

SE 4485: Software Engineering Project

Spring 2024

Detailed Design 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
Group Members	<ol style="list-style-type: none">1. Lillie McMaster2. Lauren Nguyenphu3. Alina Khan4. Hamdiya Abdulhafiz5. Iman Sheriff6. Timothy Naumov

Abstract

This Detailed Design Document serves as a guide for developers and designers of the Storybook Proof Of Concept continuation. This document details the Graphical User Interface (GUI) design, integration with design tools Storybook, Figma, and Chromatic, and specifications for components and themes. The static and dynamic models of the system, rationale for the detailed design choices, and traceability from requirements to design elements. This document aims to ensure a seamless transition between standards, guidelines, and future developers.

Table Of Contents

Introduction	4
Gui (Graphical User Interface) Design	4
Storybook Controls	4
Storybook with Figma Integration	5
Figma button specification	5
Figma Theme Specification	6
Figma Composite View	7
Static Model	9
Dynamic Model	10
Rationale For Your Detailed Design Model	12
Traceability From Requirements To Detailed Design Model	12
Functional Requirements: based of use cases from requirements documentation	12
Non-Functional Requirements	14
Evidence The Design Model Has Been Placed Under Configuration Management	15
Engineering Standards And Multiple Constraints	15
Additional References	15

List Of Figures

Figure 1: Switch Default Controls	4
Figure 2: Button Primary Controls	4
Figure 3: Switch Design Figma Integration	
Figure 4: Button Design Figma Integration	5
Figure 5: Figma Button Specification	5
Figure 6: Figma Color Specifications	7
Figure 7: Figma Typography, Shadow, and Spacing Specification	7
Figure 8: Figma Desktop View	8
Figure 9: Figma Environment	8
Figure 10: Static Model of the System	9
Figure 11: Dynamic Model with Figma, Chromatic	10
Figure 12: Dynamic Model with Storybook, Figma	10
Figure 13: Dynamic Model with StorybookGPT, Knowledge, and Storybook	11

List Of Tables

Table 1: Traceability from Requirements to Detailed Design of Functional Requirements	13
Table 2: Traceability from Requirements to Detailed Design of Non-Functional Requirements	14
Table 3: Configuration Management Table	15

Introduction

The Detailed Design Document encompasses GUI Design, Storybook Controls, Storybook with Figma Integration, Figma Button Specification, Figma Theme Specification, Figma Composite View, Static Model, Dynamic model, and rationale for each component. The purpose of the document is to illustrate component interactions in various environments, and provide documentation to future developers. Finally, the document maps the functional and nonfunctional requirements to corresponding design elements, ensuring adherence to standards given to us.

Gui (Graphical User Interface) Design

Storybook Controls

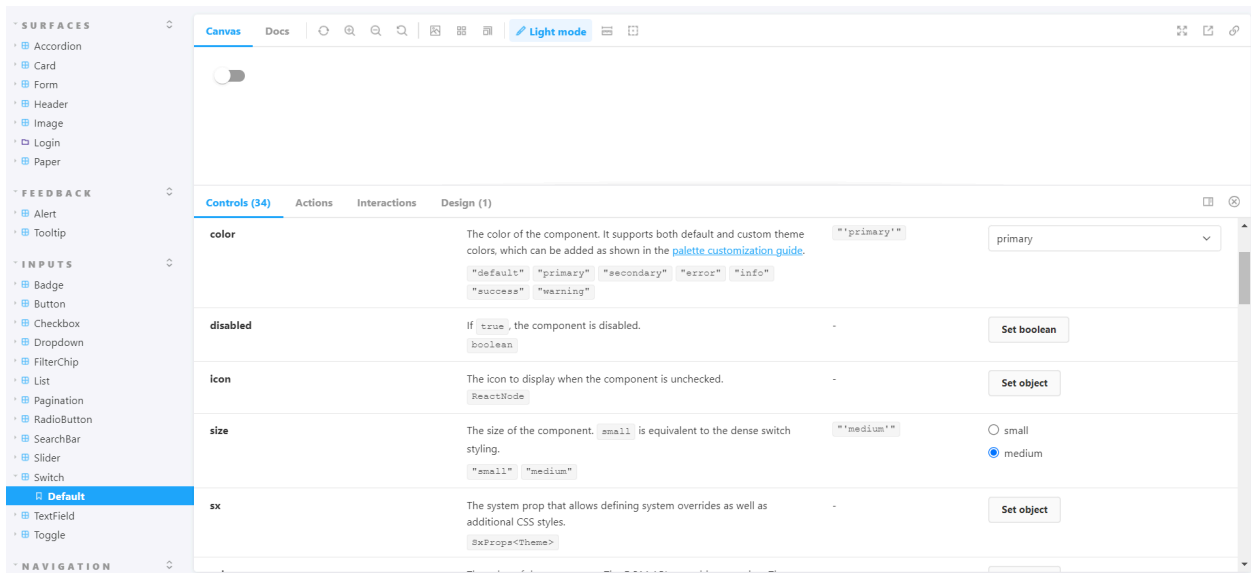


Figure 1: Switch Default Controls

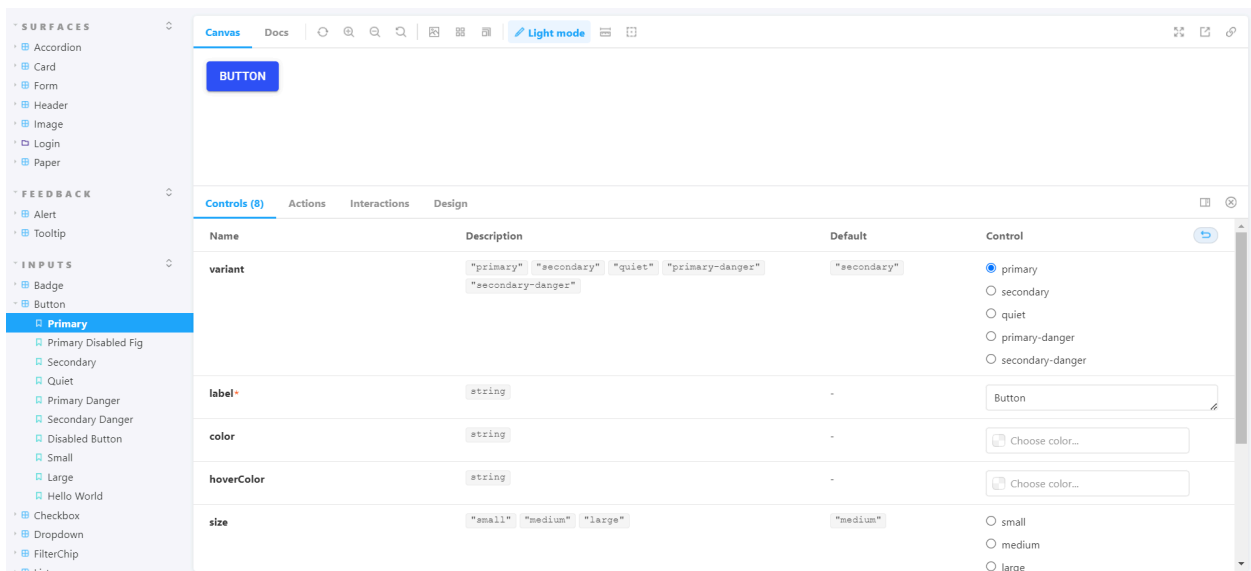


Figure 2: Button Primary Controls

The storybook controls are a Graphical User Interface that allows for developers and designers to modify component configurations and view how each component can react to different properties. The storybook controls were extended to include more options that closely align with the original behavior specified in the Material UI component library.

Storybook with Figma Integration

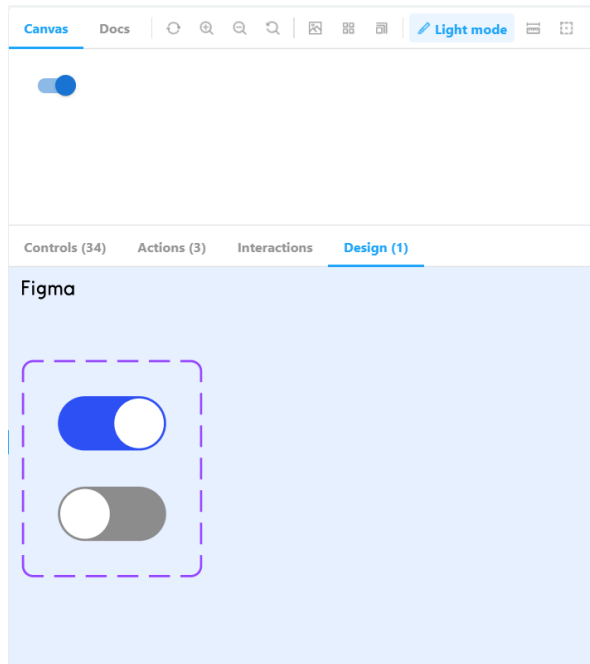


Figure 3: Switch Design Figma Integration

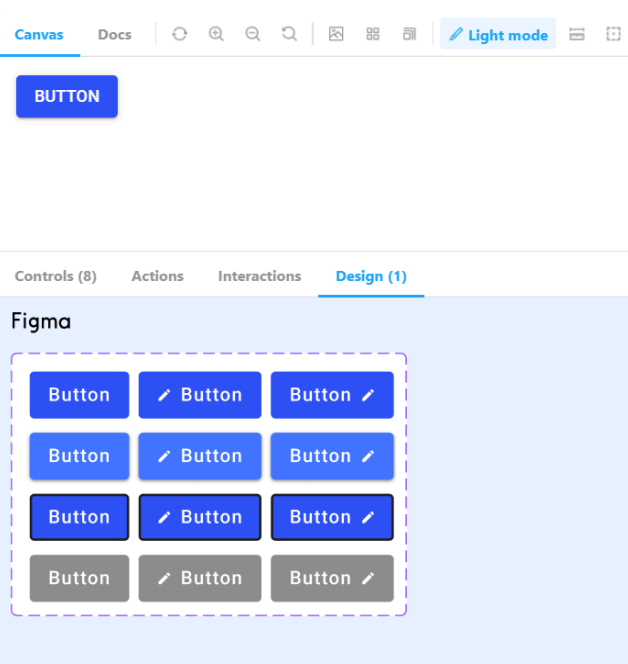


Figure 4: Button Design Figma Integration

The Storybook Figma integration enables viewers to cross reference how components were implemented with the original specifications defined in Figma. The components are linked through Storybook and the Figma integration allows for easy access to how each component was mocked up in Figma.

Figma button specification

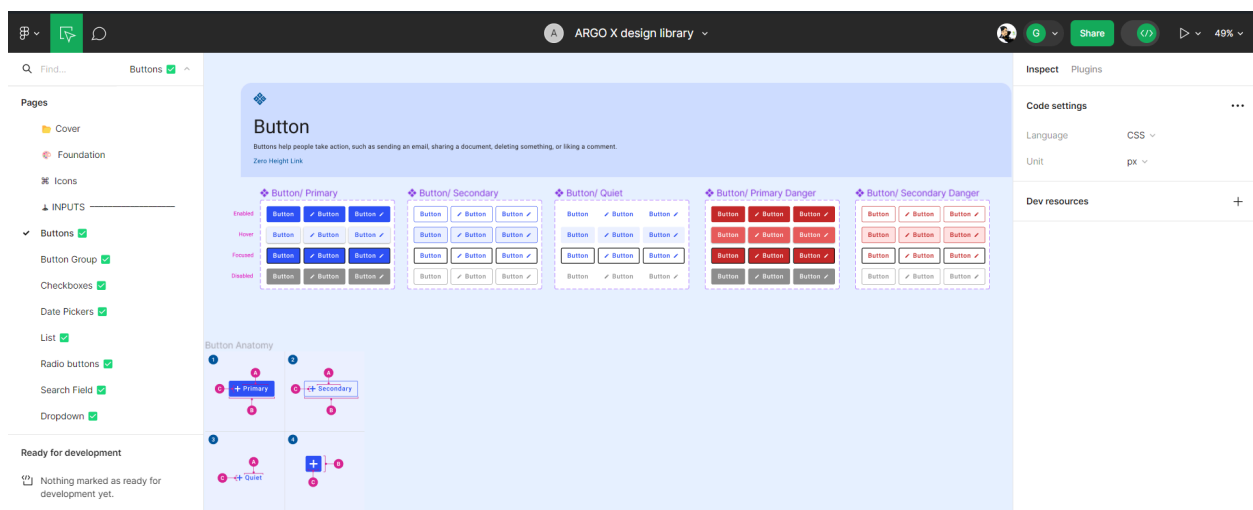


Figure 5: Figma Button Specification

These designs were provided by our Sponsor, Argo, and detail how each button component should behave

under different configurations such as disabling the button or using a primary or secondary button.

Figma Theme Specification

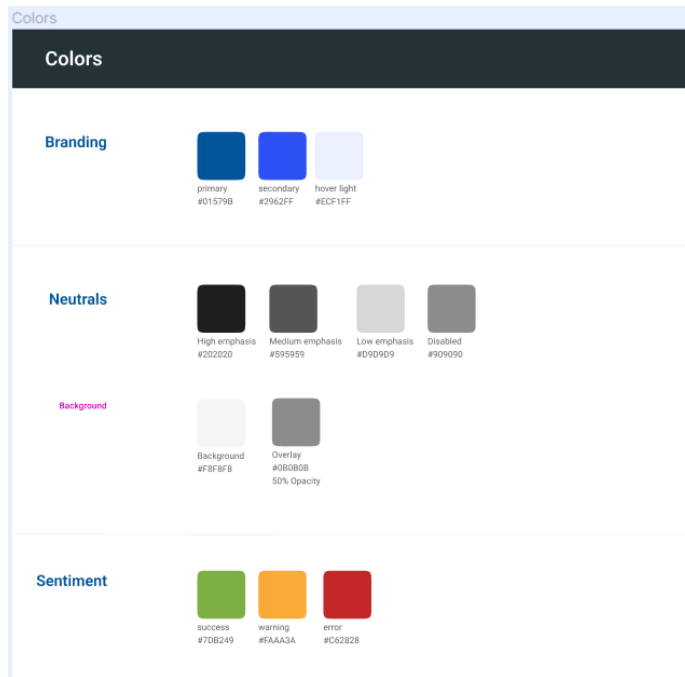


Figure 6: Figma Color Specifications

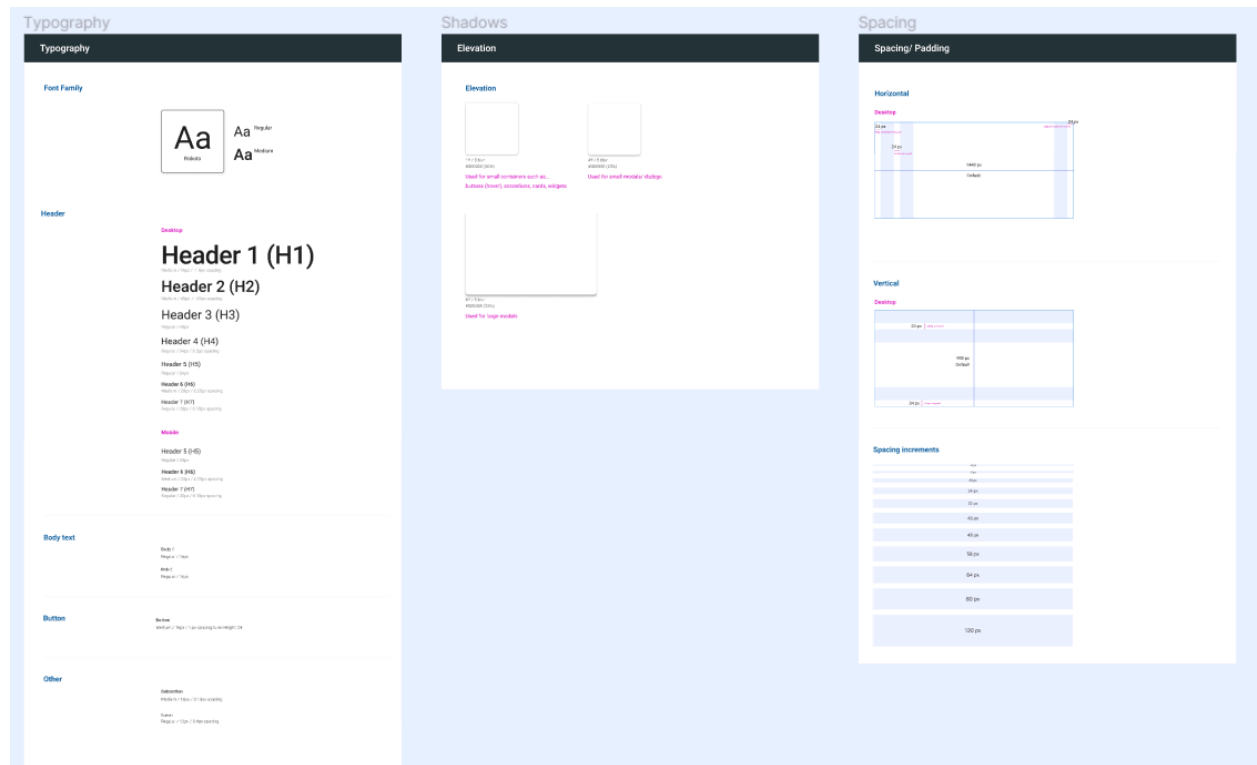


Figure 7: Figma Typography, Shadow, and Spacing Specification

This extensive specification defines what colors should be used in the component library theme as well as detailed theme specifications such as font, font weights, shadows, etc.

Figma Composite View

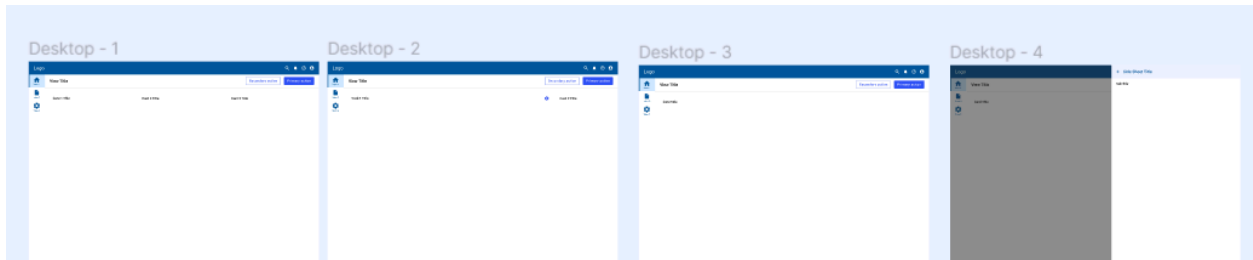


Figure 8: Figma Desktop View

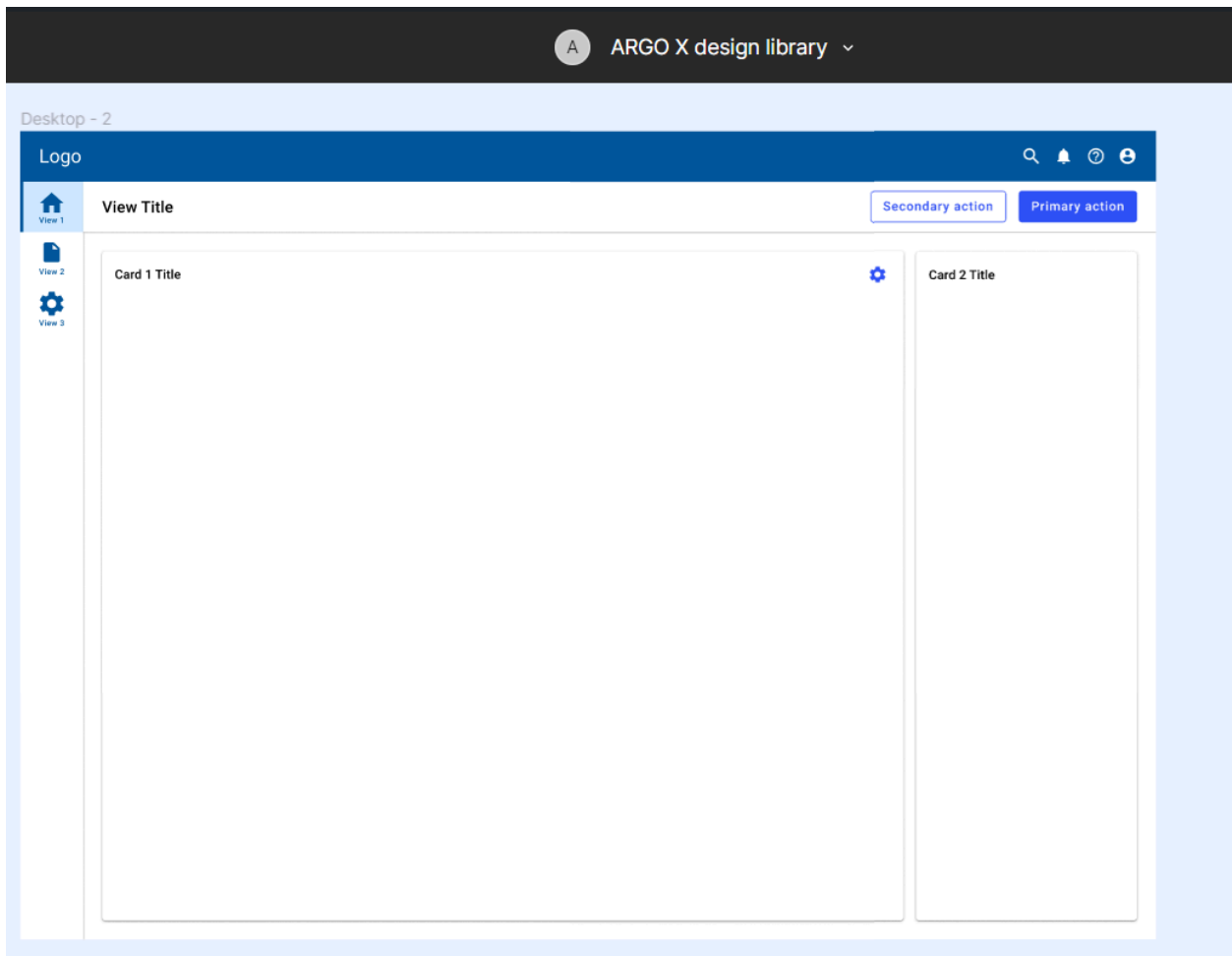


Figure 9: Figma Environment

The composite view is a collection of components that shows how they would interact with each other in a larger view. These designs illustrate how our components would work together in various environments and how larger user interfaces can be built from the component library.

Static Model

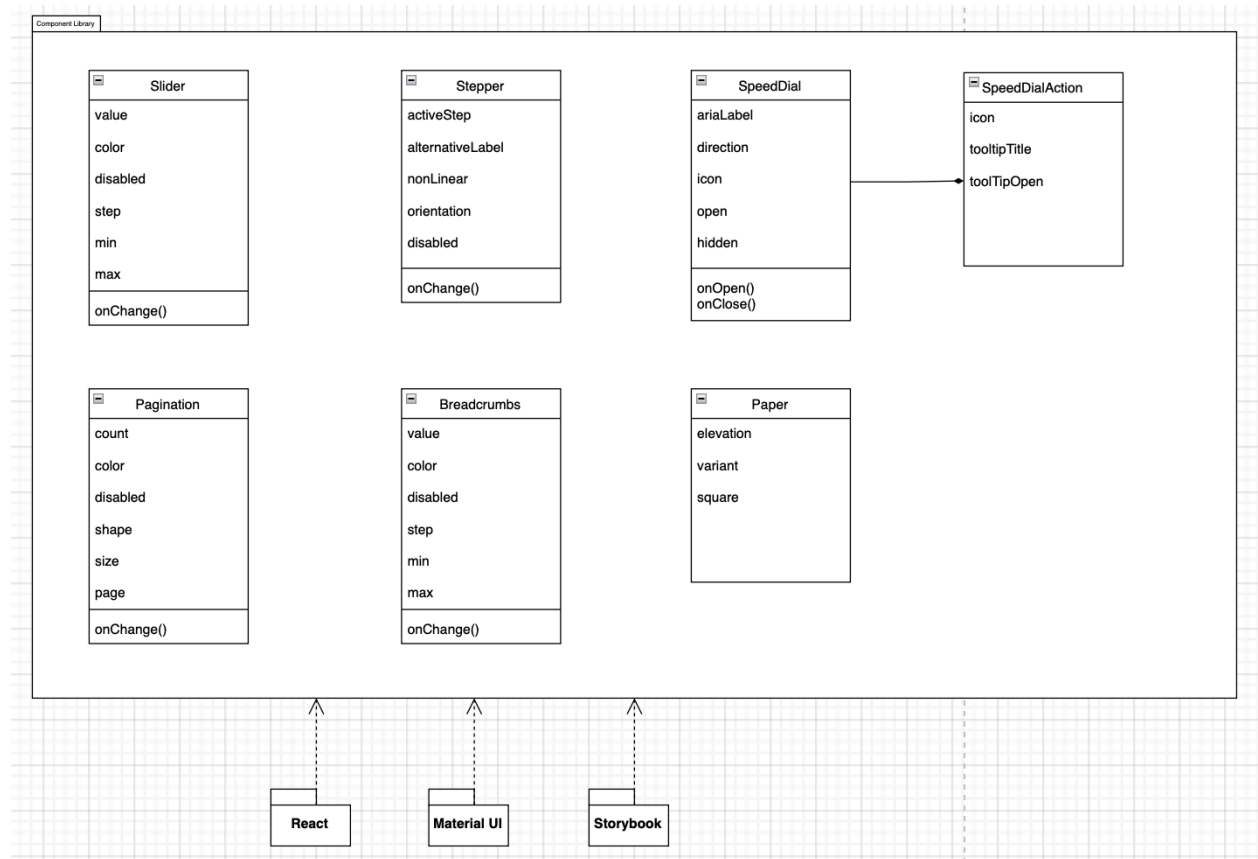


Figure 10: Static Model of the System

Dynamic Model

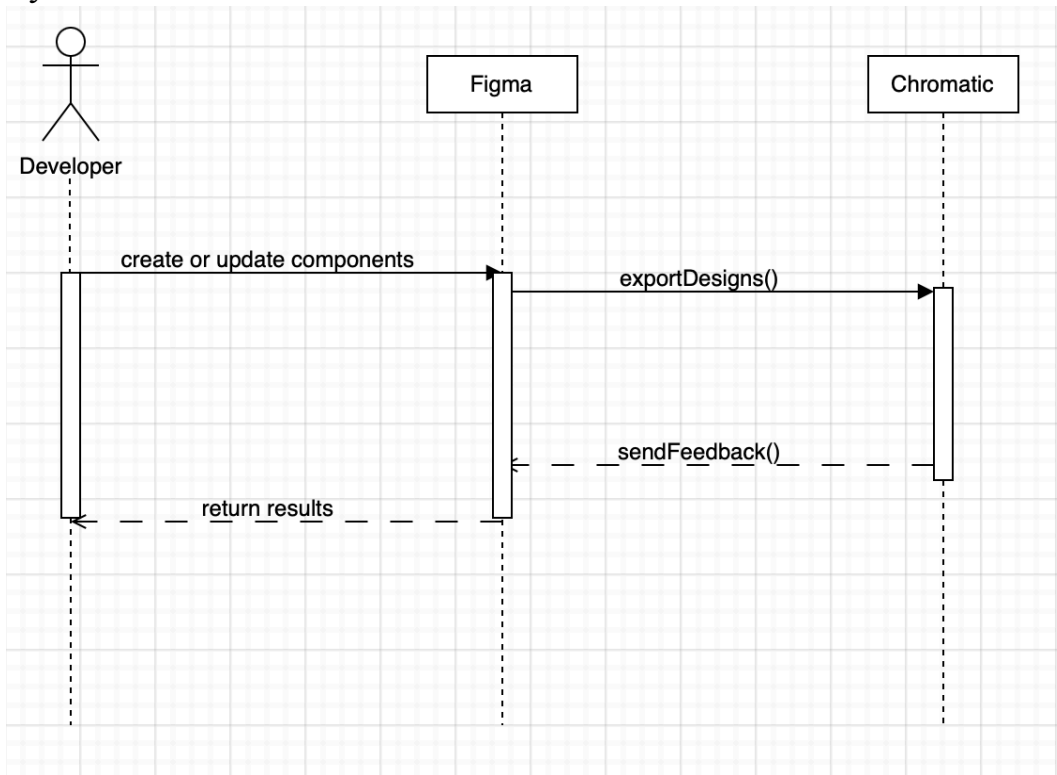


Figure 11: Dynamic Model with Figma, Chromatic

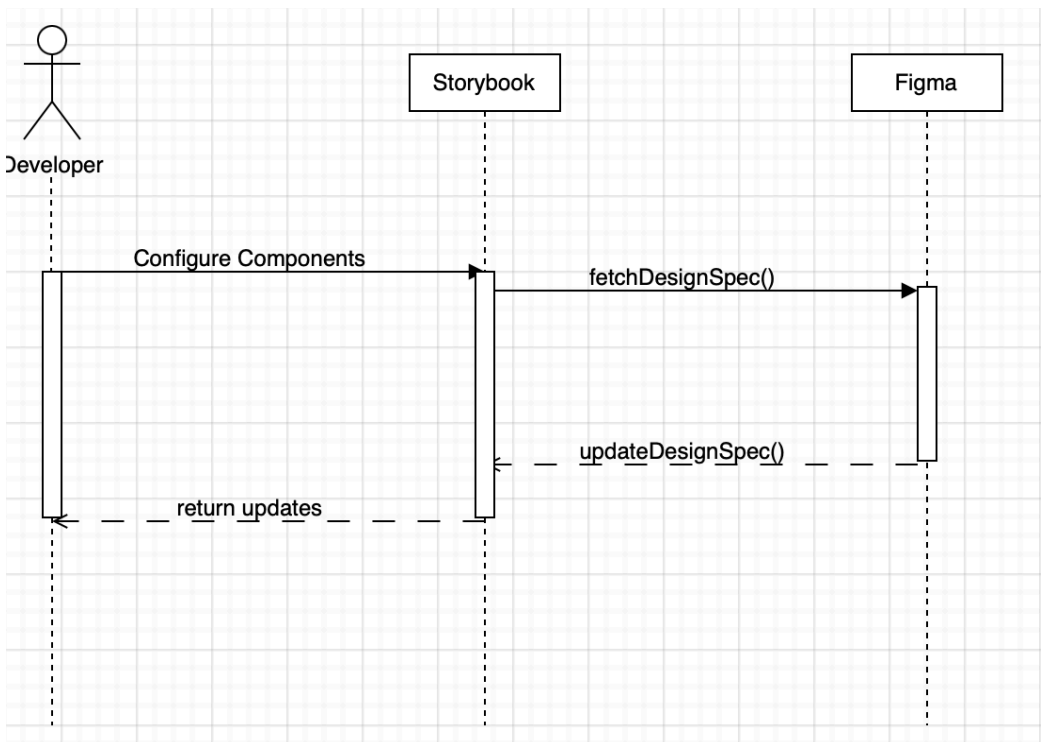


Figure 12: Dynamic Model with Storybook, Figma

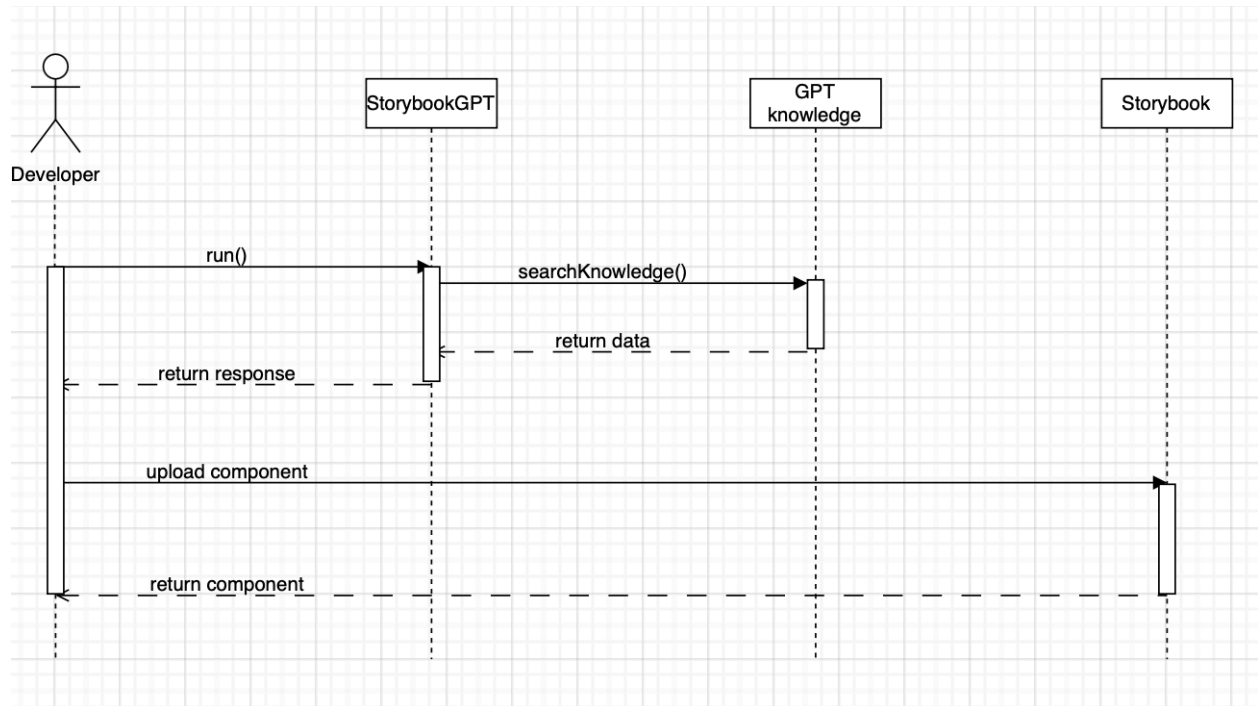


Figure 13: Dynamic Model with StorybookGPT, Knowledge, and Storybook

Rationale For Your Detailed Design Model

Graphic User Interface (GUI):

The component library requires the use of several systems, each with their own GUI's. In figures 1-8, we have outlined how the design of the GUI has impacted the development of components and other features through Storybook. Each of the GUI design's have been driven by the requirements of the features, enabling seamless component development from design to package release.

Static Model:

UML Class Diagrams provide a high level overview of the components that were built in this phase of the project. The UML Class diagram displays the component library and the objects that it compromises.

Each component in the class diagram contains a set of attributes that are associated with the component, as well as a set of methods that apply to that component. UML class diagrams also provide a visual representation of our structure, helping to understand how the components are structured and relate to one another. For example, with a component like Button or Checkbox, the UML class diagram can show the properties and their relationship with other parts of the system such as Button being used within Form.

Dynamic Model:

A sequence diagram was used to demonstrate the dynamic model because it clarifies actions and processes that the class diagram might not have included. Using a sequence diagram can provide a comprehensive perspective among entities, giving a deeper understanding of the system. For our diagram, it illustrates the interactions among a developer, Storybook, Figma, Chromatic, and GPT-4, enhancing the understanding of the system's dynamics.

Traceability From Requirements To Detailed Design Model

Functional Requirements: based of use cases from requirements documentation

Functional Requirement	Design Element	Description
The system allows designers to review, customize, and finalize components, ensuring that all designed components are easy to use and align with user expectations and behaviors.	Figma, Chromatic, Storybook	Figma allows the developer to update the components design and export them to Chromatic and get feedback to the designer to make any adjustments. Storybook allows for the configuration of the components. fig. 11,12
The system enables frontend developers to access and review documentation created, ensuring adherence to necessary standards and guidelines, and approval upon review.	Storybook	Storybook allows frontend developers to access and review documentation created within Storybook. fig. 12
The system must support Component Library Developers in reviewing design specifications and writing code for components.	Storybook, Figma	Storybook allows the enhancement of by streamlining the configuration of components. Figma works to review the design of the components.

		fig.12
The system should allow input parameters for component stories and should generate stories accordingly, also allowing developers to review and approve the generated stories.	Storybook GPT	Storybook GPT allows developers to input parameters to customize components and generate stories. fig.13
The system must enable Component Library Developers to commit code changes to the Component Library repository.	GitHub, npm	The design and utilization of GitHub guarantees reliability and adhere to best practices in version control.
The system should allow Component Library Developers to create pull requests for component changes.	GitHub	The design of Github incorporates functionalities for initiating, reviewing, and merging pull requests, allowing for collaboration and code review across all components.
The system must successfully integrate components into the Component Library after implementation.	Component Libraries, npm, Chromatic	The setup employs npm and Chromatic to integrate and deploy components seamlessly into the Component Library.
The system should support iterative refinement of components until they meet specifications.	Figma, Chromatic, React	The collaborative design setup enables iterative improvements. Figma offers design utilities for enhancing components, Chromatic aids in visual testing and feedback cycles, while React permits adjustments in implementation according to specifications..

Table 1: Traceability from Requirements to Detailed Design of Functional Requirements

Non-Functional Requirements

Non-functional Requirement	Design Element	Description
Usability	Figma	Figma ensures usability by allowing developers to create mockups and visualize components. fig.3,7
Maintainability	GitHub	The design of GitHub allows the system to be easily maintained and allows for updates to easily be made.
Extensibility	GitHub, Storybook	Storybook allows for extensibility by allowing modifications and enhancement without significant changes. GitHub allows the system to be easily updated. fig.12
Scalability	React	The design of React's architecture allows it to handle increasing complexity for growing applications.

Table 2: Traceability from Requirements to Detailed Design of Non-Functional Requirements

Evidence The Design Model 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	UG3-12
0.0	1.0	GUI Design, Static Model, Dynamic Model, Rationale, Traceability, Evidence, Engineering Standards	All Group Members	UG3-96, UG3-97, UG3-98, UG3-99
1.0	2.0	Feedback provided by ARGO	All Group members	UG3-100

Table 3: Configuration Management Table

Engineering Standards And Multiple Constraints

- IEEE Std 1016-1998-(Revision-2009): Software Design [[pdf](#)]

Additional References

- Figma Learn: Storybook and Figma. Accessed April 1st 2024.
<https://help.figma.com/hc/en-us/articles/360045003494-Storybook-and-Figma>
- Storybook Documentation: Building pages with Storybook. Accessed April 8th 2024.
<https://storybook.js.org/docs/writing-stories/build-pages-with-storybook>
- Chromatic Docs: Document. Accessed April 8th 2024.
<https://www.chromatic.com/docs/storybook/document/>
- A. Harbert, W. Lively and S. Sheppard, A graphical specification system for user-interface design, in IEEE Software, vol. 7, no. 4, pp. 12-20, July 1990. Accessed April 8th 2024.