|  |  |  |
| --- | --- | --- |
|  |  | Automated Code Grading  Project report  Zhen Li  Yue Gan  Jun Gao  Haifeng Jin  Linqian Li  Zhenhao Qian |

Dec 5, 2015

# ABSTRACT

Our customer is Bryan High School. The teacher needs an application that can grade their programming homework automatically. Besides, the application also provides interfaces for teachers and students. Firstly, for the teachers and students, they will have their own user information for login so as to collect their data information. For the teacher, we design functions for him to assign new homework including its information, upload expected answers for grading and check the homework information of each student. For students, we design functions for them to check new homework information, upload their codes and compare with the right answers uploaded by the teacher; also they will get the grades automatically after each comparison and they can edit their codes until they satisfy the grade.

# Introduction and motivation

When teaching programming courses, teachers will assign programming homework for students. However, as hundreds of students submit their code to the teachers, it is quite hard to grade them. Therefore, a platform is essential for the teachers so that the platform can provide a user interface for both students and teachers, grade the code automatically and even help the teachers to manage their courses and students’ homework records. The application is divided into two major parts: one is a user interface based on web that provides services like managing courses, checking students homework and so on; the other is the a program running in background to automatically grade the code submitted by the students and return the results or running error information to the user interface. The idea with the motivation came from Bryan High School teachers who guide the students to study basic Java and Python programming.

# Stakeholders

Customers:

Jason Cordes, Bryan High School

Developers:

1. Yue Gan – Product Owner
2. Zhen Li – Scrum Master
3. Haifeng Jin
4. Zhenhao Qian
5. Linqian Li
6. Jun Gao

End Users:

Bryan High School Students

# Important links

Video link: <https://www.youtube.com/watch?v=nB5BvSGTt3I>

GitHub repo link: <https://github.com/jhfjhfj1/automated_code_grading> Pivotal Tracker link <https://www.pivotaltracker.com/n/projects/1440602>

# User stories

## Iteration 0:

### Feature 1: Student Login Interface  5 points

As a student

So that I can record all my homework history. I want to login to my personal account.

### Feature 2: Student Code Testing Interface  5 points

As a student

So that I can test whether my program runs properly  I want to upload my source code and check the running result.

### Feature 3: View students' submissions  3 points

As a teacher

So that I can keep track of each submit of the students who takes my class  I want to obtain the view of the code, the score and the number # of each submission of these students.

### Feature 4: View homework information  3 points

As a teacher

So that I can obtain the information of the homework I assigned  I want to obtain the description, the release date, the deadline, the constraint inputs, the expected outputs and the students who submitted of each homework

### Feature 5: Edit homework  4 points

As a teacher

So that I can edit the homework I assigned  I want to obtain the original information including the description, the deadline, the inputs and expected outputs for test and I can change all these and submit the new information.

### Feature 6: Assign homework  5 points

As a teacher

So that I can assign homework to my students  I want to submit the description, the deadline, the inputs and expected outputs for test.

### Feature 7: View homework information  3 points

As a teacher

So that I can obtain the information of the homework I need to complete or I have completed I want to obtain the description, the release date, the deadline and the grade if I have accomplished of each homework

### Feature 8: Submit homework  5 points

As a student

So that I can submit one of my homework I want to obtain a form including an editor or a file upload button and a submit button to submit my code for this homework

## Iteration 1:

### Feature 1: Student Login Interface completed

As a student

So that I can record all my homework history. I want to login to my personal account.

### Feature 2: Student Register Interface  5 points

As a student

So that I can use the system to finish my homework. I want to sign up an account to the system.

## Iteration 2:

### Feature 1: View homework information completed

As a teacher

So that I can obtain the information of the homework I assigned I want to obtain the description, the release date, the deadline, the constraint inputs, the expected outputs and the students who submitted of each homework

### Feature 2: View students' submissions completed

As a teacher

So that I can keep track of each submit of the students who takes my class I want to obtain the view of the code, the score and the number # of each submission of these students.

### Feature 3: Teacher decide which class is the student in After the registration of the students, the teacher has to ways to add a student to one class. 2 points completed

First is directly add the student to a class. Second is to give the student a class code. The student then put it in the system to get into the class.

### Feature 4: One class can have more than one teacher 2 points completed

As an instructor

So that different teachers can share resources and cooperate I want one class can have more than one teacher Feature 5: Student dashboard As a student so that I can check my course and homework arranged by courses I want to have a dashboard show all my courses and unfinished courses.

## Iteration 3:

### Feature 1: One class may let the student use different languages 2 points completed

The class may use multiple languages for its homework. Also, one homework can have multiple languages.

### Feature 2: Each line in the output file is one test case 2 points completed

There is only one input & output file for each homework. Each line in the output is one case. Each case worth the same points.

### Feature 4: Keep every submission for record 1 points completed

No later submissions should cover the earlier ones.

## Iteration 4:

### Feature 1: Display detailed message for compile errors 3 points completed

As a student

So that I can get the revise my code I want to obtain the detail information of compile errors for my homework submission.

### Feature 2: Keep every submission for record 3 points completed

As a teacher

So that I can keep track of progress for any student on every homework I want that no later submissions should cover the earlier ones.

### Feature 3: Student Code Testing Interface 3 points completed

As a student

So that I can test whether my program runs properly I want to upload my source code and check the running result.

### Feature 4: Edit homework completed completed

As a teacher

So that I can edit the homework I assigned I want to obtain the original information including the description, the deadline, the inputs and expected outputs for test and I can change all these and submit the new information.

### Feature 5: Use UserID to login 2 points completed

As a student

So that I can log in even if I do not have an Email. I want to use either Email and UserID to login. Feature 6: Assign homework As a teacher So that I can assign homework to my students I want to submit the description, the deadline, the inputs and expected outputs for test.

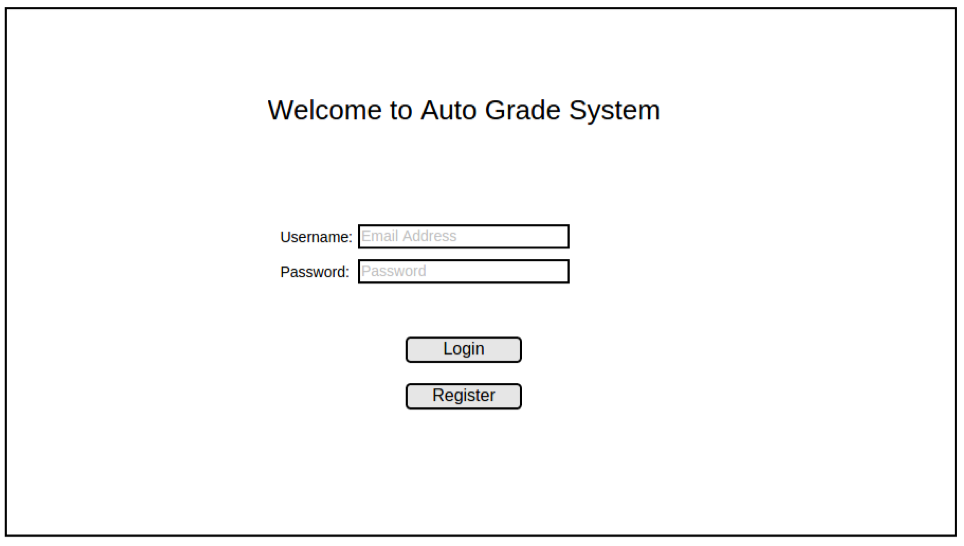
### Feature 6: Announcement for the students 4 points completed

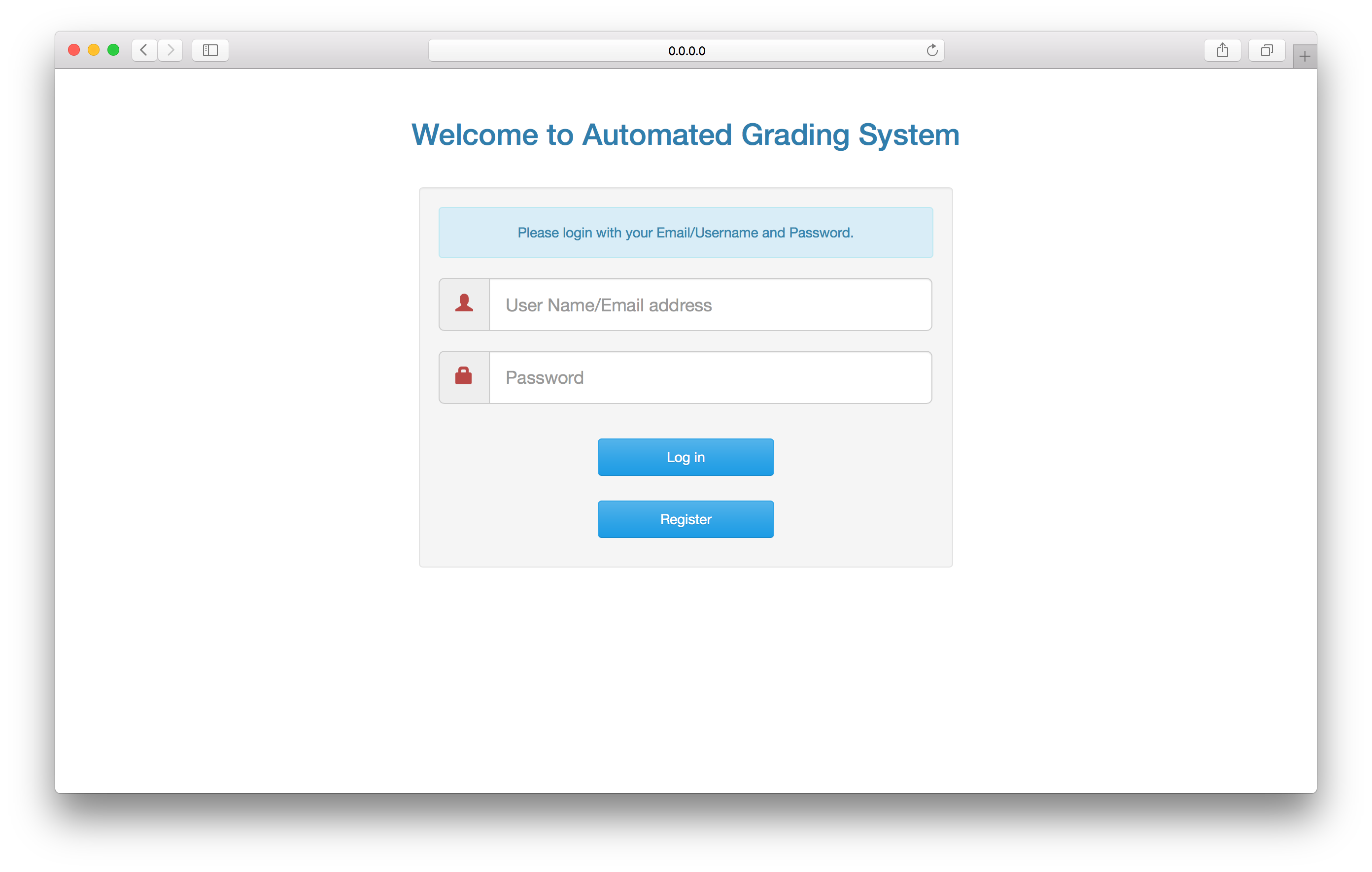
As a student

So that I can track the latest updates of the class I want to have an announcement reminding me of the latest news.

# Lofi-mockup and final screenshot

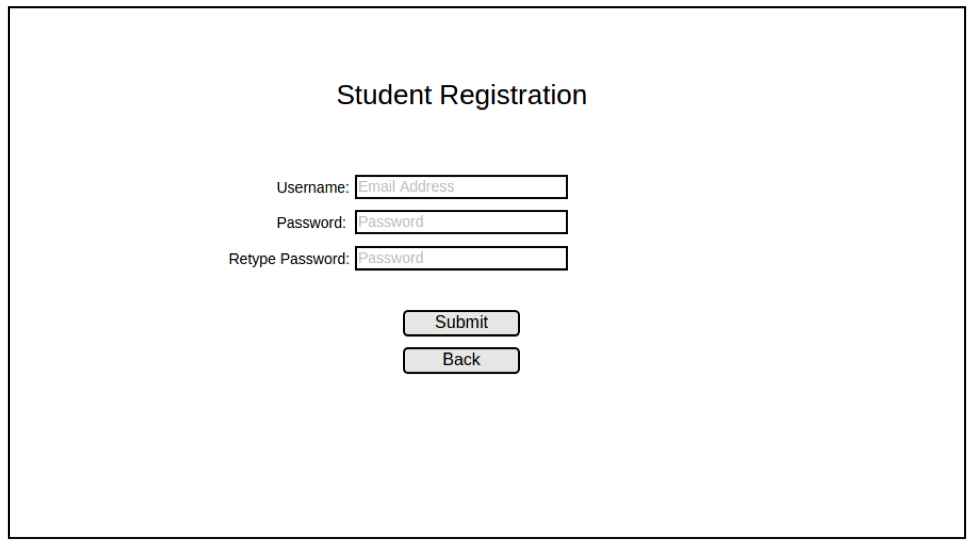
## Screen: Login

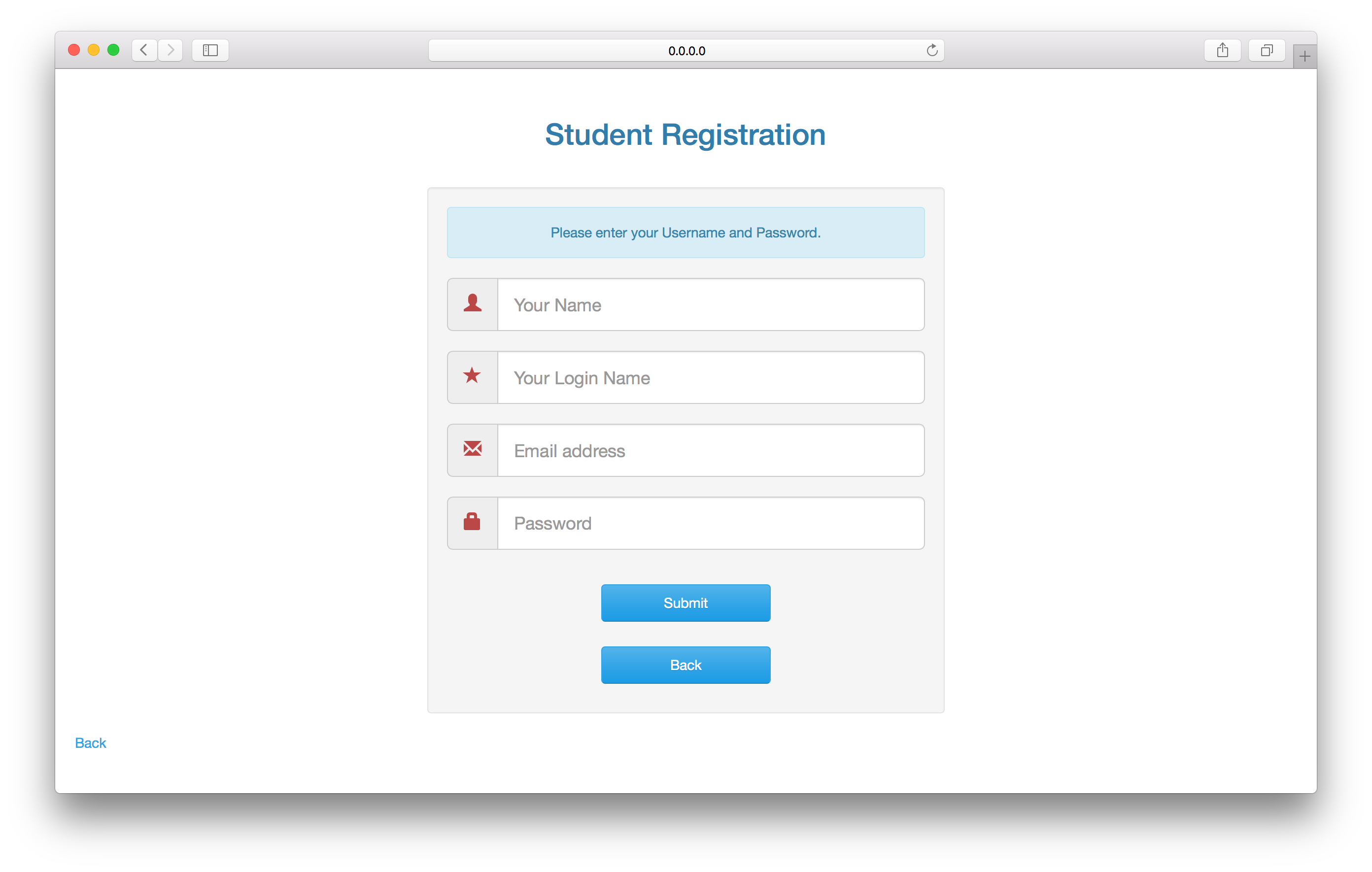




This page is mainly used for logging in the system. If the student does not have an account, he/she can create one by clicking the “register” button.

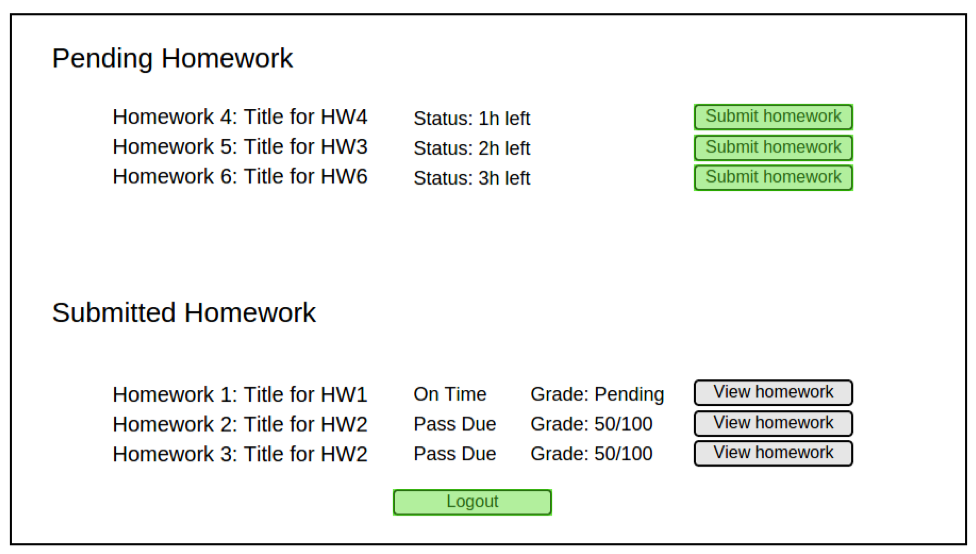
## Screen: Student Registration

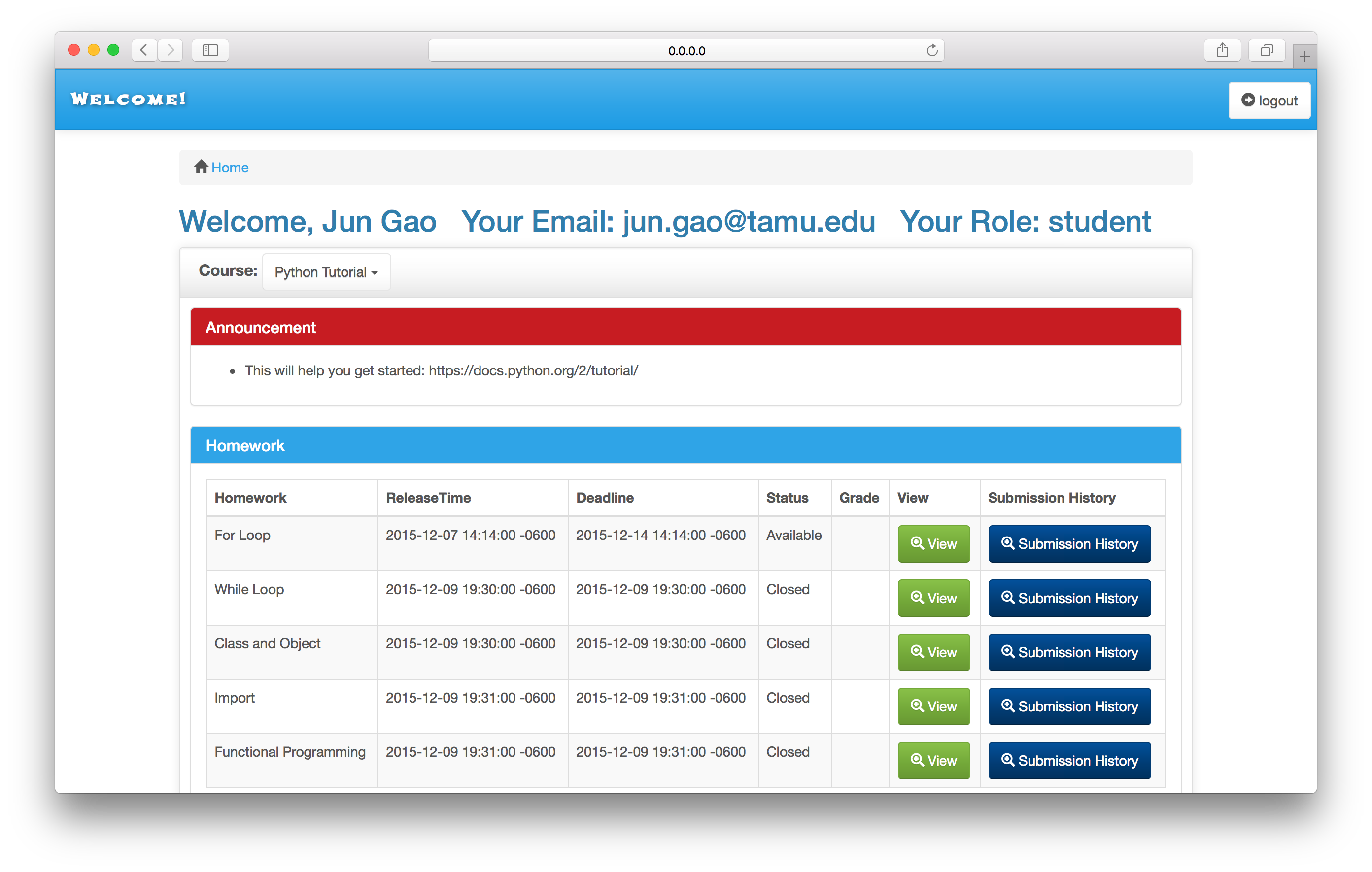




This screen is used for creating a student account.

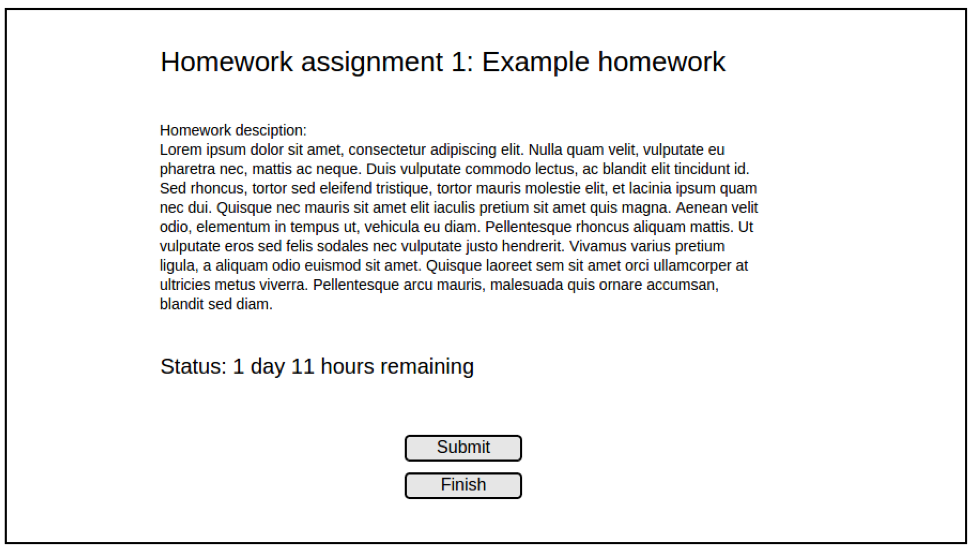
## Screen: Student\_main

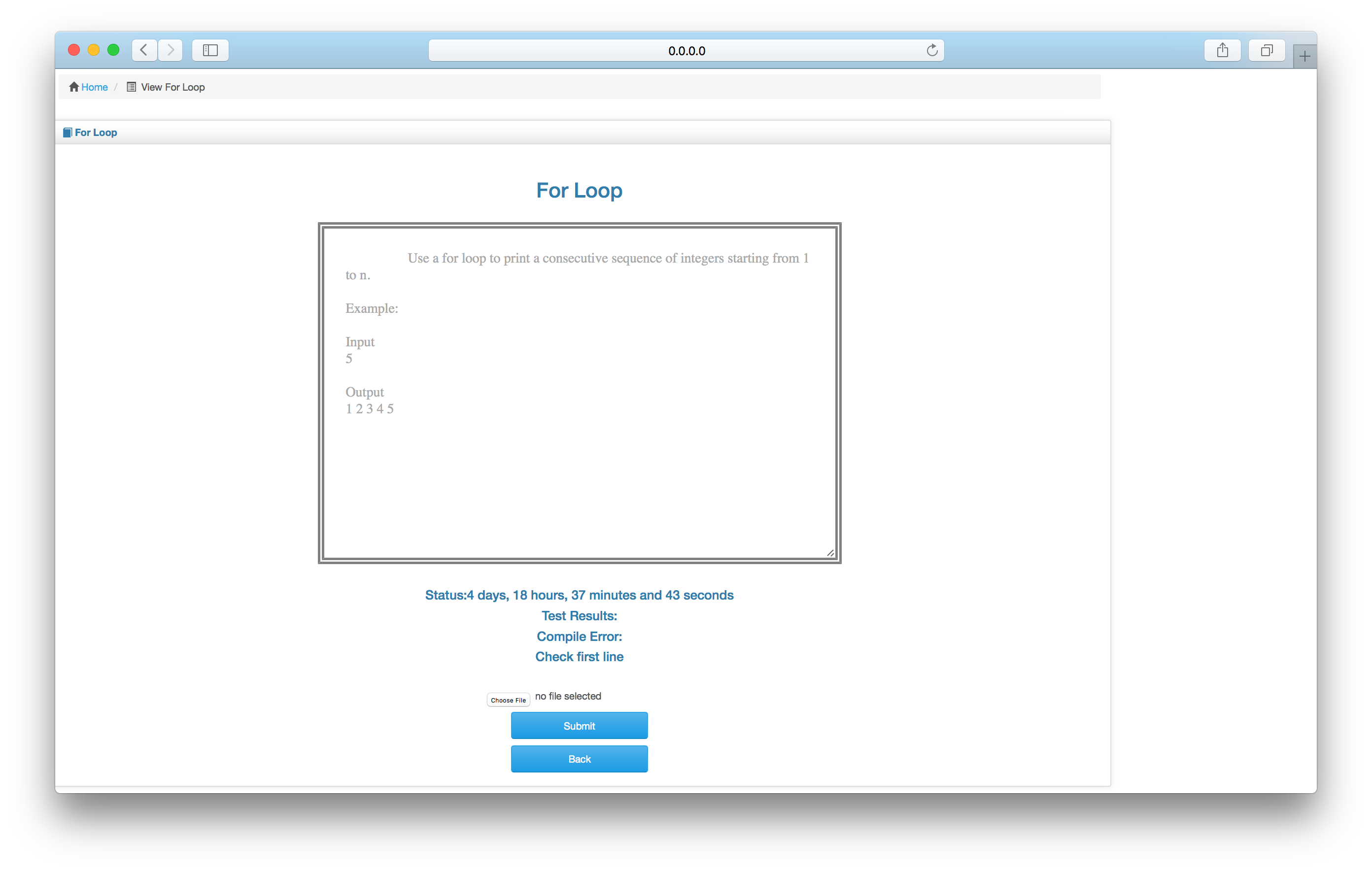




This is the student’s main screen. In this screen, student can check their submitted homework, check the due date for next homework, and submit new homework.

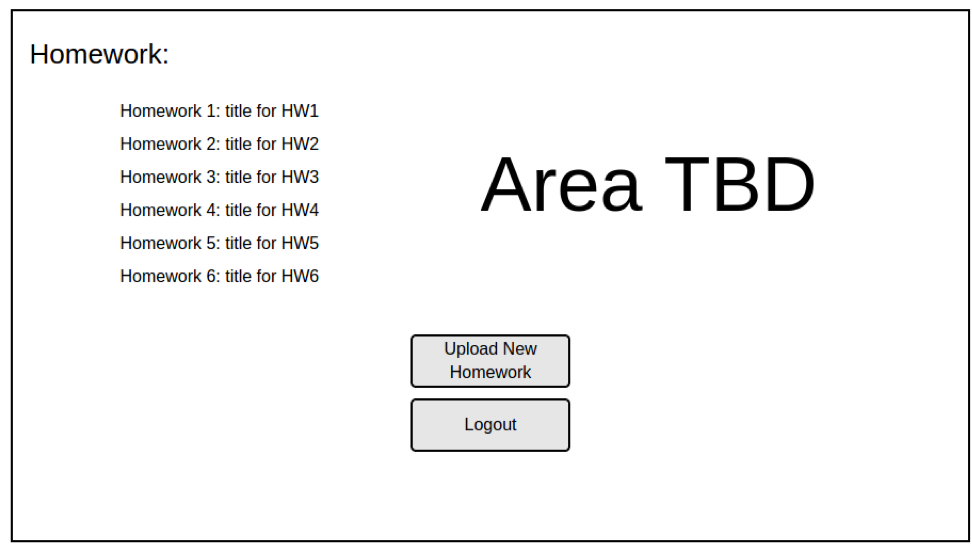
## Screen: Student\_HW\_Detail

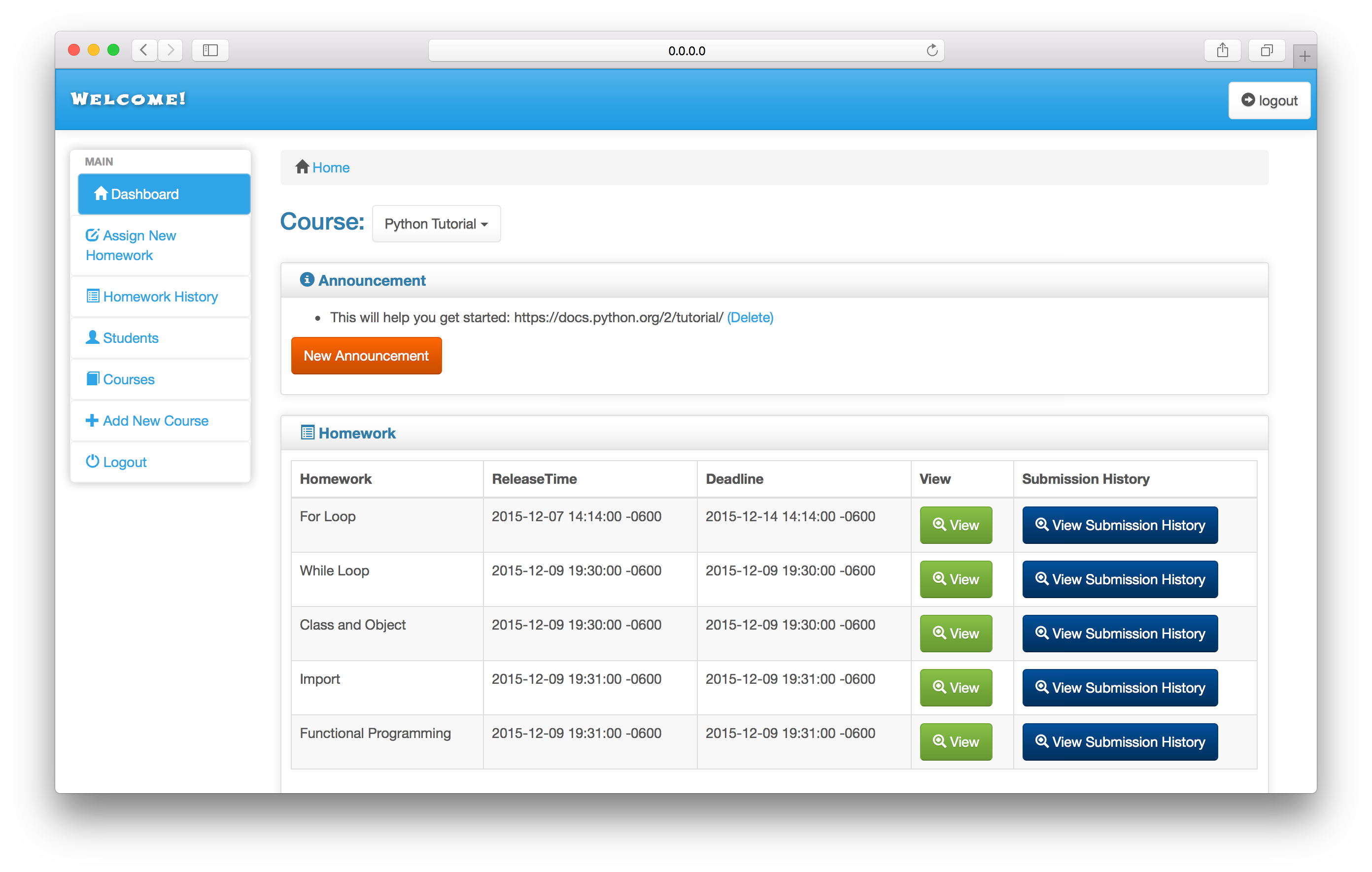




This is the homework assignment description screen. By using this screen, student can know the detail requirement about the homework, and submit the solution.

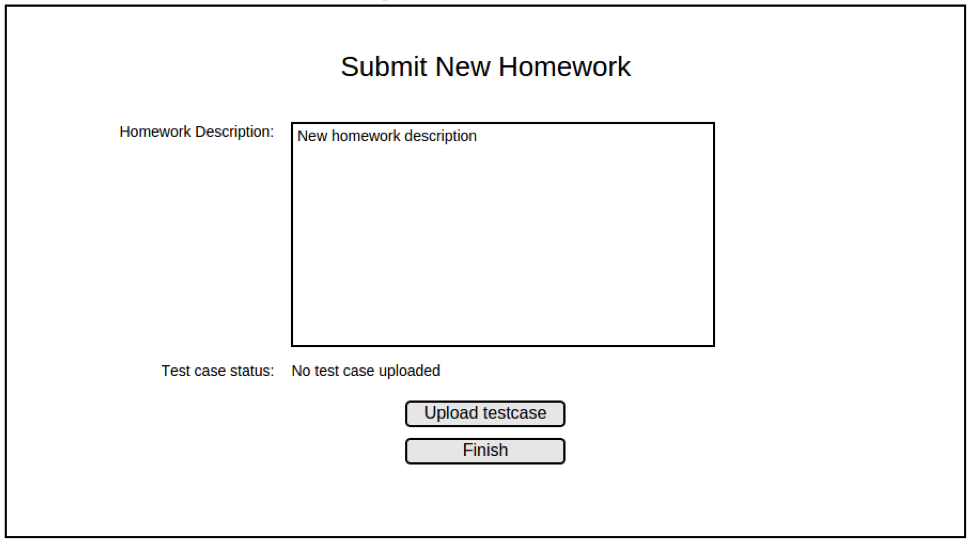
## Screen: Instructor\_main

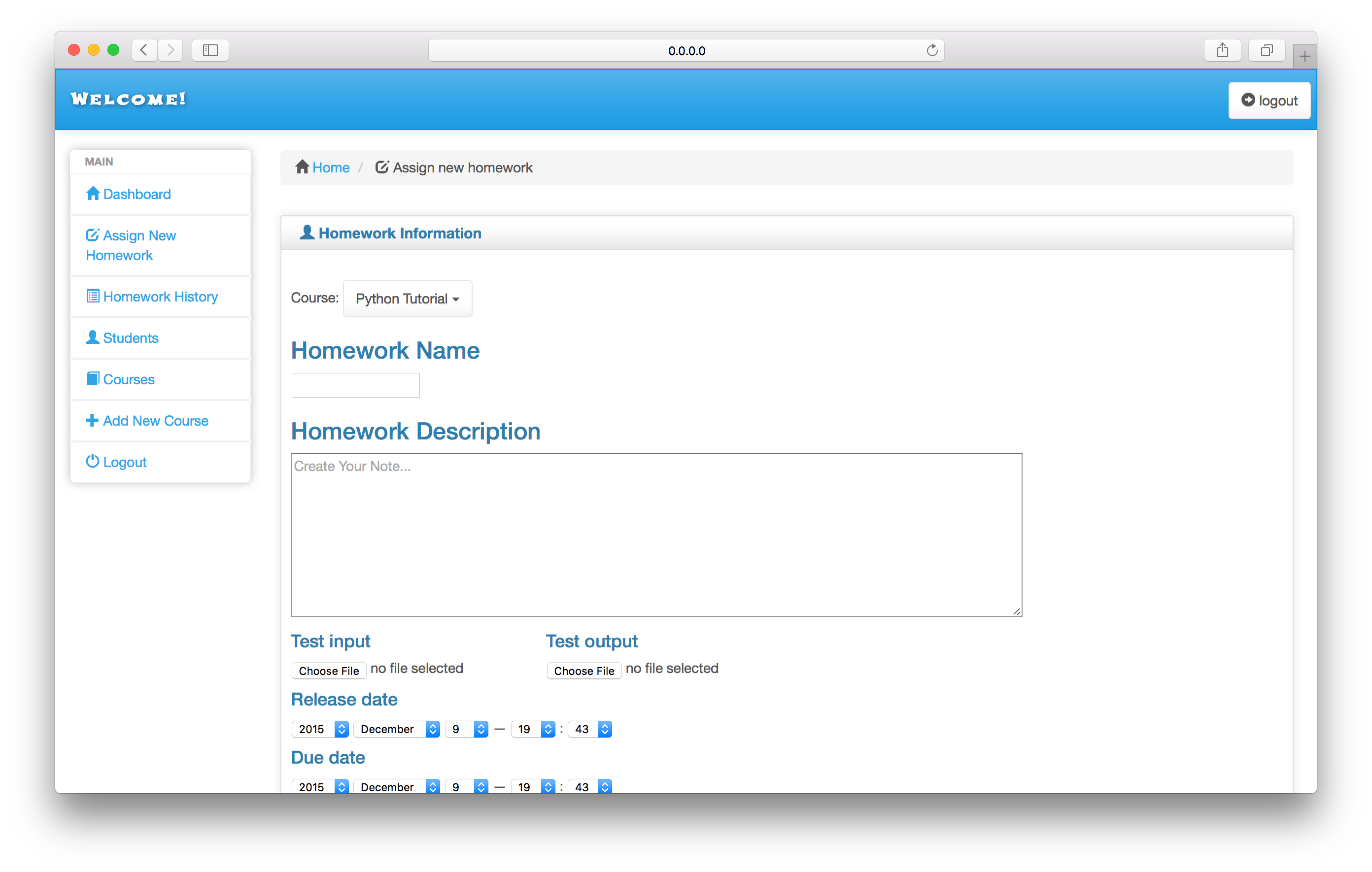




This is the instructor’s main screen. By using this screen, instructor is able to check the grade for homework done by each student, and upload the new homework assignment.

## Screen: Instructor\_new\_HW\_Assign





This is screen is mainly used for uploading new homework assignment by instructor.

# PROJECT TIMELINES AND MEETINGS

# bdd/tdd

## Benefits:

* It is of great help to ensure the correctness of the implementation of controller methods and the designs of links and JavaScript methods in the views during development.
* It reduces the time spent on positioning bugs and debugging.
* It helps us have a clear idea of the logic of dealing with the activities of users.

## Problems:

* Although it reduces the time spent on debugging, it takes much more time before we can get started on developing a new feature.
* Cucumber provides a well-designed framework to deal with BDD, but it does not provide enough methods, as a result we have to first develop the step definitions carefully. Besides, since the entity relationship is complicated, it becomes less efficient to design the test case.
* Really not sure if Cucumber and RSpec helps Agile Development since we believe these tools still need to improve so that the developers can easily implement the test case instead of developing the tedious step definitions first.

# Configuration management

The version control of the project is using Git, which is conducted by GitHub.

In order to avoid multiple files conflicts, we decided to use only one branch, the master branch. Every developer should pull from the repository each time before their development, and also push their updates immediately after development.

# TOOLS/gem used

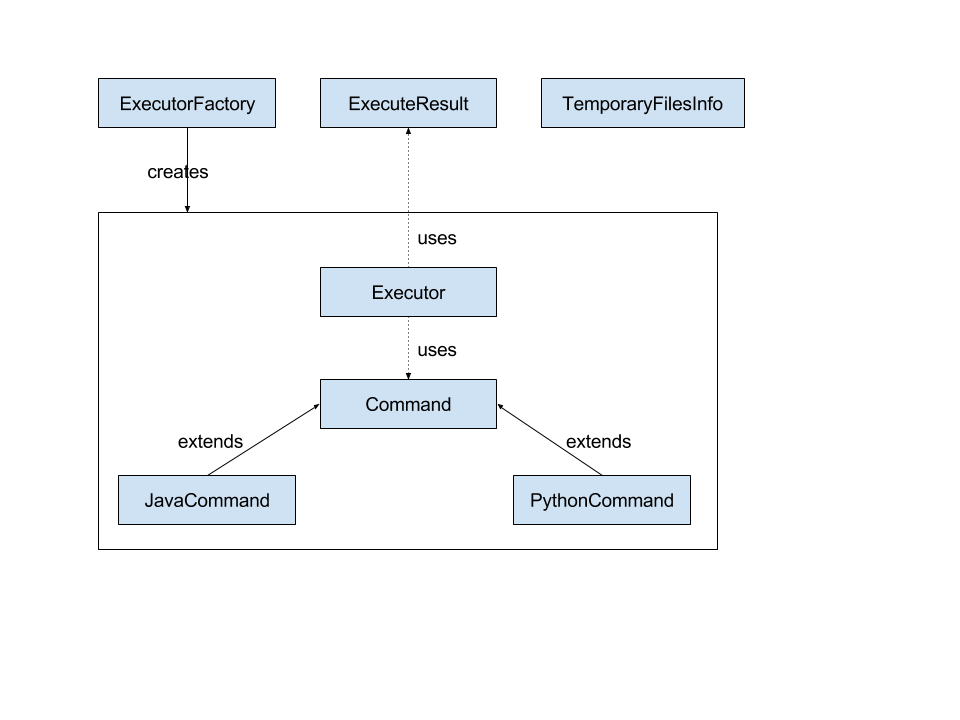
## Major Tools:

Ruby 2.2.2, Rails 4.2.1, Bootstrap 3.2.0, SQLite3, GitHub, Rspec, Cucumber, JVM.

## Gems:

*# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'*gem 'rails', '4.2.1'  
*# Use sqlite3 as the database for Active Record*gem 'sqlite3'  
*# Use SCSS for stylesheets*gem 'bootstrap-sass', '~> 3.3.5'  
gem 'sass-rails', '~> 5.0'  
*# Use Uglifier as compressor for JavaScript assets*gem 'uglifier', '>= 1.3.0'  
*# Use CoffeeScript for .coffee assets and views*gem 'coffee-rails', '~> 4.1.0'  
*# See* ***https://github.com/rails/execjs****#readme for more supported runtimes  
# gem 'therubyracer', platforms: :ruby  
  
# Use jquery as the JavaScript library*gem 'jquery-rails'  
*# Turbolinks makes following links in your web application faster. Read more:* ***https://github.com/rails/turbolinks***gem 'turbolinks'  
*# Build JSON APIs with ease. Read more:* ***https://github.com/rails/jbuilder***gem 'jbuilder', '~> 2.0'  
*# bundle exec rake doc:rails generates the API under doc/api.*gem 'sdoc', '~> 0.4.0', group: :doc  
  
*# Use ActiveModel has\_secure\_password  
# gem 'bcrypt', '~> 3.1.7'  
  
# Use Unicorn as the app server  
# gem 'unicorn'  
  
# Use Capistrano for deployment  
# gem 'capistrano-rails', group: :development  
  
group* :development, :test **do** *# Call 'byebug' anywhere in the code to stop execution and get a debugger console* gem 'byebug'  
  
 *# Access an IRB console on exception pages or by using <%= console %> in views* gem 'web-console', '~> 2.0'  
  
 *# Spring speeds up development by keeping your application running in the background. Read more:* ***https://github.com/rails/spring*** gem 'spring'  
**end***group* :test, :development **do** gem 'cucumber-rails', :require *=>* **false** *# database\_cleaner is not required, but highly recommended* gem 'database\_cleaner'  
**end**

# automated compIler design

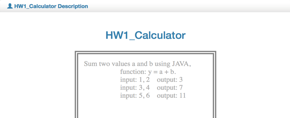
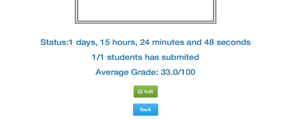
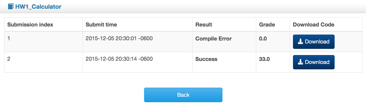
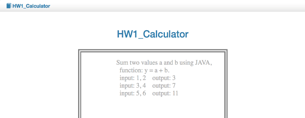
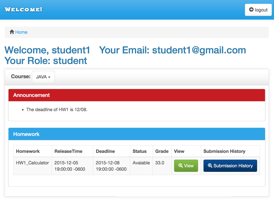
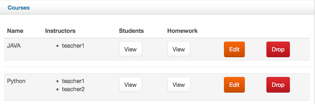
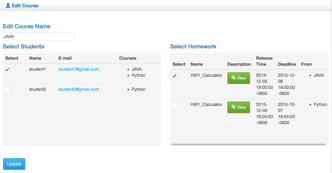
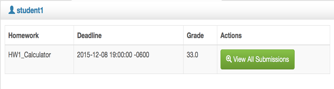
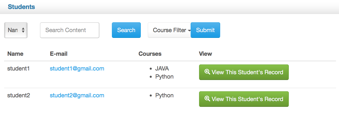
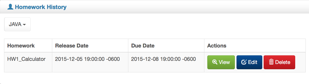
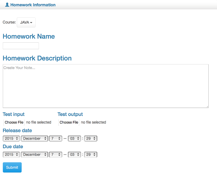
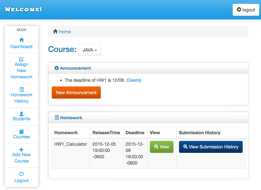
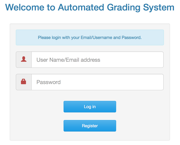
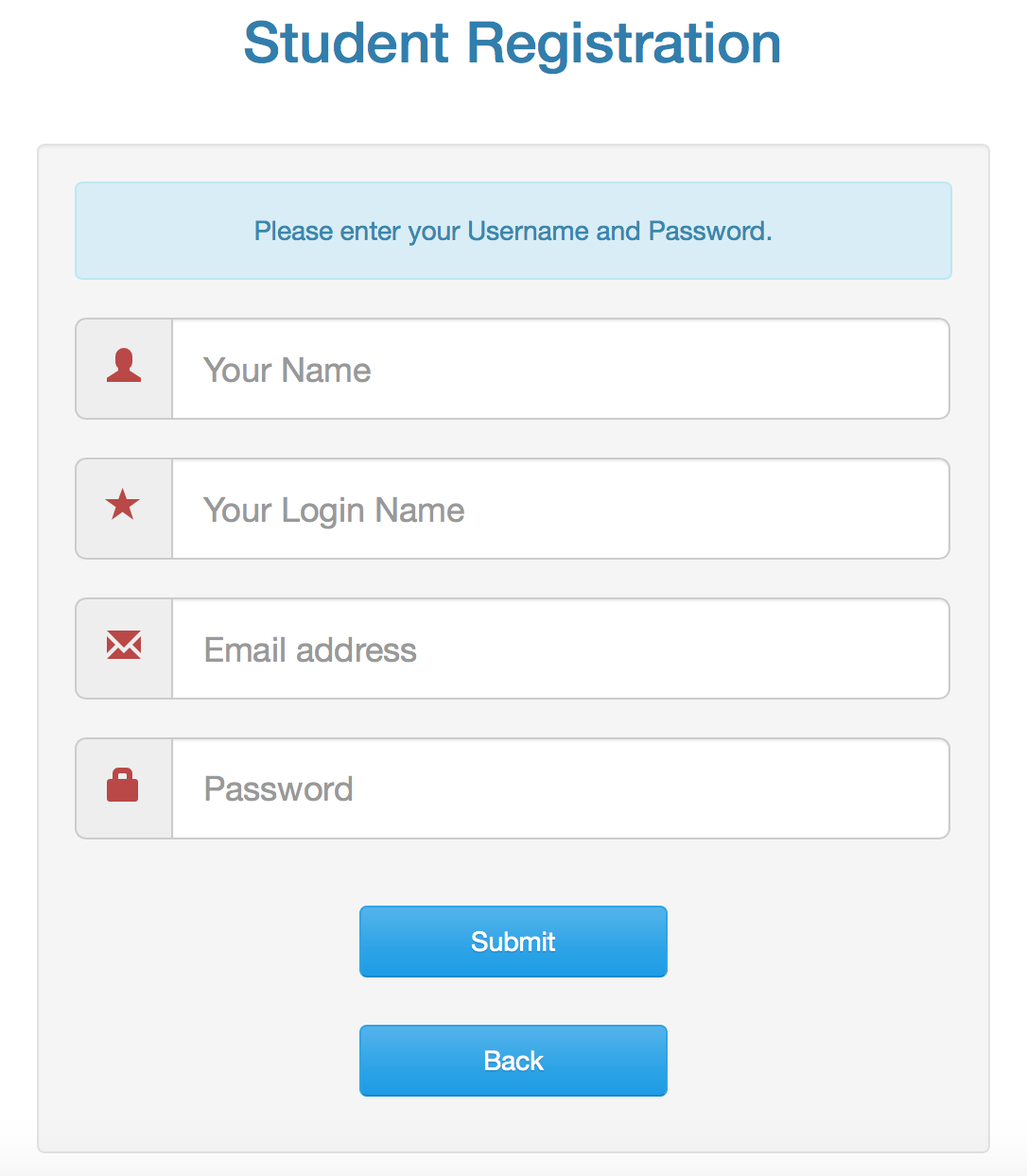


# project overview



**Teacher**

**Student**



**Add**

**new**

**course**

**Add new**

**assignment**

**Homework history**

**Check homework information**

**Students’**

**record**

**Download submissions**

**Courses**

**Edit course**

**Upload**

**homework**

**Check results**



The entire project is designed as above.

Users need to login first. Students will see the dashboard after logging in. They can see the announcements and the homework lists of their courses. Via the View buttons of the homework, students can see the details of homework and submit their code to the server. Via the Submission History buttons of the homework, students can see their submissions list which includes the results, the grades and the code of the programs they submitted.

After logging in, the teachers can see the dashboard includes the announcements and homework list of each course and they can create new announcements or delete announcements of the courses. And they can check the homework details and submissions of homework.

There is a left menu on the pages of teacher. This menu links to the other pages including assigning new homework, homework history, students, courses and creating a new course.

Teachers can assign new homework to their courses on the assigning new homework page; check, edit or delete homework on the homework history page; search for and check students’ information including homework records on students page; check and edit the information of courses or drop the courses on courses page; and creating a new course on add new course page.