Strawhacks

Rock Chalk Rendezvous

Software Development Plan

Version 1.0

Rock Chalk Rendezvous	Version: 1.0	
Software Development Plan	Date: 02/22/2024	
proj_plan1		

Revision History

Date	Version	Description	Author
02/19/2024	0.1	First draft	Shayna Weinstein, Dylan Kneidel, Delroy Wright, Ben Phillips, Andrew Reyes
02/22/2024	1.0	Draft to turn in	Shayna Weinstein, Dylan Kneidel, Delroy Wright, Ben Phillips, Andrew Reyes

Table of Contents *[keep this; say N/A when inapplicable]*

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

Software Development Plan

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

1.   Introduction

[The introduction of the **Software Development Plan** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **Software Development Plan**.]

1.1   Purpose

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.

The following people use the *Software Development Plan*:

- The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
- **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.
- Any **instructor** may use it to keep a running tab on what is currently being developed within this project and the relative dates for which each phase of the program will be released

1.2   Scope

This *Software Development Plan* describes the overall plan to be used by the "Rock Chalk Rendezvous" project, including deployment of the product.

- The details of the individual iterations will be described in the Iteration Plans.
- The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.
- The maintenance and management of our project will be based upon the phase plan and project schedule as listed in Section 4.2.4 below.

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

1.3   Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4   References

For the **Software Development Plan**, the list of referenced artifacts includes:

- ·   Iteration Plans

- ·   Development Case

- ·   Vision [you may prepare a vision statement of your own: what your vision for the project is]

- ·   Glossary

- ·   Any other supporting plans or documentation.

[1] Project Vision, version 1.0 from 2/19/2024. Team Strawhacks, accessible via team folder.

[2] Glossary of Terms, version 0.1 from 2/19/2024. Team Strawhacks, accessible via team folder.

1.5   Overview

This *Software Development Plan* contains the following information:

Project Overview   —   provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization   —   describes the organizational structure of the project team.

Management Process   —   explains the estimated schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines   —   provide an overview of the software development process, including methods, tools and techniques to be followed.

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

Annexes — includes an organized list of all relevant documents and materials pertaining to the reader of the **Software Development Plan** as well as any project technical standards and plans which apply to this project.

2.   Project Overview

2.1   Project Purpose, Scope, and Objectives

The objective of our project is to create a platform for scheduling meetings for a group by compositing schedule information from each member of the group. The purpose is to improve the functionality of existing schedule comparison software to reduce repetitive actions and mitigate the need for active responses from group members for each event.

We plan to deliver a desktop application for client users and a back-end application that will run continuously on a central dedicated server and process and respond to internet data received from the client desktop application.

2.2   Assumptions and Constraints

Assumptions:
- All team members will be able to access the development environment.
- All group members actively use and update the calendars that are accessed by this platform.
- Our platform can integrate with existing calendar applications like Google Calendar or Microsoft Outlook.
- A stable network connection will be available for our application.
- We will have access to a dedicated server to process and respond to internet data received.

Constraints:
- The Dedicated server must be continuously running to process data.
- The project must adhere to constraints defined by this Project's Scope, Schedule, and Budget.
- The platform must be able to handle multiple requests by multiple users.
- Real-time or near real-time data synchronization is necessary to ensure all users are viewing up-to-date schedules.
- The system must implement authentication and authorization mechanisms to verify user identities and manage access controls.

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

2.3   Project Deliverables

[A list of the artifacts to be created during the project, including target delivery dates. The text below is provided as an example.] <mark>Requirements, design specs, test cases, code</mark>

Deliverables for each project phase are identified in the Development Case. Deliverables are delivered towards the end of the iteration, as specified in section *4.2.4 Project Schedule*.

2.4   Evolution of the Software Development Plan

Proposed Version	Date
0.1	2/19/2024
1.0	2/22/2024

Revision Criteria

- The *Software Development Plan* will be revised prior to the start of each Iteration phase.
- The *Software Development Plan* may be revised during each Iteration phase to better fit the scope of the project.
- The *Software Development Plan* will be revised after the end of each Iteration phase.

3.   Project Organization

3.1   Organizational Structure

The project team for this initiative follows a well-defined organizational structure designed to ensure efficient collaboration and effective decision-making. The team consists of various roles, each contributing to different aspects of the project. Here's an overview:

1. Project Manager: Shayna Weinstein

The project manager is at the helm, overseeing the entire project. They are responsible for planning, coordination, and ensuring that the project stays on track in terms of timelines and objectives.

2. Development Team: Everyone

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

This team comprises software developers, engineers, and programmers responsible for the actual creation and implementation of the software solution. They work closely with other teams to meet project requirements. Everyone is a part of our development team

3. Quality Assurance Team: Andrew Reyes

This team focuses on testing the software to identify and rectify any issues before the final release. Their role is critical in ensuring the product meets high-quality standards.

4. System Architects: Ben Phillips and Delroy Wright

System architects are responsible for designing the overall structure of the software. They make decisions on the system's architecture, ensuring scalability, security, and efficiency.

5. Review Authorities: Dylan Kneidel

In addition to the project manager, there are designated review authorities who provide oversight and guidance. This could include department heads, senior executives, or key stakeholders who review progress, make strategic decisions, and ensure alignment with organizational goals.

Communication Channels:

Regular communication channels are established to facilitate information flow within the team. This includes team meetings, progress reports, and documentation shared through project management tools moderated through Discord and Github. Regular review and feedback mechanisms are in place to adapt to changing circumstances and ensure the project's success.

3.2   External Interfaces

[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]

3.3   Roles and Responsibilities *[the more details here, the easier your job; include contact info, availability info, expertise, …]*

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

[Identify the project organizational units that will be responsible for each of the disciplines, workflow details, and supporting processes. The text below is provided as an example.]

Person	Unified Process for EDUcation Role
Shayna Weinstein shayna.weinstein@ku.edu	Team Administrator
Dylan Kneidel dckneidel@ku.edu	Documentation Engineer
Delroy Wright <deli email>	Configuration Management Co-Lead Technical Lead
Ben Phillips ben.a.phillips@ku.edu	Configuration Management Co-Lead
Andrew Reyes andrewreyes@ku.edu	Quality Assurance Engineer

Anyone on the project can perform Any Role activities.

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

4.   Management Process

4.1   Project Estimates

[Provide the estimated cost and schedule for the project, as well as the basis for those estimates, and the points and circumstances in the project when re-estimation will occur.]

4.2   Project Plan

[This section contains the schedule and resources for the project.]Project artifact as well as iteration schedules]

4.2.1   Phase Plan

[Include the following:

· a Gantt chart showing the allocation of time to the project phases (Not necessarily detailed to the activity level; this type of Gantt Chart is providing along with the Iteration Plans themselves; Provide an Overview of the project Timeline with the major miles stones]

· identify **major milestones** with their achievement criteria

Define any important release points and demos.]

[If available, refer to the related **Iteration Plan Documents** for more details]

4.2.2 Iteration Objectives

[Briefly list the objectives to be accomplished for each of the iterations and Refer to the related **Iteration Plan Documents** for more details.]

1. User Authentication and Basic Calendar Functionality:
   ○ Implement user authentication system.
   ○ Develop basic calendar functionality including viewing, editing, and saving personal schedules.

- ○ Enable users to form groups and invite members.
2. Group Scheduling and Event Creation:
    - ○ Allow users to schedule events within groups.
    - ○ Integrate group scheduling algorithm to find optimal meeting times based on members' schedules.
    - ○ Implement event creation and editing functionalities.
3. Enhanced Calendar Features and User Interface Design:
    - ○ Expand calendar features to include labeling, categorizing, and color-coding options
    - ○ Refine user interface for improved user experience
    - ○ Test and iterate on UI design based on user feedback
4. Advanced Group Management and Collaboration Tools:
    - ○ Implement features for group-specific calendar modifications
    - ○ Enable group administrators to manage membership and permissions

4.2.3 Releases

- ● Release #1 (beta):
    - ○ Iteration Objectives: User Authentication and Basic Calendar Functionality
    - ○ Features:
        - ■ User authentication system
        - ■ Basic calendar functionality (viewing, editing, saving personal schedules)
        - ■ Group formation and member invitation

- ● Release #2 (beta):
    - ○ Iteration Objectives: Group Scheduling and Event Creation
    - ○ Features:
        - ■ Scheduling events within groups
        - ■ Group scheduling algorithm for optimal meeting times
        - ■ Event creation and editing functionalities

- ● Release #3 (demo):
    - ○ Iteration Objectives: Enhanced Calendar Features and User Interface Design
    - ○ Features:
        - ■ Expanded calendar features (labeling, categorizing, color-coding)

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

- Refined user interface
- User interface testing and iteration based on feedback

- Release #4 (stable):
    - Iteration Objectives: Advanced Group Management and Collaboration Tools, Optimization, and Deployment
    - Features:
        - Group-specific calendar modifications
        - Group management tools (membership, permissions)
        - Collaboration tools (file sharing, chat)
        - Optimization for scalability and performance
        - Extensive testing for reliability and security
        - Deployment on cloud infrastructure

4.2.4 Project Schedule

[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.] [Limit to major project milestone, e.g., requirements, design, implementation, and testing]

Iteration/phase	Target Date
Finish initial project planning document	18/02/2024 - 25/02/2024
Develop basic calendar functionality	25/02/2024 - 10/03/2024
Implement user login/authentication	11/03/2024 - 20/03/2024
Optimize user interface and accessibility	21/03/2024 - 29/03/2024
Incorporate cloud-synchronization feature	31/03/2024 - 10/04/2024
Develop group calendar modification feature	11/04/2024 - 20/04/2024
Complete the project user manual	23/04/2024 - 25/04/2024
Final project implementation due	30/04/2024 - 01/05/2024

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

4.3   Project Monitoring and Control

· Quality Control:

Quality will be monitored by using test cases to verify correctness during the development period. During the planning phase, the Quality Assurance engineer will be scrutinous and will verify that the plan is feasible.

Any inadequacies will be addressed on the spot and a GitHub ticket will be created by the Quality Assurance Engineer. These inadequacies will be addressed by developers and then checked by the quality assurance engineer. After corrective action is taken, the ticket will be resolved, and the issue will be removed from the ticket list.

All artifacts will go through a review process (by the team) before being submitted for release. The review process will include the aforementioned test cases, where all team members will test the system using 1-10 personal tests.

· Risk Management:

Risks will be identified through a review process carried out by team members. If a risk is identified, a team member will notify the Quality Assurance Engineer. It will then be the quality assurance engineer's duty to carry out the fix. The review process will occur once per release (project deliverable) unless further review is necessary.

Initial identified risk cases:

Risk: A team member becomes sick

Mitigation: If this happens, the team will periodically check in with the ill member. The Team Administrator will work with the ill member to reduce their workload if necessary.

Risk: A team member drops the class

Mitigation: If this happens, the Team Administrator will evenly distribute tasks that were originally delegated to that member.

Risk: A team member is unable to finish assigned tasks before a deadline

Mitigation: If this happens, the Team Administrator will work with the team member to come to some solution. If there are repeated offenses, the Team Administrator will reach out to their lab TA for support in the matter.

·   Configuration Management:

Configurations and artifacts will be managed and stored using git and Github.com. Bug fix tickets will also be managed via git and Github.com. Whenever a ticket is created, it will be administered and monitored using this service. The Quality Assurance Engineer will work in this domain. Program configurations will be updated whenever a significant change is made to the source code. When updated, program comments and documentation will also be updated. All changes will be verified and permitted by the Quality Assurance Engineer. All additions to artifacts and executables will be included in every release.

4.4   Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

4.5 Quality Control

1. Code Reviews and Testing:

   - Regular code reviews will be conducted to ensure adherence to coding standards, best practices, and to identify potential issues early in the development process.
   - Comprehensive testing will be performed, including unit tests, integration tests, and end-to-end tests, to verify the functionality, reliability, and performance of the application.

2. **Bug Tracking and Resolution:**

   - A bug tracking system will be implemented to log and prioritize reported issues.
   - Timely resolution of bugs will be ensured through a structured process involving triaging, assigning, fixing, and verifying fixes.

3. Performance Monitoring and Optimization:

   - Continuous monitoring of application performance will be conducted to identify bottlenecks and areas for optimization.
   - Performance optimization efforts will be prioritized to maintain responsiveness and scalability, especially as the user base grows.

4. Release Mangament

   - Each release will undergo thorough testing and validation before deployment to ensure stability and reliability.
   - Rollback procedures will be established in case of unforeseen issues post-deployment, with a focus on minimizing downtime and user impact.

4.6 Reporting and Measurement

Updated schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

Refer to the Project Measurements Document (AAA-BBB-X.Y.doc) for detailed information.

4.7   Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity "Identify and Assess Risks". Project risk is evaluated at least once per iteration and documented in this table.

Risks:

1. Integration Difficulty - This risks describes any difficulties encountered in building and integrating our project, leading to inaccurate or incomplete versions.
2. Data Synchronization Issues - This risk describes issues encountered in synchronizing data across multiple schedules.
3. Team Collaboration Breakdown - This risk describes a breakdown in the team's communication and teamwork, leading to inadequate progress.
4. Unexpected Technical Issues - This risk describes unexpected issues in the technology required for our project, this could be server malfunction, lack of internet connection, etc.
5. Poor Tracking and Documentation - This risk describes a lack of adequate documentation and tracking for our project, leading to ambiguity and confusion.
6. Scope and Feature Creep - This risk describes expanding the project scope to include more features than initially planned, potentially causing delays and budget overruns.
7. Poor User Interface - This risk describes a malfunctioning, or incomplete user interface that does not provide users with the necessary functionality.

Refer to the Risk List Document (CCC-DDD-X.Y.doc) for detailed information.

Rock Chalk Rendezvous	Version: 1.0
Software Development Plan	Date: 02/22/2024
proj_plan1	

4.8   Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the Change Control Manager role.

Refer to the Configuration Management Plan (EEE-FFF-X.Y.doc) for detailed information.

Tools:

- Github Repository - Keeps database of changes, versions, and project artifacts.
    - Contains source code, test scripts, and data files.
    - Contains documentation related to source code and design documentation.
    - Contains all customer deliverable artifacts.
- Google Drive
    - Contains working documents like the Project Plan and Project Description that define our Project and its scope.

5.   Annexes

The project will follow the UPEDU process.

Other applicable process plans are listed in the references section, including Programming Guidelines.