assessment is inseparable from teaching and learning, and creating online examinations which improve students’ learning experience should be an important goal for educators and researchers.

However, few researches put efforts into online multimedia examination systems based on literature review. Most of existing online examination applications focus on copying the styles of paper-based exams; it is hard to take some steps to improve the user experiences of those systems.

Software paradigm has been shifting during the last decade from “Software-as-a-Product (SaaP)” to “Software-as-a-Service (SaaS)” provided “in the cloud” [[Balci 2014](#Balci2014)]. Cloud computing is becoming a more attractive technology due to its dynamic scalability and effective usage of resources, yet how cloud computing technologies impacting online examinations is not significantly reported.

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

## 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 testing. 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]](#McGough2001) proposed a web-based testing system with high level support for advanced mathematics typesetting and equation parsing. [Yuan et al. [2003]](#Yuan2003) developed a web-based online examination system with automatic grading for objective questions. This system was successfully applied to the distance evaluation of the basic knowledge of computer science courses for the students in the local universities. [Sung et al. [2005]](#Sung2005) designed and developed a web-based self- and peer-assessment system for facilitating instructors to arrange various self- and peer-assessment procedures. [He [2006]](#He2006) presented a web-based educational assessment system by applying Bloom’s taxonomy to evaluate students’ learning outcomes. The performance of this web-based system was encouraging with experimentation in science and mathematics courses at two local high schools. [Emary and Abu [2006]](#Emary) developed an online system for tutoring and e-examination of economics courses, and it included two major modules: the first one was an online website to review and make self-assessment for all materials of economics course; the second was an online examination with a large question bank, through it the level of students could be assessed instantly and some statistical evaluations could be acquired. [Hua, Shu and Bian [2008]](#Hua2008) proposed an online training and exam system based on web-service technologies without discussing the question types. [Hlaing [2009]](#Hlaing2009) presented a framework to allow students to carry out online assessments via a mobile user agent, with emphasis on the authenticated paradigm using public key and digital signature for the mobile agents. [Fluck, Pullen and Harper [2009]](#Fluck2009) 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 by using the online examination function for assessing objective questions. Some researchers also explored developing online examination systems for general assessment purpose. For instance, [Rashad et al. [2010]](#Rashad2010) developed an Arabic web-based exam management system using open source technologies such as PHP and MYSQL for mass educational evaluation. This system included automatic grading function, exam administration function and exam report function, as well as 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, but limited multimedia elements were adopted in presenting the online questions. [Hang [2011]](#Hang2011) presented a preliminary campus-based online examination system which included instructor module, student module and administrator module, but it did not discuss the detail of question types and lacked the evaluation of the system. [Temitayo, Adebisi and Alice [2013]](#Temitayo2013) developed a computer-based examination system for Nigerian Universities using Visual Studio 2012. This system was successfully employed by several institutions, but the question types were limited. [Lu et al. [2013]](#Lu2013) designed and developed an intelligent assessment management system for the summative assessment of college courses. But it had a hybrid structure for integrating virtual online assessment management system and in-classroom examinations. [Yağci and Ünal [2014]](#Yağci2014) developed an adaptive online examination system based on measuring student’s competence level with PHP programming language for computer education. This system was successfully applied at Ahi Evran University in Turkey, but the question types and multimedia elements in presenting questions were inadequate.

There are also some commercial online examination systems such as [[ExamSoft 2015](#ExamSoft2015); [Exambuilder 2015](#ExamBuilder2015); [ClassMarker 2015](#ClassMarker2015)] and so on. Examsoft emphasizes exam delivery and management and it cooperates with institutions rather than providing service to instructors and students directly. ClassMarker is a web-based testing system for business and education; it is a quiz maker, which allows users to create an online exam with setting time limits, public or private test access, randomization of question order, instant feedback and different question types such as multiple choice, true or false, free text, grammar and essay. Exambuilder provides online test authoring and delivery, which helps generate real-time scoring results, instantly identify training gaps, and easily integrate learning management systems (LMSs). However, these commercial online exam systems do not pay much attention to the multimedia elements in an online exam and have limited functionalities for all remote users.

Although a variety of online-assessment tools have been developed, they just focus on the assessment in some specific domain or application area with limited and general functionalities, and yet there is a lack of a comprehensive software solution for supporting multimedia elements for presenting the online questions, a myriad of question types, flexible accounts to the exam-takers, randomized order in an online exam, flexible grading mechanisms, and analytical exam reports.

## Statement of the Problem

The multimedia elements have been integrated into the online teaching and learning for several years, however, fully integrating multimedia elements into the online examinations is not reported much. A lot of online examination systems still present questions with monotonous styles, and support limited question types. In addition, few existing online examination system support flexible accounts to the exam-takers, randomized order for the same question types in an online exam, flexible grading mechanisms, and constructive exam reports. Most of the existing online-assessment systems 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, and provide more friendly user experience for instructors and students.

(Cite how you’ll use APIs and what do they do)

## Statement of Objectives

The objectives of the research include the following:

1. Investigate what functionalities are necessary for ARENAS.
2. Explore what kinds of architecture could be the most appropriate for ARENAS.
3. Develop the system for instructors, students and administrators.
4. Enable instructors to effectively create an engaging online exam with multimedia elements, and get fast exam report.
5. Engage students to take an online exam and get fast exam results with friendly user experience anywhere, anytime.
6. Provide comprehensive exam reports to the administrators for helping evaluate the teaching and learning process.

## 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 AREANS functionality; Chapter 6 presents the self-evaluation of AREANS; and Chapter 7 states the concluding remarks, contributions and future work.

# VolunteerCloud Requirements Specification

**Intro(Delete)**

Requirements gathering is one of the most essential tasks when developing a software system, since they are used to describe what the system must or mustn’t do. However, truly understanding the requirements of a problem is among the most difficult tasks that face a software engineer. When you first think about it, developing a clear understanding of requirements doesn’t seem that hard. After all, doesn’t the customer know what is required? Shouldn’t the end users have a good understanding of the features and functions that will provide benefit? Surprisingly, in many instances the answer to these questions is “no.” And even if customers and end users are explicit in their needs, those needs will change throughout the project. [[Pressman 2014](#Pressman2014)]

Use cases help us understand and clarify the users’ required interactions with the system and can reveal most, if not all, functional requirements of the new system [[Dennis 2012](#Dennis2012)]. This is why despite the complexity involved in requirements gathering, use case-based requirements engineering has proven to be the best practice:

* A Use Case represents a small amount of work the software system is required to perform. Thus, decomposing a complex software system functionality into Use Cases enables the modularization needed to overcome the complexity. [[Balci 2016](#Balci2016)]
* Identifying the “real” functional requirements is always challenging for complex software systems. A Use Case describes an interaction, and based on that description, “real” functional requirements can be more successfully identified and associated with that Use Case. [[Balci 2016](#Balci2016)]
* Listing requirements one after the other, even in different categories, does not provide any help for transitioning from requirements to software system design. On the other hand, Use Cases turn themselves into classes in an object-oriented design and significantly facilitate the transition. [[Balci 2016](#Balci2016)]

## Use Case Diagrams

A Use Case specifies the behavior of a system or a part of a system and is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor. [[Balci 2016](#Balci2016)]

*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) create a question repository; 2) create an exam; 3) publish an exam; and 4) report an exam. For each major function, there are some sub-functions. For example, an instructor can add, delete, update or view a question in a question repository after he or she has created the question repository; the instructor can select a question from the question repository for creating an exam; the instructor can assign the exam to a bunch of registering students or non-registering student; and the instructor and 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 exam statistics and view exam report statistics.

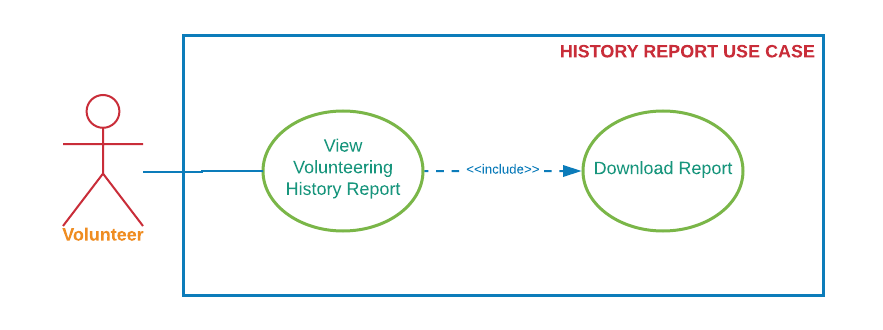




Figure 1. ARENAS Use Case Diagram

*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 to 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 question in the question repository, he or she can query the question 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 to the exam through selecting a question in the question repository which he or she has created earlier, update a question in the exam, delete a question from the exam and 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. The instructor can assign the exam to the registering student accounts and email these students. The instructor can also generate a bunch of temporal student accounts and assign the exam to these temporal 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 statistics information about the exam and write a brief summary for the exam report. The instructor can 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 when he or she is taking an exam.



Figure 7. Taking an Exam Use Case Diagram

*Figure 8* shows the exam report viewing 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. Exam Report Viewing 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 view exam statistical information or exam report statistical information.



Figure 9. Administrator Dashboard Use Case Diagram

## Use Cases and Functional Requirements

Functional requirements: 1) Are statements of services that the software, software product or software-based system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. 2) Describe functionality or system services. 3) Are requirements about the behavior and input-output transformations of the software, software product or software-based system. [[Balci 2016](#Balci2016)]

This subsection documents thirteen use cases offered in the VolunteerCloud cloud software application. Following the instructions for requirement specifications learned in [[Balci 2016](#Balci2016)], each use case is documented in terms of the following fields: description, actors, preconditions, flow of events, extension points, and post conditions. In addition, a set of functional requirements has been included for each use case.

### Use Case 1: Create an Account

Describes how volunteers and organizations can create an account in VolunteerCloud.

#### Use Case 1 Documentation

Table 1. Use Case 1 Documentation

|  |  |
| --- | --- |
| Use Case 1 Documentation | |
| Use Case ID: | 1 |
| Use Case Name: | **Create Account**  This use case describes the process by which a Volunteer or an Organization can create an account in VolunteerCloud. |
| Actors: | |
| Volunteer or Organization | |
| Preconditions: | |
| User accesses VolunteerCloud. | |
| Flow of Events of the Primary Scenario: | |
| 1. Use case starts when user accesses the VolunteerCloud home page. 2. User selects “Register”. 3. User chooses whether to register as a Volunteer or an Organization. 4. User provides: username, password, email address, and security questions/answers. 5. User clicks ***Continue***. 6. The system verifies the input:   If the username, email, or password are not valid,  The system will give an alert message.  If the username or email already exists,  The system will give an alert message.  If the confirm password differs from the password,  The system will give an alert message.   1. User provides additional basic information: name, address, mission, website, phone number. 2. User clicks ***Submit***. 3. The system stores the information into the database. 4. User has access to the new account. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * User cancels creation of the account. * User attempts registration with an existing email address or username. * Database exception occurs while account is being created. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The user account is created and the dashboard screen is displayed. | |

#### Functional Requirements Associated with Use Case 1

1. User shall be able to create at most 1 account per email address.
2. User shall be able to select a unique username that contains 4 to 14 alphanumeric characters.
3. Password shall contain 6 to 16 characters with at least; 1 uppercase, 1 lowercase, 1 number, and 1 special character.
4. User shall be able to select a user type: volunteer or organization.
5. User shall be able to select a security question and to store its answer.

### Use Case 2: Update an Account

Describes how volunteers and organizations can update their accounts in VolunteerCloud.

#### Use Case 2 Documentation

Table 2. Use Case 2 Documentation

|  |  |
| --- | --- |
| Use Case 2 Documentation | |
| Use Case ID: | 2 |
| Use Case Name: | **Update Account**  This use case describes the process by which a Volunteer or an Organization can update an account in VolunteerCloud. |
| Actors: | |
| Volunteer or Organization | |
| Preconditions: | |
| User must be signed-in  System displays ***Account Information*** screen | |
| Flow of Events of the Primary Scenario: | |
| 1. Use case starts when user selects ***Edit*** on the ***Account Information*** screen. 2. System enables editing. 3. User updates basic account information. 4. System validates changes to password 5. User clicks ***Submit***. 6. Changes are validated and stored. 7. VolunteerCloud displays ***Account Information*** screen with new information. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * User cancels update account editing. * Database exception occurs while account is being created. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The user account is updated and an updated ***Account Information*** screen is displayed. | |

#### Functional Requirements Associated with Use Case 2

1. User shall be able to update account information.
2. User shall be prevented from changing username or email address.
3. Password changes shall be verified; contain 6 to 16 characters, 1 uppercase, 1 lowercase, 1 number, and 1 special character.
4. User shall fill required fields.
5. Changes shall be visible after submission.
6. VolunteerCloud shall enable user to upload a photo to display in profile(jpg, jpeg, png).

### Use Case 3: Deactivate Account

Describes how volunteers and organizations can Deactivate their accounts in VolunteerCloud.

#### Use Case 3 Documentation

Table 3. Use Case 3 Documentation

|  |  |
| --- | --- |
| Use Case 3 Documentation | |
| Use Case ID: | 3 |
| Use Case Name: | **Deactivate Account**  This use case describes the process by which a Volunteer or an Organization can deactivate their VolunteerCloud account. |
| Actors: | |
| Volunteer or Organization | |
| Preconditions: | |
| User must be signed-in  System displays ***Account Information*** screen | |
| Flow of Events of the Primary Scenario: | |
| 1. Use case starts when user selects ***Deactivate Account*** on the ***Account Information*** screen. 2. System prompts user with action verification message. 3. User confirms action by selecting ***yes***. 4. Use case finishes when account is deactivated and user is redirected to ***Home Screen***. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * User cancels deactivation of the account. * Database exception occurs while account is being deactivated. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The user account is deactivated and the ***Home Screen*** is displayed. | |

#### Functional Requirements Associated with Use Case 3

1. Users shall be able to deactivate their accounts.
2. VolunteerCloud shall prompt user with action verification message prior to deactivation.
3. User’s information shall remain stored after deactivation of account.
4. User’s profile information shall not be accessible if account is deactivated.
5. Chat shall only be available to users with active accounts.

### Use Case 4: Search Opportunity

Describes how users can search for volunteering opportunities in VolunteerCloud.

#### Use Case 4 Documentation

Table 4.Use Case 4 Documentation

|  |  |
| --- | --- |
| Use Case 4 Documentation | |
| Use Case ID: | 4 |
| Use Case Name: | **Search Opportunity**  Describes how users can search for volunteering opportunities in VolunteerCloud. |
| Actors: | |
| Volunteer | |
| Preconditions: | |
| * Volunteer has active account. * Volunteer has logged-in to the system. | |
| Flow of Events of the Primary Scenario: | |
| 1. Use case starts when Volunteer is redirected to *Search Opportunities* screen 2. Volunteer inputs a zip code, keyword, and/or category. 3. Volunteer clicks ***Search***. 4. VolunteerCloud queries database and returns results:   If no matches were found,  The system will display a notification message.  If matches were found,  The system will return and display a list of opportunities   1. If the Volunteerwants to access more information about the opportunity,   The Volunteerclicks on the opportunity’s title   1. If the Volunteerwants to access more information about the organization,   The Volunteerclicks on the organization’s name   1. If the Volunteerwants to subscribe to the opportunity,   The Volunteerclicks ***Subscribe***   1. If the Volunteerwants to chat about the opportunity,   The Volunteerclicks ***Chat***, use “use case": Web Chat   1. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and no results are returned. * Chat disabled due to organization having deactivated their account. * Subscription to opportunity has been disabled due to organization having deactivated their account. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| 1. Web Chat | |
| Postconditions: | |
| Volunteer receives a list of volunteering opportunities that matches search criteria. | |

#### Functional Requirements Associated with Use Case 4

1. VolunteerCloud shall allow users to search for opportunities.
2. VolunteerCloud shall allow users to search opportunities by location, keyword, or categories.
3. VolunteerCloud shall allow user to subscribe to however many opportunities he/she wants.
4. VolunteerCloud shall allow users to unsubscribe from opportunity before it takes place.
5. VolunteerCloud shall display message to notify user if no results were found.
6. VolunteerCloud shall present results as a list of opportunities by displaying basic information such as title, description, and name of the organization.
7. VolunteerCloud shall enable the user to access more information about the organization.
8. VolunteerCloud shall enable the user to access more information about the opportunity.
9. VolunteerCloud shall not allow users to subscribe to opportunity if the organization’s account is deactivated.

### Use Case 5: Search Organization

Describes how users can search for organizations in VolunteerCloud.

#### Use Case 5 Documentation

Table 5. Use Case 5 Documentation

|  |  |
| --- | --- |
| Use Case 5 Documentation | |
| Use Case ID: | 5 |
| Use Case Name: | **Search Organization**  Describes how users can search for organizations in VolunteerCloud. |
| Actors: | |
| Volunteer | |
| Preconditions: | |
| * Volunteer has active account. * Volunteer has logged-in to the system. | |
| Flow of Events of the Primary Scenario: | |
| 1. Use case starts when Volunteer is redirected to “Search Organization” screen 2. Volunteer inputs a zip code, keyword, and/or category. 3. Volunteer clicks ***Search***. 4. VolunteerCloud queries database and returns results:   If no matches were found,  The system will display a notification message.  If matches were found,  The system will return and display a list of organizations   1. If the Volunteerwants to access more information about the organization,   The Volunteerclicks on the organization’s name   1. If the Volunteerwants to see the organization’s opportunities,   The Volunteerclicks ***Opportunities***   1. If the Volunteerwants to chat with the organization,   The Volunteerclicks ***Chat***, use “use case": Web Chat   1. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and no results are returned. * Chat disabled due to organization having deactivated their account. * Subscription to opportunity has been disabled due to organization having deactivated their account. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| 1. Web Chat | |
| Postconditions: | |
| Volunteer receives a list of organizations that matches search criteria. | |

#### Functional Requirements Associated with Use 5

1. VolunteerCloud shall allow users to search for organizations.
2. VolunteerCloud shall allow users to search organizations by location, keyword, or categories.
3. VolunteerCloud shall display message to notify user if no organizations that match the criteria were found.
4. VolunteerCloud shall present results as a list of organizations by displaying basic information such as name, mission, photo.
5. VolunteerCloud shall allow the user to access more information about the organization.
6. VolunteerCloud shall allow the user to access all the opportunities posted by the organization.
7. VolunteerCloud shall only return organizations with active accounts in the search result.

### Use Case 6: Access Volunteering History

Describes how users can access and download their volunteering activities history.

#### Use Case 6 Documentation

Table 6. Use Case 6 Documentation

|  |  |
| --- | --- |
| Use Case 6 Documentation | |
| Use Case ID: | 6 |
| Use Case Name: | **Access Volunteering History**  Describes how users can access and download their volunteering activities history. |
| Actors: | |
| Volunteer | |
| Preconditions: | |
| * Volunteer has active account. * Volunteer has logged-in to the system. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts when the user accesses the “Volunteering History*”* screen. 2. VolunteerCloud displays list of volunteering opportunities where user has participated. 3. If the user wants to access opportunities to which he/she subscribed but didn’t participate,   The user clicks ***Not participated***.   1. If the user wants to access all opportunities to which he/she subscribed regardless of participation,   The user clicks ***All***.   1. If the user wants to see his/her volunteering history within a range of time,   The user inputs the desired dates.   1. If the user wants to download the report,   The user clicks ***Download***.   1. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and data is not loaded properly. * Downloaded report is corrupted. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The user accesses his/her volunteering history report and has downloaded report. | |

#### Functional Requirements Associated with Use Case 6

1. VolunteerCloud shall allow users to access all of their volunteering history information.
2. VolunteerCloud shall allow users to export their volunteering history report as a pdf document.
3. VolunteerCloud shall allow users to access their volunteering history information in terms of their participation.
4. VolunteerCloud shall allow users to access their volunteering history information within their desired date range.

### Use Case 7: Manage Volunteering Opportunity

Describes how users can access and manage the volunteering opportunities posted by them in VolunteerCloud.

#### Use Case 7 Documentation

Table 7. Use Case 7 Documentation

|  |  |
| --- | --- |
| Use Case 7 Documentation | |
| Use Case ID: | 7 |
| Use Case Name: | **Manage Volunteering Opportunity**  Describes how users can access and manage the volunteering opportunities posted by them in VolunteerCloud. |
| Actors: | |
| Organization | |
| Preconditions: | |
| * Organization has active account. * Organization has logged-in to the system. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts when the user enters the “Dashboard” screen. 2. VolunteerCloud shows a list of the organization’s activities including the opportunities owned by them. 3. If the user wants to create a new opportunity,   Use “use case”: Create Volunteering Opportunity.   1. If the user wants to edit an opportunity,   User clicks ***Edit***.   1. If the user wants to cancel an opportunity,   Use “use case”: Cancel Volunteering Opportunity.   1. The user clicks on the opportunity’s title to access more information about the opportunity. 2. VolunteerCloud displays basic information about the opportunity, and lists the volunteers subscribed to it. 3. If the user wants to confirm or deny a volunteer’s participation in the event,   The user selects the appropriate option in the ***Participation*** dropdown.   1. If the user wants to see more details about one of the registered volunteers,   User clicks on the volunteer’s name.  VolunteerCloud displays detailed information about the volunteer.   1. If the Organizationwants to chat with the volunteer,   The Organizationclicks ***Chat***, use “use case": Web Chat   1. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| 1. If the user selects ***New Opportunity***, view use case “Create Volunteering Opportunity”. 2. If the user selects ***Cancel Opportunity***, view use case “Cancel Volunteering Opportunity”. | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and data is not loaded properly. * Network connectivity is dropped. * Chat with volunteer disabled due to volunteer having deactivated his/her account. * Volunteer’s participation fails to be stored correctly | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| 1. Create Volunteering Opportunity 2. Cancel Volunteering Opportunity 3. Web Chat | |
| Postconditions: | |
| User has accessed opportunity, seen information about the registered volunteers, and confirm their participation in the event. | |

#### Functional Requirements Associated with Use Case 7

1. VolunteerCloud shall allow the user to access information about their owned volunteering opportunities such as: title, category, description, date, status, address, registered volunteers.
2. VolunteerCloud shall enable the user to edit or update information about their owned volunteering opportunities such as: address, status, description, category, and title.
3. VolunteerCloud shall allow users to cancel their volunteering opportunity prior to the date of its occurrence.
4. VolunteerCloud shall allow users access more information about its registered volunteers by clicking on their names.
5. VolunteerCloud shall enable users to confirm or deny a volunteer’s participation in an event.
6. VolunteerCloud shall disable users to cancel an opportunity if the date of its occurrence has past.

### Use Case 8: Create Volunteering Opportunity

Describes how users can create a new volunteering opportunity in VolunteerCloud

#### Use Case 8 Documentation

Table 8. Use Case 8 Documentation

|  |  |
| --- | --- |
| Use Case 8 Documentation | |
| Use Case ID: | 8 |
| Use Case Name: | **Create Volunteering Opportunity**  Describes how users can create a new volunteering opportunity in VolunteerCloud. |
| Actors: | |
| Organization | |
| Preconditions: | |
| * Organization has active account. * Organization has logged-in to the system. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts after the user clicks ***New Opportunity***. 2. User provides: title, category, description, date, and address. 3. User clicks ***Submit***. 4. The system verifies the input:   If the title, category, or address are empty,  The system will give an alert message.  If the chosen date is a date prior to the current date,  The system will give an alert message.   1. The system stores the information into the database. 2. The new opportunity is accessible for subscription | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * User cancels creation of the opportunity. * Database exception occurs while opportunity is being created. * User attempts to create opportunity using date from the past. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The opportunity is created and an updated dashboard screen is displayed. | |

#### Functional Requirements Associated with Use Case 8

1. User shall be able to create as many volunteering opportunities as they desire.
2. User shall be able to select the volunteering area to which the opportunity belongs to.
3. User shall be able to indicate the address where the opportunity will occur.
4. User shall be able to include a title and a description for the volunteering opportunity.
5. User shall be able to indicate the future date when the opportunity will occur.
6. System shall verify input and provide an alert message if required fields are left blank.
7. System shall verify that the opportunity’s date is in the future.

### Use Case 9: Cancel Volunteering Opportunity

Describes how users can cancel a volunteering opportunity in VolunteerCloud.

#### Use Case 9 Documentation

Table 9. Use Case 9 Documentation

|  |  |
| --- | --- |
| Use Case 9 Documentation | |
| Use Case ID: | 9 |
| Use Case Name: | **Cancel Volunteering Opportunity**  Describes how users can cancel a volunteering opportunity in VolunteerCloud |
| Actors: | |
| Organization | |
| Preconditions: | |
| * Organization has active account. * Organization has logged-in to the system. * There exists an opportunity to be canceled. * Opportunity to be canceled has not occurred yet. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts after the user clicks ***Cancel Opportunity***. 2. System prompts user with action verification message. 3. User confirms action by selection ***yes***. 4. Opportunity is canceled and “Dashboard” screen is updated accordingly. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * User declines action verification prompt. * Database exception occurs while opportunity is being canceled. * User attempts to cancel opportunity that has already occurred. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The opportunity is canceled and data its status is changed to inactive. | |

#### Functional Requirements Associated with Use Case 9

1. VolunteerCloud shall enable users to cancel opportunities prior to date of occurrence.
2. VolunteerCloud shall prompt user for cancelation confirmation.
3. VolunteerCloud shall disable opportunity subscription after being canceled.

### Use Case 10: Search Volunteers

Describes how users can search volunteers in VolunteerCloud.

#### Use Case 10 Documentation

Table 10. Use Case 10 Documentation

|  |  |
| --- | --- |
| Use Case 10 Documentation | |
| Use Case ID: | 10 |
| Use Case Name: | **Search Volunteers**  Describes how users can search volunteers in VolunteerCloud. |
| Actors: | |
| Organization | |
| Preconditions: | |
| * User has active account. * User has logged-in to the system. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts after the user clicks ***Search Volunteers***. 2. User inputs a zip code, name, and/or category. 3. User clicks ***Search***. 4. VolunteerCloud queries database and returns results:   If no matches were found,  The system will display a notification message.  If matches were found,  The system will return and display a list of volunteers   1. If the userwants to access more information about the volunteer,   The userclicks on the volunteer’s name   1. If the userwants to chat with the volunteer,   The userclicks ***Chat***, use “use case": Web Chat   1. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and no results are returned. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| 1. Web Chat | |
| Postconditions: | |
| Organization receives a list of volunteers that match the search criteria. | |

#### Functional Requirements Associated with Use Case 10

1. VolunteerCloud shall allow users to search for volunteers.
2. VolunteerCloud shall allow users to search volunteers by location, name, or categories.
3. VolunteerCloud shall display message to notify user if no results were found.
4. VolunteerCloud shall present results as a list of volunteers by displaying at least basic information such as name, volunteering areas of interest, and photo of the volunteer.
5. VolunteerCloud shall enable the user to access more information about the volunteer.
6. VolunteerCloud shall not return volunteers whose accounts have been deactivated.

### Use Case 11: Manage Volunteering Areas of Interests

Describes how users can manage their volunteering areas of interest in VolunteerCloud

#### Use Case 11 Documentation

Table 11. Use Case 11 Documentation

|  |  |
| --- | --- |
| Use Case 11 Documentation | |
| Use Case ID: | 11 |
| Use Case Name: | **Manage Volunteering Areas of Interest**  Describes how users can manage their volunteering areas of interest in VolunteerCloud. |
| Actors: | |
| Volunteer or Organization | |
| Preconditions: | |
| * User has active account. * User has logged-in to the system. * System is currently displaying ***Account******Information***screen. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts after the user clicks ***Areas of Interest***. 2. System displays list of volunteering areas with checkboxes 3. If the useris interested in an area of interest,   User checks the volunteering category   1. If the user is not interested in an area of interest,   User unchecks the volunteering category   1. The userclicks ***Save*** 2. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and changes are not stored. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| The changes are saved and the user’s ***Account Information*** is updated. | |

#### Functional Requirements Associated with Use Case 11

1. VolunteerCloud shall display the users’ volunteering areas of interest.
2. VolunteerCloud shall enable users to update their areas of interest.

### Use Case 12: Web Chat

Describes how users can chat with other users in VolunteerCloud through the web-chat interface.

#### Use Case 12 Documentation

Table 12. Use Case 12 Documentation

|  |  |
| --- | --- |
| Use Case 12 Documentation | |
| Use Case ID: | 12 |
| Use Case Name: | **Web Chat**  Describes how users can chat with other users in VolunteerCloud through the web-chat interface. |
| Actors: | |
| Volunteer or Organization | |
| Preconditions: | |
| * User has active account. * User has logged-in to the system. * User has performed any of the use cases ***“4, 5, 7, or 10”*** beforehand. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts after the user clicks ***Chat*** with another user. 2. System displays web chat interface screen. 3. System lists previous messages held between the two users. 4. User types in message into input box. 5. The userclicks ***Send***. 6. User sends more messages or waits for response. 7. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and no results are returned. * Receiver user’s account has been deactivated. * Message fails to be sent. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| Users can successfully hold an ongoing chat. | |

#### Functional Requirements Associated with Use Case 12

1. VolunteerCloud shall allow Volunteers and Organizations to engage in communication through a web chat interface.
2. VolunteerCloud shall display previous messages held between both parties of the conversation. display their volunteering areas of interest.
3. VolunteerCloud shall allow web chat communications between active accounts only.
4. VolunteerCloud shall allow users to send messages when the receiving user is offline.
5. VolunteerCloud shall enable users to receive messages received during offline after login-in.
6. VolunteerCloud’s web chat interface shall display both of the users’ names, and their profile picture.

### Use Case 13: Linking Organization to VolunteerMatch’s Account

Describes how users can link their organization to their VolunteerMatch’s account in VolunteerCloud.

#### Use Case 13 Documentation

Table 13. Use Case 13 Documentation

|  |  |
| --- | --- |
| Use Case 13 Documentation | |
| Use Case ID: | 13 |
| Use Case Name: | **Linking Organization to VolunteerMatch’s Account**  Describes how users can link their organization to their VolunteerMatch’s account in VolunteerCloud. |
| Actors: | |
| Organization | |
| Preconditions: | |
| * User has active account. * User has logged-in to the system. * User owns account in VolunteerMatch. * System is currently displaying ***Account******Information***screen. * User’s organization has not been previously linked to another VolunteerMatch’s account. | |
| Flow of Events of the Primary Scenario: | |
| 1. The use case starts after the user clicks ***Link to VolunteerMatch***. 2. The system allows the user to perform a VolunteerMatch’s account search. 3. The user inputs a zip code and name. 4. User clicks ***Search***. 5. The system makes a call to the VolunteerMatch’s API and returns results:   If no matches were found,  The system will display a notification message.  If matches were found,  The system will return and display a list of organizations   1. The user selects his/her VolunteerMatch’s account. 2. The system stores the selected organization’s VolunteerMatch ID in the database. 3. System displays a success message. 4. The user clicks ***Import Opportunities*** 5. The system displays opportunities owned by the organization in VolunteerMatch. 6. The user selects the opportunity to be imported. 7. The system imports and stores the opportunity from VolunteerMatch into the database. 8. The use case ends. | |
| Flow of Events of the Alternative Scenarios: | |
| None | |
| Flow of Events of the Exception Scenarios: | |
| * Database exception occurs and imported opportunities are not stored. * Organization from VolunteerMatch has already been linked to VolunteerCloud * Opportunity from VolunteerMatch has already been imported into VolunteerCloud. * Network connectivity is dropped. | |
| Extension Points: | |
| None | |
| “Used” Use Cases: | |
| None | |
| Postconditions: | |
| User’s organization has been linked to his/her organization in VolunteerMatch.  User is able to import owned opportunities in VolunteerMatch. | |

#### Functional Requirements Associated with Use Case 13

1. VolunteerCloud shall allow users to link their organizations to their accounts in VolunteerMatch.
2. VolunteerCloud shall allow users to import opportunities from VolunteerMatch into VolunteerCloud after accounts have been linked.
3. VolunteerCloud shall not allow users to link their accounts to more than 1 organization.
4. VolunteerCloud shall not allow users to import opportunity duplicates.
5. VolunteerCloud shall allow calls to the Volunteer

## Non-Functional Requirements

Non-functional requirements are requirements that are unrelated to functionality of the software, software product or software-based system. For example: Requirements for Interoperability, Performance, Usability, Standards, Delivery, Portability, Privacy, Safety. [[Balci 2016](#Balci2016)]

This section displays the list of non-functional requirements that VolunteerCloud shall meet:

1. VolunteerCloud shall be accessible over HTTP request-response based connections.
2. VolunteerCloud shall be freely available for no cost.
3. VolunteerCloud shall be accessible by computers and mobile devices.
4. VolunteerCloud shall be accessible through the following web browsers:
   1. Google Chrome
   2. Mozilla Firefox
5. VolunteerCloud shall be developed in an environment using the following software:
   1. NetBeans IDE,
   2. Primefaces Framework,
   3. GlassFish Java EE Application Server,
   4. MySQL Database.
6. VolunteerCloud shall be deployed and delivered on a dedicated server meeting the following specifications:
   1. CentOS Unix operating system,
   2. Server URL: venus.cs.vt.edu
   3. Server is located in McBryde 116 Software Engineering Lab, Virginia Tech
7. VolunteerCloud shall be capable of communicating with other APIs through HTTP responses.
8. VolunteerCloud shall be stable and maintainable.
9. VolunteerCloud shall be flexible and extendable to adapt richer features in the future.
10. ARENAS shall handle these privacy issues:
    1. Ensure the user’s personal information cannot be shared with any other users or a third party.
    2. Ensure an instructor’s intellectual property rights cannot be shared with any other user or a third party without the instructors’ agreement.
    3. Enable to notify the user any change about their user information.
11. ARENAS shall handle these security issues:
    1. Identify all of its client applications before allowing them to use its capabilities.
    2. Ensure that all of the users need to change their passwords every 3 months.
    3. Enable all operations in the system to be recorded in its log.
    4. Enable all data to be backed-up daily.
    5. Enable the user’s information to be encrypted with RSA algorithm.
12. ARENAS shall handle these usability issues:
    1. Enable any graphical user interface to use an adequate font size which is usable by persons even with limited visual acuity.
    2. Enable the *Help* text to be provided in *HTML* format.
    3. Ensure all data can be exported in *XML* format.
    4. Enable any instructor to create an online exam consisting one question in 2 minutes.

## 

# VolunteerCloud Architecture Specification

Software application architecture is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application. [[Microsoft 2009](#Microsoft2009)]

The architecture is not the operational software. Rather, it is a representation that enables you to (1) analyze the effectiveness of the design in meeting its stated requirements, (2) consider architectural alternatives at a stage when making design changes is still relatively easy, and (3) reduce the risks associated with the construction of the software. [[Pressman 2014](#Pressman2014)]

Choosing the architecture of a system is highly important due to the impact it can have or not in the end product. It is important to remember that the architecture chosen for a software product will likely affect certain of its quality attributes such as modifiability, performance, security, and even usability. Choosing the right architecture is also important due to the trade-offs and constraints that a style imposes on a system. Binding it all together, George Fairbanks [[Fairbanks 2010](#Fairbanks2010)] explained how architecture affects the system in a more concrete manner by providing the following points:

* Architecture influences quality attributes.
* Architecture acts as the skeleton of a system.
* Architecture is (mostly) orthogonal to functionality.
* Architecture constrains systems.

This section introduces the architecture models used in VolunteerCloud, and why these models are a good fit for the implementation of VolunteerCloud.

## Client-Server Architecture

In the client-server architectural style, clients request services from servers. The request is usually synchronous and across a request-reply connector, but can vary. There is an asymmetry between client and server in that the client can request that the server do work, but not the reverse [[Fairbanks 2010](#Fairbanks2010)]. The client-server architecture is famous in the industry of providing software as a service because it does not require the server to know the existence of all its clients, while the clients are able to perform as many requests as possible.

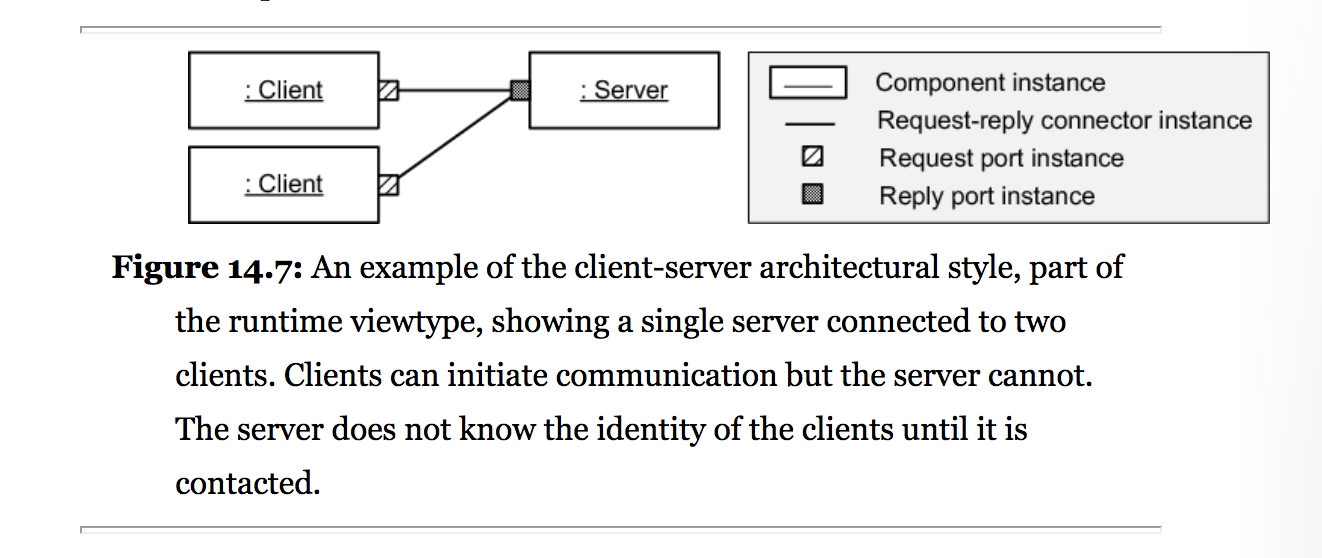


Figure X. Client-Server Architecture [[Fairbanks 2010](#Fairbanks2010)]

One common variant of this architecture style is known as the N-tier client-server architecture style. This style is regularly used because of the horizontal scalability it offers to the system as new or richer version of the requirements appear. For example, figure [N-Tier] can be easily mapped to a software solution that processes the clients’ requests through a load balancer before sending them to their respective application servers.

Client Load Balancer Server

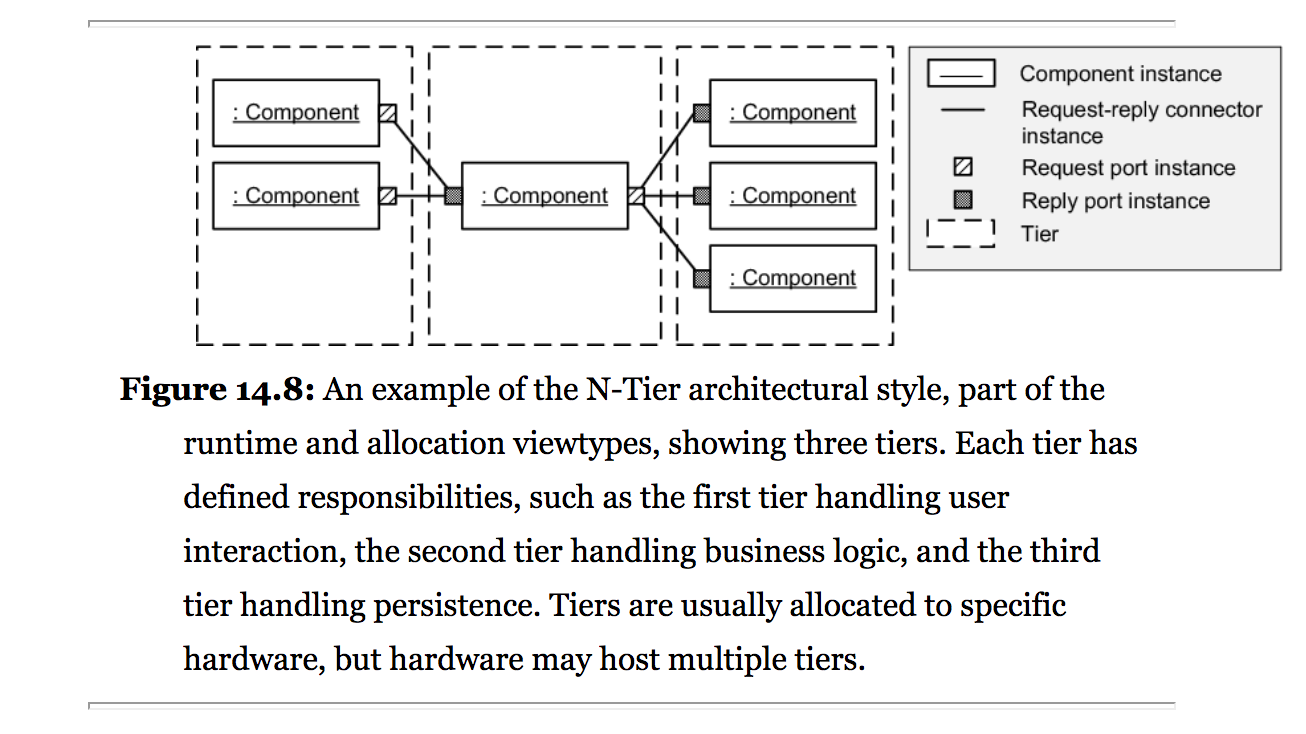


Figure X. N-Tier Client-Server Architecture [[Fairbanks 2010](#Fairbanks2010)]

Because this architectural style separates client code and server code into different components, it is possible to improve system performance by shuffling the components among computer processes [[Pfleeger 2010](#Pfleeger2010)]. This is one of the main reasons why technologies able to execute more logic on the client side such as JavaScript, have gained momentum in the past years.

## Model-View-Controller Architecture, MVC

One of the most popular architecture styles is the one known as Model-View-Controller or MVC architecture. This architecture has gained its reputation due to the benefits it provides such as good scalability and modifiability. Since the model, the view, and the controller are all independent of each other, it offers the developers the means to quickly adapt the system to new features and changes in requirements when needed.

The model contains all application-specific content and processing logic, including all content objects, access to external data/information sources, and all processing functionality that is application specific. The view contains all interface-specific functions and enables the presentation of content and processing logic, including all content objects, access to external data/ information sources, and all processing functionality required by the end user. The controller manages access to the model and the view and coordinates the flow of data between them. [[Pressman 2014](#Pressman2014)]

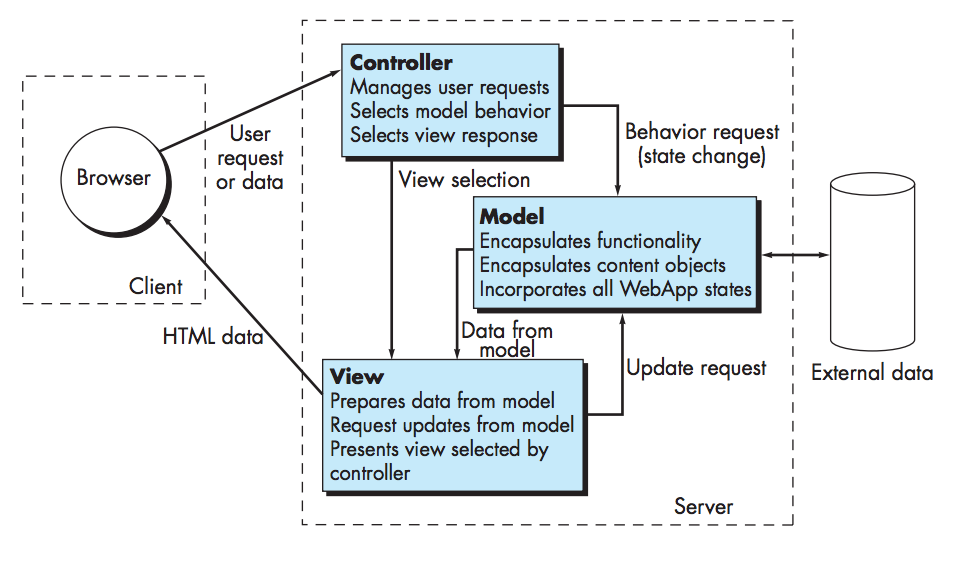


Figure X. Model View Controller (MVC) Architecture [[Pressman 2014](#Pressman2014)]

## VolunteerCloud’s Architecture

Given that VolunteerCloud is to be offered as a cloud software application, it falls under the industry of providing Software as a Service. Hence, VolunteerCloud’s software architecture consists of a mixture of the Client-Server architecture, and the Model-View-Controller architecture to benefit from the advantages these offer. As a consequence, VolunteerCloud shall display good maintainability, modifiability, and scalability.

Figure x illustrates VolunteerCloud’s software architecture; ***Label 1*** is comprised of laptops, mobile devices, or any desktop computers accessing the service through a web browser. ***Label 2*** is comprised of the controllers that handle the communication between the client and business tiers. ***Label 3*** contains all of VolunteerCloud’s business logic. ***Label 4’s*** primary purpose consists of handling the data mapping between the business and data source tiers by the use of APIs such as JDBC and JPA. ***Label 5*** is comprised of VolunteerCloud’s MySQL relational database.

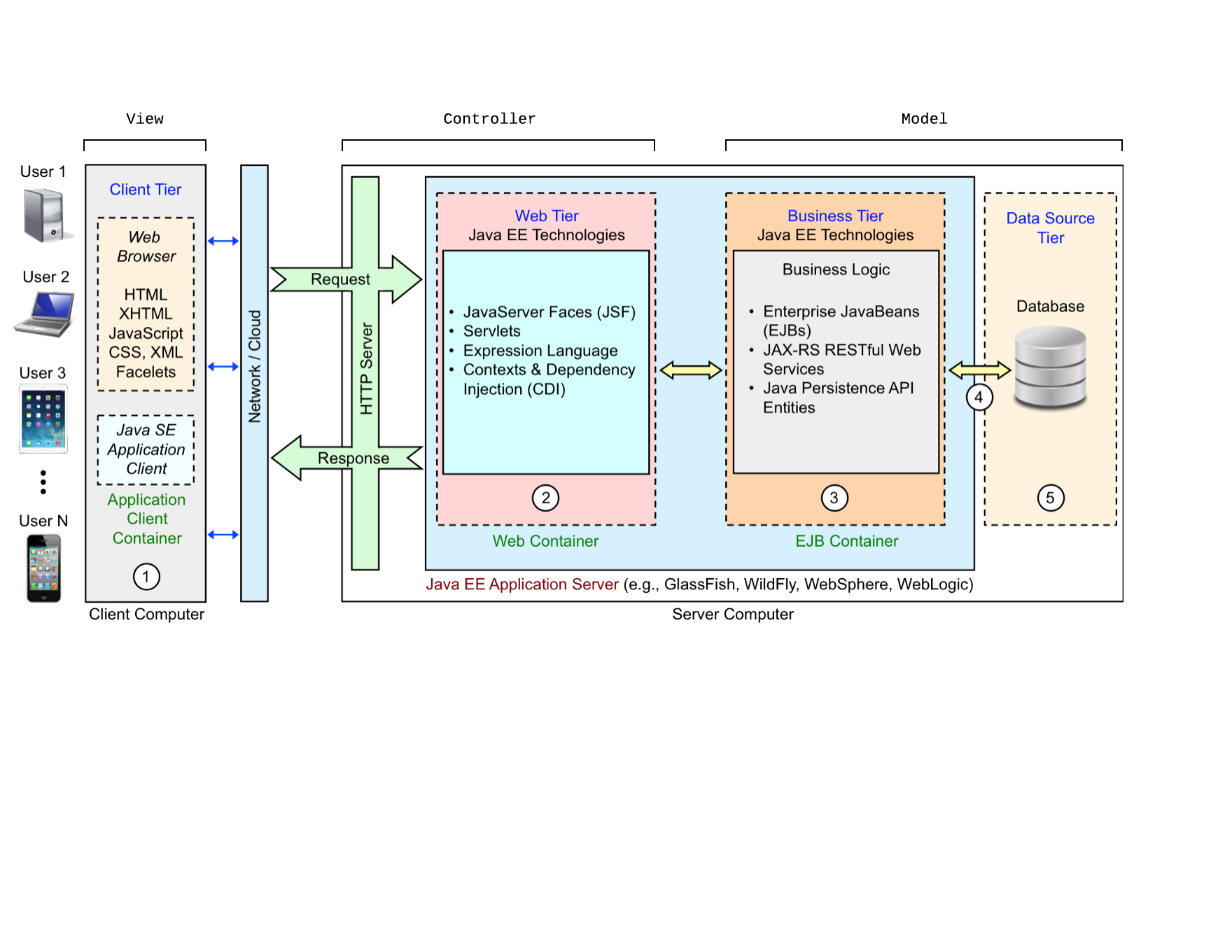


Figure X. MVC Mapping to Java EE Client-Server Architecture [[Balci 2016](#Balci2016)]

*Tier 1 - Client*: Java EE applications are unique in that not only do they support the typical web protocols like HTML, but they can also work in a dedicated client container to run Java code. Java EE can also be used as a RESTful API by returning data in XML or JSON, or any format for that matter, to the native client application to be displayed. Many of the libraries required to render JSF pages are loaded remotely as JavaScript components when the page is displayed on a browser. These libraries can either be hosted by the Java EE server or on a content delivery network (CDN) to minimize code reuse and load on a developer's infrastructure. A browser is not the only method of ingesting a Java EE web application as it can server as the backend implementation for any types of clients from smartphones to other web servers. [[Schutt and Balci 2016](#Schutt_And_Balci2016)]

*Tier 2 - Web*: The web tier for a Java EE web-application consists mainly of JSF pages (as define by the expression language) and Context & Dependency Injection (CDI). CDI provided developers the ability to dynamically request modules as needed based on the incoming request. While JSF is not required by Java EE to serve content over the web, it is highly recommended as the standard method for generating HTML and JavaScript based on the incoming request since it abstracts data bindings and application logic by with expression language. The design of the JSF framework also values component reuse and extensibility which greatly reduces overall development work for the software engineer. [[Schutt and Balci 2016](#Schutt_And_Balci2016)]

*Tier 3 - Business*: The business tier contains all logic pertaining to a specific business need by implementing various Enterprise Java Beans (EJBs). These EJBs can then respond to specific requests to JAX-RS RESTful calls that perform some predefine function on the models. EJBs can also respond to specific events that originate from a JSF page or component. The Java Persistence API is also listed here because it is used to define the models that are represented in the database. The business tier in Java EE applications is responsible for the security and the state of the current application. These sessions are managed by the application on a client-by-client basis to improve the security and continuity of the business logic components. Finally, while Java EE is mostly a request-driven framework, it also contains modules that allow for event-driven processing and messaging between the server and the client called the Java Messaging Service. [[Schutt and Balci 2016](#Schutt_And_Balci2016)]

*Tier 4 – Data Mapping*: Java EE provides several different options for data mapping technologies including Java Database Connectivity (JDBC), Java Persistence API, Connector Architecture, and Transaction API. While each of these options relies on the base Java EE APIs for connecting to a data source, each of them play a unique role in the ecosystem. JDBC and the Java Persistence API are mainly used in the management of relational data objects by defining entities and generating SQL queries based on the class definition. The Connector Architecture is primarily geared toward enterprise customers looking to integrate legacy Java applications into a Java EE web application. Transaction API is a more fine-grained approach to JDBC by managing individual methods within an entity. [[Schutt and Balci 2016](#Schutt_And_Balci2016)]

*Tier 5 – Data Sources*: Java EE provides support for all major database management systems such as Oracle, DB2, SQLServer, MySQL, and PostgreSQL. Additionally, Java EE supports NoSQL data sources like mongodb. Like other web-application frameworks, Java EE is fairly agnostic to the systems implemented as the data source. Modern data mapping technologies, like Java Persistence API, are generic enough in definition to be extended to work with any kind of data source, object- or document-based. The type of data source utilized by a web-application is entirely dependent on the requirements and expertise of the developers involved. The choice of relational or NoSQL is left to the software engineer since there has yet to be a unified consensus on the appropriate usage of data sources for a given application. [[Schutt and Balci 2016](#Schutt_And_Balci2016)]

## The Department of Defense and Architecture Framework (DoDAF)

The Department of Defense Architecture Framework (DoDAF) is the overarching, comprehensive framework and conceptual model for architectural descriptions developed within the DoD. This framework helps Department of Defense (DoD) managers at all levels make effective decisions by ensuring the sharing of consistent and common information across the Department, Joint Capability Areas (JCAs), missions, components, and programs. The DoDAF helps the DoD Chief Information Officer (CIO) develop and maintain architectures required by the Clinger-Cohen Act. It also fulfills guidance from the Office of Management and Budget (OMB) and other Departmental directives and instructions. [[DoDAF](#DoDAF)]

### OV-2: Operational Resource Flow Description

The following OV-2 model below (***Figure x)*** illustrates the resources flow in VolunteerCloud. This model is characterized as being a description of the resource flows exchanged between operational activities. [[DoDAF](#DoDAF)]

After creating an account in VolunteerCloud; 1) Organizations can link their VolunteerCloud to their VM account if desired. 2) Organizations can create opportunities or import their previously posted opportunities in VolunteerMatch. 3) Volunteers can search for opportunities. 4) Volunteers can search for organizations. 5) Organizations can search for volunteers. 6) Volunteers and organizations can start a webchat. 6) Organizations and volunteers can access their volunteering activity history.

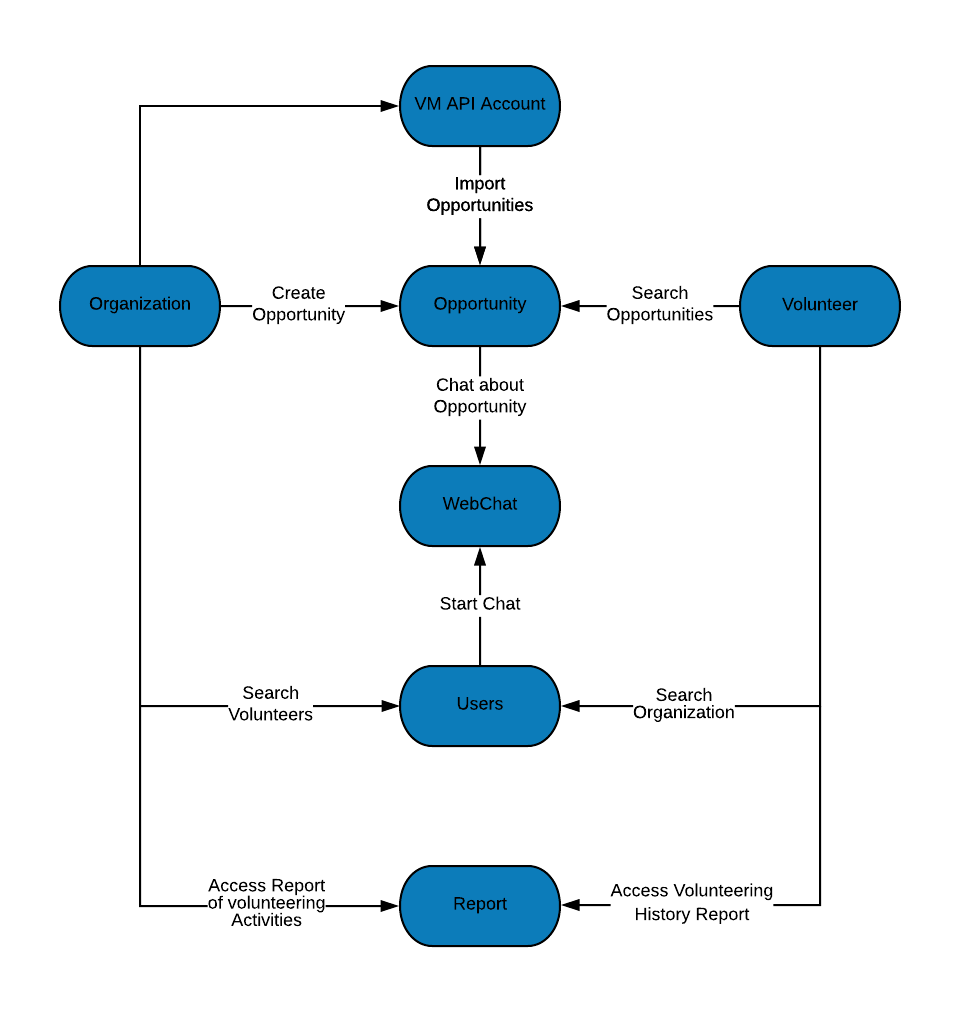


Figure X. OV-2: Operational Resource Flow Description

### SV-1: Systems Interface Description

The following SV-1 model below (***Figure x)*** illustrates VolunteerCloud’s system interface. This model is characterized for identifying an applications’ systems, and their interconnections. [[DoDAF](#DoDAF)]

VolunteerCloud has a “Control System” that consists of the application’s controllers which hold data and execute methods during the life of the session. The control system communicates with the different classes and their resources in the database, which the user views and interacts with through the user interface. In addition, VolunteerCloud communicates externally with VolunteerMatch’s and Twilio’s APIs in order to provide the *webchat* and *opportunity import* features. Finally, VolunteerCloud can interact with users and developers through its user interface and its public API web service.

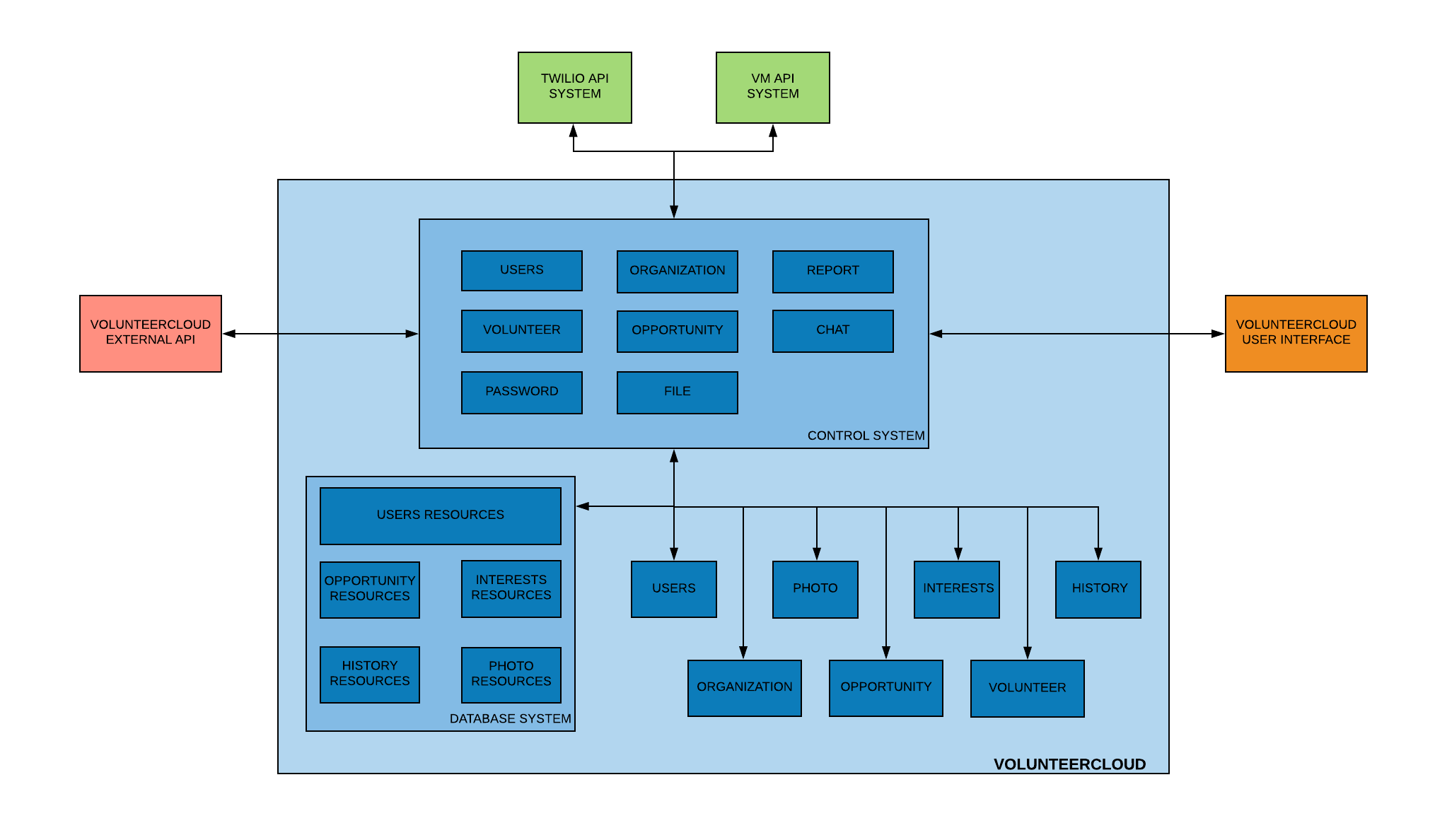
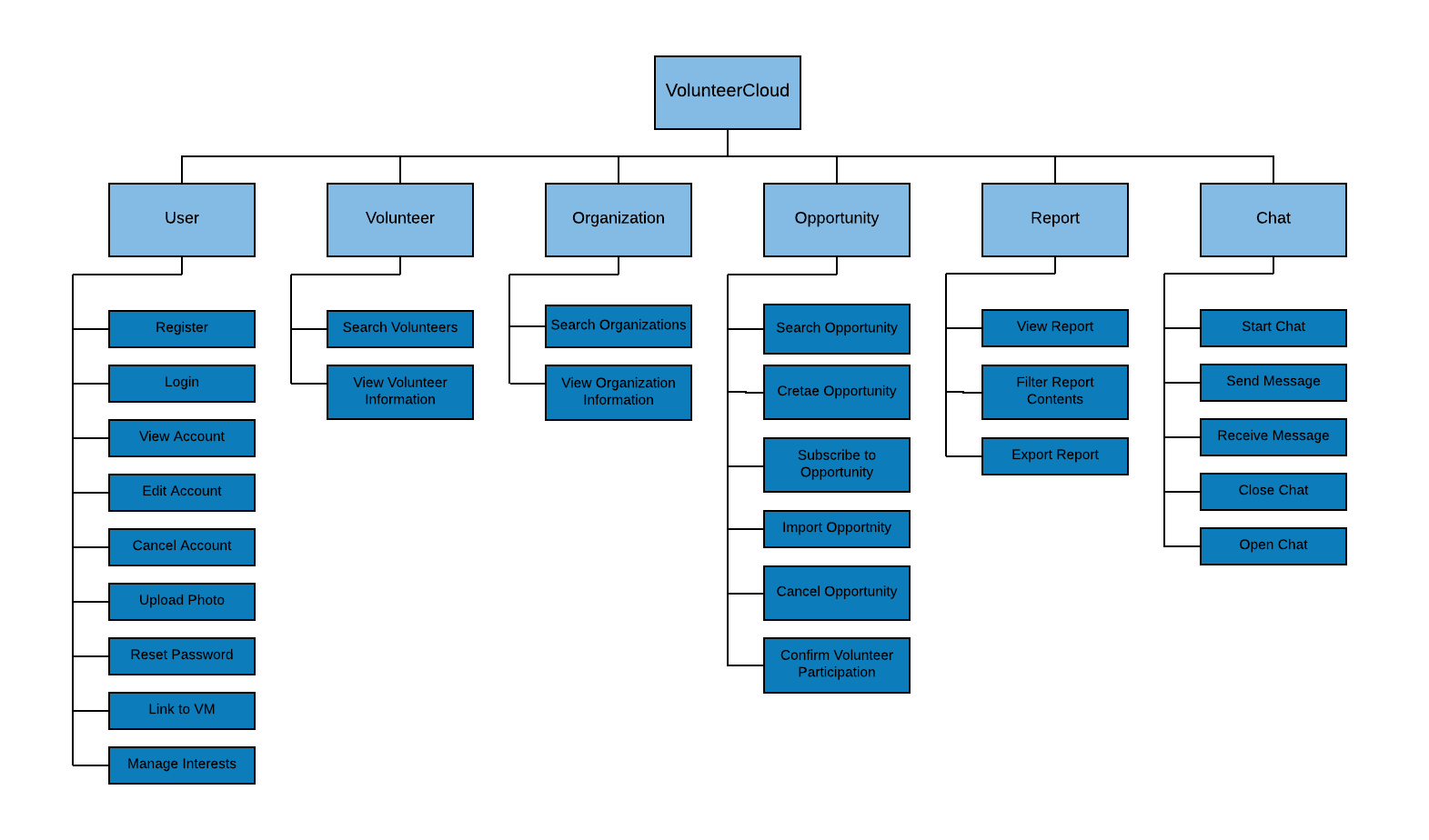


Figure X. SV-1: Systems Interface Description

### SV-4: Systems Functionality Description

The following SV-4 model below (***Figure x)*** depicts VolunteerCloud’s functions. This model is characterized for describing the functions performed by systems and the system data flows among system functions. [[DoDAF](#DoDAF)]

VolunteerCloud has a total of 6 modules where each of them has their own functions; 1) User module: register, login, view account, edit account, cancel account, upload photo, reset password, link to VM account, manage interests. 2) Volunteer module: search volunteers, view volunteer information. 3) Organization module: search organizations, view organization information. 4) Opportunity module: search opportunity, create opportunity, subscribe/unsubscribe to opportunity, import opportunity, cancel opportunity, confirm volunteer participation. 5) Report module: view report, filter report contents, export report. 6) Chat module: start chat, send message, receive message, close chat, open chat.

Figure X. SV-4: Systems Functionality Description

# ARENAS Design Specification

Intro and diagrams summary citation(Delete)

The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software based systems as well as for modeling business and similar processes. [[OMG 2011](#OMG2011)]

## Class Diagrams

The class diagram is a static model that supports the static view of the evolving system. It shows the classes and the relationships among the classes that remain constant in the system over time [[Dennis 2012](#Dennis2012)].

### ARENAS Class Diagram

The class diagram (***Figure 17*** ) shows ten classes and their relationships in ARENAS. Each class has its data fields and methods, and has some relationship with other classes. To be specific, the class “Instructor” and “Student” inherit the class “Users”, and 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”, and 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”.

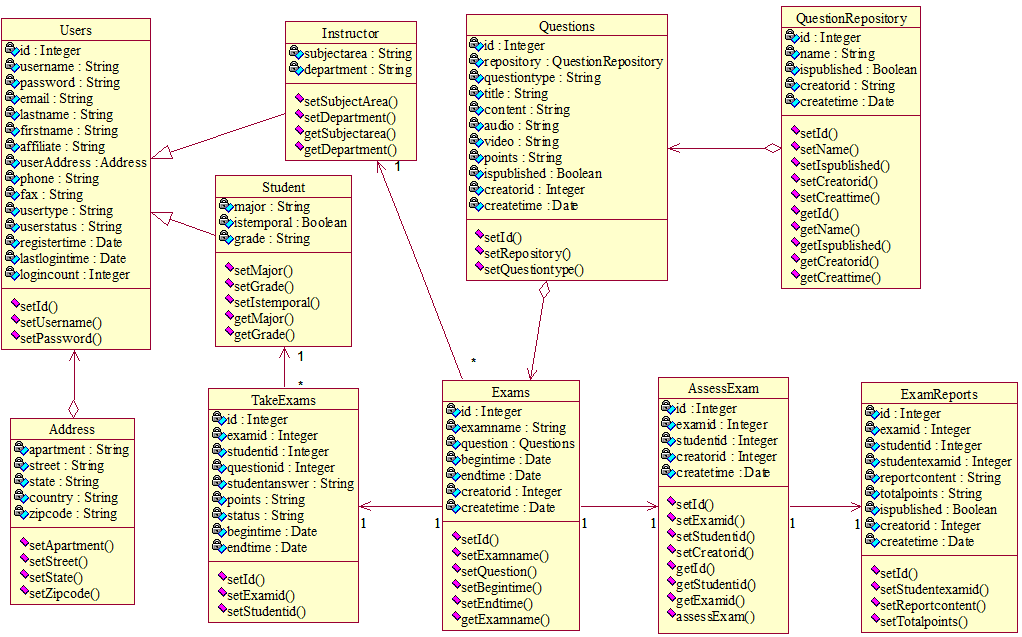


Figure 17. ARENAS Class Diagram

## Package Diagrams

Package diagrams are a subset of class diagrams. They organize elements of a system into related groups to minimize the amount of dependencies. One can think of package diagrams as a higher-level view of a system when compared to class diagrams [[Balci 2009](#Balci2009)].

### User-Exam-Database Package Diagram

The package diagram (***Figure 18***) shows four packages and their structures and dependency 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.

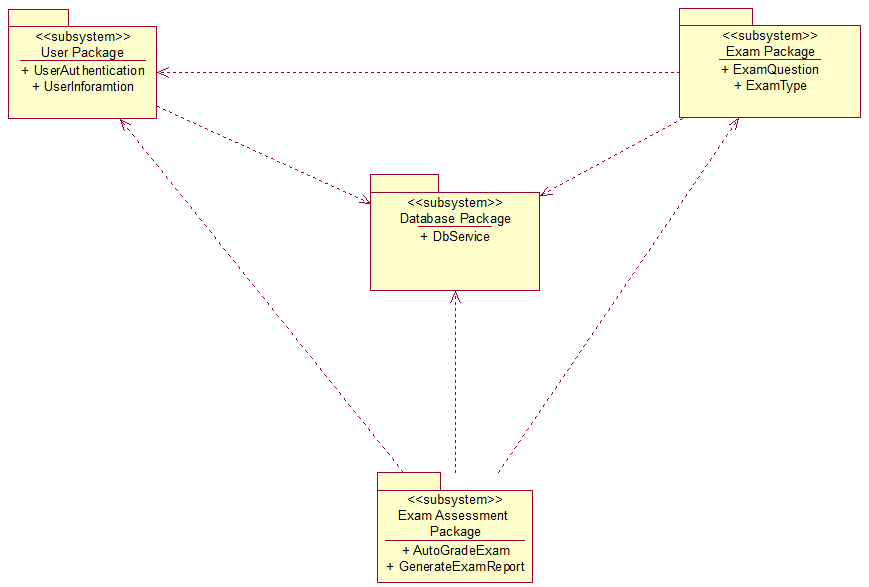


Figure 18. User-Exam-Database Package Diagram

## Component Diagrams

a component diagram shows the parts of a design for a software system. A component diagram helps you visualize the high-level structure of the system and the service behavior that those pieces provide and consume through interfaces [[Microsoft 2009](#Microsoft2009)].

### Exam-Question Component Diagram

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

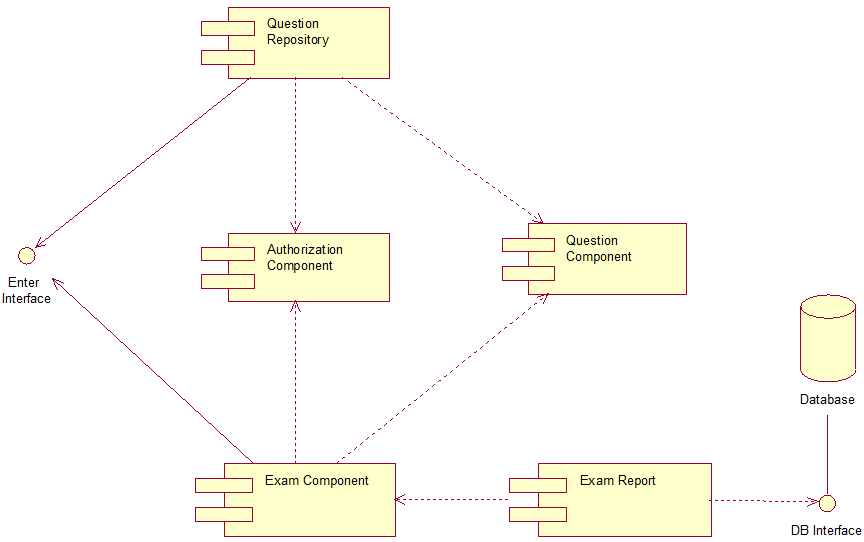


Figure 19. Exam-Question Component Diagram

## Statechart Diagrams

State diagrams or statechart diagrams are used to describe the behavior of a system by showing all the possible states of an object as events occur. Each diagram usually represents objects of a single class and tracks the different states of its objects through the system [[Balci 2009](#Balci2009)].

### Instructor Statechart Diagram

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

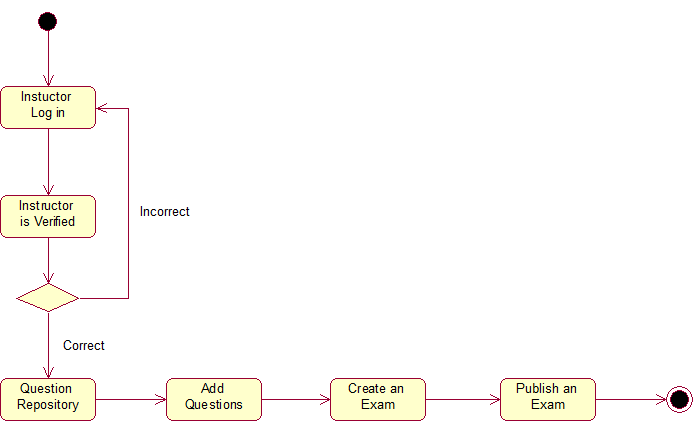


Figure 20. Instructor Statchart Diagram

### Student Statechart Diagram

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

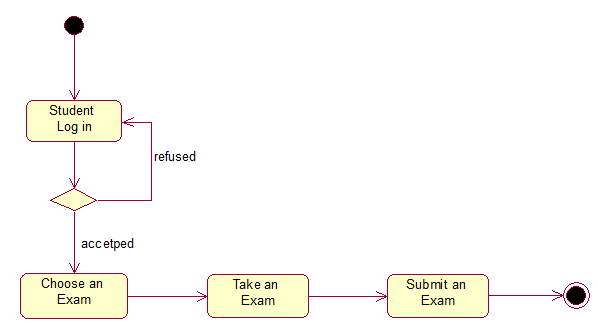


Figure 21. Student Statchart Diagram

## Activity Diagrams

Activity diagrams describe the workflow behavior of a system. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel, providing a way to model the workflow of a process similar to a flowchart. An activity diagram can also be used to model code-specific information such as a class operation [[Balci 2009](#Balci2009)].

### Instructor and Student Activity Diagram

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 to the question repository, selects questions from the question repository to create an exam, and publish the exam. Then a student takes the exam, submits the exam; the instructor generates the exam repot, and the student views the exam report, then the activity ends.

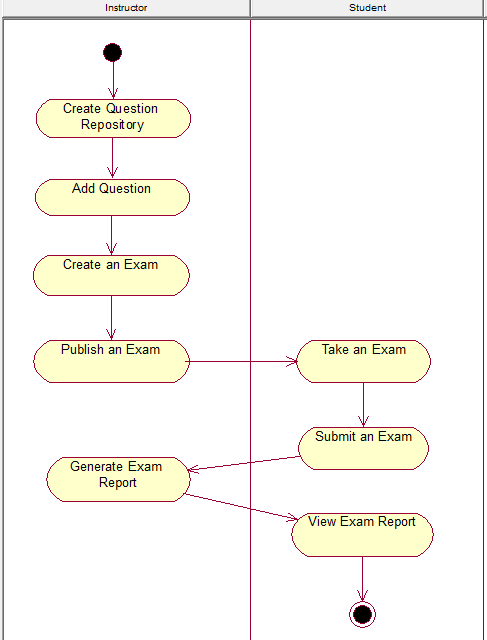


Figure 22. Instructor and Student Activity Diagram

## Sequence Diagrams

A sequence diagram is a dynamic model that supports a dynamic view of the evolving systems. It shows the explicit sequence of messages that are passed between objects in a defined interaction. Since sequence diagrams emphasize the time-based ordering of the activity that takes place among a set of objects, they are very helpful for understanding real-time specifications and complex use cases [[Dennis 2012](#Dennis2012)].

### Exam Creation Sequence Diagram

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

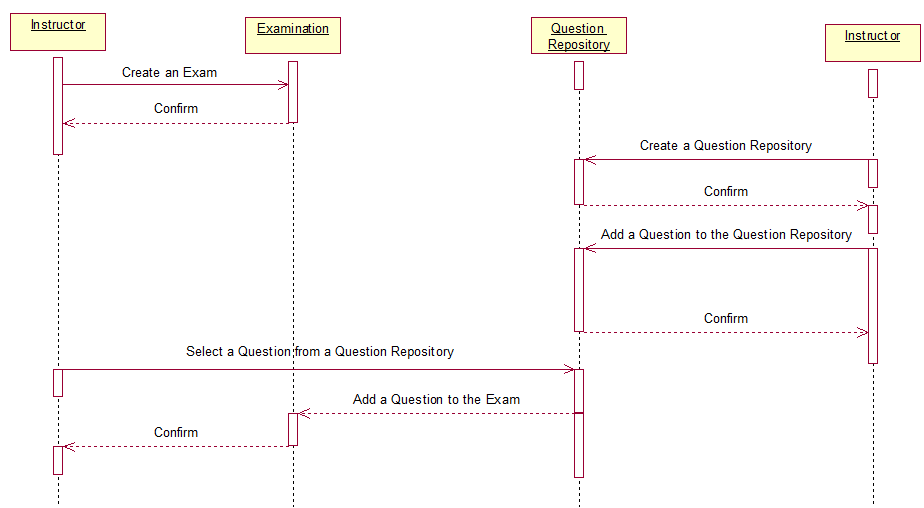


Figure 23. Exam Creation Sequence Diagram

### Exam Taking Sequence Diagram

The sequence diagram (***Figure 24*)** shows the time-based sequence interaction for taking an exam by a student. A student takes an exam, then submits the exam; the exam is assessed by the system and an exam report is generated; an instructor checks the exam report and publishes the exam report; then the student receives the exam report.

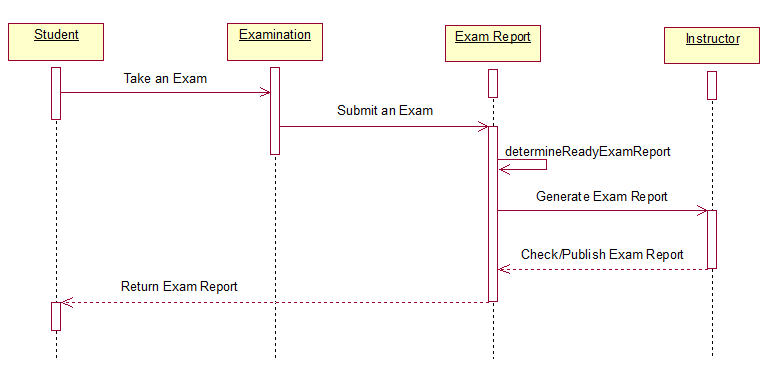


Figure 24. Exam Taking Sequence Diagram

## Collaboration Diagrams

A collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages [[Balci 2009](#Balci2009)].

### Exam Creation Collaboration Diagram

The exam creation diagram (

***Figure*** 25**)** shows the order of the message when an instructor creates an exam. When an instructor plans to create an exam, he or she firstly select a question from a question repository, then the question repository returns the question, and the instructor adds the question to an exam.

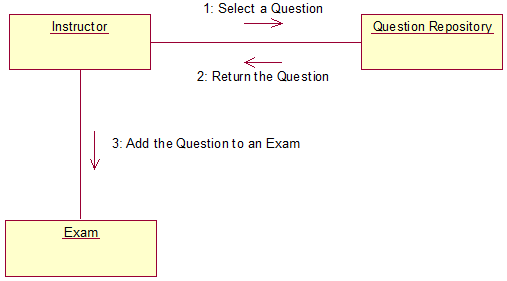


Figure 25. Exam Creation Collaboration Diagram

### Exam Taking Collaboration Diagram

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

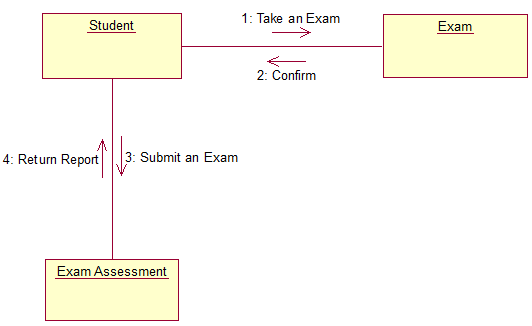


Figure 26. Exam Taking Collaboration Diagram

## Deployment Diagrams

A deployment diagram shows processors, devices, and connections [[Balci 2009](#Balci2009)].

### ARENAS Deployment Diagram

The deployment diagram (***Figure 27***) shows the physical deployment of the hardware components or the nodes in ARENAS. The Database Server is MySQL, and the Multiple Virtual Storage Mainframe like IBM MVS can be added seaml essly 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, and the protocol for other different processors is TCP/IP.

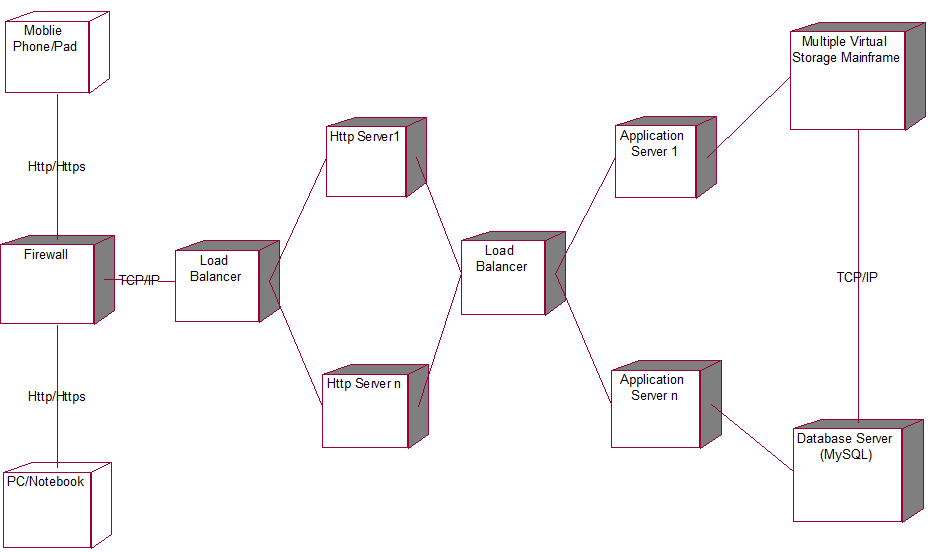


Figure 27. ARENAS Deployment Diagram

# ARENAS Functionality

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

## 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.*** User Registeration Interface).

## 

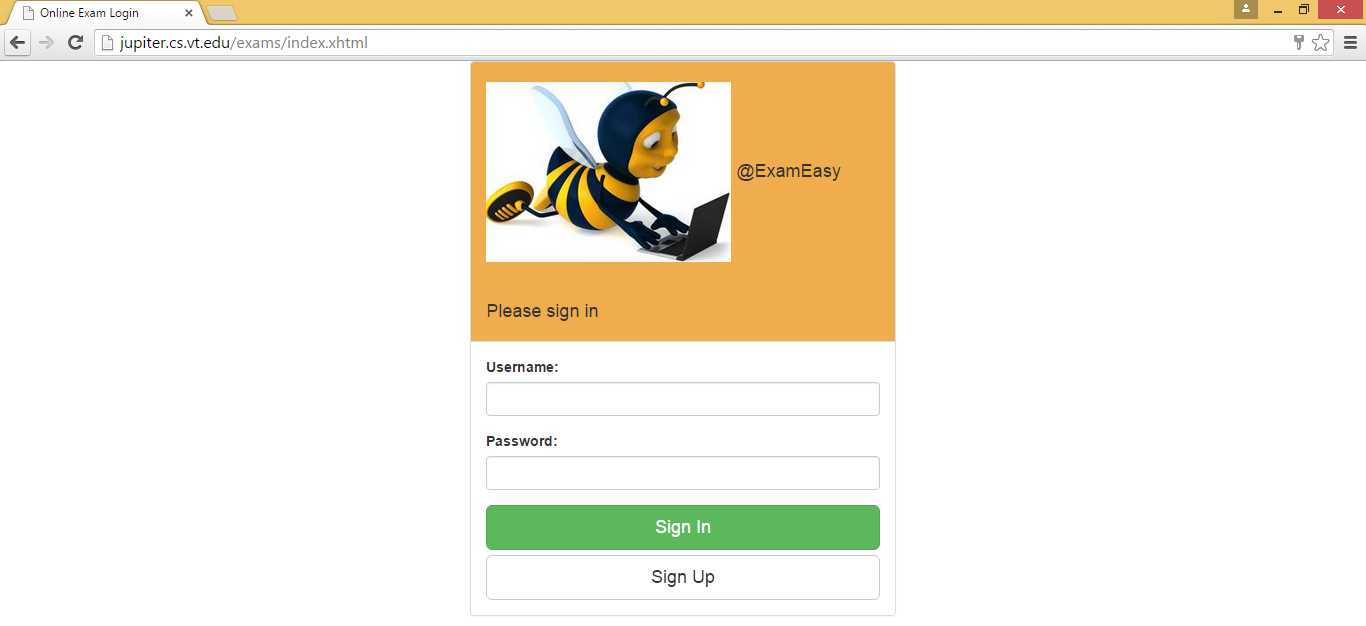


Figure 28. User Registeration Interface

## Question Repository

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

## 

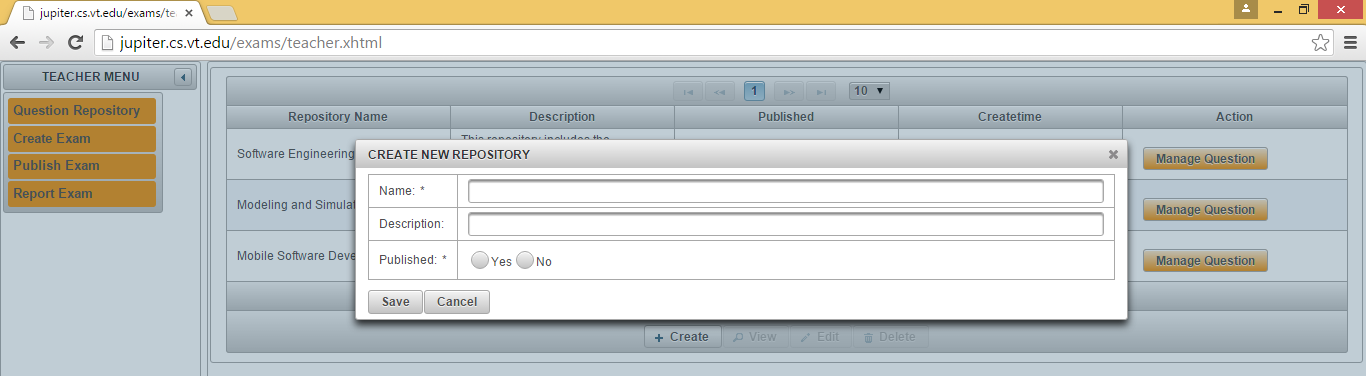


Figure 29. Question Repository Interface

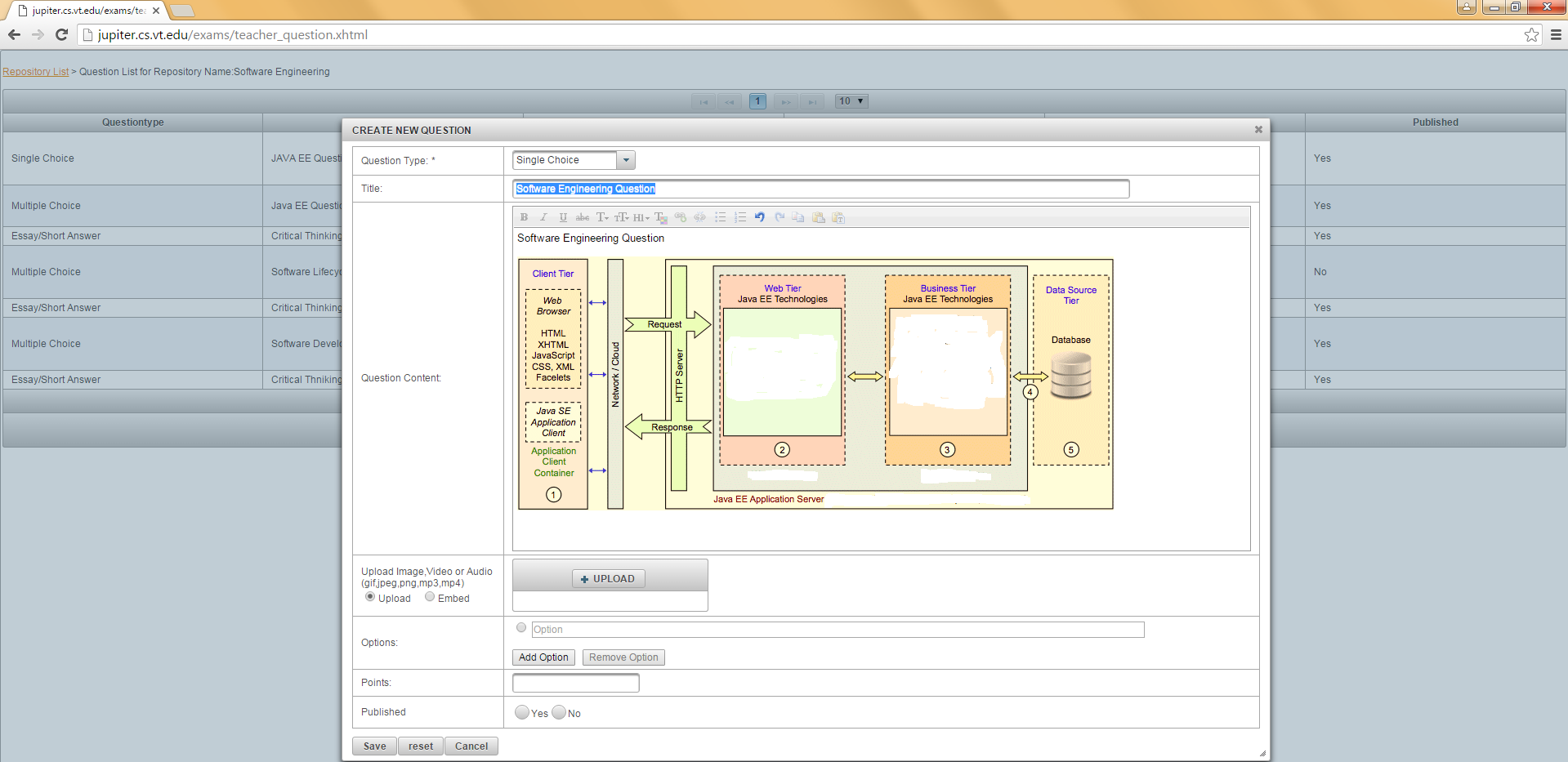
## Question Management

## After a question repository is created, an instructor can create a question for 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. When creating a question, the following information is required: question type, question title, question content, question options (only for single-choice question or multiple-choice question), points and published or not.

## When a question is created, an instructor can choose to add multimedia elements such as image, graphic, video or audio; these multimedia elements can be uploaded into the Server in which ARENAS is running if the instructor selects the “upload” choice. If the instructor selects “embedded” choice, he or she can copy a URL which directs a video in Youtube website and clicks “Submit URL” button, the video will be automatically embed in 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 databased for auto-grading and generating an exam report

## *(Figure 30. Question* Management Interface



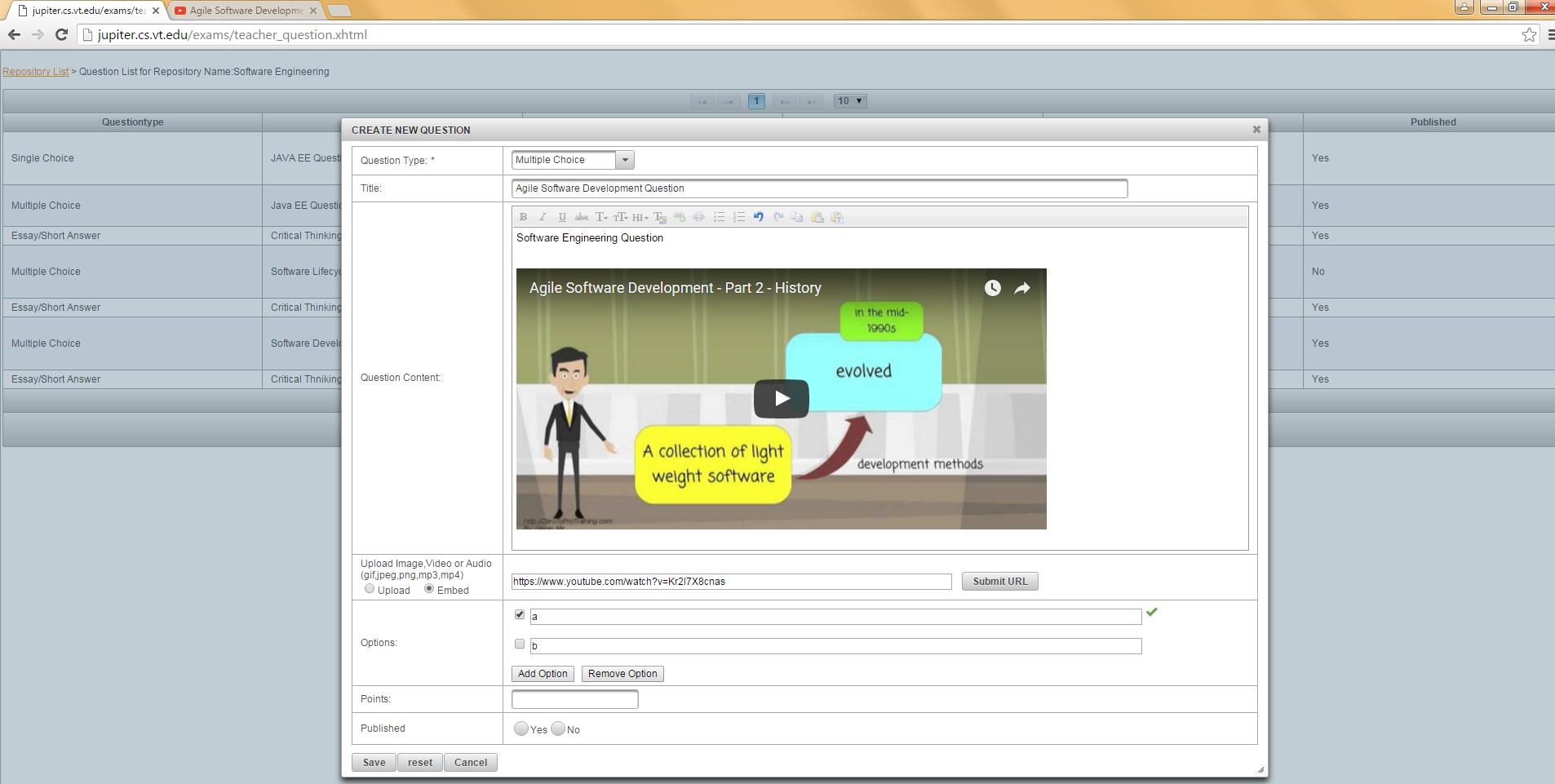
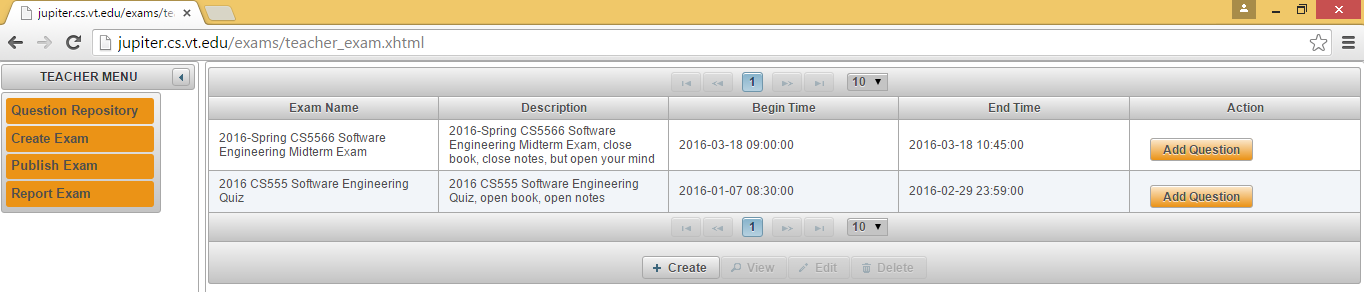


Figure 30. Question Management Interface

## Exam Creation

An instructor can create, edit, delete or view an exam anytime. The following information is 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.*** Exam Creation Interface).



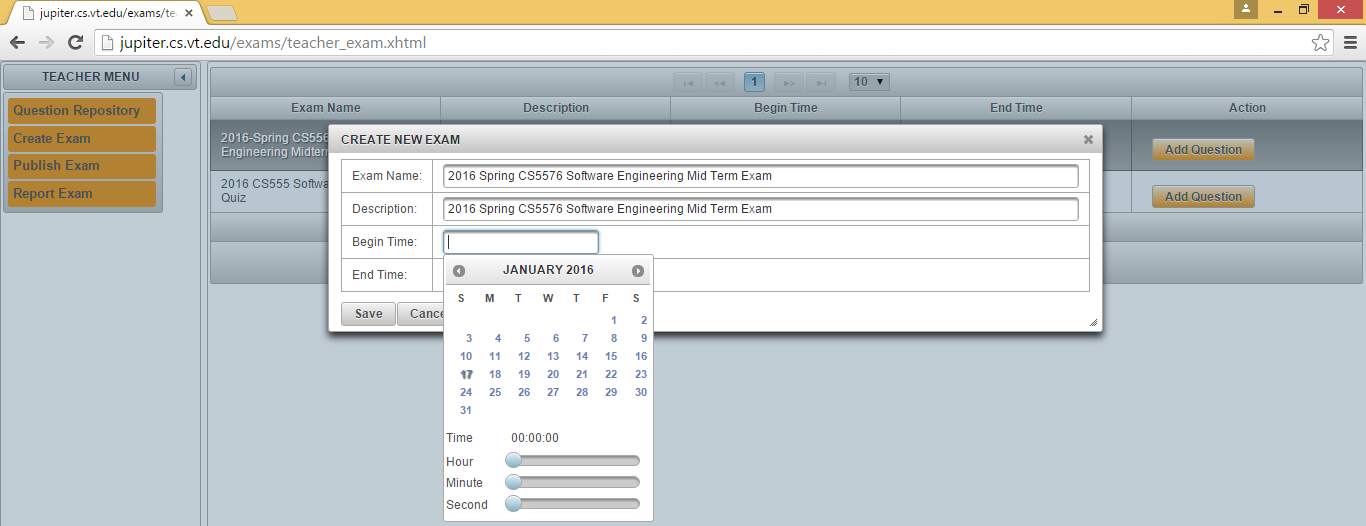
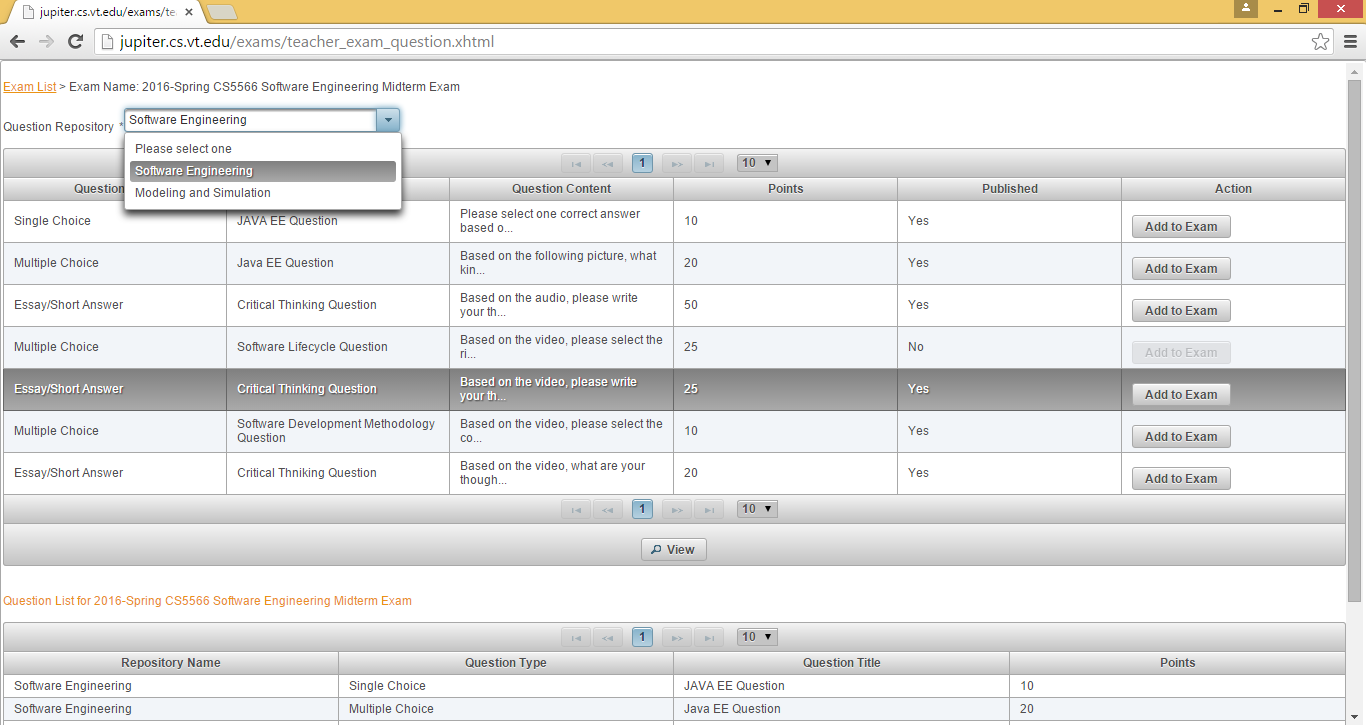


Figure 31. Exam Creation Interface

## Exam Question Management

After an exam is created, an instructor needs to select questions from the question repositories. 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 added into the exam; otherwise, the default points for that question will be used ***(Figure 32.*** Exam Question Management Interface).



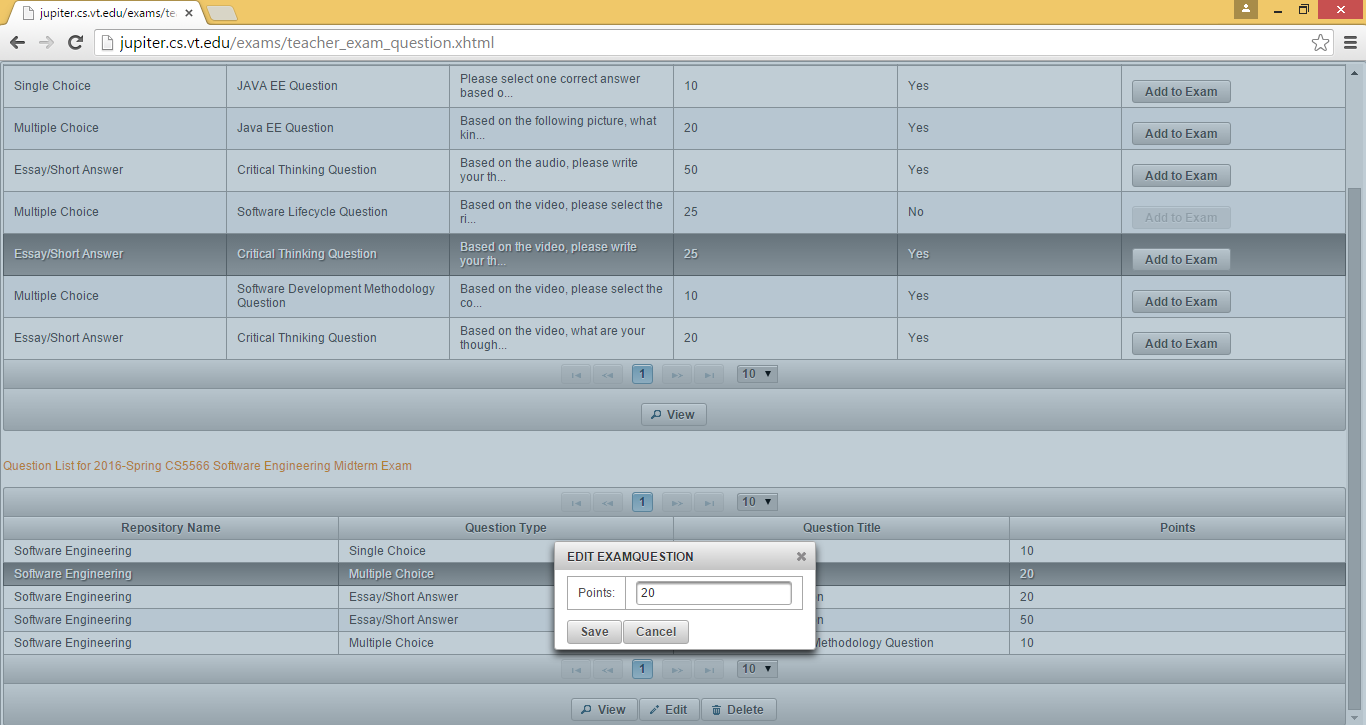
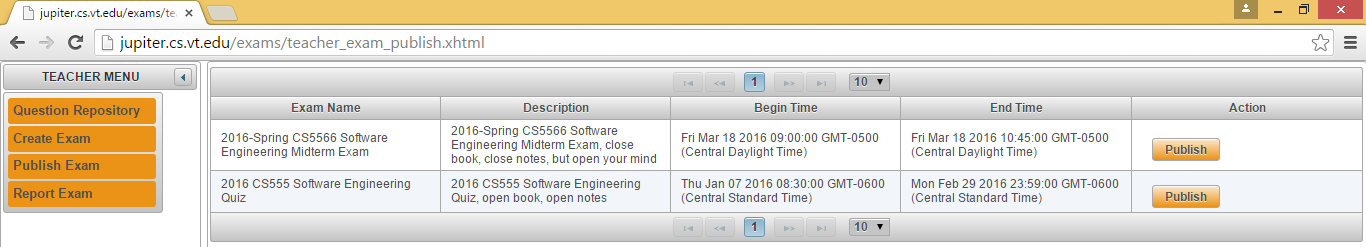
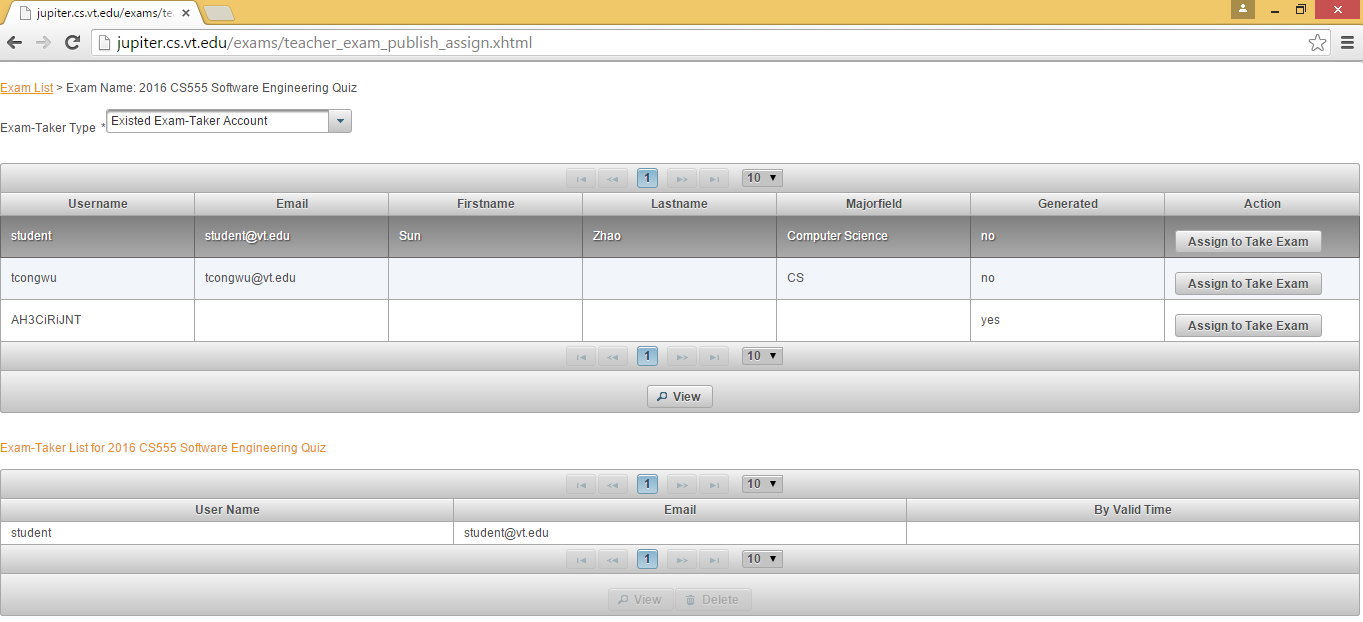


Figure 32. Exam Question Management Interface

## Publish Exam

After an exam is created by an instructor, the instructor can choose to publish the exam to existing student accounts or randomly generated student accounts. When the instructor publishes an exam to 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 period without registering the system. A generated account is locked beyond the valid period ***(Figure 33***).





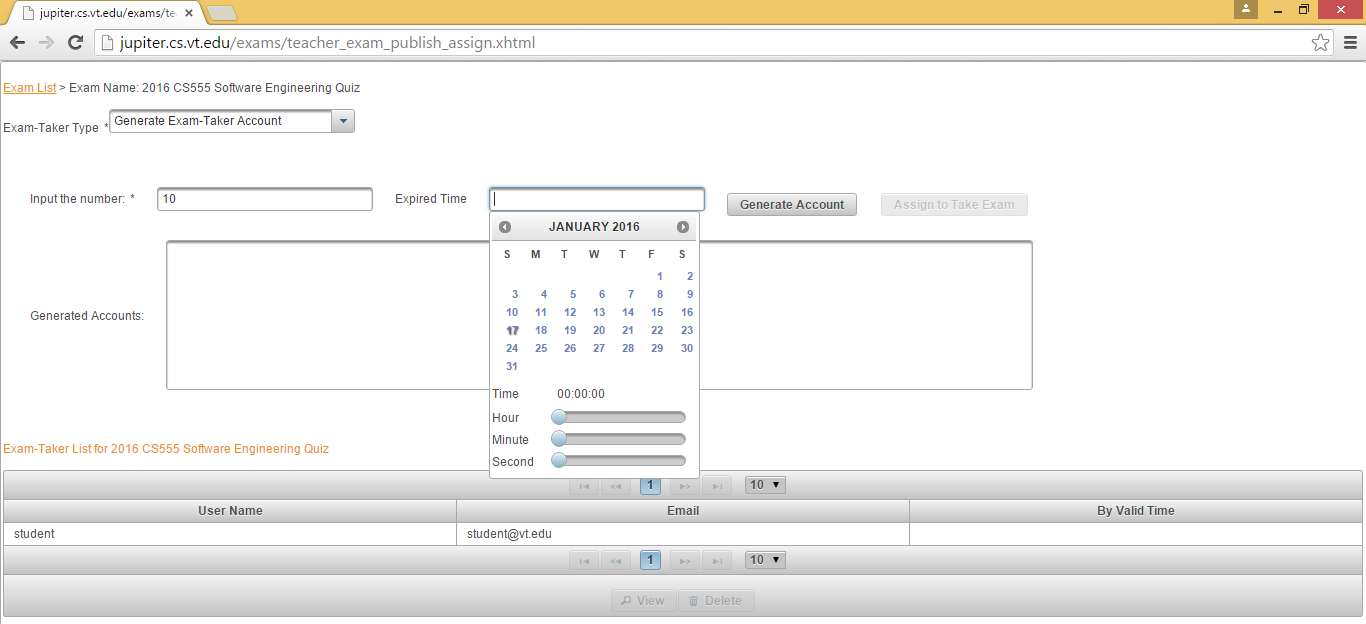
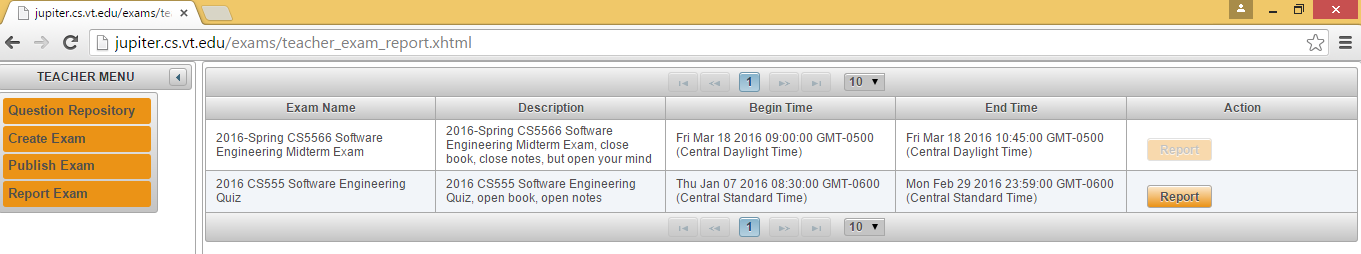
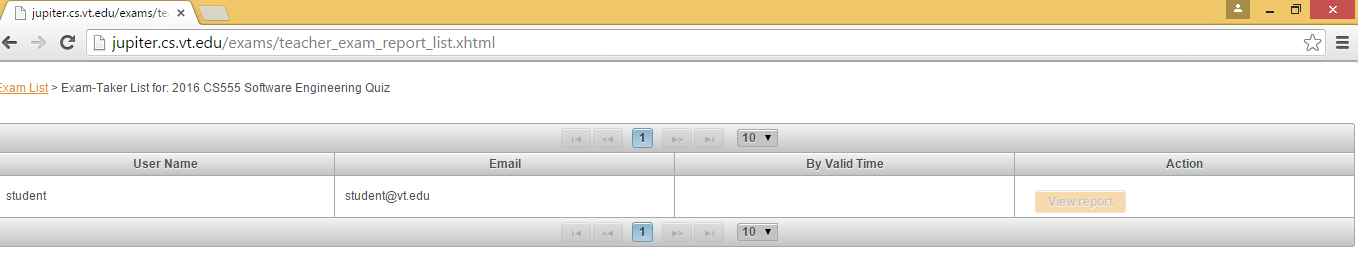


Figure 33. Publish Exam Interface

## 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 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 grade such type of questions (***Figure 34***).





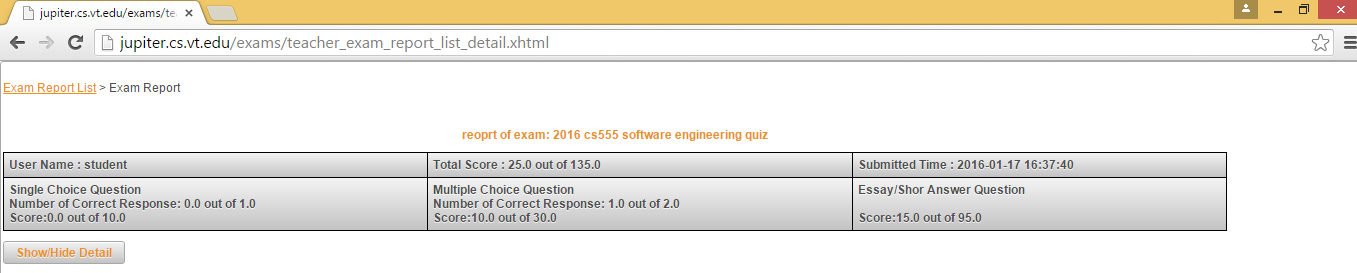
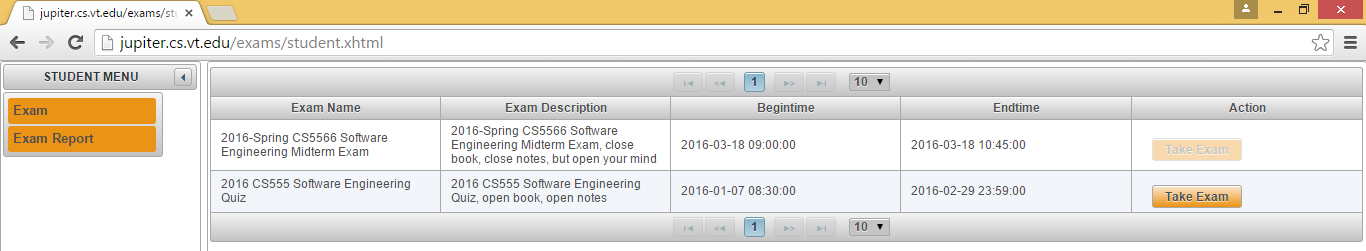




Figure 34. Report Exam Interface

## 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***).



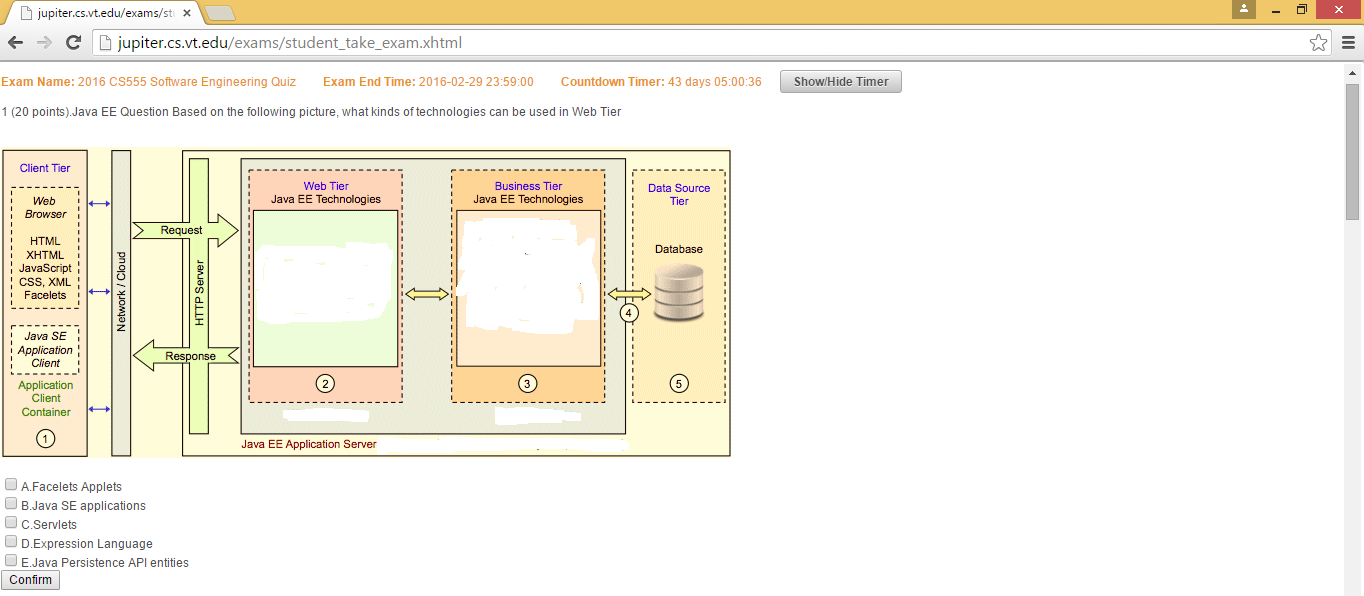
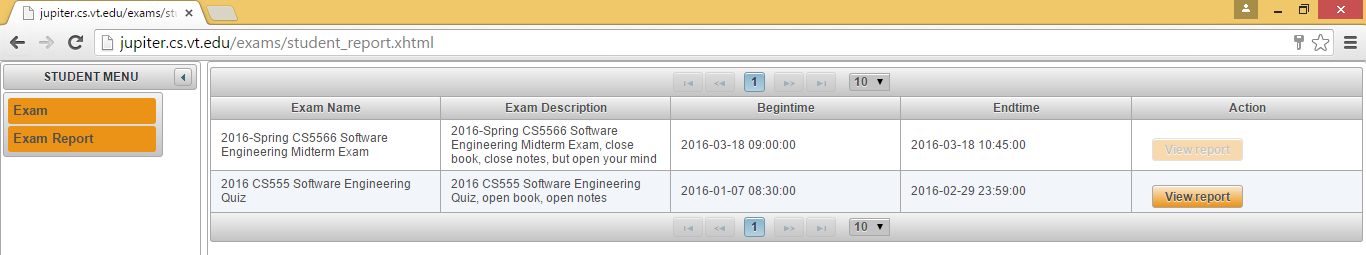


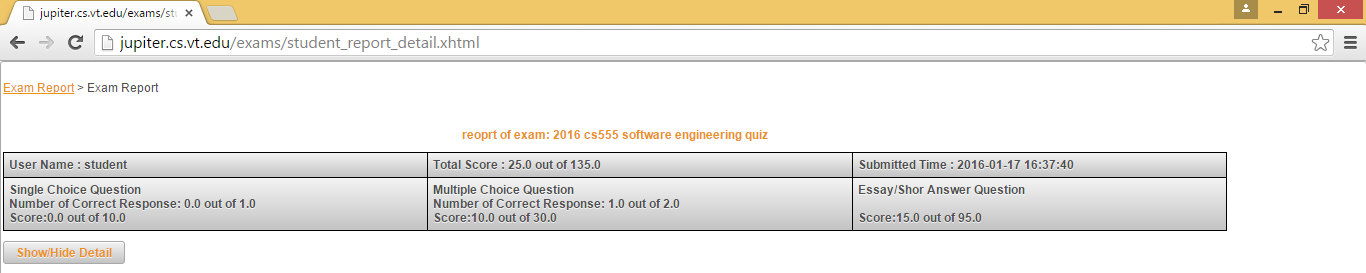


Figure 35. Take Exam Interface

## 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 summary report at the top of the web page; he or she also can click the “Show/Hide Detail” button to see the detail feedback of each question in the report (***Figure*** **36**).





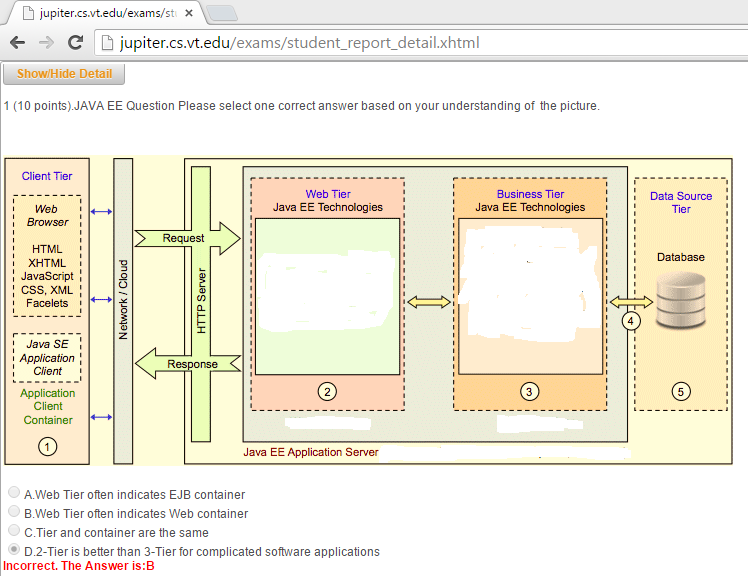


Figure 36. View Exam Report Interface

## Administrator Dashboard

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

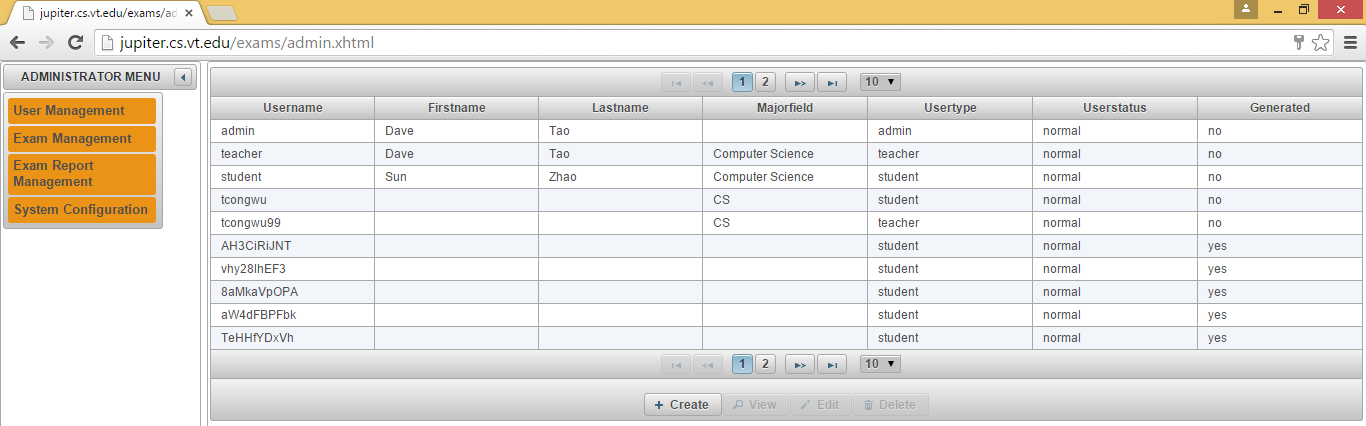


Figure 37. Administrator DashBoard Interface

# Self-Evaluation of ARENAS

This chapter discusses a self-evaluation of ARENAS in terms of these quality attributes: functionality, usability, scalability, reliability, performance, reusability and supportability [[Microsoft 2009](#Micrsoft2009)] [[Pressman 2010].](#Pressman2010)

## Functionality

Functionality defines how well a system’s features and capabilities meet customers’ needs and it can be assessed through evaluating the capabilities of the delivered program.

The capabilities were developed based on the requirement specification and the requirements were elicited from case-based requirements engineering in ARENAS. Case-based requirements engineering is considered as the best practice for creating software requirements specifications [[Balci 2014](#Balci2014)]. During the process of case-based requirement 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.

## Usability

Usability describes how well the system meets the requirements of the end users by being intuitive with an overall user-friendly experience. This attribute can be assessed by taking into account different human factors, overall consistence and aesthetics [[Pressman 2010].](#Pressman2010)

In order to make the 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 comprehensive list of JSF components emphasizing simplicity, ease of use and performance. In addition, PrimeFaces provides a myriad of theme and layouts which fully support HTML5, CSS3 and cross-browsers.

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

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

## Scalability

Scalability defines how well a system handles the increases in load without impacting its system performance, as well as how well the system can be enlarged.

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-tier architecture. Using these technologies can fundamentally guarantee this system to have 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 the third-party applications like course management system based on web-service.

## Reliability

Reliability defines how well the system can remain normal operational over time, and 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 accurate outputs of the system, the predictability of the system and the ability to recover from failure are also the features of reliability [[Pressman 2010].](#Pressman2010)

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]](#AntonioGoncalves2013) . 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 was not successful, a friendly message would be provided and lead the end users to try another possible action. 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.

## 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].](#Pressman2010)

To test the system performance, we simulate 100 hundred users to log into the system one time with JMeter [(http://jmeter.apache.org/)](file:///C:\DaveDoc\Doc\other\cs\(http:\jmeter.apache.org\)), 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; for test system performance, it is not a perfect system environment.

## Reusability

Reusability defines how well the components and subsystems can be suitable for use in other applications and in other scenarios [[Microsoft 2009](#Micrsoft2009)].

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

within each module and minimizing dependencies between modules [[Sommerville 2009](#Sommerville2009)], five subsystem components were designed and developed; there were user component, question repository component, exam component, exam report component and configuration component. The five components were loosely coupled and could be reused by any other related applications or scenarios.

## Supportability

Supportability defines how well the system can be adapted and extended. ARENAS was designed as independent cloud-based software system, and it can be easily changed, extended or adapated with API. Also, the decomposed and modularized structure of ARENAS makes it easy to be maintained.

In addition, the major programming language in this system was Java, which is an object-oriented and platform independent programming language. The features of Java programming language contribute to easy adaptment 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 Javaserver Facelets benefiting supportatlity of the system include as follows [[Antonio Goncalves 2013]](#AntonioGoncalves2013):

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

# Conclusions and Future Research

## 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](#Balci2014)]:  problem formulation, requirements engineering, architecture design (high level design), software components design and development (programming). The whole process was integrated verification and validation and quality assurance activities.

The quality attributes of the final developed system are acceptable based on self-evaluations. For instructors, the system is easy for them to create an online examination incorporating multimedia elements like image, graphic, video and audio, and it is also easy for them to assign the exam to their students whether these students have a registered account in the system. For students, they can take the online exam anywhere as long as they can connect the Internet. The exam reports can be quickly generated by the system; the instructors can tweak the exam reports or add feedback before opening the exam reports to their students and the students can easily view the exam reports after they are open. For administrators, they can easily manage and monitor the whole process.

## Contributions

The contribution of this study is the cloud-based software solution for multimedia online examinations, which utilizes the multi-tiered client-server architecture and cloud-based technology. This software solution can provide a large-scale online assessment and allow instructors to effectively design engaging online questions with multimedia elements, as well as to 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 developing other large-scale cloud-based educational software systems for educators and researchers.

## 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. Provide 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. Provide more question types when creating online questions for instructors to use in an online examination.
3. Support more file-extensions of multimedia elements when creating online questions for instructors to use in an online examination.
4. Improve the interface by providing different themes to allow the users to select the most appropriate interface for them.
5. Improve the functionality of creating questions by allowing instructors to use more third-party video-provider platform like Vimeo.
6. Improve the functionality of managing assessment criteria by allowing instructors to set flexible assessment criteria like assessment rubrics for short-answer or essay questions.
7. Integrate the third-party tools such as Respondus LockDown Browser and Respondus Monitor from <https://www.respondus.com/> into the system for helping reduce the probability of student online exam cheating.

REFERENCES

Balci, O. (2016), “CS5704 Software Engineering Course Website,” Department of Computer Science, Virginia Tech, Blacksburg, VA, <http://manta.cs.vt.edu/cs5704/>

Balci, O (2009), “A Tutorial on How to Create UML Diagrams,” Department of Computer Science, Virginia Tech, Blacksburg, VA, <https://manta.cs.vt.edu/cs5704/Handouts/UML%20Diagrams%20Tutorial.pdf>

Edwards, S. (2017), “CS5744 Software Design and Quality,” Department of Computer Science, Virginia Tech, Blacksburg, VA, <https://canvas.vt.edu/courses/53605>

Hartson, R. (2017), “CS5714 Usability Engineering,” Department of Computer Science, Virginia Tech, Blacksburg, VA, <https://canvas.vt.edu/courses/53605>

Pressman, R. (2014), Software Engineering, A Practitioner's Approach, 8th edition, McGraw- Hill Higher Education, New York, NY

Fairbanks, G. (2010), Just Enough Software Architecture, A Risk Driven-Approach, 1st edition, Marshall & Brainerd

Pfleeger, S. (2010), Software Engineering, Theory and Practice, 4th edition, Pearson, Hoboken, NJ

Hartson, R. (2012), The UX Book: Process and Guidelines for Ensuring a Quality User Experience*, 1st edition*, Morgan Kaufmann, Burlington, MA

Schutt K. and Balci O. (2016), “Cloud Software Development Platforms: A Comparative Overview”, Software Engineering, IEEE

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 III: DoDAF Meta-model Physical Exchange Specification - Developer’s Guide,” Architecture Framework Working Group, Washington, DC.

Chigani A. and Balci O. (2011), “Campus Situational Awareness and Emergency Response Management System,” Virginia Tech, Blacksburg, VA, <https://manta.cs.vt.edu/sinergy/index.html>

Microsoft (2009), “Microsoft Application Architecture Guide”, Microsoft Corporation, Redmond, WA <https://msdn.microsoft.com/en-us/library/ff650706.aspx>

Dennis A., Roth R., and Wixom, B. (2012). *Systems Analysis and Design with UML*. John Wiley & Sons, Inc.

Volunteering Related Publications & Books:

<http://journals.sagepub.com/doi/abs/10.1177/0899764011434558>

<http://su8bj7jh4j.search.serialssolutions.com/?sid=sage&iuid=307528&aulast=&date=2010&atitle=Who+gives+what+and+when%3F+A+scenario+study+of+intentions+to+give+time+and+money&title=Social+Science+Research&volume=39&spage=369&id=doi:10.1016%2Fj.ssresearch.2009.08.008&>

<http://su8bj7jh4j.search.serialssolutions.com/?sid=sage&iuid=307528&aulast=&date=2010&atitle=Social+networks%2C+recruitment%2C+and+volunteering%3A+Are+social+capital+effects+conditional+on+recruitment%3F&title=Nonprofit+and+Voluntary+Sector+Quarterly&volume=40&spage=476&>

<http://www.sciencedirect.com/science/article/pii/S1441352306700237>

<http://psycnet.apa.org/record/2008-12803-001>

<https://books.google.com/books?hl=en&lr=&id=jVgGAQAAQBAJ&oi=fnd&pg=PA73&dq=volunteer+recruitment&ots=hILWO8huyy&sig=OtmTdBhwfm4wUspT5aU-T_-RG4Y#v=onepage&q=volunteer%20recruitment&f=false>

Volunteering Related Work Commercial Applications & used APIs:

[ClassMarker](http://www.classmarker.com) (2015), “ClassMarker website(VolunteerMatch),” ClassMarker Pty Ltd, <http://www.classmarker.com>

ExamBuilder (2015), “ExamBuilder website,” ExamBuilder Company, <http://www.exambuilder.com>

ExamSoft (2015), “ExamSoft website,” ExamSoft Worldwide, Inc., <http://www.examsoft.com>

Interesting and related but not cited:

Sommerville, I. (2009). (Requirements)*Software engineering*. Boston: Pearson/Addison-Wesley.

McGovern, J., Ed. (2004). (Arquitecture)*A practical guide to enterprise architecture*. Prentice Hall Professional.

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/>

Weilkiens, T. (2011). *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Morgan Kaufmann.

Oracle (2015), “Java EE at a Glance,” Oracle Corporation, Redwood Shores, CA, <http://www.oracle.com/technetwork/java/javaee/overview/>

http://fontawesome.io/

https://www.pexels.com/

https://pixabay.com/