

# SE 4485: Software Engineering Project

Spring 2024

## Requirements Documentation

Group Number	3
Project Title	Storybook POC Continuation with Chromatic and Storybook GPT
Sponsoring Company	ARGO
Sponsor(s)	Ponchai Reainthong, Kevin Roa, Raisa Gonzalez, Mark Bentson
Group Members	<ol style="list-style-type: none"><li>1. Lillie McMaster</li><li>2. Lauren Nguyenphu</li><li>3. Alina Khan</li><li>4. Hamdiya Abdulhafiz</li><li>5. Iman Sheriff</li><li>6. Timothy Naumov</li></ol>

## ABSTRACT

This document presents the requirements documentation for the Storybook POC project sponsored by ARGO for group 3 at The University of Texas at Dallas Software Engineering Project. The document includes a use case model, which outlines functional requirements. It continues to detail functional requirements, including: Design Components, Views Documentation, Implement Components, Generate Component Stories, Publishing Components, Build and Deploy Storybook Docs, Install Component Library, and Test Components from Figma. Non-Functional requirements between the UTD-ARGO-II Storybook Repository and the ARGO Storybook GPT are detailed in categories of Usability, Maintainability, Extensibility, and Scalability.

## TABLE OF CONTENTS

<b>Introduction</b>	<b>4</b>
<b>Use Case Model For Functional Requirements</b>	<b>4</b>
Graphic Use Case Model	4
UC1: Designs Components	5
UC2: Views Documentation	5
UC3: Implement Components	6
UC4: Generate Component Stories	6
UC5: Publishing Components	7
UC6: Build and Deploy Storybook Docs	7
UC7: Install Component Library	8
UC8: Test Components from Figma	8
<b>Rationale For Use Case Model</b>	<b>9</b>
<b>Non-Functional Requirements</b>	<b>9</b>
Usability	9
Maintainability	9
Extensibility	9
Scalability	10
<b>Evidence The Document Has Been Placed Under Configuration Management</b>	<b>10</b>
<b>Engineering Standards And Multiple Constraints</b>	<b>10</b>
<b>Additional References</b>	<b>10</b>

## LIST OF FIGURES

Figure 1. Use Case Model for Functional Requirements _____	4
--	---

## LIST OF TABLES

Table 1. UC1: Design Components _____	5
Table 2. UC2: Views Documentation _____	5
Table 3. UC3: Implement Components _____	5
Table 4. UC4: Generate Component Stories _____	6
Table 5. UC5: Publishing Components _____	6
Table 6. UC6: Build and Deploy Storybook Docs _____	7
Table 7. UC7: Install Component Library _____	7
Table 8. UC8: Test Components from Figma _____	8
Table 9. Evidence The Document Has Been Placed Under Configuration Management _____	10

## Introduction

This document serves as the requirements for the Storybook POC Continuation project. This project serves as a Proof of Concept for the ARGO team to confirm that Storybook GPT would be a beneficial investment for ARGO to continue to make. The document's purpose is to outline functional requirements and non-functional requirements of the project.

The scope of the requirements is to be equivalent to the project's scope, ending May 6th. The product is a continuation of creating a GPT Component library for Storybook and Figma components. The project scope ends with a library of stories, created with GPT and with the company issues in mind, as a Proof of Concept for the use of Storybook GPT in ARGO. Scenario for using the product may look like: a user uses the created GPT to create Storybook Stories/Components, which are components of the Storybook application.

The structure of the document can be noticed within the table of contents. It follows the suggested structure given by the Professor for the course.

## Use Case Model For Functional Requirements

### Graphic Use Case Model

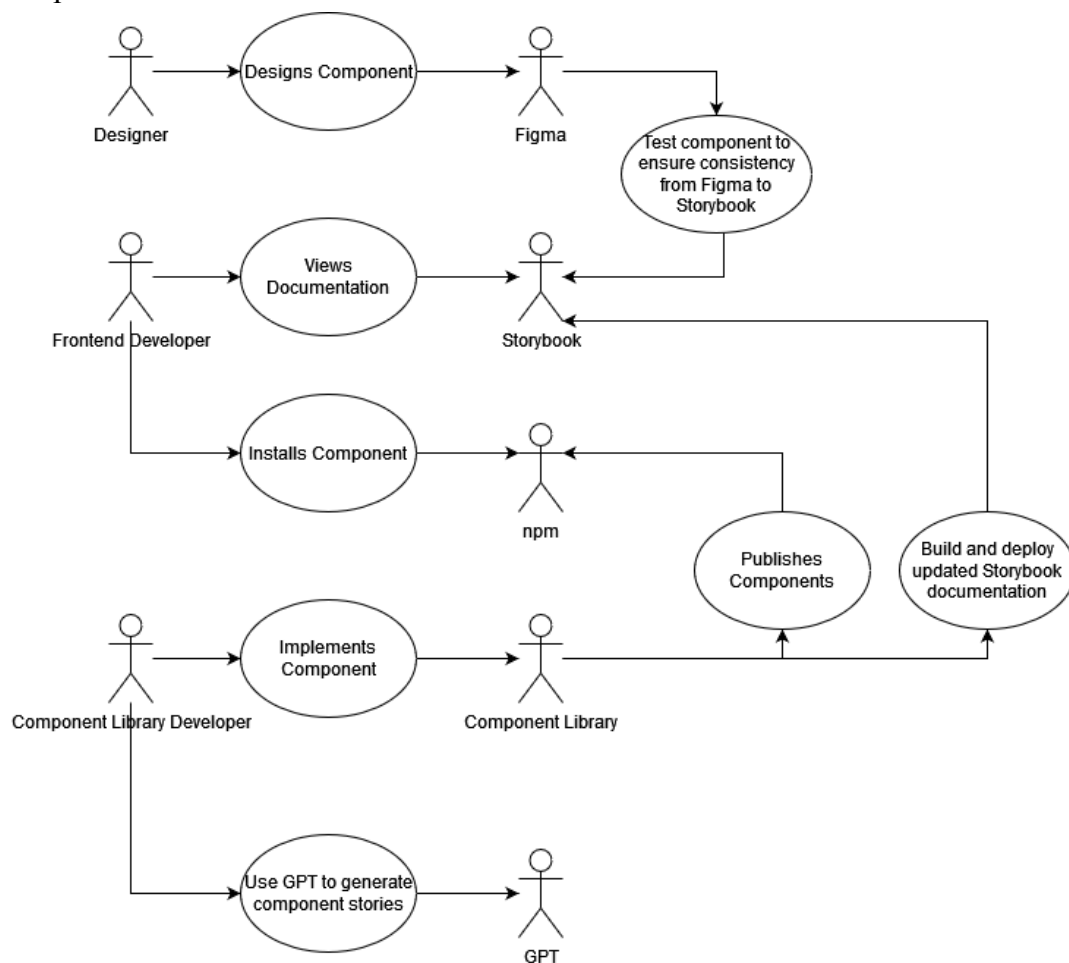


Figure 1. Use Case Model for Functional Requirements

## Textual Description Per Use Case

UC1: Designs Components	
Participating Actors	Designer, Figma
Entry Condition(s)	A list of components intended for use are created.
Normal Flow of Event	Designer reviews the list of components intended to be designed. Designer designs and customizes the components using figma. Designer reviews, and finalizes designed components.
Exit Condition(s)	The components have been designed and finalized.
Exceptions (Alternate Flow of Events)	The designer has to go back and design another component because one is missing.
Special Requirements	All designed components must be easy to use, and its functionality must align with users expectations and behaviors.

UC2: Views Documentation	
Participating Actors	Frontend Developer, Storybook
Entry Condition(s)	Documentation is created.
Normal Flow of Event	Frontend Developer opens up documentation and is able to view it. The front-end developer reviews the documentation and it's approved.
Exit Condition(s)	Documentation is viewed, and approved.
Exceptions (Alternate Flow of Events)	There are changes that need to be made in the documentation.
Special Requirements	Documentation must follow necessary standards and guidelines.

UC3: Implement Components	
Participating Actors	Component Library Developer, Component Library
Entry Condition(s)	Component design is finalized and the Component Library Developer is assigned to implement the component.
Normal Flow of Event	<p>Component Library Developer will review the design specifications and write the code for the component.</p> <p>The code is tested to ensure components are compatible and adjusted as needed.</p> <p>Developer will commit the code to the Component Library.</p> <p>A pull request is created for the component.</p>
Exit Condition(s)	Component is successfully implemented into the Component Library.
Exceptions (Alternate Flow of Events)	If the component does not meet specifications, it is reworked until it passes all tests and requirements.
Special Requirements	Component implementation must be documented and follow coding standards and guidelines

UC4: Generate Component Stories	
Participating Actors	Component Library Developer, StorybookGPT
Entry Condition(s)	<p>Component has been implemented and is available in the component library.</p> <p>The component library developer is ready to create documentation for the component.</p>
Normal Flow of Event	<p>The Component Library Developer prepares the input parameters for the component stories and inputs them into StorybookGPT.</p> <p>StorybookGPT will generate stories based off of the parameters.</p> <p>The developer will review and approve the generated stories, which will then be added to Storybook for the component.</p>
Exit Condition(s)	Component stories are generated, reviewed, and integrated into Storybook.
Exceptions (Alternate Flow of Events)	If the stories are insufficient, the Developer may need to refine inputs to regenerate stories.
Special Requirements	The input to StorybookGPT must be detailed to ensure accuracy and quality of the generated stories.

	The component library developer must validate the generated stories to match the intended function and design.
--	--

UC5: Publishing Components	
Participating Actors	Component Library, npm
Entry Condition(s)	New components are implemented
Normal Flow of Events	When new components are implemented, a new pull request is created on GitHub. After reviewing the changes, the pull request will be merged and a GitHub action will be triggered. This action will determine the next package version number from the commit history and publish the built package to the npm registry.
Exit Condition(s)	The package has been successfully deployed to the npm registry with the updated version number and newly implemented components built in.
Exceptions (Alternate Flow of Events)	Component library developers can manually trigger the release process if necessary.
Special Requirements	A new package number has to be set otherwise duplicate package version number errors will occur in the release process.

UC6: Build and Deploy Storybook Docs	
Participating Actors	Component Library, Chromatic, Storybook
Entry Condition(s)	New components are implemented
Normal Flow of Events	When a pull request is merged with new components implemented, another GitHub action is triggered which builds the new Storybook documentation and deploys it to Chromatic.
Exit Condition(s)	The updated Storybook documentation is built and deployed to Chromatic.
Exceptions (Alternate Flow of Events)	Developers can build local development storybook servers for others to see components that are in progress.
Special Requirements	Chromatic has to pass regression tests with newly implemented components as well as verifying that components match designed components in figma.

UC7: Install Component Library	
Participating Actors	Component libraries
Entry Condition(s)	Package is published
Normal Flow of Event	User opens the system and navigates to the component library installation page. Component library is selected to install. Prompts necessary configuration or customizations to be applied and the system downloads and installs the selected component library. The system confirms the installation of component libraries.
Exit Condition(s)	Selected component libraries are installed.
Exceptions (Alternate Flow of Events)	If the selected component library is not available or cannot be installed, the system displays an error message and does not proceed with installation.
Special Requirements	System must have necessary permissions to install component libraries.

UC8: Test Components from Figma	
Participating Actors	Figma, test components
Entry Condition(s)	Figma project open with test components created
Normal Flow of Event	A test is initiated for the component in Figma for the selected component. Figma runs the test and verifies the component's functionality and also provides the test results. The results from Figma get reviewed.
Exit Condition(s)	Test results are provided from Figma
Exceptions (Alternate Flow of Events)	If a test component is not found in the Figma project, notification is given that the testing process is halted. If a test fails, Figma provides specific information about the failure.
Special Requirements	Figma project with test components created.



## Rationale For Use Case Model

The use case model consists of several actors who are interacting with the different entities within the system. We separated developers into front end developers and component library developers to isolate how component library developers implement and document new components for front end developers to consume. While the component library developers largely contribute to the component library with the help of GPT, their work is propagated through the system after the contributions are built and released to the storybook deployment on Chromatic and npm. From there, front end developers are able to install package updates from npm and view the live storybook documentation for changes to components. When new storybook deployments are released, Chromatic can be used to test backwards compatibility with previous versions of the package while also ensuring that component stories adhere to design specifications from Figma.

## Non-Functional Requirements

In the Non-functional requirements, we are creating two separate systems that will work hand in hand. “The System” refers to the UTD-ARGO-II Storybook repository we are improving from the first half of the UTD-ARGO Spring 2023 project. “The GPT system” refers to the ARGO Storybook GPT we are producing using ChatGPT 4.0.

### Usability

- A. The system shall provide an interface for creating, viewing, modifying, and deleting components.
- B. The system shall provide the ability to use its components within other ARGO projects.
- C. The GPT system shall provide prompts to streamline creating new stories.
- D. The GPT system shall provide the ability to document each component deployed.

### Maintainability

- A. The system shall ensure that components will be compatible with future version upgrades.
- B. The system shall house all components within the same repository.
- C. The system shall be built using open source libraries.
- D. The system shall automatically deploy new changes to the library using Conventional Commits.
- E. The GPT system shall be built to ensure usage of proper design standards.
- F. The GPT system shall utilize the latest stable version of ChatGPT.

### Extensibility

- A. The system shall be able to accept the creation and modification of components.
- B. The system shall propagate new changes across component implementations.
- C. The GPT system shall update to the latest stable updates of ChatGPT.
- D. The GPT system shall utilize new features of stable updates of ChatGPT.

### Scalability

- A. Both systems combined shall prove the scalability of the proof of concept for ARGO.
- B. The system architecture shall be designed as stateless, allowing instance modifying without affecting functionality.
- C. The system should be resilient to individual component failures.

### Evidence The Document Has Been Placed Under Configuration Management

We use Google Docs as a tool to create our document. Following is a table to describe the version changes the document has experienced.

Version In	Version Out	Changes	Reviewed By	Notion Task Numbers
n/a	0.0	Document Creation based on Template	Lillie McMaster and Alina Khan	UG3-3
0.0	1.0	NFR and FR, Engineering Standards	All Group Members	UG3-19, UG3-20, UG3-21, UG3-22
1.0	2.0	Feedback Provided by ARGO Team	All Group Members	UG3-47
3.0	4.0	Changes based on ABET Accreditation Feedback, Added functional requirements	All Group members	UG3-80

Table 9. Evidence The Document Has Been Placed Under Configuration Management

### Engineering Standards And Multiple Constraints

- IEEE Std 830-1998: Software Requirements [[pdf](#)]
- IEEE Std 29148: Requirements Engineering [[pdf](#)]
- ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
  - Life Cycle Processes
  - Requirements Engineering [[pdf](#)]

### Additional References

- Lamsweerde, A.V., 2009. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. John Wiley
- Non-Functional Requirements in Software Engineering, L. Chung, B. Nixon, E. Yu and J. Mylopoulos, Kluwer Academic Publishing, 2000
- Scenarios, Stories, Use Cases Through the Systems Development Life-Cycle, I. Alexander and N. Maiden (eds.), John Wiley & Sons, 2004.
- Malan R, Bredemeyer D. Functional requirements and use cases. Bredemeyer Consulting. 2001 Mar.