Software Requirements Specification

for

HCGBST Tracking System

Version 0.1 approved

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

Cal Poly Humboldt

March 8-9, 2025

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

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 and 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 for 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 websites homepage

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 to scholarships

2.4 Operating Environment

The product will be created using html, css, javascript, mongodb, and react.

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

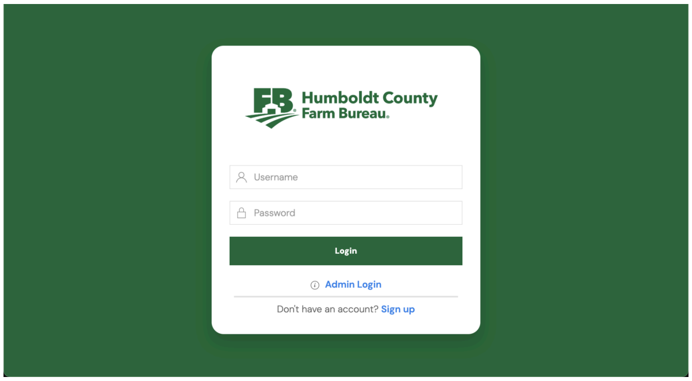
Outside documentation will be provided to allow for the upkeep of the program. This will include documentation regarding the frontend, backend, and database. Additionally visualization of the schema will be available

2.7 Assumptions and Dependencies

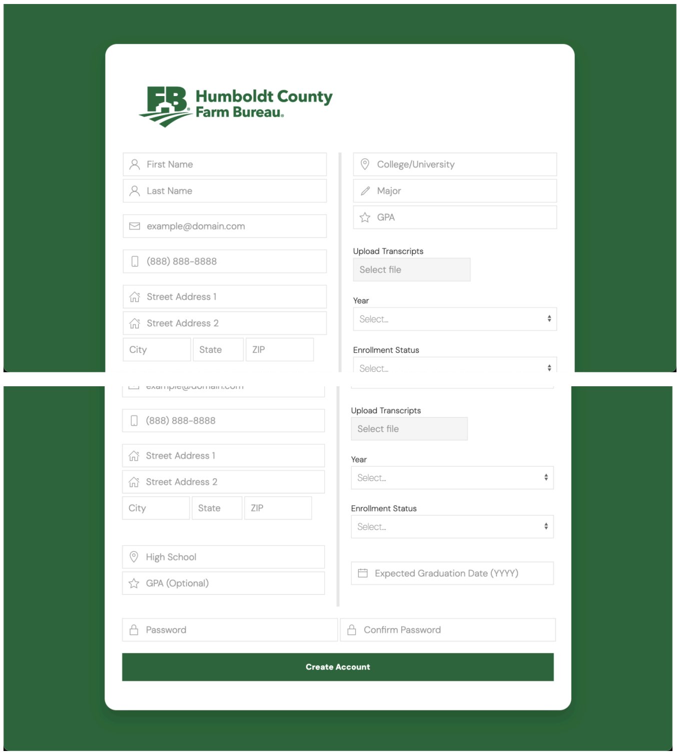
Unaware of any at this time.

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. System Features

4.1 Applicant Account Creation/Login

4.3.1 Description:

Applicant:

- Applicant is able to create and login to an account

4.3.2 Stimulus/Response Sequences:

Applicant:

- Applicant clicks login or create account
- Applicant either enters login information or information to create an account
- Applicant is then logged into that account registered to them

4.2 Displaying applicant data

4.1.1 Stimulus/Response Sequences:

Applicant:

- Applicant can view their personal information such as:
  - First Name, Last Name

Admin:

- Admin can view applicant data
- Admin can view application data
- Admin can edit applicant data
- Admin can edit application data

Reviewer:

- Reviewer can view applicant data
- Reviewer can view application data

4.1.2 Functional Requirements:

- P0: Basic Scholarship Database
  - P0.1.1: Schema for storing scholarship details (name, amount, criteria)
  - P0.1.2: CRUD (Create, Read, Update, Delete) operations for managing scholarships

4.3 Managing applicant data

4.2.1 Description:

Applicant:

- Applicant can view their stored data (personal) and view their applications
- Applicant can edit their stored data (personal)

Admin

- Admin can view applicant accounts
- Admin can view applications

4.2.2 Stimulus/Response Sequences:

Applicant:

- Applicant selects “My applications” or application goes to the Profile Icon

Admin:

- Admin selects the button to show Applicant data, and is directed to a web document containing the overall Applicant information, and Application data

4.2.3 Functional Requirements:

- P0.3.1: Overview of all applicants
- P0.3.2: Overview which includes basic reporting of applications
- P0.3.3: Simple user management
- P0.3.4: Ability to print applicant data

4.4 Managing admin data

4.2.1 Description and Priority:

Admin:

- Admin can view personal information such as First Name, Last Name, Email

4.2.2 Stimulus/Response Sequences:

Applicant:

- Applicant selects “My Profile”/”My Games”(undecided) button on the web page and gets directed to a webpage containing their game statistics in the default time frame(overall)
- Applicant chooses different timeframes and the webpage changes accordingly
- Applicant can choose to show the highscores of other Applicants as well as average scores for each statistic, to compare information

4.2.3 Admin Creation:

- Super User can create Super User, Admin, Reviewer Accounts
- Super User can add related data to created Admin, Reviewer Accounts

4.2.3 Admin Management:

- Super user can delete Admin, Reviewer Accounts
- Super user can delete/modify related data to created Admin, Reviewer Accounts

4.2.3 Functional Requirements:

- REQ-1: Applicant should be able to view personal game statistics over different time frames

4.5 User Authentication

4.3.1 Description and Priority:

Applicant:

- Applicant is able to create and login to an account
- Priority is medium

4.3.2 Stimulus/Response Sequences:

Applicant:

- Applicant clicks login or create account
- Applicant either enters login information or information to create an account
- Applicant is then logged into that account registered to them

5. Other Nonfunctional Requirements

5.1: Focus on mobile-responsive design as student often apply via phones

5.2: Implement strong data validation to prevent application errors

5.3: Ensure accessibility compliance for all users

5.4: Build with security in mind (data encryption, secure file storage)

5.5: Create clear error messages and user guidance

5.6: Document setup process for future maintenance

6. Appendix

6.1 Team Roles

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