Software Requirements Specification

collegeblend

Authors:
Adam Lee
Garret Murphy
Derek Noble

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.   Introduction

1.1   Purpose

The purpose of this requirements specification is to establish a consensus between CollegeBlend and its development team regarding the expected features of the CollegeBlend web application. By detailing all of the intended features and functionality of the application, this document helps CollegeBlend communicate precisely what it wants the software to do.
This specification is meant to be submitted by the development team to CollegeBlend for approval; the approved version serves as a reference tool for the development team so that they may ensure to their best ability that the resulting software is compliant with CollegeBlend's desires. As this specification is a clear and comprehensive list of features, it should continue to be useful to CollegeBlend in the event the development time of the application exceeds the duration of service by the development team. By extension, the requirements defined in this document are not guaranteed to be finished within the time allotted to the development team.

1.2   Scope

CollegeBlend is a web application aimed to help aid high school counselors prepare their students for other high school classes as well as their future in college. The application will also provide the students with resources to prepare for college and to gain scholarship money through an incentive system.

Counselors will have access to student's profiles which will contain data such as: Name, email, GPA, SAT and ACT scores, subjects of interest, desired college and career paths, and quiz points received.

Students will have access to resources provided by CollegeBlend such as articles related to their attributes in their profile. For example, resources can be given to a student related to their desired career and college paths inquired from their profile. They also have the ability to use the quiz system by answering questions generated by CollegeBlend in return for points which can then be converted to scholarship money for the college they want to attend.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
Admin/Administrator	System administrator who is given specific permission for managing users and counselors as well as controlling the system.
Counselor	A user of CollegeBlend that has certain permissions for viewing students, creating content, and generating reports.
Student	A user of CollegeBlend that has restricted permissions to the CollegeBlend system. They can edit their profile, view resources given, and participate in learning activities.

1.4 Overview

The remainder of this document includes two chapters. The second chapter provides an overview of the system functionality as well as more in-depth information about the types of users and their interaction with the CollegeBlend application. Chapter three details the nonfunctional requirements not mentioned in the previous parts of this specification.

2. Overall Description

2.1 Product Perspective

CollegeBlend app is a new, self-contained web application. It is not an element of a larger system, but does integrate third-party elements to implement parts of its functionality.

2.2 Product Functions

CollegeBlend must let Students:
-browse and engage in incentivised learning activities
-accrue incentives
-track incentives earned
-maintain a profile

CollegeBlend must let Administrators:
-manage users
-track application metadata
-track and manage incentives awarded
-manage articles and learning activities
-maintain a profile

CollegeBlend must let Counselors:
-maintain a profile
-track students
-manage articles and learning activities

2.3 User Classes and Characteristics

The CollegeBlend app deals with three different user classes: students, administrators, and counselors.

Administrators are the most important user class. Users of this type are essential to the health of the application because they perform user management, incentive management, global content management, and app diagnostics. To perform user management, administrators must be able to view all other registered users, edit their permissions, and suspend the activity of malicious or expired users. Administrators must also be able to perform incentive management by updating the rewards for incentivised content, approving new incentivised content, and observing "incentives awarded" reports. With the ability to create their own content and observe and edit other users' content, these users are also capable of managing all of the content in the application. Finally, in order to properly assess the health of the app, Administrators will have access to app metadata such as site traffic and page views.

Students are the second most important user class since they are the primary focus for the app. Students engage in learning exercises and accrue rewards. The learning exercises for students will consist of filling out surveys, participating in quizzes, and applying for scholarships. Awards for students are accrued by earning points from participating in incentivized content. Students are able to view and manage their accounts, view and participate in all accessible activities, and track the amount of rewards earned.

Counselors are the third most important user class since counselor satisfaction indirectly improves student well being. Counselors perform local content management, oversee students, and review reports. Counselors will be able to add new activities for students to participate in, submit incentivized activities for approval by administrators, observe progress for an assigned list of students, and generate reports of student progress and system-wide demographics.

2.4 Operating Environment

CollegeBlend will operate with modern browsers. It is hosted on an SQLite3/postgres database, with further database management support based on future deployment needs. CollegeBlend will also remain updated for compliance with the following rails gems: Pundit, Bootstrap, Devise, CarrierWave, Wysiwyg, and Survey.

2.5 Design and Implementation Constraints

The software will be built using Ruby on Rails with the Cloud9 IDE. It will be hosted on an Sqlite3 database for testing and a postgres server for deployment. The initial design will be built using the Pundit, Bootstrap, Devise, CarrierWave, Wysiwyg, and Survey ruby gems.

2.6 User Documentation

Separate tutorials will be provided for administrators, counselors, and students. Tutorials will be in the form of step-by-step guides delivered in-browser and separated by relevant tasks.

3. Nonfunctional Requirements

3.1 Performance Requirements

-Server architecture will mitigate denial of service attacks so that users enjoy timely server requests and responses.

3.2 Safety Requirements

-The user interface will avoid unnecessary stimulation to minimize the risk of seizures.

3.3 Security Requirements

-Users of all categories will be expected to provide authentication before logging in.
-Prevent unauthorized access to sensitive user data.

3.4 Software Quality Attributes

Software Usability:
-Intuitive interface and workflow for all parts of the application.
-Accommodation for the majority of common visual constraints for users.

3.5 Business Rules

-Counselors must have administrative approval to create incentivized content.