Baylor Computer Science Capstone Course Problem Statement

Spring 2020

# Background

## Capstone Content Management System v2.0

Capstone programs offer opportunities for students to demonstrate the culmination of their learnings in an education institution or environment. The entire spectrum of departments, programs, degrees, etc. across universities have the ability to design a capstone program which produces some sort of tangible asset or intellectual property. For example, art degrees culminate in a final major piece of work; life science degrees have some sort of research, engineering is some sort of research or product, computer science is software, literature is books/novels/etc., and on and on.

Many times, universities and other organizations have limited marketing material and information rolled into their existing organization wide Content Management Systems (CMS) on a single or few pages that are buried beneath the department’s own pages. The information is not centralized, and the features are lacking in depth that could be possible for a dedicated platform for hosting, managing, and showcasing the capstone programs.

Types of common problems and questions all capstone programs face:

* Handling the multiple school/department capstones
* Handling multi-disciplinary capstones
* Showcase featurettes for the results for each capstone program/team
* Centralized “Showcase Day” or equivalent where all capstone programs get to host booths to showcase their work and allow companies, students, others to peruse the materials and talk in
  + Marketing of events and “Showcase Day”
  + Handling of judges
  + Handling of awards
  + Handling of what students are tied to what programs/etc.
* Social media integration and sharing of events and capstone featurettes
* Sponsorship:
  + Why should I sponsor?
  + Who has done it in the past?
  + How to get involved?
  + Project idea/need submission
  + How much does it cost?
  + What do I get from it?
* News/blog post updates
* Ability for content management editing and configuration of pages, posts, lists and other content
* Role-level security that are tied to roles analogous to stakeholders in the education arena
* Delineation of public audience vs. internal audiences (content editors, administration, students, professors, marketing teams, etc.)

# Capstone Problem

## Context

The Cappy capstone project management system was a result of the base problem given to those before you. It was a culmination of the base problem and creativity of a team before; that base problem is outlined in Appendix A of this document.

|  |  |
| --- | --- |
| **Home Page** | **Sponsorship Page** |
| **Capstone List Page** | **Capstone Project Page** |

## Base Required Problem

V1.0 (Cappy) was an excellent start, but far from something that can be operationalized and moved into production.

Things of issue that need to be addressed:

* **Functional**
  + The name “Cappy” is not a good long-term brand name – it is too similar to another word that has a huge negative connotation. A rebranding is needed.
  + The site is built and designed like an app, but it is not an app
    - This site is meant to be a dynamic content management system more akin to a blog or information website. It needs a clean look akin to Medium.com.
    - No pull-out menu – the hamburger side menu is good for an app or a mobile version, but we need a dynamic list of nav pages that one can visit along the top – university marketing departments should be able to have configure navigation from the CMS and it dynamically shows up (Ex: bootstrap navbar).
    - Most information pages shouldn’t be hardcoded pages in the UI – there are core pages that are worth to build custom pages and components that are hardcoded in UI layer, but some pages are informational (contact us, sponsorship, about, etc.) and should be dynamic and configurable in the backend – just like creating a post on medium.com, or Wordpress, or Drupal
  + Site needs more responsive design and UI/UX enhancements
    - Need personas to be defined – think of all of them: Student, TA, Professor, University Marketing, System Admin, Potential Sponsor, Current Sponsor
      * What’s the customer journey map for each of these individuals?
    - Need to sitemap each persona’s experience and sitemap accordingly
      * Navigation is a bit clunky and should be streamlined – clean design needed for personas
    - All pages need to be reworked – evaluate usage of bootstrap HTML/CSS framework components and responsive design elements
      * Research other sites for UX ideas
      * The home page should be configurable within Strapi. It should not an advertisement for Cappy, but information about the university's capstone program
      * Categories should be shown in another fashion besides input dropdowns – if I’m a sponsor, I’ll probably care about specific departments/colleges (dynamic list) first and then specific classes? Maybe by year?
        + Research other sites for UX ideas and patterns on filtering UX patterns that are intuitive and clean
    - Spend time up front designing user interaction and flows
  + University organization – think about how students and universities are organized and how that might affect the organization of the capstone CMS frontend and backend
    - Universities are split into schools/colleges -> schools/colleges are split into departments (department of CS) -> departments have faculty and courses -> courses have teams, TA’s, professors, students -> teams have students. Capstone projects are likely tied to one or more courses (cross-disciplinary) and/or one or more teams (cross-disciplinary)
    - Cross disciplinary management – what happens if a business school capstone and CS capstone class need to collaborate? How would that work and be displayed in the system?
  + The sponsorship page is not really existing enough for me to want to take $20,000 of my budget and give it. Think of the sponsor persona and what features would speak to them? Go ask some professors in the business and CS schools on information that’s important to sponsors
    - What would the sponsor persona want to see on the project detail page? What about the potential sponsor vs. the active project sponsor?
    - Stuff I care about as a sponsor: What am I getting for my dollars?
      * What intellectual property am I getting? Deliverables? (code, documents, spreadsheets, presentations)
      * What services are adding value to my organization? (research, software development, analysis, consulting, etc.)
      * What is the timeline?
      * Access to recruiting talent and professors
      * Brand recognition
  + For sponsorship submissions, I as a department, school, marketing, admin representative, should be able to see the funnel of all requests, which ones are approved, which ones are tied to classes
    - Remember that not all features have to be implemented on the front-end UI. Some can be implemented within the Strapi portal.
  + Other schools for inspiration/research: <https://capstone.mines.edu/>, <https://capstone.gatech.edu/>
    - These are not clean designs but can be used to sample the type of functionality that should be considered
* **Technical**
  + Libraries need upgrading – the front-end UI, boostrap, the Strapi headless CMS, the DB, they all need upgrading to the latest stable releases
  + Security has not been adequately implemented nor tested
    - Should have striation of permission-to-role based security. I should be able to create an arbitrary # of roles, assign permissions to those roles, have the UI and API protected by permissions, and then assign users specific roles
    - Not HTTPS – need certificates if there’s any sort of login
  + Continuous integration – no manual deployments – use Jenkins or some other utility to take the codebase from Gitlab and deploy changes to UI/strapi/DB appropriately
  + Configurability – we should not have to recompile the applications every time we want to change a URL, server name, port, or other common configs. Externalize configs into YAML or JSON files.
  + Containerization and Kubernetes – setting up a new environment should be easy – a whole setup should be able to be bootstrapped with a deployment of services on Kubernetes with components deployed in separate docker containers (UI vs. API (strapi) vs. DB vs. other)

## Other Constraints

* Don’t re-invent the wheel – someone has likely built a library for what you’re intending. If the licensing allows for including in products, your job should just be to integrate their library.
* User stories with points, MoSCoW, sprints, tasks, hours, etc. must be stored in the team’s Gitlab project
* Source code to be checked into Git repositories stored with the Gitlab project
* Gitlab’s workflow must be followed in the Git repositories, with branches, pull requests (merge requests), and commits tied to the user story work issues
* Documentation must be built using your GitLab’s project’s Wiki using markdown

## Extra, Extra

Use your creative brains. This is your opportunity to get out of the mindset of someone telling you want to do. You have a product canvas in front of you. Offer new features, toolkits, and visualizations.

* Be Creative!
* **Do your research! Research ideas from other existing capstone programs and websites and be creative with the user experience and approach that fits the personas of your system**

## Expect Changes to Come

Since this is a model of the “real world”, we reserve the right to change or amend the requirements & scope at any time

* **NOTE**: you should expect requirement changes, as it is part of the Agile process... no waterfall here =)
* **FURTHER NOTE**: Since you will be using agile development, you should be able to accommodate any changes in scope or priority

# Appendix A: 2019 Capstone Problem: Capstone CMS v1.0 (aka Cappy)

## Base Required Problem

Each team is tasked with building a centralized, cross-organization/university platform that assists with the common problems and solutions necessary for any type of capstone program that demonstrates the following high-level requirements

**Capstone Content Management System:**

* Support for the common problems and solutions aforementioned above
* Delineation of obvious public vs. internal/authorized/secure roles and actions performed in the system
* Should have a configurable look and feel elements to match the basic style of the university or organization

## Other Constraints

* Don’t re-invent the wheel. Use a headless Content Management System like *strapi.io* or another alternative to power your backend
* The backend application must authenticate and authorize from a 3rd party system like Active Directory, LDAP -
  + You must not use the username/password store utilized by the headless CMS, nor store usernames and passwords in your application’s database – this should be federated through some other mechanism/application
  + Recommend investigating an open source Single Sign On vendor/tool like FusionAuth to broker multi-provider authentication and authorization (Active Directory, LDAP, Facebook, Twitter, OpenID, etc.)
* Components of the team project must use Docker containers to separate application components
* User stories with points, MoSCoW, sprints, tasks, hours, etc. must be stored in the team’s Gitlab project
* Source code to be checked into Git repositories stored with the Gitlab project
* Gitlab’s workflow must be followed in the Git repositories, with branches, pull requests (merge requests), and commits tied to the user story work issues
* Documentation must be built using your GitLab’s project’s Wiki using markdown