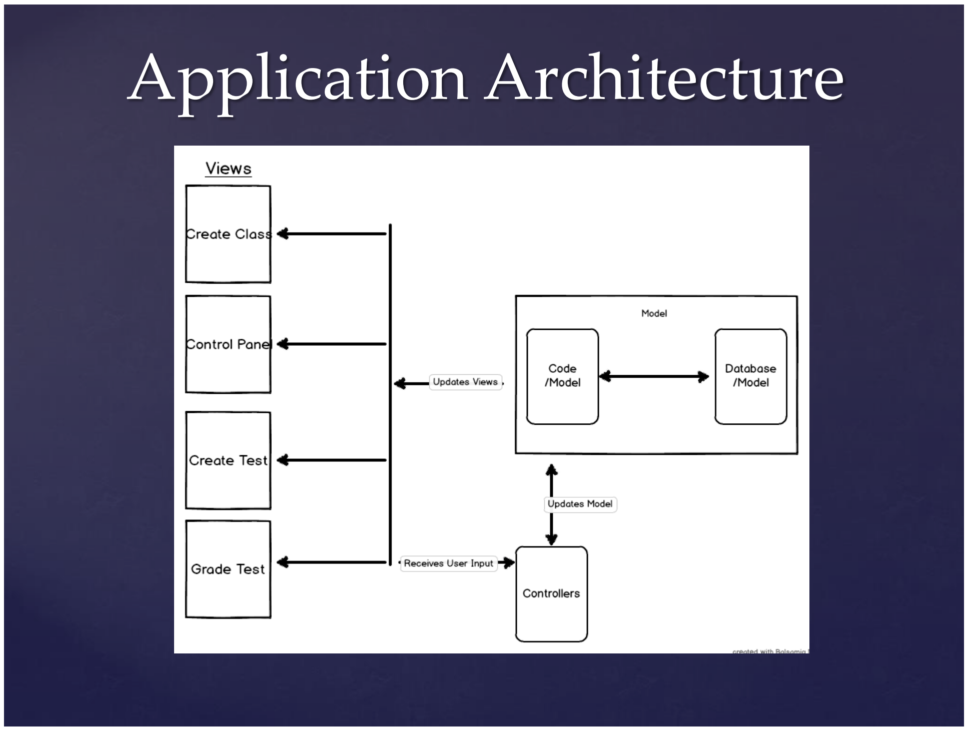
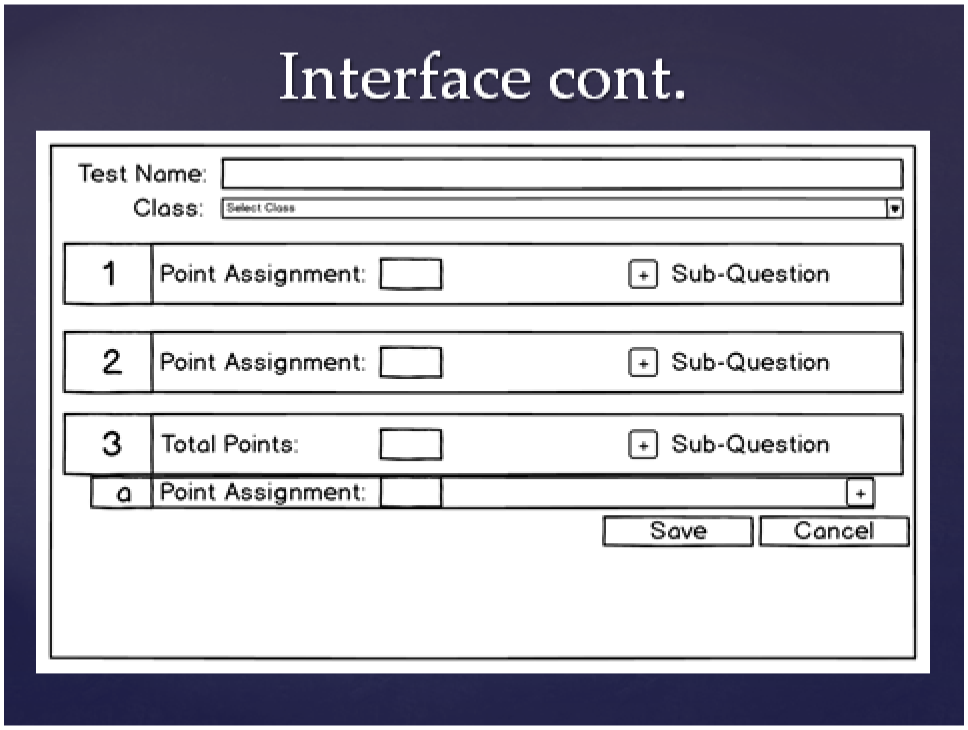
Software Design/Architecture

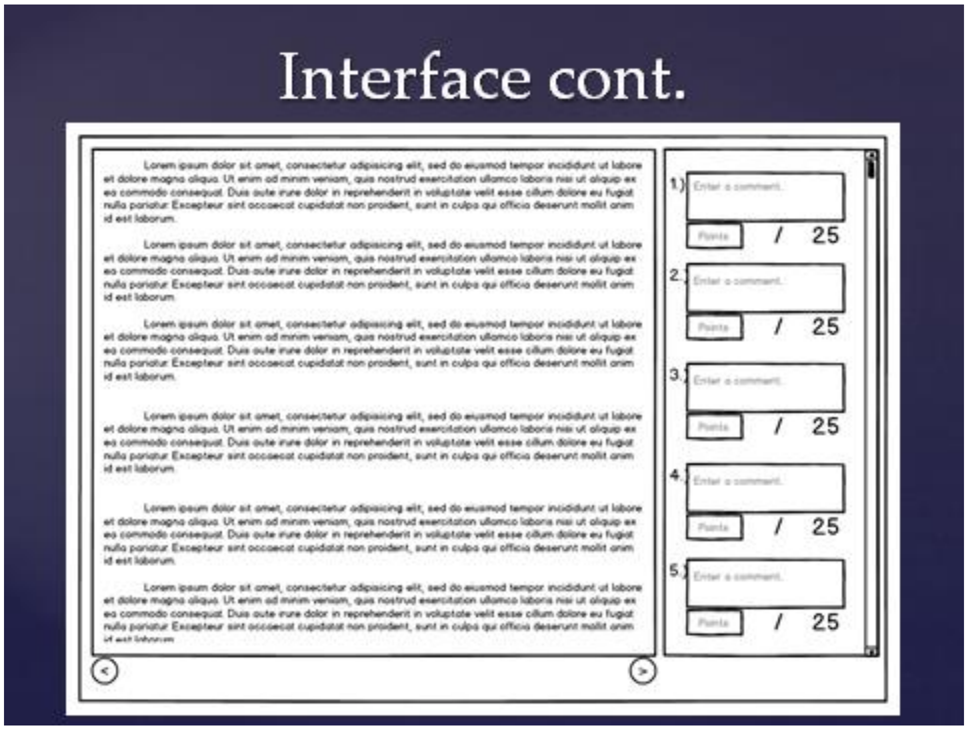


The application will follow a basic MVC framework that is enforced by the Django Framework. Views will be created that receive information from our database to synthesize a page to display to the end user. Each view will act like a client to a designated controller or set of controllers to take input from the user and update the database(model) accordingly.

User Interface Design



This is the test creation interface. A user will be able to give a test a name, assign it to a class, add questions, add sub-questions, assign point totals, and save the test once they are done.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

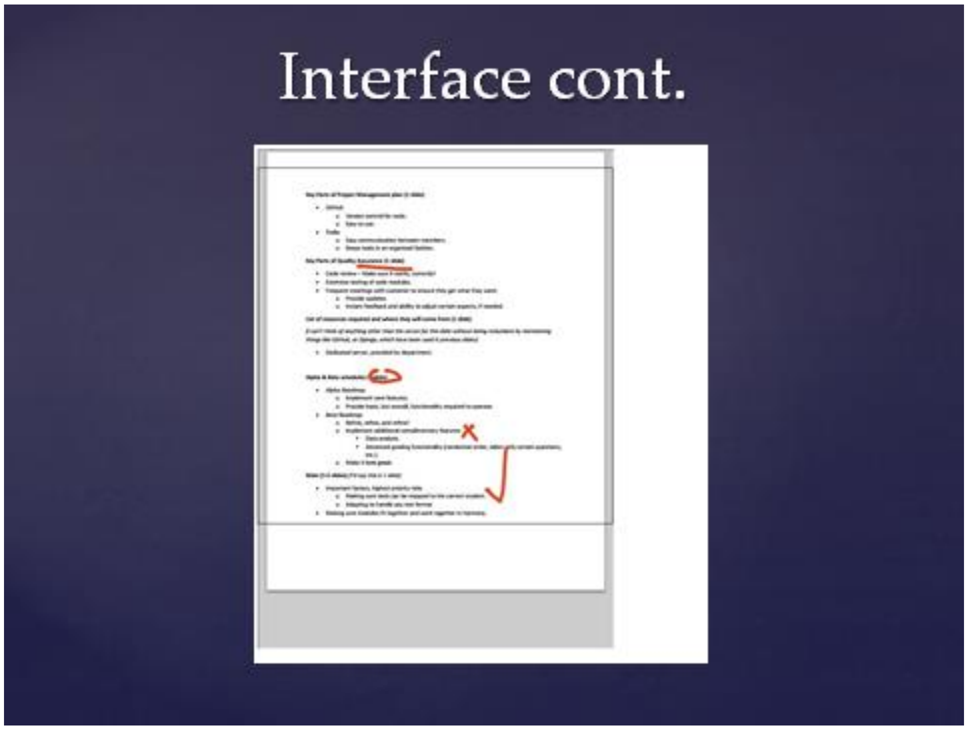
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

This is the test grading interface. On the left hand side a page from the test being graded is displayed. Below it are buttons to go forward and backwards between pages of the test. On the right hand side there is a scrollable box that contains text area inputs for leaving comments about questions on the page. There is also an input box to assign points for each question.

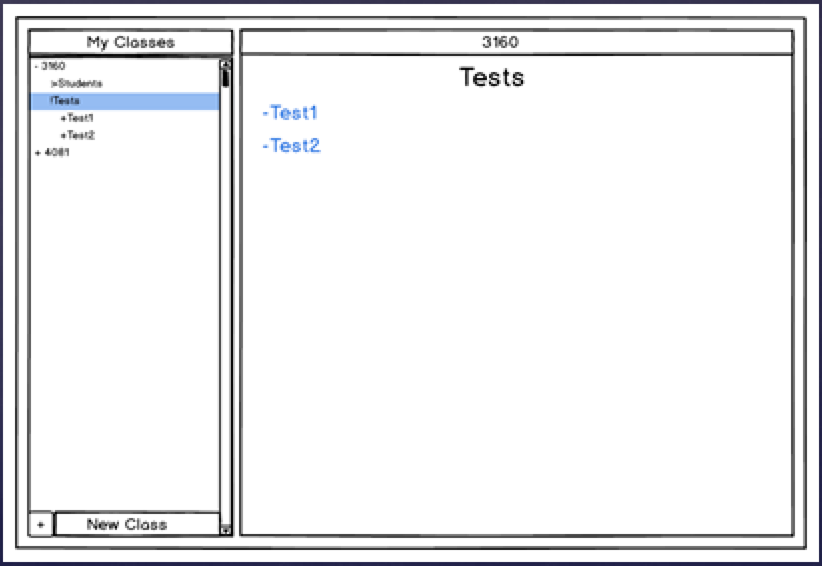


Interface cont.

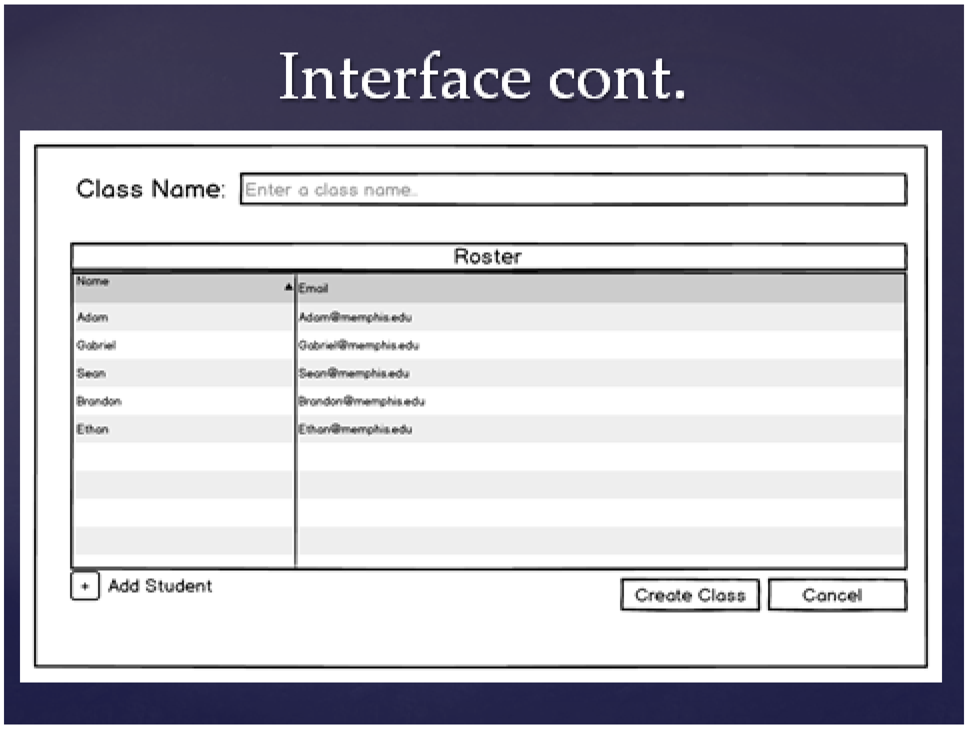
[illegible]

This is an example of our annotations feature for the test grading interface. Users will be able to draw and scribble upon the test page to leave feedback in conjunction with text comments.

Interface cont.



This is the user control panel after they login. In the left-hand pane they will be able to view a list of their classes. Class items can be expanded to reveal sub-lists of students and tests. Students can be further expanded to list each student in the class. Tests can be further expanded to list each test for the class. Students and Tests can be selected to view more details about each. The right hand pane is where all content about students, tests, and classes will be displayed. As well on the left hand pane a new class can be added by clicking the new class button.



Interface cont.

Class Name: Enter a class name...

Roster

Name	Email
Adam	Adam@memphis.edu
Gabriel	Gabriel@memphis.edu
Sean	Sean@memphis.edu
Brandon	Brandon@memphis.edu
Ethan	Ethan@memphis.edu

This is the interface to create a class. The first component is a textbox for the user to enter a class name. The second component is a list of students that can be added to by clicking the add student button. Once it is clicked the user will be prompted for the student's name and email address. Then there are buttons for creating the class or canceling the operation.

Project Management Plan

1.) Github

a.) Version Control

Our version control system will be git powered by the Github web service. We have a main project that each programmer will clone to their account. From there each programmer works on their own version and when done with their work will perform a "pull-request." After a pull-request is made their changes will be merged into the main project by another programmer and tested.

b.) Project Wiki

Github provides a project wiki which we will use to organize information about our project such as user manuals, setup instructions, etc.

c.) Issue & Bug Tracker

Github also provides an Issue & Bug tracker. This will be used by programmer to report issues they encounter with the application and file bug-fix requests. Every week the bug-list will be looked over and high-priority bugs will be assigned to a programmer to be fixed.

2.) Trello

a.) Group communication

Trello will be used to perform mostly unstructured group discussion that is not fit for our project wiki and bug tracker.

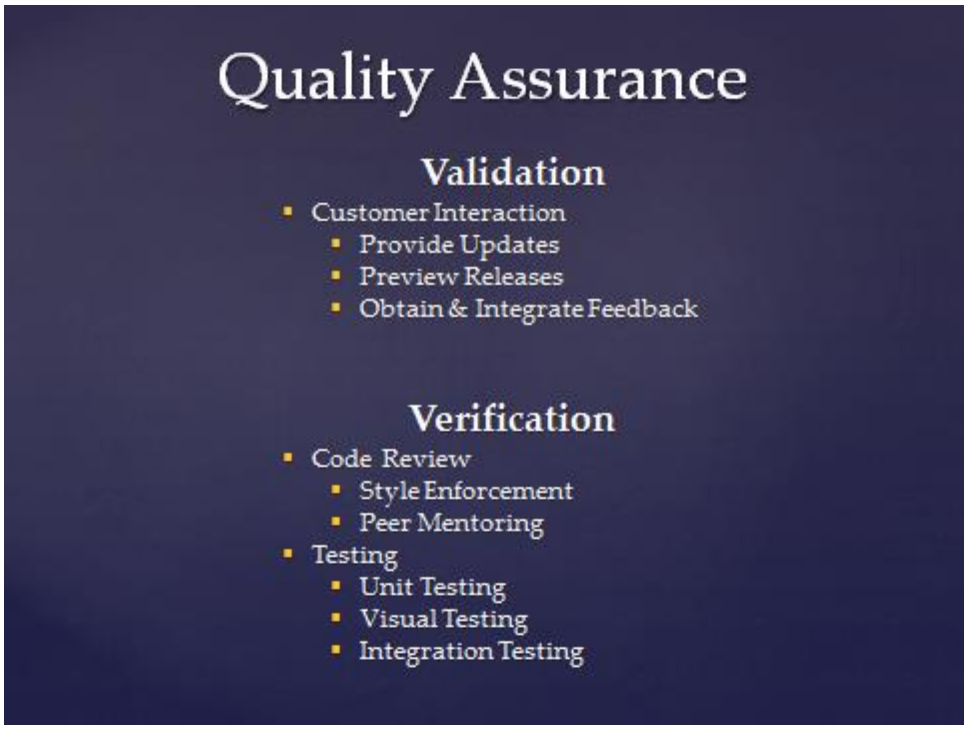
b.) To-do lists

To-do lists can be incorporated on discussions to enumerate tasks that need to be completed.

c.) File and Document sharing

We will also share screenshots, power points, etc. via Trello for collaborative work and discussion on such things.

Quality Assurance



Quality Assurance

Validation

- Customer Interaction
  - Provide Updates
  - Preview Releases
  - Obtain & Integrate Feedback

Verification

- Code Review
  - Style Enforcement
  - Peer Mentoring
- Testing
  - Unit Testing
  - Visual Testing
  - Integration Testing

1.) Customer Interaction

a.) Provide Updates

When we finish features we will provide updates to the customer in order to lead to points B and C.

b.) Preview Releases

From point A we will take new features and demo them to the customer as they are completed.

c.) Obtain & Integrate Feedback

From point B we will get feedback from the customer and integrate it into the given feature to bring it into line with their expectations. Then we will perform the Preview-Obtain-Integrate loop until it meets their satisfaction.

2.) Code Review

a.) Style Enforcement

We will maintain a consistent style for our code(Allman style bracketing) naming conventions for variables, and methods.

<u>Naming conventions</u>

Variables will follow camel-case notation.

Methods will also follow camel-case notation. Method names should be verbs.

Classes will have the first letter of each word capitalized. Class names should be nouns.

b.) Peer Mentoring

We will have code reviews and pair programming in order to expand and improve upon each team member's skillset.

3.) Testing

a.) Unit Testing

Unit tests will be created for methods in our controller code. These will be used to test our codebase after integrations of new code.

b.) Visual Testing

When a bug is reported it will be mandated, where possible, to provide a screenshot, video, or other visual means of "showing" the bug. This will make it easier for programmers to understand each other and protect us against the ambiguities of language when creating bug reports.

c.) Integration Testing

When new features are integrated into the main project our test suite will be run against the entire project and a manual test of features will also be performed.