

Stylebin

by

Madison Bintz
Ruairi Blake
Dylan Farmer
Justin Hall
Corey Zaher

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

© Copyright 2022 Bintz, Blake, Farmer, Hall, Zaher

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

<u>Madison Bintz</u>	<u>04/21/2022</u>
Madison Bintz	Date
<u>Ruairi Blake</u>	<u>04/20/2022</u>
Ruairi Blake	Date
<u>Dylan Farmer</u>	<u>04/20/2022</u>
Dylan Farmer	Date
<u>Justin Hall</u>	<u>04/19/2022</u>
Justin Hall	Date
<u>Corey Zaher</u>	<u>04/21/2022</u>
Corey Zaher	Date
<u>Yahya M Gilany</u>	<u> </u>
Yahya Gilany	Date

The authors grant to the School of Information Technology permission to reproduce and distribute copies of this document in whole or in part.

Table of Contents

Abstract	5
Project Summary:	6
Problem Statement:	6
Solution:	6
Project Source:	7
Project Objectives/Goals:	8
Project Scope:	8
Project Timeline:	9
Technologies Used:	9
Technical Architecture Diagram:	11
Data Structure Diagram:	12
User Personas:	13
Use Cases:	15
Use Case Diagram:	21
Testing Plan:	22
<i>Overview</i>	22
<i>Methodology</i>	22
<i>Scope</i>	22
<i>Objectives</i>	23
<i>Testing Log</i>	23
<i>Testing Review</i>	23
Change Management Plan:	24
<i>Purpose</i>	24
<i>Scope</i>	24
<u>Request</u>	24
<u>Review</u>	24
<u>Approve</u>	25
<u>Implement</u>	25
<u>Log</u>	25
<i>Change Log</i>	25
Budget:	26
Problems Encountered and Analysis of Problems Solved:	27
Conclusion:	28
<i>Lessons Learned</i>	28

<i>Skills Developed & Enhanced</i>	29
Appendix A	30
Appendix B	39

List of Figures

Figure 1: Stylebin System Architecture	11
Figure 2: Stylebin Data Structure	12
Figure 3: Use Case Diagram	21

List of Tables

Table 1: Project Timeline	9
Table 2: Testing Logs	30
Table 3: Change Log	39
Table 4: Stylebin Estimated Budget	26

Abstract

Often, small development teams and teams of inexperienced developers run into the problem of incorporating inconsistent frontend components into their projects. These points of inconsistency can be jarring or distracting to the user and can make a project appear messy or unprofessional. Stylebin is a web application that aims to resolve this issue by providing the user with an intuitive graphical interface for creating and styling sets of frontend React components. The user will be able to create workspaces for their projects and create ‘bins’ within those workspaces. Bins contain frontend components and a set of styling preferences that are reflected in those components. New components can be selected from a list, defined through a form, and then added to the bin. These components can then be exported and added to your codebase. The exported zip file contains the components along with a stylesheet that reflects the bin’s styling preferences. We created this project utilizing modern web development technologies to keep our product up-to-date and challenge our development team to learn a new tech stack. As it stands, our MVP serves all of our basic functionality from the frontend. Our future plans involve fleshing out the feature set that already exists within Stylebin, including adding more styling options and adding additional component types for users to work with.

Project Summary:

Stylebin is an open-source web platform for the creation, styling, and management of React component sets. Stylebin serves as an organizational and educational tool for inexperienced or hobby developers, which will allow users to create and customize React components for use in their projects. Stylebin is an open-source project, which means that our users can utilize our hosted instance to create and manage their components or pull our source code to browse and understand how our React components are created under the hood.

Problem Statement:

Having been in their position, we understand that our target audience of inexperienced or hobby developers may not know how to standardize their component styling or lack an accessible platform for creating and organizing them. Platforms like Storybook exist to be an organizational tool for development teams to maintain and document their already-created frontend components. Stylebin allows its users to create and style their components on-platform and then export them locally to use with their programming project. This platform could be used alongside a package like Storybook to add an extra layer of documentation to their frontend.

Solution:

Stylebin is a web application that will allow users to easily create and customize React components in a collaborative environment. Users will create workspaces, which encompass an entire project, bins, which encompass a set of components, and components, which can be customized and styled in-app.

Components will be grouped together in “Bins”. Bins keep components (i.e., Buttons, sliders, frames, selectors, etc...) organized in groups based on a common user-created theme. All components inside of a bin will be given the same theme, which consists of color palettes, fonts, and other common attributes.

Users will choose from a list of different template components ranging from buttons to modals with multiple functions. After choosing a component, the user will be able to customize and configure attributes to their liking. Once finished editing their components, the users will export the bin making up the components and styling information so that they can include it in their source code. The components will be exported into a “components” folder, which can be easily dragged into an existing React project.

Our application will host the attributes and styling of the components that our users create on our server. This will allow our users to access, modify, and export their components wherever they are.

Since collaboration is a cornerstone of Stylebin, role-based access control will be included. Stylebin will include user roles such as workspace administrators, contributors, and viewers.

Workspace administrators can maintain their development team by adding and removing their partners as well as modifying their permissions, restricting their ability to interact with or modify

the project. Workspace administrators can also maintain their project contents by adding, removing, and modifying bins with their development team.

Users with the contributor role will not be able to modify the permissions of users or send invitations to the project. However, they will be able to add, remove, and modify bins.

Users with the viewer role will not be able to add, remove, or modify anything about the project, but they will be able to export a bin to use in the development of their project.

By implementing Stylebin into your development project, you will be able to separately maintain sets of frontend React components. Users can experiment with the styling of their React components in a styling sandbox and be assured that their development team is entirely on the same page.

One of the ways that Stylebin stands out as a unique product against some similar platforms is its positioning as an open-source application. Developers will be able to pull the code for Stylebin, dissect it to understand how it works and use it as an interactive learning tool for frontend React development.

Project Source:

The idea for Stylebin came by considering some of the inconveniences that impact small or inexperienced development teams. Working at a small technology startup that utilized React as a frontend framework made our group aware of the difficulty that comes with organizing a team to consistently style frontend React components.

Corey, Dylan, and Justin were working at the Lindner College of Business IT Service Desk when they decided to form their senior design group. Later Corey invited Ruairi, a peer that he met during a software development co-op at Refract Labs, and Dylan invited Madison, a peer that he met during a cybersecurity co-op at Vega Americas.

Project Objectives/Goals:

The following list contains the different goals and objectives that we hope to achieve over the course of this project.

- **Workspace Organization** – Users will have the ability to create workspaces, bins, and React components. Workspaces encompass entire software projects. Workspaces contain one or more bins. These bins will contain different React components.
- **Unified Component Styling** – Users will have the ability to apply styles and themes for components within a bin.
- **Export Components** – Users will be able to easily export Stylebin stylesheets and components for easy integration with their React applications.
- **User Management** – Users will have the ability to log in through SSO using their accounts with other services like GitHub, Microsoft, or Google. Different permission schemes will be available to fine-tune how users are allowed to interact with different aspects of a workspace. Groups will be available to enable easy application of permissions to multiple users.
- **User Collaboration** – Users will be allowed to invite other users to groups/workspaces to enable a collaborative working environment.

Project Scope:

Our team will develop a web application that allows a user to:

- Create and customize ReactJS components
- Organize components into manageable workspaces and bins
- Easily collaborate with other Stylebin users

This will enable our users to create and manage a library of ReactJS components for their applications more easily than with other open-source methods available today.

We assume that our users are familiar with how to implement a React component into their code as our software will be outputting a components folder for the user. Our component creation system will be constrained by the component styling packages and assets that we implement. We will have to limit these packages based on implementation complexity and package stability.

Project Timeline:

The following table contains a preliminary timeline for the completion of certain tasks and features.

Table 1: Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Team contract completion	4 weeks	8/23/21	9/20/21
2	Environment setup	1 week	9/20/21	9/27/21
3	Prototype	1 week	9/27/21	10/04/21
4	Sign-in with Auth0	2 weeks	10/04/21	10/18/21
5	Ability to create workspaces, bins, and components	2 weeks	10/18/21	11/01/21
6	Export components	2 weeks	11/01/21	11/15/21
7	Deployment of MVP	1 week	11/15/21	11/22/21
8	Gather user feedback	1 week	11/22/21	11/29/21
9	Implement User feedback	1 week	11/29/21	12/06/21
10	Technical evaluation	1 week	12/06/21	12/13/21
11	Winter break	2 weeks	12/13/21	12/27/21
12	Join groups, email invite to the group	2 weeks	12/27/21	01/10/21
13	Admin permissions & account controls	2 weeks	01/10/21	01/24/21
14	Accessibly settings, dark mode	2 weeks	01/24/21	02/07/21
15	QA, security testing, refactoring	Remaining Time	02/07/21	IT Expo

Technologies Used:

The technologies used for Stylebin were picked to enable creative development. To start, we bootstrapped the project with React and added Typescript. We picked React because our developers were most familiar with it, and being a JavaScript library, it would be easy for the rest of the group members to pick up on. For the sake of simplicity and continuity, MaterialUI was chosen as our component library for both styling the site, and the exported components. Auth0 was picked as our authenticator, and SendGrid as our email service, as they are both widely used and trusted services.

We built the backend to use Apollo server, GraphQL and a SQLite database. This choice was made as we were mostly unfamiliar with the technologies and saw this as an opportunity for a learning experience.

The following list contains the different applications, programming languages, software frameworks, etc. that we will use while developing Stylebin.

- React
- MaterialUI
- GraphQL
- Apollo Server + Express
- SQLite
- Typescript
- Auth0
- Sendgrid
- Firebase
- Heroku
- GitHub
- VSCode

Technical Architecture Diagram:

Figure 1, below, depicts the basic architecture of our system. The flow represented in this diagram depicts how the frontend for our product is hosted, how the frontend is served to our clients, how the front and backend communicate through a GraphQL application layer, and how the backend makes requests to the database.

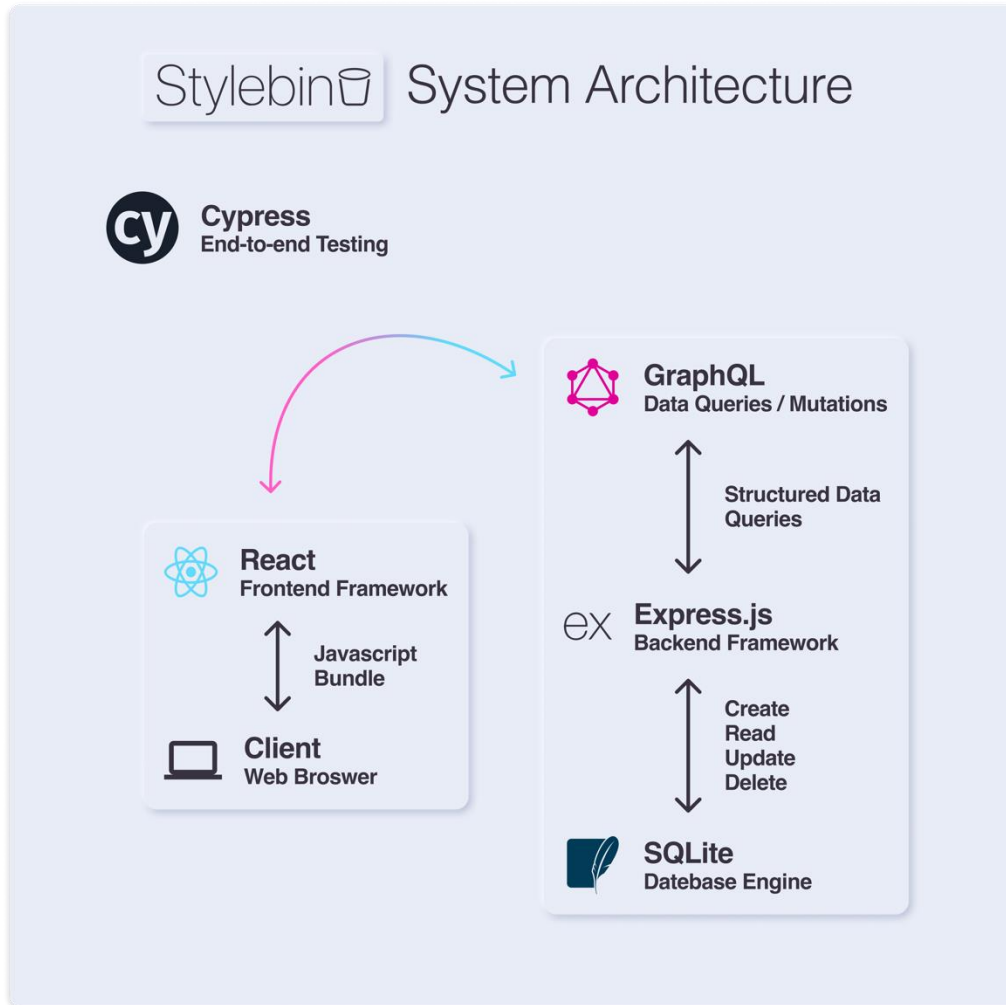


Figure 1: Stylebin System Architecture

The Stylebin architecture begins with the client web browser, which connects to the React frontend. As the user navigates around the website, and new data is loaded, the frontend interacts with the GraphQL backend. This GraphQL backend, which is run by Apollo Server + Express.js, handles all the data queries and mutations that are necessary for Stylebin to operate. Stylebin is using an SQLite database to store all application information.

Data Structure Diagram:

Figure 2, below, depicts the data structure for our MVP product's backend. The models represented in this diagram make up the data entities that exist in our product. Models directly reflect the table structure in our database. For the sake of effectiveness and space, the long list of component attributes has been shortened for the Component entity.

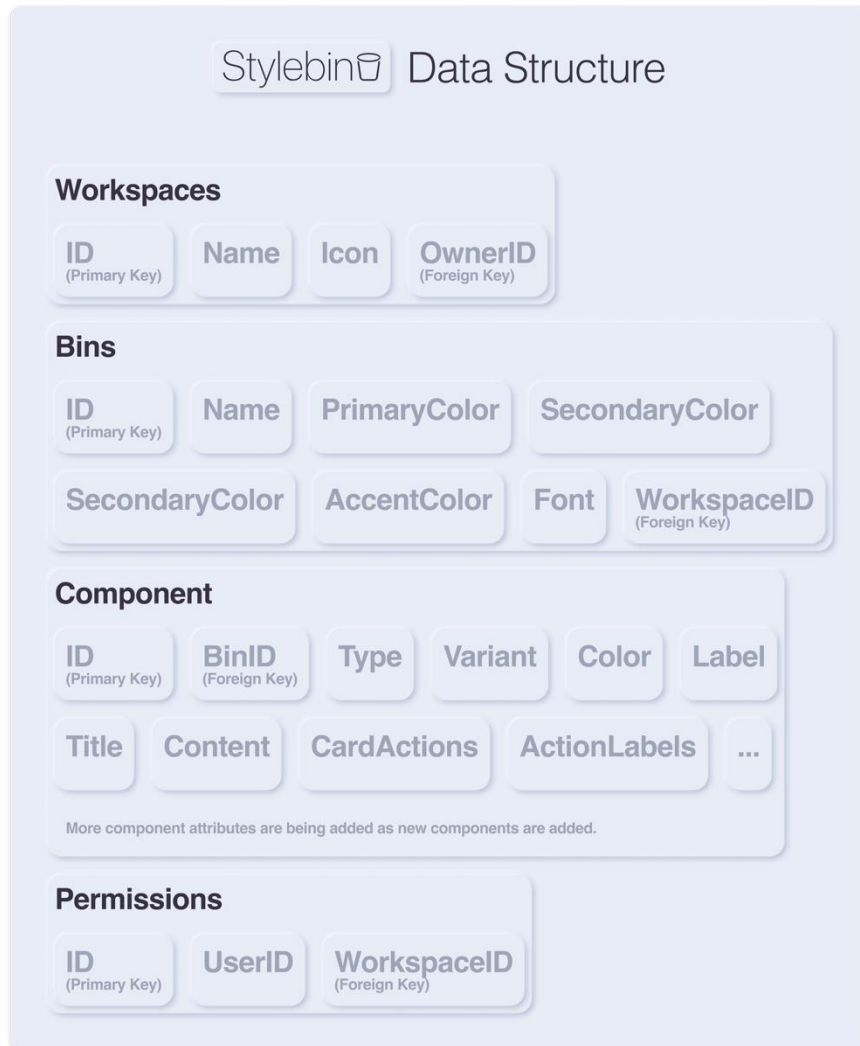

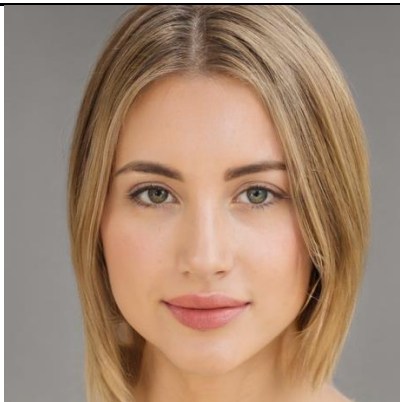


Figure 2: Stylebin Data Structure


User Personas:

User personas are hypothetical users with fleshed-out personalities, which are designed to represent not only themselves but a larger group of people who will use our application. The purpose of these personas is to provide the development team with a deeper understanding of the needs of their users. Below, we have created three user personas that represent the needs of our users.

User Persona: 1	
	Title The Busy College Student
	Name Gabriel Johnson
	Age 24
	Gender Male
Behavior	Gabriel is attending his local university to earn a bachelor's degree in software development.
Pain	Gabriel takes part in other extracurricular activities, which takes away some of his free time.
Needs & Goals	Gabriel wants to apply what he is learning outside of the classroom, to develop his skills further. He is looking for a way to practice making functional apps without spending too much time on the design aspect.

User Persona: 2	
	Title The Start-Up Developer
	Name Helena Cortez
	Age 45
	Gender Female

Behavior	Helena was hired at a start-up company as the lead developer.
Pain	Helena doesn't have any past developments of the start-up to go off of to begin her work.
Needs & Goals	Helena needs to find a consistent way to develop for all of the future developments within the company to be cohesive.

User Persona: 3	
	Title The Newb
	Name Yolanda Jones
	Age 16
	Gender Female
Behavior	Yolanda loves breaking stereotypes and learning new skills. She has a lot of free time on her hands and has been trying her hand at programming.
Pain	Yolanda does not have a lot of experience in developing, so while she may have plenty of fun projects to keep her occupied in her free time, she does not necessarily know the best way to go about creating them.
Needs & Goals	Yolanda needs the easiest and most consistent way to develop the projects that she wants to create, all so she can develop a rhythm and style that can help her when she is wanting to move forward with a developing career.

Use Cases:

The following use cases define the core functionality of Stylebin and the general process that users will follow to use those functionalities.

Use Case ID	SB_001
Use Case Name	Create Workspace
End Objective	User has a new organizational unit to hold different bins that relate to a project.
User/Actor	Stylebin Users
Trigger	User clicks the “New Workspace” button
Frequency of Use	Used whenever a user starts working on a new project
Preconditions	User must be logged in to Stylebin
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1) 2. The user clicks the “New Workspace” button 3. The user enters information about the workspace (name, icon, etc.) (See AC2) 4. User is returned to the main workspace selection screen and may select the new workspace.
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – A workspace already exists with the given name</p> <ol style="list-style-type: none"> 1. User enters a new, unique, name 2. Return to step 4
Postconditions	The new workspace is shown in the workspace sidebar.

Use Case ID	SB_002
Use Case Name	Managing a Workspace
End Objective	User has changed settings for a workspace.
User/Actor	Workspace Admins
Trigger	User clicks the settings button next to the name of the workspace they wish to edit.
Frequency of Use	Used whenever a workspace admin wants to change details about a workspace such as the display name or sharing permissions.
Preconditions	User must be logged in
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1) 2. The user selects the workspace they wish to edit 3. The user clicks the “Settings” button. 4. The user edits any settings that need to be changed in the workspace settings menu. (See AC2) 5. The user saves their changes (See AC3)
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – User is not a workspace administrator</p> <ol style="list-style-type: none"> 1. The current settings are displayed, but cannot be edited

	AC3 – User wants to discard changes 1. The user clicks the “Cancel” button 2. Return to step 2
Postconditions	

Use Case ID	SB_003
Use Case Name	Create Bin
End Objective	The user has a new bin to create components and apply like styles to them.
User/Actor	Workspace Admins, Contributors
Trigger	User clicks the “New Bin” button
Frequency of Use	Used whenever a user wants to apply different styles to a set of new components
Preconditions	User must be logged in User must have selected one workspace
Basic Flow	1. The user logs into their Stylebin account. (See AC1) 2. The user selects the workspace that the bin will live in. (See AC2) 3. The user clicks the “New Bin” button (See AC3) 4. The user enters the name of the bin 5. The user selects the styling information that applies to the bin 6. User is returned to a window with their new bin, so they can begin creating components.
Alternate Flow	AC1 – User is already logged on 1. Return to step 2 AC2 – User does not have any workspaces 1. Follow SB_001 2. Return to step 2 AC3 – User does not have workspace admin or contributor permissions on the selected workspace 1. Display message stating that user has insufficient permissions 2. Exit workflow AC4 – A bin already exists in the workspace with the given name 1. User enters a new, unique, name 2. Return to step 4
Postconditions	The new bin is shown within the workspace it was created in.

Use Case ID	SB_004
Use Case Name	Managing a Bin
End Objective	User has changed the settings for a bin.
User/Actor	Workspace Admins, Contributors
Trigger	User clicks the Settings button in the Bin they want to edit.
Frequency of Use	Used whenever a user wants to change the settings on a bin that they have access to.
Preconditions	User must be logged in Users must have at least one workspace

	User must have at least one bin
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1) 2. The user selects the workspace that contains the bin they wish to manage 3. The user opens the bin they want to edit 4. The user clicks the “Settings” button. 5. The user modifies any settings they want to be changed in the bin settings menu. 6. The user saves any settings they want to be changed (See AC2)
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – User wants to discard changes</p> <ol style="list-style-type: none"> 1. The user clicks the “Cancel” button 2. Return to step 2
Postconditions	A bin has been modified

Use Case ID	SB_005
Use Case Name	Creating a Component
End Objective	User has created a new React component which can be modified according to the user.
User/Actor	Workspace Admins, Contributors
Trigger	User clicks the “New Component” button.
Frequency of Use	Used whenever a new React component is needed for a project.
Preconditions	<p>User must be logged in</p> <p>User must have at least one workspace</p> <p>User must have at least one bin</p>
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1) 2. The user selects a workspace in the sidebar 3. The user selects a bin 4. The user clicks the “New component” button (See AC2) 5. The user selects what type of component they want to create 6. The user edits attributes for the component type 7. The user saves their changes (See AC3)
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – User does not have workspace admin or contributor permissions on the selected workspace</p> <ol style="list-style-type: none"> 1. Display message stating that user has insufficient permissions 2. Exit workflow <p>AC3 – User wants to discard changes</p> <ol style="list-style-type: none"> 1. The user clicks the “Cancel” button 2. Return to step 4
Postconditions	A new component is created in the selected bin.

Use Case ID	SB_006
Use Case Name	Modifying a Component
End Objective	User modifies a component
User/Actor	Contributor, Workspace Admin
Trigger	Component is opened
Frequency of Use	Whenever a user wishes to open a component to edit
Preconditions	User must be logged in User must have at least one workspace User must have at least one bin User must have at least one component
Basic Flow	1. The user logs into their Stylebin account. (See AC1) 2. The user selects a workspace in the sidebar 3. The user selects a bin 4. The user selects a component to open 5. The user makes the desired changes to the component 6. The user saves any changes made. (See AC3)
Alternate Flow	AC1 – User is already logged on 1. Return to step 2 AC2 – User wants to discard changes 1. The user clicks the “Cancel” button 2. Return to step 4
Postconditions	A component is modified

Use Case ID	SB_007
Use Case Name	Export a Bin
End Objective	Export a created bin to use in a project
User/Actor	All Stylebin Users
Trigger	Export button is selected when viewing a bin
Frequency of Use	Used whenever a user has finished designing the component style they wish to use in their projects.
Preconditions	User has an existing bin in their workspace
Basic Flow	1. The user logs into their Stylebin account. (See AC1) 2. The user selects the workspace containing the bin they want to export (See AC2) 3. User selects the bin they want to export 4. User clicks the export button on the Bin’s page 5. A zip file containing the Bins contents is downloaded
Alternate Flow	AC1 – User is already logged on 1. Return to step 2 AC2 – The desired workspace is already selected 1. Return to step 3
Postconditions	A zip file containing the contents of the bin has been downloaded to the User’s computer

Use Case ID	SB_008
Use Case Name	Invite a User to a Workspace
End Objective	A workspace admin will share a workspace with another user.
User/Actor	Workspace Admin
Trigger	Workspace admin types in the email address of another user, assign a role, and saves the settings for a workspace.
Frequency of Use	Used whenever a workspace admin wants to share the workspace with another user.
Preconditions	User must be logged in User must have at least one workspace
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1) 2. The user selects the workspace they wish to edit 3. The user clicks the “Settings” button. 4. The user clicks the “Invite User” button 5. The user types in the email address of the other user they wish to share the workspace with. (See AC2) 6. The user selects a role to assign to the other user. 7. The user clicks the “Invite” button (See AC3) 8. The system interacts with SendGrid to send an invitation email to the other user
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – User does not have workspace admin or contributor permissions on the selected workspace</p> <ol style="list-style-type: none"> 1. Display message stating that user has insufficient permissions 2. Exit workflow <p>AC3 – User wants to discard changes</p> <ol style="list-style-type: none"> 1. The user clicks the “Cancel” button 2. Return to step 2
Postconditions	An email with a link granting access to a workspace is shared with another user.

Use Case ID	SB_009
Use Case Name	Edit a User’s Workspace Permissions
End Objective	A workspace admin will be able to change the role assigned to a user that whom a workspace is already shared.
User/Actor	Workspace Admin
Trigger	Workspace admin changes the role assigned to a user and saves the settings for the workspace.
Frequency of Use	Used whenever a workspace admin wants to change the permissions a user has to their workspace.
Preconditions	User must be logged in User must have at least one workspace
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1)

	<ol style="list-style-type: none"> 2. The user selects the workspace they wish to edit 3. The user clicks the “Settings” button. 4. The user selects the new role that the other user will be assigned. 5. The user clicks the “Save” button (See AC3) 6. The system updates the other users’ permissions.
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – User does not have workspace admin or contributor permissions on the selected workspace</p> <ol style="list-style-type: none"> 1. Display message stating that user has insufficient permissions 2. Exit workflow <p>AC3 – User wants to discard changes</p> <ol style="list-style-type: none"> 1. The user clicks the “Cancel” button 2. Return to step 2
Postconditions	A user’s permissions will be changed according to the role assigned by the workspace admin(s).

Use Case ID	SB_010
Use Case Name	Remove a User’s Workspace Permissions
End Objective	A workspace admin removes another users’ permissions to the workspace, thereby unsharring the workspace with them.
User/Actor	Workspace Admin
Trigger	Workspace admin removes the user from the list of users that a workspace is shared with, and clicks save.
Frequency of Use	Used whenever a workspace admin wants to remove the permissions a user has to their workspace.
Preconditions	<p>User must be logged in</p> <p>User must have at least one workspace</p>
Basic Flow	<ol style="list-style-type: none"> 1. The user logs into their Stylebin account. (See AC1) 2. The user selects the workspace they wish to edit 3. The user clicks the “Settings” button. 3. The user removes the other user from the list of users that the workspace is shared with. 4. The user clicks the “Save” button (See AC3) 5. The system updates the other users’ permissions.
Alternate Flow	<p>AC1 – User is already logged on</p> <ol style="list-style-type: none"> 1. Return to step 2 <p>AC2 – User does not have workspace admin or contributor permissions on the selected workspace</p> <ol style="list-style-type: none"> 1. Display message stating that user has insufficient permissions 2. Exit workflow <p>AC3 – User wants to discard changes</p> <ol style="list-style-type: none"> 1. The user clicks the “Cancel” button 2. Return to step 2
Postconditions	A user’s permissions will be removed by the workspace admin(s).

Use Case Diagram:

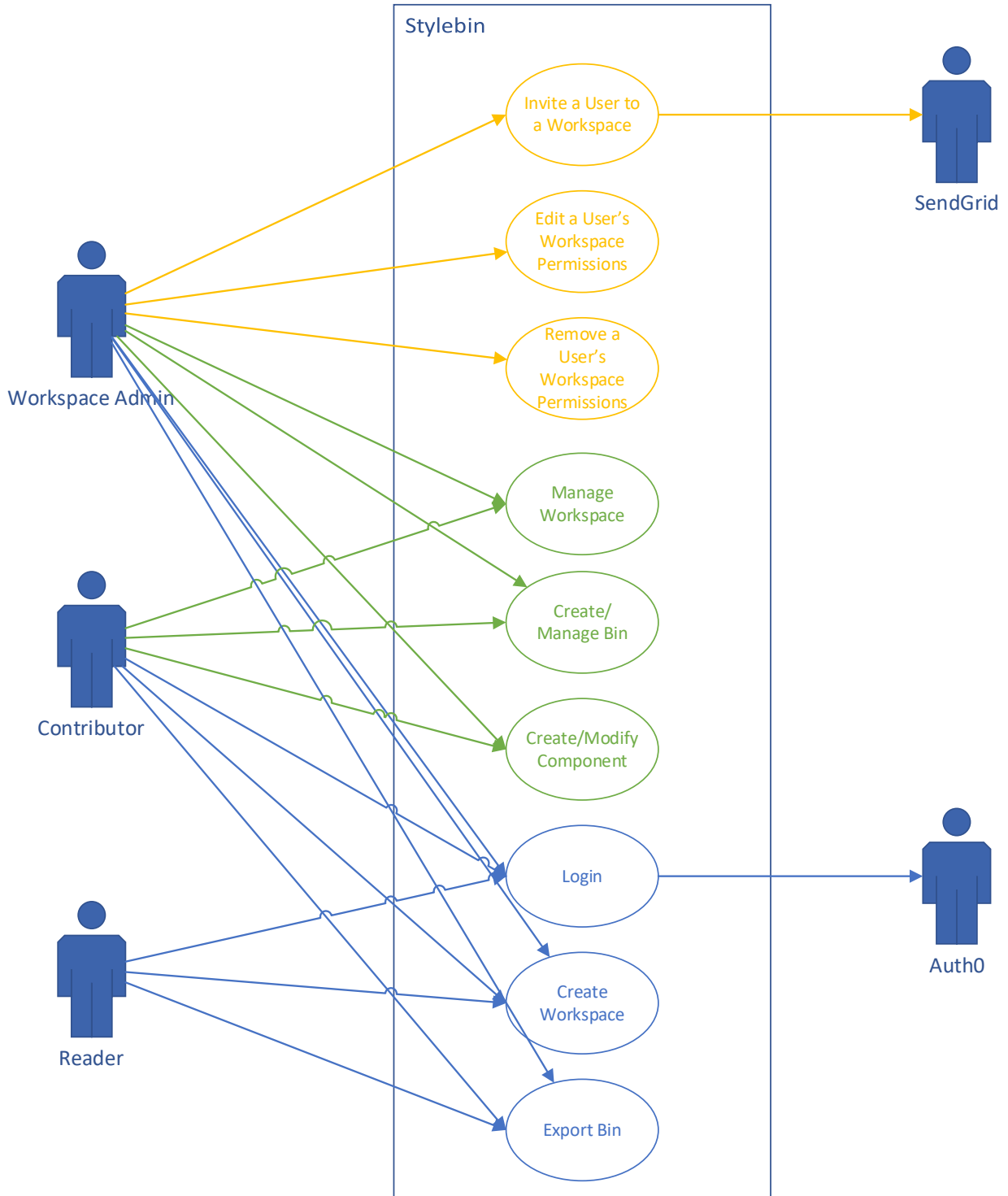


Figure 3: Use Case Diagram

Testing Plan:

Overview

This section will discuss the different aspects related to testing and quality assurance within Stylebin. We will discuss the different testing methodologies employed by our QA team that ensure the functionality of our application remains usable, the initial use cases and features that our tests will focus on, the different objectives that our testing hopes to solve, and, finally, a review of our testing progress so far.

Methodology

We are utilizing the Cypress testing framework to automate many of our tests to both ensure that our tests are being completed in a standardized fashion and free up our Quality Assurance team members to focus on more important tasks. Cypress allows us to quickly create tests so that we can ensure new functionality is being adequately evaluated.

The QA team is working on creating new tests as more features are added to Stylebin. The first step in the process is to manually create a test. From there, the QA team must decide if those tests are both functional and the best way to test its feature set. Tests are identified based on their test case number, which will be determined by the scope below. If needed, more test cases can be developed to match the needs of the scope. Each test case will be continuously tested manually by the QA team until the test no longer fails. Once a test passes, the test case will only be tested weekly to ensure that the test continues to pass. Tests will be stored in an Excel spreadsheet. A sample of the Test Log is below.

Scope

The following use cases and features are core components of Stylebin that we feel should be consistently tested. This will help us ensure that our users have a pleasant, and bug-free, experience using our application.

1. Login in / logout
 - a. We will be testing this feature to ensure that user credentials are being checked by our authentication provider and handled properly.
2. As a user, creating a workspace, style, and bin
 - a. Any user should be able to create a workspace, style, and bin at anytime
3. As a user, modify a workspace, style, and bin
 - a. Any user should be able to modify a workspace, style, and bin at anytime
4. As a user, creating a custom component
 - a. This is the core of the product; users should be able to select a component to style and customize it according to their needs
5. As a user, viewing existing workspaces, bins, and components
 - a. Users should be able to view all workspaces that they have created, as well as any workspaces that they have been invited to.
6. Exporting components
 - a. We are testing this feature to ensure that users can utilize the components created in Stylebin in their projects.

7. Inviting users to a workspace
 - a. We are testing this feature to ensure that groups can be created in the application to enable collaboration.
8. Removing users from a workspace
 - a. Users with appropriate permissions should be able to remove members of the workspace at any time.
9. Modifying permissions on a workspace (name, permissions)
 - a. Users with appropriate permissions should be able to update the permissions of a given user at any time.

Objectives

The following list contains a set of goals that we hope to achieve through proper testing and planning.

- All major use cases and features must be accounted for in the testing plan
- Squash any major bugs by early April (IT Expo)
- Ensure the user experience is consistent across different platforms
- Integrate Cypress with our CI workflow

Testing Log

For Testing Log see Appendix A

Testing Review

For our testing process, we decided to use Cypress to run tests of our workflows for our application. This allowed us to be able to see what the user can achieve, guaranteeing that our application can perform all our desired functionalities. As the development process progressed, we created tests and updated existing tests to confirm the app's functionality. Cypress allowed us to bundle these individual tests in order to test out user workflows. The important objective throughout the testing process was for our tests to cover all major use cases and features, so being able to link all these tests helped us achieve that goal.

In the early stages of development, our testing process consisted of planning what aspects of the application required testing and how we could use the functionalities of Cypress to test them. In this part of the process, we were able to see what elements of an application Cypress could identify and interact with. This allowed us to have development to create unique identifiers for the primary elements of the application. Making our scripts appear cleaner and run more efficiently.

As the development process progressed, we found that testing became even more important. We learned the importance of our core functionalities. Through testing, we were able to see how our users can manipulate Stylebin, and how we could potentially adapt Stylebin into working in a way that will better suit our users. With testing, we could see what features were fully integrated

and what features required more development over time. When we integrated our backend with our frontend, testing allowed us to find issues with how they interacted with each other. Testing proved to be invaluable at every step of development. Even in the beginning, when functionality was minimal, testing showed us ways we could improve how the application functioned. By the end of development, we were able to develop and test all necessary user functionalities using Cypress.

Change Management Plan:

Purpose

This Change Management Plan will effectively manage the process of evaluating and implementing project change as intended by the team. The intended audience of this Plan would be the project team and project advisors.

Scope

The Change Management Process is organized by a series of phases: Request, Review, Approve, Implement, and Log. The Log phase runs concurrently along the rest of the phases in order to document all phases of the requested change.

Request

Changes can be requested by any member of the project team. If there is something that requires change, the requestee will notify the team that a Change Request (CR) should be reviewed. They can communicate the request through any means of team communication. In this phase, the requestee will also determine the priority of the change. The priority levels are as follows:

- **Critical:** The change being requested affects the project significantly and must be implemented immediately. A post-mortem of the change will follow.
- **High:** The change being requested affects the project significantly and should be reviewed as soon as possible.
- **Medium:** The change being requested affects the project moderately and should be reviewed at some point throughout the week.
- **Low:** The change being requested affects the project a little and can be reviewed whenever there is time.

Assigning a priority level determines the significance of change and can determine how the team should go about reviewing the change in the next phase.

Review

Once a change has been communicated and a priority level has been assigned, the team will determine how they would like to review the request. This review can happen in person, in a video call, or even through a chat communication medium. The CR must be reviewed by all members of the team in order for the change to be considered.

Approve

The change must be approved in order for it to be implemented. The entire team must come to a mutual agreement about the change, either meaning all team members say that they agree that the change can take place, or that they abstain from indicating one way or another. If a member of the team disagrees, that member can suggest a modification to the change, which would need to be reviewed by the entire team. For a change to be approved, the team must all be on the same page about the change. If one person disagrees, it is imperative that the team communicate and determine the best course of action regarding the CR.

Implement

Once the terms of the change are agreed upon, the team can then go about implementing the change. Work can then be delegated amongst team members to implement the change.

Log

All changes must be logged in a Change Log. The Change Log is essential to having all records of changes in one place as well as showing project advisors where the team is regarding large project changes. The Change Log will include the Change's date, description, priority level, status, and what is expected of the change afterward

A change status is determined by the following categories:

- Open: The change has been suggested and is waiting for review.
- In Review: The change is being reviewed by the team.
- Denied: The change will not be taking place.
- Approved: The change was approved, and work is being delegated.
- In Progress: The change is in the process of being implemented.
- Closed: The change is complete and has been implemented.

The change status will be updated throughout the phases if need be.

Change Log

For Change Log see Appendix B

Budget:

Table 4, below, represents the estimated budget required for the development of Stylebin. This budget was calculated with labor costs and operating costs in mind. Operating costs include items like internet access fees and hardware costs for our core team members, as well as any fees incurred by our external software vendors. These external software vendors will include Google Cloud Platform, Auth0, and SendGrid. Stylebin is projected to consume minimal resources which will allow us to operate within the free tier provided by most of our external software vendors.

Table 2: Stylebin Estimated Budget

	Rate Per/Hr	Work Effort (Hours)	1 X Costs	Ongoing Annual		
				Rate Per/Hr	Work Effort (Hours)	1 X Support Cost
Labor - Developers	20	0	\$ -	20	1125	\$ 22,500.00
Labor - QA	20	0	\$ -	20	750	\$ 15,000.00
Software - Operating System			\$ 1,400.00			\$ -
Software - Office 365			\$ -			\$ 360.00
Infrastructure - Firebase			\$ -			\$ -
Infrastructure - Heroku			\$ -			\$ 300.00
Services - Auth0			\$ -			\$ -
Services - SendGrid			\$ -			\$ -
Services - Home Internet Subscriptions			\$ -			\$ 3,000.00
Hardware - Staff Computers			\$ 9,550.00			\$ -
Hardware - Computer Peripherals			\$ 3,325.00			\$ -
TOTAL			\$ 14,275.00			\$ 41,160.00

Problems Encountered and Analysis of Problems Solved:

The following list recounts some of the major technical issues that we encountered over the course of developing Stylebin.

- Next.js Server-Side Rendering with Material-UI v4
 - The first major issue we encountered involved interactions between Next.js and Material-UI (MUI). Initially, we installed MUI v5, which requires a rendering engine called emotion. This completely changes how MUI components are styled in a way that would make exporting them more difficult. This prompted us to switch to MUI v4, a more widely used version of the Material-UI package. However, Next.js and specifically Next.js' server-side rendering was not compatible with the JSS styling used by MUI. This led to about a week of trying to inject the JSS into the SSR. After a few unsuccessful attempts, the decision was made to drop Next.js in favor of simplifying the use of MUI. This switch involved us recreating our codebase using the standard React environment.
- Issues Configuring GraphQL and Linking our MongoDB Instance
 - During the process of creating the Stylebin backend, we encountered issues linking the MongoDB Cloud Atlas instance to our express-GraphQL server. These issues were related to the network whitelisting on the side of MongoDB Cloud Atlas and were resolved by initializing the MongoDB instance within the repository, containerizing it with docker, and hosting it locally. These techniques will have to be adapted once we mature out of local hosting and begin to execute formal deployments of Stylebin.
- Unable to Configure Auth0 Test Account for Cypress Testing
 - For the testing process, we created a service account for Cypress to use while authenticating with Auth0. Cypress and Auth0 both recommend using HTTP POST commands to send credentials to log in, instead of typing in credentials manually. This is where we encountered several issues that we were unable to resolve. The first of these issues being security settings within Auth0. Even though we used the recommended settings, Cypress would throw multiple errors while attempting to log in. Another issue encountered was with the commands that cypress includes in their documentation to set up Auth0 authentication. Even when copied exactly, the compiler would throw errors because of the Typescript declarations that were used in the script. We were unable to find the documentation on how to configure Cypress using Typescript instead of JavaScript, which Cypress is written in, preventing us from using the provided documentation. We tried to modify the commands to force them to use JavaScript, but they failed to run properly. After spending two weeks trying to resolve these issues, we decided that it was not worth investing any more time into resolving this problem.
 - We revisited this issue during the second semester and were able to resolve the problem. After sitting down and reading through the technical aspects of Cypress login workflow, Justin was able to generate an access token using API calls with Auth0's management API. Our frontend also required some changes to take advantage of the access token by changing the default location that the Auth0 provider scanned for authorization information. After making these changes we

were able to run Cypress tests as authenticated test users in our development environment.

- Authentication issues between frontend and GraphQL
 - During one of the key integration sessions between the software developers on the team, we ran into an issue relating to the authentication of API requests going from our frontend to our backend. To briefly summarize the issue, we could not generate an Auth0 access token for use with the Apollo client (used to programmatically make API calls) since these providers were being called at the root level of our application. The order of operations meant that the Apollo client kept trying to initialize before the Auth0 provider had loaded. This issue was a major blocker and derailed our working session while we tried to resolve the problem. The solution to the problem involved placing the Apollo client initialization code into a separate JavaScript file as an exported function, grabbing the access token from within that function, and then initializing the Apollo client after the Auth0 provider.

Conclusion:

Stylebin has come a long way since its inception back in August 2021. We as a team have learned so much in this relatively short amount of time including important lessons learned and new skillsets and tools that we'll have in our arsenal for the rest of our careers.

Lessons Learned

The most important lesson we learned stemmed from our selection of technologies. From the beginning, we decided that Stylebin will run on the Next.js framework using Material-UI as our component library of choice. During our first development cycle, we noticed a compatibility issue between Next.js and Material-UI. Ruairi, one of our Product Engineers, spent time trying to troubleshoot and resolve this issue. For more information on this, please see the **Problems Encountered and Analysis of Problems Solved** section.

In the end, we decided that it was not worth the trouble trying to get Next.js and Material-UI working together stably, so we made the decision to move away from Next.js to a more stable vanilla React-based frontend. This issue cost us valuable time troubleshooting the problem and then recreating our code to work with the standard React framework. As we look back, we feel that it would have been better to research our chosen technologies more in-depth to discover any of these issues ahead of time.

Another important lesson we learned involved the initial development schedule that we defined when we started working on Stylebin. The initial schedule loosely defined application features and timelines to get those features implemented, however, it would have been beneficial for us to revisit the timeline after scoping out each feature to break them down into individual tasks. This would have allowed us to keep track of the progress made towards each overall feature, and plan for any delays/issues that appeared.

Skills Developed & Enhanced

During our time developing Stylebin, we have challenged ourselves to learn new skills as well as demonstrate and build upon our existing skillsets. We have most notably been developing our technological skills, as shown in the list below:

- Change Control
 - Git
 - GitHub
- Frontend Web Development
 - Next.js
 - React
 - Typescript
 - React Component Libraries (Such as Material-UI)
- Backend Web Development
 - NodeJS
 - Apollo Server
 - ExpressJS
 - GraphQL
- End-to-End Testing
 - Cypress
- Web Hosting
 - Google Firebase
 - Heroku

Additionally, we have been developing and strengthening our soft skills as well. We have worked hard this semester to enhance our communication skills so that we can collaborate effectively. Being a team of five, it can be difficult to stay connected and be productive when we are separated most of the time. The Stylebin team has overcome the boundaries set by COVID-19 and our geographic locations around Cincinnati by making use of collaboration tools such as Microsoft Teams, Discord, and Notion.

Furthermore, we have strengthened our project management skills through the different deliverables that are required for senior design. Senior design allowed us to gain valuable insight into the project management process outside of a normal classroom setting like building out budgets, creating change management guidelines, developing testing procedures, and more.

Finally, we have enhanced our product demonstration skills which will be invaluable to us over the course of our careers. Over the course of this semester, we were required to not only demonstrate our weekly progress to our advisor, but also explain what was happening and why it was important to Stylebin's development. Progress reports like these will be a common occurrence for any office job, so it is important that we develop these skills early so that we can excel in our careers. The final presentation was also important since we had to not only demonstrate our project to the instructors, but also to our peers. While giving large-scale presentations like these may not be very common in our careers, it is still a key skill that we will need to draw on at some point.

Appendix A

Testing Log

Table 3: Testing Logs

Date	Test Case	Test Number	User	Role	Expected Output	Actual Output	Pass/Fail
11/19/2021	5a	1	cypress@testing.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
11/19/2021	2a	1	cypress@testing.com	Workspace Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI but not reflected in DB	Fail
11/19/2021	5b	1	cypress@testing.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
11/19/2021	5c	1	cypress@testing.com	Workspace Admin	Style details and colors are shown	Style details and colors cannot be viewed	Fail
11/20/2021	2b	1	cypress@testing.com	Workspace Admin	Styles can be created, and colors can be chosen	New styles cannot be created	Fail
11/20/2021	2c	1	cypress@testing.com	Workspace Admin	Bin is added to UI and is reflected in DB	Bin Creation window is opened but a new bin cannot be created	Fail
11/20/2021	3a	1	cypress@testing.com	Workspace Admin	A workspace is modified in the UI and changes are reflected in the DB	Workspace is modified and a pop up says changes are saved, but no changes are reflected	Fail
11/20/2021	4a	1	cypress@testing.com	Workspace Admin	Components can be created and added to bins.	Button Components can be created but not saved in DB. Other types of components cannot be created.	Fail
11/23/2021	10a	1	cypress@testing.com	Workspace Admin	Workspaces are deleted in the UI and changes are	Delete button is pressed and a pop-up appears. The	Fail

					reflected in the DB	workspace is not deleted	
11/23/2021	10b	1	cypress@testing.com	Workspace Admin	Styles are deleted and removed from the Bin	Styles are unable to be deleted	Fail
11/23/2021	10c	1	cypress@testing.com	Workspace Admin	Bin is deleted and is reflected in the DB	Bin is deleted but is not reflected in the DB	Fail
11/23/2021	10d	1	cypress@testing.com	Workspace Admin	Components can be deleted and removed from the bin.	Components are unable to be deleted.	Fail
11/23/2021	3b	1	cypress@testing.com	Workspace Admin	Style details and colors can be changed	Style details cannot be viewed and are unable to be changed	Fail
11/23/2021	3d	1	cypress@testing.com	Workspace Admin	Components can be modified, and changes can be reflected in the DB	Components are unable to be modified	Fail
11/24/2021	1a	1	cypress@testing.com	Workspace Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
11/24/2021	1b	1	cypress@testing.com	Workspace Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
11/23/2021	3c	1	cypress@testing.com	Workspace Admin	Bins can be modified, and changes can be reflected in the DB	Bins are unable to be modified	Fail
1/6/2022	2a	2	cypress@testing.com	Workspace Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI but not reflected in DB	Fail
1/6/2022	2b	2	cypress@testing.com	Workspace Admin	Styles can be created, and colors can be chosen	New styles cannot be created	Fail
1/6/2022	2c	2	cypress@testing.com	Workspace Admin	Bin is added to UI and is reflected in DB	Bin Creation window is opened but a new bin cannot be created	Fail

1/19/2022	3a	2	cypress@testing.com	Workspace Admin	A workspace is modified in the UI and changes are reflected in the DB	Workspace is modified and a pop up says changes are saved, but no changes are reflected	Fail
1/19/2022	3b	2	cypress@testing.com	Workspace Admin	Style details and colors can be changed	Style details cannot be viewed and are unable to be changed	Fail
1/19/2022	3c	2	cypress@testing.com	Workspace Admin	Bins can be modified, and changes can be reflected in the DB	Bins are unable to be modified	Fail
1/19/2022	3d	2	cypress@testing.com	Workspace Admin	Components can be modified, and changes can be reflected in the DB	Components are unable to be modified	Fail
1/26/2022	5a	2	cypress@testing.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
1/26/2022	5b	2	cypress@testing.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
1/26/2022	1a	2	cypress@testing.com	Workspace Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
1/26/2022	1b	2	cypress@testing.com	Workspace Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
1/30.2022	10a	2	cypress@testing.com	Workspace Admin	Workspaces are deleted in the UI and changes are reflected in the DB	Delete button is pressed and a pop-up appears. The workspace is not deleted	Fail
1/30.2022	10b	2	cypress@testing.com	Workspace Admin	Styles are deleted and removed from the Bin	Styles are unable to be deleted	Fail
1/30.2022	10c	2	cypress@testing.com	Workspace Admin	Bin is deleted and is reflected in the DB	Bin is deleted but is not reflected in the DB	Fail
1/30.2022	10d	2	cypress@testing.com	Workspace Admin	Components can be deleted and removed from the bin.	Components are unable to be deleted.	Fail

2/9/2022	5a	3	cypress@testin.g.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
2/9/2022	5b	3	cypress@testin.g.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
2/9/2022	1a	3	cypress@testin.g.com	Workspace Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
2/9/2022	1b	3	cypress@testin.g.com	Workspace Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
2/16/2022	2a	3	cypress@testin.g.com	Workspace Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI but not reflected in DB	Fail
2/16/2022	2b	3	cypress@testin.g.com	Workspace Admin	Styles can be created, and colors can be chosen	New styles cannot be created	Fail
2/16/2022	2c	3	cypress@testin.g.com	Workspace Admin	Bin is added to UI and is reflected in DB	Bin Creation window is opened but a new bin cannot be created	Fail
2/19/2022	3a	3	cypress@testin.g.com	Workspace Admin	A workspace is modified in the UI and changes are reflected in the DB	Workspace is modified and a pop up says changes are saved, but no changes are reflected	Fail
2/19/2022	3b	3	cypress@testin.g.com	Workspace Admin	Style details and colors can be changed	Style details cannot be viewed and are unable to be changed	Fail
2/19/2022	3c	3	cypress@testin.g.com	Workspace Admin	Bins can be modified, and changes can be reflected in the DB	Bins are unable to be modified	Fail
2/19/2022	3d	3	cypress@testin.g.com	Workspace Admin	Components can be modified, and changes can be reflected in the DB	Components are unable to be modified	Fail
2/19/2022	5a	4	cypress@testin.g.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass

2/19/2022	5b	4	cypress@testing.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
2/19/2022	1a	4	cypress@testing.com	Workspace Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
2/19/2022	1b	4	cypress@testing.com	Workspace Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
2/19/2022	10a	3	cypress@testing.com	Workspace Admin	Workspaces are deleted in the UI and changes are reflected in the DB	Delete button is pressed and a pop-up appears. The workspace is not deleted	Fail
2/19/2022	10b	3	cypress@testing.com	Workspace Admin	Styles are deleted and removed from the Bin	Styles are unable to be deleted	Fail
2/19/2022	10c	3	cypress@testing.com	Workspace Admin	Bin is deleted and is reflected in the DB	Bin is deleted but is not reflected in the DB	Fail
2/19/2022	10d	3	cypress@testing.com	Workspace Admin	Components can be deleted and removed from the bin.	Components are unable to be deleted.	Fail
2/23/2022	1a	5	cypress@testing.com	Workspace Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
2/23/2022	1b	5	cypress@testing.com	Workspace Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
2/23/2022	2a	4	cypress@testing.com	Workspace Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI and is reflected in DB	Pass
2/23/2022	2b	4	cypress@testing.com	Workspace Admin	Styles can be created, and colors can be chosen	New styles cannot be created	Fail
2/23/2022	2c	4	cypress@testing.com	Workspace Admin	Bin is added to UI and is reflected in DB	Bin Creation window is opened and is reflected in the DB	Fail
3/2/2022	3a	4	cypress@testing.com	Workspace Admin	A workspace is modified in the UI and changes are reflected in the DB	Workspace is modified and a pop up says changes are saved, but no	Fail

						changes are reflected	
3/2/2022	3b	4	cypress@testing.com	Workspace Admin	Style details and colors can be changed	Style details cannot be viewed and are unable to be changed	Fail
3/2/2022	3c	4	cypress@testing.com	Workspace Admin	Bins can be modified, and changes can be reflected in the DB	Bins are unable to be modified	Fail
3/2/2022	3d	4	cypress@testing.com	Workspace Admin	Components can be modified, and changes can be reflected in the DB	Components are unable to be modified	Fail
3/2/2022	4a	2	cypress@testing.com	Workspace Admin	Components can be created and added to bins.	Button Components can be created but not saved in DB. Other types of components cannot be created.	Fail
3/2/2022	5a	5	cypress@testing.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
3/2/2022	5b	5	cypress@testing.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
3/2/2022	5c	2	cypress@testing.com	Workspace Admin	Style details and colors are shown	Style details and colors cannot be viewed	Fail
3/7/2022	6	1	cypress@testing.com	Workspace Admin	Workspaces can be exported	The export button exists, but nothing is exported.	Fail
3/7/2022	7b	1	cypress@testing.com	Workspace Admin	Users can be invited to a workspace	This functionality does not exist yet.	Fail
3/7/2022	10a	4	cypress@testing.com	Workspace Admin	Workspaces are deleted in the UI and changes are reflected in the DB	Delete button is pressed and a pop-up appears. The workspace is not deleted	Fail

3/7/2022	10b	4	cypress@testingg.com	Workspace Admin	Styles are deleted and removed from the Bin	Styles are unable to be deleted	Fail
3/7/2022	10c	4	cypress@testingg.com	Workspace Admin	Bin is deleted and is reflected in the DB	Bin is deleted but is not reflected in the DB	Fail
3/7/2022	10d	3	cypress@testingg.com	Workspace Admin	Components can be deleted and removed from the bin.	Components are unable to be deleted.	Fail
3/9/2022	1a	6	cypress@testingg.com	Workspace Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
3/9/2022	1b	6	cypress@testingg.com	Workspace Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
3/9/2022	2a	5	cypress@testingg.com	Workspace Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI and is reflected in DB	Pass
3/9/2022	5a	6	cypress@testingg.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
3/9/2022	5b	6	cypress@testingg.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
3/9/2022	10a	4	cypress@testingg.com	Workspace Admin	Workspaces are deleted in the UI and changes are reflected in the DB	Workspaces are deleted in the UI and changes are reflected in the DB	Pass
3/9/2022	10b	4	cypress@testingg.com	Workspace Admin	Styles are deleted and removed from the Bin	Styles are deleted and removed from the Bin	Pass
3/9/2022	10c	4	cypress@testingg.com	Workspace Admin	Bin is deleted and is reflected in the DB	Bin is deleted and is reflected in the DB	Pass
3/9/2022	10d	3	cypress@testingg.com	Workspace Admin	Components can be deleted and removed from the bin.	Components can be deleted and removed from the bin.	Pass
3/9/2022	6	2	cypress@testingg.com	Workspace Admin	Workspaces can be exported	The export button exists, but nothing is exported.	Fail
3/9/2022	7b	2	cypress@testingg.com	Workspace Admin	Users can be invited to a workspace	This functionality does not exist yet.	Fail

3/16/2022	1a	7	cypress@testin g.com	Workspa ce Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
3/16/2022	1b	7	cypress@testin g.com	Workspa ce Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
3/16/2022	2a	6	cypress@testin g.com	Workspa ce Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI and is reflected in DB	Pass
3/16/2022	5a	7	cypress@testin g.com	Workspa ce Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
3/16/2022	5b	7	cypress@testin g.com	Workspa ce Admin	Bin details and components are shown	Bin details and components are shown	Pass
3/16/2022	6	3	cypress@testin g.com	Workspa ce Admin	Workspaces can be exported	The export button exists, but nothing is exported.	Fail
3/16/2022	7b	2	cypress@testin g.com	Workspa ce Admin	Users can be invited to a workspace	This functionality does not exist yet.	Fail
3/23/2022	1a	5	cypress@testin g.com	Workspa ce Admin	User is logged in and taken to the workspaces page	User is logged in and taken to the workspaces page	Pass
3/23/2022	1b	5	cypress@testin g.com	Workspa ce Admin	User is logged out and taken to the sign-in page	User is logged out and taken to the sign-in page	Pass
3/23/2022	2a	5	cypress@testin g.com	Workspa ce Admin	Workspace is added to the UI and is reflected in DB	Workspace is added to UI and is reflected in DB	Pass
3/23/2022	2b	5	cypress@testin g.com	Workspa ce Admin	Styles can be created, and colors can be chosen	Styles can be created, and colors can be chosen	Pass
3/23/2022	2c	5	cypress@testin g.com	Workspa ce Admin	Bin is added to UI and is reflected in DB	Bin is added to UI and is reflected in DB	Pass
3/23/2022	3a	5	cypress@testin g.com	Workspa ce Admin	A workspace is modified in the UI and changes are reflected in the DB	A workspace is modified in the UI and changes are reflected in the DB. Sometimes does not work until the page is reloaded	Pass/ Fail

3/23/2022	3b	5	cypress@testing.com	Workspace Admin	Style details and colors can be changed	Style details and colors can be changed	Pass
3/23/2022	3c	5	cypress@testing.com	Workspace Admin	Bins can be modified, and changes can be reflected in the DB	Bins can be modified, and changes can be reflected in the DB	Pass
3/23/2022	3d	5	cypress@testing.com	Workspace Admin	Components can be modified, and changes can be reflected in the DB	Components can be modified, and changes can be reflected in the DB	Pass
3/23/2022	4a	4	cypress@testing.com	Workspace Admin	Components can be created and added to bins.	Components can be created and added to bins.	Pass
3/23/2022	5a	6	cypress@testing.com	Workspace Admin	Workspace details and bins are shown	Workspace details and bins are shown	Pass
3/23/2022	5b	6	cypress@testing.com	Workspace Admin	Bin details and components are shown	Bin details and components are shown	Pass
3/23/2022	5c	3	cypress@testing.com	Workspace Admin	Style details and colors are shown	Style details and colors are shown	Pass
3/23/2022	6	4	cypress@testing.com	Workspace Admin	Workspaces can be exported	Workspaces can be exported	Pass
3/23/2022	7b	4	cypress@testing.com	Workspace Admin	Users can be invited to a workspace	Users can be invited to a workspace	Pass
3/23/2022	10a	5	cypress@testing.com	Workspace Admin	Workspaces are deleted in the UI and changes are reflected in the DB	Workspaces are deleted in the UI and changes are reflected in the DB	Pass
3/23/2022	10b	5	cypress@testing.com	Workspace Admin	Styles are deleted and removed from the Bin	Styles are deleted and removed from the Bin	Pass
3/23/2022	10c	5	cypress@testing.com	Workspace Admin	Bin is deleted and is reflected in the DB	Bin is deleted and is reflected in the DB	Pass
3/23/2022	10d	4	cypress@testing.com	Workspace Admin	Components can be deleted and removed from the bin.	Components are unable to be deleted.	Pass

Appendix B

Change Log

Table 4: Change Log

Date	Title	Description	Priority	Status	Next Steps
10/12/2021	Account Chip (Auth0)	Adds a frontend account chip that will display user info while logged in and contain logout functionality.	Low	Closed	N/A
10/14/2021	Migrate project from Next.js to traditional React and Node	We encountered difficulties with Next.js' server-side rendering and made the decision to revert the project to a traditional create-react-app boilerplate and then an express backend with GraphQL	High	Closed	N/A
10/14/2021	Add Material-UI v4.12.3	This is the package that will be used for creating components and styling the site itself, and for the components to be exported as the main function of the site	High	Closed	N/A
10/14/2021	Add ESLint and prettier to the project	By adding ESLint and prettier, we were able to set up formatting rules that will help us write more consistent code. This should reduce bugs in the long run	Medium	Closed	Refine linting
10/15/2021	Initialize backend with Express and GraphQL	Initialized the backend repository with Express and GraphQL, and wrote the initial schema for workspaces and bins.	High	Closed	Connect backend to MongoDB
10/16/2021	Workspace Settings Menu	Adds a new frontend menu that holds any settings related to workspace management. This will contain any settings that a user will need to update in a	Medium	Closed	As more functionality is built into the application, more options will be built into the settings page.

		workspace (ex. Name, Icon, Permissions, etc.).			
10/28/2021	Finish workspace page functions and styling	The “workspaces” page is the main page of the site. Here you can select different workspaces and bins, selecting a bin will replace the bin container with a table	High	Closed	Connect workspace and bin data to the backend. Add functions to limit the amount of bins that can be created
11/04/2021	Component creation workflow	This is the basic implementation of the site’s main function. The stepper will walk the user through the steps of creating a component, and that component is saved as a frontend object	High	Closed	Hook up to a post route to send components to the backend, Create a list of attributes for each component type, these will be used to map questions on the second screen
11/11/2021	Added MongoDB initialization script	Created a script for initializing an instance of MongoDB.	High	Closed	Create a docker-compose script for containerizing our DB
11/12/2021	Created docker-compose script	Created a docker-compose script for containerizing the initialized MongoDB instance.	Medium	Closed	Write mutators and resolvers for our workspace and bin types
11/15/2021	Bin and Workspace creation	This is the secondary “organization” function of Stylebin. Users can now create workspaces to hold their bins, and bins to hold components	High	Closed	Post these to the backend Add icon selection functionality Link workspaces and user data
11/14/2021	Create bin and workspace resolvers and mutators	Created workspace and bin resolvers and mutators for basic CRUD operations.	Medium	Closed	N/A

11/20/2021	Basic Export Functionality	Adds backend functionality to dynamically export the different components saved in a bin. Also, adds a button to the frontend to call the backend and export the contents of the selected bin.	High	Closed	Once GraphQL and MongoDB have been configured, we will need to edit the export logic to make the export process dynamic.
Date	Title	Description	Priority	Status	Next Steps
1/30/2022	Create GraphQL Client with URQL	Created a GraphQL client that uses URQL as an application layer. This stack came by recommendation from a friend.	High	Closed	Rewrite mutators and queries to use this new tech stack
1/30/2022	Initial modular component creation	Setup a modal, which prompts the user to answer generated questions based on the selected component type	High	Closed	Post to the backend on submit
2/12/2022	Implement backend auth	Require that API users be authenticated with our Stylebin application in Auth0. Deny access if not authenticated.	High	Closed	Test frontend functionality when interacting with the API
2/13/2022	Implement Type-GraphQL as an ORM	Allows our GraphQL data models to be applied to a database structure. Minimizes the amount of code needed to create new entities/columns or attributes/rows	Medium	Closed	Apply database configuration and write data models
2/13/2022	Revised entities for Workspace and Bin	Re-wrote the Workspace and Bin entities to use Type-GraphQL as an ORM. This merges our GraphQL data schema and our typescript entities into one entity.	High	Closed	Begin posting data to a database and test mutators/queries
2/20/2022	Migrated from MongoDB to SQLite	Added configuring for an SQLite database to our ORM config. This will allow us to have data	High	Closed	Project Poster Draft

		persistence in our demos.				
2/20/2022	Require auth to access all frontend routes	This change requires that a user be logged in to access our application.	Medium	Closed	Update tests to include auth step	Cypress tests to include auth step
2/27/2022	Created the first draft of the Stylebin poster for the in-class review session.	Created a new Figma document and imported our team's style library. Used those common assets to populate a poster frame that meets the dimensions discussed during the lecture.	High	Closed	Revise Draft	Poster
2/30/2022	Revised Technical Diagram and Poster Copy	Revised the list of technologies on the Stylebin poster and replaced it with a simplified technical diagram. This will allow us to have conversations about the technology during our presentation without having to get too "in the weeds" about granular details.	Medium	Closed	Final Poster Draft	
3/1/2022	Create front-end routes	Creates Apollo/GraphQL routes, that will be used to post and retrieve data from the backend	Critical	Closed	Hook up front-end routes to backend	
3/9/2022	Workspace collaboration permissions	Merged functionality for shared workspaces (manual sharing at this point)	Medium	Closed	Implement functionality for users to invite other users through the UI	
3/14/2022	Stylebin Poster Final Draft	Incorporated the new assets that I had been designing and the feedback received during the poster peer-review session to create our poster's final draft.	High	Closed	In-Class Presentation!	

3/20/2022	Modular Component Creation	Component creation modal now forms an object and posts it to the backend	Critical	Closed	Create more Component types
3/20/2022	Finalize component export functionality	Implement backend routes to export all components and styling info for a particular bin.	Critical	Closed	Testing to ensure the content provided by the frontend can be exported as expected.
3/26/2022	Workspace invitations via UI	Implement an invitation workflow to allow users to invite users to their workspace.	High	Closed	Implement a screen to show active invitations and current permissions on a workspace
3/27/2022	Add React Color Picker	React color picker is now used to select bin colors	High	Closed	Style the picker to fit the Stylebin theme
3/29/2022	Export bug fixes	Bug fixes for the export process	Critical	Closed	Continue testing the export process to ensure functionality.