Software Requirements Specification

for

Black Bear Bulletin

Version 1.0 approved

Prepared by Xander Dufour, Cooper Stepankiw, Matthew Frizzle, Michael Ness, Patrick Storer, Braden Pare,

University of Maine

02/28/2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Cooper Stepankiw	2/28/25	Creation of SRS Template	0.0.1

Cooper Stepankiw	3/6/25	Completion of first draft of SRS	1.0.0

1. Introduction

1.1 Purpose

The main purpose of this document is to provide documentation for the *BBB* (Black Bear Bulletin) system, specifically focusing on the functionalities and features included in version 1.0.0.

1.2 Document Conventions

This *SRS* uses clear and consistent formatting, with specific fonts or highlights to emphasize important elements:

- **Headers and Sections:** Use bigger or bold fonts for titles to make sections stand out.
- **Text Emphasis:** Important points can be bold for clarity. Texts will be italicized and underlined if they are to be found in the glossary.
- **Numbering:** Number everything clearly for easy reference.
- **Formatting:** Use bullet points or lists to organize ideas.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for various types of readers, each with specific purposes:

- **Project Manager:** The SRS helps track project goals, requirements, and priorities, ensuring everything stays on schedule and within scope.
- **Designer:** The document provides clear requirements and user needs, serving as a guide to create user-friendly designs and write comprehensive user manuals or help content.
- **Developers:** Developers use the SRS to understand technical and functional requirements, while testers rely on it to create test cases and ensure the software meets all outlined expectations.
- **Users:** The SRS offers insights into the app's features and functionality, explaining how the app addresses their needs and provides value.

This SRS has these sections:

- **Introduction:** Overview of the app’s purpose and goals.
- **Description:** Explains the app's features and how it helps users.
- **Interface Requirements:** Describes the app's layout and design for easy use.
- **System Features:** Details the app's main functions, like live updates and customizable dashboards.
- **Other Requirements:** Covers technical needs, security, and legal compliance.

1.4 Product Scope

This document specifies requirements for the Black Bear Bulletin, a user-friendly application designed to centralize information about UMaine sports teams.

This application allows users to:

- View live scores, schedules, and team news.
- Customize their dashboard with widgets tailored to their preferences.
- Track team progress and player statistics.
- Access updates for all UMaine sports in a single location.
- Navigate effortlessly using a clean, simple interface.
- Choose specific sports to follow and personalize notifications.
- Stay informed about program news and updates.

The application ensures an intuitive experience for users of all ages and technical backgrounds, bringing the UMaine sports community together. It is designed to operate seamlessly on mobile devices, allowing users to stay connected anytime, anywhere.

1.5 References

https://www.reqview.com/papers/ReqView-Example_Software_Requirements_Specification_SRS_Document.pdf

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Black Bear Bulletin is a new and self contained web application designed to centralize and streamline access to the University of Maine sports information such as live scores. It is not a follow up product or a replacement for an existing product or system, but it is an innovative system built to address the difficulties of navigating and finding all of the latest sports information in one place.

UMaine sports fans currently rely on a combination of social media, official team sites, etc to find updated information. Moreover, the information is not tailored to the individual. Black Bear Bulletin aims to allow users to customize their dashboard depending on what their favorite sport is, and what they want to see. The current sources of information lack that personalization, and require users to sift through information depending on their interests. The web application version will allow

users to create a personalized dashboard with customizable widgets to display live scores, updates, news, etc. It is meant to ensure users only receive the information they want to see, with an emphasis on accessibility and ease of use.

Black Bear Bulletin will be a standalone web application while utilizing external data sources for scores and information, such as the NCAA API, UMaine's Sports Database, and potentially additional third-party services. The system will depend on the backend to fetch and process the data to be displayed on the widgets.

[insert high level system diagram illustrating major components and interconnections here]

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

- *Customizable Sports Dashboard - Users can select and arrange widgets to display specific spots teams, information, etc based on the user's preferences*
- *Live Score Updates - Real-time updates on UMaine sports scores for live and completed games*
- *Team News - News displayed related to UMaine athletics*
- *Event Reminders - Displays upcoming games and allow users to set notifications for when the game time is*
- *User Profile and Settings - Enables users to save their favorite teams and their dashboard settings*
- *Simple Navigation - Clean widget based design that is easy to understand, navigable, and customizable*

[insert top level system diagram or data flow representation here]

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

Black Bear Bulletin is designed for a diverse audience of sports enthusiasts with the following identified user classes:

- *General Fans - Primary*
    - *Students, Alumni, Faculty, Family Members, Community Members*
    - *Usage Frequency: Occasional-Frequent*
    - *Technical Expertise: Entry-level, basic experience required*
    - *Key Features used: Live scores, team updates, news, schedules*

    - *Importance: High (this is the primary audience)*
- *Dedicated/Enthusiastic Sports Fans/Followers*
- *Other Student Athletes/Team Members*
- *Parents, Graduates, Older Users*
- *Sports Media/Journalists*
- *UMaine Athletics Department*

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

The software will be a web-based application that will operate on Windows OS, MacOS, and ChromeOS, as long as there is access to a web browser and the internet.

Supported Platforms:

- *Operating Systems: Windows OS, macOS, ChromeOS, Linux, Android*
- *Web Browsers: Google Chrome, Firefox, Safari, Microsoft Edge*
- *Devices: Desktops, laptops, tablets, and smartphones with web browser*

Since the application is web-based, no installation is required, and updates will be delivered seamlessly via the server.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

Many constraints will influence the development and implementation of Black Bear Bulletin:

Technical Constraints:

- *Data Access: The application will depend on pulling sports data which could be difficult without access to an open API or if UMaine Athletics Department has limits on accessibility*
- *Performance Requirements: If scores are updated in real-time, performance considerations are necessary to prevent latency issues*
- *Cross-Platform Compatibility: The web app must work across different screen sizes on web (i.e. Chrome on iPhone)*

Security Constraints:

- *Data Protection*
-

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

Available user documentation along with the release of Black Bear Bulletin

- *GitHub: Extensive update and work history stored on the projects GitHub*
- *Project Description: In depth description of the project and the general workflow*

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

Third party components:

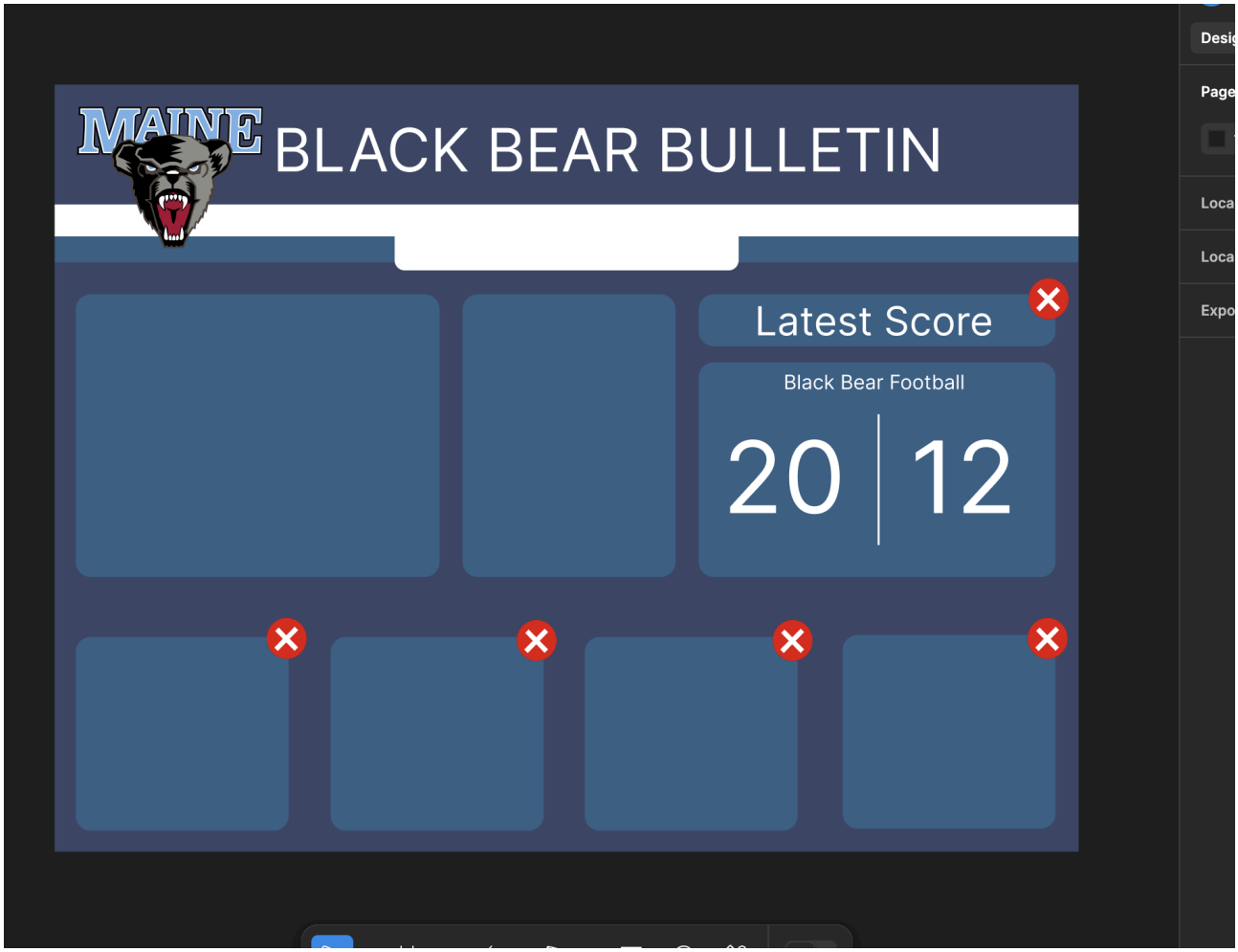
- *Umaine athletic websites(GoBlackBears.com): Scores may need to be ripped from existing Umaine websites*
- *Ncaa Scoring: Ncaa has extensive scoring tracking for many of Umaine sports that will be displayed on the website*

Potential issues:

- *Restricted API's: Ncaa and Umaine do not have open api's to access which could cause issues in gathering the data*
- *Live scoring: As games go on, the data from the third parties will change and need to be updated*

3. External Interface Requirements

3.1 User Interfaces



This is what the mockup of the website will look like. It will have interactive widgets where you can select, move and create widgets for different teams.

Within this UI, we will utilize the similar colors of blue that are involved with Umaine, so that way it looks more “Umaine-esk.” The fonts will be large enough so that they are eye-catching and you will not need to glare at the screen, but not too large where they are just obnoxious. We will also implement that when you click on a widget, it will zoom into it (or enlarge it) so that it is easier to see the information on it.

There will be a “favorite” button that will allow the user to favorite their sports teams that they would prefer to see, so that way they can have easier access to them rather than having to navigate through the entire list of teams. We will also include a help button that will allow users to get help if they are running into any issues.

3.2 Hardware Interfaces

This website will be supported on any device that can access the web, including but not limited to; desktops, laptops, phones, and tablets. As long as this device has over 1 GB of RAM, it will be able to successfully run this website.

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Black Bear Bulletin will integrate and work with many software components, databases, and APIs to ensure accurate score availability and working functionality. It is connected to other software components in many ways:

- *Databases*
    a. *Primary database: Firebase for real time updates, stores user preferences, etc*
- *External data API*
    a. *UMaine Sports API (if granted permission)*
    b. *NCAA Access API (https://github.com/henrygd/ncaa-api)*
- *Operating Systems and Deployment Platforms*
    a. *Client-side compatibility: Accessible via modern browsers (Chrome, Firefox, Edge) -web based*
    b. *Server-side compatibility: Hosted on Firebase (can use Django or Node.js for backend)*
    c. *Tools and Libraries: Frontend - React.js, Backend - Node.js, Data Handling: Redux*
    d. *Incoming Data: User Inputs (widget preferences, etc), API data*
    e. *Outgoing Data: (Possibly push notifications)*
    f. *Shared Data/Constraints: Real-time score updates on widgets using WebSockets or Polling depending on API availability (constraint) (WebSockets: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API, Polling: Client repeatedly sends requests to server)*

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

Communication requirements:

1) *Web-browser compatibility: Black Bear Bulletin will be a web-based application accessible via all modern internet browsers including: Chrome, Firefox, Safari, and Edge using HTTPs protocols for secure data transfer (Firebase has security features built in using HTTPS as a foundation for its system)*
2) *Network Server Communications: The Black Bear Bulletin website will communicate with Firebase Firestore Database for real time live score updates*
3) *Devices: laptop, phone, desktop with modern and compatible web browser installed*
4) *No installation required, updates pushed via server*
5) *end*

4. System Features

4.1 Displaying Umaine Sports Scores

4.1.1 Description and Priority

The score of Umaine sports will be displayed on the front page of the website. This is high priority as it is a core feature of the website

4.1.2 Stimulus/Response Sequences

- *User opens up the website*
- *The user will then be prompted to select what games to show on the front page*
- *The user will be able to move the widget of the score as needed*

4.1.3 Functional Requirements

REQ-1: The website shall track the scores of the Umaine sports
REQ-2: The website shall update the scores of the Umaine sports games live
REQ-3: The website shall prompt the user to add Umaine sports to be displayed on the front page as a widget
REQ-4: The website shall allow the user to move created widgets along a grid on the front page
REQ-5: The website shall allow the user to remove a widget
REQ-6: The website shall only collect and store the username, password, and provided name for the core functionality of the application
REQ-7: The website shall provide users with self-service capabilities to view, update, and delete their personal data, ensuring that no other users or unauthorized parties can access or modify it.

4.2 Storing Stats of Umaine Athletics

4.2.1 Description and Priority

The website will allow the user to view the stats of players in previous games, this is medium priority.

4.2.2 Stimulus/Response Sequences

- *User opens the website*
- *User selects a Umaine sports team*
- *Website displays recent game score and player statistics*

4.2.3 Functional Requirements

REQ-8: The website shall store the statistics of individual players and the overall team statistics

REQ-9: The website shall display the statistics of players respective to their sport
REQ-10: The website shall display the overall statistics of each Umaine sports team when selected

4.3 Viewing News of Umaine Athletics

4.3.1 Description and Priority

The website will allow the user to more easily view the news stories posted for Umaine sports. This is low priority

4.3.2 Stimulus/Response Sequences

- *User opens the website*
- *User selects a Umaine sports team*
- *Website displays most recent story*
- *User can hit see more to see all recent stories*
- *User can select a story and be redirected to the article*

4.3.3 Functional Requirements

REQ-11: The website shall store the links to recent news stories for each umaine sports team
REQ-12: The website shall display the most recent news story on a teams home page
REQ-13: The website shall display all recent news stories
REQ-14: The stories shall redirect the user to the article on it's external website

4.4 Umaine Athletics Schedule

4.2.2 Description and Priority

The website will allow the user to view the schedule for the Umaine athletics. This is medium priority

4.2.2 Stimulus/Response Sequences

- *User opens the website*
- *User selects a Umaine sports team*
- *Website displays the sports teams home page*
- *User selects the schedule*
- *Website displays the teams schedule starting on the current day*

4.2.3 Functional Requirements

REQ-15: The website shall store the schedule of each Umaine sports team
REQ-16: The website shall display the schedule of each Umaine sports team, starting from the current date
REQ-17: The website shall display current record and score of each completed game on the schedule

5. Other Nonfunctional Requirements(10-15 Required)

5.1 Performance Requirements

NFREQ-1: The website shall respond to user interactions within 2 seconds for 95% of requests under normal load.
NFREQ-2: The homepage and key pages shall fully load in under 3 seconds for 95% of users on a 5 Mbps connection.

NFREQ-3: Real time score updates shall be pushed to the UI within 2 minutes of the event for 99% of updatesThis website should respond to user actions within 2 seconds under normal circumstances.

NFREQ-4: Data consistency between frontend and backend shall be achieved within 5 seconds for 98% of operations

5.2 Safety Requirements

NFREQ-5: The system shall automatically back up users log in information and dashboards at defined intervals, store backups securely, and ensure they can be restored when needed.

NFREQ-6: The application shall restrict scheduled maintenance windows to periods with historically low traffic based on the prior 90-day average.

5.3 Security Requirements

NFREQ-7: The system shall maintain detailed audit logs of all privileged user activities, including timestamps, user IDs, and actions performed, with logs retained for at least 90 days and protected from tampering.

NFREQ-8: User sessions shall automatically terminate after 15 minutes of inactivity, and session tokens shall be securely generated, stored, and invalidated on logout.

5.4 Software Quality Attributes

NFREQ-9: The website shall have an uptime of 99.99% over every week, never being down for greater than 5 minutes at a time.

NFREQ-10: The website shall support Windows, macOS and Linux versions within the last 5 years.

NFREQ-11: The website shall allow users to customize at least 50% of their interface elements.

NFREQ-12: The website shall support up to 10,000 concurrent users without response time being greater than 2 seconds and memory utilization being greater than 90% during peak load.

NFREQ-13: The website shall manage at least 90% of all unexpected inputs and errors without failure.

5.5 Business Rules

Admin Users:

NFREQ-16: Admin users will have the ability to view, edit, approve, and publish to the website
- Manage user accounts and permissions
- Update and monitor content
- Configure application settings and features
- Access analytics for the website

End Users:

NFREQ-17: End users will have the ability to view the website
- View live scores, schedules, news, and statistics
- Customize their dashboard with widgets
- Set notifications and reminders for events
- Update their profile

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

BBB - Black Bear Bulletin
SRS - Software Requirements Specification
MFA - Multi-Factor Authentication

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>