

**Software Requirements Specification**

**for**

**Python Onboarding**

**For Incoming Students**

**(POFIS)**

**Version 0.1**

**Prepared by**

**Group Name: Saab**

|  |  |  |
| --- | --- | --- |
| **Rebecca Daniel** | **11585986** | **rebecca.daniel@wsu.edu** |
| **Andrew Cornish** | **11571297** | **andrew.cornish@wsu.edu** |
| **Samuel Dunn** | **11453733** | **samuel.i.dunn@wsu.edu** |
| **Abdi Vicenciodelmoral** | **11554779** | **a.vicenciodelmoral@wsu.edu** |

**Date:** **10/25/2019**

Contents

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 2

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3 Specific Requirements 6

3.1 External Interface Requirements 6

3.2 Functional Requirements 8

3.3 Behaviour Requirements 10

4 Other Non-functional Requirements 11

4.1 Performance Requirements 11

4.2 Safety and Security Requirements 11

4.3 Software Quality Attributes 11

Appendix A – Data Dictionary 13

Appendix B - Group Log 14

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Authors** | **Description of Version** | **Date completed** |
| 0.1 | Rebecca Daniel  Andrew Cornish  Samuel Dunn  Abdi Vicenciodelmoral | Initial version  Rough concepts added | 08/10/19 |
| 0.2 | Rebecca Daniel  Andrew Cornish  Samuel Dunn  Abdi Vicenciodelmoral | Further details and requirements added | 11/10/19 |
| 0.3 | Rebecca Daniel  Andrew Cornish  Samuel Dunn  Abdi Vicenciodelmoral | Details and steps added. Additional requirements added. | 15/10/19 |
| 0.4 | Rebecca Daniel  Andrew Cornish  Samuel Dunn  Abdi Vicenciodelmoral | Rough draft. Diagrams and other visual aids in work. | 21/10/19 |
| 0.8 | Rebecca Daniel  Andrew Cornish  Samuel Dunn  Abdi Vicenciodelmoral | Uploaded to GitHub | 24/10/19 |

**1 Introduction**

Section one provides the purpose and scope of the document, the intended audience and general structure, acronyms and abbreviations use, and a general overview of the structure of the SRS.

**1.1 Document Purpose**

The purpose of this document is to provide a detailed software requirement specification (SRS) for the Python Onboarding For Incoming Students (POFIS), release number 0.1. This document will detail the purpose, functionality, features and interface of POFIS. It will also provide general system specifications for proper operation and use of POFIS.

**1.2 Product Scope**

The goal of the POFIS system is to help incoming and current Washington State University – Vancouver (WSU-V) campus students, that already have coding experience, adjust to the intricacies of the Python language. It is intended to not only assist students, but also to benefit instructors by providing custom tutorials to help those with basic questions regarding the language.

**1.3 Intended Audience and Document Overview**

This document targets the core development team for POFIS and personnel responsible for evaluating the efficacy of POFIS. The evaluation personnel include Doctor Xinghui Zhao and any teaching assistants aiding her in grading student submissions.

This document contains four sections and two appendixes. The current section is an overview of the SRS and system. Section two provides general descriptions of the system, functionality, constraints and operating environment. Section three provides specific requirements and details for the interfaces along with behavior and functional requirements. Section four specifies the non-functional requirements such as performance and security.

**1.4 Definitions, Acronyms and Abbreviations**

PII – Personally Identifiable Information

POFIS – Presidents of Future Industry Standards

Pre-Reqs – Prerequisites

SRS – Software Requirements Specification

WSGI - Web Server Gateway Interface

WSU-V – Washington State University, Vancouver

**1.5 Document Conventions**

This document follows the IEEE formatting requirements. The document is single spaced with 1” margins. Arial font is used throughout the document. The main text is size 11, while section titles are size 18 and sub-section titles are size 14. Comments and header information are italicized.

**1.6 References and Acknowledgments**

*IEEE Guide to Software Requirements Specifications*, 1st ed., The Institute of Electrical and Electronics Engineers, Inc, New York, NY, 1984, pp.1-26

**2 Overall Description**

Section two provides an overview of the POFIS system and its functionality. It also provides general descriptions of the expected users and operating environments. Furthermore, this section describes the expected design and implementation constraints, and user documentation. Lastly, it provides the assumptions and dependencies that could negatively affect the design and implementation of the system.

**2.1 Product Perspective**

Programming in any language requires an understanding of the specific language’s syntax and behaviors. For many students, it can be difficult and stressful to be expected to write programs in a language that is not familiar to them, while learning new concepts and keeping up on other studies. The many tutorials for python available across the internet are designed for a larger audience and are more general in nature. The POFIS (Figure 1) is designed to enable students that have programming experience to quickly adapt to using Python for specific course requirements.

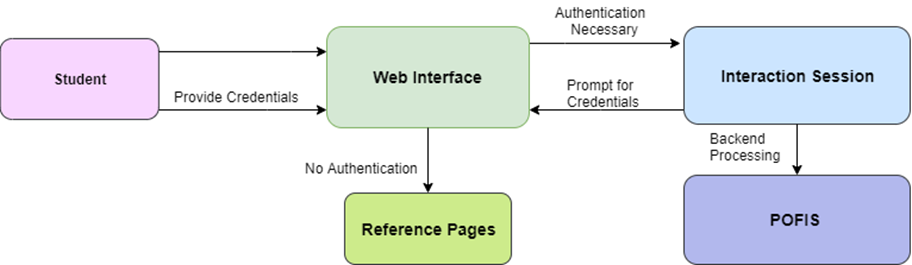


Figure 1: POFIS context diagram

**2.2 Product Functionality**

The system will provide the following major functions:

1. User functionality
   1. User log-in
      1. Optional. Provides a means for the user to save and track their progress in the interactive coding tutorials.
   2. Tutorial menu
      1. Enables users to select tutorial and bypass sections
   3. Coding window
      1. Enables user to enter Python code to attempt current tutorial step
   4. Submit code
      1. Trigger for submitting user generated code.
   5. Results area
      1. Pass/Fail results based on submitted code and expected value(s) for current tutorial step.
2. POFIS functionality
   1. User information
      1. Store and retrieve user information and progress
   2. Determine if submitted code is acceptable
   3. Course specific requirements
      1. Tutorial steps will be linked together per instructor requirements.

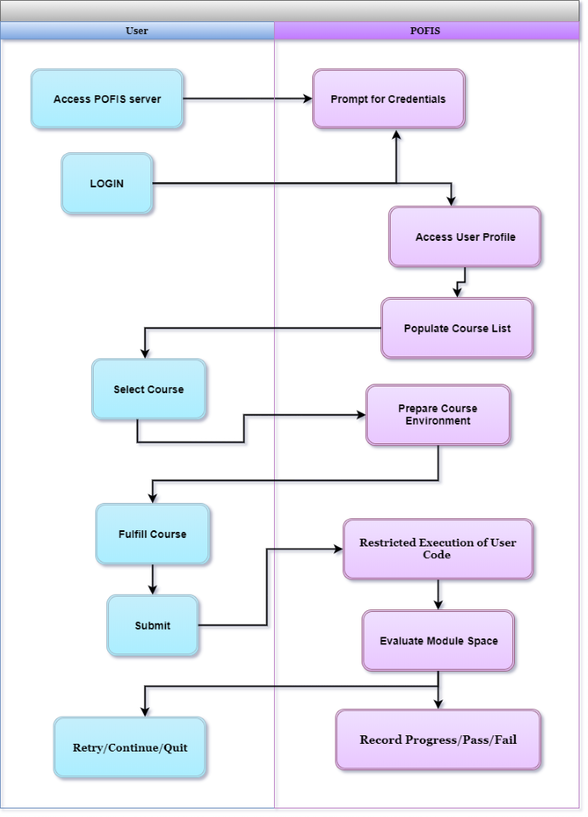


Figure 2: Data Flow Diagram

**2.3 Users and Characteristics**

The direct users of the POFIS system are WSU-V students with programming experience. It is expected that these students will be primarily Computer Science majors thought it could be helpful to any student enrolling in courses requiring Python programming.

Additionally, POFIS administrators will require additional access to the system in order to make changes and corrections. These users will be responsible for updating, creating, and modifying tutorials along with user account activities. This may be done through direct access to the server-side database.

**2.4 Operating Environment**

The POFIS software is a browser-based application. The application and associated web page will be hosted on a Unix based operating system using the Web Server Gateway Interface (WSGI).

Access to web page and application will require a computer capable of supporting a modern browser with JavaScript enabled. Mobile and tablet browsers will not be supported. POFIS will function on Mozilla Firefox v.68+, Macintosh Safari v.11+, Microsoft Edge v.44+, and Google Chrome v.7+.

* 1. **Design and Implementation Constraints**

Design constraints for the POFIS system include, but are not limited to:

* Keeping topics limited to beginner to intermediate skill sets.
* Security for user authentication
* Creating meaningful tutorials that encompass each topic covered

**2.6 User Documentation**

It is not expected that client-side documentation will be necessary, though a very simple “Hello World” exercise may be included as an introduction. It may be necessary to develop a guide for future system administrators once the system is fully operational.

**2.7 Assumptions and Dependencies**

The requirements of this SRS may be negatively affected by the following factors:

* WSU-V lab server
  + Server down time could negatively affect the outcome.
* Developer computers and laptops
  + If a personal computer is unavailable for any reason.

**3 Specific Requirements**

Section three provides detailed descriptions of external interfaces and functional requirements of the system. It also describes behavior requirements and provides use case diagrams.

**3.1 External Interface Requirements**

This sub-section provides detailed descriptions of the user, hardware, software and communication interfaces.

**3.1.1 User Interfaces**

All users will begin at the initial log-in screen with options to create account, log-in, or choose specific topics using the left side topics list. Creating an account is only necessary for saving and resuming progress. The user-interfaces will be the same for all users but those with accounts will be returned to their current progress.

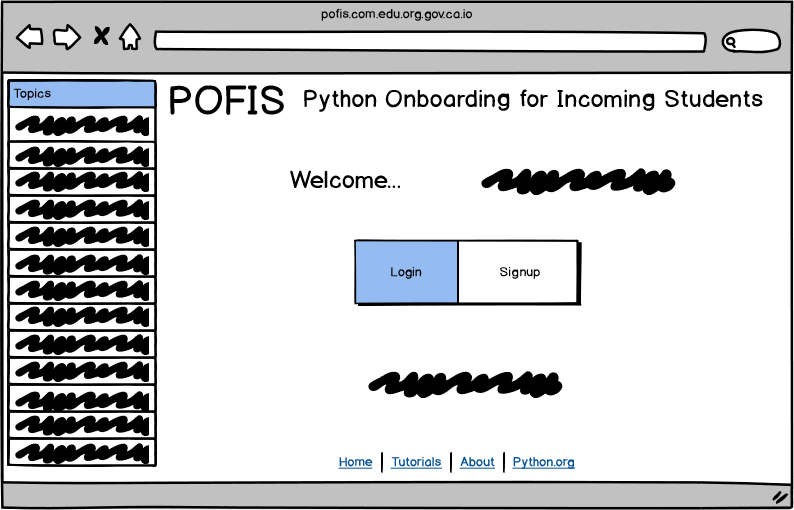


Figure 3: Home page

After the log-in screen, every user will be presented with a list of the current Python tutorials and a section with external links to related websites. The user will need to click on a tutorial link to proceed.

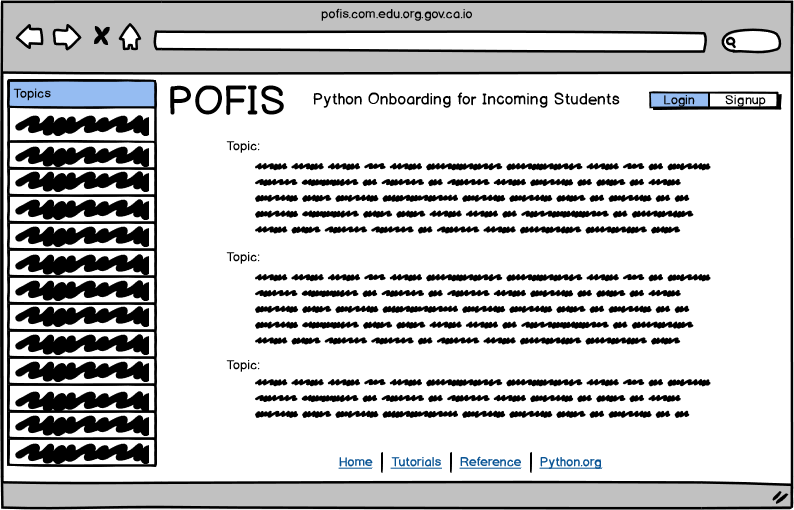


Figure 4: Topics selection

Once in a tutorial, the user will be presented with a screen split into 3 distinct sections. In the first section, instructions for the current step of the tutorial will be presented along with links to the previous or next step of the tutorial. Underneath the instructions will be the input section for the user to practice the Python code instructions. The third section will be for feedback based on the user input compared with expected input.

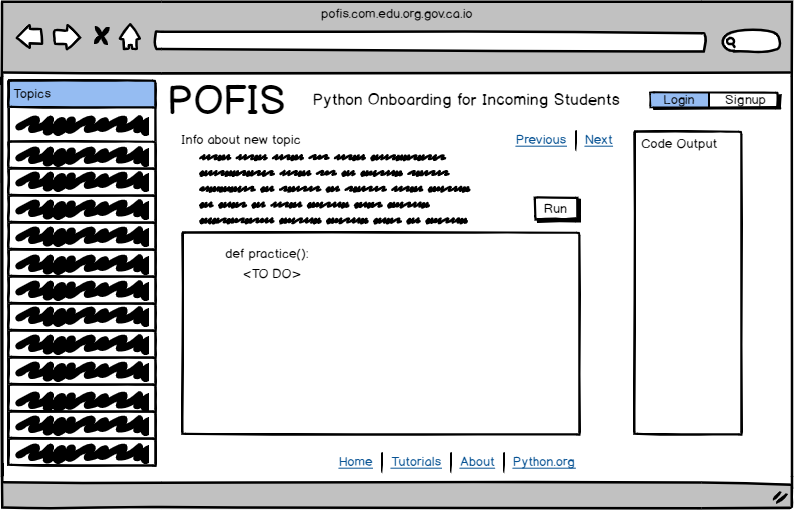


Figure 5: Tutorial selection

There will not be a different final screen upon completion of a tutorial. The layout will remain the same, but the instruction area will provide a completion statement and a link to the list of Python tutorials.

**3.1.2 Hardware Interfaces**

Access to the web page and application will require a computer capable of supporting a modern browser with JavaScript enabled. Mobile and tablet browsers will not be supported. POFIS will function on Mozilla Firefox v.68+, Macintosh Safari v.11+ Microsoft Edge v.44+, and Google Chrome v.7+.

**3.1.3 Software Interfaces**

JavaScript, HTML and CSS will be necessary for front-end development and user interfaces. Python will be utilized on the server-side to translate and process user input. Furthermore, MySQL may be utilized to store, retrieve and update user profiles and progress within tutorials.

**3.1.4 Communications Interfaces**

The POFIS system and web page will use standard HTTPS protocol for all communication between the browser and the server. No Personally Identifiable Information (PII) will be required by the user when creating a user profile.

**3.2 Functional Requirements**

This sub-section provides the intended behaviors of the system along with detailed explanations of each function.

**3.2.1 User account creation**

Note: “Pre-Reqs” is used as an abbreviation for “Prerequisites” in this section

|  |  |
| --- | --- |
| ID | UA1 |
| Actor | User |
| Priority | Low |
| Necessity | Optional |
| Pre-Reqs | None |
| Results | User account created |
| Procedure | 1. User can proceed to account creation from home and log-in screens 2. Account screen will provide a notification that PII should not be entered 3. Fields will be present to enter a unique username and password 4. Rejection of username and/or password will result in notification that username is taken, or password does not meet requirements 5. User will be returned to log-in screen or proceed directly to tutorial section if information is accepted |

**3.2.2 User account removal**

|  |  |
| --- | --- |
| ID | UA2 |
| Actor | User |
| Priority | Low |
| Necessity | Optional |
| Pre-Reqs | User has account and is logged in |
| Results | User account deletion |
| Procedure | 1. A profile link will be present on home page 2. Link will open a page with username and option to delete account 3. Clicking on Delete link will present user with a confirmation request 4. Confirmation of deletion will log out user and remove account |

**3.2.3 User log-in**

|  |  |
| --- | --- |
| ID | UA2 |
| Actor | User |
| Priority | Low |
| Necessity | Optional |
| Pre-Reqs | User has current account |
| Results | User can sign-in and save/resume progress |
| Procedure | 1. User can proceed to log-in screen from home screen 2. User will be required to enter the username and password 3. Fields will be present to enter a unique username and password 4. Rejection of username and/or password will result in notification that the information is acceptable 5. Upon verification, user will proceed directly to the tutorial section |

**3.2.4 Home Page**

|  |  |
| --- | --- |
| ID | HP1 |
| Actor | User |
| Priority | High |
| Necessity | Optional |
| Pre-Reqs | None |
| Results | Home page for POFIS |
| Procedure | 1. User will be presented log-in, account creation and tutorial options. 2. Links to additional resources about Python will be presented |

**3.2.4 Tutorial Selection**

|  |  |
| --- | --- |
| ID | TS1 |
| Actor | User |
| Priority | High |
| Necessity | Optional |
| Pre-Reqs | None |
| Results | User can select tutorial |
| Procedure | 1. User will be presented the current list of tutorials 2. Use must click on a tutorial to proceed |

**3.2.4 Tutorial Process**

|  |  |
| --- | --- |
| ID | TP1 |
| Actor | User |
| Priority | High |
| Necessity | Optional |
| Pre-Reqs | Selection of tutorial or continuation of tutorial by registered user |
| Results | User receives Python training |
| Procedure | 1. User will be presented instructions for the current learning goal 2. User enters the code per learning goal in console area 3. User must click on “Run code” button to submit code 4. User will receive feedback regarding success/failure of code 5. An option to continue will appear below instructions after an attempt 6. After first attempt, user may be presented an option for additional guidance, such as a portion of the required code or a specific tip to aid in understanding of the concept 7. User must click on the “Next” link to access further steps of the tutorial 8. If user is logged in, system will save current progress when the user proceeds to next step of tutorial |

**3.2.5 Tutorial Completion**

|  |  |
| --- | --- |
| ID | TP2 |
| Actor | User |
| Priority | low |
| Necessity | Required (upon completion of tutorial) |
| Pre-Reqs | Completion of tutorial |
| Results | User presented completion notification |
| Procedure | 1. The tutorial nomenclature and current date, along with a completion message, will be displayed upon completion. 2. A unique completion code may also be presented 3. If the user is logged in, completion of the tutorial will be saved to their account 4. A “Finish” button will be present which will return the user to the tutorial selection screen |

**3.3 Behavior Requirements**

This sub-section provides general use case diagrams showing the interaction of the users and system.

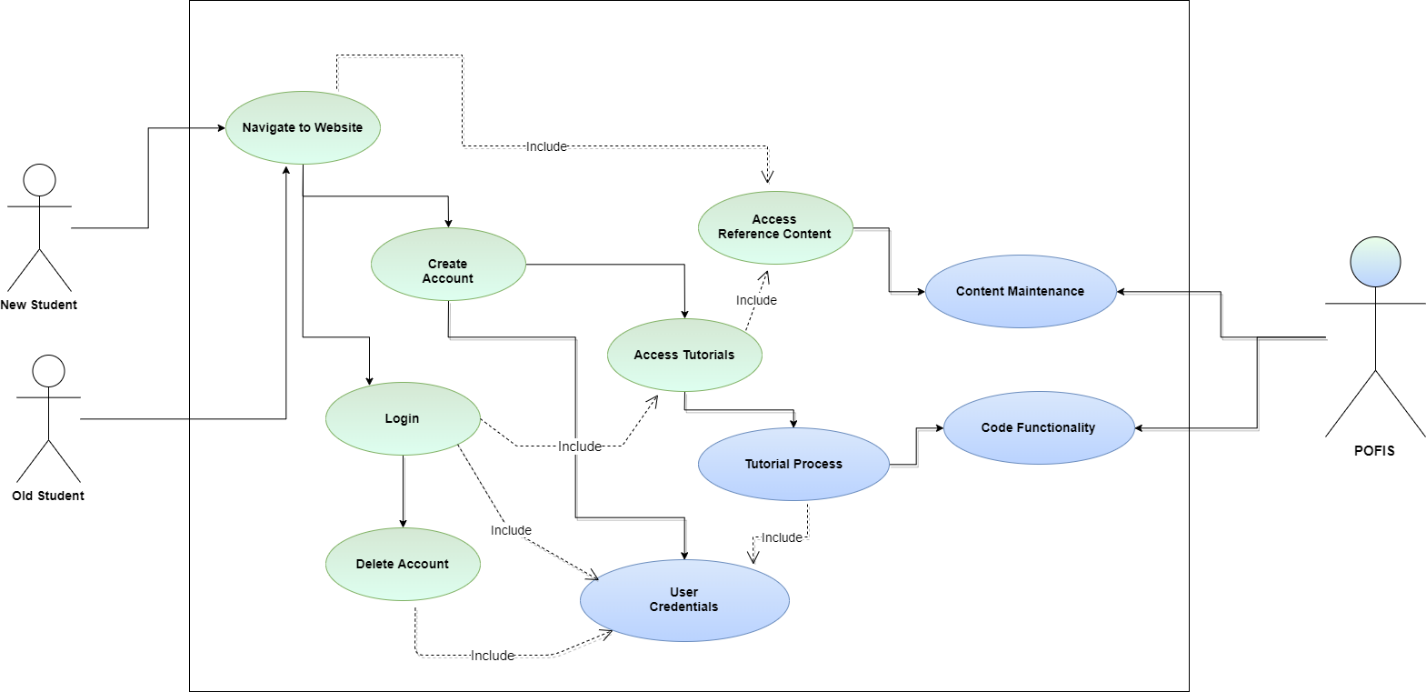
**3.3.1 Use Case View**

Figure 6: Use Case View

**4 Other Non-functional Requirements**

Section four provides descriptions about the requirements regarding performance, safety and security, and software quality attributes of the system.

**4.1 Performance Requirements**

Under normal operating conditions, the POFIS system shall be capable of the following:

1. Home and tutorial selection pages shall not take longer than 15 seconds to load.
2. User account creation shall not take longer than 10 seconds to confirm or refuse.
3. Initial session loading of any tutorial shall not take longer than 15 seconds.
4. Further steps of a tutorial should not take longer than 5 seconds to load.
5. Code submitted by a user shall take no longer than 10 seconds to receive feedback.
   1. Exceptions to this rule may occur if tutorial has extensive coding involved.
   2. If an exception is warranted, the instructions must indicate that the processing time may be longer than the norm.

**4.2 Safety and Security Requirements**

The system is to be used as an educational and informational reference only. Use of the system is not intended to be a requirement, therefore PII is not necessary for account creation. The use of passwords is still deemed necessary and will follow the following guidelines:

1. Password must be 8 to 12 characters in length
2. Passwords must contain at least one numeric character
3. Passwords may contain the following special characters: @, # and $
4. Certain passwords (such as “passw0rd”) will be refused

Safeguards will be in place to ensure that user-generated code submitted to the server does not contain malicious code and/or the server does not attempt to run suspicious code.

**4.3 Software Quality Attributes**

This sub-section provides descriptions and measurable requirements for specific software quality attributes.

**4.3.1 Availability**

The POFIS system should be available at least 95% of the time the server and associated network are functional.

**4.3.2 Correctness**

It is essential that the information in the tutorials and responses to submitted code be accurate, therefore an error rate of no more than 1% is allowed. Although there may be multiple ways create an output, the submitted code must follow the current tutorial. Code that may create the desired outcome but does not follow the tutorial will not be shown as correct. Those response will be perceived as a correct response when determining the error rate.

**4.3.3 Reliability**

The system shall return no more than two system error messages in a five-minute period while processing user submitted code at a rate of one every 15 seconds.

**Appendix A – Data Dictionary**

It was determined this section is not necessary at this time.

**Appendix B - Group Log**

10/10: Deciding Project Ideas

* Color Wheel tool and pain combinator
* Guided math quiz tool
* Python code tutorial site  Winning Idea
  + Catered towards students who have been previous introduced to coding concepts.
  + Has tutorial pages that also refer to python docs
  + Has guided tutorial pages where code is interpreted and evaluated
  + Concerns:
    - Sanitizing input for interpreter
    - Syntax highlighting in the editing environment
  + Set up project repo on github

10/17:

* Sphinx will be used to generate static HTML pages to keep a consistent theme across all pages throughout the site.

10/18:

* [https://ace.c9.io/#nav=about](https://ace.c9.io/" \l "nav=about) : Will possibly use this as the embeddable code editor within POFIS.

10/23:

* Andrew: Continue working on typing up SRS Document
* Samuel: Mock-up activity and use case diagrams and proofread SRS Document
* Abdi: Draw up diagrams and insert them in the SRS document
* Rebecca: Create website mock-up and proofread document