

# Creating the new face of Buildertrend

Final Report

**Team 21**

Buildertrend

Md Maruf Ahamed

Aashray Mehta - Team Lead

Catherine Samatas - Meeting Facilitator

Hana Franck - Meeting Scribe

Kyle Weinert - Design Engineer

Mohamed AlMansoori - Test Engineer

Saoud Alkaabi - Report Manager

**[sdmay21-21@iastate.edu](mailto:sdmay21-21@iastate.edu)**

**<http://sdmay21-21.sd.ece.iastate.edu/>**

## Table of Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Acknowledgement	5
1.2 Problem and Project Statement	5
1.2.1 General Problem Statement	5
1.2.2 General Solution Approach	5
1.3 Operational Environment	5
1.4 Project Evolution from CPRE 491	5
<b>2 Requirements</b>	<b>6</b>
2.1 General Requirements	6
2.3 Report List	6
2.4 Report Details	6
2.5 Leads and Cluster Map	6
2.6 Users	6
2.7 Use Cases	7
<b>3 Standards</b>	<b>8</b>
3.1 Buildertrend's Company Defined Standards	8
3.2 IEEE Software Development Standards	8
<b>4 Engineering Constraints</b>	<b>9</b>
4.1 Testing Constraints	9
4.2 Team Meeting Constraints	9
4.3 Schedule Constraints	9
4.4 Technical Constraints	9
4.5 Security Concerns	9
<b>5 Implementation Details</b>	<b>10</b>
5.1 Development Process	10
5.2 Report List	10
5.3 Report Details	10
5.4 Leads Map	10
5.5 Component Diagrams	12
<b>6 Testing Process and Results</b>	<b>13</b>
6.1 Process	13
6.2 Acceptance Testing	13
6.2.1 Acceptance testing for Report lists	13
6.2.2 Acceptance Testing for Report Details	13
6.2.3 Acceptance Testing for Leads Map	13

6.5 Results	14
<b>Appendix I - Operation Manual</b>	<b>15</b>
<b>Appendix II - Alternative Designs</b>	<b>17</b>
<b>Appendix III - Code</b>	<b>18</b>

### **List of Figures**

Figure 1: Use Case Diagram	6
Figure 2: Agile Development Diagram	9
Figure 3: Report List and Report Detail Component Diagram	11
Figure 4: Leads Map and Cluster Map Component Diagram	11
Figure 5: User Flow Graph	15
Figure 6: Original Report List and Report Detail Component Diagram	16

# 1 Introduction

## 1.1 Acknowledgement

This project is sponsored by Buildertrend. Our team acknowledges the technical assistance of Grant Fishburn and the management guidance of Gina Saccoman. Our team would also like to recognize our faculty advisor, Md Maruf Ahamed.

## 1.2 Problem and Project Statement

### 1.2.1 General Problem Statement

Buildertrend recently identified the need to transition application features from ASP.NET WebForms to React in order to improve application stability and reliability. This transition is also needed to simplify the development process for future features. As Buildertrend continues to grow, this transition will provide the company with the speed and efficiency to remain competitive within the industry.

### 1.2.2 General Solution Approach

Senior Design Team 21 was responsible for the conversion of three features using the latest version of React. Transitioning these features to React will provide Buildertrend with the speed and efficiency needed as the company continues to grow. The team added the React components made to Buildertrend's style guide tool, Storybook. This allows for the reuse of the components that the team designed.

## 1.3 Operational Environment

Clients of buildertrend use the buildertrend software to streamline their building operations as contractors and homeowners. The pages we are building will be used on online applications that can be accessed via desktops and mobile phones.

## 1.4 Project Evolution from CPRE 491

Our project has faced a few changes since the initial plans created during 491. These changes include the following:

- Project requirements shifted
- Buildertrends coding standards were added to our project
- Revision of the design and hierarchy of Report Details

When we met with our client, Buildertrend, in August 2020, they provided us with three features to implement for our year-long project. These features were Report List, Report Details, and Workday Exceptions. In December 2020, we were told that the requirements for our project had shifted; we were no longer responsible for implementing Workday Exceptions. Instead, we would be working on features called Leads Map and Cluster Map. This led to us rewriting the plans and requirements established in 491. Around the same time our client provided us with their company coding standards, this meant we had to go through the code we had written in 491 and update it to comply with the company's standards. We ended up spending a month or so doing this, which shifted our timeline for 492. As we were updating Report Details to comply with company standards, we decided that creating components based on chart type would better modularize our code and help Buildertrend's software team implement future reports faster.

## 2 Requirements

### 2.1 General Requirements

- All pages should be able to interface with the Buildertrend API
- All pages should match the current Buildertrend UI scheme
- All pages and features should be written in React
- All features should be well documented

### 2.3 Report List

- Report List should use the plop list template
- Be able to filter reports based on category
- Breadcrumb trails should be displayed when filtered by a category
- The user should have the ability to favorite different reports within Report List
- Display all reports in a tile and graphic format
- Tile has a favorite star button that adds the report to favorites, without reloading the page
- The user should be able to categorize reports into three predetermined categories: Sales, Project Management, and Finance
- Report list tiles should display images of the highcharts
- On the selection of a report, users should be led to a report details page corresponding to the selected report
- Integrate the report list to Buildertrend's web application.

### 2.4 Report Details

- Create new React components for each of the old Webform report types
- Report details should be implemented into the application
- Update existing components to work with new report types
- The report component should be able to handle all of the different types of reports
- Report details should be implemented with Highcharts
- Individual reports should be broken down into components as shown in the pre-implemented Work in Progress Report

### 2.5 Leads and Cluster Map

- Leads map should be a new React Component
- The grid template should be used as the base for Leads Map
- A tab functionality should load the Leads Map and Cluster Map within the page tabs
- Use the LeadTab component for other required page tabs
- Map tabs should use BTTabs and BTTabPane Components
- A user should be able to select a New Lead Opportunity Button
- Hovering over a row should make the map pin bounce
- A View Button should scope down to a pin's location
- The New Lead Opportunity Button should allow users to create new leads
- Integrate the new Leads and Cluster Maps into Buildertrend's web application

### 2.6 Users

Being that Buildertrend is a construction management company, the primary users of this project are builders, contractors, and their clients. The features, Report List, Report Details, and Leads Map, will be used primarily by project managers. Parts of the Report Details feature may be used by the project managers' employees with the managers' approval.

## 2.7 Use Cases

The use cases for both project managers and workers are shown below in Figure 1.

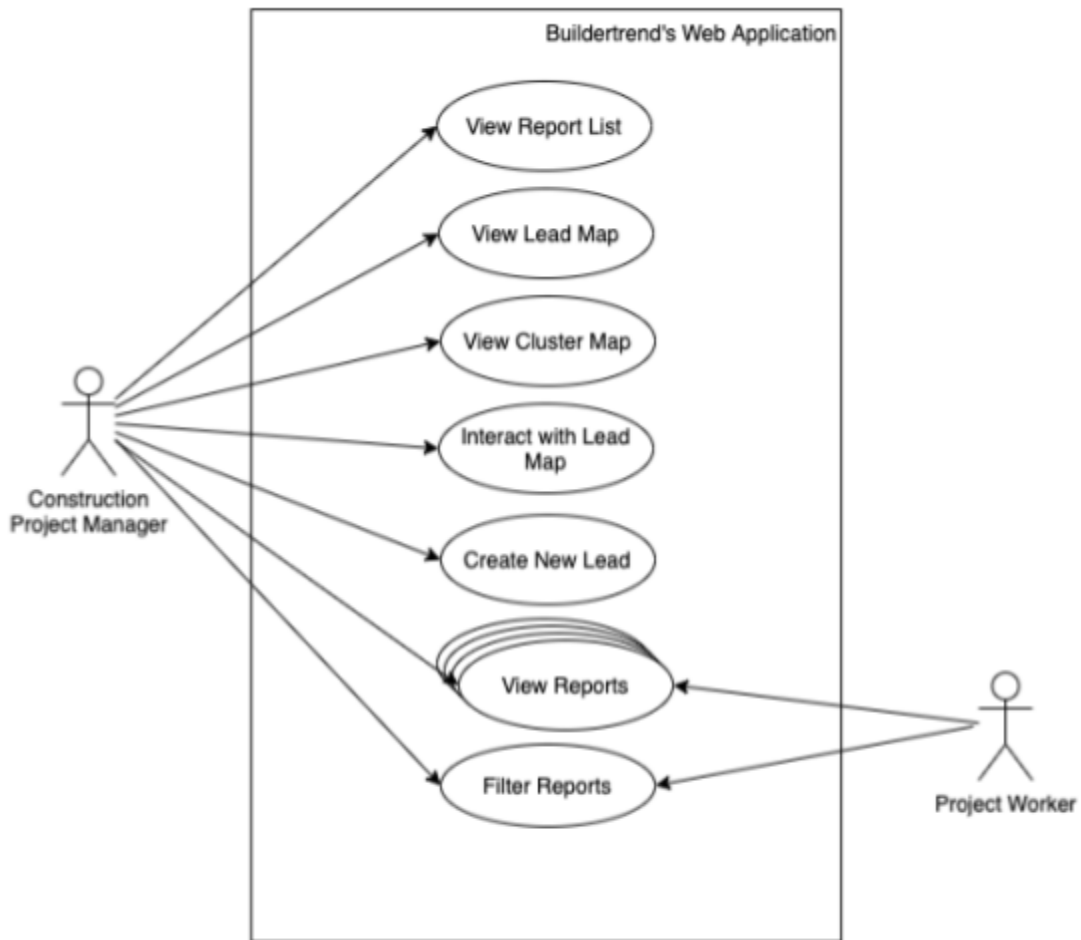


Figure 1: Use Case Diagram for Report List, Report Details, and Leads Map

## 3 Standards

### 3.1 Buildertrend's Company Defined Standards

Buildertrend has defined Component Standards, Styling Standards, and React Modal Standards that we used throughout our project. With these standards in place, we were able to write code that blended seamlessly with Buildertrend's other components and implementations. Using these standards we were able to confirm that our code would create web features that functioned and appeared as expected and would allow our code to be easily readable by Buildertrend's engineers.

Other standards and practices used by Buildertrend that the team confirmed included:

- Using TFS for code sharing
- Using AWS and virtual machines
- The use of Microsoft Teams and Outlook for all communication
- Buildertrend's design guidelines

As a software-focused project, we also used the following standards and practices:

- Version control with Git
- DevOps Model

### 3.2 IEEE Software Development Standards

#### [ISO/IEC/IEEE 12207 - Systems and Software Engineering - Software Lifecycle Processes](#)

This standard defines processes that are a part of software lifecycles. The processes that are a part of software life cycles include agreement processes, organizational project-enabling processes, technical management processes, and technical processes. The defined standards for each process ensure that the entire project runs smoothly and the outcome is what the stakeholders expect. Agreement processes organize agreements between acquirers and suppliers, which includes determining who is responsible for the organization and technical functions of the project. Organizational project-enabling processes confirm that the stakeholders' needs will be met. Technical management processes manage the allocation of resources and manage the technical team of the project. Technical processes are applied to create the model, operational product, or the expected outcomes of the project.

We used this standard to ensure that every stage of our software development project ran smoothly and met our client's expectations. This standard provided guidelines for communicating with our client, so their needs were met and our team was provided with clear requirements.

#### [IEEE/ISO/IEC 16085 - Systems and software engineering - Life cycle processes - Risk management](#)

This standard outlines a continuous process for risk management. It is intended to define a process for risk management that projects can use to ensure proper risk management. The process can be used with many different risk management and mitigation techniques. The standard lays out activities that should be completed as a part of the risk management process. These activities include:

- a. Plan and implement risk management
- b. Manage the project risk profile
- c. Perform risk analysis
- d. Perform risk monitoring
- e. Perform risk treatment
- f. Evaluate the risk management process

With any project, there are risks. Our senior design team used this standard to have clearly defined processes that would help manage and mitigate risks. Early on in the project, we defined potential risks and defined a risk mitigation strategy. In doing so we were able to complete the project on time even with major requirement changes and minimal knowledge of the coding languages used.

## 4 Engineering Constraints

Many constraints were placed on this project, which took away from some design freedom. The constraints we faced during the course of this project are listed below.

### 4.1 Testing Constraints

- The team did not have access to real users of the Buildertrend software
- User testing was done by the team and through demonstrations with the client during the biweekly meeting and milestone meetings
- The team has access older version of the software missing over 6 months of updates

### 4.2 Team Meeting Constraints

- Team members are located in different time zones.
- Adds constraints partnered work and in-person work

### 4.3 Schedule Constraints

- The project had to be completed within two semesters excluding university breaks

### 4.4 Technical Constraints

- Required to create React components
- Required to use Buildertrend's predefined templates if possible
- Required to use Buildertrend's current API, and could not request API changes
- Required to use Storybook to visualize the features
- Required to use Google Maps API

### 4.5 Security Concerns

Since the features we implemented are embedded in Buildertrend's web application and we did not update any API calls we do not have any security concerns.



## 5 Implementation Details

### 5.1 Development Process

While completing the three features, we followed an agile process similar to the process shown in Figure 2. We started in a requirements phase, where we worked with Buildertrend to define the requirements for each feature. The team then created tickets based on the recorded requirements. This led to the development phase where we began to implement the features. After some time, we began our testing phase, where we completed acceptance tests. At the end of the testing phase, we would host a demo with our company's point of contact. After each demo and round of testing, we re-entered the design phase and made adjustments according to the feedback provided during the demo. Now that all three features have been completed, Buildertrend's engineers are working to deploy our features to their active website.

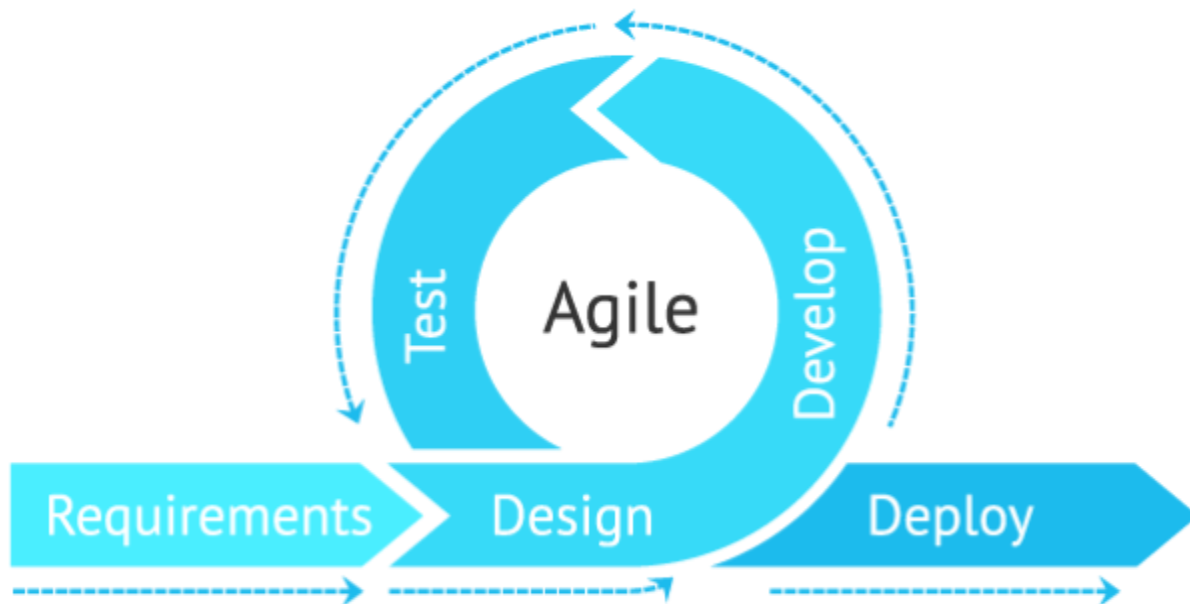


Figure 2: The agile development process.

### 5.2 Report List

The report list team started by creating a listing page and segmenting the page into the established four topics: favorites, sales, project management and financial. After the general layout was established, the team focused on populating the categories with the appropriate tiles. A breadcrumb trail was added when filtering down to a specific category. The team then worked on the functionality of choosing favorites. The general functionality of the report lists was tested manually through the use of the Storybook. The report list team ended the semester by connecting the tiles to their corresponding report detail pages and verifying the functionality.

### 5.3 Report Details

The Report Details pages were implemented using the Highcharts library and TypeScript. Each details page passes information to a react component that the team created to render a specific type of chart. Each type of chart was created with the Highcharts library and TypeScript. The code for the chart components can be found in Appendix III. To embed our charts into Buildertrend's application, we edited the file that determines which type of report and filter to render based on the information provided in the JSON response. Our edits included adding case statements for each of the report types that we implemented.

To add our components to Storybook we had to implement mock JSON files and add our components to the file that generates the stories.

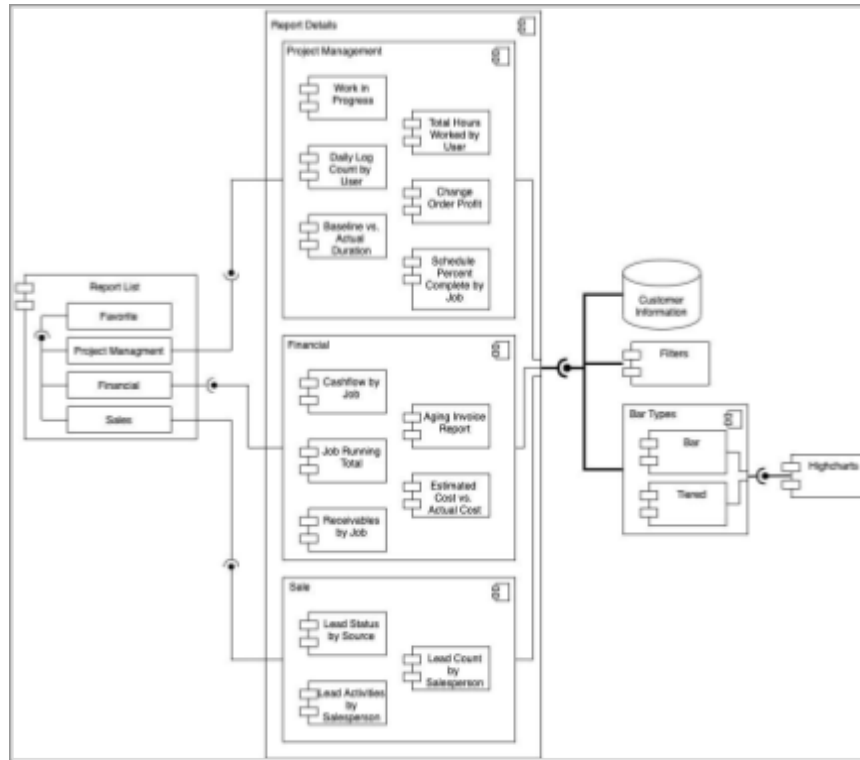
#### **5.4 Leads Map**

The Leads Map page was implemented using TypeScript, React, and many components built by Buildertrend. The Leads Map page contains two tabs for the two types of maps. These tabs are created using the React component LeadTab. When the page first loads, it pulls in data from the Buildertrend API using our API handler. This data is then passed to the component MapTab. MapTab parses the rows of data and creates objects that the page can display. The MapTab component is used to create the page under both of the tabs. By passing in parameters to the component we can decide how the page will render. For example, we can cluster the pins on the map to make it easier to look at, or we can display a scrollable list of all leads.

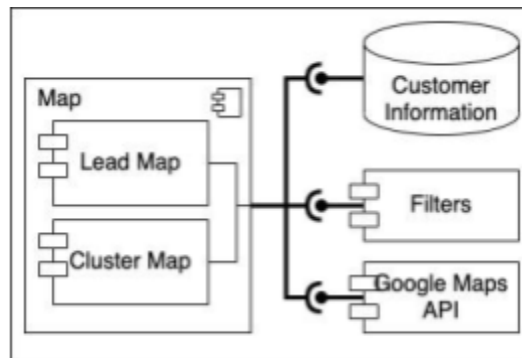
Storybook was also heavily used to test the layout of Leads Map. To make that work, we created more mock JSON files and had the page pull from that when it detected it was using Storybook.

### 5.5 Component Diagrams

The component diagrams for Report List, Report Details, Leads Map, and Cluster Map are provided below in Figure 2 and Figure 3.



**Figure 3 :** Report List and Report Detail Component Diagram



**Figure 4 :** Leads and Cluster Map Component Diagram

## 6 Testing Process and Results

### 6.1 Process

Buildertrend has a test environment that runs the application from the perspective of the user. We used this test environment to determine if the requirements were met. The acceptance tests we used are included below. During our milestone meetings, we used the test environment to demo the features we implemented. This allowed the client to provide feedback and request any changes.

### 6.2 Acceptance Testing

#### 6.2.1 Acceptance testing for Report lists

- When a star button is clicked, the star button turns yellow and adds the card to the favorites section.
  - To perform the acceptance test, we clicked on the star button of each tile and verified that the star turned yellow and the tile was added to the favorites section of the lists page.
- When view all of a certain type (sales, financial, project management, etc.) is clicked, the page repopulates with all tiles of the selected type
  - To perform the acceptance test, we clicked on view all for each card category and verified that the page repopulated with the tiles of the selected type.
- When filtered to a category, Breadcrumb should allow the user to navigate back to the reports page.
  - To Perform the acceptance test, we filtered by each category and clicked on the reports button in the breadcrumb and verified that it brought the user back to the Reports Lists page.

#### 6.2.2 Acceptance Testing for Report Details

- When a report card is selected, the user is directed to the matching report details page.
  - To perform the acceptance test, we clicked on each report card and verified that the correct details page loaded.
- A filter and a highchart is displayed on each report details page.
  - To perform the acceptance test, we navigated to each report details page and verified that each page contained a filter and highchart that matched the old implementation.
- The user can update the chart data by using a filter.
  - To perform the acceptance test, we applied different filters and verified that the chart updated to reflect the filter settings.
- The available filters match the filters in the old implementation.
  - To perform the acceptance test, we opened each filter and compared the displayed options with the old implementation options.
  - We tested the functionality of the filters to make sure that the connection was successful.
- The highchart data reflects the current data stored in the database.
  - To perform the acceptance test, we verified that the chart data matched the database data.

#### 6.2.3 Acceptance Testing for Leads Map

- When hovering over a row the corresponding map pin should bounce.
  - To perform the acceptance test, we clicked on a few rows and verified that the corresponding pin was bouncing.
- When the view button is selected the map should scope down to the corresponding pin's location
  - To perform the acceptance test, we clicked on a few view buttons and verified that the map scoped down to the corresponding pin and pin location.
- When the new lead button is clicked the user should be able to create a new lead

- To perform the acceptance test, we clicked on the new lead button and filled in the information for a new lead and verified that the lead was added to the map.
- The available filters match the filters in the old implementation.
  - To perform the acceptance test, we opened each filter and compared the displayed options with the old implementation options.
- When the filter is used in the leads map the pins reflect the updated data.
  - To perform the acceptance test, We tested the functionality of the filters to make sure that the connection was successful.
- When the cluster tab is pressed a cluster map with the equivalent pins appears.
  - To perform the acceptance test, we took note of the number of pins per area, then pressed the cluster tab. Finally verifying the numbers are the same.
- When the filter is used in the cluster map also reflects the updated data.
  - To perform the acceptance test, we first set filters in the leads map and took note of the number of pins per area. Then we selected the cluster tab. Finally verifying the numbers are the same.

## 6.5 Results

All acceptance tests were passed for Report List and Report Details and our clients were satisfied with the outcome.

## Appendix I - Operation Manual

Instructions on how to view and test our features within Buildertrend's web application.

Step 1: Open the solution in Visual Studio

When opening the solution, make sure to open the BuilderTREND.sln

- BTNet.ReactInterns -> BuilderTREND.sln

Step 2: Run NPM

Before running the solution, you need to run npm start at BT\_WebApp and Clients.App.

1. BT\_WebApp
  - Open the Developer Command Prompt in Visual Studio
    - View -> Terminal
  - Change the directory to BT\_WebApp cd BT\_WebApp
  - Run the npm start command
    - Note: If you haven't run npm install yet, you will get errors
2. Clients.App
  - Open the workspace in Visual Studio Code
  - Under the NPM Scripts section in the explorer on the left, select start

Step 3: Run the Solution in Visual Studio

1. Set the IIS Express bitness to x64 in BuilderTrend\_Web properties
  - Right-click BuilderTrend\_WEB -> Properties -> Web -> Set bitness to x64
2. Set the Visual Studio option to use 64-bit IIS Express
  - Tools -> Options -> Projects and Solutions -> Web Projects -> Check "Use the 64-bit version of IIS Express for web sites and projects"
3. Run the solution using the green play button. This should open the Buildertrend login page.

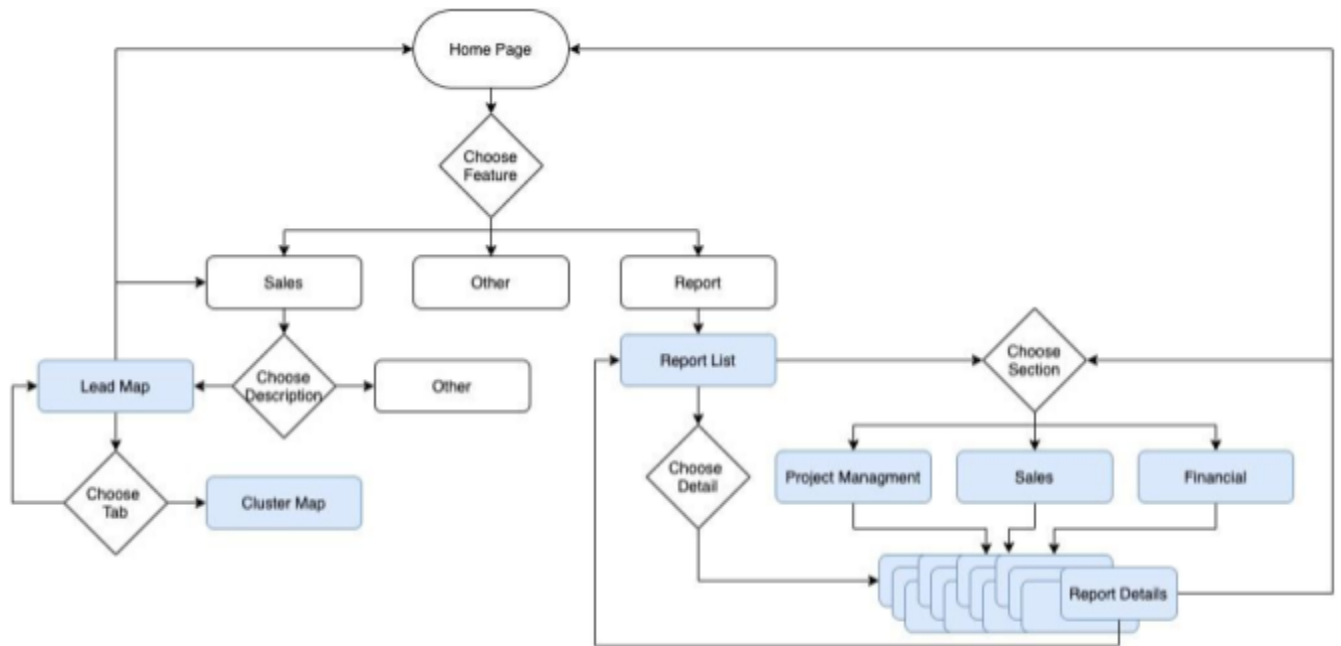
Instructions on how to view and test components in Storybook

Step 1: Run Storybook in visual studio code

1. Open the workspace in Visual Studio Code
2. Under the NPM Scripts section in the explorer on the left, select storybook
  - This should open Storybook in a browser

## Navigation

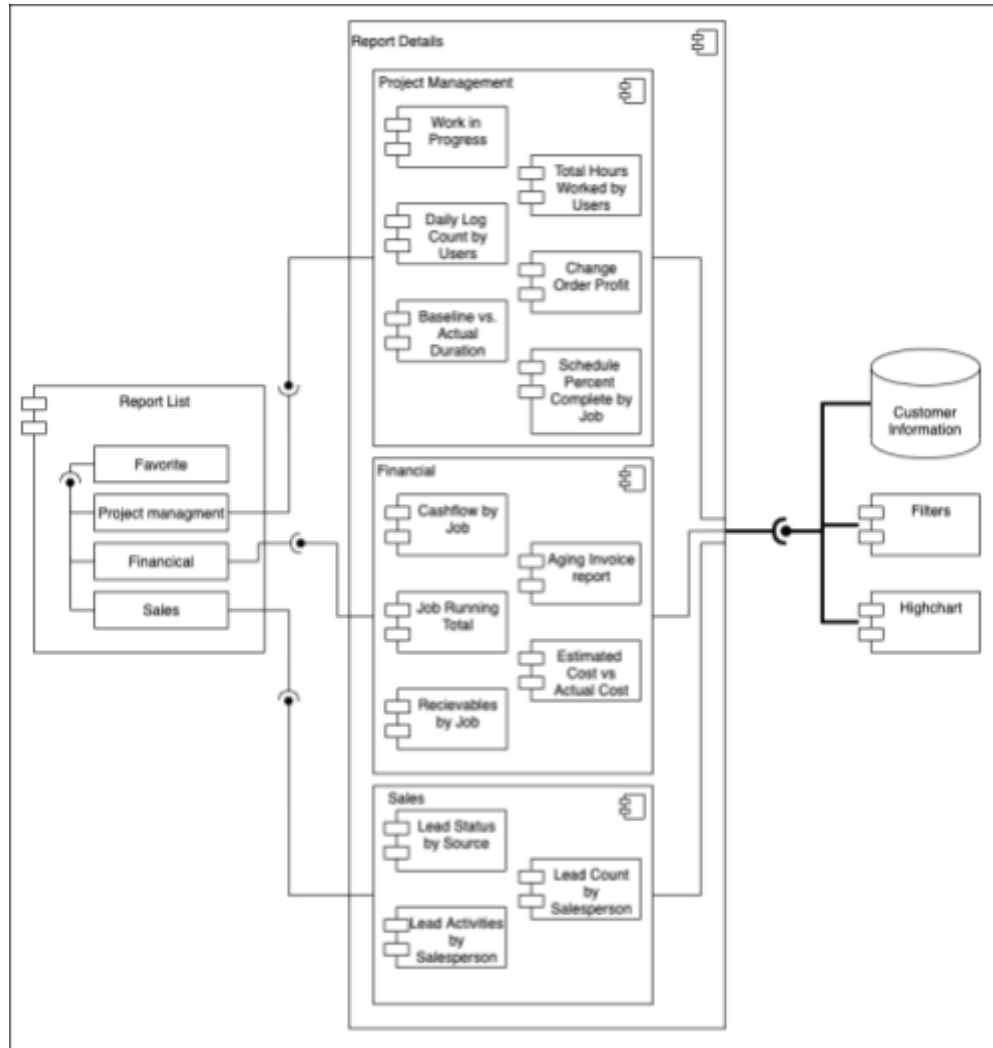
Figure 4 below provides an overview of how to navigate to our features from Buildertrend's homepage.



**Figure 5 : User Flow Graph**

## Appendix II - Alternative Designs

In a previous design, we planned to implement each Report Details Page individually. After taking a closer look we determined that we could reuse code by creating general react components for chart types that were shared by multiple details pages. Below is our alternative component diagram.



**Figure 6:** Original Lead and Cluster Map Component Diagram

To learn more about our alternative designs please visit our design document drafts:

<https://sdmay21-21.sd.ece.iastate.edu/docs.html>



## **Appendix III - Code**

Since an NDA was signed, the team can only share code that has been approved by Buildertrend.

Link to code:

<https://sdmay21-21.sd.ece.iastate.edu/docs.html>