

## **Contents**

- **Web project**
- **Desktop project**
- **Deadline : 11-2-2016**

# Web Project

Each team is required to deliver part of the final web application functionality based on team current experience.

## First Iteration (In agile terms called sprint)

- **Level A Teams:**

You're required to build a one page application containing a static image and anyone with the link is able to make comments and like that single image, and it should be an image from any slide of your choice, this page should show number of likes for that image, and should also show all the comments below the image.

- **Level B Teams:**

You're required to build a two pages application containing

1. First page: containing a login page with a textbox for username and another one for password with only available two usernames
2. Second page: After signing in you'll be directed to another page containing a static single image and only people who logged in using the username and password (not any one with the 2nd page link) will be able to view this page contents to make comments and like that single image, and it should be an image from any slide of your choice, and comments should be sorted by date in descending order, number of likes should be shown below the image.

- **Level C Teams:**

You're required to build a two or three pages application containing

1. First page: Containing a login page with a textbox for username and another one for password and a sign up details, any one can sign up for a new account, then can sign in and be able to redirected to the second page.
2. Second page: After signing in you'll be directed to another page containing a static single image and only people who logged in using the username and password (not any one with the 2nd page link only registered users) will be able to view this page contents to make comments and like that single image, and it should be an image from any slide of your choice, and comments should be sorted by date in descending order, number of likes should be shown below the image.

- **Level D Teams:**

Same as C Teams but instead of only one image, after login you'll be directed to a page containing thumbnails of all slides within a single document, with comments about the document itself, when you click on any image a third page that's the same as the second page on C teams, but you'll be able to navigate forward and backward in slides, or select a single page number

- **Level E Teams:**

Same as D teams but with a nice look and feel user experience and with a signed in person that is able to upload slides, so the structure will initially be login page -> slides ->

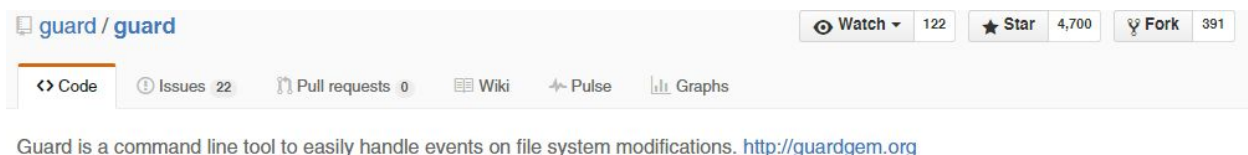
single document -> single image , @ all pages after login you can upload a new document, and then any one will be able to find it in the slides page after they log in.

Deliverables:

- Source code should be maintained on an online repository on github and all team members should have a proven collaboration on that online repo.
- Starting your github repository wiki (a wiki sample <https://github.com/guard/guard/wiki>) as your documentation for your product, and you're required to document the following at the first iteration:
  - a. Abstract section containing a general description of your product. (4 lines max)
  - b. Use case diagram for your functionality with use case specifications as specified in the lecture "A\_Software Engineering\_Requirement.pptx".
  - c. Class diagram (association relationship between your models) that you've build according to your needs (involve the framework classes that you're using within the class diagram you've made).
- A pdf containing your work distribution and who did what.

**Deadline : 22-12-2015**

Note:



What's wiki : each github repo is equipped with multiple tools:

**Issues:** where you can add your tasks -> new features, bugs, enhancements.

**Code:** your actual repo is there

**Wiki:** where you can describe your project

<https://guides.github.com/features/wikis/>

You write wiki pages using markdown syntax -> refer to the following link in case you need to add image/link ..etc

it's easy like writing normal text with very small variations.

<https://help.github.com/articles/markdown-basics/>

<https://guides.github.com/features/mastering-markdown/>

# Desktop Project 1st Milestone

Each team is required to deliver part of the final Desktop application functionality based on team previous experience + documentation.

## First Iteration (In agile terms called sprint)

- **Level A Teams:**

You're required to build a windows forms application that have a toolbox partition containing different logic gates (and, or, xor), with a click on any of these gates, the gate type is drawn at the middle of your simulator, and you can move individual gates in the simulator wherever you want, as long as there's no conflict between any two gates, the simulator window should be resizable so you can add multiple gates as you need.

- **Level B Teams:**

Same as A teams + the ability to connect different gates + the ability to drag and drop gates from the toolbox instead of inserting gates directly to the simulator.

- **Level C Teams:**

Same as B teams, in addition to adding input controllers (buttons) and output indicators (leds) and being able to simulate your design

- **Level D Teams:**

Same as C teams, in addition to ability to save designs to a file and load saved designs

## Deliverables:

- Source code should be maintained on an online repository on github and all team members should have a proven collaboration on that online repo.
- Starting your github repository wiki (a wiki sample <https://github.com/guard/guard/wiki>) as your documentation for your product, and you're required to document the following at the first iteration:
  - a. Abstract section containing a general description of your product. (4 lines max)
  - b. Use case diagram for your functionality with use case specifications as specified in the lecture "A\_Software Engineering\_Requirement.pptx".
  - c. Class diagram that you've build according to your needs (you don't have to explicitly model everything in the framework you're working with but model all your written classes).
- A pdf containing your work distribution and who did what.

## Deadline : 22-12-2015

Note:

What's wiki : each github repo is equipped with multiple tools:

**Issues:** where you can add your tasks -> new features, bugs, enhancements.

**Code:** your actual repo is there

**Wiki:** where you can describe your project

guard / guard

Watch 122 Star 4,700 Fork 391

Code Issues 22 Pull requests 0 Wiki Pulse Graphs

Guard is a command line tool to easily handle events on file system modifications. <http://guardgem.org>

<https://guides.github.com/features/wikis/>

You write wiki pages using markdown syntax -> refer to the following link in case you need to add image/link ..etc

it's easy like writing normal text with very small variations.

<https://help.github.com/articles/markdown-basics/>

<https://guides.github.com/features/mastering-markdown/>