SE 4485: Software Engineering Project

Spring 2024

Test Plan

Group Number	3
Project Title	Storybook POC Continuation with Chromatic and Storybook GPT
Sponsoring Company	ARGO
Sponsor(s)	Ponchai Reainthong, Kevin Roa, Raisa Gonzalez
Group Members	1. Lillie McMaster 2. Lauren Nguyenphu 3. Alina Khan 4. Hamdiya Abdulhafiz 5. Iman Sheriff 6. Timothy Naumov

Abstract

This document outlines the test plan for the Storybook POC Continuation project. It details test cases for both UI and system testing, encompassing both the functional and nonfunctional requirements established for this project. The document discusses techniques for test generation as well as criteria used to measure the quality of tests. Additionally, it established traceability between the test cases and the system use cases which were developed previously. Overall, the document provides a comprehensive overview of the measures taken to ensure the efficacy of the Storybook POC Continuation project, as well the engineering standards and references for comprehensive testing.

Table Of Contents

List Of Figures

List Of Tables

Introduction

This document outlines the testing plan for the Storybook POC Continuation project. which is a Proof of Concept for the ARGO team to confirm the potential benefits of using Storybook GPT as a beneficial investment for ARGO. The document's purpose is to outline the test cases to reveal any unintended side effects or bugs within the system.

This document will cover the following

1. UI Testing for the newly introduced components. These test cases ensure the components both display and function as intended.
2. The system test cases used to test that the functional and nonfunctional requirements outlined for this project.
3. The techniques that were used to generate tests and test cases as well as criteria that were used to measure the quality of the test.

A full traceability displaying how the test cases are tied to the use cases for the system is displayed, followed by standards and resources referenced during the creation of this document. For a full breakdown of this document's structure, please see the table of contents.

Requirements/Specifications-Based System Level Test Cases

UI Test Cases for Newly Introduced Components

ID	Component	Test Cases
CD1	Text Fields	a. Label displays label string. b. Context works as a text box by allowing input on click. c. Helper text displays helper string. d. Calendar icon displays when enabled.
CD2	Drop Down	a. Label moves on click. b. Drop down appears on click. c. Drop down has correct and entire contents. d. Color variants are accurate .
CD3	Paper	a. Paper displays components. b. Elevation Toggle works as values are changed. c. Sample paper displays according to controls.
CD4	Slider	a. Slider moves and actions are recorded. b. Disabled toggles correctly. c. Slider label displays a label string.
CD5	Pagination	a. Page changes on Click b. Changing boundary count reflects the correct boundary for variant page numbers. c. Changing count control reflects accurate page numbers. d. Default Page renders accurately on page refresh.
CD6	Pagination V2	a. Page changes on click

		b. Star disables when no longer applicable. c. Page numbers are sticky when clicked. d. Changing boundary count reflects the correct boundary for variant page numbers.
CD7	ARGO Button	a. Button works on click. b. Variant buttons work on click. c. Disable button does not work on click.

Table 1. UI Test Cases for Newly Introduced Components

System Test Cases

ID	Test Case
TC1	Figma designs appear under a component's design tab in Storybook
TC2	For each view change, ensure documentation is updated.
TC3	Newly created components appear in Storybook.
TC4	Prompt Storybook GPT for creation of new story based off parameters generates a new component and is reviewed to be successful by developer.
TC5	Make a new commit and confirm that version number is correct and the package is built on npm ARGO UX Master Component Library II.
TC6.1	Project is built and deployed to Chromatic after a pull request is merged.
TC6.2	Chromatic Regression Testing is approved by developer and appears in Library.
TC7	Component Library is successfully installed.
TC8	Figma design displayed in design tab matches component in Storybook

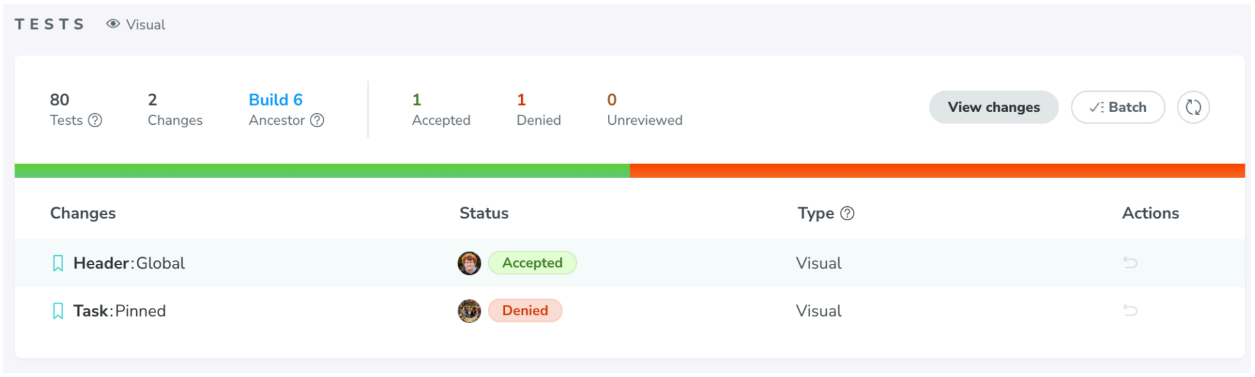
Table 2. System Test Cases

Techniques For Test Generation

Three types of test generation techniques were employed: GUI Testing, Regression Testing, and Usability testing.

GUI testing is used on icons, buttons, menus, textboxes, and all other UI components to ensure they function correctly. This is a form of black-box testing.While mostly manual work in the beginning to test all implemented components thoroughly, GUI testing on a day to day basis will be done only per component. At deployment, all components are tested once with GUI manual testing. As components are added like in TC3, UC3, only one component is manually tested and is done in less than 10 minutes. Our GUI testing is used in TC1, TC3, TC4, and TC8. We will used a successful/unsuccessful measurement on the GUI testing and our goal is 95% successful.

Regression testing is the main benefit of our Chromatic integration into our system. This is a form of black box testing. The regression testing for Chromatic allows quick and easy comparison of changes that have been made to every build sent through the main branch into the Chromatic system. As a build is finished, a developer will use Chromatics built in regression changes to review the impact of the component differences and approve/deny as needed. Regression testing through Chromatic is used in TC2, TC4, TC6.1, and TC6.2. Regression testing is measured by an approval/denial system built into Chromatic.



80 Tests	2 Changes	Build 6 Ancestor	1 Accepted	1 Denied	0 Unreviewed

Changes	Status	Type	Actions
Header:Global	Accepted	Visual	
Task:Pinned	Denied	Visual	

Figure 1. Chromatic Regression Testing.
Usability and acceptance testing were conducted by observing real users, like our developers, as they attempt to complete tasks on it. This version of white box testing ensures continuous improvement of code and development practices within the team, increasing opportunities for collaborative code. The test cases that use acceptance testing are TC5 and TC7.

Traceability Of Test Cases To Use Cases

Test Case	System Use Case
TC1	UC1: Designs Components
TC2	UC2: Views Documentation
TC3	UC3: Implement Components
TC4	UC4: Generate COmponent Stories
TC5	UC5: Publishing Components
TC6	UC6: Build and Deploy Storybook Documentation
TC7	UC7: Install Component Library
TC8	UC8: Test Components from Figma

Table 3. Traceability Of Test Cases To Use Cases
Each test case was created to meet the exit case specified in the corresponding Use Case. For example, Text Case 1 was built off the Use Case 1 exit condition "The components have been designed and finalized."

Evidence The Test Plan Has Been Placed Under Configuration Management

Version In	Version Out	Changes	Reviewed By	Notion Task Numbers
n/a	0.0	Document Creation based on Template	Lillie McMaster and Alina Khan	UG3-101
0.0	1.0	Requirements/Specification based system level test cases, Techniques for test generation, traceability of test cases to use cases.	All Group Members	UG3-102
1.0	2.0	Feedback Provided by ARGO Team	ARGO Team	UG3-103

Table 4. Evidence The Test Plan Has Been Placed Under Configuration Management

Engineering Standards And Multiple Constraints

- IEEE Std 829-1983: Software Testing [pdf]
- ISO/IEC/IEEE Std 29119-1-(Revision-2022): Part 1 - Software Testing General Concepts [pdf]
- ISO/IEC/IEEE Std 29119-2-(Revision-2021): Part 2 - Test Process [pdf]
- ISO/IEC/IEEE Std 29119-3-(Revision-2021): Part 3 - Test Documentation [pdf]
- ISO/IEC/IEEE Std 29119-4-(Revision-2021): Part 4 - Test Techniques [pdf]

Additional References

- GfG. “Software Testing Techniques.” GeeksforGeeks, GeeksforGeeks, 6 Dec. 2023, www.geeksforgeeks.org/software-testing-techniques/.
- “Ui Testing: What It Is and How to Do It.” Hotjar, www.hotjar.com/ui-design/testing/. Accessed 13 Apr. 2024.
- “Usability Testing: What It Is, Benefits, and What It Isn’t.” Hotjar, www.hotjar.com/usability-testing/. Accessed 13 Apr. 2024.
- Jorgensen, P.C., 2013. Software Testing: A Craftsman's Approach. Auerbach Publications
- Mathur, A.P., 2013. Foundations of Software Testing, 2/e. Pearson Education