## Test Plan: Job Portal SWP391

**Project Name:** Job Portal SWP391  
**Prepared By:** SWP391 Team  
**Validated By:** Tran Van Phuc  
**Dated:** 1/7/2025  
**Plan Version:** 1.0

### 1. Introduction

#### 1.1 Purpose

This Test Plan is created to: - Define a comprehensive testing strategy for the Job Portal SWP391 system. - Ensure product quality meets the requirements of employers, job seekers, and administrators. - Establish systematic testing procedures for all system functions, from basic to advanced. - Minimize risks and ensure the system operates stably across multiple platforms.

#### 1.2 Background

The Job Portal SWP391 system is developed to:

- Connect employers with job seekers.

- Create job opportunities and support efficient recruitment management.

- Support multiple platforms (PC, mobile, tablet).

- Build a community for sharing job-seeking and recruitment experiences.

#### 1.3 Scope

**Within Testing Scope:**

- All functions for three user types: Employer, Job Seeker, Admin.

- User interface on web browsers.

- Online payment integration (PayOS).

- Messaging, real-time chat, and AI chatbot system.

- Responsive design for multiple devices.

- User information security.

**Out of Testing Scope:**

- Native mobile applications (only responsive web testing).

- Internal logic of third-party payment gateways.

- Email server configuration.

#### 1.4 Project Identification

| Document | Available | Reviewed | Author/Resource | Notes |
| --- | --- | --- | --- | --- |
| Requirements Specification | Yes | Yes | SWP391 Team | Requirement Outline v1.0 |
| Functional Specification | Yes | Yes | SWP391 Team | Derived from requirements |
| Use-Case Reports | Yes | Yes | SWP391 Team | User stories documented |
| Project Plan | Yes | Yes | SWP391 Team | SWP391 project timeline |
| Design Specifications | In Progress | No | SWP391 Team | UI/UX mockups |
| Prototype | In Progress | No | SWP391 Team | Early demo version |
| User’s Manuals | No | No | SWP391 Team | To be created |
| Business Model or Flow | Yes | Yes | SWP391 Team | Recruitment ecosystem flow |
| Data Model or Flow | Yes | Yes | SWP391 Team | Database ERD |
| Business Functions and Rules | Yes | Yes | SWP391 Team | Business logic rules |
| Project or Business Risk Assessment | Yes | Yes | SWP391 Team | Risk matrix created |

### 2. Test Approach

#### 2.1 Testing Stages

* **Unit Testing:**
  + Test individual components/modules (Servlets, DAO classes, Models, Utilities).
  + Validate business logic methods and database CRUD operations.
  + Target: 80% code coverage.
* **Integration Testing:**
  + Test interaction between frontend and backend.
  + Validate API endpoints and data flow.
  + Test database integration and third-party services (PayOS, Google OAuth, email).
* **System Testing:**
  + End-to-end testing of complete user workflows.
  + Cross-browser compatibility testing.
  + Performance and security testing.
* **Acceptance Testing:**
  + User Acceptance Testing (UAT) with stakeholders.
  + Business process validation.
  + Usability testing with real users.
  + Final verification before deployment.

#### 2.2 Testing Types

**2.2.1 GUI Testing:**

|  |  |
| --- | --- |
| Test Objective: | Ensure all user interface elements are displayed correctly and consistently across supported browsers and devices. Validate layout, colors, fonts, icons, and controls for user-friendliness. |
| Technique: | Manual testing on Chrome, Firefox, Edge, Safari (desktop and mobile). Responsive design testing using browser dev tools and real devices. Accessibility testing (WCAG 2.1 AA) with screen readers. Visual regression testing (screenshot comparison). |
| Completion Criteria: | All UI elements render correctly on all target browsers and devices. Responsive breakpoints work as designed. Navigation is intuitive and error-free. Accessibility standards are met. |
| Special Considerations: | Test on both emulators and real mobile devices. Verify touch interactions and gestures on mobile. Check loading states, error messages, and modals or popups. Validate language or locale if multi-language is supported. |

**2.2.2 Functional Testing:**

|  |  |
| --- | --- |
| Test Objective: | *Verify that all functional requirements are implemented correctly, including registration, login, profile management, job posting, application, payment, chat, and admin features.* |
| Technique: | Black-box testing based on requirements and user stories. Positive and negative test scenarios. Boundary value analysis for input fields. State transition testing for workflows. |
| Completion Criteria: | All user stories and requirements pass acceptance criteria. Error handling works as expected. Data validation prevents invalid inputs. Business rules are properly enforced. |
| Special Considerations: | Test multi-language support if applicable. Verify email notifications and third-party integrations. Test concurrent user scenarios, for example, multiple users applying for the same job. |

**2.2.3 Performance Testing:**

**2.2.3.1) Performance Profiling:**

|  |  |
| --- | --- |
| Test Objective: | Identify performance bottlenecks, measure response times for critical actions, and analyze database query performance. |
| Technique: | Application profiling using Java profilers such as VisualVM. Database query analysis with SQL Server Profiler. Frontend performance monitoring with browser developer tools. |
| Completion Criteria: | Page load times under 3 seconds for 95 percent of requests. Database queries execute within acceptable timeframes. Memory and CPU usage remain stable during extended use. |
| Special Considerations: | Profile both development and production-like environments. Monitor resource consumption during peak usage. Focus on critical user flows such as login, job search, application, and payment. |

**2.2.3.2) Load Testing:**

|  |  |
| --- | --- |
| Test Objective: | Verify system performance under expected and peak user loads. Test scalability, resource utilization, and identify breaking points. |
| Technique: | Gradual load increase testing using JMeter or similar tools. Concurrent user simulation, for example, 100 or more users. Stress testing beyond normal capacity. |
| Completion Criteria: | System handles 100 concurrent users without significant degradation. Response times remain acceptable under normal load. System recovers gracefully from overload conditions. |
| Special Considerations: | Simulate realistic user behavior patterns. Include database and file system stress testing. Monitor for memory leaks and resource exhaustion. |

**2.2.4 Configuration Testing:**

|  |  |
| --- | --- |
| Test Objective: | Ensure the system works correctly in different server and environment configurations. Validate deployment and configuration parameters. |
| Technique: | Deploy on multiple server configurations such as Windows and Linux. Test with different database connection settings. Verify environment-specific configurations for development, test, and production. |
| Completion Criteria: | Application deploys successfully on all target servers. Configuration files are properly managed and documented. Environment-specific features work correctly. |
| Special Considerations: | Test both development and production configurations. Verify backup and recovery procedures. Check for configuration errors and missing dependencies. |

**2.2.5 Installation Testing:**

|  |  |
| --- | --- |
| Test Objective: | Verify installation procedures work correctly, including system deployment, configuration, database setup, and initial data loading. |
| Technique: | Fresh installation testing on clean environments. Upgrade testing from previous versions. Installation on different operating systems and server setups. |
| Completion Criteria: | Installation completes without errors. All required components are properly installed and configured. System starts and functions correctly after installation. |
| Special Considerations: | Document installation prerequisites and steps. Test rollback procedures if installation fails. Validate uninstallation and cleanup. |

**2.2.6 Browser Testing:**

|  |  |
| --- | --- |
| Test Objective: | Ensure cross-browser compatibility and consistent user experience across all supported browsers. Verify JavaScript and CSS functionality. |
| Technique: | Manual and automated cross-browser testing on Chrome, Firefox, Edge, and Safari. JavaScript compatibility testing. Responsive and mobile browser testing. |
| Completion Criteria: | Application works correctly on all target browsers and devices. No browser-specific bugs or rendering issues. JavaScript and CSS features function consistently. |
| Special Considerations: | Test on both desktop and mobile browsers. Include testing on older browser versions if required. Validate third-party integrations such as Google OAuth and PayOS on all browsers. |

### 3. Test Deliverables

* Test Plan (this document)
* Test Cases (detailed specifications)
* Test Scripts (automated scripts if applicable)
* Test Data (sample data for scenarios)
* Test Reports (execution results, defect reports)
* Summary Report (final test summary and recommendations)
* Deployment Guide (production deployment instructions)
* User Acceptance Test Report (UAT results and sign-off)

### 4. Environmental Needs

#### 4.1 Hardware

| Machine | Configuration |
| --- | --- |
| Server | CPU: 4 cores, RAM: 8GB, SSD: 500GB, Network: 1Gbps |
| Client | CPU: 2 cores, RAM: 4GB, Display: 1920x1080, Mobile devices for testing |

#### 4.2 Software

| Machine | OS | Browser | Software Required for Testing |
| --- | --- | --- | --- |
| Server | Windows Server 2019 / Linux Ubuntu 20.04 | N/A | Java 11+, Tomcat 9+, SQL Server 2019 |
| Client | Windows 10 / macOS / Android / iOS | Chrome, Firefox, Safari, Edge | Browser dev tools, Postman |

#### 4.3 Tools

* Test Management: Jira, Google Sheets
* Automation: Selenium WebDriver
* Performance: JMeter
* Database: SQL Server Management Studio
* API Testing: Postman
* Bug Tracking: Jira
* Version Control: Github

### 5. Staffing

#### 5.1 Responsibilities

| Role | Responsibility | Person/Team |
| --- | --- | --- |
| Test Plan | Create and maintain test strategy | Test Lead |
| Test Cases | Design and document test cases | Test Analysts |
| Manual Testing | Execute manual test cases | Testers |
| Automated Testing | Develop and maintain automation scripts | Automation Engineer |
| Performance Testing | Design and execute performance tests | Performance Tester |
| UAT Coordination | Coordinate user acceptance testing | Business Analyst |

#### 5.2 Training

* Testing Tools Training: Selenium, JMeter, SQL Server
* Domain Knowledge: Recruitment processes and workflows
* Security Testing: OWASP guidelines and security testing practices
* Agile Testing: Continuous integration and testing practices

### 6. Dependencies/Risks

#### 6.1 Dependencies

* Complete development environment setup
* Availability of realistic test data
* Test accounts for third-party services (PayOS, Google)
* Test servers and database instances

#### 6.2 Risks

| Risk | Impact | Probability | Mitigation Strategy |
| --- | --- | --- | --- |
| Incomplete requirements | High | Medium | Regular requirement reviews and clarifications |
| Performance issues | High | Medium | Early performance testing and optimization |
| Security vulnerabilities | High | Low | Security code reviews and penetration testing |
| Browser compatibility issues | Medium | Medium | Continuous cross-browser testing |
| Database performance | Medium | Medium | Database optimization and indexing |
| Timeline delays | Medium | High | Parallel testing activities and resource allocation |

### 7. Schedule and Milestones

#### 7.1 Schedule

| Testing Type | Unit Testing | Module Testing | System Testing |
| --- | --- | --- | --- |
| GUI | 01/07/2025 | 08/07/2025 | 15/07/2025 |
| Functional | 02/07/2025 | 09/07/2025 | 16/07/2025 |
| Performance | 03/07/2025 | 10/07/2025 | 17/07/2025 |

#### 7.2 Milestones

| Module Name | Release Date |
| --- | --- |
| User Management Module | 05/07/2025 |
| Job Posting/Experience | 12/07/2025 |
| Application/Payment Module | 19/07/2025 |
| Admin Management Module | 26/07/2025 |
| Full System Integration | 31/07/2025 |

### 8. Approvals

| Project Manager | Quality Manager |
| --- | --- |
| Tran Van Phuc | Tran Van Phuc |