Buildertrend Zoom Integration

Design Document

Team Number: 27

Client: Buildertrend
Advisor: Dr. Lotfi ben Othmane

Team Members:
Jack Wilkinson – Project Developer
Axel Zumwalt – Project Developer
Casey Brandt – Project Developer
Nitin Nagavel – Project Developer
Rohan Mishra – Project Developer
Adam Banwell – Project Developer
Keven Lin – Project Developer

Team Email: sdmay22-27@iastate.edu

Team Website: https://sdmay22-27.sd.ece.iastate.edu/

Revised: 12/5/2021

Executive Summary

Development Standards & Practices Used

- IEEE/EIA 12207 - Software Life Cycle Processes
- IEEE/IEC 29119-2: Test Processes

Summary of Requirements

- A backend using .NET
- A frontend using REACT
- The ability to schedule Zoom calls within the application
- The ability to host Zoom calls within the application with minimal additional software installation
- The ability to generate transcripts of completed Zoom calls once they finish
- Basic user functionality to support multiple accounts creating separate scheduled zoom calls
- Basic login functionality (No security requirements as application is proof of concept and will not be utilized in a capacity that could expose actual Buildertrend customers)

Applicable Courses from Iowa State University Curriculum

- Com S 309
- Com S 319

New Skills/Knowledge acquired that was not taught in courses

- .NET experience
- REACT experience
- Zoom Api/Library implementation experience

Table of Contents

List of figures/tables/symbols/definitions:

1 Team

1.1 TEAM MEMBERS
- CASEY BRANDT
- JACK WILKINSON
- AXEL ZUMWALT
- ROHAN MISHRA
- KEVEN LIN
- NITIN NAGAVEL
- ADAM BANWELL

1.2 REQUIRED SKILL SETS FOR YOUR PROJECT
- Creating and accessing a SQL relational database.
  - Client preferred database framework
- Frontend Webapp User Interface design with React.
  - Client's application utilizes React for frontend design and functionality and are looking to keep the project as compatible to their full product as possible.
- .NET for backend scripting and data management.
  - Client's application utilizes .NET for backend functionality and was looking for its use for the above listed reason.

1.3 SKILL SETS COVERED BY THE TEAM
- SQL: Adam Banwell, Nitin Nagavel, Axel Zumwalt, Jack Wilkinson, and Casey Brandt
- React: Keven Lin, Nitin Nagavel, Axel Zumwalt, Rohan Mishra, and Jack Wilkinson
- .NET: Adam Banwell, Rohan Mishra, Keven Lin, and Casey Brandt

1.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

The project management style adopted by the team is a hybrid of waterfall and agile approaches. This hybrid will utilize many of the overarching planning benefits of waterfall with the consistent communication benefits of Agile as we will be utilizing frequent meetings to assure team understanding and adjust project velocity as needed.

1.5 INITIAL PROJECT MANAGEMENT ROLES

Product Owners:

- Rohan Mishra

- Axel Zumwalt
- Jack Wilkinson

Scrum Masters:

- Keven Lin
- Adam Banwell

Developers:

- Jack Wilkinson
- Adam Banwell
- Nitin Nagavel
- Casey Brandt
- Axel Zumwalt
- Keven Lin
- Rohan Mishra

2   Introduction

2.1 PROBLEM STATEMENT

Buildertrend is a construction software company that provides homeowners and construction companies/contractors the tools that are necessary to take a project from conception to project management to billing. Our group is aiming to solve the communication aspect of that value delivery system. In order to improve customer quality of life Contractors, remodelers, and commercial companies require a method of communication integrated into the Buildertrend Application Suite. Therefore, Buildertrend has asked our group to create a feature that will meet these requirements.

2.2 REQUIREMENTS & CONSTRAINTS

Our Capstone project will be a web service that would be able to schedule Zoom meetings via the Buildertrend Application. For example, this app could use the path: /meetings/meetingID/invite in order to allow users to invite others to a Zoom room. We will be integrating with the Zoom API to be able to connect with other Buildertrend users. An aspect of this integration will include the ability to save transcriptions of past meetings for future reference. Additional requirements include the ability to schedule

meetings, meeting management (settings for scheduled meetings) and user interface in line with what Buildertrend has already established in their other applications.

We have identified several constraints that we will have to work within throughout this project, many of which are development requirements requested by our client. First, our database backend needs to be a relational database to store User and Zoom Data. The front-end of our integration is preferred to be written in React.JS. And lastly, the server side needs to be written in either .NET or Spring. Additionally, we will be restricted by some of the features available to us through the Zoom API suite and adding additional functionality will require the incorporation of additional APIs or tools that may affect the performance of the product.

2.3 Engineering Standards

Our team will be implementing IEEE/EIA 12207, Software Life Cycle Processes, for our project. We'll be placing extensive focus on Planning, Defining, Designing, Building, Testing, & Deployment throughout the life of API development.

2.4 Intended Users and Uses

This Project acts as a proof of concept for adding video chat functionality to the Buildertrend construction management software platform. Buildertrend benefits from this implementation because improving communication on their platform is a key focus area for them in 2021.

With fully integrated chat features, users won't have to worry about creating and scheduling meetings on another platform which greatly improves the user experience. The voice chat can be used to create communication links between construction offices and sites or owners and builders directly on the software those people use the most. Users will also take advantage of meeting scheduling to plan important conversations and meeting recordings to look back and reference those conversations.

3 Project Plan

3.1 Project Management/Tracking Procedures

Which of agile, waterfall or waterfall+agile project management style are you adopting? Justify it with respect to the project goals.

We will be employing a mix of waterfall and agile project management methodologies. In respect to project goals Agile likely makes more sense due to the loose nature of requirements sent to us by Buildertrend. However, given that we were required to create a full project plan and Gantt chart for this assignment it doesn't make sense not to utilize the plan we've already constructed. Additionally, while loose we have been provided with enough specifications for elements of waterfall to make sense within the context of this project.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

Our group will be utilizing Git, Trello and Slack for project management tracking and communication. Given that Git is an industry standard and is hosted by the university it makes sense to use for this project. Additionally Trello provides a robust suite of project board features along without requiring any kind of purchase of software to use makes it ideal for our project. Finally, we've identified that Slack is the industry standard along with being a popular choice for communication outside of the project among team members and have decided to use it for our primary source of communication.

3.2 Task Decomposition

- Set up a project repo
- Set up architecture
  - Create a database
  - Stand up a webserver
  - Create a react app
- User Authentication and Log-in
  - Create database table for users
  - Create React component to log in as a user
  - User encryption and authentication logic
- Schedule meetings and add attendees
  - Add a meeting relation to the users data table

- ○ Create a React component to create meetings and add attendees
- ○ Create a React component to view upcoming meetings as a user
- Add APIs to generate meeting signatures
- Add meeting logic
  - ○ Join a meeting
  - ○ Start a meeting as the host
- Add invite logic
- Add recording / transcription logic
  - ○ Create a meeting recording attribute to the meetings data table.
  - ○ Add a way to view recorded meetings from the meetings view.

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

Project Milestones:

1. Set up a project repo
   a. Project repo should be accessible by all members of the group
2. Set up architecture
   a. Create a database
      i. The database will be set up with SQL as a relational database storing user information for future meetings.
   b. Stand up a webserver
      i. Web Server should be able to receive traffic and maintain an average response time of less than 300 ms
   c. Create a react app
      i. The react app will display the interactive meeting to the users and provide general communication with microphone, chat, and video.
3. User authentication and log in.
   a. User authentication algorithms should authenticate the right credentials for any user 100% of the time.
4. Schedule meetings and add attendees
   a. Creating a new meeting should add it to the database and should be scheduled for all invited attendees 100% of the time.
   b. Upcoming meetings view should show all meetings the user is invited to.
5. Add APIs to generate meeting signatures
   a. APIs will create unique meeting signatures for future meetings
6. Add meeting logic

a. Join a meeting
   i. Using created architecture, add React Components and web server logic
b. Start a meeting as the host
   i. Using created architecture, add React Components and web server logic

7. Add invite logic
   a. Algorithm to invite people should have a response time of less than 200 ms and should have a time to live of 10 minutes before needing to be hit again.
8. Add recording / transcription logic
   a. All recordings should be transcribed so that there are subtitles in every video.
   b. Every meeting that was recorded should be viewable by clicking the event on the meetings view.

3.4 PROJECT TIMELINE/SCHEDULE
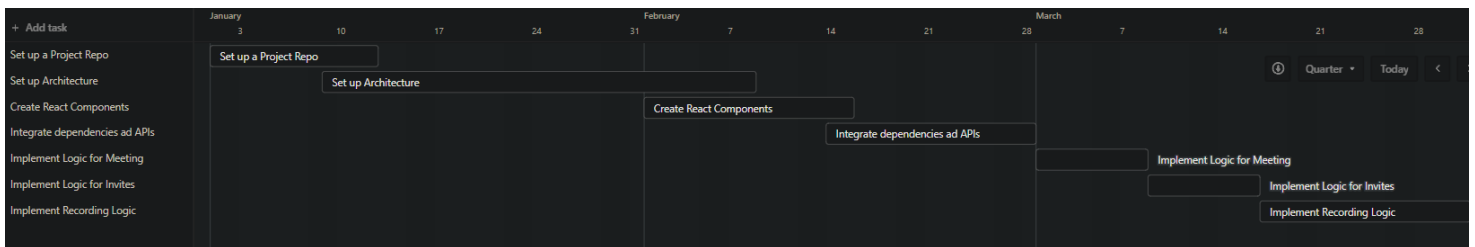
Figure 1: Project Timeline/Schedule Gantt Chart

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

Agile projects can associate risks and risk mitigation with each sprint. We attributed very little risk with this project because we find very little technical or time issues to completing the project. All of the technical work required of the project is either known by a member of our team or is well vetted and documented online so we foresee having any major hang-ups. Due to this we also don't see issues with going too long over schedule even if we do have a couple of delays.

Task	Risk Level (0.0 - 1.0)

Set up a project repo	**0.1**
Set up architecture	
Create a database	**0.5**
Stand up a webserver	**0.4**
Create a react app	**0.3**
Create a React component to log in as a user	**0.2**
Create a React component to schedule meetings and add attendees	**0.4**
Add packages to the project	**0.2**
Add APIs to generate meeting signatures	**0.2**
Add meeting logic	
Join a meeting	**0.3**
Start a meeting as the host	**0.3**
Add invite logic	**0.3**
Add recording / transcription logic	**0.5**

Table 1: Risk Management Plan

3.6 PERSONNEL EFFORT REQUIREMENTS

Task	Effort(hrs)	Explanation
Set up a project repo	**3 - 5**	**Project Repo on Gitlab Setup with CI/CD Environment**
Set up architecture		
Create a database	**15 - 20**	**Covers the implementation and Bugs**
Stand up a webserver	**10 -15**	**Covers the implementation and Bugs**

Create a react app	**15 - 20**	**Covers the implementation and Bugs**
Create a React component to log in as a user	**15 - 20**	**Time covers the implementation and integration with Buildertrend**
Create a React component to schedule meetings and add attendees	**15 - 20**	**Time covers the implementation and integration with Buildertrend**
Add packages to the project	**15 - 20**	**Time covers the implementation and integration with Buildertrend**
Add APIs to generate meeting signatures	**15 - 20**	**Integration with Zoom along with any QA checks**
Add meeting logic		
Join a meeting	**10**	**Integration and QA**
Start a meeting as the host	**10**	**Integration and QA**
Add invite logic	**10**	**Integration and QA**
Add recording / transcription logic	**10**	**Integration and QA**

Table 2: Personal Effort Plan

3.7 OTHER RESOURCE REQUIREMENTS

The primary resource requirements will be time and computers. Given the nature of the project and the fact that it is entirely software, it will require little else to complete. Additionally Buildertrend has offered us a Git repository and a server that we will be able to utilize for development, making these resources already readily available.

- React
- .NET/Spring
- SQL
- Github
- Zoom API

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

Area	Description	Examples
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	EX: Increasing/reducing exposure to pollutants and other harmful substances, increasing/reducing safety risks, increasing/reducing job opportunities
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	Development or operation of the solution would violate a profession's code of ethics, implementation of the solution would require an undesired change in community practices
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	Increasing/decreasing energy usage from nonrenewable sources, increasing/decreasing usage/production of non-recyclable materials

Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	Product needs to remain affordable for target users, product creates or diminishes opportunities for economic advancement, high development cost creates risk for organization

Table 3: Broader Context Design Areas

Areas	Considerations
Public health, safety, and welfare	Providing the opportunity for people to come together in multiple ways. I.e. Subcontractors can communicate with each other to verify if something that's being done is safe and follows correct practice.
Global, cultural, and social	The development of this solution should not affect any communities or groups since this feature is a part of a suite of products for construction management.
Environmental	Since this is a software product, there is no direct environmental impact. However the devices this application will run on could be.
Economic	This product will be a part of a suite of products offered by Buildertrend. Buildertrend's pricing is derived from extensive market research and understanding their niche.

Table 4: Design Area Considerations

4.1.2 User Needs

- Construction industry professionals need a way of communicating between onsite and in office workers because on-demand, fast communication is necessary to prevent costly construction mistakes.
- Clients need a way of communicating with construction professionals to have input in the construction process of their buildings.
- Users of the buildertrend app need a way to open Zoom meetings in the app so

that they can be done intuitively with very little hassle.

4.1.3 Prior Work/Solutions

Slack's integration of the Zoom API as well as Google Workspace and Salesforces' use of the Zoom APi are the main complete projects conducted by reputed companies**.** We will be utilizing these implementations as reference when developing our project.

4.1.4 Technical Complexity

Involved Components:

- .NET backend
  - C# based backend system that utilizes MVC architecture and will provide as a connection to the API and the database
- SQL relational database
  - A relational database that will store all relevant information about the users and project.
- REACT frontend landing page
  - Frontend will allow the users to interact with the application and will provide context for the functionality in the project
- Zoom API
  - This API will allow us to integrate zoom directly into the web application
- Potentially Teams API
  - This is a consideration for if we finish the project ahead of schedule

The project will utilize all of the components listed above. It will do this by connecting them with the backend and frontend working to provide functionality and UI design that will then utilize the Zoom API (and potentially the teams API) and relational database to provide underlying functionality to the page. For example, we will store user login information in the database along with information regarding the meeting details. The most complex aspect of this project when compared to a normal web application will be the Zoom integration. This will require us to utilize the Zoom API in order to host the call within the Buildertrend application itself. This will require research on our end towards finding either first or third party software that will allow us to accomplish integrating a zoom call into the web application. We will also encounter complexities while designing the UI for the project as we will need to build it from scratch and make sure that it provides enough context and clarity for users to use it effectively.

4.2.1 Design Decisions

1. We made a design decision to use zooms api for integration over teams.
2. For our Relational Database requirement, we decided to use SQL.
3. For our Server-side technology, we decided to use .NET as recommended due to how it mirrors Buildertrend's tech stack.

4.2.2 Ideation

When deciding our Server-Side Technology, we finalized on using .NET as our framework because it mirrors the exact same tech stack as Buildertrend. Some other options include:

- Apache HTTP Server
- KS Software
- Perl
- Qt
- Ceylon

4.2.3 Decision-Making and Trade-Off

Design Decision	Pros	Cons
Zooms api integration	Better documentation More intuitive implementation	Some functionality may not exist
SQL for relational database	Mirrors buildertrend tech stack Team is familiar with it	SQL could also be missing some functionality
.NET backend	Mirrors buildertrend tech stack	Less experience working with .NET

Table 5: Design Decision Trade-Offs

Discuss what you have done so far – what have you tried/implemented/tested?

Due to the minimal technical requirements and similarity of the project to other existing projects we have felt our time this semester would be better spent developing the project plan and related deliverables. To this end we haven't done much in the way of technical experimentation and are reserving much of this for the Spring at our client's request.

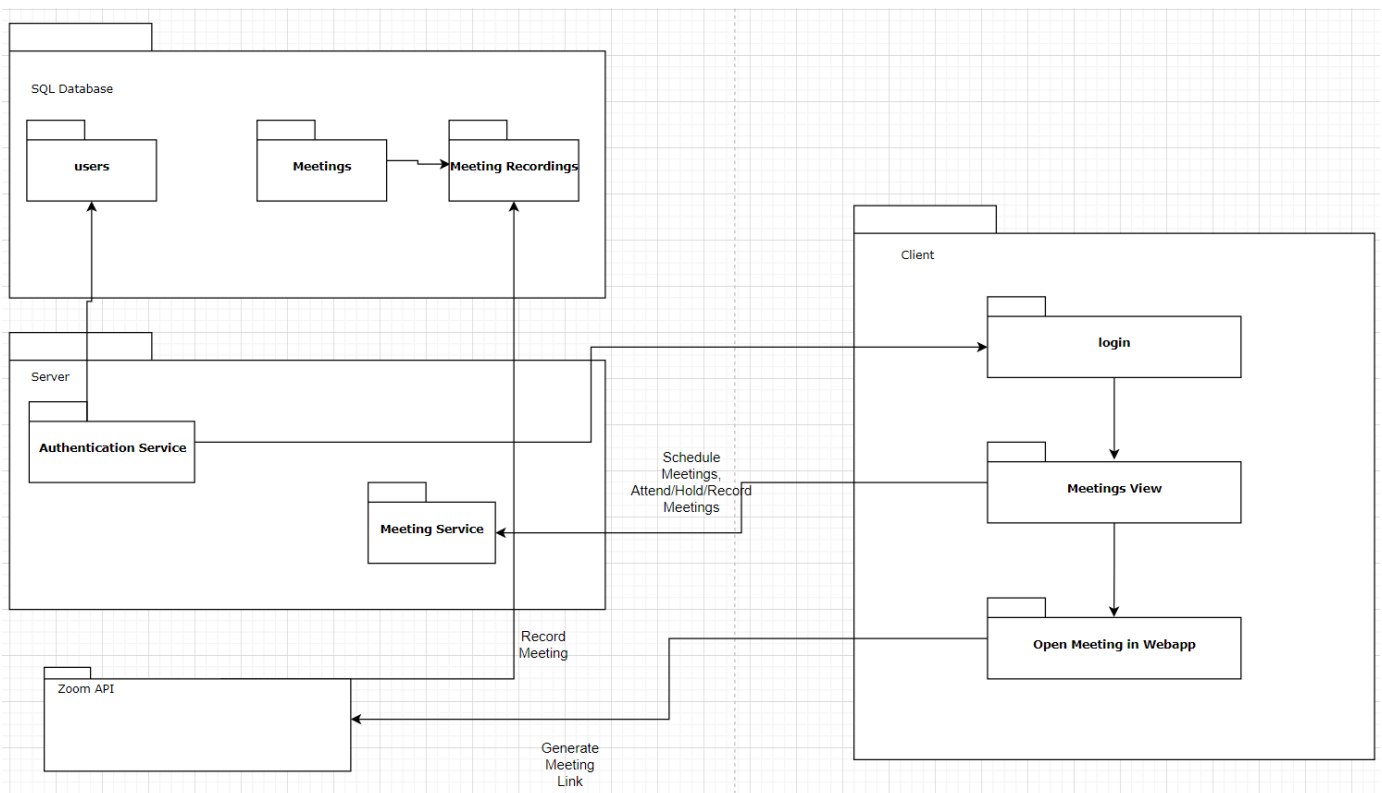### 4.3.1 Design Visual and Description
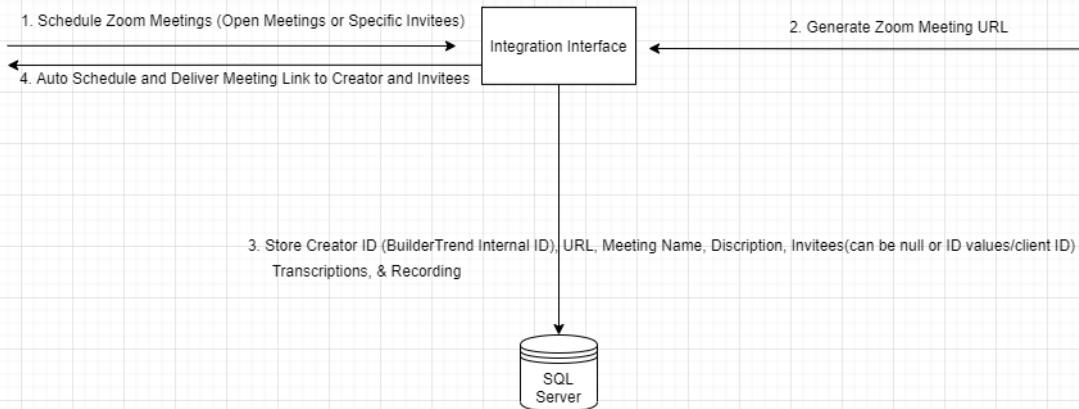

Figure 2: Design Visual I

Figure 3: Design Visual II

4.3.2 Functionality

Describe how your design is intended to operate in its user and/or real-world context. This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

Buildertrend has requested that we design for them a zoom meeting place integrated into their preexisting software. They would like to be able to schedule zoom meetings, send meeting links, and host zoom calls all from within their own software without having to download another application. We are permitted to download a minimum amount of third party software if needed however. They have also mentioned wanting us to integrate with a calendar/scheduling application in order to set meeting times all from within their own software.

How well does the current design satisfy functional and non-functional requirements?

The design satisfies these requirements to the best that it is able to at the moment. We have not been provided a lot of specifics when it comes to requirements, just the general needs listed above so we have not been able to fully flesh out a design yet as there still a lot that needs to be known, but we have tried to fit our design to best suit the requirements as we know them.

4.3.3 Areas of Concern and Development

Based on your current design, what are your primary concerns for delivering a product/system that addresses requirements and meets user and client needs?

1. We do not have much experience with .NET which was recommended by the client, there could be some issues brought forth by that lack of experience during the development process. If we were to run into a problem with .NET it may take longer than it should to fix the problem.
2. Another concern is that buildertrend has not provided us with more specific requirements about the project. There is a lot more information we need from them in order to make larger design decisions. They have told us that they will be providing some more information in a few weeks, but at the moment we do not have much.

What are your immediate plans for developing the solution to address those concerns? What questions do you have for clients, TAs, and faculty advisers?

1. We plan on learning how to use .NET prior to development so that we will be less roadblocked by lack of experience. It is not a perfect solution but one that will definitely make it easier to get the work done efficiently.
2. We have requested more information from buildertrend. As mentioned earlier they have promised to provide us with more information soon so that problem may not be too impactful. We will also be in open communication with buildertrend if we still need verification on design choices or need to know how they would prefer a specific element of the design.

4.4 TECHNOLOGY CONSIDERATIONS

Highlight the strengths, weaknesses, and trade‑offs made in technology available.

Strengths:

- REACT is a robust frontend framework and has wide array of optional libraries that allow for highly specialized implementation
- .NET is a widely used backend framework and therefore has a great deal of documentation surrounding it, something that will be helpful due to the lack of .NET experience for some team members

Weaknesses:

- REACT is not based in the same language as .NET and would be better paired with a javascript backend library that would allow for less context switching
- .NET is not a backend framework that many of the team members are experienced with. A language like Java where all of the team members had experience would cause less future learning issues

Trade-Offs:

- We are able to choose between the Zoom API and Zoom integration libraries. The integration libraries would be able to provide an implementation closer to what the client is looking for but utilizing an API to make calls to Zoom accounts would drastically reduce the overall amount of work required in the project

Discuss possible solutions and design alternatives

There is very little that we can do regarding fixing and particular weaknesses or tradeoffs. The technology utilized for this project in its entirety will have been selected by the client and given this we have decided to simply adhere to their requests regardless of any downsides in order to better suit their expectations.

4.5 DESIGN ANALYSIS

Did your proposed design from 3.3 work? Why or why not?

As of now the design is complete and has allowed us to make further design and planning decisions. We have not begun implementing code as of yet so we have not been able to further test its effectiveness at this time.

What are your observations, thoughts, and ideas to modify or iterate over the design?

As of right now we have noted a need to increase the overall amount of detail in the visual design. This would include more input from the client on design practices incorporated into their application, we haven't been able to discuss many of these requirements with them yet as they have indicated a preference for discussing project specifics more over the course of next semester as development begins.

We are also planning on utilizing the hybrid project planning implementation of our project in order to refine our project velocity and planning as we encounter each of the identified project objectives. This would include further expanding out objectives as we make progress on currently identified objectives.

4.6 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

Using our component diagram, our architectural overview describes how we will implement the zoom api into Buildertrend's existing system. We will use a SQL Database to store user and meeting information. Using Buildertrend's server, clients will be able to login and use the zoom features that will be integrated into the application.

5 Testing

5.1 Unit Testing

Ticket tasks & features will be broken down in such a way to comprise one "unit" of development, including testing. In order to assist in achieving complete test coverage we will be utilizing a test driven development philosophy where the developers will develop the unit test for each class or methods concurrently during the coding process. We will utilize JUnit testing in order to perform unit tests on the back end of our application.

5.2 Interface Testing

The Interface will be a ReactJS Front-End application. The React application will be tested using Jest and React Testing Library. Tests will be written concurrently with the front-end application in a similar manner to our development philosophy for the backend aspect of the project.

5.3 Integration Testing

We have one critical integration path. Our primary concern will be connecting our React.js front end to our Java Spring backend. We will need to write a combination of Jest & RTL (for front end) and Java Test Suite to make sure that our front and back end are playing nicely and working properly. This will include tests that check for integrity and functionality of the front end and proper backend functionality as the front-end is interacted with. There will be some testing surrounding potential integration with the BuilderTrend environment. However, this will not be a primary concern as we will be utilizing a different backend framework than them and are focused primarily on developing a proof-of-concept. We will develop the frontend and backend testing suite to ensure that there are no issues if a full-scale integration were to happen.

5.4 System Testing

System testing requires a high level of coverage while working on new features. Even in cases where we are not achieving 100% test coverage in our project we have determined that bare minimum two or three tests per identifiable feature or component is adequate for checking for base functionality. These tests should be written in a way that takes in random or black box values in order to try and catch edge cases and make sure that values aren't being selected for their ability to make a test pass. This use of 2-3 tests per component applies to unit, interface and integration tests alike and will be written prior to or as code is being written. We will utilize the tools outlined above for each component of the testing scheme and tie them together by having a "total" testing script that will run all tests that will be run before code is added to the codebase.

5.5 Regression Testing

We will ensure that new additions do not harm existing functionality by constructing tests in a way that will refer to specific components and values and in the case of black-box testing verify that values are adhering to the appropriate data flow while moving through the application. The primary critical features are those surrounding the zoom call functionality (scheduling, calling and transcription). This choice of critical features is driven by the requirements of the project but the criteria for this testing philosophy will apply to all aspects of the application and will align with our dedication to test driven development in that we will strive to write code to make tests pass rather than writing tests to make code pass. For our regression testing we will utilize the same frameworks as our other regions of testing but will reinforce it will support tools that allow the application to spin up locally or hit production to verify that total application functionality is maintained.

5.6 Acceptance Testing

We will verify that design and functional requirements are being met by writing tests that focus on specific components and application states that are being attained by the application. These will focus on customer-facing data and application states and may rely on some amount of local application "simulation" but will primarily focus on what is visible and developing it in a way that may be easily rerun by developers using tools through our development environment.

5.7 Security Testing (if applicable)

The primary concern for this project is functionality. We will be providing a small amount of testing around making sure that accounts are working properly but our client has specified that this is not the primary concern of the project and has recommended that we largely ignore security concerns outside of general front-end security issues.

5.8 Results

The results of our future testing will help us accomplish tasks and features with unit testing, frontend functionality with interface testing, overall backend and frontend functionality with integration testing, component system functionality with system testing, newly implemented features with regression testing, and finally customer approval and security with acceptance and security testing.

6 Implementation

Implementation described in section 3.3. Exact ordering of implementation will be followed due to requirements.

7 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

SE Engineering Code of Ethics and professional Practice

Public	Shall act in the interest of the public	The SE and NSPE standards are very similar as it ensures that the work and the product is produced to serve in the best interests of the public.
Client and Employer	Shall act in the best interest of the client and the employer	Client and Employer standard from SE is very similar to work compensation and financial responsibility from NSPE as these standards ensure the final product meets the specification of both the client and employer. In addition, the final product will meet professional standards and in cost, delivery, and

		production.
Product	ensure the product meets professional standards	The SE and NSPE standards are very similar as it ensures that the product meets professional standards.
Judgement	maintain professional judgement	Judgement in both the SE and NSPE standards look for integrity as well as not engaging in deceptive acts.
Management	managers and leaders shall promote ethical management and development	Management in the SE and NSPE standards look to approach management with ethical approaches that lead to a good product and deceptive acts being reported.
Profession	maintain the integrity and reputation of the profession	This SE standard and NSPE Social Responsibility Standards and Health, Safety, and Well being standards are very similar as it ensures that individuals working in this industry need to maintain the integrity and reputation of the profession. In addition, ensure the work and product of the industry is to work in the best interest of society.
Colleagues	Be fair and supportive of their colleagues	The Colleagues standards from SE and the Health, Safety, and well being standard from NSPE are very similar as both are promoting a fair, safe, and supportive work environment.
Self	maintain ethical approach in the profession	The SE and NSPE standards for self are similar as they both require an ethical approach from oneself in an engineering setting.

Table 6: Areas of Professional Responsibility

7.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

- **Work Competence**
  - High - Work Competence applies to our project's professional context. We have done a good job maintaining this quality of work by maintaining consistent standards and clarity when it comes to project documentation and planning. This is extremely important for our project as it will be utilized in a professional context by Buildertrend customers and they will need it to work consistently and effectively as they will rely on it to operate their businesses.
- **Financial Responsibility**
  - N/A - Financial Responsibility does not apply to our project because this project is not one that we are charging for. There is no money being passed between developers and clients so there are no financials to be responsible about. Our team performance is N/A

- **Communication Honesty**
  - High - Communication honestly does apply to our project because we are in communication with a client that is expecting a product. For most projects in this class that means truthfully communicating what work has been done to a professor, however our project was requested by an actual company and part of what we make may actually be used by the company. Therefore the truthfulness of our communication has a direct impact on a real world business. Our team has been very open with the client and advisors about what has been completed so far so we are performing very high.
- **Health, Safety, Well-Being**
  - N/A - Health, Safety, Well-Being does not apply to our project. While our project could have safety implications for the general public, our client is looking for a proof of concept/functional prototype and won't be utilizing any kind of login implementation that we design. In this regard we just need to create a system that functions and since our requirements don't extend outside of that and it has no ability to harm the public with its functionality it is completely disconnected from this requirement.
- **Property Ownership**
  - N/A - Property Ownership does apply to our project because we actively working with a company's software. Because of this we will be actively having to respect their software and not look into areas that are not pertinent to the project if we have access to them. Our team has not yet had to work with their software yet so we have not come into the issue of Property Ownership therefore the team performance right now is N/A.
- **Sustainability**
  - N/A - This doesn't apply to our project, sustainability doesn't affect our project in the sense that there is little that we can do to make our project more or less sustainable. The hosting service that we will use along with the data management solutions will be provided by Buildertrend making it not possible for us to improve the environmental impact in any way. Additionally, we don't have any physical components making sourcing a non-issue. Due to the nature of our project, it is largely unaffected by requirements for sustainability.
- **Social Responsibility**
  - High - Social Responsibility applies to our project as it requires us to make sure that the system we are developing is safe and doesn't collect data or perform any actions maliciously or promote negative behaviors within the user base. Our team has performed the Social Responsibility aspect very well so far and have prioritized assuring that design decisions are made with the best interests of the client and the user in mind.

Of the areas of professional responsibility outlined in the paper "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment" we identified "Communication Honesty" as being both important to our project and important for our team to maintain a high level of proficiency in for the project.

Communication honesty is important in executing this project because the software development nature of the project requires a lot of coordination as all seven of our team members will be working on and making changes to the same codebase. Without clear and constant communication there will be a lot of personal and technical conflicts as a result of miscommunications that are easily avoidable.

In the forming and storming phases of our team's life cycle we have already identified communication issues as a possible force that could cause issues moving forward. To combat that we have had discussions about how we should handle communication and demonstrated professional responsibility in fixing those issues too.

8 Closing Material

8.1 Discussion

Our project is fully designed and ready to be implemented. As the design stands all requirements are fully met and should be manageable when it comes time to begin implementing the code. No code has been written at this point so at the moment none of the requirements have been tangibly met, but every requirement has been fully planned out and will be completed.

8.2 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals. What constrained you from achieving these goals (if something did)? What could be done differently in a future design/implementation iteration to achieve these goals?

So far we have accomplished creating a full project design and project plan. We have selected the frameworks necessary to implement our project along with the storage and management software that we will need to accomplish full implementation. We hope to create an application that can fully support Zoom calls within the app. This application will have the ability for users to login with distinct accounts that have Zoom meetings and calls tied to them. In order to accomplish this we will follow the implementation

order outlined in 3.3, we will follow this very closely due to the high requirements of each; they are required to be completed sequentially in order to properly accomplish each. I think that in the future we will definitely be exploring activity parallelization opportunities as we further develop and encounter technical issues and milestones. Parallelization would allow us to potentially increase project velocity exponentially but is difficult without further exploring the technical aspects of the project.

8.3 REFERENCES

Not applicable due to lack of technical project work done. Listed technical references are largely an indicator of the overall concept being employed in the project, otherwise team experience is enough to cover the required planning elements performed so far.

8.4 APPENDICES

8.4.1 Team Contract

Team Contract

Team Name: *Hermes*

Team Members:

1) Adam Banwell   2) Rohan Mishra

3) Keven Lin   4) Nitin Nagavel

5) Axel Zumwalt   6) Jack Wilkinson

7) Casey Brandt

Team Procedures

Day, time, and location for regular team meetings:

All regular team meetings will be held virtually on zoom. A meeting link will be posed in the team Slack channel at the time of the meeting.

Monday: 10AM - 11AM: Beginning of week check-in, discuss plans for work to be done over the week, discuss any issues faced by team members over the weekend (if applicable) and review current progress.

Friday: 2PM - 3PM: End of week check-in, discuss progress made over the week and evaluate any issues that were encountered and how that affects project velocity.

For any valid reason determined by the majority of the team a weekly meeting may be cancelled or moved to an in-person location. **Team meetings must be confirmed with the entirety of the team beforehand to make sure that all team members are aware and can provide input.** If decisions are made outside of a scheduled meeting that affect the entire team in significant ways, it must be brought up again during a scheduled or planned meeting in order to keep all team members up to date on what is happening.

Preferred method of communication updates, reminders, issues, and scheduling:

The preferred primary method of communication between team members will be Slack. Communication with members outside of the team including but not limited to, Jacob Conn (TA), Lotfi ben Othmane (Faculty Advisor), and Kayla Gilleland (Client point of contact), will be done over email using the team email sdmay22-27@iastate.edu.

Decision-making policy:

Any decisions impacting the team will be discussed during a weekly meeting and decided upon with a majority vote.

Procedures for record keeping:

Meeting minutes will be recorded on a shared Google document that all members of the team can access and refer to. More important takeaways and action items will be captured in Slack or Trello.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

   ➔ **Attendance** - Team members are expected to attend all meetings for the project without valid extenuating circumstances. If a team member is expected to be absent from a meeting they should notify the team in advance and are expected to follow up after the meeting and consult meeting minutes to catch up on any missed information.

➔ **Punctuality** - Given the busy schedules of team members it is fine if a member is occasionally late or only late by a few minutes. However, they are expected to be there for a majority of a meeting in order to properly contribute.
➔ **Participation at all team meetings** - Team members are expected to contribute to all topics covered by the team during meetings. If they do not have anything additional to add to a specific topic due to agreement with the course of action chosen that is fine but they are expected to let members know this and are expected to follow the content of the meeting to reduce questions.

2.   Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

➔ **Team assignments** - Team members are expected to complete all assignments and work assigned to them without appropriate extenuating circumstances.
➔ **Timelines** - Team members are generally encouraged to work on assignments and work assigned to them over the course of the time allotted for said work. However, given the busy nature of senior year it is fine if they work on it more sporadically if they are able to assure continued quality and meet provided deadlines.
➔ **Deadlines** - In the event that a deliverable or work package being worked up by a team member or members is constrained by a deadline, those individuals are expected to deliver on time. If unforeseen circumstances prevent this, those members must communicate with the rest of the team the issue as soon as possible so the team can adjust project plans accordingly.

3.   Expected level of communication with other team members:

Team members are expected to communicate primarily through Slack in a way that clearly notifies all other members of current developments and decisions. Secondary to Slack, email is also an acceptable way to contact and communicate with other team members with the understanding it may yield slower responses.

4.   Expected level of commitment to team decisions and tasks:

Team members are expected to commit to and involve themselves in the conversation surrounding decisions and tasks that involve them or the entire team to at least a point where their lack of commitment does not adversely affect other team members.

Leadership

1.   Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Individual members of the team will take ownership over areas that most interest them as those areas become apparent during the project. For example, Adam and Rohan will take ownership of team organization, Axel and Jack will take ownership of project planning, and Nitin will head up testing initiatives. However, these roles are not solely defined to anyone and the team can share in these roles.

2.   Strategies for supporting and guiding the work of all team members:

Team members are expected to rely on one another for project related issues. The primary avenues for this are through Slack or during the planned meetings for the project. If these areas are insufficient for resolving issues that team members are encountering then they are encouraged to utilize the wide range of online resources that are available for the primary technologies we are utilizing in this project.

3.   Strategies for recognizing the contributions of all team members:

We will be separating work into workable tickets that team members are expected to take on and resolve. If they would like to collaborate on specific tickets that is fine but this doesn't absolve them of completing their allotted work.

Collaboration and Inclusion

1.   Describe the skills, expertise, and unique perspectives each team member brings to the team.

Adam: Java, Spring, SQL, C, Python

Rohan: Java, Spring, terraform, Azure & AWS, Spark, Kubernetes, Redis

Keven: Java, Spring, React.js, AWS, SQL (Oracle, Microsoft), GraphQL, NoSQL

Nitin: Java, Spring, SQL (Oracle and Microsoft), C, JavaScript, Python

Axel: JavaScript/TypeScript (React framework, NextJS, GraphQL), CSS (Styled Components), Java (OSGi framework), Python (GUI testing), C, C++, SQL

Jack: JavaScript (React Framework, GraphQL, NextJS), CSS, Java (Spring Framework), Python, C, C++, SQL

Casey: Java (Spring Framework), C, C++, SQL

2.   Strategies for encouraging and support contributions and ideas from all team members:

- Team meeting every Monday and Friday
- Slack messages
- Additional brainstorming meetings

3.   Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

- Team meeting discussion

Goal-Setting, Planning, and Execution

1.   Team goals for this semester (Will be further defined after initial meeting with client):

- Research/select tool to integrate through APIs into sample site
  - (Zoom or Teams)
- Plan an Interface
  - User login
  - Scheduling, adding, inviting, recording
  - Hosting, joining, viewing recordings
- Plan APIs
  - Storing login information
  - Meeting signatures

2.   Strategies for planning and assigning individual and team work:

- Project planning meetings
- Project planning tools (trello, etc.)

3.   Strategies for keeping on task:

- Consistent communication with team members

- Update Trello/Git Boards upon completion of sprints
- Follow project plan and deadlines when planning work execution

Consequences for Not Adhering to Team Contract

1.   How will you handle infractions of any of the obligations of this team contract?

Discuss infractions and issues in team meetings and retrospectives

2.   What will your team do if the infractions continue?

 Contacting our project TA or Faculty Advisor

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*

b) *I understand that I am obligated to abide by these terms and conditions.*

c) *I understand that if I do not abide by these terms and conditions, I will suffer the*

consequences as stated in this contract.

1) Adam Banwell   DATE: 09/16/2021

2) Nitin Nagavel   DATE: 09/16/2021

3) Keven Lin   DATE: 09/16/2021

4) Rohan Mishra   DATE: 09/16/2021

5) Jack Wilkinson   DATE: 09/17/2021

6) Axel Zumwalt   DATE: 09/17/2021

7) Casey Brandt   DATE: 09/17/2021