TCU Sports Broadcasting

FrogCrew
Software Requirements Specification

Version 1.2

Revision History

Date	Version	Description	Author
03/03/2025	1.0	Initial Draft	Michala Rogers
03/10/2025	1.1	Created additional diagrams needed	Michala Rogers
03/12/2025	1.2	Round 1 of edits	Michala Rogers

Table of Contents

Software Requirements Specification

1. Introduction

The FrogCrew Software Requirements Specification (SRS) outlines the functional and non-functional requirements of the system. This document serves as a detailed reference for developers, project stakeholders, and the TCU IT team, ensuring a clear understanding of the system's capabilities, constraints, and expectations.

The SRS provides a structured breakdown of FrogCrew's purpose, operating environment, design constraints, functional requirements, external interfaces, quality attributes, and deployment strategy. It serves as a foundation for development, implementation, and future maintenance, ensuring that the system aligns with the needs of administrators, crew members, and event organizers.

This document includes references to supporting materials and follows a structured format to maintain clarity and consistency. Each section systematically describes the system's architecture, interactions, and technical specifications. The goal is to facilitate the seamless development, deployment, and long-term maintenance of FrogCrew as it transitions to the management of the TCU IT team.

1.1 The Purpose of FrogCrew

The FrogCrew system is built to solve scheduling problems for TCU Sports Broadcasting. Currently, the scheduling process is manual, time-consuming, and error-prone. This leads to scheduling conflicts, miscommunications, and a higher administrative workload. Mike Martin, an Associate Professor of Professional Practice at Texas Christian University, manages these schedules.

FrogCrew's main goal is to automate and simplify this process. Crew members can enter their availability directly into the system. Administrators can then generate schedules and assignments more efficiently. By centralizing communication and automating data handling, FrogCrew reduces errors, saves time, and boosts efficiency. The system is meant for TCU Sports Broadcasting administrators and crew members to ensure accurate and efficient staffing for all athletic events.

1.2 The Purpose of this Document

The purpose of this document is to fully describe the functional and nonfunctional requirements for the FrogCrew system. This document outlines what the system must do, not how it will be implemented, providing a comprehensive description of the requirements necessary to build the software. It includes specifications for the system's behavior, performance, security, and usability, as well as any design constraints that must be adhered to. In addition; this document is intended to be used by the development team responsible for building the FrogCrew system, TCU IT staff who will oversee the infrastructure and then take over the project after the development team's graduation, and other parties of interest (ie. TCU Sports Broadcasting administrators). All requirements specified here are committed for the initial release of FrogCrew unless otherwise noted.

1.3 Document Conventions

This document follows a structured format to ensure clarity and consistency. Each section is numbered according to its hierarchy within the document. Requirement identifiers follow a structured format with a two-letter prefix that corresponds to the section (e.g., SEC-1 for security requirements, PER-1 for performance requirements).

All headings and subheadings use a bold format to enhance readability. Key terms and system names such as FrogCrew are capitalized consistently throughout the document. When referring to external documents or references, hyperlinks are provided where applicable.

1.4 References

- Project Glossary: 1 Project Glossary
- Vision and Scope: 2 Vision and Scope
- Software Architecture: URL
- Use Cases: 3 Use Cases
- Business Rules: 4 Business Rules
- User Interface Wireframe/Prototypes: Figma FrogCrew Prototypes
- Business Domain Model: 6 Diagram Master List
- API Document: API Documentation

2. Project Glossary

The project glossary is available here: 1 Project Glossary

3. Vision and Scope

The vision and scope document is available here: 2 Vision and Scope

4. Software Architecture

The FrogCrew system is designed to streamline the scheduling and management of crew members for TCU athletic events. Its architecture is built around a modular and scalable approach, using a centralized backend with distinct layers for user interface, business logic, and data management. The system's design ensures seamless integration with TCU's existing infrastructure, including Single Sign-On (SSO) and email systems, to provide a secure and efficient user experience. This architecture supports the key functionalities of FrogCrew, such as schedule creation, availability management, shift exchanges, and financial reporting, while ensuring data security, performance, and maintainability.

4.1 System Context Diagram

Reference Diagram 2.1: 6 Diagram Master List

The Level 1: Context Diagram for the FrogCrew system provides a high-level overview of its interactions with users and external systems. The FrogCrew system is designed to streamline crew scheduling for TCU Athletics Broadcasting, facilitating efficient management of scheduling, shift swaps, and notifications. The system interacts with Administrators, Crew Members, and Gmail for automated communication. Administrators, such as the TCU Athletics Broadcasting Director, use FrogCrew to post schedules, assign crews, and approve shift swaps, ensuring proper staffing for athletic events. Crew Members, including TCU students and freelancers, submit availability, view schedules, and manage shift exchanges. FrogCrew integrates with Gmail to send automated email notifications, keeping all users informed of schedule updates and changes. By centralizing scheduling and automating notifications, FrogCrew enhances efficiency, reduces scheduling conflicts, and streamlines communication. Further technical details on system interactions are covered in the External Interfaces section.

4.2 Container Diagram

Reference Diagram 2.2: 6 Diagram Master List

The Level 2: The FrogCrew Container Diagram outlines its internal architecture and interactions. The system consists of a **Vue.js front-end**, a **C#/.NET REST API back-end**, an **MS SQL Database**, and **Gmail integration** for notifications. The Vue.js front-end provides an interactive web interface for crew members and administrators to submit availability, view schedules, manage shift swaps, and receive notifications. This front-end communicates with the REST API, which is built using C# and .NET, to process user requests, enforce scheduling rules, and manage authentication. The MS SQL Database securely stores crew profiles, game schedules, shift assignments, and financial data, with all CRUD (Create, Read, Update, Delete) operations managed by the REST API. Additionally, FrogCrew integrates with Gmail via SMTP, automating email notifications for schedule updates, shift assignments, and approvals, ensuring seamless communication between crew members and administrators. Here are the detailed steps:

1. User Accesses FrogCrew: Crew members and administrators access the FrogCrew system via a web browser, where the Vue.js front-end serves as the primary user interface.
2. Front-End Requests Data from Backend: The Vue.js front-end communicates with the C#/.NET REST API by making requests to retrieve or update data related to crew schedules, availability, and shift assignments.
3. REST API Processes Requests: The REST API handles user authentication, scheduling logic, and data processing, ensuring that all business rules, such as availability validation and shift conflict resolution, are enforced.
4. Database Operations: The REST API interacts with the MS SQL Database to store and retrieve crew member profiles, availability submissions, game schedules, shift assignments, and financial reports.
5. Automated Email Notifications: The REST API integrates with Gmail via SMTP to send email notifications to crew members and administrators about schedule updates, shift assignments, and swap approvals.
6. User Receives Notifications and Updates: Crew members and administrators receive email notifications and can log into the system to confirm assignments, adjust availability, or take necessary actions.

This sequence ensures efficient crew scheduling, accurate record-keeping, and seamless communication between users, minimizing manual intervention.

4.3 Operating Environment

OE-1: The FrogCrew system shall function correctly on modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari on both desktop and mobile devices.

OE-2: The frontend is a Vue.js-based Single Page Application (SPA) that operates in the browser and communicates with the backend via API calls.

OE-3: The backend REST API, built with C# and .NET, shall run on a Windows Server environment, utilizing IIS (Internet Information Services) as the web server.

OE-4: The database is hosted on Microsoft SQL Server (MS SQL) and operates within the same secured network as the backend application.

OE-5: The system shall be accessible by TCU Athletics staff and freelance crew members from any location with an Internet connection, provided they have valid login credentials.

OE-6: Email notifications will be handled via Gmail SMTP integration, ensuring automated communication for schedule updates and shift swaps.

OE-7: The FrogCrew application shall coexist with other TCU Athletics infrastructure and services without interfering with existing scheduling and communication systems.

OE-8: If hosted on cloud infrastructure, the system must comply with TCU's IT security policies and use firewall protections and encrypted communication (TLS/SSL) for all data exchanges.

OE-9: The system should support mobile and desktop accessibility, ensuring users can manage schedules from a variety of devices, including Windows, macOS, Android, and iOS.

4.4 Design and Implementation Constraints

CO-1: The FrogCrew system shall adhere to TCU Athletics' IT security and data privacy policies, ensuring compliance with institutional guidelines for user authentication and data protection.

CO-2: The frontend must be developed using Vue.js, as it is the selected framework for the project.

CO-3: The backend shall be implemented using C# and .NET, following standard coding practices and best practices for API development.

CO-4: The database must use Microsoft SQL Server (MS SQL) as the primary data storage solution.

CO-5: The system must integrate with Gmail's SMTP service for automated email notifications, limiting email handling to this platform.

CO-6: Authentication and user management must align with TCU's existing authentication protocols and, if required, integrate with TCU's single sign-on (SSO) system.

CO-7: All API endpoints must use RESTful principles, ensuring scalability and easy integration with potential future applications.

CO-8: The system must support responsive design to ensure usability across different devices, including desktops, tablets, and smartphones.

CO-9: All source code must follow standard software development best practices, including version control (Git), proper documentation, and adherence to code review processes.

CO-10: The hosting environment must be compatible with Windows Server and IIS, ensuring smooth deployment within TCU's infrastructure.

CO-11: Any third-party libraries or dependencies used in development must be actively maintained and compatible with the system's core technologies.

CO-12: Performance constraints require that user requests (such as schedule retrieval or availability updates) be processed within two seconds under normal system load.

CO-13: The system must be built with scalability in mind, ensuring it can handle an increasing number of users without significant performance degradation.

CO-14: Security measures, including SSL/TLS encryption, must be implemented for all data transmissions between the frontend, backend, and database to protect user information.

CO-15: Potential future limitation is the system's ability to handle database storage limitations, such as implementing data archiving or automated deletion of outdated records, to prevent performance issues when nearing maximum capacity.

4.5 Assumptions and Dependencies

Assumptions:

AS-1: The FrogCrew system assumes that all users (crew members and admins) have access to a stable internet connection to interact with the system effectively.

AS-2: It is assumed that TCU Athletics will provide and maintain the necessary server infrastructure for hosting the application.

AS-3: Crew members are expected to submit accurate availability and schedule preferences, ensuring that the scheduling process runs smoothly.

AS-4: The system assumes that Gmail's SMTP service will remain available and operational for email notifications.

AS-5: User authentication and security measures will align with TCU's existing IT policies and infrastructure.

AS-6: The system assumes that Microsoft SQL Server (MS SQL) will be available and properly maintained for data storage.

AS-7: Admins will review and approve schedule changes promptly to maintain scheduling efficiency.

Dependencies:

DE-1: The FrogCrew system depends on Gmail's SMTP service to send automated notifications regarding schedule updates and shift swaps.

DE-2: The system relies on Microsoft SQL Server for storing user and schedule data, and any disruptions to the database service could impact system functionality.

DE-3: The backend, developed in C# and .NET, depends on proper hosting and support for Windows Server and IIS to ensure system uptime.

DE-4: The system requires Vue.js and its associated libraries for the frontend, meaning updates or changes to these libraries could affect development and functionality.

DE-5: TCU Athletics IT infrastructure must support and maintain network security measures, ensuring safe access to the system for authorized users.

DE-6: The availability and performance of the system depend on TCU's IT department providing technical support, including database backups and server maintenance.

5. Functional Requirements

The FrogCrew system must provide functionalities for managing crew schedules, submitting availability, approving shift swaps, and generating reports. Crew members should be able to update their availability, view assigned shifts, and request schedule changes, while administrators must have the ability to create schedules, assign crew, and approve modifications. The system should also send automated notifications via email to inform users of schedule updates and approvals. All functionalities must ensure seamless interaction between the web interface, backend processing, and database management.

5.1 Use Cases

The use cases are available here: 3 Use Cases

5.2 Non-Use Case Functional Requirements

N/A - all needed information is found in the Use Case document

6. Business Rules

The business rules are available here: 4 Business Rules

7. Data Requirements

7.1 Business Domain Model

Reference Diagram 1.1: 6 Diagram Master List

7.2 Data Acquisition, Integrity, Retention, and Disposal

DI-1: The FrogCrew system shall retain crew member profiles, game schedules, shift assignments, and availability records for the duration of a user's active employment within the system.

DI-2: The system shall maintain a historical archive of past game schedules and crew assignments for a minimum of one year to support financial reporting and administrative review.

DI-3: Shift swap and availability change records shall be retained for six months to ensure auditability and dispute resolution.

DI-4: The system shall implement routine data backups to prevent data loss, ensuring that all critical scheduling and user data is securely stored.

DI-5: When a crew member is removed from the system, their active scheduling data shall be deleted, but historical records related to past assignments and payments will be preserved for reporting and compliance purposes.

DI-6: Any temporary or cached data used for scheduling calculations or notifications shall be cleared after 24 hours to maintain system performance and security.

8. External Interface Requirements

The FrogCrew system must effectively communicate with users and external software to facilitate scheduling, availability management, and notifications. The system's user interface (UI) must ensure ease of use for both crew members and administrators, while external integrations, such as email notifications, must function seamlessly to keep users informed

8.1 User Interfaces

UI-1: The FrogCrew system shall provide a web-based graphical user interface (GUI) that adheres to usability best practices, ensuring intuitive navigation for both crew members and administrators.

UI-2: The UI shall be responsive, adapting to different screen sizes for usability on desktops, tablets, and mobile devices.

UI-3: Crew members shall have a dashboard that displays their assigned shifts, availability status, and shift swap requests in an organized manner.

UI-4: Administrators shall have a control panel that provides an overview of crew assignments, shift approvals, and schedule modifications with filtering and search capabilities.

UI-5: The system shall include standard buttons for submitting availability, requesting shift swaps, and confirming assignments, ensuring consistency across all screens.

UI-6: Error messages shall be displayed in a clear, user-friendly format, specifying the issue and providing guidance on corrective actions.

UI-7: The system shall provide notifications within the UI for pending shift swap requests, schedule updates, and administrative approvals.

UI-8: The UI shall include a help section or tooltips to guide users on key functionalities related to scheduling and availability management.

8.2 Software Interfaces

SI-1: The FrogCrew system communicates with an MS SQL database to store and retrieve crew member profiles, availability, game schedules, shift swaps, and financial records

SI-2: CRUD operations are executed via the REST API to ensure consistent data management.

SI-3: The system sends automated email notifications via SMTP through Gmail to inform users of schedule updates, approvals, and other important alerts.

SI-4: Emails are triggered by system events such as schedule publishing, shift swap approvals, and availability updates.

SI-5: The frontend communicates with the REST API via HTTP requests to retrieve and update scheduling data, user profiles, and notifications.

SI-6: JSON is used as the standard data format for communication between the frontend and backend.

SI-7: The system may integrate with an authentication service to manage user login, access control, and session handling.

SI-8: Role-based access control (RBAC) is enforced to ensure that only authorized users can perform administrative actions.

8.3 API Document

The API document is available here: API Documentation

8.4 Hardware Interfaces

No hardware interfaces have been identified for the FrogCrew system. The system operates as a web-based application, accessible via standard web browsers, and does not require direct interaction with specific hardware devices beyond typical user devices such as desktops, laptops, tablets, and smartphones.

8.5 Communications Interfaces

CI-1: The FrogCrew system shall send automated email notifications via Gmail SMTP to inform users of schedule updates, shift assignments, shift swap approvals, and other important events.

CI-2: Emails shall be formatted in plain text or HTML and will not include attachments.

CI-3: The system shall be accessible through modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

CI-4: All communication between the frontend (Vue.js) and backend (REST API) shall use HTTPS to ensure secure data transmission.

CI-5: The frontend shall communicate with the backend using JSON-formatted HTTP requests and responses.

CI-6: API requests shall follow RESTful conventions, with appropriate authentication and authorization mechanisms in place.

CI-7: The system shall enforce secure authentication using role-based access control (RBAC) to ensure only authorized users can access certain functionalities.

CI-8: Sensitive data transmissions shall be encrypted using TLS (Transport Layer Security) to protect user information.

9. Quality Attributes

The FrogCrew system is designed to ensure high-quality performance, usability, and reliability for both crew members and administrators. These quality attributes focus on making the system efficient, easy to use, and accessible while maintaining data integrity and seamless communication.

9.1 Usability

USE-1: The FrogCrew system shall provide an intuitive user interface that allows crew members and administrators to navigate scheduling, availability, and shift swaps with minimal learning effort.

USE-2: 90% of new users shall be able to complete core tasks such as submitting availability and requesting shift swaps without external assistance.

USE-3: The system shall include clear error messages and tooltips to guide users in resolving issues and understanding system functionalities.

USE-4: The UI shall be designed for accessibility, ensuring compatibility with screen readers and keyboard navigation to support users with disabilities.

USE-5: The system shall be optimized for both desktop and mobile access, ensuring responsive design and usability across different screen sizes.

9.2 Performance

PER-1: The FrogCrew system shall support up to XXX registered users, with a maximum of XXX concurrent users accessing schedules, submitting availability, or requesting shift swaps during peak usage periods.

PER-2: XXX of webpages shall load within XXX seconds over a XXX Mbps or faster internet connection to ensure a seamless user experience.

PER-3: The system shall process and display confirmation messages within an average of XXX seconds and a maximum of XXX seconds after a user submits availability, shift swap requests, or schedule modifications.

PER-4: The system shall handle schedule updates and database queries with an average response time of under XXX seconds to maintain efficiency in real-time crew management.

9.3 Security

SEC-1: The FrogCrew system shall require user authentication for all access, ensuring that only authorized users can view or modify scheduling and shift data.

SEC-2: Role-based access control (RBAC) shall be enforced to restrict administrative actions, such as approving shift swaps or modifying game schedules, to authorized personnel.

SEC-3: All sensitive data, including user credentials and financial records, shall be stored securely using industry-standard encryption methods.

SEC-4: The system shall log all access and modification actions related to scheduling and financial transactions for auditing and security monitoring purposes.
SEC-5: User sessions shall automatically expire after a defined period of inactivity to reduce unauthorized access risks.

SEC-6: All communications between the FrogCrew system and external services, such as Gmail and the database, shall be secured using encryption protocols.

9.4 Safety

SAF-1: The FrogCrew system shall ensure that only authorized administrators can modify game schedules and shift assignments to prevent unintended disruptions.

SAF-2: The system shall prevent accidental deletion of crew schedules and financial records by requiring confirmation before executing such actions.

SAF-3: The system shall log all administrative changes related to scheduling, shift swaps, and financial data for audit and recovery purposes.

SAF-4: Error handling mechanisms shall be in place to prevent system crashes or data corruption in case of unexpected failures,

9.5 Availability

AVL-1: The FrogCrew system shall maintain high availability to ensure crew members and administrators can access scheduling and shift management features without disruption.

AVL-2: System uptime shall be maximized, with scheduled maintenance periods communicated in advance to minimize operational impact.

AVL-3: Automated monitoring and alerts shall be implemented to detect and respond to downtime issues promptly.

9.6 Robustness

ROB-1: The system shall handle unexpected errors gracefully, preventing data loss or corruption in case of connectivity issues.

ROB-2: If a session is interrupted, users shall be able to resume scheduling actions without losing previously entered data.

ROB-3: The system shall implement failover mechanisms to maintain operational stability during peak usage periods.

10. Deployment

The FrogCrew system will be deployed on a TCU owned and operated Windows Server, ensuring seamless integration with the university's existing infrastructure. The deployment process is being carefully coordinated with the TCU IT team to ensure a smooth transition and long-term sustainability. This includes aligning the system with TCU's security policies, server configurations, and maintenance schedules. Deployment will follow a structured approach, beginning with a development environment where new features and updates will be tested before moving to a staging environment for final validation. Once verified, the system will be deployed in the production environment, ensuring minimal disruption to users.

Throughout the development process, the FrogCrew team is actively collaborating with TCU IT to ensure that deployment meets university standards for security, performance, and availability. The team is working closely with IT staff to configure the server environment, database access, authentication systems, and network settings required for a stable and scalable deployment. Additionally, documentation and training resources are being prepared to facilitate a smooth handoff to the TCU IT department, which will assume responsibility for system maintenance and support following the graduation of the original senior design team.

To maintain system reliability, TCU IT will handle regular updates, security patches, and server monitoring, following university-wide IT protocols. Any necessary upgrades, bug fixes, or feature enhancements will be evaluated and implemented in coordination with relevant stakeholders. This structured deployment approach ensures that FrogCrew remains a robust, secure, and efficient platform for crew management at TCU, even after the original development team transitions out.

11. Internationalization and Localization Requirements

The FrogCrew system is designed specifically for TCU and its internal operations, meaning internationalization and localization requirements are minimal. The system will adhere to U.S. standards for date and time formats (MM/DD/YYYY), currency (USD), phone number formatting (XXX-XXX-XXXX), and English language spelling and grammar. Since FrogCrew is intended for use by TCU staff and crew members, no multilingual support or adjustments for different regional conventions are required at this time.

However, the system is built with scalability in mind, allowing for potential future localization if needed. Should TCU expand its operations or require support for additional time zones or language preferences, the system architecture could accommodate those changes with minimal adjustments. For now, FrogCrew remains focused on meeting the needs of TCU's local user base, ensuring consistency with the university's existing IT and communication standards.

12. Appendix A