

# Software Engineering Fall-2016

## Team A

### Team Members and Primary Abilities

#	Name	Primary Task
1	<u>Justin Goulet</u>	<u>Project Manager</u>
2	<u>Steven Clark</u>	<u>Development Team</u>
3	<u>John Orcino</u>	<u>Development Team</u>
4	<u>Jeremy Pietersma</u>	<u>UI Designer</u>
5	<u>Chris Childers</u>	<u>UI Designer</u>
6	<u>Christian Mathias</u>	<u>Analytics Manager</u>

## Role Descriptions

The participant roles are subject to change at any time. Each participant will fulfill each role at some point in time during the project. As a precursor, each participant will be assigned a primary function, denoted in the table above. Below is a description of each role.

### **Project Manager / SCRUM Master - (1 Person)**

- Oversees every part of the project. Ensures project members stay on task, meet deadlines, and fulfill requirements as assigned. Product manager is in charge of scheduling meetings, project documentation and oversees the project at every step - keeping a particular close eye on version control, commenting and code quality. Here are some example tasks that the **project manager** will take a large role in:

- Competing product design wireframes
- Managing set communication methods
  - Slack
  - GitHub Issues
- Arranging team deadlines
  - May be different than actual deadlines to increase productivity
- Quality Assurance (QA)
- Assists with any portion of the project that needs further work to meet assigned deadlines.

## **UI Designer(s) - (2 People)**

- Tasked with managing the interface of the final project. The UI Designer(s) will work with team members to bring a high quality product, in terms of design, to the portfolio.
  - Designated person to ensure design of product meets that of project manager's design documentation
  - Takes feedback from team to ensure reliability and ease of User Interface
  - Works directly with project manager to create useful and intuitive design
    - Directs project manager on overall UI using feedback from team members
    - Provides status updates as designs are completed to the team
  - Must take every detail into careful consideration. Project manager may change designs, or provide critical feedback, if change may be required.
  - Creates code for overall UI
    - Ensures panels work as expected in terms of layout
    - Ensures classes may be accessed when needed by development team

## **Analytics Manager - (1 Person)**

- Develops and maintains integrated analytics for overall project. Major tasks include creating necessary classes for custom analytic events, analytics db management where required and notifying development team of critical events that need attention. Task details include:
  - Integrating solutions such as Google Analytics, Google Firebase, or other analytics engine
  - Catching all exceptions and reporting them to the analytic event handler
  - Logging critical events throughout the project to ensure critical debugging is (almost) pain-free
    - Local file events may be helpful to determine common bugs
    - Sorting of log files may be a handy tool to create as well (independent program - written in Analytic Manager's preferred language)
  - Creating a data-flow process to inform team of common routes taken by users and where users are having issues
  - Common analytic references may be helpful:
    - Time spent on page
    - Times application loaded per day
    - Events per button
    - Sequence of views
    - Number of users per day
  - Analytics should be independent of project and should be usable on other projects without changing code.

## **Development Team - (2 People)**

- Team in charge of overall development of project. This team is designated the task of creating the functional methods and procedures of the application and will integrate the use of the developed UI from the design team. So the work is spread out, the following tasks are designated to this team:
  - Work directly with the UI Designer(s) to ensure code is synchronous and integrate smoothly
  - Creates driver class to control all aspects of the program

- The Driver program should initialize:
  - The Main Frame
  - Analytics References
  - Static/Const variables
- Develops the functionality for each action in the project
  - If buttons are required, buttons will need event listeners

## Project Description

Functions of the project (including original requirements and additional features):

### Summary of Project:

To create a Test-Taking application that can be used for content management and studying. We prioritized the basics of functionality and simplicity.

### Group-Set Requirements:

- Flash Cards
  - User can use study sets from current exams and study the questions before they are quizzed on them.
- Clean Design
  - Needs to be pleasant to use by the user. It should be clean, flat, and modern to match current design principals
- Course Management
  - The user should be able to separate courses and tests so that there is proper management of questions and they can be sorted accordingly. Users shouldn't have a bunch of rambled tests with no way of easily looking for them.
- Mac / Windows Compatible
  - We will use Oracle's Java Development Language. This way, the installed JRE on either MAC or Windows will properly run the files and there is no need for additional information/compilation mechanisms.
- Quizzes

- Users should be able to practice quizzes/tests within courses for additional studying. They should be able to take a quiz and have their scores recorded.
- Efficient
  - The program should open and close seamlessly and quickly. Should not be a large burden on older machines.
- Ease of Use
  - Users should be able to navigate easily and manage content easily. The user should not need an additional instruction manual to learn how to use this product.
- Single-User Database
  - To allow branching in the future, we must have a database that stores content for a particular user. This database will hold all materials related to said user and should load/save content when required.

## **Additional Features**

- Administration Login System
  - To branch from our single user implementation, we would plan to add an administration system where users and admins would have different permissions. This would be so admins to courses can create tests/quizzes and would authorize particular students (users) to take them. Data would then be shared between the two parties.
- Scheduling / Calendar Implementation
  - Along with the admin/user implementation, there would be an option for scheduling events, such as deadline for quizzes to be taken. This would be managed by admins and abided to by users.
- Study Game
  - A fast paced study game for users to ensure they know the material they were yearning to learn.
    - This would be a sheet of questions, each with a drop down with possible answers.
    - Each correct answer provides a point and additional time
    - Each incorrect answer reduces a point and time from the clock
    - The game ends when time runs out

- Scores can be opted to be sent to admins for possible EC/points

## Links

All links associated with the project will go here. These include items such as repository instructions, IDE setups, additional downloads and similar. For example, I placed the GitHub repo in the list below:

- [GitHub Repository](#)
- [Slack Team](#) - Send [email](#) for invite
- [GitHub Help](#)
- [Markdown Syntax Assistant](#)
- [Google Docs](#)

## Repository Info

The repo is currently set to an organization, [Software-Engineering-CSUSM](#) so that it is not tied to a single individual, for credit purposes. As such:

- All contributors will be known as **collaborators** instead of owners.
- All contributors should create their own branches, which will be merged to the master when required.

## Slack Info

Our team has been using [Slack] (<https://slack.com>), a team-based communication platform.

- Our team link is: [csusm-se-f2016.slack.com](https://csusm-se-f2016.slack.com)
- To join, please send a message to the [Project Manager](#)

We may use Slack integrations. Feel free to add any you like

## Project Start

Official Project Start Date: August 30, 2016