Quiz Maker

Requirements Specification and Analysis

<Version>

09.11.2018

Atilla Ozder

Mert Çetinkaya

Begüm Mina Bilgin

Selin Atalay

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc496873294)

[1.1. Purpose of the System 1](#_Toc496873295)

[1.2. Scope of the System 1](#_Toc496873296)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc496873297)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc496873298)

[1.5. Overview 1](#_Toc496873299)

[2. Current System 1](#_Toc496873300)

[3. Proposed System 1](#_Toc496873301)

[3.1. Overview 1](#_Toc496873302)

[3.2. Functional Requirements 2](#_Toc496873303)

[3.3. Nonfunctional Requirements 2](#_Toc496873304)

[Usability 2](#_Toc496873305)

[Reliability 2](#_Toc496873306)

[Performance 2](#_Toc496873307)

[Supportability 2](#_Toc496873308)

[Implementation 2](#_Toc496873309)

[Interface 2](#_Toc496873310)

[Packaging 2](#_Toc496873311)

[Legal 2](#_Toc496873312)

[3.4. System Models 2](#_Toc496873313)

[Scenarios 2](#_Toc496873314)

[Use case model 2](#_Toc496873315)

[Object model 2](#_Toc496873316)

[Dynamic model 2](#_Toc496873317)

[User interface—navigational paths and screen mock-ups 3](#_Toc496873318)

[3.5. Project Schedule 3](#_Toc496873319)

[4. Glossary 3](#_Toc496873320)

[5. References 3](#_Toc496873321)

REQUIREMENTS ANALYSIS DOCUMENT[1]

The Requirement Analysis Document (RAD) should be written after the use case model is stable, that is, when the number of modifications to the requirements is minimal. The requirements, however, are updated throughout the development process when specification problems are discovered or when the scope of the system is changed.

Please set your word processor’s language to English, enable spell checker to detect the misspellings, and follow the formatting in this document.

# Introduction

The purpose of this section is to provide a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context (e.g., reference to the problem statement written by the client, references to existing systems, feasibility studies). The introduction also includes the objectives and success criteria of the project[1].

## Purpose of the System

## Scope of the System

## Objectives and Success Criteria of the Project

## Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the RAD.

## Overview

This subsection should:

* Describe what the rest of the RAD contains
* Explain how the RAD is organized.

# Current System

If the new system will replace an existing system, this section describes the functionality and the problems of the current system. Otherwise, this section describes how the tasks supported by the new system are accomplished now.

# Proposed System

Documents the requirements elicitation and the analysis model of the new system

## Overview

Presents a functional overview of the system.

## Functional Requirements

Describes the high-level functionality of the system.

## Nonfunctional Requirements

Describes user-level requirements that are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, interface, operational, packaging, and legal requirements.

### Usability

### Reliability

### Performance

### Supportability

### Implementation

### Interface

### Packaging

### Legal

## System Models

Describes the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

### Scenarios

Scenario Name: Login

Participating Actor : Mert: User

**Flow of Events:**

1. He is at login page screen .He enterd email and passsword so, he login button click.
2. User can see 3 option;
3. If user leave blank any area that is email/user name or password, user sees message that Usernema or password recheck please.
4. If user entered false email/user name or password, , user sees message that Usernema or password recheck please.
5. İf user dont do any mistake, user login in system.
6. He clicks logo ut button and exit account and he returns login page.

Scenario Name: Log out

Participating Actor : Atilla: User

**Flow of Events:**

1. He is at login page screen and he entered email/username and password and click “Log In”

Button.

1. He is click log out button, and he is exit system in addition he returns login page.

Scenario Name: Join Quiz

Participating Actor : Begüm: User

**Flow of Events:**

1. She is login at login page screen and she goes home page.

2.) She sees home page for user.She click “public quiz” button after she sees list of public quiz. She selects one quiz after she join a quiz. She starts answer to question. She finishs quiz she return list of public quiz page.

1. He is click log out button, and he is exit system in addition he returns login page.

Scenario Name: Create Quiz and Search User

Participating Actor : Selin: User

**Flow of Events:**

1. She is login at login page screen and she goes home page.
2. She click create quiz and she goes create create quiz page.
3. She selects a question type and she enters to correct answer in additon she entered question.
4. She as much as question number replys.
5. She selects start time and end time.
6. If she searchs user that is user name/email. She sees user profil and she adds user fort his quiz.
7. She click start quiz button and quiz will start.

Scenario Name: Edit Profile

Participating Actor : Deniz: User

**Flow of Events:**

1. She is login at login page screen and she goes home page.
2. She clicks profil button.
3. She sees that information of own.she clicks edit button, so she changes user email, password, first name and last name or gender.
4. She clicks saves button and amendents save, after she returns profil page.
5. She click home page button, so she click log out button. She will log out of system.

Scenario Name: Result Quiz

Participating Actor : Ali : User

**Flow of Events:**

1. He is login at login page screen and she goes home page.
2. He clicks Result Quiz of Participation button click.
3. He sees list of participation quiz. He selects any quiz.
4. He sees which question is true or false. If question is classic, he will get point of question.
5. He clicks home page button and he clicks log out button, so its log out of system.

Scenario Name: Result Quiz2

Participating Actor : Veli : User

**Flow of Events:**

1. He is login at login page screen and she goes home page.
2. He clicks My Create Quiz Result button click.
3. He clicks finished quiz button.
4. He sees list of participation quiz.
5. He selects any quiz.
6. He sees who is participation, answer of question, percent of correct answer.
7. He clicks home page button and he clicks log out button, so log out of system.

Scenario Name: Edit Quiz

Participating Actor : Veli : User

**Flow of Events:**

1. He is login at login page screen and she goes home page.
2. He clicks My Quizzes Result button click.
3. He clicks not started quiz button.
4. He sees list of not started quiz. He will changes quiz question or quiz start or end date.
5. He click homa page button and he clicks log out button, so he log out of system.

Scenario Name: Delete Quiz

Participating Actor : Veli : User

**Flow of Events:**

1. He is login at login page screen and he goes home page.
2. He clicks My Create Quiz Result button click.
3. He clicks not started quiz or finished quiz button.
4. He sees list of quiz.
5. He wants to Delete “ Quiz1” and He clicks delete button that is near by quiz name Delete button.
6. System asking a user is “ Are you sure” and system shows yes or no button.
7. He clicks yes button and he deletes Quiz1.
8. He clicks home page button and he clicks log out system, So he log out of system.

Scenario Name: Register

Participating Actor : Ahmet : User

**Flow of Events:**

1. He is at login page. He wants to login in to system but he hasnt any account for Quiz Maker.
2. He clicks register label. He entered information of name, e-mail, password for this application and he selects any user name this application.He clicks register button and;
3. He enteres wrong/false/used e-mail address, so he sees error message and he returns register page.
4. He enteres used user name or select bad user name, so user sees error message and returns register page.
5. He enteres short password or not entered any password, so he sees error message and returns register page.
6. He will register.

|  |
| --- |
| *Scenario Name:* AddCourse |
| *Participant actor instanecs:* Begüm: Admin |
| *Flow of events:*  1. Begüm opens the application and logs in as admin. She sees AddCourse button at the first page she arrived by logging in.  2. Begüm clicks on the AddCourse button.  3. She selects the course’s name from the given course list which has all the possible classes that can be added to the system and clicks on the “Add” button.  4. There comes a notification saying “Course is added successfully.” and sees that the course is added to the system. |

|  |
| --- |
| *Scenario Name:* CreateLectureQuiz |
| *Participant actor instanecs:* Atilla: Instructor |
| *Flow of events:*   1. Atilla opens the application and logs in as instructor. He sees CreateLectureQuiz button at the first page he arrived. 2. Atilla clicks on the CreateLectureQuiz topic. 3. He picks the course’s name from the course list and enters the topic of the quiz and continues with creating the questions by the default template of the system allows him to do by clicking on the “Continue” button. 4. He picks one of the question types such as classical questions, true/false questions, fill in the blanks questions, multiple choice questions and writes the questions he wants to ask and he clicks on the “Create Quiz” button. 5. System asks him if he is sure creating the quiz or returning back to editing, he picks “Yes” if he is done editing or “No” to return back editing the questions. 6. If he clicks on the “Yes” button, there comes a notification saying “Creating the quiz is completed successfully.” and sees the quiz is added to the system. |

|  |
| --- |
| *Scenario Name:* UpdateQuizzes |
| *Participant actor instanecs:* Mert: Instructor |
| *Flow of events:*   1. Mert opens the application and logs in as instructor. 2. He clicks on “MyQuizzes” session from the page he arrived. 3. He can only update thouizzes which are not yet started. 4. If he tries to update a quiz which is already started or finished, he sees an error message saying “You can’t update a quiz which is already started or finished.” 5. After he updates the quizzes, he sees a notification saying “Updating the quiz is successful”. |

|  |
| --- |
| *Scenario Name:* DeleteQuizzes |
| *Participant actor instanecs:* Begüm: Instructor |
| *Flow of events:*   1. Begüm opens the application and logs in as instructor. 2. She clicks on “MyQuizzes” session from the page she arrived. 3. She can only delete those quizzes which are not yet started. 4. If she tries to delete a quiz which is already started or finished, she sees an error message saying “You can’t delete a quiz which is already started or finished.” 5. If she tries to delete those quizzes which are not yet solved, she clicks on the Delete button. 6. Then she sees a notification saying “Deleting the quiz is successful”. |
|  |
| *Scenario Name:* ShowQuizzes |
| *Participant actor instanecs:* Mert: Instructor |
| *Flow of events:*  1. Mert opens the application and logs in as instructor.  2. He clicks on “MyCourses” session and opens up a page that he can see the list of courses.  3. After Mert displays the list of courses he clicks on “Details” to see the students who took the quizzes.  4. After he clicks on one of the students’ names, he sees the list of quizzes he made of the student he chose. |

|  |
| --- |
| *Scenario Name:* ReadAnswers |
| *Participant actor instanecs:* Begüm: Instructor |
| *Flow of events:*   1. Begüm opens the application and logs in as instructor. 2. She clicks on “MyCourses” session and opens up a page that she can see the list of courses. 3. After seeing the list of courses, Begüm picks a course to display the students which she wants to see the answers of. 4. After she clicks on one of the students’ names, she sees the list of quizzes she had of the course she chose. 5. After she displayed the list of quizzes, she clicks on the quiz which she wants to display. 6. After she displays the quiz of the student she picked, she decides the true answer for those which are classical questions, and responds as true or false. 7. After responding the quizzes she clicks on the “Save” button to save the results. 8. Then she sees a notification saying “Changes Saved” as a feedback. |

|  |
| --- |
| *Scenario Name:* AddStudent |
| *Participant actor instanecs:* Atilla: Instructor |
| *Flow of events:*   1. Atilla opens the application and logs in as instructor. 2. He clicks on “MyCourses” session and opens up a page that he can see the list of courses. 3. After he sees the list of classes, he picks one of the classes he wants to add students to. 4. He clicks on the AddStudent button near the student’s name. 5. After that, he clicks on the “Save Changes” button. 6. He is asked if he is sure of adding the students he picked with an alert screen. 7. If he says “Yes”, he gets the “Changes Saved” feedback. |

|  |
| --- |
| *Scenario Name:* DisplayStatistics |
| *Participant actor instanecs:* Atilla: Instructor |
| *Flow of events:*  1. Atilla opens the application and logs in as instructor.  2. He clicks on “MyCourses” session and opens up a page that he can display the list of courses.  3. Then Atilla clicks on the “Details” button that seen next to the course’s name which Atilla wants to see the list of quizzes.  4.a If Atilla wants to see all quizzes average, it will seen top of the window in related course.  4.b. If Atilla wants to see the average of a quiz, it will seen next to each quizzes (after all students are graded.).  4.c. If Atilla wants to see a students grade, Atilla must click the “details” button next to related quiz.  4.d. If Atilla wants to see any of the students exam paper to see students mistakes. |

|  |
| --- |
| **Scenario name:** Login Student |
| **Participant actor instances: Student:Ali** |
| **Flow of events:**   1. Ali joins QuizSystem/student on application. 2. Ali sees login pages and she enter UserName into UserName text field also enter password into Password text field on this pages.   a) if Ali leaves one or more fields empty, system displays a warning message, like "This area can not be empty.".   1. If UserName,password and role match to information which are in Quiz Maker database Ali can login to the Quiz Maker System.   a)If the informations checked from database are not true, system displays a warning message, like "Wrong username or password, please retry.". |

|  |
| --- |
| **Scenario name: Take quiz** |
| **Participant actor instances: Student:Ali** |
| **Flow of events:**   1. Ali logs into the QM. 2. He sees puclic quizzes and lecture quizzes button on the screen. 3. If he wants to take a public quiz, he clicks public quizzes button. Then, he sees all public quizzes. He clicks join button what he want to take, and quiz opens on the screen. He answers questions and than he clicks save button.   -He sees a tools message, Are you sure?.   * 1. If Ali wants to click the yes button. The message shown on screen that “it is saved successfully”.   2. If Ali does not want to click the no button, and he continues to quiz.  1. If he wants to take a lecture quiz, he clicks lecture quizzes button. He clicks join button which lecture quiz to take, and quiz opens on the screen. He answers question adn than he clicks save button.   -He sees a tools message, Are you sure?.   * 1. If Ali wants to click the yes button. The message shown on screen that “it is saved successfully”.   2. If Ali does not want to click the no button, and he continues to quiz.   3. He logs of. |
|  |

|  |
| --- |
| **Scenario name:** ListQuizzes |
| **Participant actor instances: Student:Ali** |
| **Flow of events:**   1. Ali logs into the QM. 2. Ali clicks the "List Quizzes" button on students main page. 3. He sees all quizzes and the results. 4. Ali clicks "Back" button. 5. The system directs students to his/her main page. 6. Ali logs of. |

|  |
| --- |
| **Scenario name:** Edit Profile |
| **Participant actor instances: Student: Ali** |
| **Flow of events:**   1. Ali logs into the QM. 2. Ali selects “Edit profile” button. 3. Then Ali click “edit and save” button. 4. Quiz Maker finds what Ali choose on databased change its all informations on system. 5. The message shown on screen that “It is edited successfully”. 6. Ali logs of. |

|  |
| --- |
| **Scenario name:** Create Quiz |
| **Participant actor instances:** Student: Ali |
| **Flow of events:**   1. Ali logs on in the QM. 2. This flow starts with Ali’s click to the create quiz button. 3. He clicks quiz type button. Then, Ali selects which one he wants. 4. System shows to Ali fields to fill with quiz info. 5. He fills fields with the quiz information which is date, time, etc. Then, he clicks save button. 6. Ali adds questions button. After he create his questions, clicks save button. 7. Ali logs of. |

|  |
| --- |
| **Use case name:** Logout |
| **Participant actor instances: Student:Ali** |
| **Flow of events:**   1. Ali logs on. 2. He clicks “Logout” button. 3. System redirects to the login page. |

|  |
| --- |
| **Scenario name:** Deleting Instructor |
| **Participant actor instances:** Admin:Atilla User**:**Begüm |
| **Flow of events:**   1. Atilla logs on the system. 2. He clicks List of Instructors button. 3. He sees all Instructors who registered in the system. 4. Atilla sees instructors id,name,surname and delete,edit buttons in the row, then selects Begüm. 5. Then Atilla clicks “delete and save” button. 6. Quiz Maker finds the Begüm on database and delete her all informations on system. 7. The message shown on screen that “The Instructor is deleted successfully”. 8. Atilla logs of. |

|  |
| --- |
| **Scenario name:** Deleting Student |
| **Participant actor instances:** Admin:Atilla User**:**Selin |
| **Flow of events:**   1. Atilla logs on the system. 2. He clicks List of Student button. 3. He sees all student who registered in the system. 4. Atilla sees students id,name,surname and delete,edit buttons in the row, then he selects Selin. 5. Then Atilla clicks “delete and save” button. 6. Quiz Maker finds the Begüm on database and delete her all informations on system. 7. The message shown on screen that “The Instructor is deleted successfully”. 8. Atilla logs of. |

|  |
| --- |
| **Scenario name:** Deleting course |
| **Participant actor instances:** Admin:Atilla Course: SE301 |
| **Flow of events:**   1. Atilla logs on the system. 2. He clicks List of Course button. 3. He sees all courses which registered in the system. 4. Atilla sees course name and delete, edit buttons in the row, then he selects SE301. 5. Then Atilla clicks “delete and save” button. 6. Quiz Maker finds the Se301 on database and delete its all informations on system. 7. The message shown on screen that “The course is deleted successfully”. 8. Atilla logs of. |

|  |
| --- |
| **Scenario name:** Edit course |
| **Participant actor instances:** Admin:Atilla Course: SE301 |
| **Flow of events:**   1. Atilla logs on the system. 2. Atilla clicks list of course button, then he sees all courses which registered in the system. 3. He sees course name and edit, delete buttons in the row, then select Se301. 4. Then he clicks “edit and save” button. 5. Quiz Maker finds the Se301 on databased change its all informations on system. 6. The message shown on screen that “The course is edited successfully”. 7. Atilla logs of. |

|  |
| --- |
| **Scenario name:** Deleting Quiz |
| **Participant actor instances:** Admin:Atilla Quiz: SE301 |
| **Flow of events:**   1. Atilla logs on the system. 2. Atilla clicks List of quiz button, then he sees all quiz which registered in the system. 3. He sees quiz name, which course belongs to the quiz and edit, delete buttons in the row, then select Se301. 4. Then Atilla clicks “delete and save” button. 5. Quiz Maker finds the Se301 on database and delete its all informations on system. 6. The message shown on screen that “The quiz is deleted successfully”. 7. Atilla logs of. |

|  |
| --- |
| **Scenario name:** Approve Request to Instructor |
| **Participant actor instances:** Admin:Atilla Instructor:Selin |
| **Flow of events:**   1. Selin send request to be instructor to Atilla. 2. Atilla logs on the system. 3. He clicks the "Request of Instructors" button. 4. He sees List of Requests of Instructors. 5. Atilla choose Selins and click Approve button. 6. He clicks the "Save" button. 7. He see a "Saved successfully!" message on a dialogue box. 8. Atilla logs of. |

|  |
| --- |
| **Scenario name:** Reject Request to Instructor |
| **Participant actor instances:** Admin:Atilla Instructor:Mert |
| **Flow of events:**   1. Mert send request to be instructor to Atilla. 2. Atilla clicks the "Request of Instructors" button. 3. He sees List of Requests of Instructors. 4. He choose Mert and click Reject button. 5. Atilla clicks the "Save" button. 6. He see a "Saved successfully!" message on a dialogue box. 7. Atilla logs of. |

|  |
| --- |
| **Scenario name:** ListInstructors |
| **Participant actor instances:** Admin:Atilla |
| **Flow of events:**   1. Atilla logs into the system. 2. Atilla clicks the "List Instructors" button on administrator main page. 3. He sees all Instructors who registered in the system. 4. Atilla clicks "Back" button. 5. The system directs administrator to his/her main page. 6. Atilla logs of. |

|  |
| --- |
| **Use case name:** Logout |
| **Participant actor instances:** Admin:Atilla |
| **Flow of events:**   1. Atilla logs into the system. 2. Atilla clicks "Logout" button. 3. System redirects administrator to the login page. |

|  |
| --- |
| **Scenario name: Add course** |
| **Participant actor instances:** Admin:Atilla |
| **Flow of events:**  1. Atilla opens the application and logs in as admin. She sees AddCourse button at the first page she arrived by logging in.  2. Atilla clicks on the AddCourse button.  3. She selects the course’s name from the given course list which has all the possible classes that can be added to the system and clicks on the “Add” button.  4. There comes a notification saying “Course is added successfully.” and sees that the course is added to the system. |

### Use case model

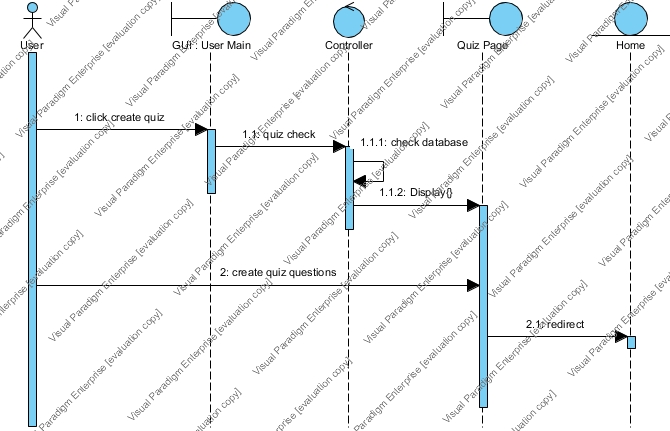
### 

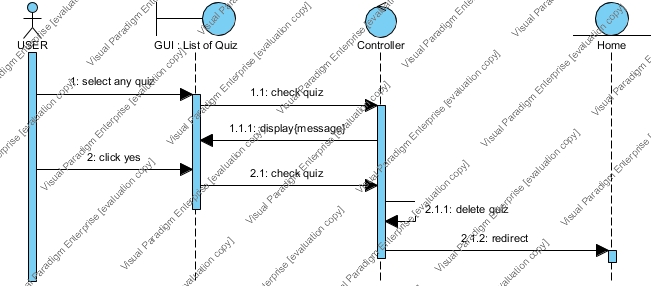
### Object model

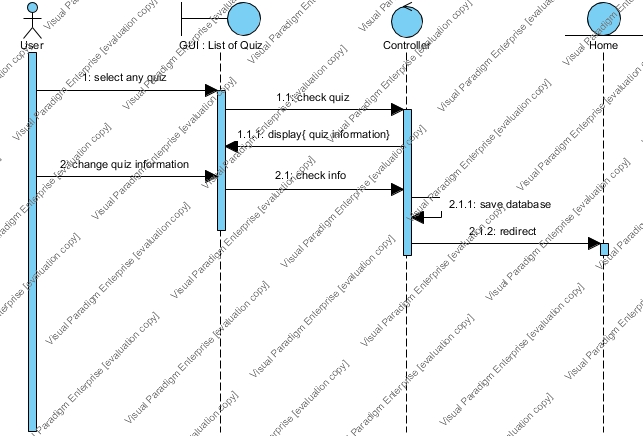
The analysis object model, depicted with UML class diagrams, includes classes, attributes, and operations. The analysis object model is a visual dictionary of the main concepts visible to the user.

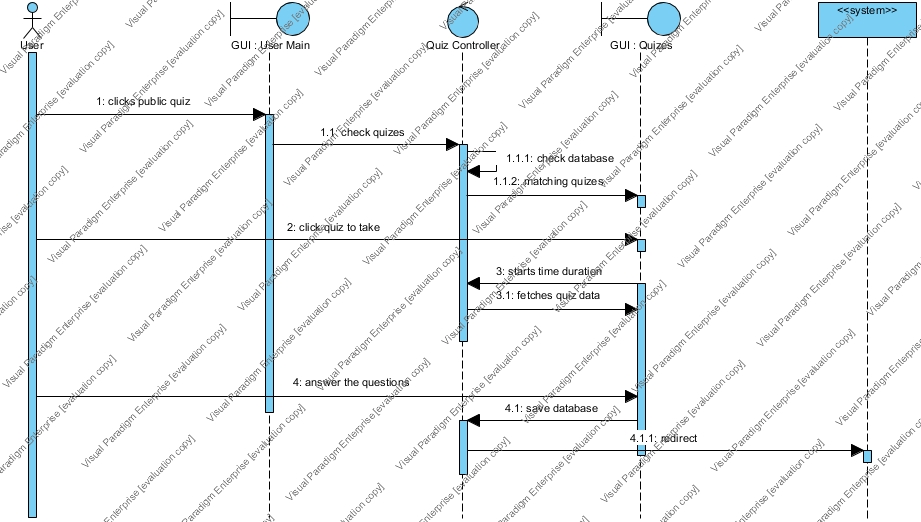
### Dynamic model

### 

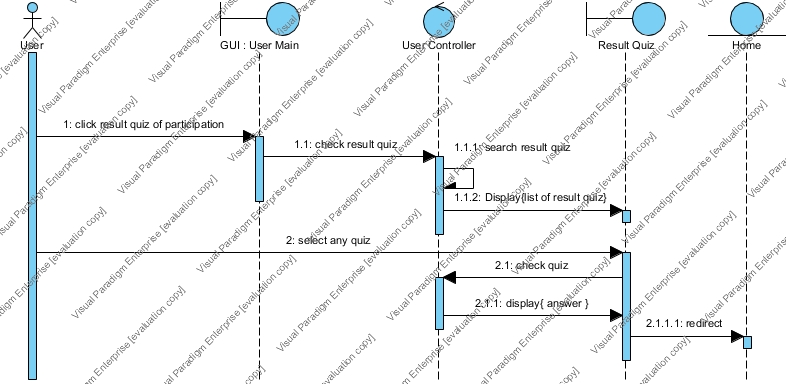


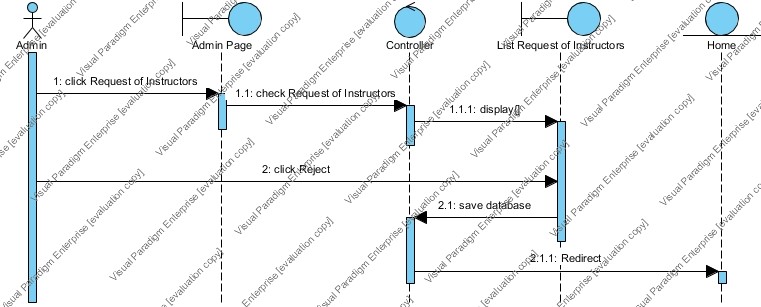
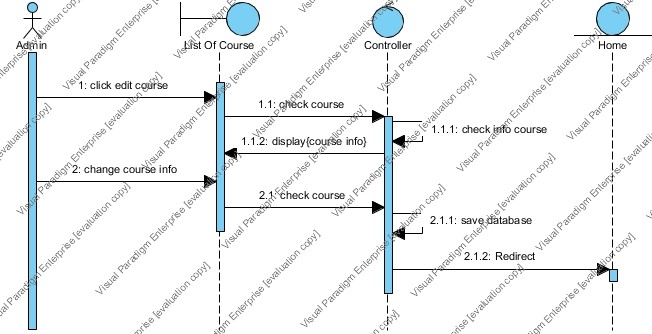
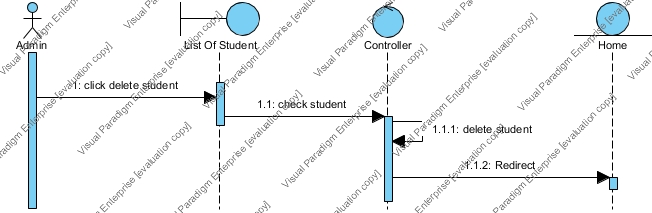
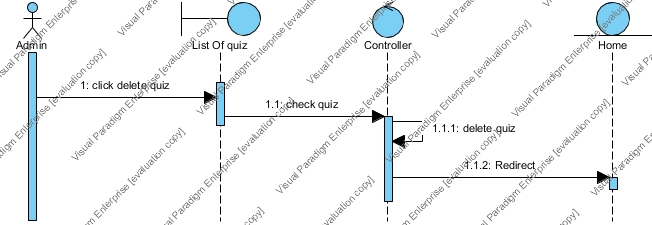
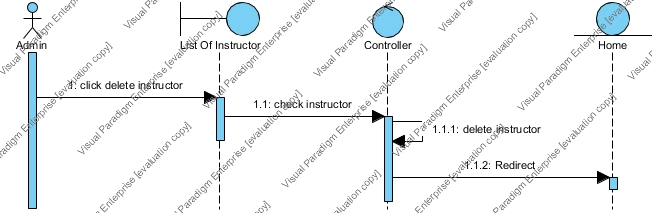
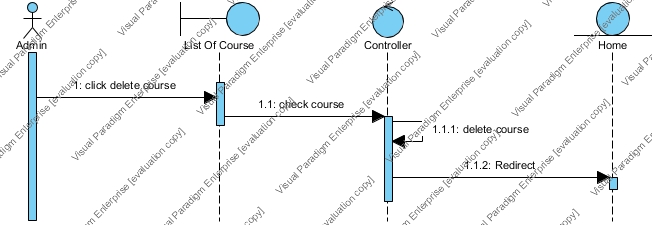
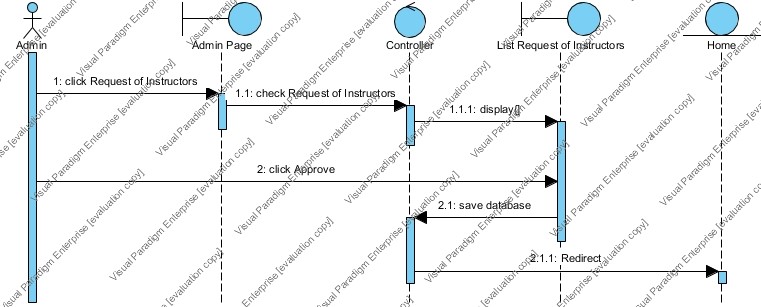
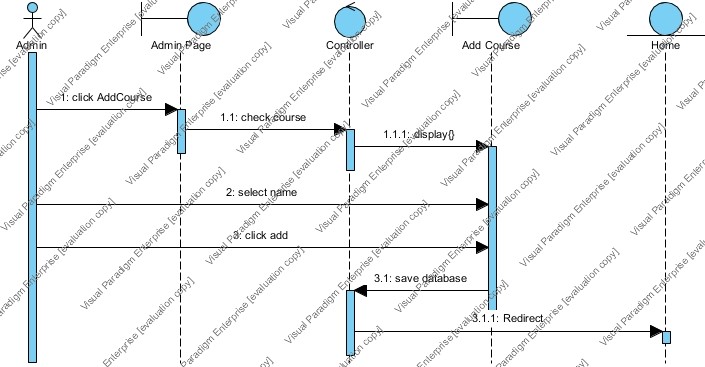


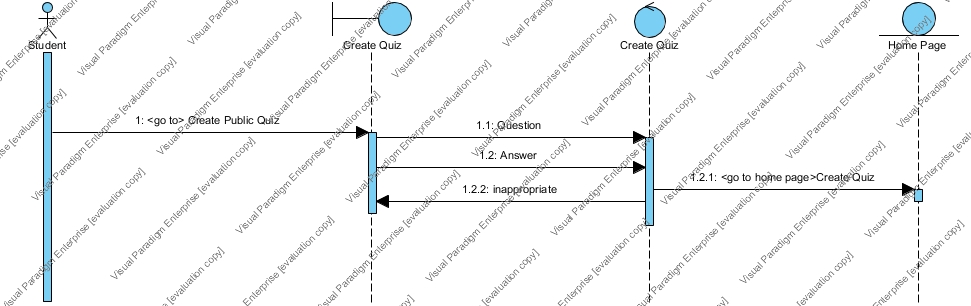


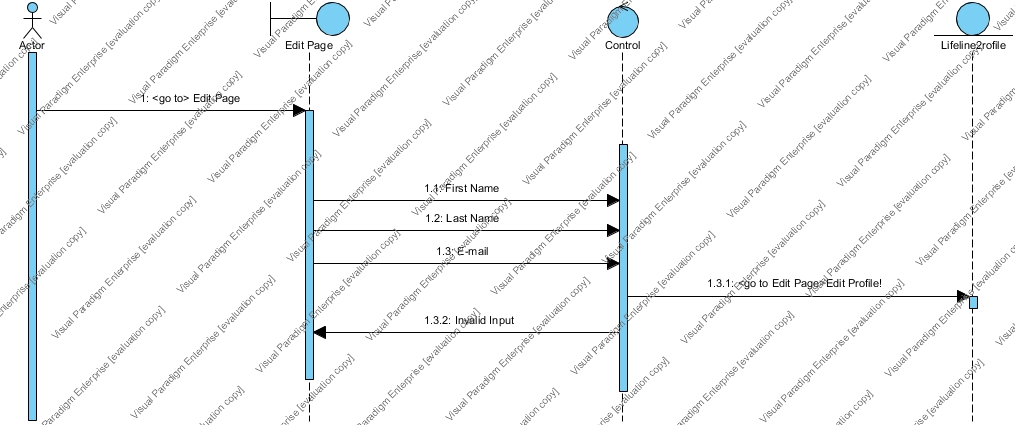


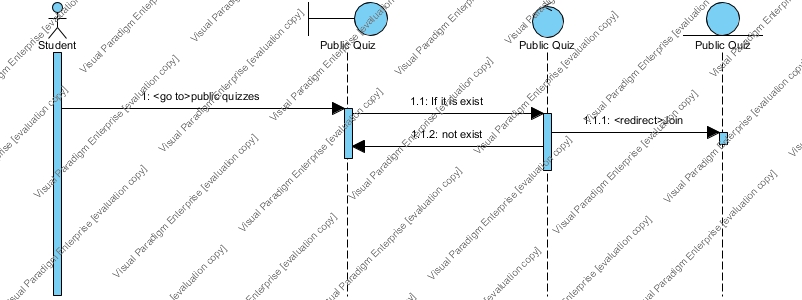
### 

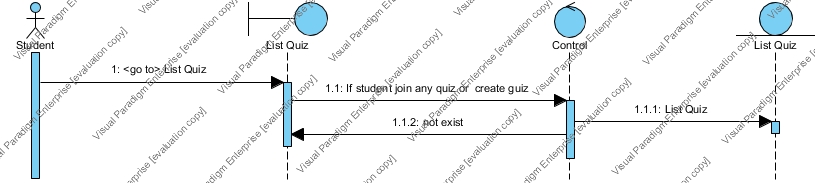


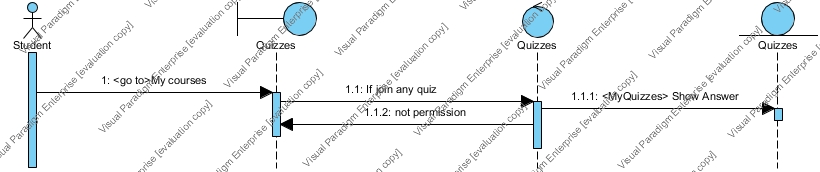






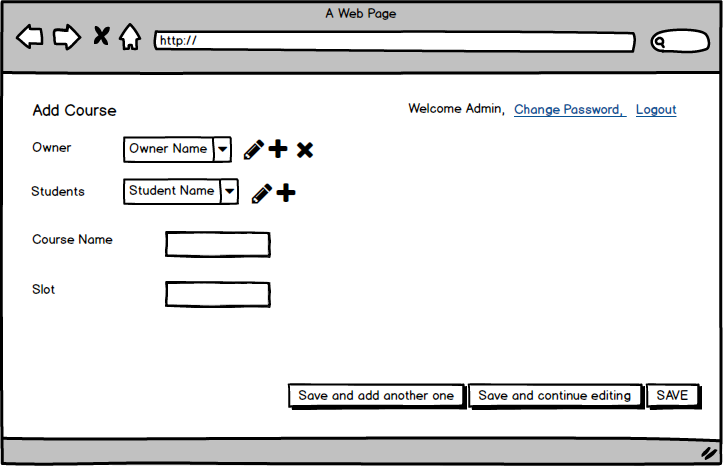
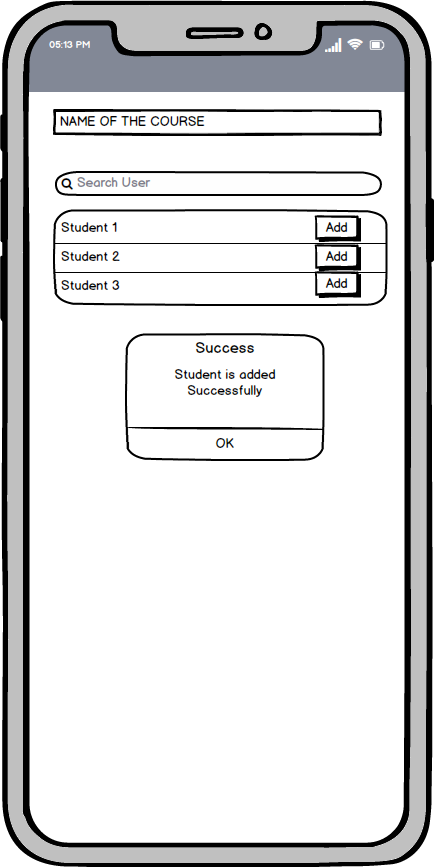
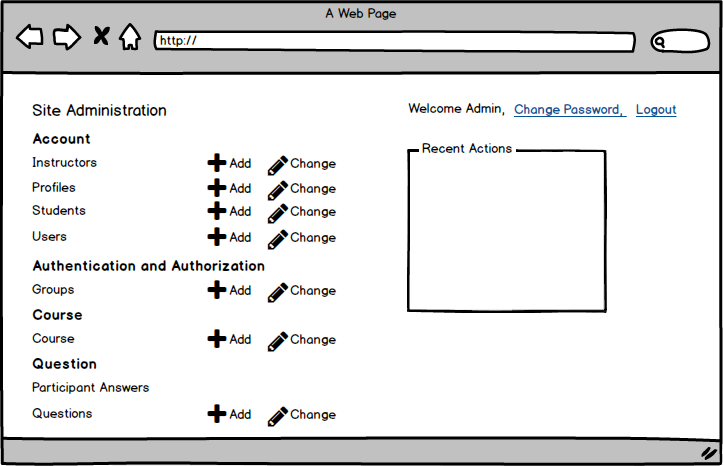
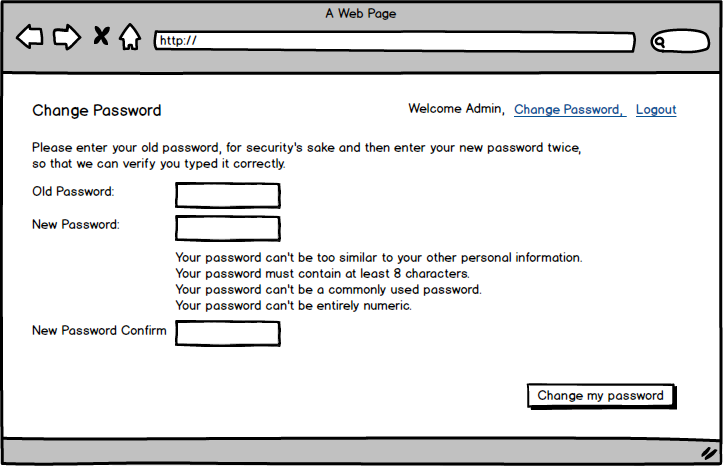
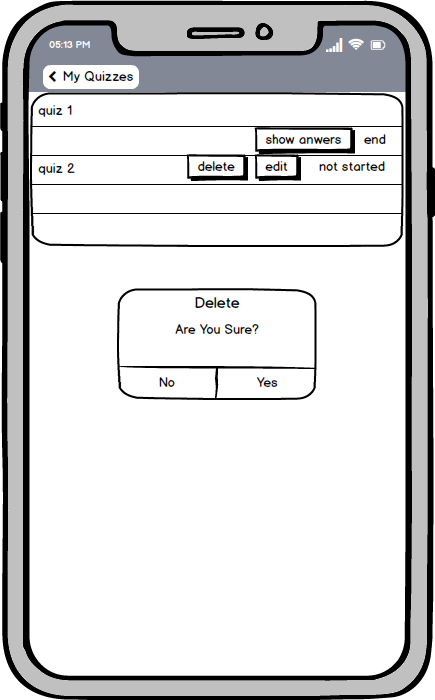
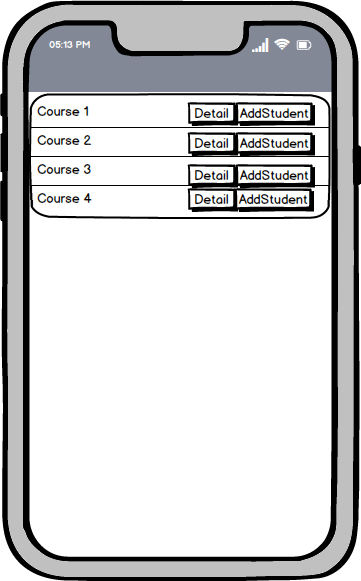
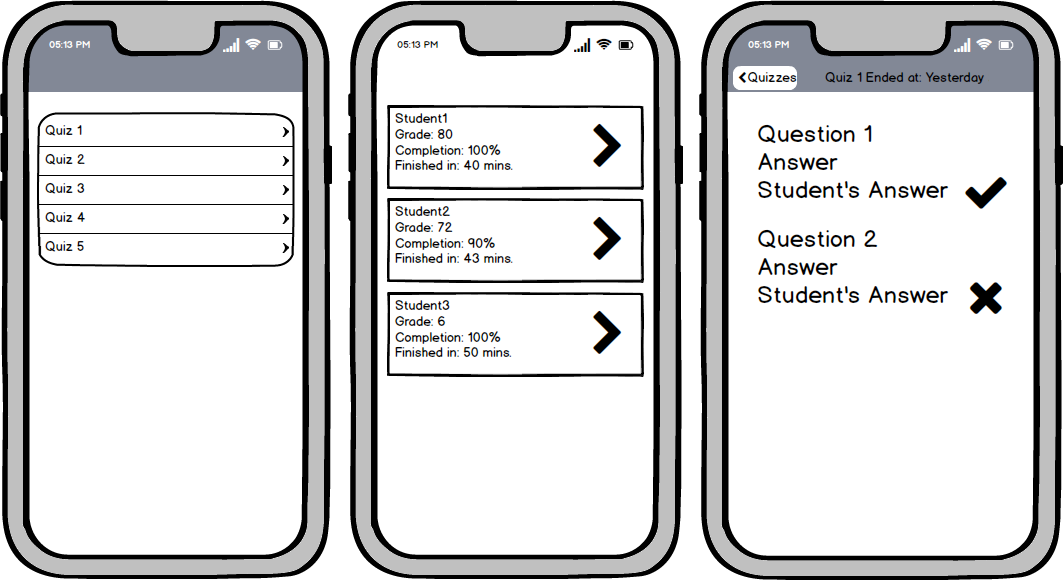
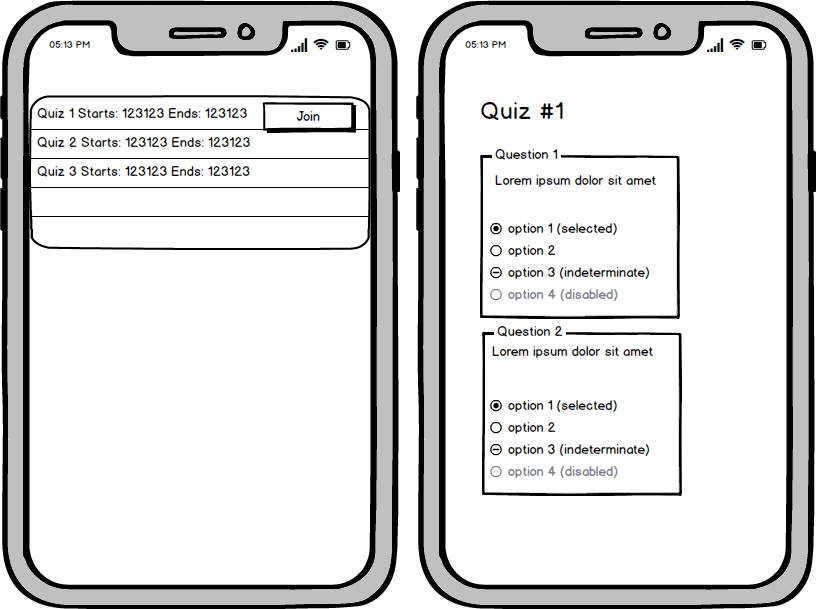
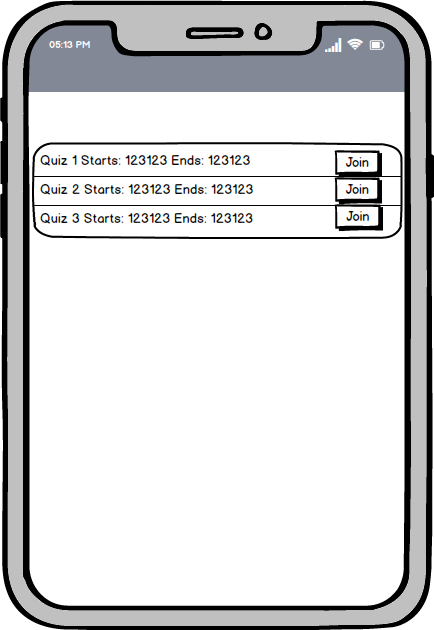
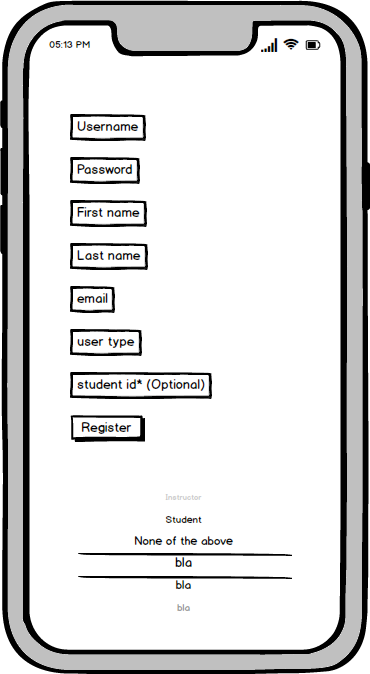
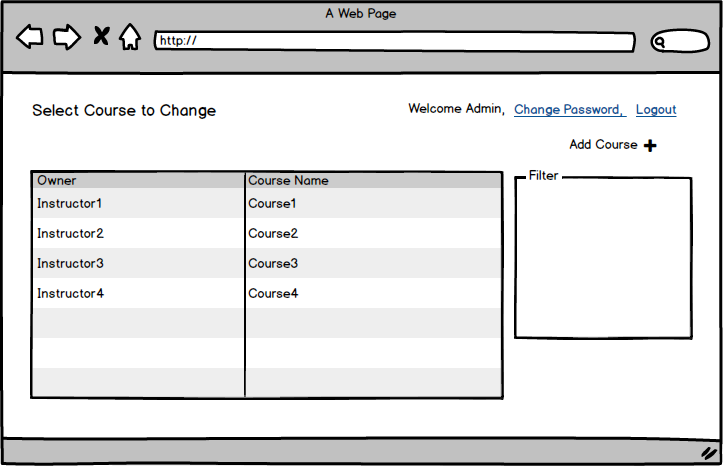
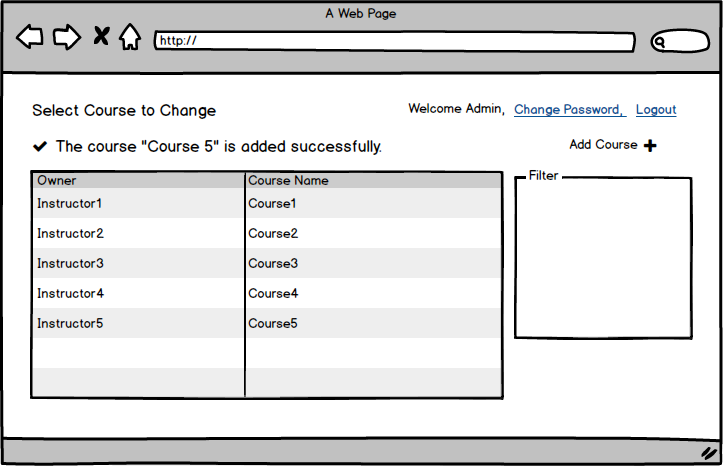
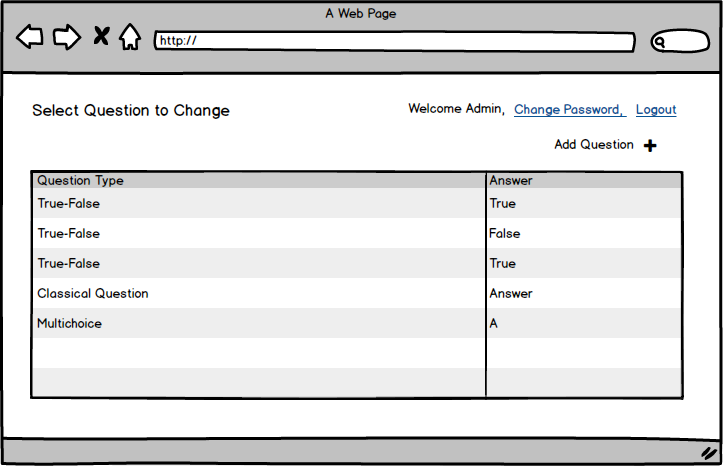
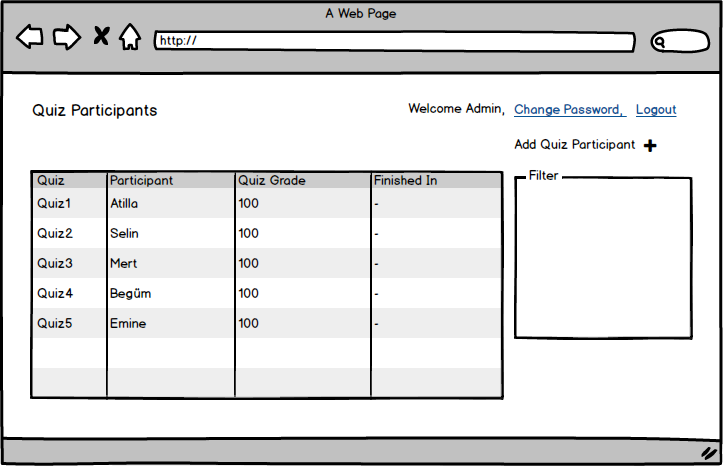
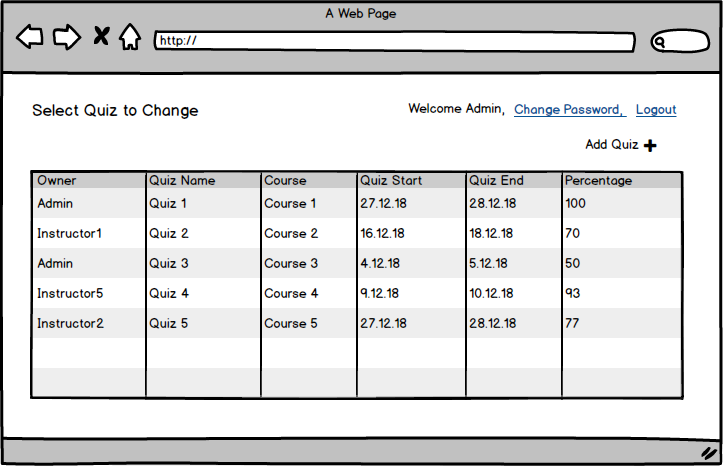
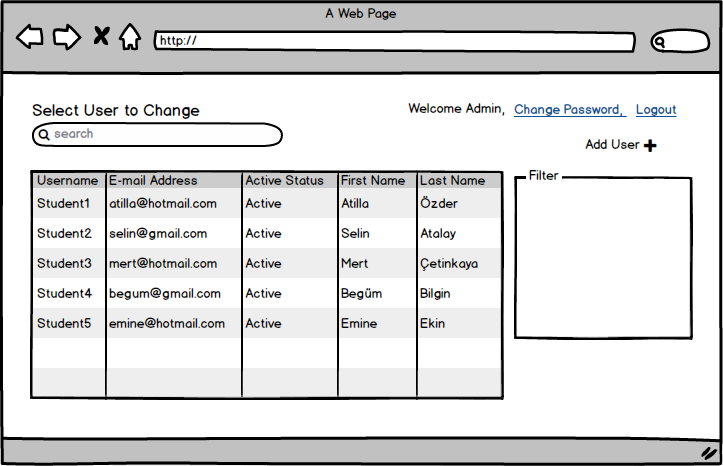
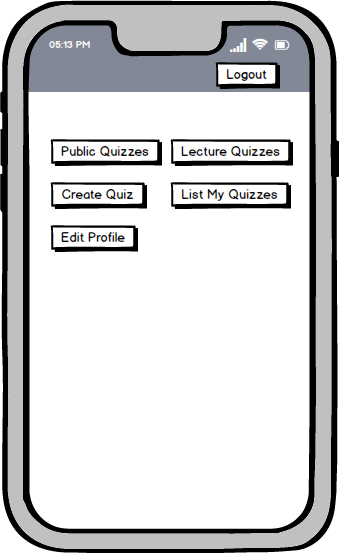
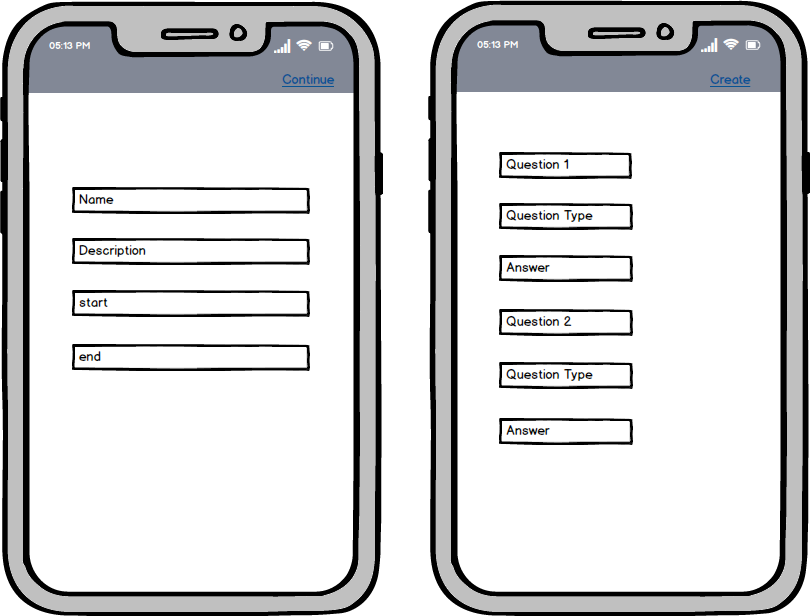
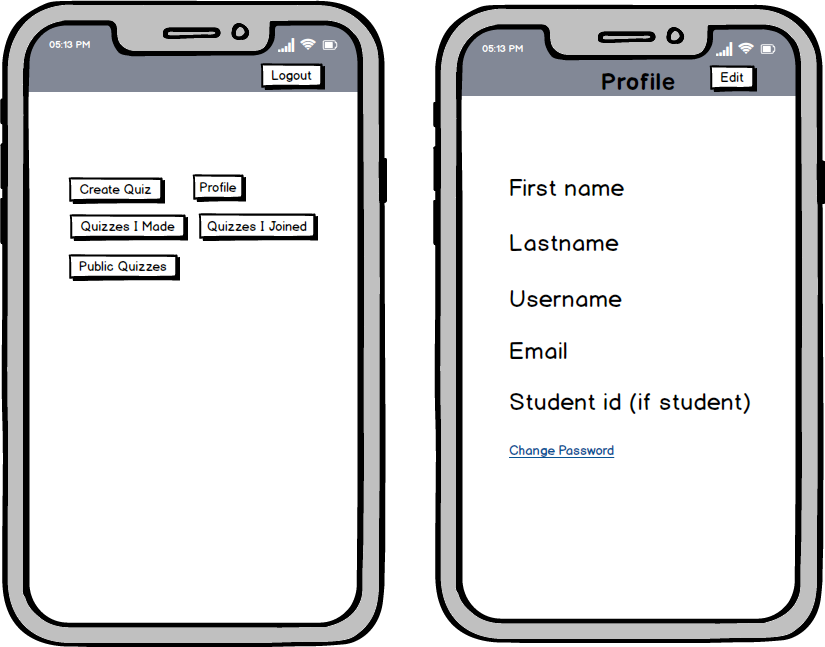






### 

### User interface—navigational paths and screen mock-ups

****

## Project Schedule

Prepare Gannt Chart, and add it to this section.

# Glossary

To establish a clear terminology, developers **identify the participating objects** for each use case. Developers should **identify, name, and describe them** unambiguously and collate them into a glossary.

# References

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.