Software Test Document for Social Butterfly

Version <1.0> Approved

Prepared by:

Group Name: Code Crafters

Juliana Unger	z23618477	junger2021@fau.edu
Matthew Henao	z23685608	henaom2022@fau.edu

Instructor: Safak Kayıkcı

Course: Principles Software Engineering

Section: 4010-002

Teaching assistant: Preston Billion Polak

Date: July 28, 2024

1.0 Introduction

1.1 Test Plan Identifier

- **Test Plan ID:** SB-TP-001

1.2 Purpose

The purpose of this Software Test Document is to outline the testing strategy, objectives, resources, schedule, and deliverables required to validate the quality and functionality of the Social Butterfly project. This document will serve as a reference for the testing team to ensure comprehensive and effective testing of the system.

1.3 Scope

The scope of testing for the Social Butterfly project includes functional testing, performance testing, security testing, and usability testing. The primary focus will be on validating the data retrieval, data storage, data analysis, and reporting components, as well as the user interface. The scope excludes testing of future integrations with additional social media platforms and advanced machine learning algorithms for predictive purposes.

2.0 Test Plan

2.1 Test Items

- **Data Retrieval Component**
  - Retrieve data from Twitter (X) using Tweepy.
  - Retrieve data from Reddit using PRAW.
- **Data Storage Component**
  - Store retrieved data in the database.
  - Ensure data integrity and prevent duplication.
- **Data Analysis Component**
  - Perform sentiment analysis.
  - Identify trends.
- **Reporting Component**
  - Generate reports in PDF format.
  - Generate reports in CSV format.
- **User Interface**
  - Test all user interface elements for functionality.
  - Assess usability and user experience.

2.2 Features to be Tested

- **Data Retrieval**: Ensure data can be retrieved accurately from Twitter (X) and Reddit using the Tweepy and PRAW libraries.

- **Data Storage**: Verify that the retrieved data is stored correctly in the database without loss or duplication.

- **Data Analysis**: Test the algorithms for sentiment analysis and trend identification to ensure they work as expected.

- **Report Generation**: Ensure that reports can be generated in both PDF and CSV formats and that the data within the reports is accurate.

- **User Interface**: Evaluate the functionality and usability of the user interface, ensuring that all elements work as intended and provide a good user experience.

2.3 Features Not to be Tested

- **Future Integrations**: Testing for potential future integrations with additional social media platforms is outside the current scope.

- **Advanced Machine Learning Algorithms**: Testing of advanced machine learning algorithms for predictive analysis is not included in the current scope.

2.4 Approach

- **Functional Testing:** Verify each component of the system against the functional requirements.
- **Performance Testing:** Assess the system's performance under various load conditions.
- **Security Testing:** Ensure user authentication, data protection, and compliance with data privacy laws.
- **Usability Testing:** Evaluate the user interface for ease of use and accessibility.

2.5 Item Pass/Fail Criteria

- **Pass:** The test case is considered passed if the actual results match the expected results without any critical or major defects.

- **Fail:** The test case is considered failed if the actual results do not match the expected results or if there are any critical or major defects.

2.6 Suspension Criteria and Resumption Requirements

- **Suspension Criteria:** Testing will be suspended if critical issues are found that block further testing.

- **Resumption Requirements:** Testing will resume once the critical issues have been resolved and verified.

2.7 Test Deliverables

- **Test Plan Document**: A comprehensive document outlining the testing strategy, objectives, resources, schedule, and deliverables.

- **Test Cases and Test Scripts**: Detailed test cases and scripts for manual and automated testing, specifying inputs, actions, and expected outcomes.

- **Test Execution Reports**: Reports detailing the execution of test cases, including pass/fail status, time taken, and any deviations from expected results.

- **Defect Reports**: Documentation of any defects found during testing, including severity, steps to reproduce, screenshots, and status updates.

- **Final Test Summary Report**: A summary of all testing activities, results, lessons learned, and any outstanding issues or recommendations for future testing.

2.8 Testing Tasks

- **Develop Test Plan and Test Cases**: Create a comprehensive test plan and detailed test cases that cover all aspects of the system.

- **Set Up Test Environment**: Prepare the necessary hardware, software, and network configurations for testing. This includes setting up servers, installing required software, and configuring network settings.

- **Execute Test Cases**: Run the test cases according to the test plan, document the results, and compare them against expected outcomes.
- **Log and Track Defects**: Record any defects found during testing, categorize them by severity, and track their resolution through a defect management system.

- **Retest Resolved Defects**: Verify that defects have been fixed correctly and that no new issues have been introduced. Conduct regression testing as necessary.

- **Prepare Test Summary Report**: Compile the results of the testing activities into a final report that includes an overview of the testing process, key findings, and recommendations for improvement.

2.9 Environmental Needs

- **Hardware:** Servers, client machines for testers.

- **Software**: Operating systems, browsers, database management systems, Tweepy and PRAW libraries.

- **Network:** Stable internet connection for API interactions.

- **Test Data:** Sample data for testing purposes.

2.10 Responsibilities

- **Test Manager:** Oversee the testing process, manage the testing team, and ensure timely completion of testing tasks.

- **Testers:** Execute test cases, log defects, and retest resolved defects.

- **Developers:** Fix defects reported by the testing team.

- **Project Manager:** Coordinate between the testing and development teams.

2.11 Staffing and Training Needs

- Ensure sufficient staffing with experienced testers.

- Provide training on the Social Butterfly project, testing tools, and libraries used (Tweepy, PRAW).

2.12 Schedule

- **Planning Phase:** [7/06/24] - [7/08/24]
- **Test Case Development:** [7/08/24] - [7/10/24]
- **Test Execution:[**7/10/24] - [7/14/24]
- **Defect Resolution:** [7/14/24] -7/20/24]
- **Final Validation and Reporting:** [7/20/24] - [7/28/24]

2.13 Risks and Contingencies

- **Risk:** API rate limiting may affect data retrieval testing. Mitigation: Implement caching and use test accounts with higher rate limits.

- **Risk:** Network issues may disrupt testing. Mitigation: Ensure backup internet connection and local test environment setup.

2.14 Approvals

- Team Manager: [Matthew Henao, Signature, Date]
- Project Manager: [Juliana Unger, juliana unger , 7/27/24]
- Development Lead: [Name, Signature, Date]

3.0 Test Cases

3.1 Test Case Identification

Each test case will be uniquely identified with a test case ID following the format TC

3.2 Test Case Description

Test Case ID: 1
Description: Verify data retrieval from Twitter using the *fetch_twitter_data* method.
Preconditions: Valid Twitter API credentials.

Steps:

1. Call *fetch_twitter_data* with valid parameters.
2. Verify the API response status.

3. Validate the structure and content of the retrieved data.

Expected Results:
- Should return a list of structured tweet data with no errors.

Test Case ID: 2
Description: Verify data storage functionality using the *store_data* method.
Preconditions: Retrieved data from the data retrieval component.

Steps:

1. Call *store_data* with valid data.
2. Verify that data is correctly stored in the database.
3. Check for data duplication or conflicts.
   Expected Results:
   - Data should be stored correctly and retrievable without duplication.

4.0 Test Scenarios

4.1 Normal Scenarios

- **Scenario 1:** Successful data retrieval and storage.
- **Scenario 2:** Accurate sentiment analysis and trend generation.
- **Scenario 3:** Proper report generation in PDF and CSV formats.

4.2 Edge Cases

- **Scenario 1:** API rate limiting handling.
- **Scenario 2:** Handling of empty or malformed data.
- **Scenario 3:** User authentication failures.

4.3 Error Cases

- Scenario 1: Network or API errors during data retrieval.
- Scenario 2: Database connection issues.
- Scenario 3: Invalid user input during search or report generation.

5.0 Item Pass/Fail Criteria

- **Pass:** A test item is considered passed if all associated test cases meet the expected results without any critical or major defects.

- **Fail:** A test item is considered failed if any associated test case does not meet the expected results or if there are any critical or major defects that impact functionality or user experience.

6.0 Suspension Criteria and Resumption Requirements

6.1 Suspension Criteria

- **System Crashes or Hangs:**
    - The Social Butterfly application crashes or hangs during execution, preventing further testing activities.
    - Multiple instances of system instability that interrupt the testing process.

- **Critical Functionality is Non-Operational:**
    - Very important features such as data retrieval, data storage, data analysis, or reporting components are not functioning as expected.
    - User interface components necessary for navigation or important operations are broken or unresponsive.

- **API Rate Limits Reached:**
    - API rate limits for Twitter (X) or Reddit are reached, preventing data retrieval and stopping further testing.
    - Issues with API authentication or authorization that block access to necessary data.

- **Security Vulnerabilities:**
    - Discovery of significant security vulnerabilities that could compromise user data or system integrity.
    - Unauthorized access or data breaches detected during security testing.

- **Network Issues:**
    - Persistent network connectivity issues affecting interactions with external APIs or internal components.

- Network outages or disruptions that interrupt testing activities.

6.2 Resumption Requirements

- **Resolution of the Blocking Issue:**
  - The development team must address and resolve the important issues identified during testing.
  - Implement appropriate fixes or workarounds to restore system functionality.

- **Verification of the Fix by the Development Team:**
  - The development team should verify the implemented fixes to ensure that the critical issues have been resolved.
  - Do initial testing to confirm the effectiveness of the fixes.

- **Confirmation from the Test Manager to Proceed with Testing:**
  - The test manager needs to review and approve the resolution of critical issues and the verification results.
  - Ensure that all blocking issues have been addressed and documented and fixed.

- **Regression Testing:**
  - Do regression testing to ensure that the fixes have not introduced new defects or issues in other parts of the system.
  - Very thorough testing of related functionalities to maintain overall system integrity.

- **Update Test Documentation:**
  - Update the test plan, test cases, and defect reports to reflect the resolution of critical issues and any changes made.
  - Document any new test scenarios or additional test cases required based on the fixed issues.

- **Stakeholder Communication:**
  - Communicate the resolution status and readiness to resume testing to all relevant stakeholders, including the project manager and development team.
  - Provide a summary of the actions taken and any remaining risks or concerns.

- **Monitoring and Contingency Planning:**
    - Establish monitoring mechanisms to detect any reoccurrence of the issues.
    - Develop contingency plans for quick response in case similar issues arise during subsequent testing phases.

7.0 Test Deliverables

7.1 Test Plan Document

- **Overview:**
    - A comprehensive document outlining the overall testing strategy, objectives, scope, and approach for the Social Butterfly project.
- **Sections:**
    - Introduction, test items, features to be tested, features not to be tested, testing approach, pass/fail criteria, suspension and resumption criteria, test deliverables, testing tasks, environmental needs, responsibilities, staffing and training needs, schedule, risks and contingencies, and approvals.

7.2 Test Cases and Test Scripts

- **Detailed Test Cases:**
    - Test cases for each feature and component, including:
        - Test case ID, description, preconditions, steps, expected results, and postconditions.
        - Functional test cases, performance test cases, security test cases, and usability test cases.
- **Automated Test Scripts:**
    - Scripts developed for automating repetitive and regression testing tasks using tools like Selenium and JMeter.
- **Review and Approval:**
    - Test cases and scripts reviewed and approved by stakeholders to ensure accuracy and completeness.

7.3 Test Execution Reports

- **Daily/Weekly Test Execution Reports:**
    - Reports generated during the testing phase detailing the execution of test cases, including:
        - Number of test cases executed, passed, failed, and blocked.
        - Defects identified, their severity, and current status.

        - Test coverage and progress against the planned schedule.
- **Final Test Execution Report:**
    - A comprehensive report summarizing the entire test execution phase, including:
        - Overall test execution status, major findings, and key metrics.
        - Defects summary and their impact on the project.
        - Recommendations for improvement and next steps.

7.4 Defect Reports

- **Defect Logging:**
    - Detailed documentation of defects found during testing, including:
        - Defect ID, description, severity, steps to reproduce, expected results, and actual results.
        - Screenshots, logs, and other relevant attachments to support defect reproduction.
- **Defect Tracking:**
    - Continuous tracking and updating of defect status, including:
        - New, open, in-progress, resolved, verified, and closed statuses.
        - Assigning defects to responsible developers and tracking their resolution.
- **Defect Metrics:**
    - Metrics and analysis of defects, including:
        - Defect density, defect discovery rate, defect closure rate, and defect age.
        - Trends and patterns in defect occurrences and their impact on the project.

7.5 Final Test Summary Report

- **Test Summary:**
    - A high-level summary of all testing activities, including:
        - Objectives, scope, and approach of the testing.
        - Test coverage, executed test cases, and test results.
- **Defects Summary:**
    - A detailed summary of all defects identified during testing, including their severity, status, and impact.
- **Key Metrics:**
    - Key performance indicators and metrics to provide a quantitative assessment of the testing process, such as:
        - Test case execution rate, pass/fail rate, defect discovery rate, and defect closure rate.
- **Risk Assessment:**
    - An assessment of any remaining risks and their potential impact on the project.
- **Lessons Learned:**

    - Documentation of lessons learned, best practices, and areas for improvement identified during the testing process.
- **Stakeholder Review:**
    - Presentation of the final test summary report to stakeholders for review and approval, including recommendations for next steps and final sign-off.

8.0 Testing Tasks

8.1 Develop Test Plan and Test Cases

- **Requirement Analysis:**
    - Gather and analyze requirements to understand the system's functionality and identify key areas for testing.
- **Test Plan Creation:**
    - Create a comprehensive test plan outlining the testing scope, objectives, approach, resources, schedule, and deliverables.
- **Test Case Design:**
    - Develop detailed test cases for each identified feature and component, specifying preconditions, steps, expected results, and postconditions.
- **Review and Approval:**
    - Review the test plan and test cases with stakeholders to ensure completeness and accuracy, and obtain necessary approvals.

8.2 Setup Test Environment

- **Environment Configuration:**
    - Prepare and configure the hardware, software, and network components required for testing.
- **Test Data Preparation:**
    - Create or procure the necessary test data, ensuring it covers various scenarios and edge cases.
- **Environment Validation:**
    - Validate the test environment setup to ensure it accurately mirrors the production environment and is ready for testing activities.
- **Access Management:**
    - Ensure testers have the necessary access permissions to all required systems and tools.

8.3 Execute Test Cases

- **Test Execution:**
    - Execute the test cases according to the test plan, following the specified steps and recording the results.
- **Defect Identification:**
    - Identify and document any defects or discrepancies between expected and actual results.
- **Test Case Prioritization:**
    - Prioritize test cases based on criticality and impact on the overall system, focusing on high-risk areas first.

8.4 Log and Track Defects

- **Defect Reporting:**
    - Log identified defects in a defect tracking system, providing detailed information including steps to reproduce, severity, and impact.
- **Defect Triage:**
    - Triage and prioritize defects based on their severity and impact on the system.
- **Defect Assignment:**
    - Assign defects to the appropriate developers for resolution.
- **Defect Tracking:**
    - Track the progress of defect resolution, ensuring timely fixes and retesting of resolved issues.

8.5 Retest Resolved Defects

- **Defect Verification:**
    - Verify that defects reported as fixed by developers are resolved correctly without introducing new issues.
- **Regression Testing:**
    - Conduct regression testing to ensure that fixes have not adversely affected other parts of the system.
- **Continuous Monitoring:**
    - Continuously monitor the system for any new issues that may arise during the defect resolution process.

8.6 Prepare Test Summary Report

- **Test Execution Summary:**
    - Compile a summary of the test execution activities, including the number of test cases executed, passed, failed, and blocked.
- **Defect Summary:**

    - Provide a detailed summary of the defects identified, their status, severity, and impact.
- **Metrics and KPIs:**
    - Include key metrics and performance indicators to provide a quantitative assessment of the testing process.
- **Lessons Learned:**
    - Document any lessons learned, best practices, and areas for improvement identified during the testing process.
- **Stakeholder Review:**
    - Present the test summary report to stakeholders, highlighting key findings and obtaining their feedback and approval.

8.7 Continuous Improvement

- **Review and Feedback:**
    - Conduct review sessions with the testing team and stakeholders to gather feedback on the testing process.
- **Process Improvement:**
    - Identify areas for improvement and implement changes to enhance the efficiency and effectiveness of future testing efforts.
- **Training and Development:**
    - Provide training and development opportunities for the testing team to keep them updated with the latest testing tools and methodologies.

9.0 Environmental Needs

- **Hardware:**
    - Servers:
        - High-performance servers to host the application, database, and testing environments.
        - Backup servers to ensure data redundancy and availability during testing.
        - Load balancers to distribute the load during performance testing.

    - Client Machines for Testers:
        - Laptops or desktops with sufficient processing power and memory to run test scripts and tools.

        - Backup systems to ensure continuity in case of hardware failure.

    - Storage:
        - Sufficient storage capacity for storing test data, logs, and reports.
        - Backup storage solutions to prevent data loss.

- **Software:**
    - Operating Systems:
        - Compatible operating systems for servers and client machines.
        - Virtualization software for creating isolated test environments.

    - Browsers:
        - Latest versions of popular web browsers (e.g., Chrome, Firefox, Safari, Edge) for cross-browser testing.
        - Browser automation tools

    - Database Management Systems:
        - SQL-based databases for storing and retrieving test data.
        - NoSQL databases for testing unstructured data storage.

    - Testing Tools:
        - Functional Testing: Selenium, TestComplete.
        - Performance Testing: JMeter, LoadRunner.
        - Security Testing: OWASP ZAP, Burp Suite.
        - Usability Testing: UserTesting, Lookback.

    - Libraries:
        - Tweepy: A Python library for accessing the Twitter API.
        - PRAW (Python Reddit API Wrapper): A Python library for accessing the Reddit API.
        - Pandas: A Python library for data manipulation and analysis.
        - Matplotlib: A Python library for data visualization.

- **Network:**
    - Stable Internet Connection:
        - High-speed internet connection to ensure seamless API interactions and data retrieval.
        - Backup internet connection to minimize downtime during network issues.

- Network Configuration:
  - Secure network configurations to protect against unauthorized access and data breaches.
  - Firewalls and VPNs for secure communication between servers and client machines.

- **Test Data:**
  - Sample Data:
    - Mock social media posts from Twitter and Reddit for data retrieval testing.
    - User information data for testing user authentication and authorization.
    - Structured datasets for testing data storage and analysis components.
    - Test data for sentiment analysis, including labeled positive, negative, and neutral posts.
  - Data Generation Tools:
    - Tools for generating synthetic test data.
    - Scripts for creating and populating test databases with sample data.
  - Data Privacy:
    - Anonymized data sets to ensure compliance with data privacy regulations.

- **Additional Infrastructure needs:**
  - Testing Environment:
    - Dedicated test environments that mirror the production setup to ensure accurate testing results.
    - Continuous Integration/Continuous Deployment (CI/CD) pipelines for automated testing and deployment.
  - Monitoring and Logging:
    - Monitoring tools to track system performance and identify issues during testing.
    - Logging solutions to capture and analyze logs for debugging and issue resolution.
  - Backup and Recovery:
    - Regular backups of test environments and data to ensure quick recovery in case of failures.
    - Disaster recovery plans to handle major disruptions during testing.

10.0 Responsibilities

- **Test Manager:** The Test Manager is responsible for overseeing the entire testing process to ensure that all testing activities are completed on time and meet the quality standards of the project. Specific responsibilities include:
    - **Planning:** Develop the overall test strategy and plan, including defining the scope, objectives, resources, and schedule.
    - **Team Management:** Manage the testing team, including assigning tasks, monitoring progress, and providing guidance and support.
    - **Coordination:** Act as a liaison between the testing team, development team, project management, and other stakeholders to ensure clear communication and coordination.
    - **Quality Assurance:** Ensure that all testing activities adhere to the test plan and that testing is conducted according to industry best practices.
    - **Issue Resolution:** Identify, track, and resolve any issues that arise during the testing process, including resource constraints, technical challenges, and schedule delays.
    - **Reporting:** Provide regular updates to stakeholders on the status of testing activities, including test execution progress, defect status, and risk assessments.
    - **Final Validation:** Oversee the final validation and acceptance of the system before release, ensuring that all critical issues have been resolved.
- **Testers:** Execute test cases, log defects, retest resolved defects, and report progress to the test manager.Specific responsibilities include:
    - **Test Case Execution:** Execute test cases according to the test plan, including functional, performance, security, and usability tests.
    - **Defect Logging:** Identify and document any defects found during testing, including detailed steps to reproduce the issue, severity, and impact.

    - **Defect Tracking:** Track the status of defects from discovery through resolution, ensuring that all critical defects are addressed in a timely manner.

    - **Retesting:** Verify that defects have been fixed by retesting the relevant functionality and ensuring that no new issues have been introduced.

    - **Documentation:** Maintain detailed records of test execution, including test results, defect logs, and any other relevant documentation.

    - **Reporting:** Provide regular updates to the Test Manager on the status of testing activities, including any issues encountered and progress towards completion.

- **Developers:** Fix defects reported by the testing team and provide technical support during testing. Specific responsibilities include:

    - **Defect Resolution:** Analyze and fix defects reported by the testing team, ensuring that the underlying issues are resolved without introducing new problems.

    - **Code Quality:** Ensure that code changes adhere to the project's coding standards and do not negatively impact other areas of the system.

    - **Technical Support:** Provide technical support to the testing team, including answering questions, clarifying requirements, and assisting with environment setup.

    - **Integration:** Ensure that any code changes are properly integrated into the system and do not disrupt existing functionality.

    - **Verification:** Conduct unit tests and other forms of verification to ensure that code changes are correct and complete before handing off to the testing team for further validation.

- **Project Manager:** The Project Manager is responsible for ensuring the overall success of the project by coordinating between the testing and development teams, managing project timelines, and addressing any resource issues. Specific responsibilities include:

- **Project Coordination:** Facilitate communication and coordination between the testing and development teams, ensuring that everyone is aligned on project goals and timelines.

- **Resource Management:** Identify and address any resource constraints, including staffing, hardware, and software needs, to ensure that the testing and development teams have what they need to succeed.
- **Timeline Management:** Develop and manage the project schedule, including setting milestones, tracking progress, and adjusting timelines as necessary to accommodate changes or delays.

- **Risk Management:** Identify potential risks to the project, including technical, schedule, and resource risks, and develop mitigation strategies to address them.

- **Stakeholder Communication:** Provide regular updates to stakeholders on the status of the project, including progress, risks, and any issues that need to be addressed.

- **Issue Resolution:** Address any issues that arise during the project, including conflicts between teams, schedule delays, and resource constraints, to ensure that the project stays on track.

11.0 Staffing and Training Needs

- **Staffing:** Ensure sufficient staffing with experienced testers who have a good understanding of the Social Butterfly project and the technologies involved.

- **Training:** Provide training sessions on the Social Butterfly project, testing tools, and the Tweepy and PRAW libraries. Ensure testers are familiar with the testing environment and processes.

12.0 Appendices

- **Appendix A:** Glossary of Terms
  - **API:** Application Programming Interface
  - **Twitter (X):** A social media platform for microblogging
  - **Reddit:** A social media platform and forum for community discussions
  - **Tweepy:** A Python library for accessing the Twitter API

    - **PRAW:** Python Reddit API Wrapper, a Python library for accessing the Reddit API
    - **SDD:** Software Design Document (Software Design)

- **Appendix B:** References
    - Twitter API Documentation
    - Reddit API Documentation
    - Tweepy Documentation
    - PRAW Documentation