COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

CSCI 370 Final Report

Tri-Code

Jacob Kilmer
Mallorie Mackay
Robby Serna

Revised Dec. 12, 2025

CSCI 370 Fall 2025

Mr. Bartel

Table 1: Revision history

Revision	Date	Comments
New	8/27/2025 - 9/7/2025	Updated Sections: • Introduction • Requirements • Risks • Definition of Done • Team Profile
Rev – 2	9/11/25 - 9/21/25	Updated Sections: • Software test and quality • Project ethical considerations • System Architecture
Rev – 3	9/28/25-10/12/25	Updated Sections: • Project Ethical Considerations
Rev - 4	10/13/25-10/26/25	Updated Sections: • Software Test and Quality ○ Test Results • Project Completion Status • Future Work • Lessons Learned
Rev – 5 (Final)	11/3/25- 12/12/25	Updated Sections: • Future Work • Project Completion Status • Software Test and Quality • System Architecture • Acknowledgments • Appendix

Table of Contents

I. Introduction

- This project is a bid/data dashboard website to help better organize and visualize contract data. The client is Infinity Technology Systems, a subset of a concrete construction company.
- Currently, they have too much data to be successfully kept and organized in excel, so they would like a place to easily view and insert bidding data.
- The main goal of the dashboard is to provide visualizations, advanced filtering, and an overall user-friendly alternative to excel.
- Data is user entered or provided by Infinity. Their team will maintain the software after this semester, which means that documentation is a focus.
- Users will be Infinity employees who manage estimations and bidding.

II. Functional Requirements

The end goal is a website that allows for easy access and insight into contract data. This would include:

- Login page to prevent entry of malicious or inaccurate data
- Home page that displays different projects to view
- Project page to show project specific data
- Custom filtering and data page that allows users to see data across all projects based on specific metrics
- Project entry page for adding a project to the database

III. Non-Functional Requirements

- The website must interface with necessary existing Infinity services
- The program must be easily maintained and thoroughly documented
- The program must interact with Infinity's existing database
- The program must have user Authentication or prevention from SQL injection

IV. Risks

Technical Risks:

- **Database Integration Issues**
  - Likelihood: Moderate
  - Impact: Major
  - Mitigation: Maintain close communication with the client for database scheme access and test database connections early in development.
- **Data Visualization Complexity**
  - Likelihood: Likely
  - Impact: Moderate
  - Mitigation: Start with simpler charts/graphs before implementing more advanced visualizations and rely on established visualization libraries.

Skill Risks:

- **Learning Curve associated with new API's and code environments**
  - Likelihood: Likely
  - Impact: Moderate
  - Mitigation: Encourage team to research and experiment early, share knowledge amongst one another
- **Version Control Conflicts**
  - Likelihood: Unlikely
  - Impact: Minor
  - Mitigation: Establish strong communication between team members and utilize strong Git branching conventions
- **Team Availability/Coordination**
  - Likelihood: Likely
  - Impact: Minor
  - Mitigation: Set clear communication expectations and maintain strong scrum/stand-up habits

V. Definition of Done

The finalized product should include:

- A front-end dashboard is implemented using React.js that connects to Infinity's existing SQL database
- Users can view and filter bid data by various fields
- Functional data visualization based on bid data
- Interface supports efficient navigation and uses a clear, user-friendly design

Client Testing/Acceptance:

- Client can successfully connect the dashboard to their SQL database and retrieve data without errors
- Client can run filtering and view visualizations correctly
- Client can verify that the data displayed on the dashboard match the data in their SQL database

Product Delivery:

- The final product will be delivered as a GitHub repository containing all source code and documentation.
- Files that were not included in the GitHub for security purposes have been sent to the client separately.

VI. System Architecture

Below are images of the final iteration on the Bid Dashboard. These figures highlight the key aspects of the design and goals for a final product that we achieved. Our original wireframe designs which did not meet all of the goals, but gave us a rough idea of how we wanted the site to look, are included in Appendix A.

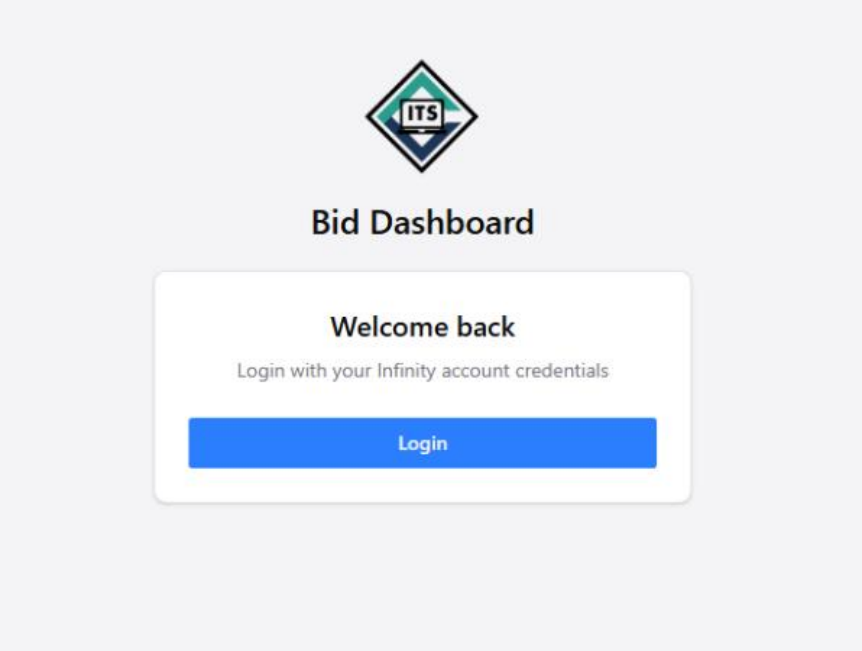
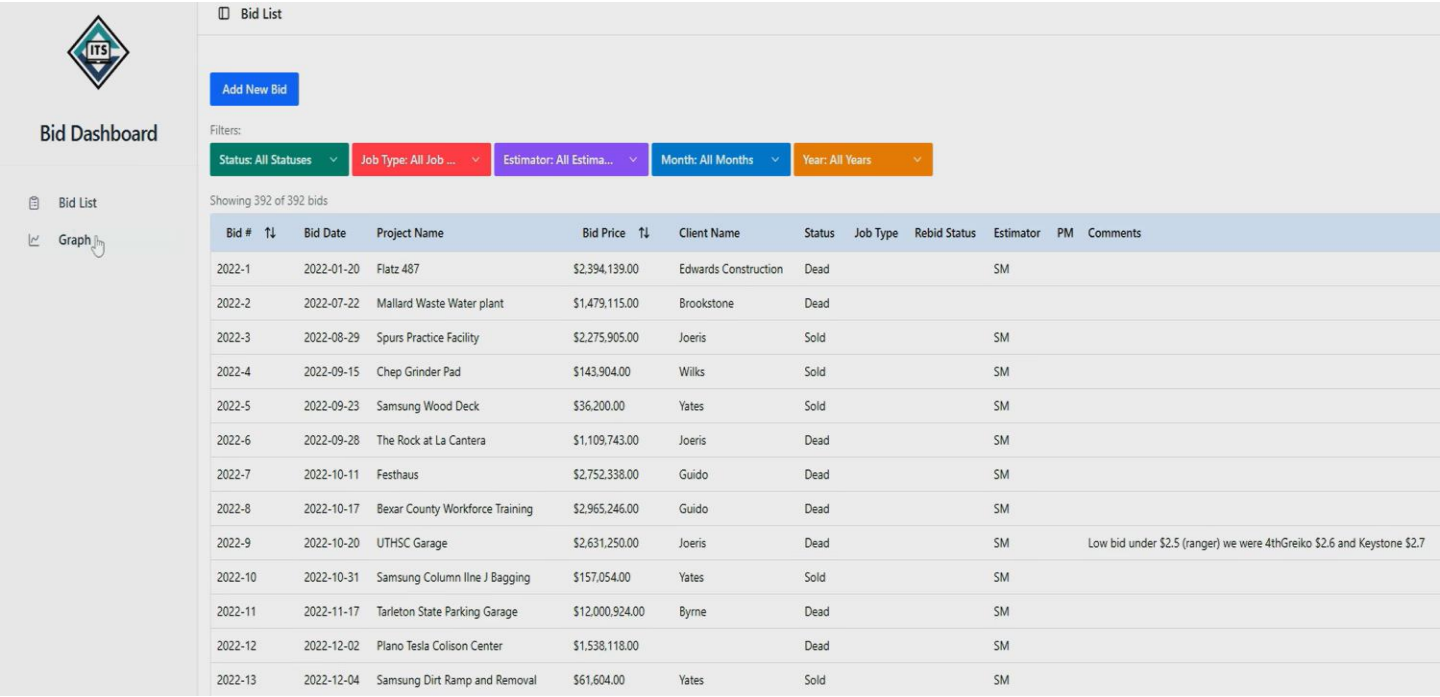


Figure 1: Final Login Page



Bid #	Bid Date	Project Name	Bid Price	Client Name	Status	Job Type	Rebid Status	Estimator	PM	Comments
2022-1	2022-01-20	Flatz 487	$2,394,139.00	Edwards Construction	Dead			SM		
2022-2	2022-07-22	Mallard Waste Water plant	$1,479,115.00	Brookstone	Dead					
2022-3	2022-08-29	Spurs Practice Facility	$2,275,905.00	Joeris	Sold			SM		
2022-4	2022-09-15	Chep Grinder Pad	$143,904.00	Wilks	Sold			SM		
2022-5	2022-09-23	Samsung Wood Deck	$36,200.00	Yates	Sold			SM		
2022-6	2022-09-28	The Rock at La Cantera	$1,109,743.00	Joeris	Dead			SM		
2022-7	2022-10-11	Festhaus	$2,752,338.00	Guido	Dead			SM		
2022-8	2022-10-17	Bexar County Workforce Training	$2,965,246.00	Guido	Dead			SM		
2022-9	2022-10-20	UTHSC Garage	$2,631,250.00	Joeris	Dead			SM		Low bid under $2.5 (ranger) we were 4thGreiko $2.6 and Keystone $2.7
2022-10	2022-10-31	Samsung Column IIne J Bagging	$157,054.00	Yates	Sold			SM		
2022-11	2022-11-17	Tarleton State Parking Garage	$12,000,924.00	Byrne	Dead			SM		
2022-12	2022-12-02	Plano Tesla Colison Center	$1,538,118.00		Dead			SM		
2022-13	2022-12-04	Samsung Dirt Ramp and Removal	$61,604.00	Yates	Sold			SM		

Figure 2: Final Bid List Page

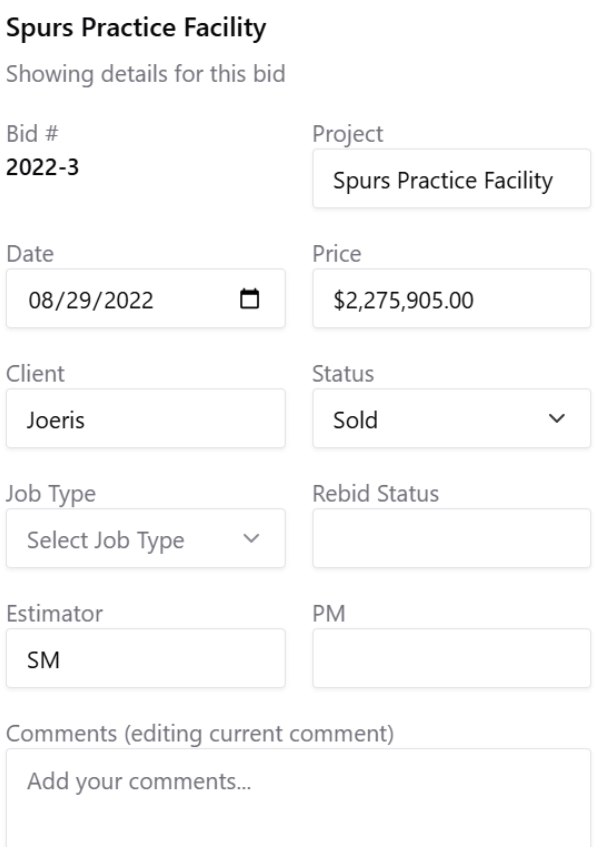


Figure 3: Final Bid Edit Pane

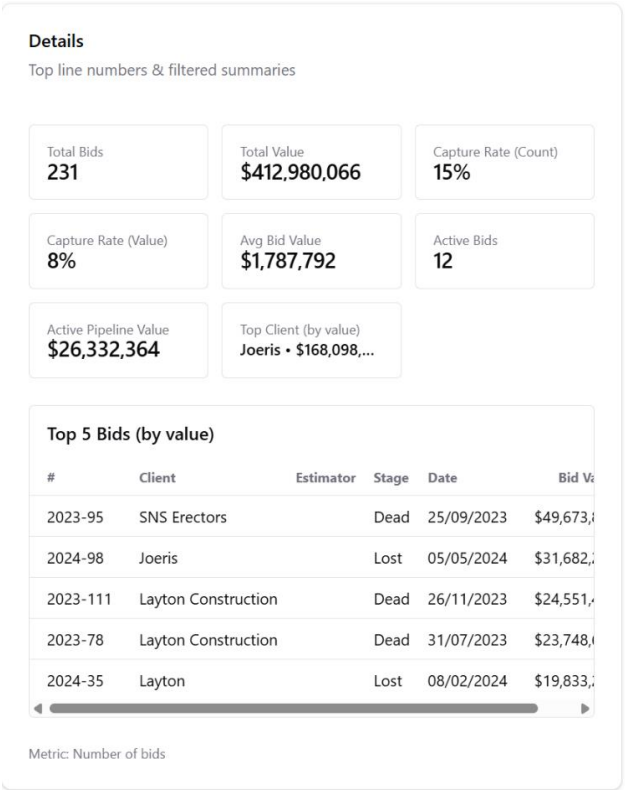


Figure 4: Final Important Details Pane

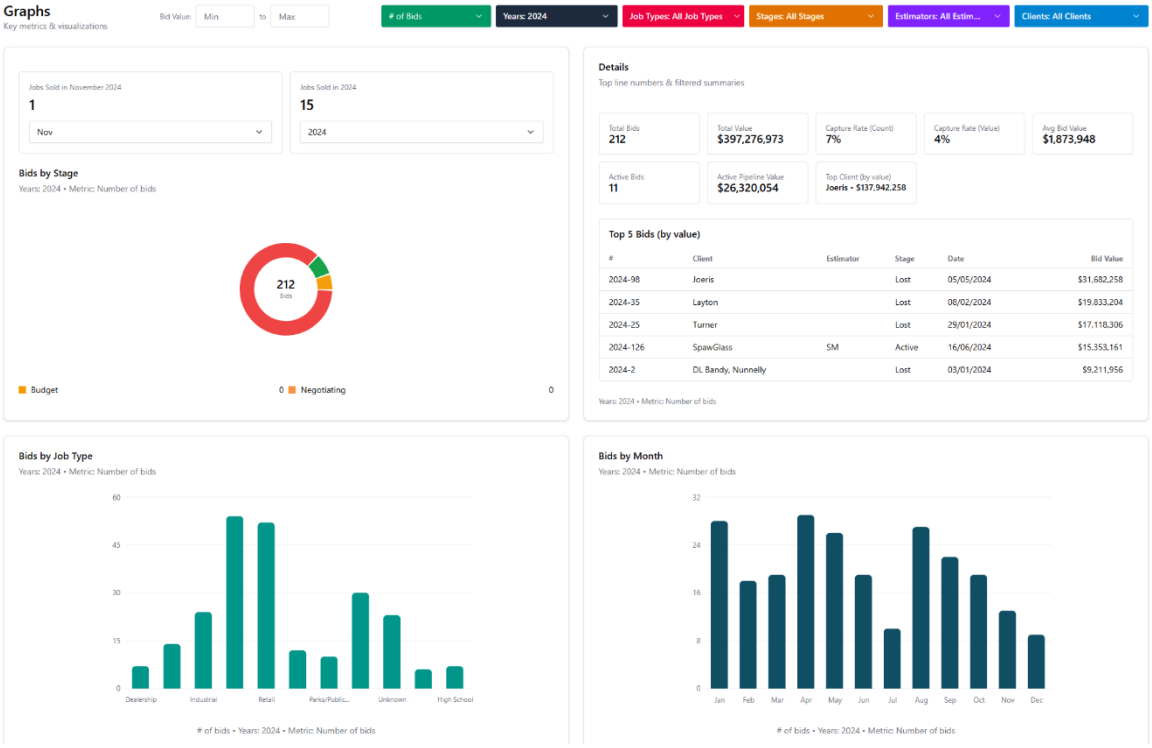


Figure 5: Final Bid Visuilization Page

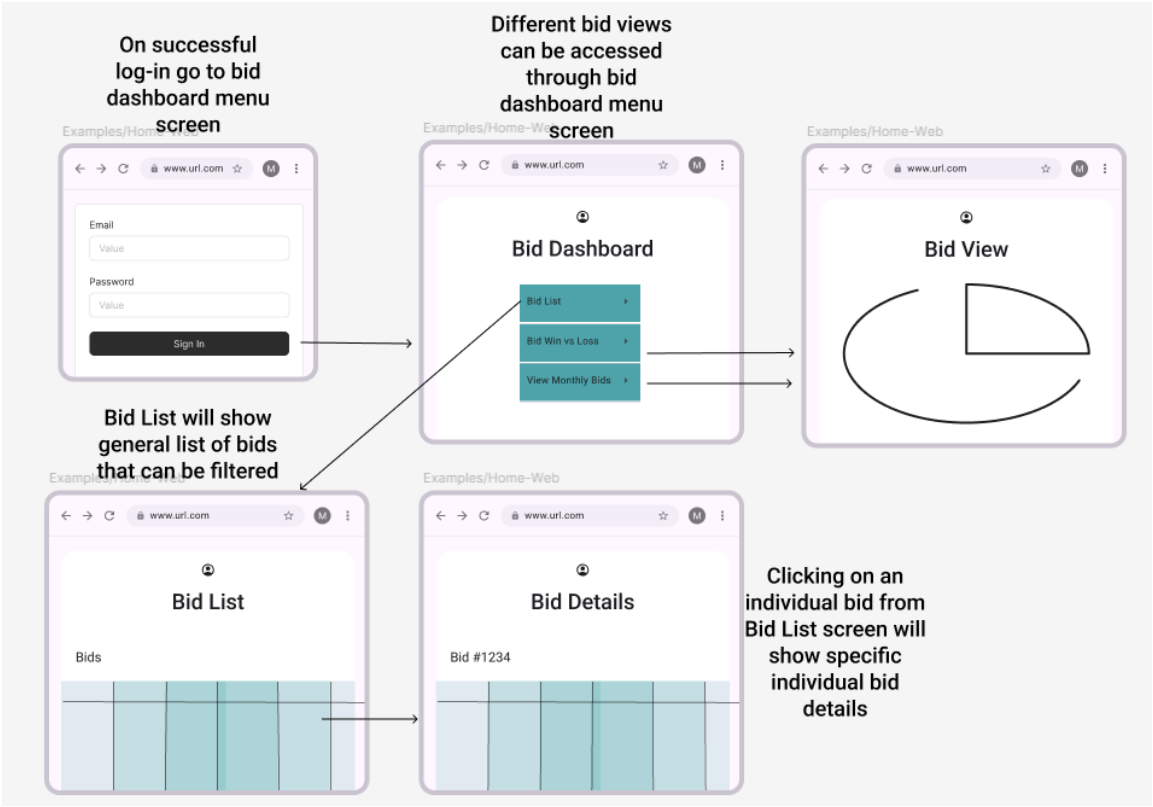


Figure 6: UI Storyboard

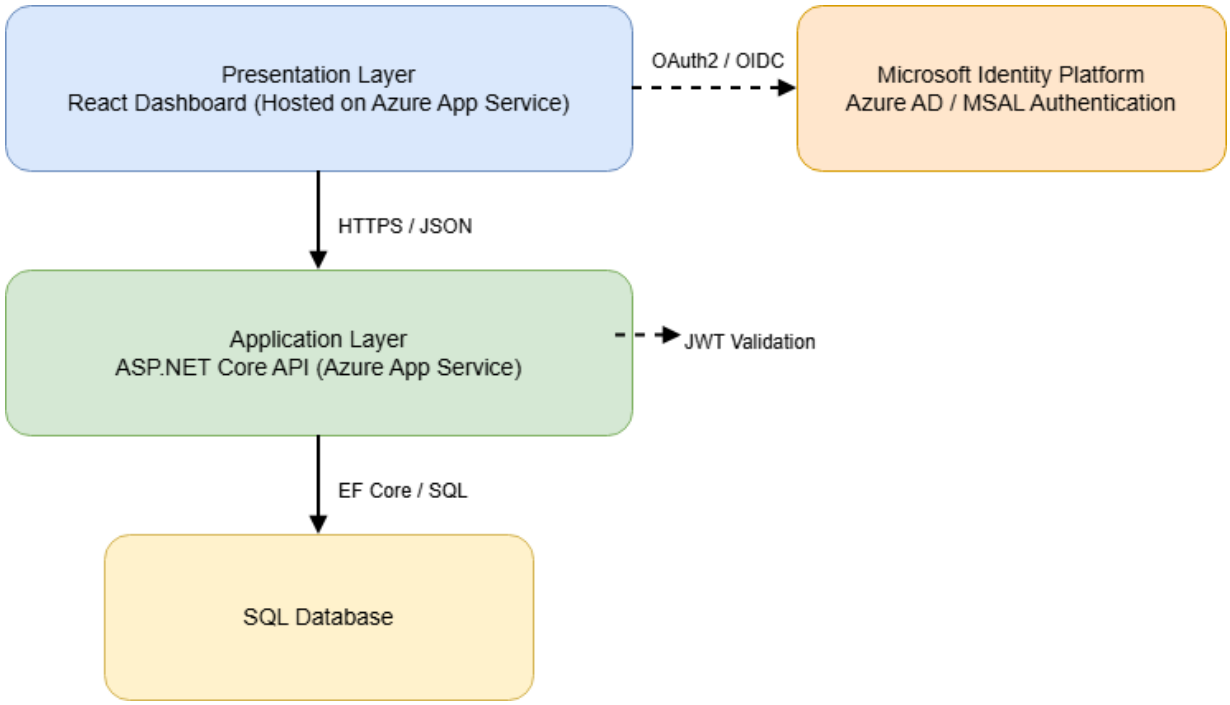


Figure 7: Architecture Overview

VII. Software Test and Quality

Unit Testing

- Implementation*:* Individual React components (such as filtering functions, data tables, and chart components) were tested in isolation using a testing framework, React Testing Library.
- Purpose*:* Detects defects early by verifying that each component behaves correctly with valid and invalid inputs.
- Quality Aspect*:* Defect detection, verification of component logic.

User Interface Testing

- Implementation*:* Manual walkthroughs of the dashboard confirmed that charts, filters, and navigation behave as expected. If feasible, automated UI tests (e.g., with Selenium or Cypress) may be added.
- Purpose*:* Ensures usability and correct behavior of interactive features.
- Quality Aspect*:* Validation from a user perspective.

Integration Testing

- Implementation*:* We tested how the front end communicates with the database. This included verifying that queries return the expected data and that errors are handled correctly.
- Purpose*:* Ensures that separate parts of the system work together as intended.
- Quality Aspect*:* Verification of system interactions and data integrity.

Code Reviews

- Implementation*:* Team members conducted peer reviews before merging code into the main branch on GitHub. Reviews focused on readability, adherence to coding standards, and maintainability.
- Purpose*:* Improves code quality by catching defects and enforcing consistency.
- Quality Aspect*:* Coding standards, defect detection, maintainability.

Documentation

- Implementation: Team members included many comments and descriptions throughout all code files. A final demonstration of the site with instructions for how to use all features was also completed and delivered to the client per their request.
- Purpose: Improves code readability and maintainability allowing for easy maintenance and future improvement by the client
- Quality Aspect: The client is deploying and maintaining the product in the future so making it easier to understand the existing code and iterate is very important.

Test Results:

User Acceptance Testing:

- We conducted user acceptance tests with the client during demos of progress on the dashboard. We were able to validate that the dashboard's layout, functionality, and visualizations met user expectations. Feedback provided valuable insight into how users interpret bid data and what metrics are most meaningful for day-to-day decision-making.
- One user suggested adding filters for specific metrics, while also confirming that the existing filtering options and design effectively capture key information. Another user suggested updating the types of graphs being used for data visualization to better meet employee needs.

User Interface Testing:

- We conducted UI testing by using multiple different web browsers on different devices to confirm a responsive and consistent layout. The testing consisted of a manual walkthrough of all the main functionalities.
- It was also important for the dashboard to be accessible on an iPad while outside in the sun, so walkthroughs were completed to test readability and other interface aspects on an iPad. Minor layout inconsistencies were identified while testing the UI; these issues will be addressed in the next version.

Integration Testing:

- We verified the proper interaction between the front-end and the database through the back-end connection. All queries tested were done in a live capacity on a mock database that contains older client data.
- Testing ensures that all CRUD operations, specifically data retrieval, filtering, and updates, function as intended.
- Error handling was also evaluated by simulating malformed queries and invalid inputs (incorrect bid creation or updates) to confirm that the dashboard handles and responds appropriately.

Unit Testing:

- We conducted unit tests on individual React components such as filtering functions, data tables, and chart modules to ensure all were working as expected.
- Each component was tested in isolation to ensure correct behavior with both valid and invalid inputs. These tests were focused on element behavior rather than interface and layout issues (these were covered in UI testing).

VIII. Project Ethical Considerations

ACM/IEEE Code of Ethics and Professional Practice:

Principles which are particularly pertinent to the development of our product are listed below [1].

Client and Employer

- Confidentiality: Creating the bid dashboard requires that we have access to company data, so it is especially important to maintain confidentiality and keep sensitive information private.
- Integrity: Our team will follow all procedures to ensure that the software is created legally, ethically, and with proper authorization at all stages.
- Artificial Intelligence: Any use of A.I. will be done in accordance with client policies. Code will never be entered into personal generative A.I. accounts, only the authorized client account when necessary.

Product

- Accuracy, reliability, and security of the dashboard's outputs are essential. Having accurate bid information is integral to business operations, making our professional practices surrounding the product a top priority.

Management

- Management will play the biggest role within our team throughout the project. Having clear practices for version control, testing, and data protection will ensure accountability and a reliable product.
- Our team will maintain transparency about timelines and uncertainties, while also keeping open communication with the client.

Principles which are most in danger of being violated during the development of our project, as well as ways that we will avoid violating them, are listed below [1].

Product

- The biggest potential danger is lacking some testing and validation due to the time constraints placed on us. This could have a negative impact on the accuracy and security of the dashboard. To avoid this, we will uphold all aspects of the quality assurance plan to the highest standards.

Judgment

- Judgement may be suppressed when pressured by deadlines, resulting in overpromising features or ignoring risks in the system design. Our team will avoid this by documenting concerns relating to feasibility and risk, so that we can communicate amongst ourselves, and raise concerns when necessary.

Tests articulated by Michael Davis have been applied to the Bid Dashboard below. [2]

- The **Harm Test** states that the benefits of the bid dashboard must outweigh the potential harm. We believe the potential harms relate to data privacy and correctness. Having access controls for the data, as well as testing the data accuracy across the dashboard, will prevent potential risks.
- The **Publicity Test** asks how the choices made would look on the front page of the newspaper. We have designed the system and are committed to building it in a way that would reflect responsible and ethical software engineering choices, given any publicity it would receive.
- The **Professional Test** asks what the ACM code of ethics says about the problem being solved. We have already gone over many of the ACM principles above, which we are committed to upholding throughout the development of the dashboard.

A poorly implemented quality plan could lead to incomplete data visualization, inaccurate bid data, or system crashes. The dashboard will handle sensitive financial and strategic information, and without proper quality controls, vulnerabilities could expose client data. A lapse in software quality would compromise our ethical responsibilities to the client, product, and public, and will not be tolerated within the team. We view the quality plan as strictly non-negotiable and will uphold it through testing, peer code reviews, and thorough documentation. Ethically, we are committed to ensuring that the dashboard is trustworthy and secure.

IX. Project Completion Status

- The goal of this project was to create a web-based dashboard with a responsive front end and a backend that connects the UI to Infinity's existing database. The dashboard is designed to replace the company's spreadsheet-based system, providing a more efficient way to edit, filter, and visualize bid data.
- In our final delivered version, the core structure and user interface have been fully implemented in React, including navigation, data tables, bid adding/editing/deleting, and chart visualization components. A structured API layer was completed for the backend, defining standardized endpoints for all CRUD operations.
- A context file was created for state management to cache bid data, reducing backend load and improving dashboard responsiveness. This structure supports dynamic filtering and real-time UI updates.
- A user authentication system has been fully implemented through Infinity's existing Microsoft Sign In featuring secure token handling, protected route enforcement, and session-aware UI rendering of bid data.
- Responsive design principles were applied to ensure the dashboard functions across varying screen sizes, including large buttons for ease of use on tablets and smartphones.
- The front end also has client-side form validation with descriptive error messaging safeguards to ensure data integrity during bid creation and editing.
- Advanced filtering and visualization features have been implemented and tested using real bid data, ensuring the functionality performs as expected. All filters and charts implemented align with Infinity's workflow including desired chart types, filtering metrics, and highlighted values such as capture rate and top bids.
- The delivered product meets all goals set in the definition of done.

X. Future Work

- For the future expansion of the dashboard, a feature to allow users to create custom graph visualization based on bid data could be added.
- Another feature discussed with the client that could be implemented in a future version is the ability to save filters for a specific user, allowing them to store and use the filters across different sessions or devices.
- While developing the dashboard, we had the idea of implementing an export feature that would give users the ability to save key visualization or filtered data to a pdf. This would not be very difficult to implement, but exporting a data summary was not a priority for the client.
- A future version of the dashboard could also have more role-based access control to determine what elements or bids can be viewed or edited by different employees.

XI. Lessons Learned

- React turned out to be a powerful and flexible framework. When we first started working on this project, we took some time to decide on a framework to use for the front-end. After some debate we decided on React, which has proved to be very helpful throughout the development process. The component-based setup made it easy to organize the user interface into reusable parts, which greatly simplified development, as well as testing. The underlying responsiveness of react made adapting the dashboard to different devices much easier than past projects some of us have worked on. Overall, React was a very good choice for the dashboard.

- UI component libraries, such as shadcn significantly sped up development. These prebuilt and customizable components ensure consistent styling and improved ease of use across the dashboard. Before doing some research and deciding to use shadcn, we tried to create some components from scratch because of the customization. We learned quickly that many component libraries exist which allow for complete customization, without having to build all the components from the ground up. Instead of spending time coding repetitive interface elements from scratch, we were able to focus on higher level functionality, which gave us a much more polished and user-friendly product.

- Client feedback was crucial throughout the development process. Having communication with the product users during client meetings was helpful in making sure the dashboard was using metrics and visualization styles that are valuable to the client's daily workflow. The feedback loop led to many iterations of the dashboard, each improving usability for end users. Being involved with the client and end users early on in development led to much less backtracking on design iterations as well as allowing us to fit the product as best we could to the client's needs.

- SCRUM methodologies have improved the overall workflow of our team. The class structure of two-week sprints allowed us to set specific goals and focus on delivering small iterations. This also made it easier to adapt to new requirements or ideas thrown out by the client much quicker, and with less work being redone. We also learned quickly that regular stand-up and sprint planning meetings were very important. Having these meetings kept our team on the same page and allowed us to identify potential issues before they became real. Overall, the standups kept us on track to meeting deadlines for class and kept the product development on track to meet the definition of done that we set at the beginning of the project.

XII. Acknowledgments

We would like to give a huge thank you to Infinity Technology Systems for bringing us this project and providing extensive insight into their daily workflows and processes as it relates to tracking and analyzing bid data. We want to offer a special thank you to Jennifer Wood, Deron Wood, and Jonas Edelstein, who have been our mentors in all stages of the software development life cycle throughout this semester. Their constant support and guidance were integral to the development of the Bid Dashboard as well as our personal growth in software development practices. The Infinity team took time out of their busy schedules to meet with us weekly for check-ins, product demos, and code debugging, all of which we are extremely grateful for.

We would like to extend our sincere appreciation to our field session advisor, Caleb Bartel. Throughout the semester, Caleb also took time out of his busy schedule to meet with us every sprint, being very flexible in doing so. The sprint retrospectives we completed with him helped us improve our communication, backlog management, and client relationship. Caleb also provided much needed feedback on class assignments and the design proposal presentation which helped us improve the final report and presentation.

Finally, we would like to thank the CSCI 370 professors, Kathleen Kelly and Donna Bodeau, for all their work in setting up an extremely valuable computer science field session. Professor Kelly has designed this class and brought in a wide variety of clients as well as guest speakers from industry who have provided insight into software development and life after graduation. All of which allowed us to have a great experience this semester. Thank you!

XIII. Team Profile

Robby Serna

Senior - Computer Science

Hometown: Aurora, Colorado

Work Experience: KCCA SW Consultants LLC Software Developer, Warehouse Associate

Hobbies: Guitar, Snowboarding, Gym

I am very excited to be part of the project and be able to gain hands on experience developing a product that will help used in industry.

Jacob Kilmer

Senior – Computer Science and Computer Engineering

Hometown: Aurora, Colorado

Hobbies: photography, hiking, playing/watching sports

I am looking forward to working with mentors from Infinity Technology Systems and creating a real-world product.

Mallorie Mackay

Senior – Computer Science

Hometown: Houston, Texas

Work Experience: CACI Software full stack dev intern, 2 years

Hobbies: Art, gaming, fishing

I am looking forward to applying my previous knowledge to new frameworks and languages and creating a product in an industry outside of my comfort zone.

References

[1] D. Gotterbarn, K. Miller, and S. Rogerson, “The Software Engineering Code of Ethics and Professional Practice,” Association for Computing Machinery, https://www.acm.org/code-of-ethics/software-engineering-code (accessed Sep. 17, 2025).

[2] M. Davis, “Ethics across the curriculum,” *Teaching Philosophy*, vol. 16, pp. 205–235, 1993, doi: 10.5840/teachphil199316344.

Appendix A – Original System Design

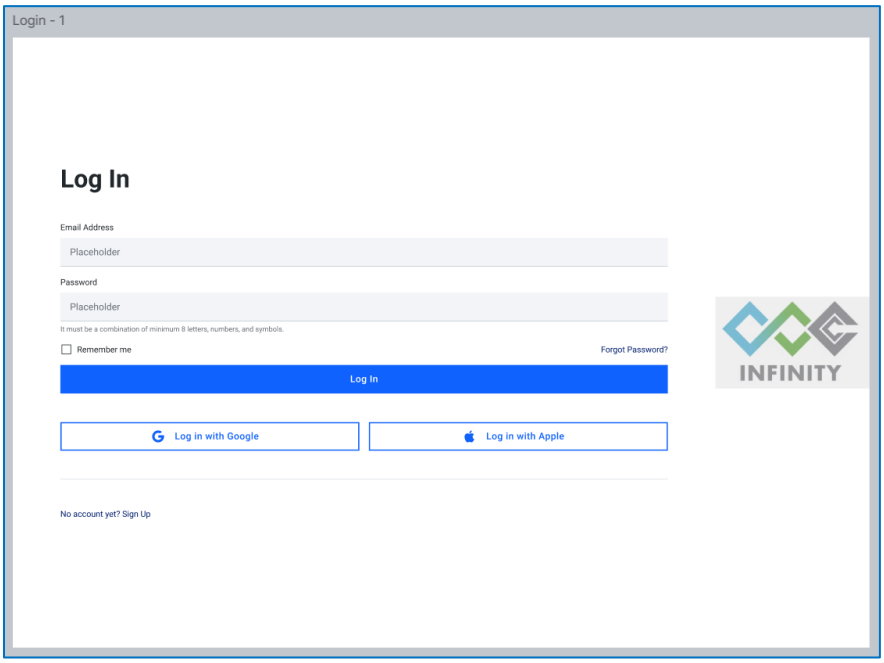


Figure A1: Log In page wireframe

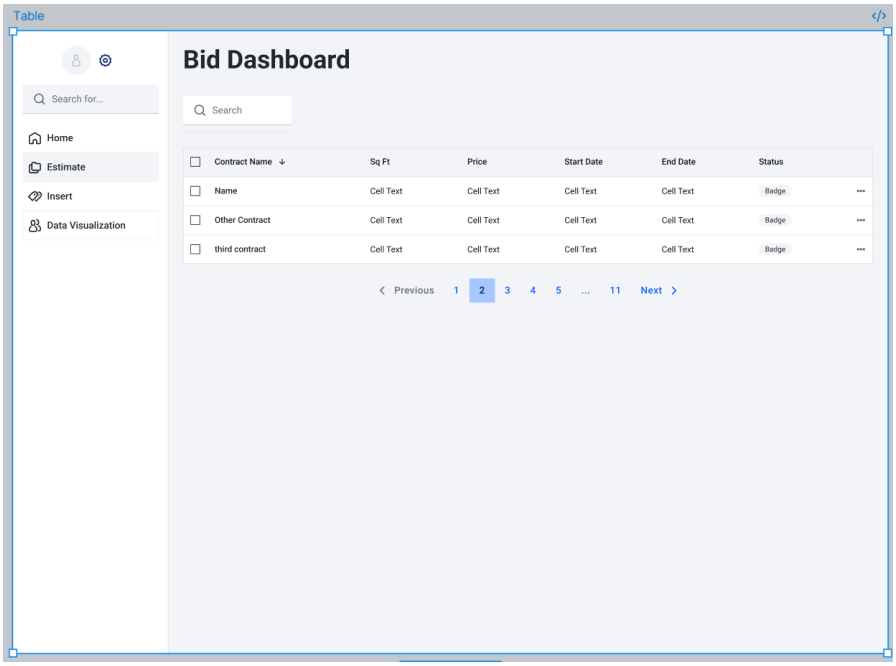


Figure A2: Bid Dashboard Wireframe

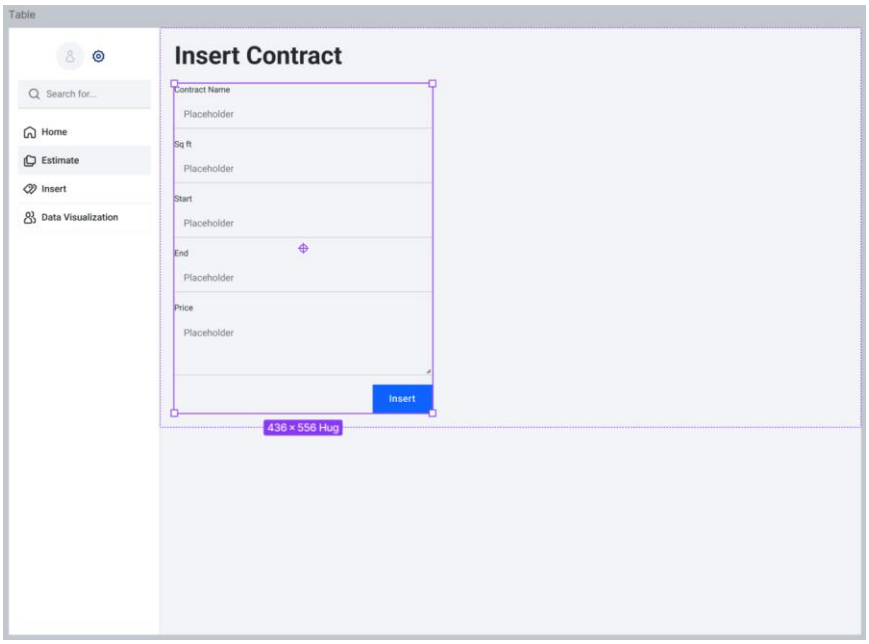


Figure A3: Contract Insertion page wireframe

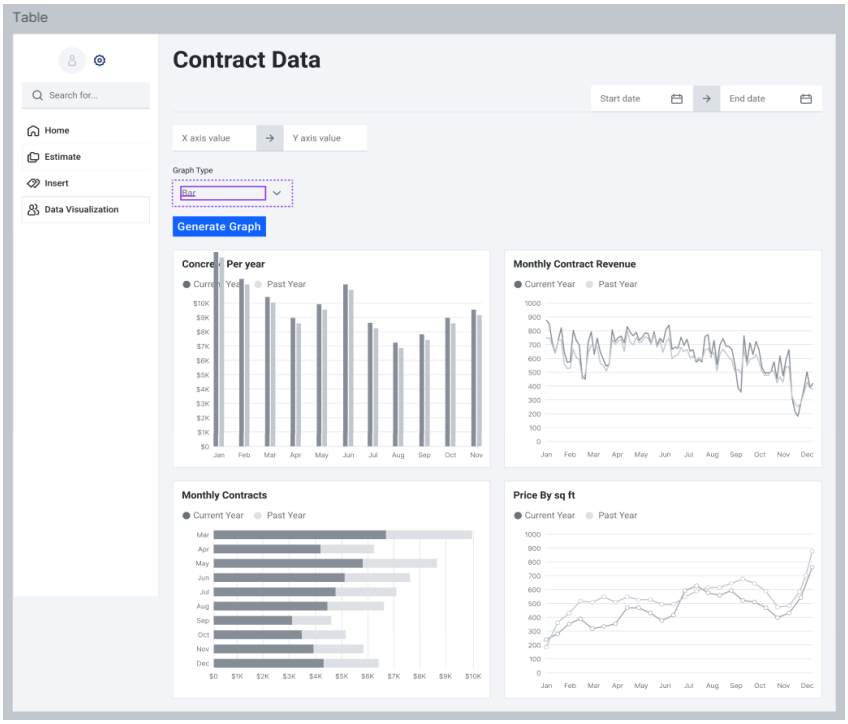


Figure A4: View Contract Data wireframe

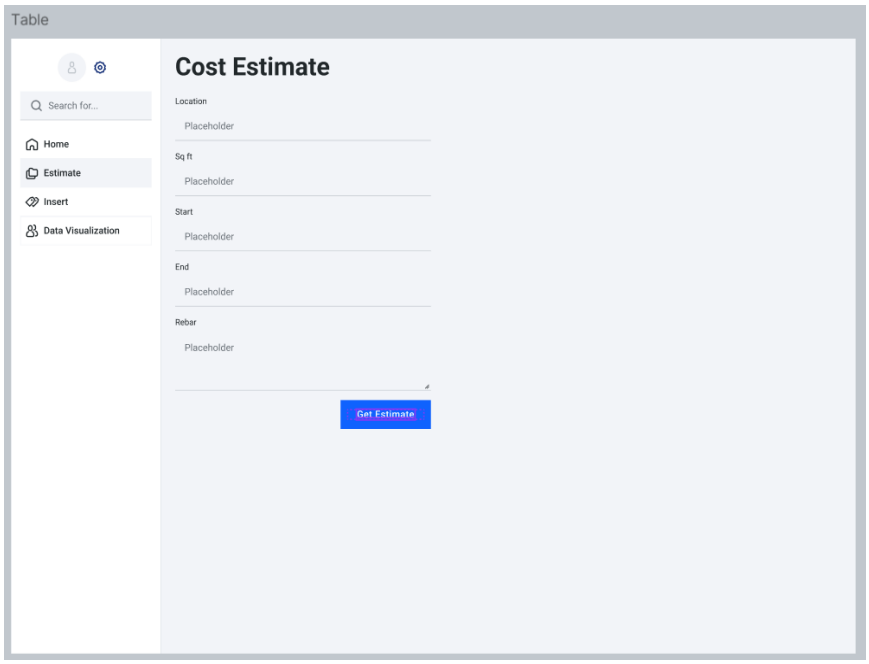


Figure A5: Contract filtering wireframe