

Gamified Food Tracker - GFT

Revision History

Date	Version	Author	Description
27/02/2024	0.0.1	Gnox Team	First Draft of test Strategy

Table of Contents

- 1. Scope
- 2. Test Approach
- 3. Test Environment
- 4. Testing Tools
- 5. Release Control
- 6. Risk Analysis
- 7. Review and Approvals

1. Scope

It defines parameters like

- Who will review the document?
 - Kim & Steffen (Project Leader, Senior Developer)
- Who will approve this document?
 - Kim & Steffen (Project Leader, CEO)
- Testing activities carried out with timelines
 - Unit Testing - Automated testing run everytime a new feature is implemented.
 - Integration Testing - Automated testing run everytime a new component is added, or updated.
 - System Testing - manual testing run when the system is ready to be tested.
 - UI/Acceptance Testing - Manual testing run when we have a UI Prototype, major updates to the prototype and when we have run system tests.

2. Test Approach

It defines

- Process of testing
 - Unit Testing - We will develop our methods and classes using a TDD approach, this will ensure that we will use unit testing on all our code with an almost 100% code coverage.
 - Integration Testing - Integration Testing will be implemented when more than 1 component is required. The tests will cover all the tests needed to make sure we have a complete and functioning integration of the components.
 - System Testing - System Testing will be implemented when the system is functional, it will cover most of the basic functions of the system.
 - UI/Acceptance Testing - UI/Acceptance Testing will be implemented when the prototype is launched, and after major updates, it will also be conducted as the project finishes.
- Testing levels
 - Unit Testing.
 - Integration Testing.
 - System Testing.
 - UI/Acceptance Testing.
- Roles and responsibilities of each team member
 - Project Manager - Responsible for overall project planning, coordination, and delivery.
 - Scrum Master (Agile) - Facilitates Agile ceremonies (e.g., sprint planning, daily stand-ups, retrospectives).
 - Developers (Backend, Frontend, Full-stack) - Write code based on specifications.
 - UI/UX Designer - Designs the user interface and user experience.
- Types of Testing (Load testing, Security testing, Performance testing etc.)
 - Regression Testing - Verifies that new code changes do not negatively impact existing functionalities.
 - Exploratory Testing - Testers explore the software to discover defects without predefined test cases.
 - Performance Testing - Evaluates the software's responsiveness, scalability, and stability under different load conditions.
- Testing approach & automation tool if applicable
 - TDD
 - GitHub Actions
 - JUnit

3. Test Environment

- Define number of requirement and setup required for each environment
 - Development Environment:
 - Purpose: To allow developers to build and test the software.
 - Requirements:
 - Integrated development environment (IntelliJ).
 - Version control system (Github).
 - Build tools (Maven).
 - Development libraries and frameworks.
 - Test data for unit testing (Testcontainers).
 - Testing Environment:
 - Purpose: To conduct various levels of testing (unit, integration, system, etc.).
 - Requirements:
 - JUnit.
 - Access to a testing database (PostGress TestContainer).
 - Automated testing tools (PostGress TestContainer).
 - Staging Environment:
 - Production Environment:
- Define backup of test data and restore strategy
 - Backup Storage - Cloud storage, with backups every 7 days, in case of corrupted data we go back to the latest backup.

4. Testing Tools

- Automation and Test management tools needed for test execution
 - JUnit
 - GitHub Actions.
- Figure out number of open-source as well as commercial tools required, and determine how many users are supported on it and plan accordingly
 - JUnit - free open source for automation of tests.

- TestRail: Typically priced based on the number of users. A free trial is available for a limited number of users, for test management tools
- Budget Constraints - as we have a budget of 0, we will have to use only free tools, this however does mean that we don't have any restrictions on what user can use what tools.
- using JUnit is good for scalability, it is not hard to scale a program using JUnit

5. Release Control

- Release management plan with appropriate version history that will make sure test execution for all modification in that release
 - Run all tests, and release the app through app distributors like app store and google play

6. Risk Analysis

- List all risks that you can estimate
 - Market Changes: Shifts in market trends or user preferences may affect the relevance of the software.
 - Natural Disasters: Unexpected events such as earthquakes, floods, or other disasters can disrupt project timelines.
 - Sickness in the team: Sickness in the team can lead to delays in the programming
 - War: war can completely stop the programming process.
 - Resource Constraints: Insufficient resources, such as manpower, tools, or technology, may impact project progress.
 - malicious attacks: viruses and hacking can ruin the program
 - Unexpected pregnancies: pregnancies can result in a employee being away from the project for a longer period.
- Give a clear plan to mitigate the risks also a contingency plan
 - Stay up to date on current market trends and try to incorporate it into our project, to secure relevancy
 - Have backup server in different country

- for short period of sickness, we will have to have room in our deadline to be able to handle it, for longer periods of sickness get a substitute programmer
- In case of War we have a backup of the program in another country, we may need to put the programming process on hold while we fight for our country and people!
- For malicious attacks we will have a backup that we can roll back to
- for pregnancies we will get a substitute

7. Review and Approvals

- All these activities are reviewed and sign off by the business team, project management, development team, etc.
- Summary of review changes should be traced at the beginning of the document along with approved date, name, and comment