Software Requirements Specification

for

HCGBST Tracking System Database

Version 0.1 approved

Prepared by:
Rachel Brower
Blake Culbertson
Enrique Lopez
Chan Rain
Garrett Willis

Cal Poly Humboldt

March 8-9, 2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Creation	3/8/25	Created the initial document	.01

1. Introduction

1.1 Purpose

HCGBST is a system meant for storing scholarships, as well as their applications, to streamline the application process for students and simplify the application processing for the organization. The system also acts as a database to manage and use the applications to quickly access information for applicants.

1.2 Document Conventions

The document follows the number scheme 1.1, 1.2, 1.3 to help anyone reading follow the flow of the document.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is the employees of the organization *Humboldt County Farm Bureau* and acts as the documentation for use of the software.

1.4 Product Scope

The scope of this product is to allow for the storage of scholarships offered by the *Humboldt County Farm Bureau* and act as the application portal for their provided scholarships. Additionally, the portal functions as a space to moderate the scholarships and applicants by the organization.

2. Overall Description

2.1 Product Perspective

This product is the application portal and tracker for the organization. It is to be associated with the organization itself and acts in tandem with the homepage of the website.

2.2 Product Functions

- Application Form
    - Interface for applicants to submit scholarship forms.
    - Interface for applicants to view submitted and in-progress scholarship forms.
- Student Management
    - Interface for admins to manage students within the database and their related data.
- Scholarship Management
    - Interface for admins to manage scholarships, applicants for said scholarships, scholarship history and data.

2.3 User Classes and Characteristics

The main user class for this product will be students applying for scholarships.

2.4 Operating Environment

The product will be created using HTML, CSS, JavaScript, and MongoDB.

2.5 Design and Implementation Constraints

The sole constraint of this project is the time limit imposed by the nature of the Cal Poly Humboldt Hackathon.

2.6 User Documentation

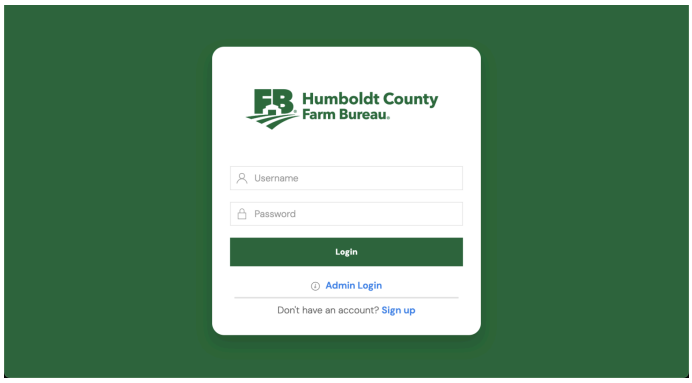
Outside documentation will be provided to allow for the upkeep of the program. This will include documentation regarding the front-end, back-end, and database. Additionally, visualization of the schema will be available

2.7 Assumptions and Dependencies

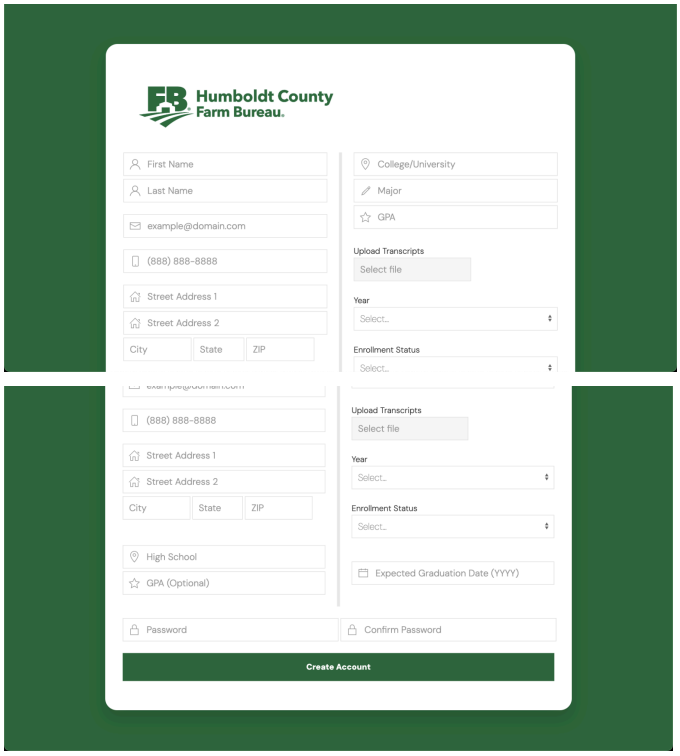
Unaware of any at this time.

3. External Interface Requirements

3.1 User Interfaces



- Example of log-in page, with admin login link, sign-up links



- Example of sign-up page

3.2 Hardware Interfaces

No hardware interfaces at this time.

3.3 Software Interfaces

There will be a connection from the front-end and the back-end of the product. The front-end will contain the search features for the applicant to access different scholarships and the backend will store and process this data. The administration users will be able to view all of the data associated with the applicants and be able to quickly process applications

3.4 Communications Interfaces

This product will require a web browser and an internet connection in order to properly function

4. Database Features

4.1 Database Structure and Schema

All applicant tables exist on their own, and if they submit an application, they are to be linked to said application. Applications are linked to individual scholarships and contain the applicant's data. The scholarships contain application fields, which link to the application table, and the application table links to an applicant field, which links to an individual applicant. The admin field is allowed to oversee and modify the fields. Admin fields may contain super fields; admin users with the ‘super’ distinction are allowed to create other ‘super’ and/or admin users. The read/write fields are linked to respective read/write permissions of the admin users and allow for an implicit differentiation between admins and reviewers without adding additional fields for reviewers, creating a more minimal approach.

4.1.1 Detailing of Schema

Applicant:

- The applicant table contains data regarding scholarship applicants.

Admin:

- The admin table contains data regarding administrative users. The administrative users may fall under: super-admin, admin, and reviewer.
    - Super-Admin: Has the ability to create other administrative users, can manage scholarships, can manage applicants
    - Admin: Has the ability to manage scholarships and has the ability to manage applicants
    - Reviewer: Has the ability to view scholarships and the ability to view applicants but does not have editing permissions

Application:

- The application field contains data regarding the scholarships and contains specific data regarding the applicants in order to log the applicants that applied to the scholarship.

Scholarship:

- The scholarship table contains data regarding the scholarships in order to properly log their existence. Some of these fields include: created date, number of applicants, and funding type.

4.2 Use of MongoDB

4.2.1 Usage of MongoDB:

The structure of MongoDB will allow for the database to be changed to easily reflect consumer requirements and expectations. MongoDB allows for easy scalability and quick access of data to allow the client to easily scale up the database if necessary.

5. Appendix

6.1 Team Roles

- **Lead Designer: Rachael Brower**
- **Technical Designer/Front-end Manager: Chan Rain**
- **Security Specialist: Blake Culbertson**
- **Quality Assurance, Documentation: Enrique Lopez**
- **Team/Backend Manager: Garret Willis**