

Creating the new face of Buildertrend

Design Document

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

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

Executive Summary

Development Standards & Practices Used

As a software-focused project we will be using the following standards and practices:

- Version control with Git
- DevOps Model

The project team will be conforming to the standards and practices used by buildertrend, including, but not limited to:

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

Summary of Requirements

- Migrate the report list feature from ASP.NET to React
- Migrate the report details feature from ASP.NET to React
- Migrate the workday exceptions feature from ASP.NET to React
- New React code should be modular
- The new React code should be well documented

Applicable Courses from Iowa State University Curriculum

- | | |
|------------|------------|
| - ComS 227 | - ComS 319 |
| - ComS 228 | - SE 339 |
| - SE 329 | - SE 396 |
| - ComS 309 | |

New Skills/Knowledge Acquired

Hard Skills

- TypeScript
- React
- Microsoft Visual Studio and Microsoft Visual Studio Code
- ASP.NET Web Forms

Soft Skills

- Effective Communication
- Organization
- Team management
- Adaptability
- Collaboration

Table of Contents

1.1	Acknowledgement	5
1.2	Problem and Project Statement	5
1.3	Operational Environment	5
1.4	Requirements	5
1.5	Intended Users and Uses	6
1.6	Assumptions and Limitations	7
2	Project Plan	8
2.1	Task Decomposition	8
2.2	Risks And Risk Management/Mitigation	9
2.4	Project Timeline/Schedule	14
3	Design	19
3.1	Previous Work And Literature	19
3.2	Design Thinking	19
3.3	Proposed Design	19
3.4	Technology Considerations	19
3.5	Design Analysis	19
3.7	Design Plan	20
4	Testing	23
4.1	Unit Testing	23
4.2	Interface Testing	23
4.3	Acceptance Testing	23
4.4	Results	24
5.1	Implementation Semester 1	24

6 Closing Material	25
6.1 Conclusion	25
6.2 References	25

List of figures/tables/symbols/definitions

Figure 1: Risk Analysis Table	9
Figure 2: Risk Map	10
Figure 3: Risk Map Levels	11
Figure 4: Risk Occurrence Levels	11
Figure 5: Impact Level Table	12
Figure 6: Metrics of Subtasks	13
Figure 7: Gantt Chart 1	14
Figure 8: Gantt Chart 2 Semester 1 (Fall 2020)	14
Figure 9: Gantt Chart 2 Semester 2 (spring 2021)	15
Figure 10: Gantt Chart Legend	16
Figure 11: Personnel Effort Requirements table	17
Figure 12: The agile development process	19
Figure 13: Workflow of user	20
Figure 14: Component Diagram for Report List and Report Details	21
Figure 15: Component Diagram Workday Exception	22

1 Introduction

1.1 Acknowledgement

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

1.2 Problem and Project Statement

General problem statement - Buildertrend is moving to a service once model where all the business logic and api are written once for web and native clients. Buildertrend has identified the need to convert old ASP.NET code into React code to improve stability and performance.

General Solution approach - Senior design team 21 is responsible for the conversion of pages using the latest version of react. The minimum scope is three detail page rewrites. The team will be adding components to the Buildertrend style guide tool, Storybook. Tests will be written using Jest and Enzyme. The team will work with Buildertrend provided documentation and their codebase away from buildertrend main branches. The complete pages will go live at the end of the senior design 492.

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 Requirements

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

Report List

- Report List should use the plop list template
- Be able to filter reports based on 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 should include elements from Highcharts in the display of the tiles.
- On the selection of a report, users should be led to a report details page corresponding to the selected report

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

Workday Exceptions

- Create React components for the modal and sub-modals
 - Plop details should be used as a base
- The modal needs to be implemented into the application using ReactDialogHandler
- The New Workday Exception Category Modal should open when the user clicks the **Add** button on the **Category** line
- The input field for the New Workday Exception Category should be a text box, where the user can enter a title for the Workday Exception
- The save button within the New Workday Exception Category Modal should save the new Workday Exception Category and select the category for the current Workday Exception
- The Edit Workday Exception Category Modal should open when a user clicks the **Edit** button on the **Category** line
- Within the Edit Workday Category Modal, a warning should appear when the user attempts to delete a category that is not in-use
- Within the Edit Workday Category Modal, a warning should appear when the user attempts to Edit a Default Category
- The save button within the Edit Workday Category Modal should save the new Workday Exception Category and selects it as the category for the current Workday Exception
- The delete button within the Edit Workday Category Modal should open a delete dialog depending whether the category is in use
 - If the category is in use when the user clicks **OK** the Category should be deleted and the Workday Exception Category Editor modal should be closed and they Category dropdown should be set to “Select an Option”
 - If the category is not in use when the user clicks **OK** nothing happens
- The user input field for the Edit Workday Category modal should be a text box, where the user can enter a title for the Workday Category
- If the user attempts to edit a default category a message will appear stating that the user is unable to edit default categories

1.5 Intended Users and Uses

Intended Users:

Being that Buildertrend is a construction management company, the primary users of this project are construction company workers and their employees. The report features and workday exception would mainly be used by managers and above. Reports lists (and by association report details) displays information on sales, project management, and financial reporting. This means the intended user job title/type can vary depending on the size of the company. For example, if the client company is large, individual departments such as sales and marketing will be using the reports in addition to the expected project manager. If the client company is one, the smaller side, the intended users for all the reports, can be a single manager or owner. Workday exceptions has a more limited user base due to its intended use (see below).

Uses:

Report List

- Display all reports in a tile and graphic format
- Reports are pre-organized into three categories
 - Sales
 - Project management
 - finance

- Allow for the organization of favorited reports based on the user input.
- On the selection of a report, users are led to a report details page corresponding to the selected report.

Report Details

- Displays a graph relevant to the type of report
- Filtering options are available for the graph

Workday Exceptions

- Allows managers to create workday exceptions
- The form allows many specifications, including:
 - Title
 - Category
 - Date specifications
 - Projects for exception
 - Notes
- Information inputted is reflected in the project management schedule calendar

1.6 Assumptions and Limitations

Assumptions

- A minimum of three features of the Buildertrend will be complete over the next two semester
 - Report List
 - Report Details
 - Workday Exceptions
- All React features implemented will be identical to Webform implementation of the features
 - This includes functionality such as buttons, organization features, warnings and redirects
- Story Book will be used for format and color matches
 - This ensures that react features are cohesive with pre-existing features
- Features will be implemented with quality code. The code will be well documented and easy to follow
 - The code will follow an open-closed programming principle.
 - Well documented
 - Names/Titles (variable and function names) within the code will be self-explanatory and follow a snake case format

Limitations

- User testing
 - The team does not have access to real users of the Buildertrend software
 - User testing will be done only by Team and through demonstration with BT point of contact and BT shareholders during the biweekly meeting and milestone meetings
- Team member geographical constraints
 - Team members are located in different time zones.
 - Adds constraints partnered work and in-person work
- Schedule
 - The project should be done within the scope of two semesters
 - This excludes university breaks (as of now)

1.7 Expected End Product and Deliverables

At the end of our senior design year, our team will have delivered three web pages for Buildertrend. The pages are report lists, report details, and workday exception pages. For the documentation, each code will have commented within the code. Each Web page should look and function in the same way described by our client. We expect to deliver two of these pages at the end of the Fall 2020 semester; however, we might have to extend the delivery date for one of these pages since this is a short semester. If everything went well, the team will deliver the report list and report details pages in November 2020. Lastly, the team will deploy the last page and the final product in March 2021.

2 Project Plan

2.1 Task Decomposition

The project consists of converting three features into React; the tasks of the project can easily be broken down by according to features. While converting the code of each feature, we will follow the same plan consisting of 5 subtasks.

The task decomposition can be described as follows:

Project Prep:

- Understanding the requirements for the project
- Get set up in the Buildertrend environment
- Research React and Webform

Report List:

- Explore the old implementation
- Write the new React Code
- Test the new code
- Demo the completed feature
- Make edits according to demo and test findings

Report Details:

- Explore the old implementation
- Write the new React Code
- Test the new code
- Demo the completed feature
- Make edits according to demo and test findings

Workday Exceptions:

- Explore the old implementation
- Write the new React Code
- Test the new code
- Demo the completed feature
- Make edits according to demo and test findings

2.2 Risks And Risk Management/Mitigation

Risk Analysis							
ID	Probability of Occurrence	Risk Impact	Risk Level	Risk Description	Project Impact	Risk Impact	Risk Response Strategy
R1	Occasional	Major	High	Knowledge of coding languages used	This risk could cause delays within our project because extra time would need to be allocated to learning. If not handled correctly this risk could also cause code quality issues.	Technical, Scheduling	In order to prevent this risk, we have allocated time to review necessary coding languages and to explore Buildertrend's existing application. Any future occurrence of this risk will be handled by communicating with other team members and continued research.
R2	Expected	Negligible	Low	Early Completion	In general, this risk would not affect our project negatively. But being that the project/class is about completing work that takes two semesters, it is necessary that the risk is addressed.	Technical, Contractual, Scheduling	If features are implemented faster than expected, the client is prepared to assign more features. Early completion the scope of the project will change along with the timeline which won't be an issue because currently we are set to complete early.
R3	Unlikely	Major	Medium	Loss of Code	The project plan will be set back to the point of the remaining code.	Technical, Scheduling	This will be mitigated by committing changes to the git repository frequently. If confronted with this risk, we will work together to repopulate the code. We have extra time in our schedule to accommodate.
R4	Expected	Minor	Medium	Minor requirement change	minor code changes may delay the project plan	Technical, Contractual, Scheduling	Double-checking all requirements with the client team will mitigate this risk. Upon coming across this risk we will make tickets and fit them into our

							timeline. We have extra time in our schedule to accommodate.
5	Possible	Fatal	Medium	Major requirement change	major code changes that will delay the project timeline	Technical, Contractual, Scheduling	Double-checking all requirements with the client team will mitigate this risk. If this risk occurs, we will quickly respond with new tickets and timelines. If applicable, we will use past code as a jumping point in order to make up time. In general, we have time built into our schedule to accommodate extra tasks.
6	Unlikely	Minor	Medium	The functionality of ASP.NET incompatible with functionalities of React	re-write of react or ASP.NET will cost time and set the timeline back	Technical, Scheduling	Testing functionality frequently will help mitigate this risk. On discovery of the risk, the team will research alternative solutions and present them to Buildertrend.

Figure 1: This table analyzes the risks associated with the project.

			Risk Map			
			Occurrence Level			
			Possible	Unlikely	Expected	Occasional
			1	2	3	4
Impact Level	Negligible	1			R2	
	Minor	2		R6	R4	
	Major	3		R3		R1
	Fatal	4	R5			

Figure 2: Assigns risk levels based on risk occurrence and impact level

Risk Map Levels		
Low	Medium	High
A risk in this category has little effect on the progress of the project. Basic prevention should be taken. However because the risks in this category are minimal, they can be easily corrected and overall not feared.	A risk in this category has a medium effect on the progress of the project. Because of this prevention must be taken. These risks will affect the project but can be avoided.	A risk in this category could affect the progress of the project greatly and should be actively avoided through prevention.
Figure 3: This table describes, in detail, the levels within our Risk Map.		

Risk Occurrence Level				
Level	Title	Definition	Probability (%)	Justification
1	Possible	Technically can happen but probably will not	1	With category includes items that are easily avoided and will not happen with basic prevention
2	Unlikely	If everything goes as planned, it will not happen	3	Prevention needs to be taken and needs to be completed correctly to avoid this category
3	Expected	Most likely will happen	10	Active Prevention needs to be taken but will have little effect despite efforts; this occurrence can still happen
4	Occasional	Will happen a few times throughout the course within the whole project	20	Unpreventable but only will happen a few times.
Figure 4: This table estimates the risk occurrence				

Impact Level		
Level	Title	Time Delay (or less for all)
1	Negligible	0
2	Minor	One week
3	Major	Two week
4	Fatal	One month

Figure 5: This table assigns levels based on time delays caused by risks

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Project Proposed Milestones

Our project will be split into three main milestones. After reviewing the project and discussing expectations with the client, we have split the milestones into the following. The first iteration of the report list and report details (M1), the final iteration of the report list(M2), and report details and final iteration of workday exception(M2). Each of these three milestones will end in a milestone demonstration meeting with BT. The project is split into these three segments to ensure code completion is timely and accurate. Within the three main milestones, we hope to accomplish various mini-milestones/or subtasks.

Milestone 1 (October 19):

Report list:

- Explore the old implementation
- Write the first iterations of the new React Code

Report Details:

- Explore the old implementation
- Write the first iterations of the new React Code

M1 demo purpose: During this demo, we will be presenting tickets that have been made and the basic layout that has been implemented. At this time, we will be able to address any initial questions we have. We will also use this meeting as an opportunity to explain that program architecture we will be using for all three features.

Milestone 2 (November 6):

Report list:

- Write the second iterations of the new React Code
- Begin testing

Report Details:

- Write the second iterations of the new React Code
- Begin testing

M2 demo purpose: Through this demo, we will be able to show significant progress in the project. This milestone meeting will allow for the team to discuss issues being faced. As well as discuss possible adjustments or new features and address missing features to ensure a fully functional end product for demo three.

Milestone 3 (November 18):

Report list:

- In-depth unit test
- The final iteration of code
- Fix remaining bugs
- Complete documentation

Report Details:

- In-depth unit test
- Edits based on demo discussions and tests
- Fix remaining bugs
- Complete documentation

M3 demo purpose: During this demo, we will be presenting a fully functional report list and report details. We will use feedback from the demo to make necessary edits and to put us on the right track as we enter the second-semester phase of our project.

Metrics of Subtasks	
Subtasks	Evaluation Criteria
Explore the old implementation	In-Depth notes on the abilities of each feature. With this information, tickets should be made and organized by priority.
Write first iterations of the new React Code	The code should account for at least 30% of the initial tickets created.
Basic unit test created	Tests should account for 50% of lines in file
Write second iterations of the new React Code	80-90% of initial tickets to be completed
In-depth unit test	The test should account for 90-100% of lines in file
The final iteration of code	100% of tickets finished. This also includes tickets added for demo discussions
Fix remaining bugs	Completion of all remaining tickets found by testing
Complete documentation	Every file, function, and class has a brief description of its purpose

Figure 6: Describes the evaluation criteria of all subtask (all three projects have the same subtasks)

2.4 Project Timeline/Schedule

Gantt Chart 1

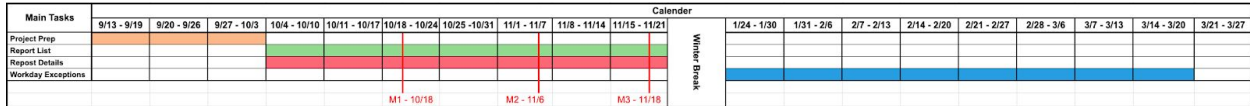
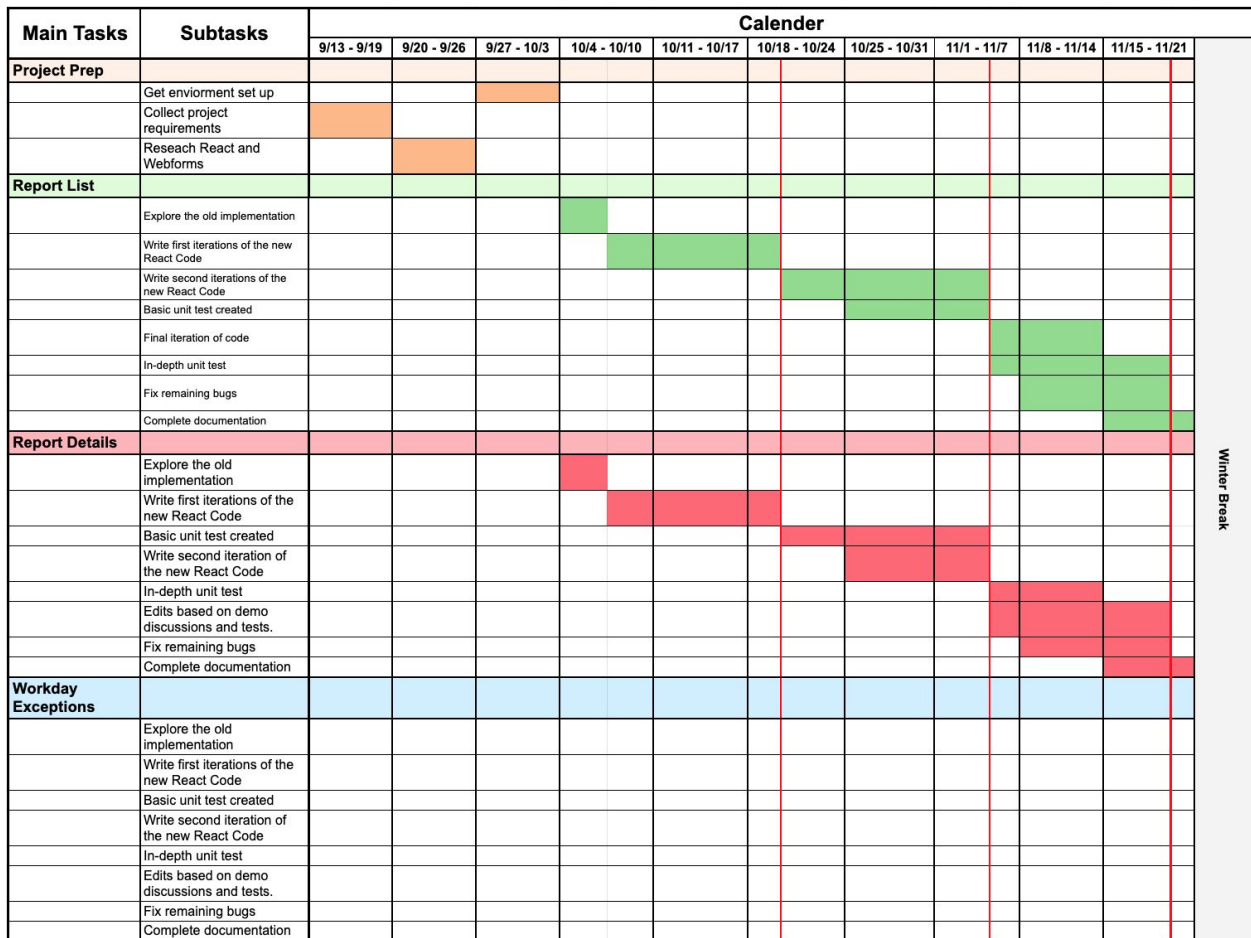


Figure 7: Displays over break down of time spent on each feature and estimated dates for milestone meetings

Gantt Chart 2 and 3 - Displays subtask, and how long each subtask is estimated to take for fall semester. Please note that the number of cells filled per column corresponds to the number of people working the subtask (see the legend under spring semester 2021)



Fall 2020 Semester

Figure 8: Displays over break down of time spent on each feature and estimated dates for milestone meetings

Semester 2 - Spring 2021

Main Tasks	Subtasks	Calender									
		1/24 - 1/30	1/31 - 2/6	2/7 - 2/13	2/14 - 2/20	2/21 - 2/27	2/28 - 3/6	3/7 - 3/13	3/14 - 3/20	3/21 - 3/27	
Project Prep											
	Get enviornment set up										
	Collect project requirements										
	Reseach React and Webforms										
Report List											
	Explore the old implementation										
	Write first iterations of the new React Code										
	Write second iterations of the new React Code										
	Basic unit test created										
	Final iteration of code										
	In-depth unit test										
	Fix remaining bugs										
	Complete documentation										
Report Details											
	Explore the old implementation										
	Write first iterations of the new React Code										
	Basic unit test created										
	Write second iteration of the new React Code										
	In-depth unit test										
	Edits based on demo discussions and tests.										
	Fix remaining bugs										
	Complete documentation										
Workday Exceptions											
	Explore the old implementation										
	Write first iterations of the new React Code										
	Basic unit test created										
	Write second iteration of the new React Code										
	In-depth unit test										
	Edits based on demo discussions and tests.										
	Fix remaining bugs										
	Complete documentation										

Figure 9: Displays over break down of time spent on each feature and estimated dates for milestone meetings

Gantt Chart Legend	
Number of Cells Filled	Number of Team Members Working on Task
1	6
2	3
3	2
4	(distributed based on need)
5	(distributed based on need)
6	1
Figure 10: Specifies the number of team members working on a task based on the number of cells filled in the Gantt charts above.	

As our timeline is now, we have extra time that will be utilized in case of unexpected problems or implementation of a new feature.

The first few weeks of class were dedicated to prep work. During that time we will set up our work environments and get a feeling for the project. During the remaining weeks of the first semester, we will work on developing components of the Report Details and Report List pages. Our work schedule will be based on the agile development model. During that time we will divide our work based on the number of cells filled in in each column. Information about that can be found in the legend above.

2.5 Project Tracking Procedures

In order to track the progress of this project, our team will be using Git, Github, Microsoft Teams, and Trello. We plan to track our hours in weekly status reports that we will send to our advisor on Thursdays. In addition to meeting with our advisor, we will be meeting with Buildertrend representatives regularly. This includes a biweekly meeting (on every other Friday) and three milestone meetings that track our overall progress.

2.6 Personnel Effort Requirements

Task	Approx time per person(hours)
project prep(environment setup, research)	18
Explore the old implementation	54
Write first iterations of the new React Code	33
Basic unit test created	27
Write second iterations of the new React Code	36
In-depth unit test	27
Final iteration of code	18
Fix remaining bugs	18
Complete documentation	24
Figure 11: Approximation of man-hours spent on each task	

The time estimate shown in Figure 11 above is an approximation of the number of hours that will be spent working on the task. Some tasks are the same for every page so this breakdown adds up the total time spent on the task for each page.

2.7 Other Resource Requirements

This project requires the use of AmazonWorkspaces, access to WorkSpaces will be provided by Buildertrend. There are no other required resources for this project.

2.8 Financial Requirements

There are no financial requirements for this project.

3 Design

3.1 Previous Work And Literature

Buildertrend developers and other senior design teams have already started the process of transition from ASP.NET to React implementations. The most significant of these implementations is the vast UI component library implemented through storybook. The library contains common features seen throughout the Buildertrend application and more specifically our features. These components include filters, basic feature outlines and colors. In addition to the storybook, one of the report details pages was already implemented. The general format of the work in progress report however is not fully applicable to the other details pages because it has a list format and the other details pages are graph focused. We will be using previous work as a model for our work. This will help us implement the requested modals faster and more accurately as well as ensure that our implementation fits cohesively with the rest of the application. A downfall to following previous work is that our creativity is limited. We do not have the same creative freedom as some other senior design teams. There aren't any other products on the market that we will have to worry about, but we will need to research React.

3.2 Design Thinking

When the project was first introduced to our team, Buildertrend had already worked through the “define” stage of design thinking. We will be using the problem statement that has been defined by Buildertrend.

Problem Statement: Buildertrend is moving to a service once model where all the business logic and APIs are written once for web and native clients. Buildertrend has identified the need to convert old ASP.NET code into React code to improve stability and performance.

Design Choices:

1. Object-Oriented Design
2. Focus on Modularity
3. Abstractions

3.3 Proposed Design

We plan to follow a design previously implemented by buildertrend. We will migrate all features already implemented in ASP.NET to React by reverse engineering existing pages. If overlooked problems/bugs may be found, they will be resolved in the new code. We will use APIs provided by Buildertrend; if one is not available, it will be requested. Once the team has completed the page, it will be sent to BT to review, should there be any problem/bugs, it will be sent back to the team to fix.

3.4 Technology Considerations

Buildertrend has chosen the technology used in this project; we will not need to consider other technologies.

3.5 Design Analysis

The proposed design described above works since we will be using Buildertrend's previous designs as a template. When designing features for a client, it is important to maintain consistency with their designs and business models.

As we implement and iterate over our design we plan to make modularity and scalability top priorities. With these priorities in mind, we will propose any software architecture changes that we see fit.

3.6 Development Process

We will be following an agile methodology (Figure 12) when completing our various features by completing in an agile cycle per feature. We found that this development process works best with this type of project. As described in the previous section we will start each feature in the requirements stage. The requirements stage of each feature has been completed by Buildertrend. Our team has received requirement pages for each feature that includes basic functionality and tools to be used. These pages will be the jumping point for our design and plan stage. During this stage we will go through each of the already created features that is made through webform. We then will create tickets based on the information collected. This will lead us into the development phase where we will start to implement the features. After a predetermined amount of time (see Ghant chart Figure 8 and 9) we will shift our focus to testing. In terms of our project our testing phase also includes our demos because it will be at this time where we can check that our implementation is on track. After each demo and round test we will re-enter the design phase and make adjustments according to the new information. Once each feature is at a point of completion we will deploy.

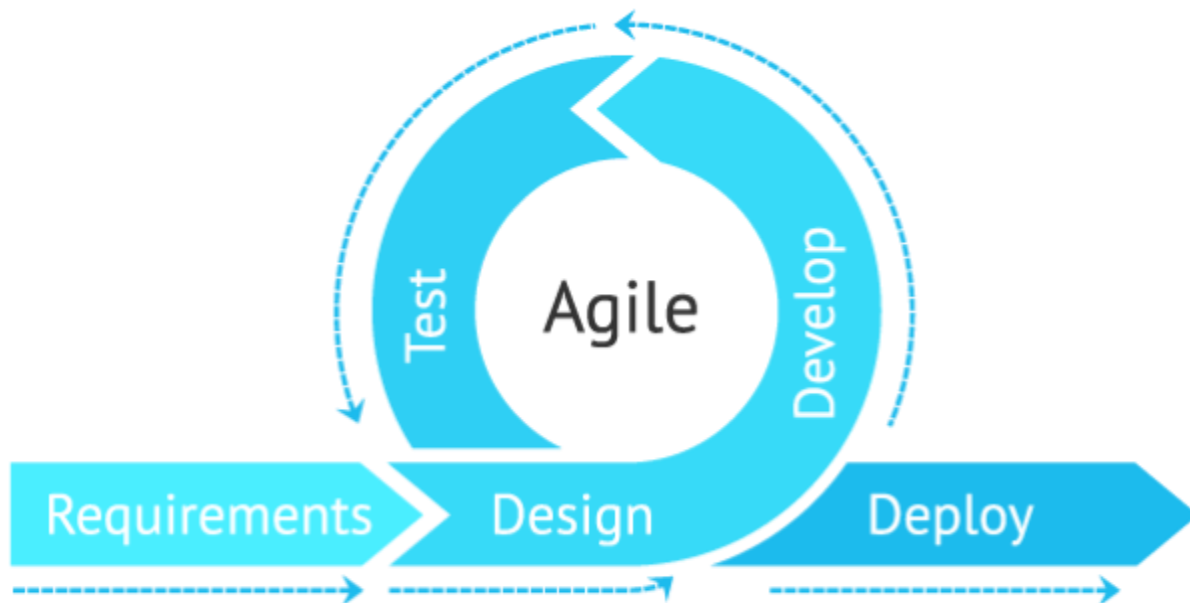


Figure 12: The agile development process.

3.7 Design Plan

Users will be able to look at all available reports from the report list page, when one is selected, the user is taken to the details page of report. Report details use Highcharts to show graphs. Users can add and remove a report from their favorites section. Workday exceptions will be part of the schedule page and once completed it redirects the user

to the schedule page. All components will use existing BT templates and APIs. The design plans for these features are shown below in Figure 13, Figure 14, and Figure 15.

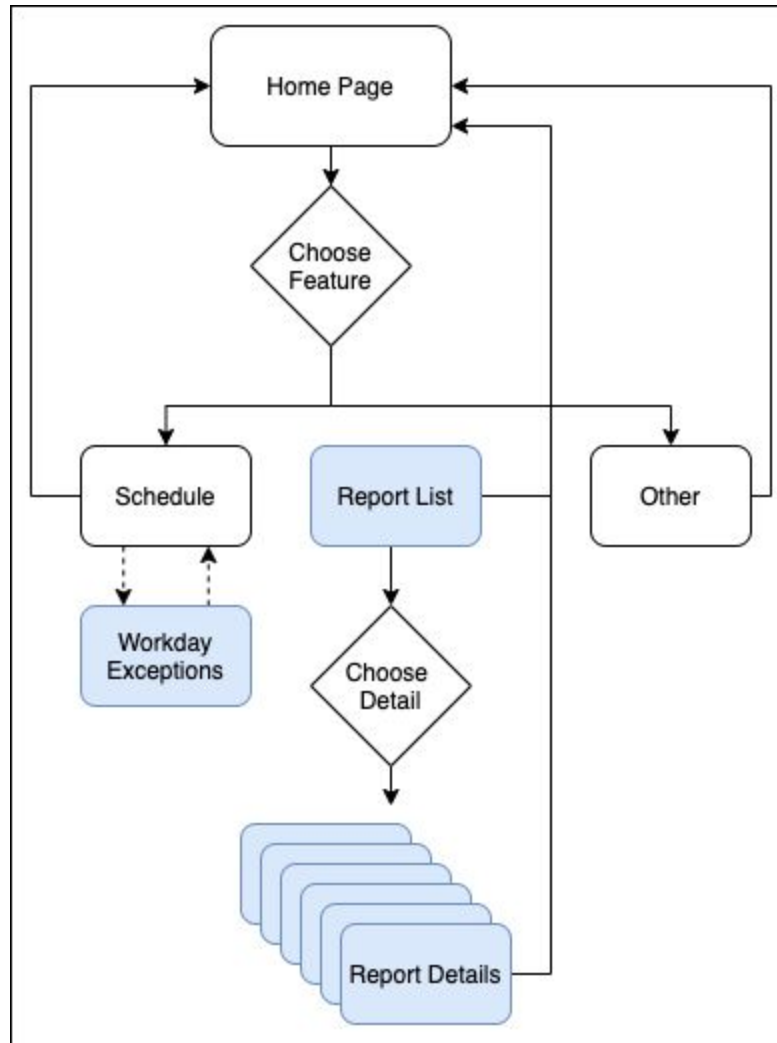


Figure 13: Workflow of user. Blue files indicate features we will be implementing

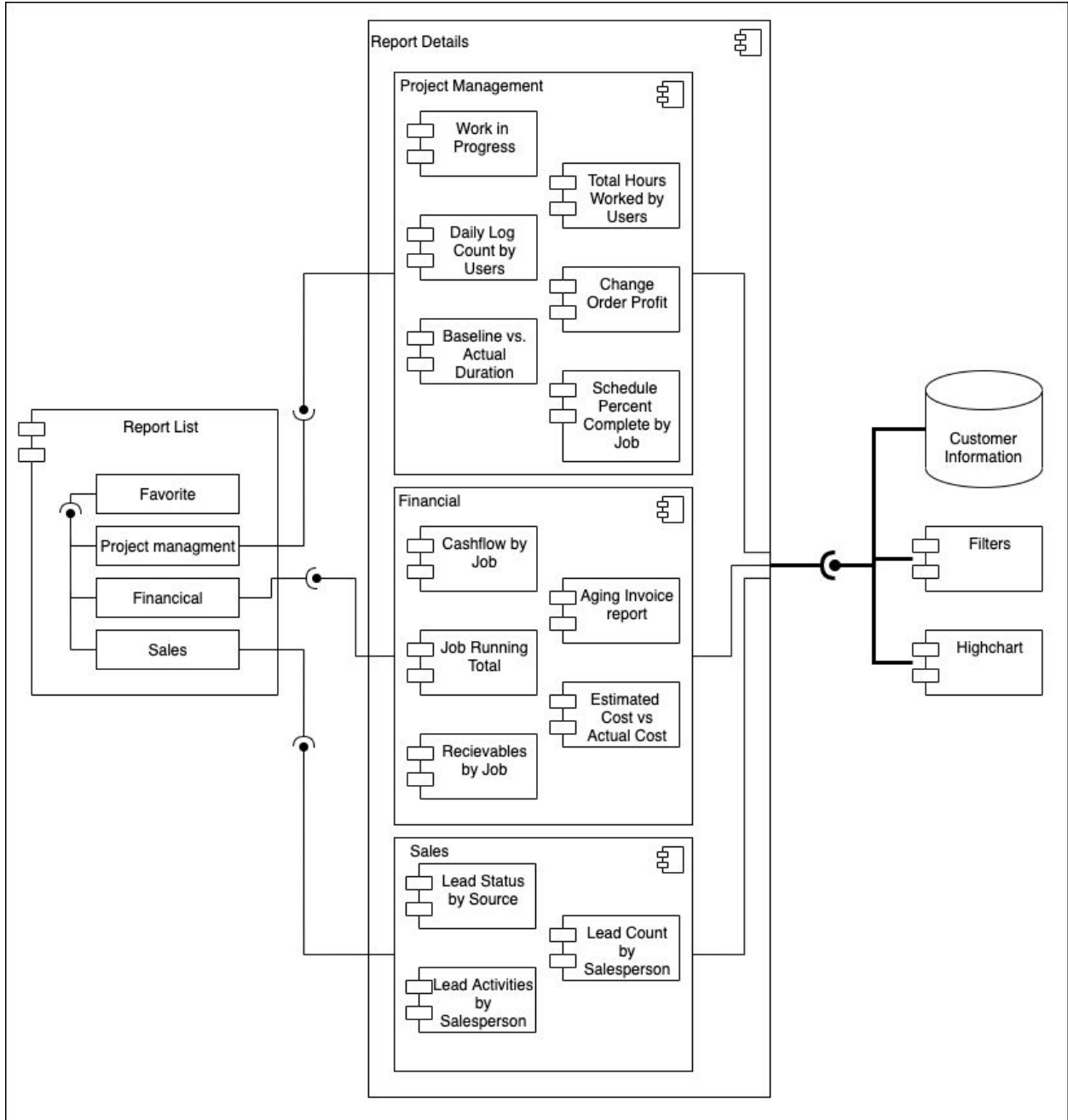


Figure 14: Component Diagram for Report List and Report Details

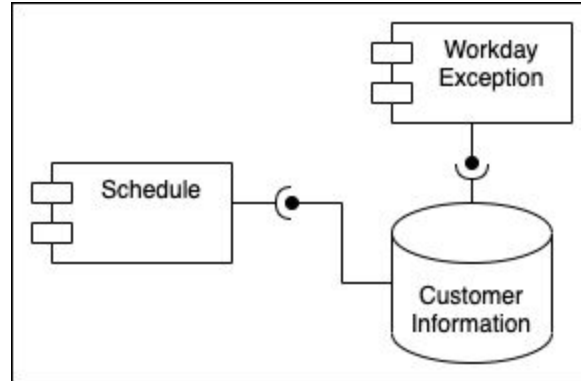


Figure 15: Component Diagram Workday Exception

4 Testing

The senior design team will be testing the features and components described above using Storybook. Storybooks has a hot reloading feature that will help us optimize our workflow. This saves time because Storybook will not make actual API calls; instead, it will use fake handlers while testing is being conducted. We plan to have each person use Storybook to test the features they develop; then, once we imbed our features into Buildertrends application, we will all work through acceptance testing. Our goal is to catch all bugs before the features make it to the application.

4.1 Unit Testing

We will not be conducting unit tests as per the client's request. This is not necessary because our features have been implemented with previously defined and tested components. For example, in the details pages the two main components are the Highcharts and filters. The filters component has already been implemented and tested by Buildertrend engineers and the Highcharts library has been tested by its creators.

4.2 Interface Testing

For this project, we will not be designing or implementing any interfaces. We will be using previously defined interfaces and API calls. We will be using acceptance tests to verify the interfaces work with the features that we create.

4.3 Acceptance Testing

Buildertrend has a test environment that runs the application from the perspective of the user. We will be using this test environment to determine if the requirements are being met. The acceptance tests we will be using for Report Lists and Report Details are included below. During our milestone meetings, we will use the test environment to demo the features we have implemented. This will allow the client to provide feedback and request any changes.

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 will click on the star button of each tile and verify that the star turns yellow and the tile was added to the favorites section of the lists page.

- When view all of a certain types (sales, financial, project management, etc.) is clicked, the page repopulates with all tiles of the selected type
 - To perform the acceptance test, we will click on view all for each card category and verify that the page repopulates with the tiles of the selected type.

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 will click on each report card and verify that the correct details page is loaded.
- A filter and highchart is displayed on each report details page.
 - To perform the acceptance test, will navigate to each report details page and verify that each page contains a filter and highchart that matches the old implementation.
- The user can update the chart data by using a filter.
 - To perform the acceptance test, we will apply different filters and verify that the chart updates to reflect the filter settings.
- The available filters match the filters in the old implementation.
 - To perform the acceptance test, we will open each filter and compare the displayed options with the old implementation.
- The highchart data reflects the current data stored in the database.
 - To perform the acceptance test, we will verify that the chart data matches the database data.

4.4 Results

So far, none of our acceptance tests have failed. If future tests fail we will assign a member of our team to determine the point of failure and make necessary adjustments. Once we complete all of our acceptance tests, our code will be handed off to a Buildertrend engineer who will incorporate our changes in the deployed application.

5 Implementation

5.1 Implementation Semester 1

For semester 1, the team worked together on the design documentation and awareness talks. We also split up into 2 teams that were in charge of completing a page each. Both teams have successfully implemented report lists and report details.

Report List

The report list's implementation started with understanding the vast code base and getting acquainted with the report list page. Soon after, tickets were made, and work was started. 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 topics with the appropriate tiles. 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.

Report Details

The implementation of report details started with the creation of work tickets for various detail pages. To generate these tickets, we discussed which details were most important and what the workflow would look like for the

semester. Due to unfamiliarity with the code base and new technologies, the team decided to work together to get one successful report details page completed. This page was the backbone for the implementation of the rest of the pages. The first hurdle faced during the implementation process was figuring out how to include the filters and generate the Highcharts. Once the outlines of these necessary elements were included, we focused on incorporating the appropriate types, colors, and labels. After completing the first report details page, we divided the remaining reports among Catherine, Saoud, and Hana. Throughout this process, we focused on consistency and readability. To remain consistent with Buildertrend's already implemented features, we structured our files using the same hierarchy and data structures. In terms of readability, we took the time to discuss appropriate file and variable names and included the appropriate documentation. The final stage of our implementing report details was adding connections to the database.

5.2 Implementation Semester 2

For Semester 2, the team will continue in the sub teams to complete work day exceptions and any other pages asked for by Buildertrend. Once the work day exception page is complete the minimum scope of the entire project is achieved.

6 Closing Material

6.1 Conclusion

We split up our team of six into two team sub-teams to work on two features, report lists and report details. This way, the team can work concurrently on two pages. Both sub-teams are nearing completion. This plan of action worked best for us as each member had enough time to work on their task along with other school work. It reduced the chances of a member's work being blocked due to another's work being incomplete. We found this plan of action more viable than six team members working simultaneously on each page, one after the other.

6.2 References

- [1] F. Wilson, "How to Use the Risk Assessment Matrix in Project Management," *Task*, 22-Jul-2020. [Online]. Available: <https://www.ntaskmanager.com/blog/risk-assessment-matrix/>. [Accessed: 12-Nov-2020].
- [2] "Gantt Chart: The Ultimate Guide (with Examples)," *ProjectManager.com*, 07-Oct-2020. [Online]. Available: <https://www.projectmanager.com/gantt-chart>. [Accessed: 12-Nov-2020].
- [3] "ISO/IEC/IEEE International Standard - Systems and Software Engineering - Life Cycle Processes - Risk Management," in *Std ISO IEC 16085 - 2006*, vol., no., pp.1-46, 15 Dec. 2006, doi: 10.1109/IEEESTD.2006.288594.
- [4] "ISO/IEC/IEEE International Standard - Software and systems engineering --Software testing --Part 2:Test processes," in *ISO/IEC/IEEE 29119-2:2013(E)*, vol., no., pp.1-68, 1 Sept. 2013, doi: 10.1109/IEEESTD.2013.6588543.
- [5] "What is AGILE? - What is SCRUM? ," *Cprime*. [Online]. Available: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>. [Accessed: 12-Nov-2020].