Software Development Plan

for

<GET HOTEL SYSTEM>

Version 1.0

Prepared by <H21\_GREEN>

<RIVERFRONT SENTRAL HOTEL>

<7 March 2024>

Table of Contents

Table of Contents ii

Revision History iv

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms and Abbreviations 1

1.4 Overview 2

2. Project Overview 2

2.1 Project purpose, scope and objectives 2

2.2 Assumptions and constraints 2

2.3 Project deliverables 3

2.4 Evaluation of the Software Development Plan 3,4

3. Project Organization 4

3.1 Organization Structure 4

3.2 External Interfaces 4

3.3 Roles and Responsibilities 5,6

4. Management Process 6

4.1 Project Estimates 6

4.2 Project Plan 6

4.2.1 Phase Plan 6,7,8

4.3 Project Monitoring and Control 8

4.4 Risk Management Plan 8,9

4.5 Close-out Plan 10,11

5. Technical Process Plans 12

5.1 Methods, Tools and Techniques 12,13,14

6. Supporting Process Plans 15

6.1 Evaluation Plan 15,16,17,18

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| 1. Anaz Azzamuddin 2. Tan Jin Wan 3. Luqman Hazim 4. Nabil Aiman | 19 July 2020 | Initial release | 1.0 |

# Introduction

The Get Hotel System Project or Software Development Plan (SDP) serves as a comprehensive blueprint for the successful development and implementation of the cutting-edge hotel booking system, aiming to revolutionize the way travelers engage with accommodation services. This document outlines the purpose, scope, definitions, acronyms, abbreviations, references, and an overview of the project, setting the stage for a structured and efficient development process.

## Purpose

The primary purpose of the Get Hotel System is to create a state-of-the-art platform that seamlessly facilitates hotel bookings, providing users with a user-friendly, efficient, and secure experience. This SDP articulates the strategic objectives, goals, and key deliverables essential for the success of the project.

## Scope

The scope of the Get Hotel System encompasses the entire lifecycle of software development, from conceptualization to deployment. This includes requirements analysis, design, coding, testing, deployment, and maintenance. The system aims to cover a wide range of functionalities, including user registration, room selection, payment processing, and real-time availability updates.

## Definitions, Acronyms and Abbreviations

Hotel System: Refers to the software developed as part of the project to manage various aspects of hotel operations, including reservations, check-ins, check-outs, and other related functions.

## Overview

This section provides a high-level summary of the Get Hotel System project, detailing its objectives, key features, and anticipated benefits. It serves as a brief yet informative snapshot, allowing stakeholders to grasp the essence of the project before delving into the finer details outlined in subsequent sections.

In essence, the Get Hotel System Project or Software Development Plan sets the stage for a systematic and well-coordinated effort to bring the vision of an advanced hotel booking system to fruition, ensuring a streamlined development process and the delivery of a cutting-edge solution to the end-users.

# Project Overview

## Project purpose, scope and objectives

The primary purpose of the Get Hotel System project is to develop a sophisticated and user-centric hotel booking system that enhances the overall travel experience for users. This system aims to streamline and simplify the process of finding, reserving, and managing hotel accommodations, offering an intuitive platform that caters to both seasoned travelers and first-time users.

## Assumptions and constraints

Assumptions:

1. Data Accuracy: The accuracy and completeness of data provided by external sources, such as hotel details, prices, and availability, are assumed to be reliable and regularly updated.
2. User Accessibility: It is assumed that users accessing the Get Hotel System will have basic internet access and will be using devices that support modern web browsers.
3. Stable Technology Environment: The successful development and deployment of the Get Hotel System assume a stable and consistent technology environment, including reliable internet connectivity, server infrastructure, and third-party service providers.

Constraints:

1. Budget Constraints: The project operates within predefined budgetary constraints, requiring careful management of resources to ensure efficient use of funds throughout the development lifecycle.
2. Equipment and Infrastructure: The project is constrained by the availability and reliability of necessary equipment and infrastructure, including servers, hosting services, and development environments.
3. Security and Compliance Requirements: The project is constrained by the need to adhere to rigorous security standards and compliance regulations, influencing the design and implementation of the Get Hotel System.

## Project deliverables

1. User Interface (UI) Design: Visual representations of the system's user interface, showcasing the layout, navigation, and design elements.
2. User Account Management System: A feature-rich user account system allowing users to create profiles, manage bookings, and receive personalized recommendations.
3. Booking Engine: A fully functional and efficient booking engine that allows users to search for hotels, view details, and complete reservations.
4. User Account Management System: A feature-rich user account system allowing users to create profiles, manage bookings, and receive personalized recommendations.

## 2.4 Evaluation of the Software Development Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Changes |
| 1.0 |  | Anaz Azzamuddin | Initial Draft |
| 1.1 |  | Luqman Hazim | Added User Interfaces |
| 1.2 |  | Tan Jin Wan | Updated Scope and Objectives |
| 1.3 |  | Nabil Aiman | Stakeholder Feedback |
| 2.0 |  | Luqman Hazim | Finalized Version for Approval |

# Project Organization

## Organization Structure

Project Team:

1. Project Manager: Oversees the overall execution of the Get Hotel System project. Responsible for planning, coordination, resource allocation, and ensuring the project aligns with organizational goals.
2. Lead Developer: Responsible for the technical direction of the project, guiding the development team, and ensuring the adherence to coding standards and best practices.
3. Software Engineers: Execute the actual development tasks, including front-end and back-end development, database management, and integration of third-party APIs.
4. Change Control Manager: Manages change requests and ensures that any modifications to the project scope, requirements, or plan are thoroughly evaluated, approved, and documented.

## External Interfaces

Project Interfaces with External Groups:

1. Hotel Partners:

External Contact: Hotel Partnership Manager

Internal Contact: Stakeholder Liaison

Description: The Stakeholder Liaison serves as the primary point of contact between the project team and hotel partners. The Hotel Partnership Manager oversees the relationship with individual hotels, gathers information, and ensures seamless integration with the Get Hotel System.

1. Third-Party API Providers:

External Contact: API Integration Coordinator

Internal Contact: System ArchitectDescription: The System Architect coordinates with API Integration Coordinators to seamlessly integrate third-party services, such as maps and weather information, into the Get Hotel System. This ensures enhanced functionality and user experience.

1. Regulatory Compliance Authorities:

External Contact: Compliance Officer

Internal Contact: Project Manager

Description: The Project Manager works closely with Compliance Officers to ensure that the development and deployment of the Get Hotel System comply with regional and industry-specific regulations. Regular updates and documentation are maintained to address compliance requirements.

## Roles and Responsibilities

**Requirements Gathering and Analysis**

Organizational Unit: Project Leader

Workflow Details:

* Engage with external stakeholders to gather system requirements.
* Collaborate with hotel partners to understand specific needs and preferences.
* Document and analyze gathered requirements for development.

**Software Development**

Organizational Unit: Programmer

Workflow Details:

* Code and implement front-end and back-end functionalities.
* Ensure adherence to coding standards and best practices.
* Conduct code reviews and collaborate on version control.

**Deployment and Release Management**

Organizational Unit: Project Manager

Workflow Details:

* Develop a deployment plan for the Get Hotel System.
* Coordinate with stakeholders to ensure a smooth deployment process.
* Monitor system performance post-deployment and address any issues.

**Change Management and Documentation**

Organizational Unit: Document Maker

Process Details:

* Evaluate and approve change requests to the project scope or requirements.
* Review and approve major technical decisions.
* Ensure documentation is updated and aligned with any approved changes.

# Management Process

## Project Estimates

Cost laptop and documentation

## Project Plan

### Phase Plan

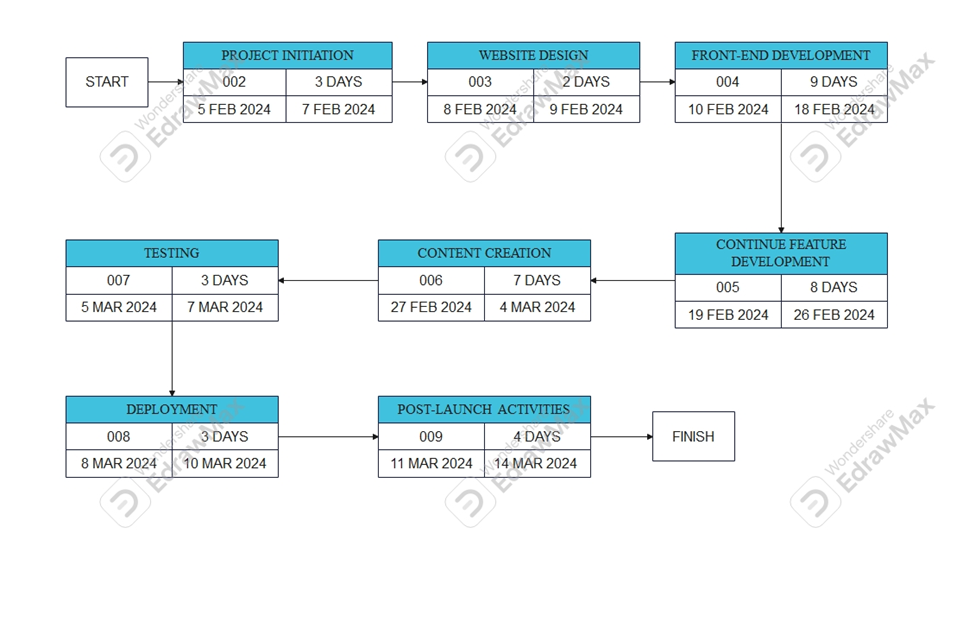
GANTT CHART

A screenshot of a computer program

Description automatically generated



### Project Schedule



## Project Monitoring and Control

Our system in programmer laptop. It makes other members cannot access in other laptop for our

references.

## Risk Management Plan

**Risk Identification and Response:**

1.1 Scope Creep:

Likelihood: Moderate

Impact: High

Mitigation: Clearly define project scope and objectives. Implement a robust change control process for any modifications.

Response: Strict change control process.

Contingency: Additional resources allocated for scope changes.

1.2 Technical Complexity:

Likelihood: High

Impact: High

Mitigation: Conduct thorough technical feasibility assessments. Allocate experienced developers and foster a culture of knowledge sharing.

Response: Prioritize technical feasibility studies.

Contingency: Access to specialized expertise for complex issues.

1.3 Third-Party Dependencies:

Likelihood: Moderate

Impact: Moderate

Mitigation: Identify critical dependencies early. Have contingency plans and alternative solutions in place.

Response: Regular communication and coordination.

Contingency: Identify alternative third-party providers.

1.4 Resource Constraints:

Likelihood: Low

Impact: High

Mitigation: Regularly review resource allocations. Have a contingency plan for key roles. Cross-train team members.

Response: Regular resource reviews and adjustments.

Contingency: Cross-training team members for key roles.

1.5 Security Vulnerabilities:

Likelihood: Moderate

Impact: High

Mitigation: Conduct regular security assessments. Implement encryption and secure coding practices.

Response: Regular security assessments and training.

Contingency: Swift response plan for identified vulnerabilities.

## Close-out Plan

**Staff Reassignment:**

Evaluate the workload and assign team members to other projects or responsibilities.

Conduct knowledge transfer sessions to share insights and lessons learned with relevant team members.

**Archiving of Project Materials:**

Archive all project documentation, including requirements, design specifications, source code, and testing documentation.

Ensure proper storage and categorization for easy retrieval in the future.

Post-Mortem Debriefing:

Conduct a post-mortem debriefing session with the project team.

Reflect on successes, challenges, and areas for improvement.

Gather feedback from team members on the project management process.

**Documentation:**

Compile a document summarizing insights gained during the project.

Document both positive practices and areas where improvements can be made.

Share this document with the team and relevant stakeholders.

**Client/User Feedback Analysis:**

Review and analyze user feedback received during and after the project.

Identify areas of satisfaction and areas for potential enhancements.

Incorporate valuable feedback into future project planning.

**Documentation Handover:**

Ensure that all relevant project documentation is handed over to the appropriate stakeholders.

Provide guidance on how to access and utilize archived project materials.

**Financial Closure:**

Finalize and close all financial aspects of the project.

Ensure that all invoices are settled, and budget allocations are reconciled.

**Project Report:**

Prepare a comprehensive project report summarizing the entire project lifecycle.

Include details on objectives, milestones achieved, challenges faced, and the overall project outcome.

Share the report with key stakeholders and senior management.

**Team Evaluation and Recognition:**

Evaluate the performance of team members and provide constructive feedback.

**Continuous Improvement Recommendations:**

Solicit suggestions for process improvements from team members.

Compile recommendations and integrate them into the organization's continuous improvement initiatives.

**Closure of Project Management Tools:**

Close project management tools, repositories, and communication channels.

Ensure proper documentation and communication of any project-related changes for future reference.

**Stakeholder Communication:**

Communicate the project's successful completion to all stakeholders.

Provide information on the final project status, outcomes, and any relevant post-project activities.

**Project Finalization:**

Complete any remaining tasks or outstanding items.

Ensure that all project deliverables meet the defined acceptance criteria.

# Technical Process Plans

## Methods, Tools and Techniques

The project adheres to a set of documented technical standards to ensure consistency, efficiency, and quality throughout the development process of the GET HOTEL SYSTEM. These standards encompass various aspects of the project life cycle, providing clear guidelines for different phases. The Business Modeling Guidelines establish a structured approach for representing and analyzing business processes, while the User-Interface Guidelines ensure a user-friendly and intuitive design. Use-Case Modeling Guidelines guide the identification and description of system functionalities.

Moving forward in the development process, Design Guidelines dictate the architectural and structural considerations for the system, ensuring scalability and maintainability. Programming Guidelines set coding standards to promote uniformity and readability in the source code. Test Guidelines define the procedures and criteria for rigorous testing to validate the system's functionality and performance. Finally, the Manual Style Guide dictates the standards for creating comprehensive and consistent documentation.

By referencing these documented standards, the project maintains a systematic and well-defined approach, fostering collaboration among team members and ultimately contributing to the successful development and deployment of the GET HOTEL SYSTEM.

**TECHNIQUES**

1. **HTML** (Hypertext Markup Language):

Main Function: Used to define the elements on a web page, such as headings, paragraphs, images,

links, and forms.

Extra Function

i. Interactive Order Summary - Create a real-time, interactive order summary that dynamically

updates as customers add or remove room .

ii. Real-time Price Calculation - Calculate the total order price in real-time as customers

customize their orders.

iii. Order History and Reordering - Offer registered users a user-friendly order history section to

view past orders and easily.

iv. Multi-Payment Options: Support multiple payment methods, including credit/debit cards,

online banking, and cash.

**TOOLS**

I. HARDWARE

i. Computer.

ii. Internet Connection. iii. Input Devices.

iv. Monitor.

**II. SOFTWARE**

i. Visual Studio Code, Notepad++ - Write Coding

ii. Google Chrome, Mozilla Firefox, and Safari - for testing and debugging the website.

iii. XAMPP - Web Server or Web Hosting Service

iv. MySQL - PHPMyAdmin - For a database management

v. Operating System - Windows

vi. Adobe Dreamweaver - Test Coding

vii. Animate.css - Use for CSS animation code

viii. cssbuttons.app - Use for Intuitive button design code

ix. Flaticons - Find High-Level Icons

x. Neumorphism.io - Help to generate soft UI CSS Colour for elements.

xi. Bootstrap - contains HTML, CSS and JavaScript-based design templates

xii. Figma - Use as interface design tool

13. Microsoft Word - Document Maker xiii.

Canva – Presentation or Slid

# Supporting Process Plans

## Evaluation Plan

Product Evaluation Plan: Get Hotel System Project

1. Introduction:

The Product Evaluation Plan outlines the strategies and methodologies for assessing the quality, functionality, and performance of the Get Hotel System. This plan is distinct from the Test Plan, which focuses on the testing processes, environments, and criteria. Product evaluation encompasses a broader perspective, including walkthroughs, inspections, and reviews, to ensure a comprehensive assessment of the product's overall success.

2. Evaluation Techniques:

2.1 Walkthroughs:

- Purpose: To provide an informal review of the system, emphasizing knowledge transfer and early identification of issues.

- Participants: Development team members, stakeholders, and subject matter experts.

- Procedure: Iterative walkthrough sessions focusing on different aspects of the system, such as user interface, functionality, and security.

2.2 Inspections:

- Purpose: Formalized reviews with a focus on defect identification and prevention.

- Participants: Cross-functional teams, including developers, testers, and relevant stakeholders.

- Procedure: Structured inspection meetings using checklists and predefined criteria to identify and address issues in the product.

2.3 Reviews:

- Purpose: Collaborative assessment of project artifacts and deliverables.

- Participants: Project team members, stakeholders, and external experts as needed.

- Procedure: Regular reviews of documentation, design specifications, and other project deliverables to ensure alignment with project goals and quality standards.

3. Evaluation Criteria:

3.1 Functionality:

- Assess the completeness and correctness of system features and functionalities.

- Criteria include adherence to requirements, ease of use, and consistency.

3.2 Performance:

- Evaluate the system's response time, scalability, and overall performance under different user loads.

- Criteria include response times, system resource utilization, and scalability metrics.

3.3 Security:

- Assess the security measures implemented in the system to protect user data and prevent unauthorized access.

- Criteria include adherence to security best practices, encryption, and vulnerability assessments.

3.4 Usability:

- Evaluate the user interface design and overall user experience.

- Criteria include intuitiveness, accessibility, and user satisfaction.

3.5 Reliability:

- Assess the system's reliability and stability under normal and extreme conditions.

- Criteria include system uptime, error handling, and recovery processes.

4. Evaluation Metrics:

4.1 Defect Density:

- Measure the number of defects identified per unit of code or functionality.

- Identify areas with higher defect density for focused improvement.

4.2 Code Review Effectiveness:

- Measure the efficiency of code reviews in identifying and addressing issues.

- Evaluate the percentage of defects found and resolved during code reviews.

4.3 User Satisfaction Surveys:

- Collect feedback from end-users to assess their satisfaction with the system.

- Utilize metrics such as Net Promoter Score (NPS) and user ratings.

5. Evaluation Procedures:

5.1 Regular Walkthroughs and Inspections:

- Conduct walkthroughs and inspections at key milestones throughout the development lifecycle.

- Use feedback gathered to inform iterative improvements.

5.2 Periodic Reviews:

- Schedule regular reviews of project artifacts and deliverables.

- Ensure that all documentation aligns with the project's evolving requirements.

5.3 Post-Implementation Review:

- Perform a comprehensive review after system deployment.

- Assess real-world performance, user feedback, and any issues arising in the production environment.

6. Continuous Improvement:

6.1 Feedback Mechanism:

- Establish a feedback loop to collect insights from evaluation processes.

- Use feedback to identify areas for improvement in both the product and development processes.

6.2 Process Adaptation:

- Adapt evaluation processes based on lessons learned and emerging best practices.

- Ensure continuous alignment with industry standards and user expectations.

7. Reporting:

7.1 Evaluation Reports:

- Generate regular evaluation reports summarizing findings, identified issues, and recommendations for improvement.

- Share reports with the project team, stakeholders, and senior management.

8. Roles and Responsibilities:

8.1 Evaluation Team:

- Identify team members responsible for leading and participating in evaluations.

- Include representation from different functional areas to ensure a holistic assessment.

9. Approval:

This Product Evaluation Plan is subject to approval by the Project Manager and relevant stakeholders. Sign-off indicates acknowledgment and agreement with the outlined strategies and procedures for product evaluation in the Get Hotel System Project.