**Software Requirements Specification (SRS) Document**  
**Project Name: Automated Software Testing System (ASTS)**

### ****1. Introduction****

#### ****1.1 Purpose****

Automated Software Testing System (ASTS) offers a cost-effective, efficient, and scalable automated software testing system with minimal human intervention. ASTS test cases are automated, bugs are caught, and they are reported for improvement in software quality, automating the process, and reducing time-to-market. ASTS is tailored for effective CI/CD of agile development teams.

#### ****1.2 Scope****

ASTS is an API-based, web-based developer, tester, and project manager application. It provides:

* Improved efficiency and accuracy by automating the testing.
* Reliability in software through tracking and detection of defects.
* Secure version control test case management.
* Rich data dashboards for analytics and reporting.
* Testing paradigms support (unit, regression, integration, performance, security).
* Test case generation and optimization using AI (long-term objective).

#### ****1.3 Definitions, Acronyms, and Abbreviations****

* **ASTS**: Automated Software Testing System
* **CI/CD**: Continuous Integration/Continuous Deployment
* **RBAC**: Role-Based Access Control
* **SRS**: Software Requirements Specification

#### ****1.4 References****

* IEEE 830-1998 (Software Requirements Specification Standard)
* IEEE 829-2008 (Software Test Documentation)
* ISTQB Software Testing Standards
* ISO/IEC 25010:2011 (Software Quality Model)
* GDPR Compliance Standards

#### ****1.5 Overview****

This specification gives the functional and non-functional requirements of ASTS, defining an ordered guide for its development, deployment, and maintenance.

### ****2. Functional Requirements****

#### ****2.1 User Authentication and Authorization****

* Username and password-based secure login.
* Third-party integration authentication with OAuth 2.0.
* Admin, Tester, Developer, Guest-based implementation of RBAC.

#### ****2.2 Test Case Management****

* Generation of tickets for create, update, delete, and group of test cases.
* Support for parameterized test cases.
* Categorization based on priority and application module.

#### ****2.3 Test Execution****

* Fetching executions on schedules/triggers.
* Parallel support for execution to perform better.
* Support for integration with CI/CD pipelines so that auto tests execute on each deployment.

#### ****2.4 Defect Logging & Tracking****

* User reporting, tracking, and updating of defect status (Open, In Progress, Resolved, Closed) feature.
* Attach screenshot/logs option for debugging feature.

#### ****2.5 Test Reporting & Logging****

* Execute reports with pass/fail result and failure reason.
* Export reports as input in alternative file formats (PDF, CSV, HTML).
* Trend visualization analysis dashboards.

#### ****2.6 Integration with Development Tools****

* Git, GitHub integration for version control.
* Included Jira, Trello integration for issue tracking.
* API-based custom integration support.

#### ****2.7 Notification System****

* In-app and email alerts for test result.
* Major failure alert.

#### ****2.8 AI-Based Test Automation (Future Scope)****

* Test case generation based on execution history by using AI.
* Defect prediction and automatic optimization of test cases.

### ****3. Non-Functional Requirements****

#### ****3.1 Performance Requirements****

* Finishing unit tests in 5 seconds.
* Focusing regression tests in 30 minutes.
* Support for executing 100+ tests simultaneously.

#### ****3.2 Usability Requirements****

* Natural and intuitive navigation of UI.
* Support for cross-platform (mobile, web).

#### ****3.3 Reliability & Availability****

* System availability of 99.9% to facilitate seamless operation.
* Failover features for automating test runs.

#### ****3.4 Security Requirements****

* AES-256 encryption for storage of data.
* TLS 1.2 encryption for data transport.
* Role-based access control (RBAC) for denying access.
* Encrypted APIs for data integrity.

#### ****3.5 Maintainability & Scalability****

* Backend structure with easy updates and insertions possible.
* Horizontal scaling for high test loads.
* Few-maintenance design, i.e., practically no developer interference.

#### ****3.6 Compatibility Requirements****

* Support for Windows, macOS, Linux.
* Support for Chrome, Firefox, Edge, Safari.

#### ****3.7 Compliance Requirements****

* ISTQB, IEEE 829, GDPR compliance.
* Follows standard industry data policies to maintain user privacy.

### ****4. Justification for Non-Functional Features****

#### ****4.1 Maintainability****

* Modular design makes it very simple to add changes without system-wide modification.
* Microservices architecture enables independent scaling of various modules.

#### ****4.2 Reliability****

* Failover procedures in an automated way ensure system reliability.
* Cloud infrastructure enhances availability and disaster recovery.

#### ****4.3 Security****

* Secure storage of test case data is ensured by AES-256 encryption.
* TLS 1.2 encryption avoids data transmission interception.
* Role-based access control grants proper user permissions to preserve data integrity and confidentiality.

#### ****4.4 Reusability****

* Test items are reusable by other projects with object-oriented design.

##### **Examples of Test Cases:**

* Generation of purchase orders
* Tracking of ships

### ****5. Appendix****

* **GitHub Repository:** [GitHub Repository](https://github.com/AribaMumtaz/SE_Assignment1/blob/main/Final%20Doc.docx)
* **Notion Document Link:** [Notion Document](https://www.notion.so/199a647e4f75808cb8b3ebf1cfde741c?v=199a647e4f75808cb864000cb3ce6885)
* **Project Members & Contact Details:** To be included.

### ****6. Conclusion****

The Automated Software Testing System (ASTS) is envisaged to revolutionize software testing by combining automation, defect tracking, and AI-driven enhancement. By offering robust security, compliance, and scalability, ASTS simplifies software quality assurance processes and improves team productivity.

This paper will be formally submitted on Notion and GitHub.

**Submission Date:** Friday, 28th February 2025, 11:00 PM.  
**Instructor Email:** [usman.younis@itu.edu.pk](mailto:usman.younis@itu.edu.pk)  
**TA Email:** [ayeshaaman.ta@itu.edu.pk](mailto:ayeshaaman.ta@itu.edu.pk)