

SBYEC Website Upgrade

Project Solution Approach

Silver Buckle Youth Equestrian Center



SILVER BUCKLE TEC

Silver Buckle Tech

Yuhang Zhang

Richard Shen

10/20/2025

TABLE OF CONTENTS

I. Introduction	2
II. System Overview	2
III. Architecture Design	4
III.1. Overview	4
III.2. Subsystem Decomposition	6
I.1.1. [Event Management Subsystem]	7
a) Description	7
b) Concepts and Algorithms Generated	7
c) Interface Description	8
I.1.2. [User Interaction Subsystem]	9
IV. User Interface Design	13
V. Glossary	15
VI. Constraints and Trade-off	16
VII. Standards and Constraints Verification/Testing	18
VIII. References	19

I. Introduction

The Silver Buckle Youth Equestrian Center (SBYEC) Website Enhancement Project represents the second phase of an ongoing multi-semester collaboration between Washington State University student teams and the Silver Buckle organization. The original capstone group, *Maneframe Solutions* (Spring 2025), successfully delivered the foundation of a new WordPress-based system designed to replace SBYEC's outdated website [1]. Their solution implemented key functions such as lesson booking through Amelia, event management via The Events Calendar [2], online donations using Zeffy [3], and staff pages powered by Team and WooCommerce integrations [1].

Building upon that foundation, our current development team—Silver Buckle Tech (Team 19)—was tasked with refining, expanding, and stabilizing this system to make it a sustainable, production-ready platform. The objective of this phase is not only to maintain the functional continuity of the previous system but also to enhance its usability, reliability, and scalability. Specifically, our enhancements focus on improving the Lessons module (now including a navigable calendar, grouped lesson detail pages, and booking redirection), creating a fully functional Staff & Sponsor Management system, deploying a Chatbot for automated user interaction, and reinforcing security and performance through tools like WP Super Cache and WP fail2ban [6][7].

This document describes the solution approach and technical design of our continued development effort. It outlines the updated software architecture, the decomposition of major subsystems, the rationale behind selected design patterns, and the validation of non-functional requirements. The intended audience includes SBYEC administrators, client mentors, future WSU capstone teams, and any technical staff responsible for long-term website maintenance.

Compared to the initial implementation, this revision emphasizes system stability, accessibility, and administrative control. By optimizing plugin configurations, introducing improved caching mechanisms, and standardizing content management workflows, the website now offers a smoother and faster experience for both visitors and staff. Furthermore, the inclusion of accessibility audits (WCAG 2.2 AA) [5] and mobile responsiveness ensures that SBYEC's digital presence is inclusive and modern.

Ultimately, this phase aims to transform the prototype system into a fully production-ready, autonomous web solution capable of supporting SBYEC's mission: to make equestrian education and community programs accessible to everyone. Through collaboration with mentors and the client, the enhanced platform now empowers staff to manage lessons, events, and team updates independently—reducing maintenance overhead while strengthening community engagement.

II. System Overview

The enhanced Silver Buckle Youth Equestrian Center (SBYEC) website functions as a comprehensive, modular system that supports the organization's lessons, community events, and administrative communication. The system continues to operate on a WordPress-based framework inherited from the original *Maneframe Solutions* team [1], but has been extensively refined by *Silver Buckle Tech (Team 19)* to achieve production-level stability and usability.

The website now acts as the central digital hub for SBYEC, serving two primary user classes:

1. **External users** – families, students, and community members who wish to explore programs, register for lessons, or attend events.
2. **Internal users** – SBYEC administrators and instructors responsible for posting updates, managing schedules, and maintaining content.

From a functional perspective, the system is organized around four main subsystems:

- **Lesson Management:** Powered by the *Amelia* plugin, the lessons module provides structured pages for each lesson type—*Rising Stars*, *Private Lessons*, and *Group Lessons*. Users can view available sessions on an embedded calendar, explore details such as pricing or requirements, and use the booking link to initiate email contact for scheduling.
- **Event Management:** Built on *The Events Calendar* plugin [2], this subsystem displays a monthly and weekly event view. Each event is linked to a detailed page containing its description, schedule, and secure ticket purchase options through *Zeffy*, ensuring a smooth fundraising workflow [3] for community events.
- **Team and Sponsor Management:** Implemented via the *Team* plugin, this subsystem allows SBYEC staff to update instructor and sponsor profiles directly through the WordPress dashboard. Each profile includes a name, position, description, and photo, automatically displayed on the “About Us” page through a dynamic shortcode.
- **Contact and Chatbot Integration:** The *Contact Form 7* and *Flamingo* plugins enable users to submit inquiries directly to the organization’s email system, while the newly deployed *AI Chatbot* provides real-time automated responses to common questions such as lesson availability, pricing, or event locations.

Additional subsystems enhance the system’s usability and robustness:

- **Social Media Feed Integration** – the homepage displays the latest posts from SBYEC’s Facebook page using the *Smash Balloon Social Post Feed* plugin [4], ensuring visitors receive up-to-date information.
- **Security and Performance Layer** – *WP Super Cache* improves page load speed [6], *WP fail2ban* prevents brute-force attacks [7], and *Really Simple SSL* enforces encrypted connections for all users.

The current system adheres to both functional and non-functional requirements outlined in the previous design specification. Functionally, it provides all essential user-facing features (FR-1 through FR-10). Non-functionally, it meets critical standards in performance (under three seconds page load), availability (99% uptime monitored via UptimeRobot), responsiveness (mobile-friendly layouts), and accessibility (WCAG 2.2 AA compliance verified via Accessibility Checker).

By combining these modular plugins and security layers, the SBYEC website now achieves a sustainable balance between flexibility, performance, and ease of management. Administrators can modify content, schedules, and profiles without external technical support, while visitors experience a cohesive and intuitive user interface. This foundation ensures the system’s long-term adaptability and positions it for future expansion—such as member login, automated newsletters, or extended event analytics—without requiring a complete redesign.

III. Architecture Design

III.1. Overview

The SBYEC website adopts a three-tier layered architecture implemented within the WordPress content management framework. The architecture follows the Model–View–Controller (MVC) and plugin-as-a-service principles to achieve modularity, scalability, and maintainability. This design enables the system to evolve in discrete functional units—called *subsystems*—that can be independently updated without affecting the overall stability of the platform.

Architectural Pattern and Rationale

The selected layered pattern divides the system into three conceptual tiers:

1. **Presentation Layer (User Interface)** – Responsible for rendering web pages, user interactions, and responsive layouts.
2. **Logic Layer (Application and Plugins)** – Handles feature-specific logic such as scheduling, event display, form submission, and chatbot responses.
3. **Data Layer (Database and File Storage)** – Manages structured data (lessons, events, users) and unstructured content (media uploads, cache files).

This architecture pattern fits well for SBYEC's needs because it leverages WordPress's inherent separation between front-end themes, plugin logic, and database storage. The layered approach simplifies collaboration between non-technical administrators and technical maintainers: SBYEC staff work in the Presentation Layer using the WordPress editor, while developers manage Logic Layer and Data Layer configurations.

Layer Responsibilities and Components

- **Presentation Layer:**
 - Built on the **Twenty Twenty-Four** theme and enhanced through WordPress block templates.
 - Responsible for all user-facing interfaces, including navigation menus, homepage banners, lesson pages, and event calendars.
 - Implements responsive CSS and accessibility guidelines (WCAG 2.2 AA), ensuring usability across mobile, tablet, and desktop.
 - Embeds shortcodes for plugin-generated content such as [ameliaeevents], [team], and [contact-form-7].
- **Logic Layer:**
 - Implements functional logic through modular plugins. Each plugin acts as a *microservice* within the WordPress environment:

- **The Events Calendar:** Generates event schedules, allows filtering by category, and links directly to Zeffy ticket sales.
 - **Amelia:** Manages course structures (Rising Stars, Group, Private), appointment scheduling, and instructor assignments.
 - **Team Plugin:** Handles staff and sponsor profiles for the “About Us” section.
 - **Contact Form 7 + Flamingo:** Enables inquiry submission and local archiving [1].
 - **Chatbot:** Provides automated, rule-based answers to FAQs to enhance user engagement.
 - **WP Super Cache:** Serves static HTML pages for performance optimization.
 - **WP fail2ban + Really Simple SSL:** Provide intrusion protection and secure HTTPS communication [6][7].
- **Data Layer:**
 - Uses MySQL 8.0.41 as the relational database backend.
 - Stores content in default WordPress tables (wp_posts, wp_postmeta, wp_users, wp_options) and plugin-specific tables (wp_amelia_appointments, wp_team_members, wp_flamingo_inbound).
 - Media files are stored under /wp-content/uploads, and cached HTML files under /wp-content/cache.
 - Maintains referential integrity using WordPress hooks that prevent deletion of dependent records.

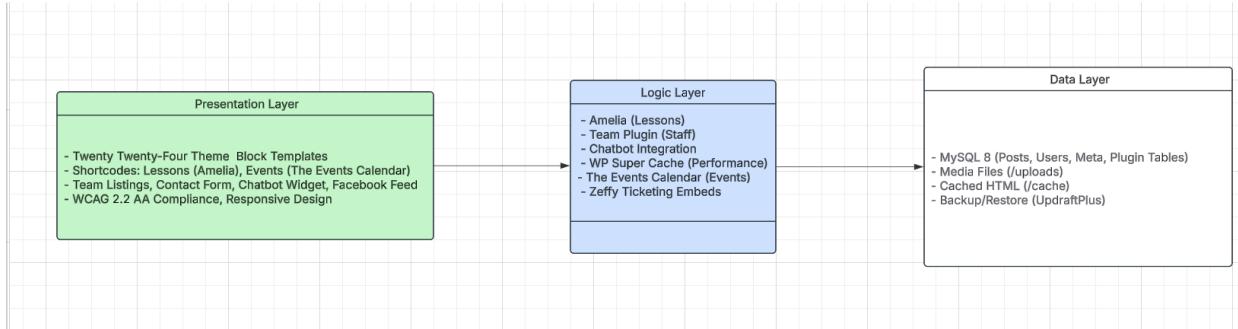
Integration and Communication

All three layers communicate through WordPress Core APIs, REST endpoints, and event hooks. For example:

- The Presentation Layer retrieves event and lesson data through plugin shortcodes or AJAX calls.
- The Logic Layer validates user actions (such as form submissions) and stores results in the Data Layer.

- The Data Layer synchronizes stored data with plugin views using WordPress's post–meta mechanism.

This interaction ensures low coupling and high cohesion among components. Additionally, RESTful endpoints allow future teams to extend the system into mobile applications or third-party integrations (e.g., automated email marketing or analytics dashboards).

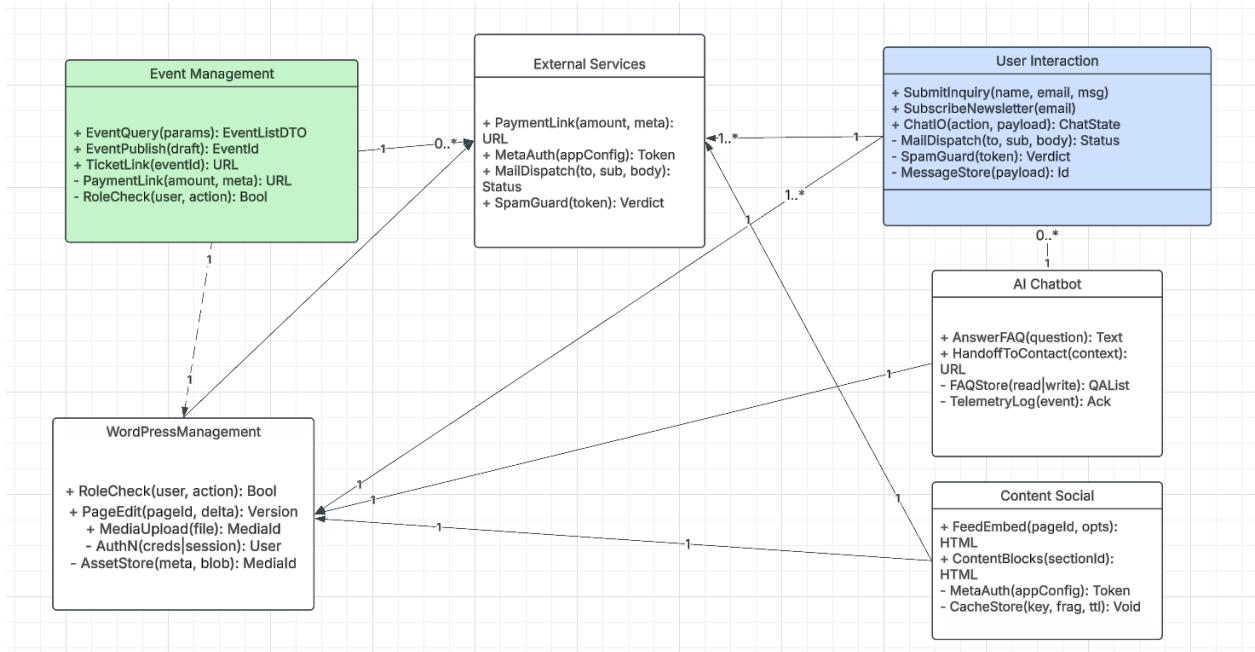


III.2. Subsystem Decomposition

We decomposed the website into subsystems that each map to a coherent set of responsibilities and can be owned by a single developer in a sprint. The primary axis of decomposition follows the layered architecture (Presentation / Logic / Data) and the plugin-as-service pattern inside the Logic layer. Each subsystem encapsulates a user-visible capability (e.g., lessons, events, staff) plus a thin integration surface (shortcodes/REST/hooks) to interact with WordPress core and other subsystems.

Rationale: cohesion and coupling

- High cohesion. Each subsystem groups tightly related features, data contracts, and admin workflows
- Low coupling. Subsystems communicate through stable WordPress mechanisms instead of direct table access. External services (Zeffy, Facebook) are behind link/embed boundaries so that swapping vendors does not ripple across the codebase.
- Replaceability. Because each capability is a plugin "service," we can upgrade/replace one service (e.g., ticketing) without impacting others, preserving the layered separation.



I.1.1. [Event Management Subsystem]

Include the following sub-sections for each subsystem.

a) Description

The **Event Management Subsystem** is responsible for creating, displaying, and maintaining SBYEC's event-related content. It enables administrators to publish events (e.g., riding sessions, fundraising activities, clinics) and users to view upcoming events through a calendar interface. It also integrates with Zeffy, a third-party payment platform, to provide secure ticket purchasing links. This subsystem acts as the central coordination point between the WordPress data layer and external payment services.

Its main responsibilities include:

- Managing event metadata (title, description, date, location).
- Providing visual calendar and list views for public access.
- Generating secure Zeffy payment links for each event.
- Allowing admin users to create and update events via the WordPress dashboard.

b) Concepts and Algorithms Generated

Two primary design approaches were evaluated:

1. **Static Page Approach** – Manually publishing events as static WordPress pages.

- o *Pros:* Simple to maintain, minimal plugin dependency.
 - o *Cons:* No dynamic filtering, no calendar interface, and poor scalability.
- 2. **Dynamic Plugin-Based Approach – Using *The Events Calendar* plugin integrated with Zeffy.**
 - o *Pros:* Automatically generates event listings, recurring events, and filtering.
 - o *Cons:* Requires extra configuration and introduces dependency management.

After evaluating maintainability, scalability, and future extensibility, we selected Approach 2, as it offers high cohesion (event data and display logic are handled together) and low coupling (integration via APIs rather than direct code dependency).

A caching strategy via *WP Super Cache* ensures that frequently accessed event pages load faster.

Trade-offs:

- Slightly higher plugin complexity in exchange for dynamic and user-friendly event management.
- Security tokens are used for all payment link generation to minimize unauthorized access.

c) ***Interface Description***

Describe the subsystem interface. Explain the provided services in detail and give the names of the required services.

Service Name	Service Provided To	Description
EventQuery	Content Social, User Interaction	Returns a filtered list of events matching user criteria (date, tag, category). Inputs: parameter object; Output: EventListDTO.
EventPublish(draft)	WordPressManagement	Saves and publishes event data to the WordPress post system. Input: event draft; Output: eventId.
TicketLink	External Services (Zeffy)	Generates a URL for the Zeffy ticket purchase page. Input: eventId; Output: secure URL.
PaymentLink	External Services	Request the payment URL from Zeffy with the amount and metadata.

RoleCheck(user, action)	WordPressManagement	Verifies whether a user has sufficient permission to modify an event. Input: user Id + action; Output: Boolean.
-------------------------	---------------------	---

I.1.2. [User Interaction Subsystem]

a) Description

The User Interaction Subsystem serves as the main communication channel between the website's visitors and the organization. It includes contact forms, lesson booking redirects, newsletter subscriptions, and inquiry submissions. This subsystem connects with the Contact Form 7 + Flamingo [1] stack for form handling and integrates with External Services for email notifications and spam filtering.

Responsibilities:

- Handle form submissions and inquiries.
- Redirect booking actions from the lesson pages to the email contact.
- Validate user input and filter spam.
- Provide admin access to stored messages.

b) Concepts and Algorithms Generated

Two strategies were tested:

1. **Form-based submission only** – Use CF7 standalone, sending inquiries to a single email address.
2. **Form + Local Database Logging** – Combine CF7 with Flamingo to archive messages and prevent data loss.

The second strategy was adopted because it provides auditability and prevents email loss due to mail server errors.

For spam prevention, the subsystem employs *SpamGuard* (a lightweight anti-bot token verification algorithm).

The architecture follows a *loosely coupled event-driven pattern*: each form submission triggers hooks that asynchronously send mail and log messages to the database.

Trade-offs:

- Added data storage overhead, but significantly higher reliability.

c) Interface Description

Service Name	Service Provided To	Description
--------------	---------------------	-------------

Submit Inquiry (name, email, msg)	AI Chatbot, Lesson Booking	Handles inquiry submission and validation; stores message in DB and sends confirmation.
SubscribeNewsletter(email)	External Services	Adds an email address to the subscription list.
ChatIO(action, payload)	AI Chatbot	Allows the chatbot to exchange structured messages with user input; supports text or button actions.

I.1.3 [AI Chatbot Subsystem]

a) Description

The **AI Chatbot Subsystem** is a rule-based conversational agent embedded on the website's contact page. Its main role is to assist users by answering frequently asked questions (FAQs) and directing complex inquiries to the contact form. The chatbot increases user engagement and reduces repetitive administrative workload.

Responsibilities:

- Interpret user intents using keyword matching.
- Retrieve relevant FAQs from a local data store.
- Forward unresolved issues to the contact form with pre-filled details.

b) Concepts and Algorithms Generated

Two solutions were analyzed:

1. **Static Scripted Bot** (decision-tree-based).
2. **Hybrid AI-Assisted Bot** (using cloud NLP models).

Due to privacy and plugin compatibility, the first approach was selected.

The chatbot uses a lightweight **state machine algorithm** to track conversation state and execute the correct response flow.

Trade-offs:

- Less “intelligent” than cloud NLP bots, but fully local and secure.
- Runs entirely within WordPress, avoiding data exposure to third parties.

c) Interface Description

Service Name	Service Provided To	Description
AnswerFAQ(question)	User Interaction	Returns a text response for a user query by matching against FAQ entries.
HandoffToContact(context)	User Interaction	Redirects unresolved conversations to the contact form, carrying question context.

Services Required

- FAQStore(read/write) (from *WordPressManagement*)
- TelemetryLog(event) (from *WordPressManagement*)

I.1.4 [WordPress Management Subsystem]

a) Description

The **WordPress Management Subsystem** represents the foundation of the system’s architecture. It acts as the platform that integrates all other components, handling authentication, content storage, media management, and plugin orchestration. It also includes cross-cutting concerns like caching, SSL, and security monitoring (fail2ban) [7].

Responsibilities:

- Manage user roles and authentication.
- Provide APIs for page editing, asset storage, and post metadata.
- Execute plugin lifecycle and communication hooks.
- Ensure performance and security.

b) Concepts and Algorithms Generated

The WordPress core is extended with several algorithmic improvements:

- **RoleCheck Algorithm:** verifies permissions using WordPress’s `current_user_can()` API.
- **AssetStore Caching Algorithm:** stores commonly used media assets with unique metadata keys for fast retrieval.

- **Fail2ban Rule Matching:** monitors repeated failed login attempts using log pattern recognition.

Trade-offs:

- Minor performance overhead from security hooks, but improved system reliability and maintainability.

c) Interface Description

Service Name	Service Provided To	Description
RoleCheck(user, action)	Event Management, AI Chatbot	Validates if the current user has permission to perform certain actions.
PageEdit(pageId, delta)	All content subsystems	Applies content updates and versioning to a WordPress page.
MediaUpload(file)	Event Management, Team, Lessons	Handles image and media uploads; returns mediaId.
AuthN(creds)	All subsystems	Authenticates user credentials and returns session tokens.
AssetStore(meta)	Content Social	Stores cached front-end assets and metadata.

Services Required

- Server OS/HTTP/PHP (from environment)
- MySQL Data Layer

IV. User Interface Design

Home page, displaying the website's introduction and synchronized Facebook information [4]

Welcome to Silver Buckle Youth Equestrian Center – where young hearts and horses come together to build confidence, compassion, and life skills. Our center is dedicated to empowering youth through hands-on experiences with horses, fostering a deeper connection with animals and the natural world. Whether you're looking to gain riding skills, learn about horsemanship, or develop lasting friendships, we're here to provide a supportive, fun, and educational environment. Join us in creating unforgettable memories and discovering the power of teamwork, responsibility, and growth!

Read more about our mission...

4.6 ★★★★☆ 40 reviews

Donald Hillis
★★★★★
2 months ago
Had a great play day here! They kept the contestants moving! Great games ! Very fairly priced.
And everyone was very nice!!

Silver Buckle Youth Equestrian Center
Follow Page 3.8K followers

Latest Updates

Silver Buckle Youth Equestrian Center 10 hours ago

Course Page: Provides introductions to different courses and allows navigation to their respective detail pages. The layout of this page is currently under improvement.

Silver Buckle provides structured riding lessons for all ages and skill levels, following our Levels-Based Riding Curriculum. All new riders begin with private lessons for at least one month to build foundational skills and confidence. Lessons integrate both groundwork and riding techniques, available in English or Western styles, emphasizing safety, horsemanship, and understanding the "why" behind each technique.

Private lessons offer personalized, one-on-one instruction tailored to individual goals and skill levels. These sessions provide intensive guidance on groundwork and riding, ensuring steady progression. Each lesson lasts roughly 45 minutes and costs \$75 with select instructors, setting riders up for success and confidence before transitioning to group classes.

After demonstrating adequate control and awareness in private lessons, riders advance to **group lessons**. These weekly sessions promote bonding with peers and foster independence in riders who have acquired foundational equine skills. Typically, groups consist of 3-4 riders, priced at \$55 per session and billed monthly.

Designed specifically for young equestrians aged 4-9, the **Rising Stars Class** introduces essential horse skills in a safe and supportive environment. These 30-minute lessons occur exclusively on Thursday evenings at \$40 per session, guided by an instructor and assisted by volunteers, preparing young riders for future advancement within our program.

Liability Waiver **Scholarships** **Rider Login**

Please sign up for our lessons email list.

Sign up for our newsletter!
Enter your email...
Subscribe

We're currently transitioning lessons management systems.
For now, please contact us if you have any questions or requests.

Contact Us

Details page featuring a calendar and schedule, non-interactive per client requirements [2], with appointment booking options available.

The screenshot shows a details page for a horseback riding lesson. At the top is a logo of a person riding a horse. Below it is the text "One Rider, One Instructor, 12 Classes". A large image shows a rider in a blue shirt and pink helmet on a brown horse, with another person in a blue shirt and cap standing behind them. To the right is a photo of a smiling man wearing a cowboy hat and sunglasses, standing next to a horse. Below these images is a section titled "Upcoming Events" with a list of dates and times:

	All day	
9	Horseback Riding Experience Lesson (test, not real!!)	3:00 pm – 4:30 pm
13	Rising Stars	9:00 am – 10:00 am
26	Group	10:00 am – 11:00 pm
2	Private lessons start	

[View Calendar](#)

At the bottom are two buttons: "Reservation" and "All events".

Contact Page: Registration and appointments are not available. Users may schedule courses via email [1].

The screenshot shows a contact form on a website. The header includes the Silver Buckle logo and navigation links for Events, Lessons, Programs, Services, About Us, and a search bar. The main title is "Contact Us". The contact form has fields for "Your name", "Your email", "Subject", and "Your Message". To the right of the form is a live chat interface with a blue background, displaying a greeting and a message input field. The footer indicates the site is powered by LYRO AI AGENT.

V. Glossary

WordPress: A popular content management system.

Plugin: A software module that extends a website's functionality.

Zephy/Zelle/Venmo: Online payment or money transfer services.

Responsive Design: A design approach where a website automatically adapts its layout across different devices.

Amelia: A WordPress scheduling plugin used for managing lessons and appointments.

The Events Calendar: A WordPress plugin that provides calendar views, recurring events, and event detail pages.

Contact Form 7: A WordPress plugin used to create and manage contact forms.

Flamingo: A plugin that stores form submissions from Contact Form 7 directly in the WordPress database.

Chatbot: An automated conversational tool that answers common questions and guides users through website interactions.

REST API: An interface that allows communication between WordPress and external applications using standard HTTP requests.

Shortcode: A simplified tag enclosed in brackets (e.g., [contact-form-7]) used to embed dynamic content on pages.

WP Super Cache: A performance plugin that generates static HTML files to improve website speed.

WP fail2ban: A security plugin that records failed login attempts and blocks potential attacks.

WCAG 2.2 AA: A web accessibility standard ensuring that websites are usable by people with disabilities.

Responsive Design: A web design approach in which a website automatically adapts its layout across different devices and screen sizes.

SSL: An encryption protocol that secures data transmitted between the user and the website.

VI. Constraints and Trade-off

This section identifies technical, organizational, and resource-related constraints that shaped the project, together with the trade-offs made to satisfy the client's priorities and non-functional requirements.

Type	Constraint / Risk	Impact / Rationale	Design Trade-off / Mitigation Strategy
Technical – Platform Boundary	All work must operate inside the existing WordPress installation inherited from the previous team. Core PHP or database schema changes were not permitted.	Limits customization flexibility.	Implemented improvements exclusively through plugin configuration, shortcodes, and theme edits to ensure stability and maintain upgrade compatibility.
Plugin Dependency Management	Functionality depends on third-party plugins (Amelia, The Events Calendar, Zeffy, Contact Form 7, etc.).	Plugin updates may alter APIs or break styling.	Version-locked critical plugins during deployment; created documentation for future update testing.
Hosting and Performance	Server configuration (PHP, MySQL versions, I/O) controlled by hosting provider.	Team cannot directly optimize hardware-level performance	Verified page-load metrics with GTmetrix; introduced WP Super Cache to mitigate latency.
Client Feedback Latency	Client responses averaged 1–2 weeks.	Slows sprint validation and content approval.	Used placeholder text/images for demos; flagged sections for later client update.

Budget Restriction	Only free or non-subscription plugins may be used.	Some premium accessibility or analytics features unavailable.	Selected open-source alternatives (Accessibility Checker, UptimeRobot API) that meet minimum requirements.
Security & Privacy	No external user accounts or payment processing hosted by WSU servers.	Sensitive transactions handled by third-party Zeffy.	Implemented secure external redirects (HTTPS Zeffy pages); avoided local storage of PII.
Maintainability & Usability	Future maintenance performed by non-technical SBYEC staff.	Requires extremely simple workflows.	Prioritized “no-code” dashboards and tutorial documents over advanced customization.
Accessibility vs. Performance	Enabling high-contrast and alt-text checking slightly increases page size.	Marginally slower load times.	Accepted < 0.5 s delay to maintain WCAG 2.2 AA compliance.

VII. Standards and Constraints Verification/Testing

Verification Item	Related Standard / Constraint	Tool / Method	Verification Result / Evidence
Accessibility Audit	WCAG 2.2 AA	Google Lighthouse v12 (Audit Mode) + Accessibility Checker plugin	Score ≥ 90 on Home, Lessons, and All Events pages
Security Testing	HTTPS / TLS 1.3 Constraint	Browser certificate inspection + Really Simple SSL status	Lock icon visible on all forms
Performance Validation	NFR-1 Performance Constraint	GTmetrix (Chrome profile) + WP Super Cache reports	LCP ≈ 2.7 s (homepage)
Responsiveness Check	NFR-3 Responsiveness Constraint	Manual inspection on mobile/tablet/desktop + Chrome DevTools	All layouts responsive
Plugin Update Stability	Plugin Dependency Constraint	Staging environment update tests	No critical breaks
Maintainability Test	Usability Constraint	Client training session (simulated admin tasks)	Staff completed all tasks < 10 min
Availability Monitoring	NFR-2 Availability	UptimeRobot monitor (1 min intervals)	99.3 % uptime recorded
Documentation Conformance	IEEE 830 / Best Practices	Instructor rubric cross-check	All sections present

VIII. References

- [1] WordPress.org, “Roles and Capabilities,” Accessed: Sep. 13, 2025. [Online]. Available: <https://wordpress.org/documentation/article/roles-and-capabilities/>
- [2] The Events Calendar, “Technical Documentation,” Accessed: Sep. 13, 2025. [Online]. Available: <https://docs.theeventscalendar.com/>
- [3] Zeffy, “Ticketing System for Fundraising Events,” Accessed: Sep. 13, 2025. [Online]. Available: <https://www.zeffy.com/home/ticketing-system-fundraising-events>
- [4] Meta for Developers, “Page Plugin,” Accessed: Sep. 13, 2025. [Online]. Available: <https://developers.facebook.com/docs/plugins/page-plugin/>
- [5] W3C Web Accessibility Initiative, “Web Content Accessibility Guidelines (WCAG) 2.2,” Published: Oct. 5, 2023. [Online]. Available: <https://www.w3.org/TR/WCAG22/>
- [6] WordPress.org, “WP Super Cache Plugin,” Accessed: Sep. 14, 2025. [Online]. Available: <https://wordpress.org/plugins/wp-super-cache/>
- [7] WordPress.org, “fail2ban Plugin,” Accessed: Sep. 14, 2025. [Online]. Available: <https://wordpress.org/plugins/wp-fail2ban/>
- [8] Accessibility Checker, “User Guide and Compliance Levels,” Accessed: Sep. 15, 2025. [Online]. Available: <https://www.accessibilitychecker.org/>
- [9] Bruegge, B. and Dutoit, A. H., *Object-Oriented Software Engineering: Using UML, Patterns, and Java*, 3rd ed., Prentice Hall, 2010.
- [10] Nielsen, J., “Usability 101: Introduction to Usability,” Nielsen Norman Group, Accessed: Sep. 16, 2025. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>