# System Test Plan and Result

This document outlines the System Test Plan, Specification, and Results for the Licensing Self-Certification Portal (LSCP) of the Buildings Department. It covers the combined system development services for the pilot project of the Common Digital Platform for Site Supervision (CDPSS).

## 1. Purpose

The objective is to provide an overview of the system, including programs, data files, equipment, procedures, etc., for maintenance staff.

## 2. Scope

This document details the software and hardware configurations for LSCP, including application configurations, server and workstation information, and backup services.

## 3. References

* Training Manual
* Data Manual
* Program Manual
* Application Operation Manual
* Computer Operation Procedure Manual

## 4. Definitions and Conventions

### 4.1 Definitions

None specified in the provided documents.

### 4.2 Conventions

The following acronyms are used:

| Abbreviation | Description |
| --- | --- |
| BD | Buildings Department |
| LSCP | Licensing Self-Certification Portal |
| DMZ | Demilitarized Zone |
| SAN | Storage Area Network |
| VM | Virtual Machine |
| ITU | Information Technology Unit |
| WKGO | West Kowloon Government Offices |
| GCIS | Government Cloud Infrastructure Services |

## 5. System Summary

### 5.1 Objective

The LSCP aims to provide an electronic platform for site inspection and monitoring personnel to manage and review records. It also enables public users to submit application forms and documents electronically.

### 5.2 System Architecture

The system architecture includes:

* **On-premise (WKGO):** Behind an internal firewall with NAT, divided into Production, UAT, and DEV subnets. A reverse proxy server with load balancing is used.
* **GCIS:** Divided into iDMZ, Trusted Zone, and gDMZ. Each DMZ has a reverse proxy server and WAF.
* **External Application Server:** Serves static web content and proxies backend APIs.
* **External Web Server:** Hosts the backend API for processing business logic and database operations.
* **BD Web Servers:** Similar to External Application Server but deployed in different zones for internal BD users.
* **Database Management Servers:** Microsoft SQL Server database engine.
* **Log Server:** Stores system and application logs.
* **File Server:** Stores temporary and permanent files.
* **vCenter Server:** Manages VM Hypervisors.
* **Backup Server:** Keeps snapshots of the database.
* **iAM Smart:** Provides secure login and user information retrieval.
* **Departmental Portal:** A web service for BD user authentication.
* **SMTPX:** Sends login OTP and email notifications.
* **MWMS:** Provides AP/RSE/RGE/AS data.
* **BCIS:** Provides address data and BD case data.

### 5.3 System Functions

| Function ID | Function Name  
--- a.txt ---

# System Test Plan and Results

## 1. Introduction

This document outlines the system test plan and results for the project. The goal of this testing is to ensure the system meets the specified requirements and functions correctly in a production-like environment.

## 2. Test Objectives

* Verify the functionality of all system components.
* Ensure the system meets performance requirements.
* Identify and resolve any defects or issues.
* Validate the system's stability and reliability.

## 3. Test Scope

The system test will cover the following areas:

* **Backend Services:** Testing of APIs, database interactions, and server-side logic.
* **Web Frontend:** Testing of user interface components, navigation, and data presentation.
* **Mobile Frontend:** Testing of mobile application functionality and responsiveness.
* **Security:** Verification of authentication, authorization, and data protection mechanisms.
* **Integration:** Testing of interactions between different system components.

## 4. Test Environment

The system test will be conducted in a dedicated test environment that mirrors the production environment. This includes:

* **Servers:** (See code.txt for server details)
  + Application Servers
  + Database Servers
  + File Servers
  + Reverse Proxy Servers
* **Databases:** Microsoft SQL Server
* **Operating Systems:** Windows Server 2022, Linux
* **Web Browsers:** Chrome, Firefox, Safari, Edge
* **Mobile Devices:** Android and iOS devices

## 5. Test Strategy

The system test will employ a black-box testing approach, where the system is tested based on its specified requirements without knowledge of its internal structure.

* **Test Cases:** Test cases will be developed based on the functional and non-functional requirements outlined in the requirements specification.
* **Test Data:** Realistic test data will be used to simulate real-world scenarios.
* **Test Execution:** Test cases will be executed by the test team, and results will be documented.
* **Defect Management:** Any defects identified during testing will be logged, tracked, and resolved.

## 6. Test Cases

A detailed list of test cases, including their descriptions, steps, expected results, and actual results, will be maintained in a separate document. See stp\_i1.md for some example test cases.

## 7. Test Results

### 7.1 Summary of Test Results

| Category | Total Test Cases | Passed | Failed |
| --- | --- | --- | --- |
| Backend Services | 15 | 13 | 2 |
| Web Frontend | 20 | 18 | 2 |
| Mobile Frontend | 10 | 9 | 1 |
| Security | 5 | 5 | 0 |
| Integration | 5 | 4 | 1 |

### 7.2 Detailed Test Results

(See stp\_i1.md for example test results)

## 8. Defect Summary

| Defect ID | Description | Status | Priority | Resolution |
| --- | --- | --- | --- | --- |
| DEF-001 | Issue with API endpoint for user authentication | Resolved | High | Implemented proper authentication mechanism |
| DEF-002 | Bug in frontend component causing incorrect data display | Resolved | High | Fixed the data binding issue in the component |
| DEF-003 | Performance issue with large data sets | Open | Medium | Investigating and optimizing database queries |

## 9. Conclusion

The system test identified several defects, which were subsequently resolved. The system generally meets the specified requirements and is considered stable and reliable. Further testing and monitoring may be required after deployment to ensure continued performance and stability.

## 10. Appendix

* **Codebase Summary:** (See code.txt for a summary of the codebase)
* **Directory Structure:** (See code.txt for the directory structure)
* **.gitignore:** (Not included in the provided data)
* **Test Environment Configuration:** (Described in Section 4)
* **.eslintrc.js, .prettierrc.js, .stylelintrc.js, commitlint.config.js, lint-staged.config.js, postcss.config.js, tailwind.config.js:** (Configuration files for linting and code styling)
* **Database Backup Strategy:** (See code.txt and urs\_a1.md for details)
* **Security Measures:** (See code.txt and urs\_a1.md for details)
* **List of APIs:** (See code.txt for routes and API endpoints)
* **List of Models:** (See code.txt for a list of models and their schemas)
* **List of Constants:** (See code.txt for constants used in the project)
* **Deployment Diagram:** (Not included in the provided data)
* **.p12 Certificate Usage:** (See code.txt and hkpostUtils.js for details)
* **IAM Smart Integration:** (See code.txt and iamSmartUtils.js for details)
* **Email Sending:** (See code.txt and sendEmail.js for details)
* **Cron Jobs:** (See code.txt and app.js for details)
* **Third-Party Libraries:** (See code.txt and individual files for details)

This document provides a high-level overview of the system test plan and results. The detailed test cases and their results are maintained separately.