Degree Audit Form

for Computer Engineering Students and Advisors

Julian Callin | Pranav Bheda | Carlos Rivera

Abstract and introduction

- SCU COEN students currently use ECampus degree audit
- Problems with the current solution
  - Single font
  - ASCII art formatting
  - One long page of text
  - No tables or interaction
- Our one-webpage solution
  - Enter classes, see what has been fulfilled
  - Instant results
  - Shareability
  - Readability

Functional Requirements

- Take a student's input and give information regarding completion of degree requirements.
- Tell students whether a major requirement is complete or incomplete.
- Tell students whether a core requirement is complete or incomplete.
- Tell students whether they possess extra class credits which can be used toward education enrichment requirements.
- Persist a student's degree audit between sessions.
- Allow a student or adviser to download or print the degree report.

Requirements

Non-Functional Requirements

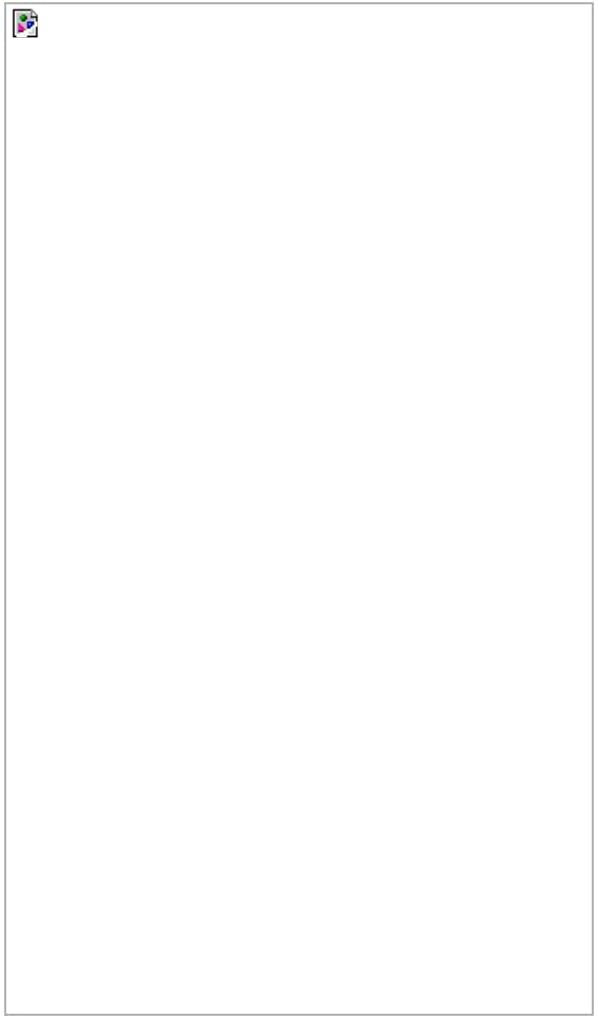
- User friendly.
- Easily testable.
- Easily maintainable.

Design Constraints

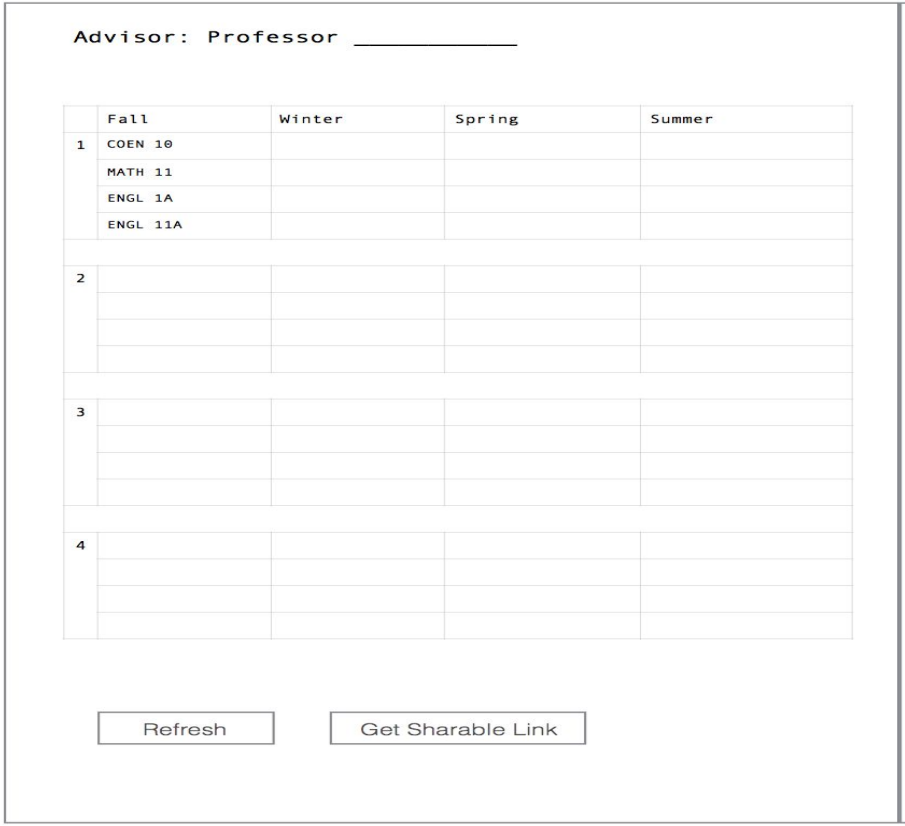
- The system must be web based.
- The system must run on the SCU Engineering Compute Center Linux and Windows machines.
- The system must retain functionality on both web browsers, Firefox and Chrome.
- The system must be completed by the week 10 presentation date.

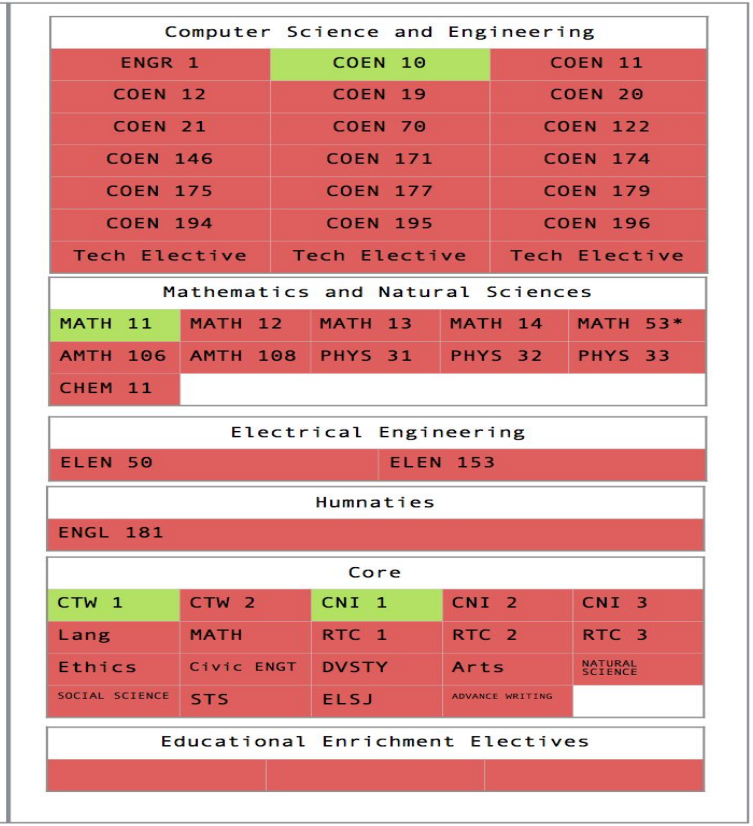
Use Case Diagram

Activity Diagram



Conceptual Model





Left side of webpage

	Fall	Winter	Spring	Summer
1	COEN 10			
	MATH 11			
	ENGL 1A			
	ENGL 11A			

Right side of webpage

Computer Science and Engineering		
ENGR 1	COEN 10	COEN 11
COEN 12	COEN 19	COEN 20
COEN 21	COEN 70	COEN 122
COEN 146	COEN 171	COEN 174
COEN 175	COEN 177	COEN 179
COEN 194	COEN 195	COEN 196
Tech Elective	Tech Elective	Tech Elective

Technologies Used

HTML: A simple, universal web markup language

Bootstrap CSS: A framework for web design and formatting

JavaScript: A popular web language

JSON: A text format for storing hierarchical data

Git: A method of source code control

Github: A host for source code and a collaborative environment

Architectural Diagram

Design Rationale

- One page UI
  - Eliminates navigation and scrolling
  - Easy cross-referencing
  - No memory of previous data
- Basic, lightweight web technologies
  - Good looking
  - Portable
  - Easy to run in any browser
- No relational database
  - Don’t need advanced queries

Test Plan

- Logic Testing
- Firefox and Chrome tests on Linux and Windows computers
- Acceptance testing
- Alpha testing with our own schedules
- Beta testing with other COEN students’ schedules

Risk Analysis

Risk	Consequence	Probability	Severity	Impact	Mitigation Strategy
Run out of time	Cut features	.4	9	3.6	Ensure team members are on schedule
Unexpected difficulty of feature implementation	Increased time to production	0.6	6	3.6	Allocate more time than necessary for component completion
Test cases will not completely cover all possible requirement combinations	User may experience unexpected behavior	0.3	8	2.4	Write unit tests or smoke tests using data, do more integration and user acceptance testing

Development Timeline

- Key Dates
  - Week 7 Demo
  - Week 10 Final Presentation
- Design Weeks 4-6
  - Frontend - Pranav
  - Algorithm - Julian
  - Backend - Carlos
- Implementation Weeks 5-9
- Testing Week 5-10
  - Browser Capability
  - Algorithm Correctness
  - Ease of Use

Thank you

Questions?