

Product TRACkER

DESIGN DOCUMENT

Team:

SdMay20-02

Client:

Collins Aerospace

Adviser:

Mai Zheng

Team Members/Roles:

Ryan Connolly — Subject Matter Expert

Tyler Krueger — Meeting Scribe

Clayton Nida — Frontend Engineer

Jamie Raught — Team Lead

Nathan Shull — Test Engineer

Nathan Yasosky — Report Manager

Team Email:

sdmay20-02@iastate.edu

Team Website:

sdmay20-02.sd.ece.iastate.edu

Revised:

10/12/19 --Version 1

Executive Summary

Development Standards & Practices Used

- Using SCRUM Development Style
- Continuous Integration
- Agile Software Programming
- Clean Coding Practice
- Model View Controller Setup
- Controller, Service, Repository Backend Design

Summary of Requirements

1. Maintainable electronic database

- Edit data manually
- Upload new/updated data via Excel file
 - Excel file can be converted to .csv or .txt

2. Search by field values (incl partial) or upload CPN list via Excel

- Display results on screen
- Download results to Excel file

3. Integration of product and contracts data

- One product can be on multiple contracts
- One contract can include multiple products

4. Interactive

- Linking allows moving around between products/contracts

Applicable Courses from Iowa State University Curriculum

- ComS 227
- ComS 228
- ComS 309
- ComS 319
- SE 329
- Engl 314
- SE 409

New Skills/Knowledge acquired that was not taught in courses

- Front end framework: React
- Connecting Spring Project to MySQL database
- Controller, Service, Repository Backend layout
- SCRUM development standards
- Agile Software Programming Style
- Project Management
- Designing a Gantt Chart

Table of Contents

1 Introduction	4
1.1 Acknowledgement	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	4
1.5 Intended Users and Uses	4
1.6 Assumptions and Limitations	5
1.7 Expected End Product and Deliverables	5
2. Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	6
2.3 Development Process	6
2.4 Design Plan	6
3. Statement of Work	6
3.1 Previous Work And Literature	6
3.2 Technology Considerations	7
3.3 Task Decomposition	7
3.4 Possible Risks And Risk Management	7
3.5 Project Proposed Milestones and Evaluation Criteria	7
3.6 Project Tracking Procedures	7
3.7 Expected Results and Validation	7
4. Project Timeline, Estimated Resources, and Challenges	8
4.1 Project Timeline	8
4.2 Feasibility Assessment	8
4.3 Personnel Effort Requirements	8
4.4 Other Resource Requirements	8
4.5 Financial Requirements	9
5. Testing and Implementation	9
5.1 Interface Specifications	9
5.2 Hardware and software	9

5.3	Functional Testing	9
5.4	Non-Functional Testing	9
5.5	Process	10
5.6	Results	10
6.	Closing Material	10
6.1	Conclusion	10
6.2	References	10
6.3	Appendices	10

List of figures/tables/symbols/definitions

(This should be similar to the project plan)

1 Introduction

1.1 ACKNOWLEDGEMENT

Throughout the course of the project, we will be assisted by our faculty advisor Mai Zheng. His guidance will help us stay on track, and develop the best plan of attack. Not only will he help keep our team running smoothly, but has already helped tremendously to keep us on track and make sure our project is following and meeting the expectations of the senior design class. Additionally we have and will continue to be in close communication with Jenny Smith. She has already helped out tremendously to relay the problem that Collins has enlisted our help to solve. Constant communication with her has provided us with guidance on design decisions along with the ability to understand the aspects more closely relating to Collins issue we are trying to solve.

1.2 PROBLEM AND PROJECT STATEMENT

1.2.1 General Problem Statement

The original reason for this project and why Collins Aerospace enlisted our help was they did not feel that they had a great system to manage not only their products but also their contracts. The way that they had a lot of important information for both contracts and products stored was through a mess of a wide variety of different excel documents, text files, and other forms of media. They could not find a lot of the important information they needed on products when drafting contracts with customers, and if they could find the information it was disorganized that it was hard to get the appropriate feedback. This then resulted in contracts that were being created, but could not be met. One example of this was the product was no longer being produced, but the contract would state that they would be producing this product for a while outside of the products end of production date. This caused conflicting information, and often times put Collins Aerospace into tough predicaments or into positions where they could not hold up their end of the contract and would have to figure out a way to still fulfill the contract.

1.2.2 General Solution Approach

In order to help solve this problem for Collins Aerospace our goal is to develop a web application that will manage their contracts and products information. This will allow them to not only have all of their data in one place, but will allow them to have a way to search their data and make sure they are coming up with contracts that they know without a doubt will be able to be met for the customer. Therefore our application will need to be able to list all contacts and products collins currently has. Then it needs to be able to navigate this application easily and search for the exact product or contract that the user is looking for. Additionally the admins of the system will need a way to add more data into the system, that way the application can be long lasting and stay up to date as the company evolves. The last important key about this application is it will need to be able to tell the user the connections that a product has to however many contracts or on the other side a contract and how many products it has associated with it. That way Collins can easily assure customers that their contracts can still be met even as time progresses.

1.3 OPERATIONAL ENVIRONMENT

The operational environment for this project is a computer that is able to receive access to the collins network, in order to hit the website that our application will be deployed on. Since our application is strictly software, the environment on which it is deployed is not going to affect the application. Therefore, there are no physical hazards that we will need to take into consideration while developing this application. However, it is important that we design an application and deploy it an environment that allows the application to have the necessary security regulations on it along with a low no down time environment.

1.4 REQUIREMENTS

List all requirements for your project – functional requirements within your project context, economic/market requirements, environmental requirements, UI requirements, and any others relevant to your project.

1.5 INTENDED USERS AND USES

– To properly design an end product that will provide the maximum satisfaction and perform in the most efficient manner, it is essential to understand the end user and the associated end uses.

1.6 ASSUMPTIONS AND LIMITATIONS

- Two separate lists, with a short justification as needed.
- Extremely important, as it can be one of the primary places where the client can go to determine if the end product will meet their needs.
- Examples of assumptions: The maximum number of simultaneous users/customers will be ten; Blue is the best background color and will be used; The end product will not be used outside the United States.
- Example of limitations: The end product shall be no larger than 5”x8”x3” (client requirement); The cost to produce the end product shall not exceed one hundred dollars (a market survey result); The system must operate at 120 or 220 volts and 50 or 60 Hertz (the most common household voltages worldwide).
- For limitations, include tests not performed, classes of users not included, budget/schedule limitations, geographical constraints, etc.

1.7 EXPECTED END PRODUCT AND DELIVERABLES

These tie in with the goals. What deliverables are necessary to meet the goals outlined in the introduction?

List the end product and any other items, along with a brief description, that will be delivered to the client prior to the end of the project.

- If the end product is to be commercialized, the description shall be of the commercialized end product.
- It shall be in the form of a technical product announcement, as opposed to a product advertisement, and shall not include a list of technical specifications.
- Any other items that will be delivered to the client shall also be included and described unless their definition and description are obvious.
- Examples might include a household power supply to eliminate the need for batteries, a user's manual, or other project reports.
- There shall be at least a one-paragraph description for each item to be delivered.
- Delivery dates shall also be specified.

2. Specifications and Analysis

2.1 PROPOSED DESIGN

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested, etc?
- We want to know what you have done
- Approach methods should be inclusive of **functional and non-functional requirements** of the project, which can be repeated or just referred to in this section

If your project is relevant to any **standards** (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here

2.2 DESIGN ANALYSIS

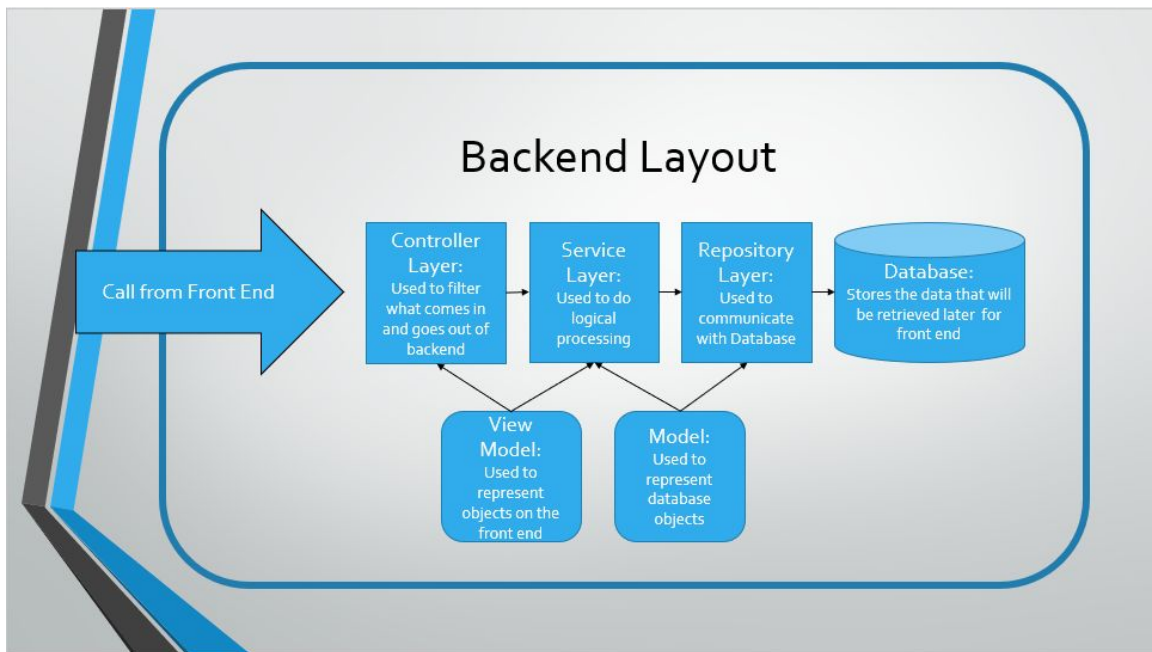
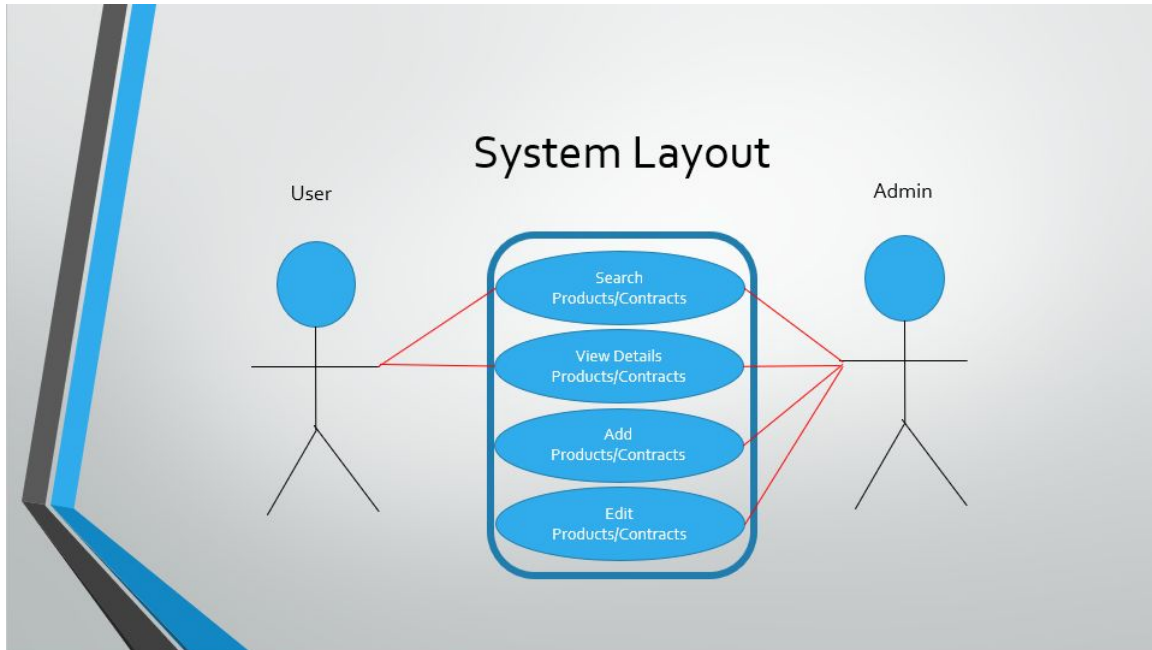
- Discuss what you did so far
- Did it work? Why or why not?
- What are your observations, thoughts, and ideas to modify or continue?
- If you have key results they may be included here or in the separate “Results” section
- Highlight the **strengths, weakness**, and your observations made on the proposed solution.

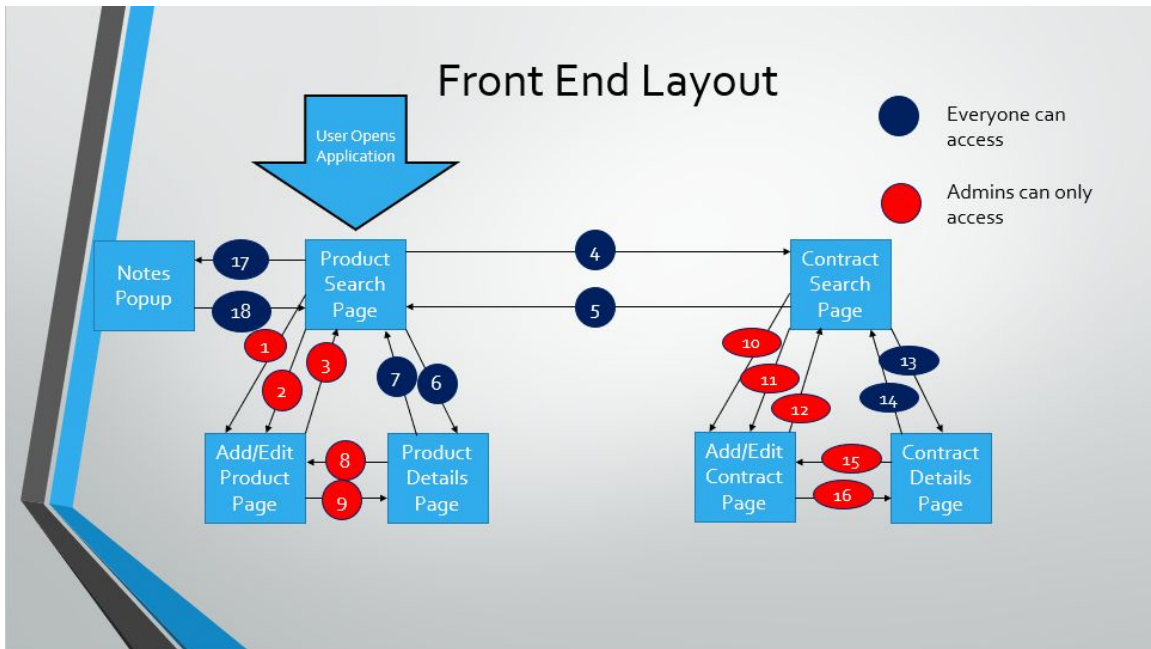
2.3 DEVELOPMENT PROCESS

We are following an Agile process with elements of a SCRUM framework. This allows us to break up the tasks into monthly Increments and weekly Sprints.

2.4 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.



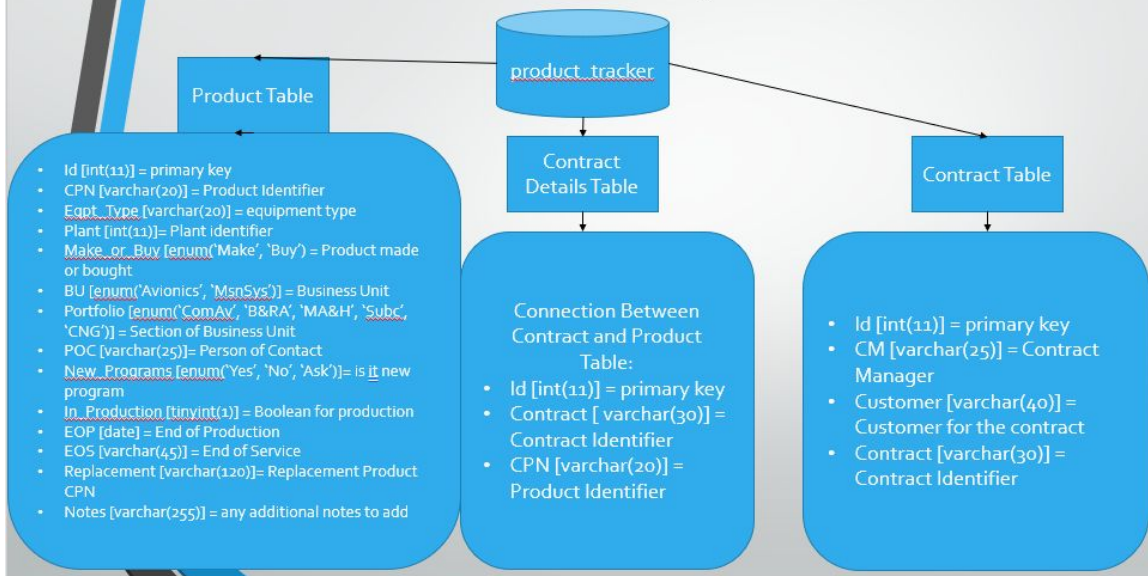


- ## Front End Transitions
1. Click the add button to insert a new product
 2. Click on a product and clicking the edit button to edit the product
 3. Click the x on the add/edit popup to go back to search page
 4. Clicking on the Contract Tab to bring you to the contract search page
 5. Clicking on the Product Tab to bring you to the product search page
 6. Clicking on the CPN of the Product to bring up the products detail page
 7. Clicking on the x to take you back to the product search page
 8. Clicking on the Edit Button on the Product Details Page
 9. Clicking the x on the edit popup to take you back to the product details page

Front End Transitions (Cont.)

10. Click the add button to insert a new contract
11. Click on a contract and clicking the edit button to edit the contract
12. Click the x on the add/edit popup to go back to search page
13. Clicking on the Contract Number of the Contract to bring up the contract detail page
14. Clicking on the x to take you back to the contract search page
15. Clicking on the Edit Button on the Contract Details Page
16. Clicking the x on the edit popup to take you back to the contract details page
17. Open the notes on a product by clicking notepad on the side of the listing
18. Clicking the x to go back the Product Search Page

Database Layout



Product Details

CPN Plant

Eqpt Type Make Buy

POC BU

New Programs Yes No Ask

Notes

Associated Contracts

Portfolio

EOP

EOS

Replacement

Close

Notes

Notes

Close

Add Product

CPN Plant

Eqpt Type Make Buy

POC BU

New Programs Yes No Ask

Notes

Associated Contracts

Portfolio

EOP

EOS

Replacement

Upload Data Add Data Cancel

Edit Product

CPN Plant

Eqpt Type Make Buy

POC BU

New Programs Yes No Ask

Notes

Associated Contracts

Portfolio

EOP

EOS

Replacement

Save Cancel

3. Statement of Work

3.1 PREVIOUS WORK AND LITERATURE

Include relevant background/literature review for the project

- If similar products exist in the market, describe what has already been done

- If you are following previous work, cite that and discuss the **advantages/shortcomings**
- Note that while you are not expected to “compete” with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

3.2 TECHNOLOGY CONSIDERATIONS

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.3 TASK DECOMPOSITION

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

3.4 POSSIBLE RISKS AND RISK MANAGEMENT

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

3.5 PROJECT PROPOSED MILESTONES AND EVALUATION CRITERIA

What are some key milestones in your proposed project? Consider developing task-wise milestones. What tests will your group perform to confirm it works?

3.6 PROJECT TRACKING PROCEDURES

Our team is using the GitLab Issues board to manage tasks and track our progress.

3.7 EXPECTED RESULTS AND VALIDATION

What is the desired outcome?

How will you confirm that your solutions work at a **High level**?

4. Project Timeline, Estimated Resources, and Challenges

4.1 PROJECT TIMELINE

- A realistic, well-planned schedule is an essential component of every well-planned project
- Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity
- A detailed schedule is needed as a part of the plan:
 - Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.
 - Annotate the Gantt chart with when each project deliverable will be delivered
- Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.

4.2 FEASIBILITY ASSESSMENT

Realistic projection of what the project will be. State foreseen challenges of the project.

4.3 PERSONNEL EFFORT REQUIREMENTS

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just "X" hours per week for the number of weeks that the task is active

4.4 OTHER RESOURCE REQUIREMENTS

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

4.5 FINANCIAL REQUIREMENTS

If relevant, include the total financial resources required to conduct the project.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case
5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 INTERFACE SPECIFICATIONS

- Discuss any hardware/software interfacing that you are working on for testing your project

5.2 HARDWARE AND SOFTWARE

- Indicate any hardware and/or software used in the testing phase
- Provide brief, simple introductions for each to explain the usefulness of each

5.3 FUNCTIONAL TESTING

Examples include unit, integration, system, acceptance testing

5.4 NON-FUNCTIONAL TESTING

Testing for performance, security, usability, compatibility

5.5 PROCESS

- Explain how each method indicated in Section 2 was tested
- Flow diagram of the process if applicable (should be for most projects)

5.6 RESULTS

- List and explain any and all results obtained so far during the testing phase
 - - Include failures and successes
 - - Explain what you learned and how you are planning to change it as you progress with your project
 - - If you are including figures, please include captions and cite it in the text
 - This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place
- Modeling and Simulation:** This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.
- List the **implementation Issues and Challenges.**

6. Closing Material

6.1 CONCLUSION

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 REFERENCES

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

6.3 APPENDICES

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.