# Problem

Universities desire to teach software security but it becomes difficult to do this without hands on training. Setting up multiple computers for students to target is expensive and time consuming. Even if this is done, students don’t have a way to reset machines on their own. Even if students complete the challenges, there isn’t a good way to grade their performance without watching them. Students need a portal to launch exercises, reset machines that they break, and have unique answers to submit. Teachers can then login to the application to see exercises students have completed. The application also informs the teacher if any students submit answers that are from another student.

# Requirements

1. GUI interface for students to login, launch exercises, revert machines, and submit answers
2. GUI interface for teachers to login and view answers of students
3. Each exercise has a unique hash based on user, exercise, and admin private key
4. There must be at least 3 web exercises
5. There must be at least 3 desktop application exercises
6. The application should only allow a student to launch one exercise at a time
7. The application should be multi-threaded with locks on critical functions
8. Buttons pressed should give a “waiting” sign, not receive input, and have a “kill” button.
9. The application must be developed securely with static analyzer and must undergo scanning from OWASP ZAP. This application should be difficult to exploit or DOS.
10. This application should be extremely easy to setup. Every time the project is updated, Jenkins will run a Sonqarqube scan to ensure no new findings have been added. Then it will do a full build on a bare Centos7 system and run all tests to ensure functionality. If all tests pass, the production build will be updated. Anyone wanting to use this application should just have to download my repository and run a build script in bash. All dependencies will be installed.
11. This application can manage KVM machines. Users can put exported virtual machines from virtual box in here and they will show up for all students

# Architecture Design

This will be a Django project that interfaces with docker to launch virtual machines. See below for architecture drawings.

# Development Prerequisites

* Jenkins and Sonarqube
* Testing Driven Development
* Agile principles
* Django framework knowledge

# Development Plan

Agile development

What I will be doing on a daily basis

* Test driven development
* Working 2 hours per day during the week
* Am I learning and working efficiently?
* Does what I’m doing add value? Does it look and feel nice?
* What are my biggest roadblocks?
* How much am I actually accomplishing per sprint?

How should I track this and report status?

* I will put this in TFS
* I will give you a demo and summary bi-weekly of my progress. You will give me feedback and Ill update my backlog

# Senior Project Documentation Requirements

What do I have to document for Fall?

What do I have to document for Spring?