Quiz Maker

System Design

<V 1.0>

<02.12.2018>

Atilla Özder

Mert Çetinkaya

Begüm Mina Bilgin

Selin Atalay

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc433996772)

[1.1. Purpose of the System 1](#_Toc433996773)

[1.2. Design Goals 1](#_Toc433996774)

[1.3. Definitions, Acronyms, and Abbreviations 1](#_Toc433996775)

[1.4. References 1](#_Toc433996776)

[2. Current Software Architecture 1](#_Toc433996777)

[3. Proposed Software Architecture 1](#_Toc433996778)

[3.1. Overview 1](#_Toc433996779)

[3.2. System Decomposition 1](#_Toc433996780)

[3.3. Hardware Software Mapping 2](#_Toc433996781)

[3.4. Persistent Data Management 2](#_Toc433996782)

[3.5. Access Control and Security 2](#_Toc433996783)

[3.6. Global Software Control 2](#_Toc433996784)

[3.7. Boundary Conditions 2](#_Toc433996785)

[4. Subsystem Services 2](#_Toc433996786)

[5. References 2](#_Toc433996787)

SYSTEM DESIGN DOCUMENT

# Introduction

Design is the abstraction of an answer; it's the common description of the answer to an issue without a details. Design is view sample seen within the evaluation part to be a sample in a design part. After design part we are able to reduce the time required the implementation.

## Purpose of the System

As said within the Requirements Analysis Doc, the aim of the system is to offer system administrator, instructors, student and users with a central location for organizing varied quiz occasions. The aim of this technique supplies the next causes: No physical presence wanted for quizzes, no wastage of time throughout analysis, prompt availability of the outcomes, beginning and completion time are each recorded by the server, a collection of on-line assessments are provided by the software program for the good thing about the scholars. Normally the system is far more preferred as a consequence of these causes across various organizations and the world.

## Design Goals

The design objectives signify the specified qualities of quiz maker system and supply a constant set of standards that should be thought of when making design choices. Based on non-functionality requirements the next design targets must achieved as a way to qualify the system as profitable:

* **Security**

The system security is one of the most important non-functional requirements.

* **Reliability**

The system has to carry out the quiz maker operations with no errors. The application developed needs to be extremely dependable and safe in order that details about any questions and many others shouldn’t be leaked earlier than the precise quiz is held.

* **End User**

Students can take quiz only throughout the previously allotted time slots, however can open application anytime to access different data. System should be able to deal with a number of users. This system should run IOS operating system.

* **Performance**

The system has to be sturdy enough to manage any valid input from the users.

* **Flexibility:**

It requires Internet.

* **Implementation:**

System should be modified and readable. The number of errors must be minimized.

## Definitions, Acronyms, and Abbreviations

**Student**: The system user who will attends to the quiz. At the same time will be able to create the quiz.

**Admin**: The system administrator who will manage all data system data and user controls.

**Instructor**: The user who will prepare and present the questions to the students and calculate their scores by entering the answers of the questions.

**Registered User:** Only public quizzes are allowed. He/ She will be create the public quiz.

**Visitor:** Visitor can only register and than login in the application.

## References

- There are many quiz maker applications. For example; <https://www.quiz-maker.com/>

<https://www.easypromosapp.com/quiz/>. But our system some different from others.

# Current Software Architecture

There are some websites and applications which have similar purposes as our system. These systems are allowing users to create or join some quizzes. The applications mentioned are iSpring QuizMaker[2], Easypromos Quiz[3] and Typeform Quiz[4], and the websites are Quiz-Maker[5] and SurveyMonkey[6]. iSpring QuizMaker is made for measuring the competency levels of staff members. Easypromos Quiz is made for testing the followers’ knowledge of a brand and products or a specific topic using a quiz. Typeform Quiz is based on business growth, personal or professional growth and it is designed to teachers or corporate trainers helping people with new knowledge and skills. The website Quiz-Maker is a site where you can start building your own quizzes at the first page you see, you can prepare your own quizzes and share it with anyone you’d like to share. SurveyMonkey is a website where you can prepare your own surveys and share it with the people you want, it can be used like our system but the main idea is different. There is no system like our QuizMaker System which is based on education and providing instructors to create lecture quizzes to the students which are related with those classes and also to the registered user to create quizzes by their own and to make people join to the quizzes they have prepared.

# Proposed Software Architecture

We suppose a system that contains design pattern. According to this design pattern, we do clearly implementation in addition, the strategy design pattern leads to our project. Finally, maintenance and operate become easier and proper.

Also, we suppose a system that has been implemented the layered architecture. According this the layered architectural is a hierarchy of layer, each layer using services offered by the lower layers. We implement our project the layered architecture, all of functions are called from services. Files of webservices contains all functions that are called web-service. All methods will be here.

The system is object-oriented. Therefore, there is model so, database of our system will be query-based. With completely, this is back-end of the project. In front-end, we have logical page. These page use web-services with page way.

## Overview

The system is user friendly system and all user can be used the system easily. The system has four interfaces.

**Admin Interface:** It provides services to manage for system. They lists such as add course, delete course, approve instructor, reject instructor, delete quiz, delete user (instructor, student, registered user).

**Instructor Subsystem:** It provides services that they must be at login, create lecture quizzes, update lecture quizzes, delete lecture quizzes, add students, edit profile and change password.

**Student Subsystem:** It provides services that join lecture quizzes, create public quizzes, join public quizzes, edit profile, update public quizzes and change password.

**Registered User Subsystem:** It provides services that create public quizzes, join public quizzes, edit profile, update public quizzes and change password.

**Visitor Subsystem:** It provides view quizzes time and name.

**Login Component:**  It provides services for system administrator, instructor, student or registered user in to the system.

**Registered Component:** It provides service for visitor to register the system.

**Create Lecture Quiz Component:** It provides instructor add quiz into the system.

**Update Lecture Quiz Component:** It provides instructor update quiz for students into the system.

**Delete Lecture Quiz Component:** It provides instructor and admin delete quiz for students on the system.

**Add Student Component:** It provides instructor add student for course on the system. Instructor can search that user name, email or student id for add student on the system.

**Edite Profile Component:** It provides every user changes own information that is first name, last name and e-mail on the system.

**Change Password Component:** It provides every user changes password into the system. (except visitor).

**Join Lecture Quiz Component:** It provides student for lecture quiz. Student join quiz on the system.

**Create Public Quiz Component:** It provides student and registered user. There are creating public quiz on the system.

**Join Public Quiz Component:** It provides student and registered user that join public quiz on the system.

**Update Public Quiz Component:** It provides student and registered user that update quiz into the system.

## System Decomposition

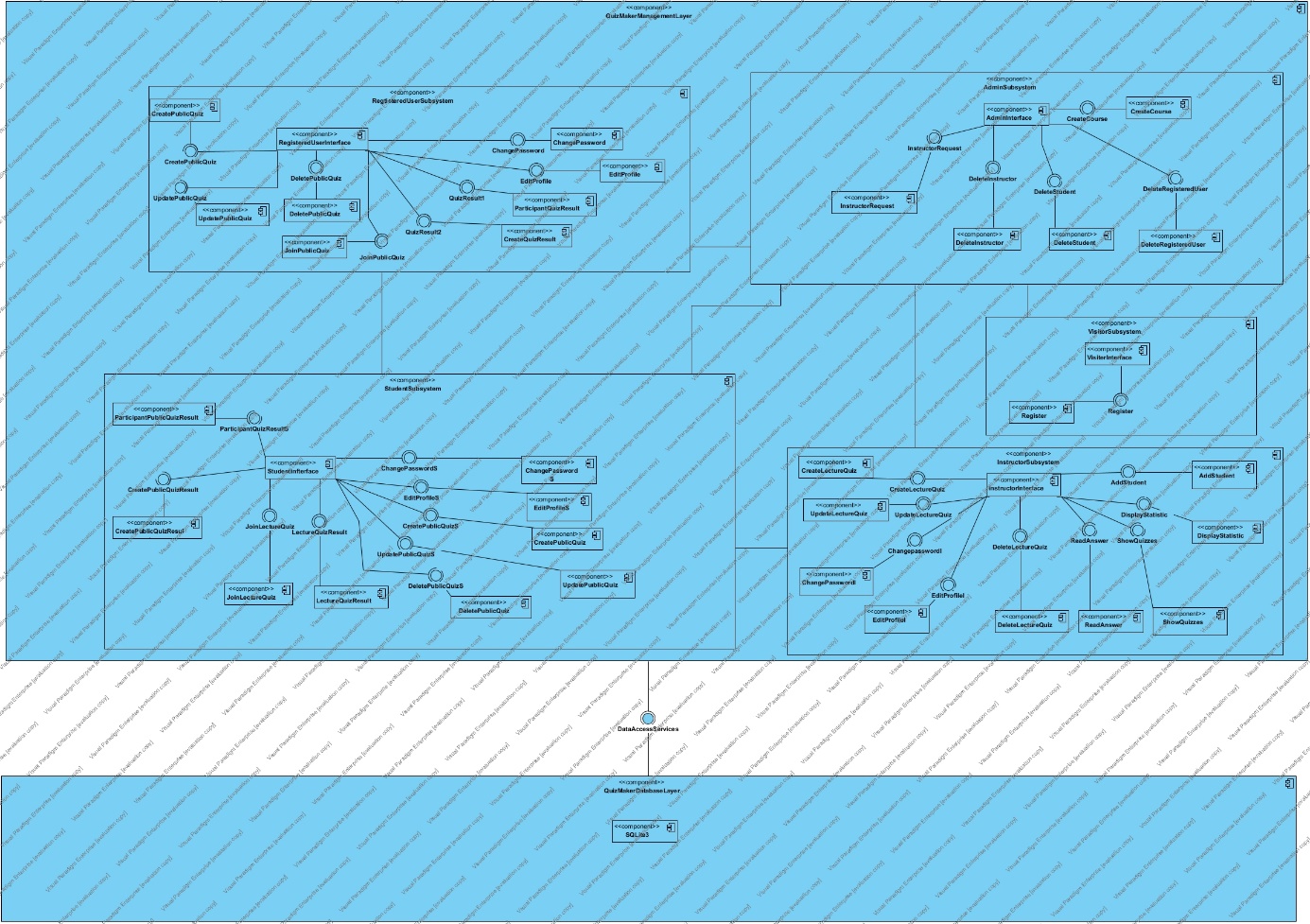


Figure 1: Coupling view of Subsystem Decomposition

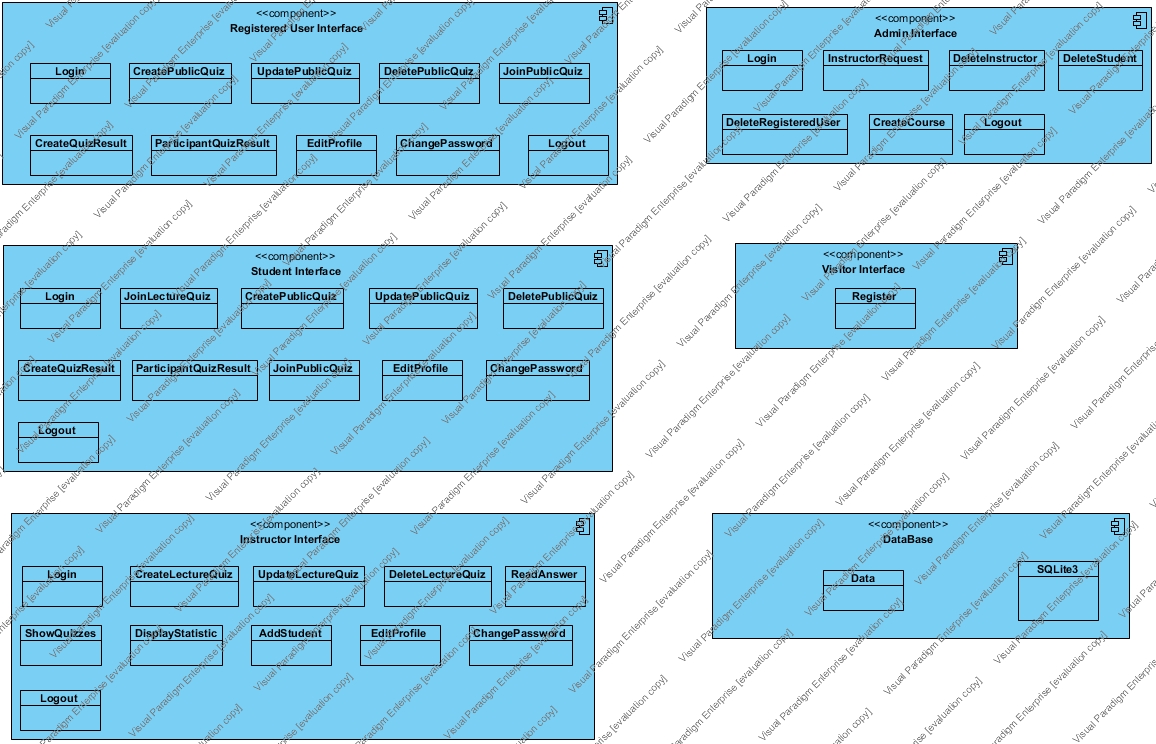
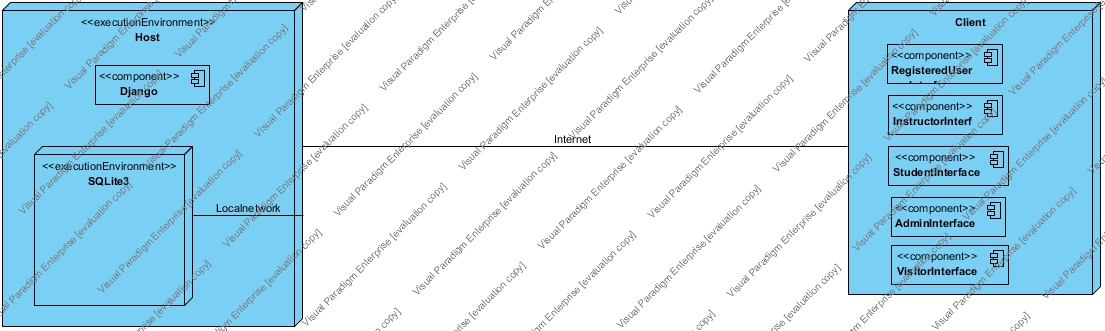
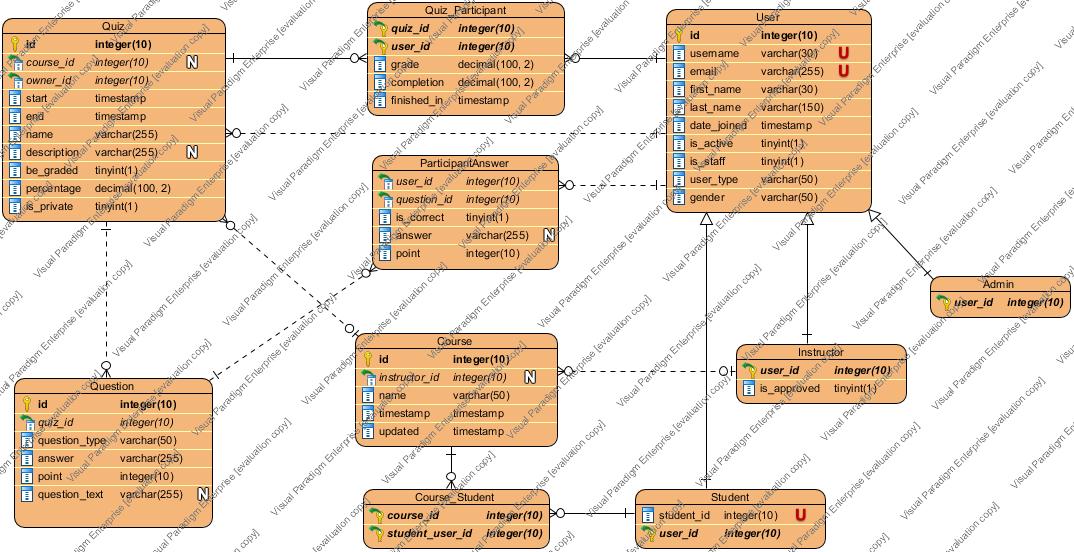


Figure 2: Cohesion view of Subsystem Decomposition

## Hardware Software Mapping



## Persistent Data Management



***User, Instructor, Student, Admin***

This tables stores all information about user. User tables has ten columns that is id, username, email, first\_name, last\_name, date\_joined, is\_active, is\_staff, user\_type and gender. id is an integer unique primary key it is generated by system. username and email are the unique ones.

***Course***

Course table stores all information about course. The table has five columns that is id, instructor,\_id, name, timestamp, updated. id is an integer unique primary key and instructor\_id is an integer foreign key that isn’t null.

***Question***

Question table stores all information about question. The table has five columns that is id, quiz\_id, question\_type, answer, point and question\_type. id is an integer unique primary key and quiz\_id is integer foreign key.

***Quiz***

Quiz tables stores all information about quiz. The table has ten columns that is id, course\_id, owner\_id, start, end, name, description, be\_graded, percentage, is\_private. id is an integer unique primary key. course\_id is an integer foreign key from course tables. owner\_id is an integer foreign key from user table.

## Access Control and Security

Quiz maker is a mobile application running with swift. It is created with object-oriented programing so, there are classes. It is classical mobile application. It is a multiuser application that are more than one actor. The actors are admin, instructor, student, registered user and visitor.

There has been objective oriented architecture so, we implemented classes that are organized by actor. In this project, we are setting the role in database. When a registered person tries to log in to the system, if person makes successful login, person will be redirected to main page as person role. Actually, person accesses services by actor role.

Person access to the system on unique define e-mail, user name and password at registration case.

## Global Software Control

The system controls the authentication with defined unique mail, username and password. The synchronization is satisfied among queries on server. Thus, dynamic content will be displayed.

## Boundary Conditions

Quiz Maker is active by the system admin that uploaded app store this project. Upload stage does project compiler and our select to upload on app store after the upload project, we do some changes for project and we take a screen shot for an upload app store and we published on app store. Now this project waiting for a confirmation who is app store officials. App store officials approve project after this project will publish app store.

The errors could be originated from attempting to login with invalid e-mail or password or trying to update the personal information with empty information. All these exceptions are getting caught by the system and handled properly.

When the system is terminated by system our remove application on app store. All user that has download in app store continue to use this application but these users don’t get an update in addition, others users don’t download this application.

# Subsystem Services

The subsystem decomposition of Quiz Maker application, we divide the system into smaller subsystem with strong coherence.

The subsystem separation shows the entities of following subsystem

* Visitor
* Registered User Subsystem
* Instructor Subsystem
* Student Subsystem
* Admin Subsystem
* Database Subsystem

**Registered User Subsystem**

This subsystem manage quiz activity with user type is registered user.

This subsystem provided by:

* Login
* CreatePublicQuiz
* UpdatePublicQuiz
* DeletePublicQuiz
* EditProfile
* ChangePassword
* ShownAnswer
* DisplayAnswer
* Logout

**Instructor Subsystem**

This subsystem provides to manage quiz activity for student.

This subsystem provided by:

* Login
* CreateLectureQuiz
* UpdateLectureQuiz
* DeleteLectureQuiz
* EditProfile
* ChangePassword
* ReadAnswer
* AddStudent(with search student)
* ShownAnswer
* DisplayAnswer
* Logout

**Student Subsystem**

This subsystem provides to manage quiz activity and participant quiz activity.

This subsystem provided by:

* Login
* JoinLectureQuiz
* CreatePublicQuiz
* JoinPublicQuiz
* EditProfile
* ChangePassword
* QuizResultforCreated
* ShownAnswer
* DeletePublicQuiz
* Logout

**Admin Subsystem**

This subsystem provides to manage all system.

This subsystem provided by:

* Login
* CreateCourse
* UpdateCourse
* DeleteCourse
* ApproveInstructor
* RejectInstructor
* DeleteRegisteredUser
* DeleteStudent
* DeleteInstructor
* DeleteQuiz
* Logout

**Visitor Subsystem**

This subsystem provides to registeractivity.

This subsystem provided by:

* Register

# References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. <https://www.ispringsolutions.com/ispring-quizmaker>
3. <https://www.easypromosapp.com/quiz/>
4. <https://www.typeform.com/quizzes/>
5. <https://www.quiz-maker.com/>
6. <https://www.surveymonkey.com/mp/quiz/>