Test Plan Document

For

No-Kill Louisville Animal Shelter

Sebastian Burman

Xiaokun Li

Jonathan Roberts

1 Introduction 3

2 Business Background 3

3 Test Objectives 3

4 Scope 3

5 Test types Identified 3

6 Problems Perceived 3

7 Architecture 3

8 Environment 3

9 Assumptions 3

10 Functionality 3

11 Security 4

12 Performance 4

13 Usability 5

14 Test Team Organization 6

15 Schedule 6

16 Defects Classification Mechanism 6

17 Configuration Management 6

18 Release Criteria 6

**Test Plan**

# Introduction

This document outlines the test strategy for the No-Kill Louisville animal shelter's digital check-in service. The plan covers various test phases designed to validate the integration of Twilio SMS services with the shelter's scheduling system, ensuring robust, secure, and user-friendly operations.

# Business Background

The No-Kill Louisville animal shelter utilizes Acuity for scheduling pet food pickups. The digital check-in service is designed to streamline the pickup process, allowing clients to notify the shelter of their arrival via SMS, which in turn prompts an immediate response from the volunteers

# Test Objectives

# Verify SMS reception and correct parsing of client messages.

# Ensure accurate API communication with the Acuity scheduling system.

# Confirm the reliability of volunteer notification dispatch.

# Assess overall system performance, security, and usability.

# Scope

***Inclusions***

* End-to-end testing of the check-in process.
* API integration with Acuity scheduling.
* SMS processing by Twilio and backend systems.
* Notification system accuracy and reliability.

***Exclusions***

* Detailed testing of third-party services internals (Twilio/Acuity).
* Physical hardware testing (e.g., servers, routers).

# Test types Identified

* Unit Tests: Validate individual code modules.
* Integration Tests: Ensure modules work together as expected.
* System Tests: Confirm the system meets all requirements.
* Security Tests: Check for vulnerabilities.
* Performance Tests: Assess system behavior under load.
* Usability Tests: Evaluate the user experience for both clients and volunteers.

# Problems Perceived

* Potential latency in SMS reception.
* Rate limiting or downtime from third-party APIs.
* Data inconsistencies between the Acuity system and the local database.

# Architecture

* SMS Reception: Twilio service to WebAPI server.
* API Communication: WebAPI server to Acuity scheduling system.
* Notification Dispatch: WebAPI server to volunteer interface.

# Environment

* Test will be conducted on a staging server mirroring the production environment.
* Twilio and Acuity APIs will be accessed using test accounts/modes.

# Assumptions

* Acuity scheduling system and Twilio provide stable test environments
* Test data is representative of live operational data.

# Functionality

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 | Limited test accounts for Twilio. | Use mock services for initial testing stages. |
| Constraint 2 | Shelter Acuity current subscription does not support API service | Use Data.json file to mock API calls and responses |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

* Risk: SMS delivery delays.
* Mitigation: Implement retries and fallback mechanisms.

***Test Strategy:*** Step-by-step verification of all documented workflows.

***Automation Plans:*** Automate SMS reception and API call simulations.

***Deliverables:*** Test reports detailing outcomes for each workflow.

# Security

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 | Encryption requirements for data at rest and in transit. | Use TLS for data in transit and AES for data at rest. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

* Risk: Unauthorized access to the API.
* Mitigation: Implement API Authentication with stringent policy enforcement.

***Test Strategy:*** Conduct penetration testing and vulnerability scanning.

***Automation Plans:*** Integrate security testing tools with the CI/CD pipeline.

***Deliverables:*** Security audit report.

# Performance

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 | Limited resources for load testing | Use cloud-based load testing services |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

***Test Strategy:*** Simulate peak loads and measure system responsiveness.

***Automation Plans:*** Automated load tests as part of the regular testing cycle.

***Deliverables:*** Performance test results with recommendations.

# Usability

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Infosys Limitations** |
| Constraint 1 | The Acuity notification must be intuitive for volunteers of varying technical skill levels. | Conduct beta testing with a small group of volunteers to gather feedback and iterate on the design |
| Constraint 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Risk Identified & Mitigation Planned***

* Risk: Poor user experience leading to volunteer errors.
* Mitigation: Conduct user experience testing with real volunteers

***Test Strategy:*** Heuristic evaluation and user testing sessions.

***Automation Plans:*** N/A, as usability tests will be conducted manually.

***Deliverables:***

* Usability Test Report summarizing feedback and recommendations
* User Testing Video Recordings
* Action Plan for Interface Improvements

# Test Team Organization

Jonathan Roberts will oversee the test planning, execution, and reporting and focus on complex test scenarios and automation.

# Schedule

Testing is scheduled to commence on March 1st and conclude by May 4th.

# Defects Classification Mechanism

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type of Defects | Functionality | Performance | Security | Usability | Compatibility |
| Critical | Defects that prevent client check-ins or notifications | A defect that results in unacceptable response times or system downtime. | A defect that leads to a breach of sensitive data or unauthorized system access. | A defect that renders the system unusable or leads to a significant number of user errors. | A defect that causes the system to be completely non-operational on certain platforms or devices. |
| Major | Defects that cause significant delays or confusion | A defect that causes noticeable performance degradation during peak load times. | A defect that could potentially be exploited to gain unauthorized access or data but requires specific conditions to exploit. | A defect that makes important system features difficult to use or understand. | A defect that limits functionality or alters the user experience on specific browsers or devices. |
| Minor | A defect that affects minor features that have easy workarounds or seldom-used functionalities. | A defect that causes minor performance issues that do not significantly affect user experience. | A defect that poses a low security risk, such as a vulnerability in a non-critical system component. | A defect that affects user convenience but does not significantly impact the overall usability. | A defect that causes minor layout issues on certain devices but does not affect functionality. |
| Cosmetics | A defect that does not affect the system's functionality, like typos in the UI. | A performance issue that is imperceptible to the user, like slight delays in UI animation. | Security defects that are theoretical in nature with no practical exploitation scenario. | Minor UI/UX issues that do not affect the functionality or user's ability to complete tasks. | A defect that causes very minor visual discrepancies across different browsers or devices, like font size differences. |

***Defects Logging and Status Changing Mechanism***

* Defects will be logged into a centralized bug tracking system such as Serilog form .net core framework
* Each defect will be categorized by type and severity upon logging
* The status of each defect will be tracked through its lifecycle: New, Confirmed, In Progress, Fixed, Verified, and Closed.

***Turn Around Time for defect fixes***

Critical: Immediate attention; resolution expected within 24-48 hours.

Major: High priority; resolution expected within 3-5 business days.

Minor: To be addressed in the next sprint cycle.

Cosmetic: Can be addressed in subsequent releases or when resources are available.

# Configuration Management

Purpose and Scope:

The configuration management plan for the No-Kill Louisville animal shelter's digital check-in system ensures that all software and hardware configurations are identified, documented, and maintained systematically, and that changes to configurations are controlled and tracked.

Configuration Identification:

Configurations that require management include:

* Software code repositories and branches.
* Hardware specifications for the server and networking equipment.
* Third-party services configurations (Twilio, Acuity Scheduling API).
* Application settings and environment variables.
* Database schema versions and data migration scripts.
* Documentation, including API specs and user manuals.

Version Control System:

* Tool: Git will be used as the version control system.
* Repositories: Separate repositories for front-end and back-end code.
* Branching Strategy: The main branch for production, development branches for ongoing work, feature branches for new features, and hotfix branches for urgent fixes.

Change Management Process:

* Request: Changes must be submitted via a change request form and approved by the project manager.
* Review: The impact and necessity of each change are reviewed by all of us.
* Approval: Changes with significant impact require sign-off from all of us.
* Implementation: Approved changes are scheduled for implementation during non-peak hours and with appropriate rollback plans.
* Documentation: All changes are documented, including the change description, rationale, implementation details, and testing results.
* Notification: Stakeholders are notified of significant changes, especially those that may affect system availability or functionality.

Build and Release Management:

* Automated Builds: Continuous Integration (CI) tools GitHub Actions is used for automated builds.
* Testing: All builds are subjected to automated unit tests and, upon success, are deployed to a testing environment.
* Release: Releases are versioned using Semantic Versioning and deployed to production after thorough testing and stakeholder approval.

# Release Criteria

The release criteria for the No-Kill Louisville animal shelter's digital check-in system define the specific conditions under which the software is considered stable and ready for deployment to the production environment. These criteria ensure that the system meets all the necessary quality standards and business requirements.

Functional Completeness

* All scheduled features and functionalities as per the requirements document are implemented and verified.
* User acceptance testing (UAT) is completed with the involvement of actual users from the shelter.

Defect Status

* All critical defects identified during testing phases are resolved and verified.
* Major defects are resolved or have acceptable workarounds with agreed-upon timelines for permanent fixes.
* Minor and cosmetic defects are documented, with a plan for resolution in future updates.

Performance Benchmarks

* The system meets predefined performance metrics such as response time, throughput, and resource utilization under expected load conditions.
* Stress testing verifies that the system maintains acceptable performance levels under peak loads.

Security Assurance

* Completion of security testing, including vulnerability assessments and penetration testing.
* No known security vulnerabilities that could lead to data breaches or unauthorized access.
* All data transmission is secured, and sensitive data is encrypted.