MSU Developers Group

College Blend

Software Design Documentation

Date: 2/26/2017

TABLE OF CONTENTS

SYSTEM OVERVIEW SYSTEM ARCHITECTURE

Architectural Design Decomposition Description Design Rationale

DATA DESIGN

Data Description Data Dictionary

COMPONENT DESIGN HUMAN INTERFACE DESIGN

Overview of User Interface Screen Images
Screen Objects and Actions

REQUIREMENTS MATRIX APPENDICES

1. INTRODUCTION 1.1 Purpose

This document defines the architecture and technology that MSU Developers Group will employ to fulfill the requirements specified by CollegeBlend, LLC. It is intended for members of MSU Developers Group as well as CollegeBlend.

-this document lays down the design for the CollegeBlend app, developed by Adam Lee, Garret Murphy, and Derek Noble for CollegeBlend, LLC

-intended for MSUDevelopers group and future associated developers as a guide for development, and CollegeBlend, LLC to communicate how the requirements of their software is being fulfilled

1.2 Scope

The software application is intended for Counselors, and Students involved with CollegeBlend. The application will provide an online system where CollegeBlend can implement their incentive system. This system will aid in helping counselors mentor students.

-not intended to be used by non-clients except CollegeBlend itself

<do I list what the app is supposed to do here? This seems like just a regurgitation of the requirements. Perhaps I isolate the island of requirements by identifying all of the ocean that surrounds it>

1.3 Overview

Provide an overview of this document and its organization.
<Think more about this!>

1.4 Reference Material

This section is optional.

List any documents, if any, which were used as sources of information for the test plan.

1.5 Definitions and Acronyms

This section is optional.

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

2. SYSTEM OVERVIEW

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

- This software application, will be developed using the Ruby on Rails Framework.

- Version controlled will be managed through git, a repository will be hosted on Github.
- IDE used will be Cloud9 IDE
-

  -system for students <track content to learn, earn rewards, apply for scholarships>
  -system for counselors <pull student demographics, view specific student profiles>
  -system for CollegeBlend <manage rewards, administration, content creation>

3. SYSTEM ARCHITECTURE 3.1 Architectural Design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

3 database agents: students, counselors, admins
Students gain rewards by taking quizzes (and completing other tasks)
Counselors track assigned students
Admins manage counselors, students, incentives, and incentivized content

3.2 DecompositionDescription

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an objectoriented description. For a functional description, put toplevel data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

3.3 Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

4. DATA DESIGN

Admins [highest level]
Counselors [medium level]
Students [low level]

Admin -> many to many Counselors and Students
Counselors -> have many Students
Students -> Belong to one counselor

4.1 Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

4.2 Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

5. COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

6. HUMAN INTERFACE DESIGN 6.1 Overview of User Interface

If a first-time user or a user that is not logged in enters CollegeBlend they will be directed to the main static home page. The body and contents of the home page will display …..??? This home page will also contain a navigation bar at the top margin where the user can navigate to the registration, login, and the about us pages.

When a user that is already registered and logged into CollegeBlend enters the site they will be directed either to their profile page or the user index page. If the user is an admin they will be directed to the user index page, also known as the admin dashboard. If the user is a student, they will be directed to their profile page.

Every student will have a profile page as it is setup and uses the information when they first registered with CollegeBlend. They will be able to view and edit their own profile information such as email and other information that will describe their personal preferences concerning academics when they are logged in. This view of their profile will be displayed in the body of the page.

Once a user logs in, whether they are an admin, counselor, or a student, the top margin navigation bar will still be displayed but will be modified slightly. The log in button will be switched to log out and a profile button will be added to the bar. Also, a new side navigation bar will be displayed. At the top

Top margin navigation bar

Side navigation bar

Profile page

User index (admin dasboard)

6.2 Screen Images

Display screenshots showing the interface from the user's perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

6.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

7. Requirements Matrix

Provide a crossreference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

8. Appendices

This section is optional.

Software Design Document

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

Reqs:
nonfunctional:
-secure storage and validation for personal information (of students)
-TDD
-scalability

Functional:
-track incentives earned by students
-track and manage incentives awarded by admins
-quiz administration for admins
-quiz participation by students
-store and display student info
-manage users for admins
-personalized experience for all
-student view for counselors

Currently a platform to administer CollegeBlend incentives to registered students, with assistance from counselors to ensure students are benefiting as much as possible
More stuff here

Questions:

- Heroku and AWS, are they going to comp for that
    - Lets do Heroku for now
- Sqlite3 or PostGres?
- What Each counselor should see?
    - Admin: Everything
    - Counselor:
        - Listing Students
        - Student Scores
        - Student Profiles
    - Student
        - Only their profile
- Reports are a must
- Live Quiz
- Providing articles, or external articles, or both? Which one has more priority.
    - For now CollegeBlend Long run external
    - WYSIWYG Text editor? Or uploading a pdf reading.
    - Upload or Link(priority)
- What should be displayed on the home page?
- CollegeBlend or collegeblend?

1. Counselor can see all student profile pages and but have a group of students set assinged to them.
2. Home page: Contact info, about, links, articles displayed on the application.
3. Square Space and articles
4. Student page: access to resources, scholarship searching, link to CollgeBlend square space site
5. collegeblend logo
6. Reports of information about student. For our scope, develop a report with simple info such as point value earned, and registration info.
7. Add more info about student: race, ethnicity, annual income, gpa
8. Replicate DFS (https://denverscholarship.org/) question structure (collegeblend will most likely provide a list of questions asked by counselors to students)
9. Meta data (future goal)
10. Square space student log in questions for ideas on more student info
11. Priority: Build quiz, point system, and articles on site