

CSCE 606: Software Engineering

Project title: CSE PhD Qual Practice System

Team: WeCode

Scrum Master: Shruthi Sampathkumar

Product Owner: Abhishek Taur

Team Members:

Abhishek Taur

Pranav Kulkarni

Arif Arman

S M Farabi Mahmud

Raj Vardhan

Shruthi Sampathkumar

Shibin Tazhe Veetil

Meeting with Customer:

We met with Dr. Walker (customer) on Thursday 27th February, 2020 at 2pm. We discussed the customer requirements for the Project.

Summary:

Our Customer is the Computer Science and Engineering Department of Texas A&M University. The Customer requires a software solution which is a web based user interface for PHD Qualification exam practice and testing. The platform can be used by students as well as Faculty. An existing solution is present which has basic features implemented. The existing solution provides features like user registration as an admin or regular user, take quiz or practice

questions, add questions of different categories and add new categories. There are also some unresolved bugs in the previous implementations.

The customer requires additional features to be implemented so that the practice session and quiz can be timed, Add tags to go to previous questions, ability for the admin to assign other users as admin and approve questions added by other users, ability to quit the session in between, deploy the site on a paid account, login using google or any other social media login, support for mobile and tablet, password recovery. Our solution will include a user friendly interface which shows the timer when you are starting the quiz/practice session and the user can set it to his desired limit. We will be providing options to tag questions which will be displayed to the user to go back. Additional options to assign admin will be provided. The design will include additional options to use google or facebook for login, password recovery. The design will be compatible on Mobile phones as well as tablets. These implementations are as per the customer requirements.

Links:

Github: <https://github.com/Shruthi-Sampathkumar/PhD-Qual-Practice.git>

Pivotal Tracker: <https://www.pivotaltracker.com/n/projects/2436597>

User Stories to be implemented :

1. Feature : Add Timer
As a PhD Qual User
I should be able to customize the time limit for my quiz/practice
So that I can get a simulation of the actual qual exam
2. Feature : Tag Questions
As a PhD Qual User
I should be able to tag questions and go back to them during test
So that I can answer questions in any order
3. Feature : Only an admin can add admins
As a PhD Qual Admin
Only I should be able to add new admins to the system
So that users do not have the privilege of becoming admins themselves
4. Feature : Deploy to Production
As a PhD Qual Admin
I should be able to use a paid Heroku application cloud platform

So that I have features like free SSL and automated certificate management.

5. Feature: Ability to Quit the session

As a PhD Qual User

I should be able to quit the session while it's running

So that I can restart the quiz or practice from the start

6. Feature : Multi user support

As a PhD Qual User

I should be able to use the system even when several other students are using it

So that my work is not impacted when the system is experiencing heavy traffic

7. Feature : Omni-login

As a PhD Qual User

I should be able to login to the system using my Google or Facebook account

So that I don't have to manage separate credentials for this service

8. Feature : Admin Password Recovery

As a PhD Qual Admin

I should be able to create a new password if I have forgotten my current password

So that I don't lose access to the system in case I forget my password

User interface:

Below you can find the initial ui mockup. These mockup is not finalized, especially the color schemes. Please consider it as an initial reference for the discussion.

1. UI with new login option.

Log in

Email

Password

Forgot password?

Log in

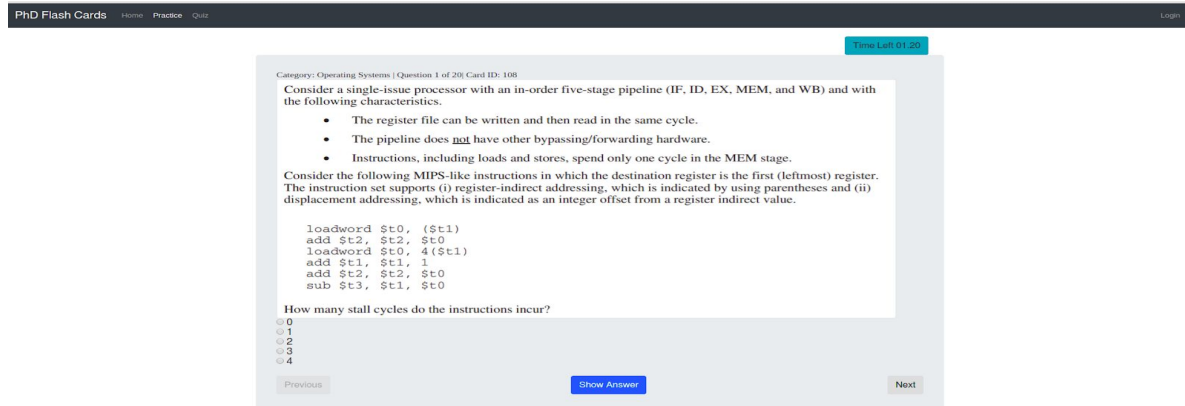
"Not a user?" Register



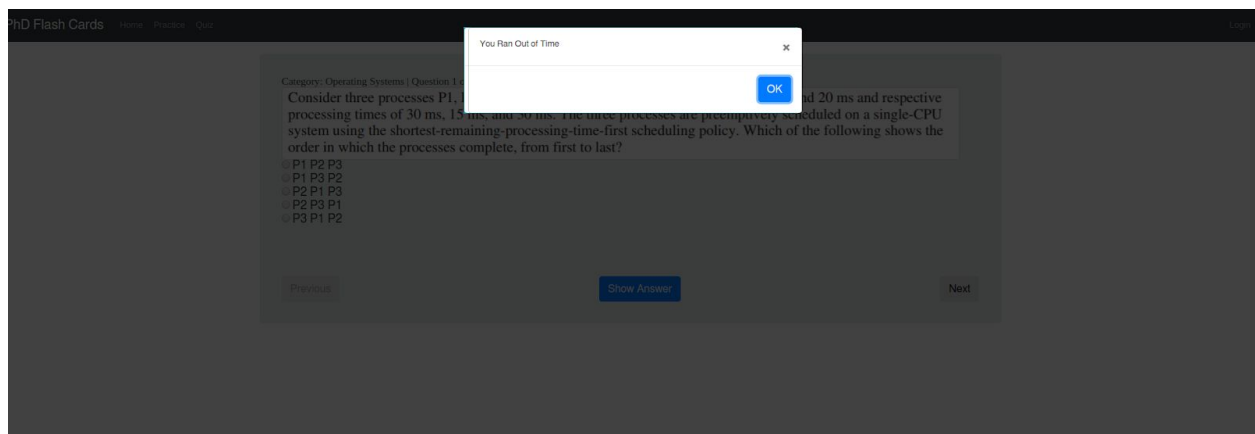
Continue with Google



Continue with Facebook



b. Timeout screen




3. Tag question to come back to later

A star icon to mark this question to review again. On click to the “marked questions” button it shows the list of marked questions. The user can hide the list by clicking the same button (works as a toggle button).

PhD Flash Cards

HomePracticeQuiz

Category: Algorithms | Question 3 of 20 | Card ID: 2



Consider the following pseudocode.

```
x := 1;
i := 1;
while (x ≤ 1000)
begin
  x := 2x;
  i := i + 1;
end;
```

What is the value of `i` at the end of the pseudocode?

☐ 4

☐ 5

☐ 6

☐ 7

☐ 8

Previous

Next


Marked Questions

Submit

PhD Flash Cards

HomePracticeQuiz

Category: Algorithms | Question 3 of 20 | Card ID: 2



Consider the following pseudocode.

```
x := 1;
i := 1;
while (x ≤ 1000)
begin
  x := 2x;
  i := i + 1;
end;
```

What is the value of `i` at the end of the pseudocode?

☐ 4

☐ 5

☐ 6

☐ 7

☐ 8

Previous

Next

Marked Questions:

Q1

Q2

Q3

Q17

Q18

Hide Marked Questions

Submit

Figure: Mockup for tagging a question

Strategy to Approach Legacy Code:

We received the legacy code in the form of GitHub repository. We deployed the code in heroku test server to see in details how the codes were implemented with relevance to the different existing features. Existing codebase does not cover some of the requested features of the client. So we have decided to break down the feature set to be in two groups, where the first ones are

completely new features and the second group consists of existing features which need improvements. The new features which needed to be added are -

1. Addition of a mechanism for setting up timer before taking a quiz
2. Creation of a new interface between user and the system to mark questions during the quiz
3. Integration of 'Omni login authorization' with the system

We also need to improve some of the existing features. These are -

1. Modification of the bookmarking mechanism in look and feel
2. Incorporating bookmarking mechanism in the practice mode
3. Deploying the system to highly scalable server
4. Stress Testing the deployment with large number of concurrent users
5. Incorporating admin password recovery system with the current password recovery system