



# WPI

## Creating and Updating Effective Internal UIs for a Large Financial Company

### **Project Team:**

Ido Katz            ikatz@wpi.edu

Caitlin Kuzma    cmkuzma@wpi.edu

Christopher Lobo    calobo@wpi.edu

Callan Moriarty    cmoriarty@wpi.edu

### **Project Advisors:**

Professor Wilson Wong, Department of Computer Science

Professor Robert Sarnie, School of Business

Professor Renata Konrad, School of Business

*This report represents the work of WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, please see <http://www.wpi.edu/academics/ugradstudies/project-learning.html>*

# Abstract

Our team's project was completed in collaboration with a large financial investment firm. We worked with a software development team in the workplace investing sector on two projects. First, we worked on revising the user interface of an ID cross referencing tool to condense the features, make it more user friendly, and update the user interface to the Angular framework—the current company application standard. Then, we created a similar Angular-based user interface design for a brand new cache refreshing tool. Our work will assist in the company's efforts of moving its internal tools onto the cloud and the Angular framework so they can more easily be updated. Throughout our project, we utilized an Agile Kanban Methodology to organize our team.

# Table of Contents

<b>Abstract</b>	<b>ii</b>
<b>Table of Contents</b>	<b>iii</b>
<b>Table of Figures</b>	<b>v</b>
<b>Table of Tables</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Company Background	1
1.2 User Interface	1
1.3 Project Goals	1
<b>2 Background</b>	<b>2</b>
2.1 Domain Research	2
2.2 User Interface Design Rules	3
2.3 Similar Applications	4
2.4 Purpose and Impact	4
<b>3 Methodology</b>	<b>5</b>
3.1 Software Development Methodology	5
3.2 Agile Scrum Methodology	6
<b>4 Software Development Environment</b>	<b>7</b>
4.1 Project Management Software and Programming Environment	7
4.2 Data Sources	8
<b>5 Software Requirements</b>	<b>8</b>
5.1 Requirements Gathering	8
5.2 Functional and Nonfunctional Requirements	8
5.3 Epics and User Stories	9
<b>6 Design</b>	<b>10</b>
6.1 ID Cross Reference Tool	10
6.1.1 Current Model and Proposed Redesign	10
6.1.2 Architectural Design	12
6.1.3 Final Design	15
6.2 Refresh Cache Tool	15
<b>7 Software Development</b>	<b>16</b>
7.1 Week 1 (10/25-10/29)	16

7.2 Week 2 (11/1-11/5)	17
7.3 Week 3 (11/8-11/12)	19
7.4 Week 4 (11/15-11/19)	21
7.5 Week 5 (11/22-11/24)	23
7.6 Week 6 (11/29-12/3)	24
7.7 Week 7 (12/6-12/10)	27
7.8 Overall Burndown Charts	29
<b>8 Business Perspective</b>	<b>30</b>
8.1 Primary Business Risks	30
8.2 Business Rewards	31
8.3 Other Risks	31
8.4 Risk Culture	32
8.5 Financial Savings	32
8.6 Business Learnings	36
8.7 Key Business Takeaways	37
<b>9 Assessment</b>	<b>38</b>
9.1 Achievements	38
9.2 Major Takeaways	39
9.3 Technical Learnings	39
9.4 Self Reflection for Industrial Engineering Majors	40
<b>10 Future Development</b>	<b>41</b>
<b>11 Conclusion</b>	<b>41</b>
<b>Bibliography</b>	<b>42</b>

## Table of Figures

Figure 1: UXDI Process Model for Traditional Settings	2
Figure 2: Virgin America's Simple UI	4
Figure 3: Example Jira Kanban Board	6
Figure 4: Architecture Diagram	12
Figure 5: Facade Diagram	13
Figure 6: Context Diagram	14
Figure 7: Final ID Cross Reference Tool	15
Figure 8: Overall Burndown Chart by Day	29
Figure 9: Overall Burndown Chart by Week	29
Figure 10: ID Cross Reference Tool Copy and Paste ID Time trials	33
Figure 11: ID Cross Reference Tool File Upload Time Trials	34

## Table of Tables

Table 1: ID Cross Reference Tool Copy and Paste ID Time Trials	33
Table 2: ID Cross Reference Tool File Upload Time Trials	34
Table 3: ID Cross Reference Tool Average Time Trial Results	35

# 1 Introduction

## 1.1 Company Background

Our team completed this project for a Fortune 500 financial investment firm. The company has more than 25 million customers around the world and manages trillions of dollars of customer assets. Customers include individuals who are looking for advice in financial planning and investing, employers who are looking for dynamic benefit programs for employees (retirement, health, and 401K plans), and institutions looking to grow their business to be able to navigate industry changes (<Financial Company> 2021). The company has three major investing sectors: personal investing (PI), workplace investing (WI), and business investing (BI). Our team worked with a software development team in the WI sector that mainly focuses on internal tools for the company's employees.

## 1.2 User Interface

A User Interface (UI) is the space where users interact with an application or software. A well-designed and easy-to-use UI is critical because it creates a better experience for the user and can keep customers interested in a product and company. A well-designed UI is not only important for customers but also for employees using a company's internal system. A good UI is less confusing and will cause less frustration for anyone using it. A good design will also minimize the likelihood of users misinterpreting how to use the tool and making errors, increasing overall user (in our case employee) efficiency.

A strong UI may be modeled off of many different design standards, depending on the purpose and context. In general, a strong UI includes simple methods of interaction to guide the user, avoid potential use issues, and add interest to the user's experience. There are 3 important parts of a User Interface: visual design (making the UI appealing), interactive design (the actions of the UI), and information architecture (the structure and organization of the UI) (Berezhnoi 2019). We will discuss these elements further in Chapter 2.

## 1.3 Project Goals

The goals of this project were to redesign and implement user interfaces for two internal company tools: an ID Cross Reference tool and a refresh cache tool. The ID Cross Reference tool helps employees who need to find different ID types associated with one (or multiple) IDs that they already have. Typically, this tool assists employees with troubleshooting customer issues. The Refresh Cache tool will help employees with refreshing various cloud-based applications which may have several instances running at once.

Our team concentrated on creating simple interactive designs for each UI with similar themes. We ensured that the information architecture and visual design of our UIs for both projects was done intuitively and at a professional level. Our goal for the first project was to design and execute an improved, user friendly UI which was written in Angular. For the second project, our goal was to create another straightforward Angular-based UI for a new Refresh Cache tool. By doing this, the company would be able to apply both of these user interfaces to the applications to make them more user friendly (<Financial Company> Labs 2021).

## 2 Background

### 2.1 Domain Research

Our project followed the traditional model for user experience driven design seen in Figure 1 below. This model consists of two major worlds—the design world and the user world. The three phases in the model are the inspiration phase, the ideation phase, and the implementation phase. Each phase has its own set of steps. In the inspiration phase, developers of a project or product owners identify an opportunity to build or improve a UI. Next, they figure out what customers want from the UI. After the ideation phase, developers design, build, and test the UI to make sure it meets user expectations and user needs. Lastly, in the implementation phase the team deploys the UI to wherever it is needed (Djamasbi et al. 2019).

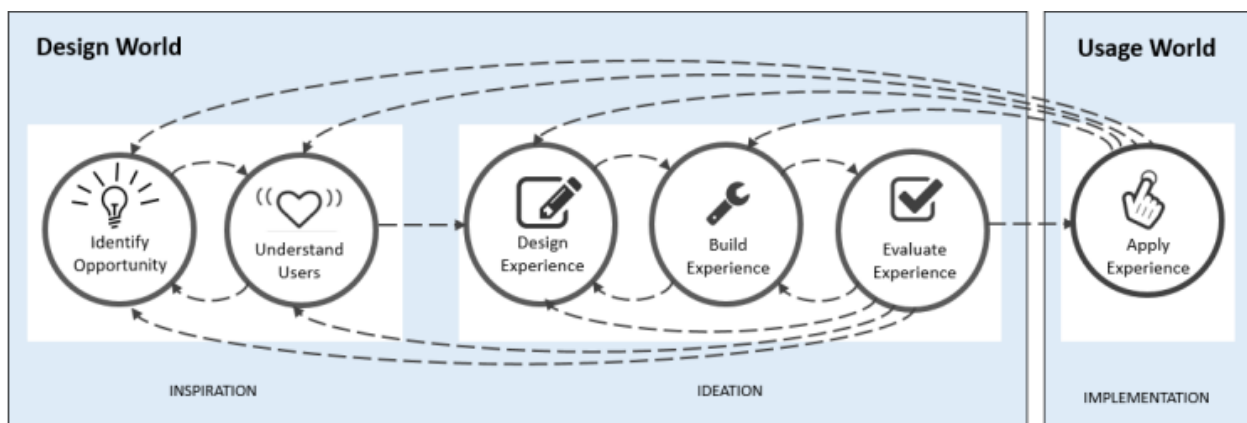


Figure 1: UXDI Process Model for Traditional Settings (Djamasbi et al. 2019)

## 2.2 User Interface Design Rules

To help build our UIs, we followed some best-practice design rules. These rules are important because if they were not followed, the UI could give users a less-than-ideal experience with the tools, and by extension, the company itself. While neither of the UIs we created are customer facing, the rules are still important because if an internal UI is bad it could create operational bottlenecks, angering users (the employees) and resulting in lost time and money. The below rules we referenced in our project came from [tutorialspoint.com](https://www.tutorialspoint.com) (a well-referenced CS info database), which has several of the same rules as [usability.gov](https://www.usability.gov)—a website run by the US General Services Administration ([tutorialspoint.com](https://www.tutorialspoint.com) 2021; [Usability.gov](https://www.usability.gov) (2) 2021).

1. *Consistency*. Any chains of command or terminologies that our team used throughout building the UIs must work the same and function the same way every time. This keeps the UI simple.
2. *Shortcuts*. These are used in order to help users save time when interacting with a UI. Our ID Cross Reference tool's UI allowed users to input multiple IDs at a time unlike the previous version where only one could be manually typed at a time.
3. *Feedback*. Every time an action is performed, the UI should provide feedback that is consistent with how significant the action inputted by the user was. In the Refresh Cache tool's UI, if the user presses the refresh button the UI displays a loading bar as the tool works in the background to refresh the cache. This feedback shows the user what is happening and helps to avoid repeated user actions.
4. *Simple Error Handling*. If a user makes a mistake, a UI should correct or guide them in some way so that the mistake can be fixed. Our ID Cross Reference tool's UI can detect if an ID is entered incorrectly (ex. If there are too many characters entered) and provide a pop-up message to help the user.
5. *Reversal of Actions*. Like the error handling rule, this helps relieve user anxiety and frustration. In our ID Cross Reference UI, if the user makes a typo when inserting an ID there is a little "X" to the right of each ID which helps them easily delete individual IDs.
6. *Control*. In order for a system to be user-friendly it should give users as much control over what they do in the UI as possible. In the ID Cross Reference tool's UI, when a user opens the UI they have many options to manually enter or upload a file to enter data, identify what they are entering, and get help from a tutorial.
7. *Simplicity*. UIs should be kept simple to not frustrate users and keep their time using the UI to a minimum. UIs should not have any unnecessary pages and complex actions should provide the user with guidance. The ID Cross Reference tool's UI is only one page with a few simple actions and a tutorial button if the user needs it ([tutorialspoint.com](https://www.tutorialspoint.com) 2021).



## 2.3 Similar Applications

Although our UIs were made for an internal company system, employees who use them want a good experience and UIs should be made for them as if they were customers of the company. There are some existing applications that take advantage of robust UIs in order to satisfy their customers. First, Virgin America uses its UI design—seen below in Figure 2—to its advantage by making the process of booking flights quick and easy for the user. The company does this with a simple UI that is straightforward, visually appealing, and has a structure that allows the user to retain less information and keep all unnecessary parts of the process out of view until needed. Similarly, Dropbox also has an effective UI for a tough technical based task which is storing and synchronizing files across cloud software. Dropbox makes its UI easy to understand and learn which helps increase the productivity of its customers and makes the process much easier (Beaubain 2018). Both of these UIs also follow many of the rules of building UIs such as feedback, simple error handling, control, and simplicity.

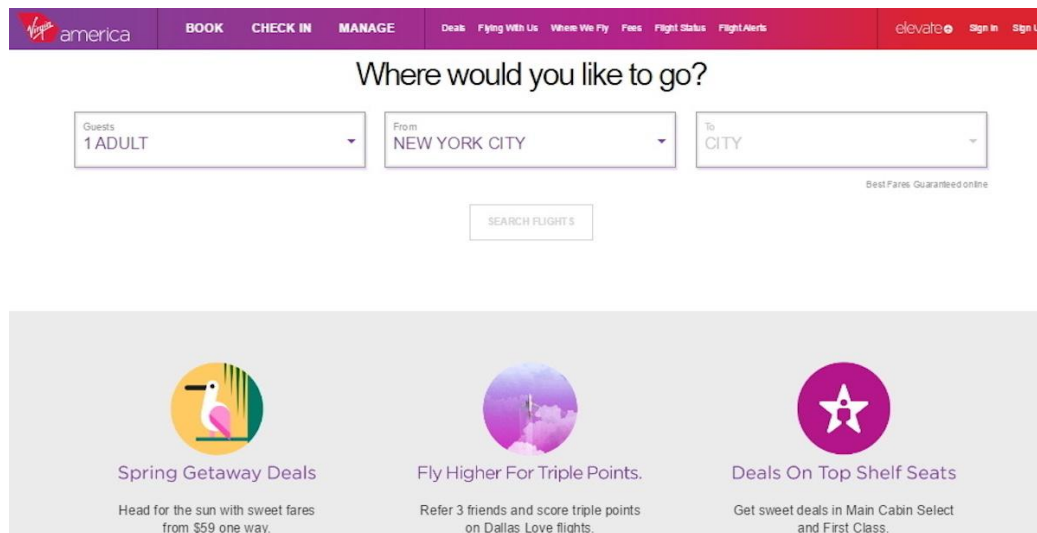


Figure 2: Virgin America's Simple UI (Beaubain 2018)

## 2.4 Purpose and Impact

The aforementioned two companies, along with many others, make their UIs with some common goals of understanding what their users need and want. Even though the UIs we built are for an internal company system, it is still important for them to be held to the same standards as public-facing UIs made for customers. Simple and well-designed UIs save users time and minimize frustration. This is because using poorly-designed UIs can lead to an ineffective use of technology. This can cause poor return on investment on the money and time spent to create the UI and it can also affect the organization's overall goals and outcomes in the short and long term (Djamasbi et al. 2019). Effective UIs provide clear solutions to solving problems and completing tasks while keeping visual design consistent. Our team built on fundamental UI principles when

designing the UI for our two projects, so that our project created UIs which increase efficiency and are easy to use. Our team aimed to develop software to accomplish the goals of both projects in a way that was straight-forward, simple, and easy to understand (Peterson 2019).

## 3 Methodology

### 3.1 Software Development Methodology

Our project utilized the Agile Kanban development methodology, which is a popular methodology used for project management and software development. In brief, this methodology is centered around the creation of a Kanban board and a backlog of stories (Radigan 2021). By prioritizing stories, assigning points, and assigning stories, our team was able to efficiently segment and complete our work in the time given.

Daily standup meetings were a driving force for our work. Every morning, one team member (the Agile Leader) led a meeting where we discussed team updates and tasks for the day, and each team member gave updates answering the following questions:

- What did I do yesterday?
- What am I working on today?
- Am I facing any blockages?

At the end of each week, our team discussed how the week went, our major accomplishments, and our goals for the following week. Additionally, we created a weekly progress report detailing the stories we completed and our progress throughout the week. All of our progress reports and completed stories can be seen in Chapter 7.

To organize our work, we utilized a Kanban board through Jira, an example of which is shown in Figure 3 below. Our board used three swimlanes/columns—to do, in progress, and done—into which we sorted our stories, or cards. After determining the requirements of each tool and making mockup designs, we created stories based on each task we would need to complete. Each story included some identifying information and some details about the task(s): a title, description of task/sub-tasks, acceptance criteria (for determining when each story is complete), and points (approximate time allotted). We created stories in relation to the completion of our report, the completion of business analyses, as well as our technical development.

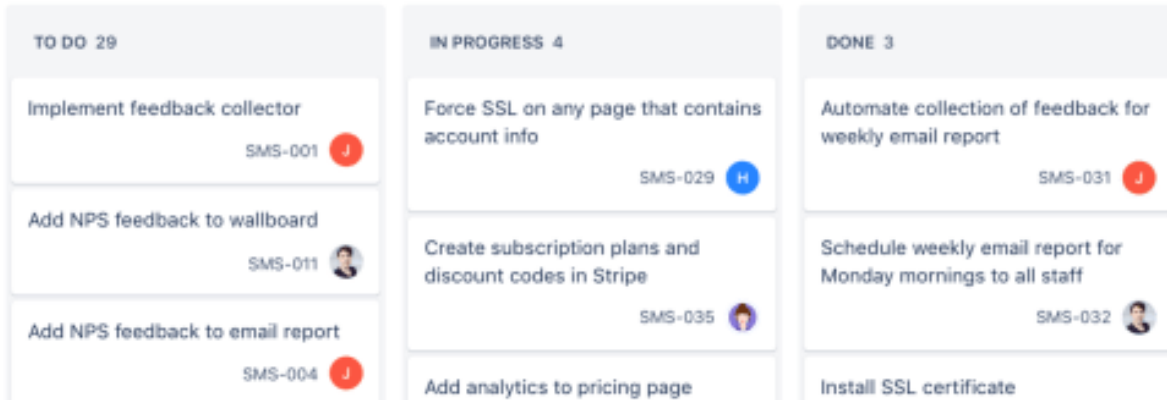


Figure 3: Example Jira Kanban Board (Sa 2018)

Most of our technical development stories were created from user stories, which define the requirements of a product in terms of the functionality required by a user (i.e. As a [User Role] I want [specific functionality] so that [business purpose]). These stories were then broken down into tasks and stories in Jira. A list of our user stories can be found in Section 5.3. After stories are added to Jira, they automatically start on the left side of the board in the “to do” column. Each week, we assigned stories of highest priority and moved them into the “in progress” column as each team member started to work on them. During standup meetings, the Agile Leader moved any completed stories into the “done” column once they were discussed and confirmed done by the group. At the end of each week we had a short discussion about how the week went and our goals for the following week.

Two other important aspects to our stories and Kanban board were points and our Work in Progress (WIP) limit. As previously mentioned, when stories were created they were assigned a certain number of points: 1, 3, or 5. We decided on this system because it is the same one our sponsors use within their software development teams. The number of points corresponding to a story or activity is a measure of the story’s difficulty, or amount of time to complete, with 1 being the easiest/shortest time and 5 being the most difficult/longest time. At the end of each week, the Agile Leader could sum the number of points to track the team’s process and create a burndown chart, which tracks the velocity (number of points completed per week) of the team. Additionally, we set a limit of 10 stories in the “in progress” lane at a time. The purpose of this limit was to ensure that each team member did not take on too many tasks at once, and rather than having several tasks going at once, we were each focused on a maximum of 2-3 tasks at a time. This Kanban practice is commonly known as a WIP Limit (AgileThought 2021).

### 3.2 Agile Scrum Methodology

Although we ended up using Agile Kanban, we had initially planned to use the Agile Scrum methodology. We wanted to use Agile Scrum because it is the most common methodology used by similar MQP teams, but at the beginning of the project we decided that

Kanban would be a better fit because our project was so dynamic. Our sponsors' teams use both Kanban and Scrum, but Scrum is the more popular of the two. The Scrum Methodology centers around short 1-2 week "sprints" of iterative development between versions of a product. This iterative framework allows for plenty of feedback sessions from ancillary team supporters, and helps split large projects up into smaller sections of development (Business News Daily 2021).

The main difference between Kanban and Scrum is that Agile Scrum is focused on the creation of 1-4 week long sprints, where stories are preassigned to both team members and sprints. In our version of Kanban, we did not prescribe to set sprints with formal product reviews at the end. Instead, our team had short review meetings at the end of each week. Instead of assigning every story to our developers at the beginning of sprints, we assigned a few stories and developers were able to "take from the top" of the "to do" column once the assigned stories were completed.

## 4 Software Development Environment

### 4.1 Project Management Software and Programming Environment

Since our project was completed fully online, we used a few different types of project management software. For text and voice communication within our team, we used Discord (0.0.264). We used Microsoft Teams (1.0) through our Virtual Machines to communicate with our sponsors, and Zoom to meet with our sponsors. To compile files and complete our paper, we used Google Drive. For code management, we used BitBucket (7.18)—the same software used by our sponsors. As mentioned in the previous section, we used Jira (8.20) to create a Kanban board and manage our stories.

Throughout our project, we used the Visual Studio Code (1.62) IDE through our Virtual Machines and wrote code in HTML, SCSS, and TypeScript. To create our UIs, we used the Angular (10.2.3) framework. Angular is a web component-based framework built on TypeScript that includes integrated libraries for routing, form management, client-server communications, and more. We used this framework because our sponsor is shifting all of their existing and new applications to Angular. Our team used Angular to develop the individual parts of our UIs to tie them all together into a functioning web application. Within Angular's libraries, we used the material library to style the majority of our components. We also used the following tools to help develop our UIs:

- Restful API: In order to get corresponding IDs when given one ID, we needed to use two APIs. The first one made a request to get an access token. Using this token, a list of valid IDs, and an ID type, the second API retrieves the data that we need to display to the user. If any request returns an error, a message is displayed to the user.
- JS-XLSX Library: We used the xlsx library to allow users to interact with data upload and download using Microsoft Excel files. We felt this feature was important because users may want to save their results from the ID Cross Reference tool so that they do not

have to input the same information multiple times. This feature also allows users to send results to other users without having to reuse the tool. Excel also has additional data organization and analysis features to allow users to shape ID data to their desired format.

## 4.2 Data Sources

For the ID Cross Reference project, the necessary data comes from the two APIs mentioned in the previous section. The first important piece of data is an access token which gets passed as a parameter to run the second API request. The second API retrieves the data of interest which is immediately displayed to the user. This information does not get stored anywhere on the page because the data displayed changes often.

# 5 Software Requirements

## 5.1 Requirements Gathering

The main sources of requirements for our team's UIs were our sponsors, who we met with consistently throughout the project term. We gathered and documented the majority of the requirements for the ID Cross Reference tool in the first week of the project during these sponsor meetings. As we progressed through the software development process, we asked for feedback. At the end of Week 3, we conducted informal interviews with a couple of business analysts in the WI sector. They are also the primary users of the ID Cross Reference tool so their input was very useful in both the development of the UI and understanding why the project is beneficial for the business overall. For the Refresh Cache tool, our supervisor reached out to the users of the Refresh Cache tool in order to meet with us and give us similar requirements and advice that we got for the ID Reference Tool. After learning about the tool from our supervisor, we presented her with our designs to get feedback.

## 5.2 Functional and Nonfunctional Requirements

Based on previous understanding of the projects assigned to us, we created functional and nonfunctional requirements. For the first project, our team improved the tool's existing UI. Overall, we focused on improving the look of the UI while preserving the functionality and keeping the design simple. The functional requirements of the ID Cross Reference tool were:

1. Have the ability to export a list of IDs into a .csv or .xlsx file
2. Let a user delete a single ID at a time from the entire list
3. Display all of the IDs before submitting them
4. Show the correct information related to each ID in a table when submitted
5. Allow the user to search through more than 10 IDs at a time

6. Allow users to insert IDs in more ways than just plain text: CSV/Excel upload
7. Keep the tool's batch processing ability
8. Combine single and batch entry options onto the same page (rather than two tabs).

Our first project also had a couple of non-functional requirements for the UI which were:

1. Keep the plain layout of the tool while improving the way it looks: get positive feedback from current users on the design
2. Add more verbiage to the UI to avoid user error: an error message if an ID is not inputted correctly, a button to display a tutorial

Our team also created requirements for the second tool that was assigned to us. Since this Refresh Cache tool did not have an existing UI, the requirements were more basic than the first project. The purpose of the tool is simple in that it helps internal company software developers refresh applications quickly and easily. The functional requirements for the Refresh Cache UI were:

1. There should be a dropdown list of application identifiers that a user can choose from
2. When an application is selected from the list, lists of its environments and endpoints are displayed for the user to choose from
3. There should be a second screen that allows selected users to add, update, or delete endpoints for a selected application
4. After selecting an application and endpoint, a button will initiate the refresh and give the user immediate feedback in the form of a message

Our second project did not have any non-functional requirements.

### 5.3 Epics and User Stories

As discussed in Section 3.1, our team created a set of user stories before creating stories in Jira. We created these user stories based on informal interviews with our sponsors and the requirements listed in the previous section. We focused on creating user stories from two perspectives: that of a user/employee, and that of a developer. Since we had two different tools to work on, we created user stories in two different epics, one for the first tool and one for the second tool. Our initial user stories are listed below.

#### ***Epic 1: Update UI for ID Cross Reference tool***

- a. As an user (Business Analyst for financial company)
  - i. I want to search multiple IDs for corresponding IDs
  - ii. I want clear error messages if I enter invalid data
  - iii. I want to be able to differentiate between which ID I input so that I accurately enter data

- iv. I want to be able to paste in multiple IDs at once so that I do not need to spend time entering in each one
- v. I want to be able to import a .csv or .xlsx file of IDs so that I can save time
- vi. I want to be able to export the data that I get to a .csv or .xlsx file so I can save it somewhere
- vii. I want to be able to individually delete one ID from the input if I make a typo
- viii. I want to be able to input as many IDs as I want so that all of my information is in one place
- ix. I want the option to view a tutorial on the tool on the same page as the tool
- b. As a developer
  - i. I want to have clear API calls to make the code as readable as possible

***Epic 2: Create UI for Refresh Cache***

- a. As a user (Software Developer for financial company)
  - i. I want to see a list of all running applications that I have the ability to refresh
  - ii. I want the ability to select an application and endpoint to refresh from available lists
  - iii. I want to be able to edit (add, update, delete) the endpoints of any given application (if I am given the permissions)
  - iv. I want to be able to click a refresh button to refresh the cache.
- b. I want visual feedback from the Refresh Cache tool so I know that it is working

## 6 Design

As we discussed in Section 2.1, our work followed a typical UXDI Process Model. After our sponsors identified the ID Cross Reference tool and the Refresh Cache tool as projects, our team started with identifying areas for improvements and learning about the purposes of the tools. This chapter details our initial thoughts on the design of each UI, as well as the architectural diagrams for the ID Cross Reference tool.

### 6.1 ID Cross Reference Tool

#### 6.1.1 Current Model and Proposed Redesign

The ID Cross Reference tool we worked on had a very simple UI. The tool itself is an internal tool accessible by employees through a web browser, and the original front end code had been lost throughout the years. One of the company's main motivations for updating this tool was to bring the UI onto the Angular framework, so the code could be archived and the tool would be on the same framework as most other internal applications. Aside from this reason, the

UI itself was quite simplified and did not utilize the company's standard color scheme. Overall, our team's main design critiques were:

- Different ID entry types (single vs. batch/upload) were on two separate pages
- Color scheme and general design did not reflect the company's visual identity
- ID entry box was inefficient and not user friendly
  - Users had to enter IDs one by one in a large text box, and separate each ID with a comma
- There was no option to export the results into an Excel or CSV file
- There was no tutorial or help button to provide users with additional help and information about the tool

Along with our list of functional and non-functional requirements, we used our design critiques to draft a mockup design of a new UI for the tool which we presented to our sponsors. Our new UI combined the existing two pages into one page. It also implemented the use of a switch to identify the type of ID that is input into the tool. Instead of a large text box of IDs, the new UI creates a list of IDs from user entries before submitting them all at once. By doing this, the user is able to more easily identify any errors before submitting a list of IDs—instead of looking at a block of text in a rectangular box, they see a scrollable list with each ID in a bubble. We also added a clickable info button at the top of the tool to provide tutorial information, and a button to give the user the option of exporting the output into an Excel file or a CSV. Finally, we referenced the financial company's branding guidelines to implement an appropriate color scheme.



## 6.1.2 Architectural Design

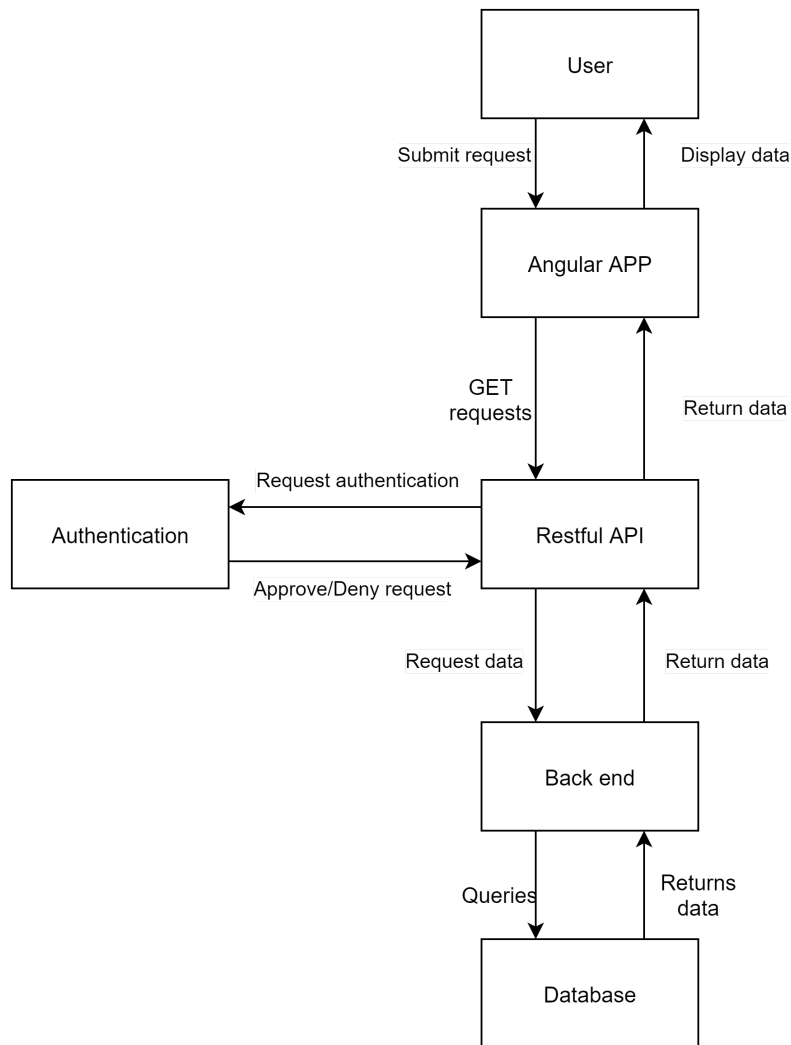


Figure 4: Architecture Diagram

The above diagram shows how data/information is passed between different parts of the ID Cross Reference tool. The first step in using the tool is the user entering their input (IDs) and requesting the associated information. The application then compiles and formats the user inputs and sends a request to the Restful API to get the information associated with the given IDs. The Restful API checks to make sure that the user is authorized to use the Cross Reference tool, and if this authentication is approved the API will then make a call to a function in the back end. The back end function queries the database for the information and sends the database's response to the application through the Restful API. The application then parses the response data and displays it to the user.

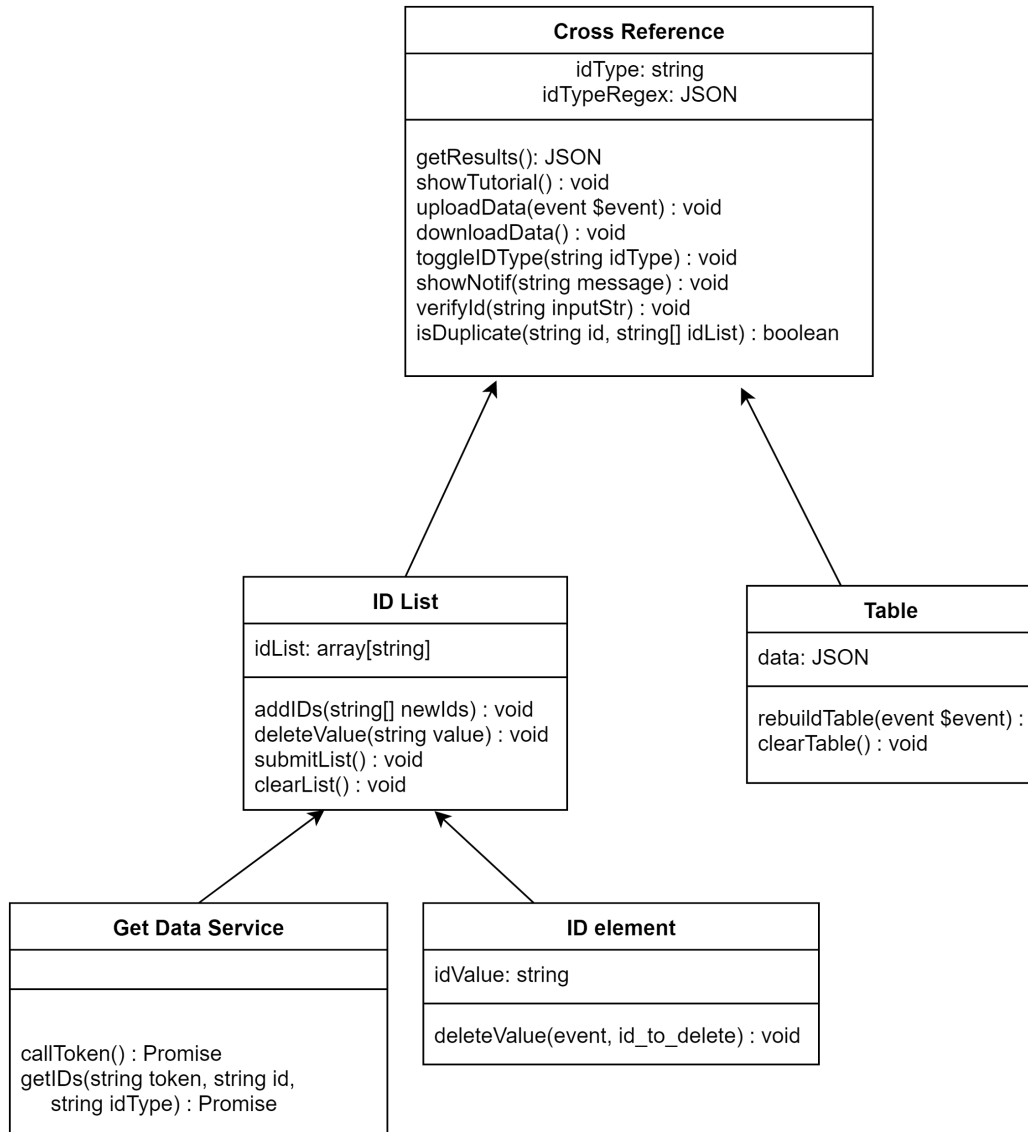


Figure 5: Facade Diagram

The above Facade design pattern diagram shows how our app utilizes components and services to accomplish its goals. The main class has a number of very simple elements that did not justify creating a new component. These include an info button, upload/download buttons, `idType` toggle, and ID input field. The main component interacts with the ID List which handles saving data and submitting/clearing the IDs. This component interacts with a service that upon clicking the submit button, calls two APIs to get the necessary data. The main component ties everything together as it passes the API results to the table which updates.

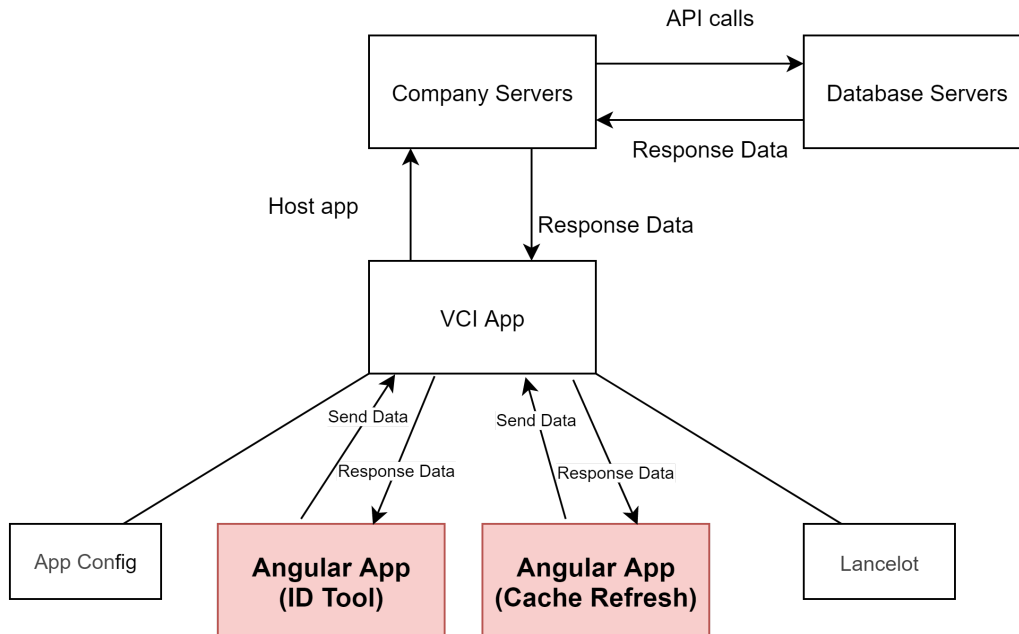


Figure 6: Context Diagram

The context diagram above in Figure 6 shows how the ID Cross Reference tool interacts with its system environment. The ID Cross Reference tool is part of a larger Voice Configure Interface (VCI) Application that also contains the App Config, Lancelot, and Cache Refresh tools. All outgoing requests for data and responses of incoming data go through the VCI application. This is the method through which the Cross Reference tool accesses database and authentication information.

### 6.1.3 Final Design

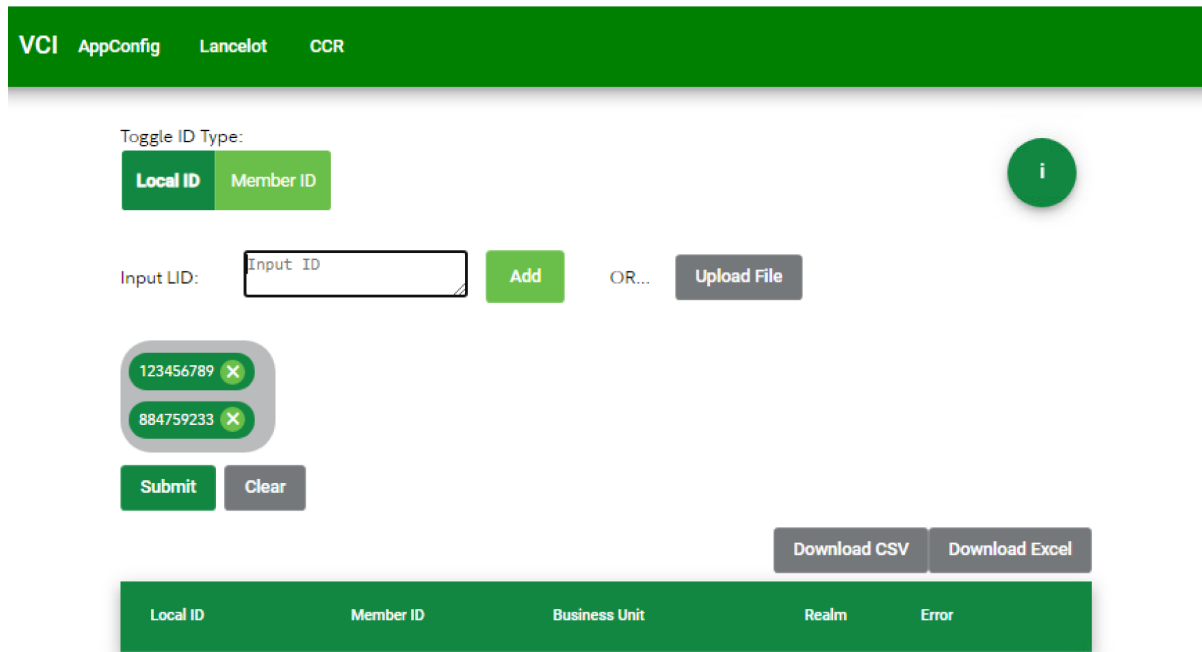


Figure 7: Final ID Cross Reference Tool

Figure 7, above, shows our final developed design for the ID Cross Reference tool. It includes the elements discussed in 6.1.1– a switch for ID type, all of the features on one page, a clear list of IDs prior to submission, export buttons, and a tutorial button. The design also includes colors standard to the visual identity of the company. After clicking the submit button, the corresponding information appears in the table at the bottom of the screen, underneath the correct headers.

### 6.2 Refresh Cache Tool

The Refresh Cache tool was in development and without any existing UI when it was presented to our team, so we had a lot of creative freedom when it came to the initial designs. Our sponsor tasked us with creating a simple and user-friendly UI which fit the function of the tool and included some specific features: a drop down list of applications and environments, a list of endpoints, instant visual feedback after the user chooses what to refresh, and a separate page for editing endpoints. When creating our mockup designs for this UI, we focused on implementing these requirements first, then played around with the visual layout, shapes, and icons on the page. We also modeled this UI with similar design elements to the ID Cross Reference UI since both tools will be part of the same website.

Since we were ultimately unable to begin any development of the Refresh Cache tool, we did not create any architectural design models for the tool.

## 7 Software Development

As we have mentioned, we utilized a Kanban Methodology for this project. Although our work was not structured around sprints (like the Agile Scrum methodology utilizes), we held an informal retrospective at the end of each week and wrote weekly progress updates. By documenting and discussing our work each week, our team was able to assess our progress, create goals for the following week, and handle problems as they arose.

Throughout our eight weeks, we spent the first two on initial setup of our development environments and designing the first tool's UI. Weeks 3 through 7 were focused on the development of the first tool, while Weeks 6 through 7 were focused on the requirements gathering and designing the mock-up of the second tool.

### 7.1 Week 1 (10/25-10/29)

Our team had a slow start to our project. Throughout the first two weeks we faced many issues with gaining access to company systems, which prevented us from doing most of the work we had planned on completing during this period. In Week 1, we obtained access to our Virtual Machines, learned more about the ID Cross Reference tool, and had some initial meetings with our sponsors. We also completed some onboarding training modules, and learned how the company's teams implement Agile into their workflows. We discussed our team's use of Agile, and decided on a general meeting schedule. While we were not able to code yet, we created an initial mockup of the ID Cross Reference tool as well as some user stories, so we could get a better picture of how we would need to modify the existing UI. Since we did not get access to Jira this week, we were unable to complete the final two stories listed below. In Week 2, we hoped to gain access to Git to begin coding and access to our sponsors' Jira board so we could create detailed stories.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 22. Our overall burndown charts can be found in Section 7.8.

There was one potential risk the team encountered that could have had a moderate impact on the project this week. Our team lost the first few days of potential development due to the team having to gain access to the VMs and other softwares. As a result, we could only make a mock-up and start making a couple of user stories. Due to this setback, our work on the first tool was extended and we were not sure if we would have time to get to the third project that had been presented to us by our sponsor. Ways to mitigate the risks were mostly out of our control since it was up to the company to get us access to the necessary softwares. Although if the team had minimal access problems down the road we could work a little harder on the code and paper in order for us to still be able to do the third project.

One of the things we could have done better this week was be more prompt with getting access to everything we needed so that way we could have been completely prepared to start

developing the ID Cross Reference tool in Week 2. Following up on our access issues faster might have sped up the process overall.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Get access to virtual machines	- Call IT support	5	All	Yes	10/26
Get more details about Tool 1	- Meet with sponsors - Demo ID referencing tool - Get use cases and requirements	1	All	Yes	10/26
Download/access necessary software	- Bitbucket, VScode, postman	3	All	Yes	10/26
Create general user stories for Tool 1	- Review use cases - Create Epic 1	3	All	Yes	10/27
Complete onboarding trainings	- 6 online trainings through our VMs	3	All	Yes	10/28
Create mockup of planned UI for Tool 1	- Brainstorm and research design ideas - Review requirements	3	All	Yes	10/28
Create our board on Jira	- Access to Jira	1	All	No	(11/01)
Add user stories to Jira	- Decide on points system - Decide on Kanban vs. Scrum	3	All	No	(11/01)

## 7.2 Week 2 (11/1-11/5)

Through Week 2, we were still experiencing access issues, although we did gain access to our own Jira board and had our initial design approved by our sponsors as well as the first tool's main architect. We also gathered some additional UI requirements for the tool, and spent time updating content in our report. We updated our UI design and showed it to our sponsors, and created an Architecture design and Facade diagram. Additionally, we planned out daily standup meetings and discussed Scrum roles. In Week 3, our team hoped to begin team stand ups every morning, gain access to the code repository, and revise our stories in Jira to be more efficient.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 21. Our overall burndown charts can be found in Section 7.8.

This week had a moderate-high risk of impacting the whole project since our first project got off to a much slower start than expected and our team had more access problems this week. There was a likely chance that it could change the project scope in that our team would be unable to begin the third optional project described by our sponsors. Although our team completed all the user stories this week, our access issues had the potential to slow our team down on the project even more. The only mitigation risks we could do and did was to email our sponsors immediately if there was an access problem and then wait to get the problem resolved. The rest of it was up to the company and was out of our team's control.

Similar to Week 1, our team could have been more proactive in getting access to all of the necessary programs so we could have started developing the ID Reference tool this week. Another thing that we could have done better this week is started scheduling daily standup meetings with our sponsors to help speed up resolutions to our access issues.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Create our board on Jira	- Access to Jira - Add preliminary user stories	3	All	Yes	11/02
Get mockup approved	- Approval from direct supervisors - Approval from architect	3	All	Yes	11/02
Revise mockup design		1	All	Yes	11/04
Paper - Create and format sections for the paper		1	Caitlin	Yes	11/04
Paper - update introduction		3	Callan	Yes	11/04
Paper - update methodology		3	Caitlin	Yes	11/04
Make to do list for next week		1	Caitlin	Yes	11/04
Choose Scrum leader		1	All	Yes	11/04
Plan standup		1	Caitlin	Yes	11/05

meetings					
Paper - create diagrams	- Architecture Design - Facade Diagram	1	Chris, Ido	Yes	11/05
Get access to Bitbucket		3	All	No	(11/08)

### 7.3 Week 3 (11/8-11/12)

In Week 3, we were able to resolve the majority of our access issues. By Thursday, we all had access to the code repository, and we were able to start coding some basic UI elements including the ID Cross Reference tool's landing page. We started having daily standup meetings every morning, and updated our story system in Jira to include points and some story assignments. On Friday, we met with two frequent users of the ID Cross Reference tool, who walked us through their usage of the tool and looked at our mockup design. We talked about the big picture purpose of the tool, what features they felt were missing, and some benefits of the tool and our project as a whole. We also created our first status report covering the first three weeks of work. In Week 4, we hoped to finish coding a prototype of the new UI so we could do some testing and get additional feedback from our sponsors. We were also hoping to begin reaching out into the business aspects of the tool and the project in general, and work on fleshing out more of our final report.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 33. Our overall burndown charts can be found in Section 7.8.

This week's risks were low since our team finally resolved the vast majority of the access issues. However, access issues could still come up at any time and could have an impact on how many projects we were able to get done. The more of a project the team gets done the less risk an access problem is to impact the scope of the first project and the other two.

One of the things our team could have done better this week was reach more into the business aspects of the tools and start to get data and metrics for the ID Cross Reference tool. Our team also could have done more to look into the reasons why we faced so many access issues to prevent them from happening again.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Get access to repo	- Chris and Ido get access - Caitlin and Callan get access	5	All	Yes	11/08



Paper - add SDLC section	- SDLC research	3	Caitlin	Yes	11/08
Paper - revise research section		5	Callan	Yes	11/09
Create landing page for the ID referencing tool		3	Chris, Ido	Yes	11/09
Update Jira board	- Start assigning stories - Add points to stories - Add paper-related stories - Breakup larger stories	3	All	Yes	11/09
Paper - update citations and development environment	- Change citation style - Update list of development environments	1	Caitlin	Yes	11/09
Decide on Kanban vs. Scrum		1	All	Yes	11/09
Set up meeting with users	- Communicate with Scott	1	Caitlin	Yes	11/10
Complete and send forms		1	All	Yes	11/12
Meet with users	- Demo ID referencing tool - Show our design - Q&A	1	All	Yes	11/12
Create status report		1	All	Yes	11/12
Put Paper into past tense		1	Callan	Yes	11/12
Paper - Clean up Citations (1)		3	Callan	Yes	11/12
Create tutorial button	- Make popup component - Make tutorial button	1	Chris	Yes	11/12
Create ID input section	- Make text field with functioning "Add" button	3	Ido	Yes	11/12

## 7.4 Week 4 (11/15-11/19)

This week we ran into some access issues again when we lost access to Git, but were still able to code in our existing branches throughout the week. Once we had access again, our developers pushed their code and successfully merged the branches with only a few errors to correct. We made a lot of changes and updates to almost every chapter of the paper throughout the week. Our developers were able to create all of the necessary features of the ID Cross Reference tool's UI. On Friday, we started tying features together and working on the styling and visual design of the page. Next week before our Thanksgiving break, we hoped to meet with one of our sponsors to get another demo of the Refresh Cache tool. After doing this, we would be able to confidently create our user stories and can start adding those to our Jira board.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 55. Our overall burndown charts can be found in Section 7.8.

This week's risks were high. Our team ended up having trouble with code access again and it took the company a few days to fix it. Like in previous weeks, this impacted the project because our team most likely would have to remove one of the projects from the whole MQP scope. Although the access issues were fixed, another issue could still come up at any point and cause more delays and may continue to affect the scope of the project. For the project removal risk, our team asked our sponsors which projects we should prioritize. This prioritizing could also reduce risk to the team since our team would be able to complete the most important two projects and not have to rush to complete a third.

One of the things our team could have done better this week is obtain background information and start doing other pre-development tasks for the Refresh Cache tool such as making user stories or drawing up mock-ups in order to get ahead on the project. We also had some trouble with prioritizing the projects throughout the week since our sponsor for a potential third project suddenly wanted us to get their project done as soon as possible which interrupted our progress.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Create Input field		3	Ido	Yes	11/15
Create popup for tutorial		3	Chris	Yes	11/15
Create switch to change ID type	- Make toggle switch component	1	Chris	Yes	11/16
Get currently selected input type		3	Chris	Yes	11/16
Paper - revise Chapter 3		3	Caitlin	Yes	11/16

Create ID list		3	Ido	Yes	11/16
Create ID element		3	Ido	Yes	11/16
Paper - write Chapter 5		5	Callan	Yes	11/16
Paper - write sections 8.1, 8.2, 8.3		5	Callan	Yes	11/16
Ability to delete IDs		1	Ido	Yes	11/17
Create a table for IDs to show up in		1	Chris	Yes	11/17
Paper - revise Chapter 7		1	Caitlin	Yes	11/17
Paper - edits & formatting	- Edit Chapter 5 - Edit bibliography - Add numbers to chapter headers	3	Caitlin	Yes	11/17
Paper - revise figures 1 and 2		1	Chris, Ido	Yes	11/17
Paper - create context diagram		1	Chris, Ido	Yes	11/17
Paper - update the abstract		1	Caitlin	Yes	11/18
Paper - draft section 6.1		1	Caitlin	Yes	11/18
Paper - update Chapter 4		1	Caitlin	Yes	11/18
Merge all code		3	Chris, Ido	Yes	11/18
Paper - write section 8.4		1	Callan	Yes	11/18
Paper - add new citations for business section		1	Callan	Yes	11/18
Button styling	Implement angular material	3	Ido	Yes	11/18
Paper - update section 4.1	(secondary updates post-advisor meeting)	1	Caitlin	Yes	11/19
Develop burndown chart		3	Ido	Yes	11/19
Paper - add risk paragraphs (Chapter 7)		1	Callan	Yes	11/19

Parse ID input and add separately to id list		1	Chris	Yes	11/19
Prepare status report		1	Caitlin	Yes	11/19

## 7.5 Week 5 (11/22-11/24)

This week was very short (2 days) because of the Thanksgiving holiday. We worked on getting in contact with future users of the Refresh Cache tool but had not received a reply yet. We had a brief access issue on Monday which was resolved in the middle of our standup meeting that morning. Our developers completed all of the error input user stories and worked on some of the styling for the ID reference tool's UI. We also inserted an Excel import option into the ID reference tool and the tutorial button and began working on exporting to .csv and .xlsx files.

After getting feedback regarding confidentiality within our paper, we modified our text to remove direct mentions of our sponsoring company and revised a few different sections. Our team also reached out to a software development team leader to ask for information regarding the standard rate of labor. This was done in order to develop financial metrics for a new section in Chapter 8.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 28. Our overall burndown charts can be found in Section 7.8.

Next week we hoped to finish the ID Cross Reference tool and then move on to the Refresh Cache tool. In order to start, our team hoped to meet with users to obtain background on it so our team could write up user stories. We also hoped to obtain some financial metrics for a business analysis.

This week's risks were high. First, our team lost access to the Git repo once again although this time it was fixed within the hour and had no impact on the project. Another major risk that affected the whole project scope is that if the third project is not started before the paper's deadline then the third project will not be able to be completed. To mitigate the access risk, our sponsors worked on finding the root cause of the problem so the chances of more access problems occurring would decrease. To mitigate this week's scope risk our sponsors made it clear to the team guiding us on the testing project that the ID Cross Reference and Refresh Cache tools must be done before the testing project. To mitigate the project scope risk, our team and sponsors clearly defined in our project scope that the testing project is optional so if it cannot be included it will have no effect on the entire project.

One thing our team could have done better this week is look into what was causing our access issues to the code repository to reduce future problems. Our team also could have started getting business data for both the ID Reference tool and Refresh Cache earlier so our team could have had that information sooner.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Input error checking		3	Chris	Yes	11/22
Error/warning messages notification box		3	Chris	Yes	11/22
Style ID chips		3	Ido	Yes	11/22
Update and annotate mockup		1	Caitlin	Yes	11/22
Paper - fix company mentions (pt. 1)		3	Callan	Yes	11/22
Paper - revise 3.1		3	Caitlin	Yes	11/22
Paper - fix company mentions (pt. 2)		3	Caitlin	Yes	11/23
Paper - draft 6.1		3	Caitlin	Yes	11/23
Prepare status report		1	Callan	Yes	11/23
Import csv feature		5	Ido	Yes	11/23

## 7.6 Week 6 (11/29-12/3)

This week our team accomplished a lot. Our developers focused on getting the API endpoints for the ID Cross Reference tool and successfully getting a response. We also did a lot of styling and bug fixing. We got the functional and non-functional requirements for the Refresh Cache tool, wrote some user stories, and developed a mock-up for the Refresh Cache tool.

Our team also did a lot on the paper. We acquired the standard rate of labor from our sponsor and received time trial data from a user of the ID Cross Reference tool. However, before we could do any analysis we had to finalize our ID Cross Reference tool's UI and run time trials on it. In addition to that our team edited the whole paper and submitted our final draft on Thursday. Our team also edited the paper based on comments received from our advisor and submitted a first draft to him for a closer look.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 71. Our overall burndown charts can be found in Section 7.8.

This week's risks were high. First, our team had trouble setting up the back end API which made it impossible for our team to test the ID Cross Reference tool and do any time trials

on it. Another risk that has affected the project scope was the extended amount of time we had to spend on finishing up the ID Cross Reference tool's UI, which cut our available time to work on the Refresh Cache UI and eliminated the possibility of working on the third project. In order to mitigate these risks, our team reached out to the sponsor immediately after we started having trouble setting up the ID tool's API. Our team also wrote a description of our existing data in Chapter 8.5 in case we were not about to set up the API in time to run a second set of time trials. Our team already took steps to mitigate the project scope risk by prioritizing all three projects back in Week 5 and in addition to that our team also set priorities for things that should be done for the Refresh Cache tool project. Since we needed to adjust our Refresh Cache design,

One thing our team could have done better was looking into business data earlier in the MQP so we could have had more time to find specific metrics for the ID Cross Reference and Refresh Cache tools.

Next week our first goal was to finish all development. This includes finishing the ID Cross Reference tool and getting as much as we can in the Refresh Cache tool. For the business portion, our goal was to get time data from the updated ID Cross Reference tool and then insert specific metrics for the ID Cross Reference tool. Once our advisor and sponsor got back to us on the paper, we wanted to update the paper based on all the feedback we received. Our last goal was to complete our presentation and start rehearsing for it.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Get more info about Tool 2	- Meet with sponsor - Get use cases	1	All	Yes	11/29
Email Cross Reference users		1	Caitlin, Callan	Yes	11/29
Paper - Chapter 5 and 7 review/edit		3	Caitlin	Yes	11/29
Paper - Update 5.2	- Review requirements for Refresh Cache tool - Create Epic 2	1	Caitlin, Callan	Yes	11/29
Style ID list		3	Ido	Yes	11/29
Style ID element		3	Ido	Yes	11/29
Implement API		5	Ido	No	
Style Table		3	Ido	Yes	11/29
Create User Stories for Refresh Cache	- Add to paper - Update Jira board	3	All	Yes	11/30

Paper - Finish Chapter 5	- Complete 5-1 - Complete 5-3	3	Callan		11/30
Create mockup for Refresh Cache tool		3	All	Yes	11/30
Paper - Outline Chapter 8, 9, 10		3	All	Yes	12/1
Paper - Write Chapter 10		3	Caitlin	Yes	12/1
Paper - Finish Writing Chapter 8	- Write 8.5 - Add to 8.6 - Add to 8.7	5	Callan	Yes	12/1
Paper - Write Chapter 9	- Write 9 - Write 9.1	5	Callan	Yes	12/1
Paper - Full edit and review		5	Caitlin	Yes	12/2
Write Chapter 11		5	Callan	Yes	12/2
Revise Chapter 8	-Write ID Reference Tool Data Paragraph - Add intro paragraph to 8.6 - Add intro paragraph to 8.7	3	Callan	Yes	12/2
UI bug fixes		5	Ido/Chris	Yes	12/2
Add the "Should have done sections"	- Add them all to 7.1-7.5 - Add a summary to 9.1	3	Callan	Yes	12/2
Outline final presentation		1	Caitlin	Yes	12/3
Prepare status report		3	Callan	Yes	12/3
Get Refresh Cache mockup approved by supervisor		1	All	No	12/10

## 7.7 Week 7 (12/6-12/10)

This week our team accomplished a lot and experienced a major project scope change. Our developers focused on finishing off the ID Cross Reference tool by doing all the final debugging, writing the tutorial, and by testing it with time trials to compare it to the old data. Since we knew we wouldn't have time to start and complete development on the Refresh Cache tool, we decided to just complete a final version of our mockup design for future developers to reference.

Our team also worked a lot on the paper and final presentation to the company. We were able to get the data analysis done for the ID Cross Reference tool metrics, correct feedback from both our advisor and sponsor, correct final grammatical and formatting errors in the paper, and send our final draft to our sponsor. In addition to the paper, our team also completed the bulk of our final presentation's content.

Our Work in Process (WIP) limit was 10, as we typically each took 1-3 stories per day. The total number of points we completed this week was 70. Our overall burndown charts can be found in Section 7.8.

This week our team's risks were high. The main risk our team had was that we needed more time to complete the ID Cross Reference tool, which ultimately changed our project scope. The team mitigated these risks by asking our sponsor what our team should prioritize and adjusting our project scope again. We were also approved for an extra day of development to complete our work. Our team planned to work over the weekend to get the ID Cross Reference tool completely done and complete the Refresh Cache mockup.

One thing our team could have done better is our team could have tried to figure out how long it would take to complete both projects so that our project scopes would have not changed on the spot.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Outline presentation slides		1	Callan	Yes	12/6
Rough presentation draft (Callan's Part)		5	Callan	Yes	12/6
Fixed styling for ID input area		1	Ido	Yes	12/6
Complete API integrations		5	Ido	Yes	12/7
Update advisor feedback		3	All	Yes	12/7
Format presentation		3	Callan	Yes	12/7



(Callan's Part)					
Presentation slide creation (Caitlin's Part)		5	Caitlin	Yes	12/7
Paper corrections		5	Caitlin	Yes	12/7
Get CCR metrics time trial data		1	All	Yes	12/8
Data analysis and input for CCR Metrics	- Calculate time and money saved - Insert information into paper - Insert information into presentation	3	Callan	Yes	12/8
Transform API response data to fit table		3	Ido/Chris	Yes	12/8
Fix ID verifications		5	Ido/Chris	Yes	12/8
Add 'Error' column and logic to table		5	Ido/Chris	Yes	12/8
Fix Business Unit render		3	Chris	Yes	12/9
Fix data analysis		1	Callan	Yes	12/9
Adjust chapter 9 and 11 based on project scope change		1	Callan	Yes	12/9
Write tutorial		1	Ido	Yes	12/9
General presentation editing		3	Caitlin	Yes	12/9
General paper editing		5	Caitlin	Yes	12/10
Status Report		3	Callan	Yes	12/10
Presentation Rehearsals		3	All	Yes	12/10

## 7.8 Overall Burndown Charts

The burndown charts below in Figures 8 and 9 display our story completion progress throughout the seven weeks of our project. Figure 8 displays this data by day, and Figure 9 displays the data by week. In addition to weekends, note that November 24-26 were non-working days due to the Thanksgiving holiday.

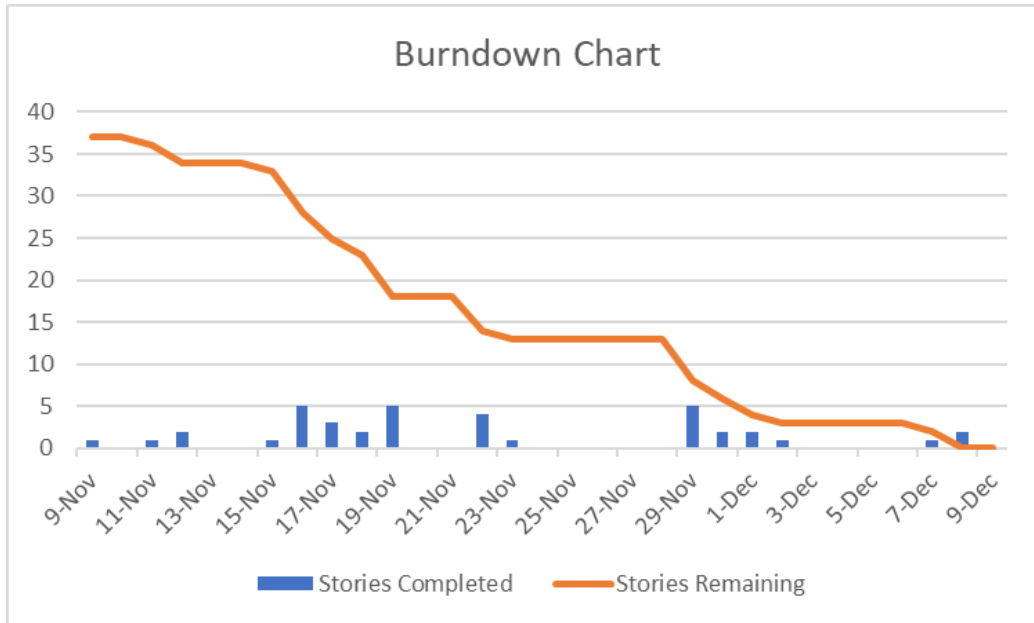


Figure 8: Overall Burndown Chart by Day

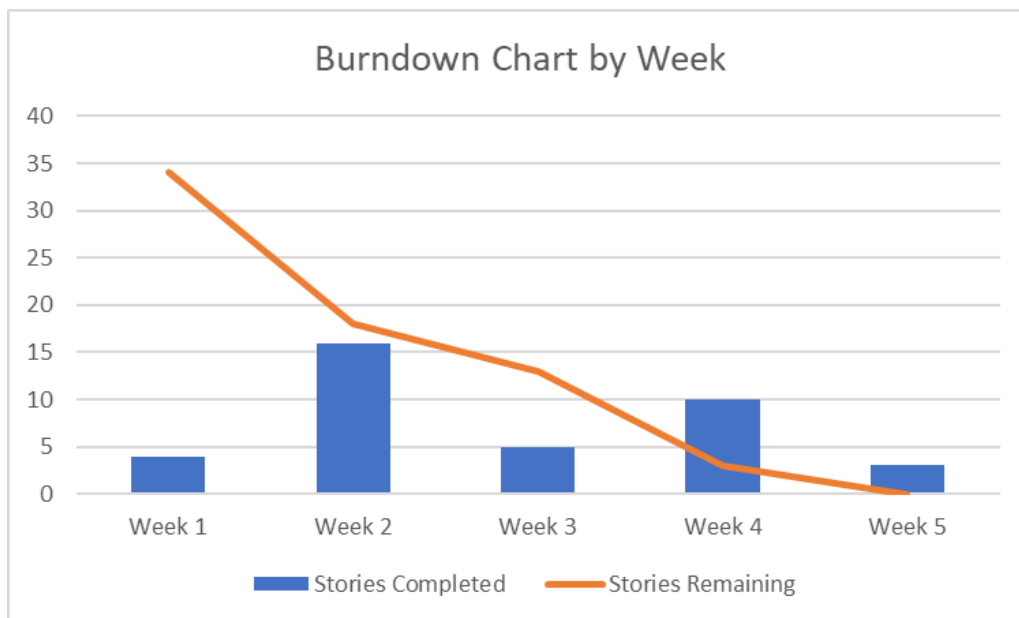


Figure 9: Overall Burndown Chart by Week

## 8 Business Perspective

Separate from our group's technical work, a different aspect of this project is the business side of things. Doing a project like this comes with different risks if the project goes wrong but also different rewards if the project goes well. The projects we worked on for the company included risks and rewards from multiple different sides of the business realm such as operational, innovative, financial, and reputational.

### 8.1 Primary Business Risks

There were two main operational risks associated with our project: the risk of deployment and the risk it has to any customers involved with it. If the deployment of a UI goes wrong, it could disrupt the company's operations and potentially cause the company to lose control of its internal systems. These failures would cost the company a significant amount of money and time to fix affected systems. These operational risks also affect the company's customers because a disruption in service could cause customers to wait for a long time or not get serviced at all. This would cause them to get frustrated and could cause the company to lose customers which would only worsen any financial losses (Morgan 2021).

Innovative risks of deployment and customer involvement are closely attached to one another. If the UI that is deployed fails to meet the expectations of users or fails to effectively address already known user needs then it has a direct impact on the users. If a UI deploys poorly it would have a direct impact on both the company's employees and could have an indirect but serious effect on customers. If employees struggle to use a tool they will waste time on a process that is more difficult than before. As employees' time is now wasted, they would be unable to spend time resolving customer issues, leading to customers waiting longer and becoming frustrated. Similar to operational risks, it could result in a loss of customers and money for the company. More money as well as time would be lost trying to build a new system to replace a poor performing one (Wood 2021).

Reputational risks are the biggest risk category out of all of them due to the company's size and current reputation. If a deployment went wrong not only could it affect the employees using the tools and the customers who need the employees, but the whole company. Frustrated employees and customers could lead to a bad reputation which could significantly drop the number of new customers coming into the company. A bad deployment and operational issues could cost the company hundreds of millions of dollars in customer assets if not more. They could also cost the company's employees because they would not want to work in a company that deployed a bad product and has a bad reputation. Acquiring new talent would also be harder (ScienceDirect 2021). Recovering from a damaged reputation takes years and costs many millions of dollars (Gonzalez 2015).

The deployment of the ID Cross Reference tool helps mitigate all of these risks by allowing users to test the tool with customer data before actually applying the UI to their larger

system. By testing customer data it allows the system to be thoroughly checked over for any errors, bugs, glitches, and potential breaches of data or privacy before deployment. Although the Refresh Cache is not involved in the same risks as the ID tool, it is still important for it to go through all of the same checks that the ID tool goes through so the company can experience the rewards of a successful deployment.

## 8.2 Business Rewards

Successful deployment of both UIs will result in significant rewards along similar perspectives as those evaluated for risk. Operational rewards are the most important rewards to the company because the redesigned ID tool's UI was updated to current standards and saves time for users. The Refresh Cache project also saves time since it standardizes what was previously a very varied process. At our sponsoring company, saving time is very important so the improvement of these UIs will reduce the amount of time needed for the troubleshooting process associated with the ID tool and eliminating the restart process with the Refresh Cache. Our project can help make the company's overall operations more efficient and avoid employee frustration and confusion. It may also make the customers happier too since they will be able to access employees and the company's products quicker. Our project will save the company money and potentially get them new customers.

There are also rewards associated with innovation from the project. The ID Cross Reference tool is being built on a completely new infrastructure that will allow it to be updated numerous times. Since the new system can be continuously improved by the company's development team (our sponsors) for years to come, there is room for new potential ideas for innovation on the tool in the future. These ideas can be used to further decrease employee frustration as well as increase the company's productivity, profitability, competitiveness in a fast changing technology world, and increase its brand recognition with a better reputation. The Refresh Cache will also constantly need to be updated so it can keep up with current technology updates and innovations (NiBusinessInfo.co.uk 2021).

This type of innovation can strengthen the company's reputation. When a company has a reputation of being innovative it shows that the company easily changes and adapts to meet customer needs especially from a technology standpoint. This can lead to more business opportunities especially in new countries that the company has not already started doing business in. A good reputation sets a company apart from competitors, increases profits, and attracts better quality employees (ReviewPoint 2021).

## 8.3 Other Risks

In terms of other risks there are two others that are not as significant but still important to mention: change management risks and training risks. For the former, if the deployment goes wrong and ends up causing the company to lose customers there could be serious impacts on management. This could cause employee morale to plummet and it may cause some to leave the

company altogether. In order to ease this transition or prevent it from happening, it is important for everyone to be ready to adapt to changes on any project team. The most important thing is to communicate clearly what, when, why, and how the changes will happen. This will minimize the risk of bad management risk change and will make any transitions much easier to deal with when they happen.

Training risk can also come with change management risk since if anybody is let go it will take time and money to hire and train new employees and management. Not only that but any previous efforts done on a poorly deployed tool will be wasted. The people using the technology will have to be re-trained on how to use a new one.

There are two types of risks that are not prevalent to either one of our team's projects: marketing and ownership risks. Both projects are internal tools that are only used by specific employees, and will not be available for widespread or customer use. As a result, there is no need to market these tools (Lashford 2016). Since both UIs are internal and owned by the company, there is nobody outside the company sharing or making any money off of them. There is only one owner which means there is no ownership risk.

## 8.4 Risk Culture

The risk culture of the company is extremely risk averse. It handles all sorts of highly confidential information which if leaked could put it in serious legal trouble. The company is risk averse to the degree in which our team was not able to access the back end information during the time of our project. Our team also did not have access to the fully finished tools once they were done either because information security is one of the company's top priorities. For example, in order to gain access to the ID Cross Reference tool, employees who are allowed to use the tool have to go through multiple layers of security authentication before they can use the tool. All people who work with the company including our team must complete several training modules about how to protect sensitive information. This risk averse philosophy also affected what projects our team is doing because the company gave us very low risk projects and did not give us much access to company data. In addition, we worked through virtual machines in order to protect confidential information.

Since the company is highly risk averse, our team functioned in the same way. Our team was extremely careful with the way we handled the information, tools, and data we had access to. The company and the team go with this high risk averse philosophy because one little mistake could put the company in serious trouble that would take substantial time and money to solve.

## 8.5 Financial Savings

In order to gather information about how much time and money our sponsoring company will save with these projects, our team reached out to a manager to get the standard rate of labor, which is on average \$85/hour (including benefits). From there, our team contacted users of the ID Cross Reference tool in order to gather information about how much time it takes them to use

the current UI for both processes involved with it: manually entering (copying and pasting) IDs and uploading a CSV file with a list of IDs. The first set of time trials timed how long it took a user to enter and cross reference one ID, five times. The results for this set of trials are shown below in Table 1 and Figure 10.

Table 1: ID Cross Reference Tool Copy and Paste ID Time Trials

Trial 1	Time (seconds) of Old UI	Time (seconds) of New UI
1.1	6.8	5.6
1.2	6.7	5.0
1.3	7.1	5.5
1.4	6.4	5.6
1.5	6.7	5.4

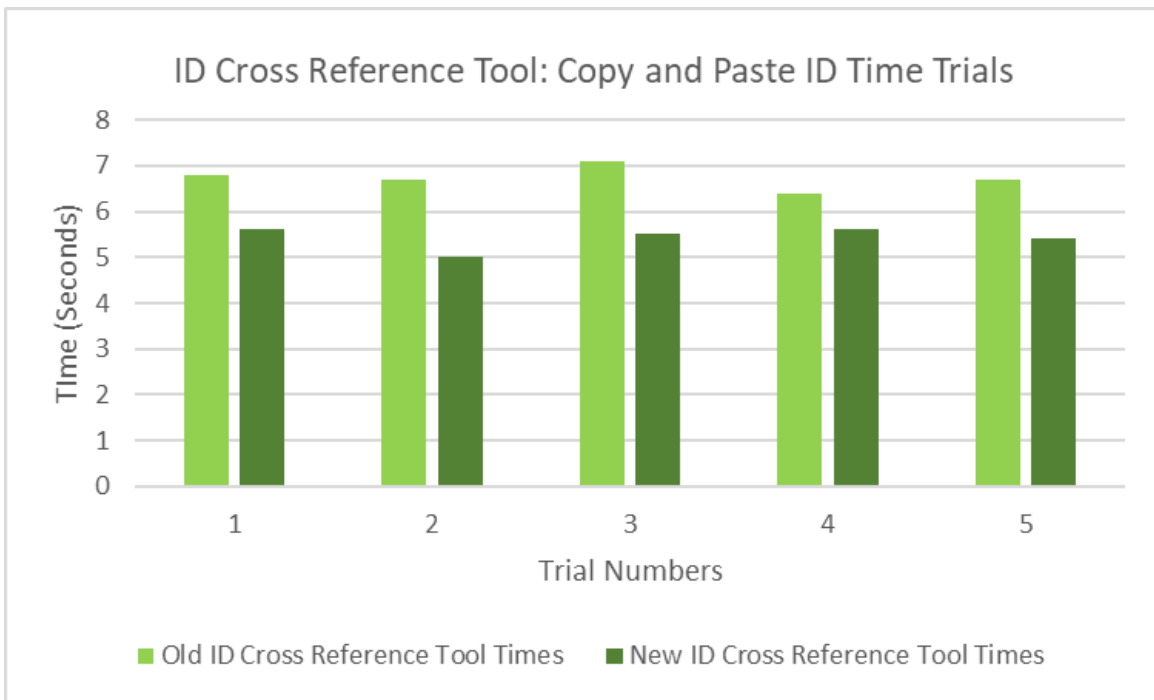


Figure 10: ID Cross Reference Tool Copy and Paste ID Time Trials

In the second set of time trials, we measured the time it takes a user to upload and cross reference a CSV file of IDs, five times. The results for this set of trials are shown below in Table 2 and Figure 11.

Table 2: ID Cross Reference Tool File Upload Time Trials

Trial 2	Time (seconds) of Old UI	Time (seconds) of New UI
2.1	16.8	5.0
2.2	13.2	4.2
2.3	13.4	4.4
2.4	14.1	4.9
2.5	13.2	5.0

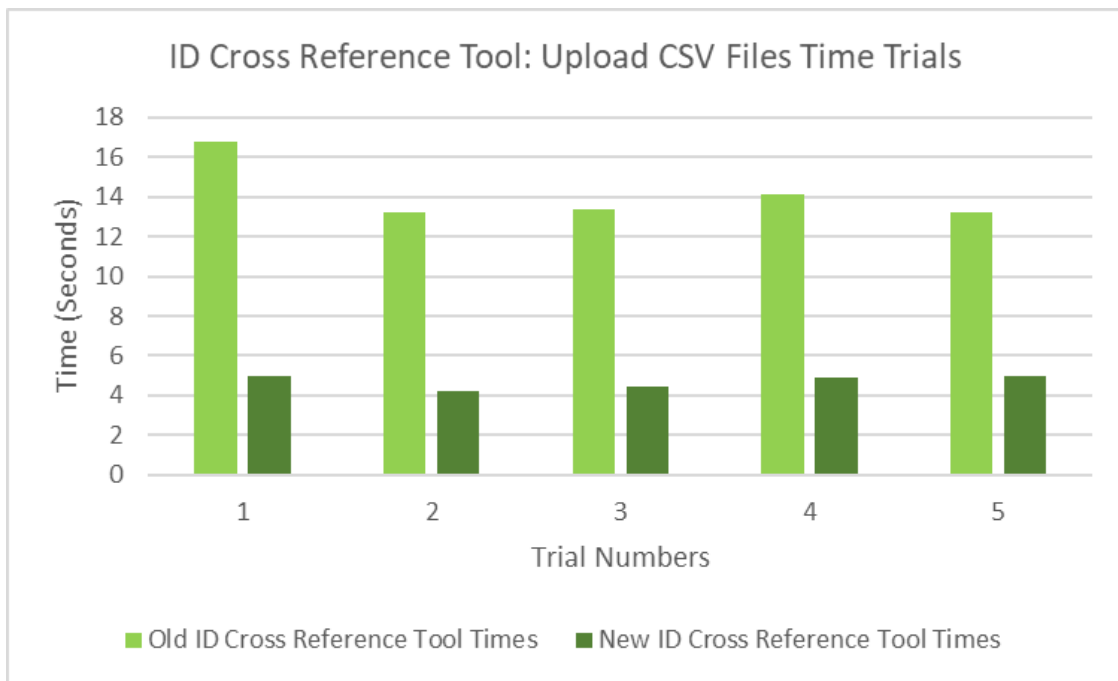


Figure 11: ID Cross Reference Tool File Upload Time Trials

After getting both sets of results, we did some analysis. First, we took the averages of each time trial which are shown below in Table 3.

Table 3: ID Cross Reference Tool Average Time Trial Results

	Old UI Average Time	New UI Average Time	Difference
Copy and Paste	6.7 seconds	5.4 seconds	-1.3 seconds
File Upload	14.1 seconds	4.7 seconds	-9.4 seconds

We then subtracted each of the averages to get how many seconds were saved which was 1.3 seconds per use for the copy and paste process and 9.4 seconds for the upload CSV file process. Our team then figured out how much money was saved using the new tools based on the number of seconds saved on average. Assuming a usage of 4 times per week, the company saves \$6.40/year per person for the copy and paste process and \$69.25/year per person for the file upload process. According to the company, up to 4000 employees will have access to the ID Cross Reference tool, meaning there is a potential savings of \$25,600-277,000 per year depending on type and frequency of usage.

Since the code for the old UI was lost, the new UI was built on new information architecture so the tool can be updated constantly. This new UI also eliminates the mild frustrations that the previous tool had such as not being able to delete separate IDs, a buggy CSV/excel file import process, and the ability to input multiple IDs at once. The tool also combined the manual entry and file upload tabs into a single page. All of these fixes will save users time and unnecessary work if they make an error, overall making the tool more efficient.

Beyond the tool being easier to use, the project achieved the sponsor’s goal of shifting all internal tools to the angular framework and hosting them on a cloud based system. Since the source code of the ID Cross Reference tool was lost, the application needed to get built from scratch. We estimate that requirements gathering, architecture planning, and development took around 200 man hours which translates to \$17,000 saved for the sponsor.

For the Refresh Cache tool, there were no specific metrics that could have been given in terms of time or money saved because there is not a specific process that the tool is replacing. Prior to the tool’s existence, employees who needed to refresh an application endpoint did not have a specific way to do it, so they may waste minutes or hours on various ways of refreshing. Since the tool is still in development and there was no standard way to refresh a cache prior to the tool’s existence, there were no metrics available to us besides a generalization that the new tool will make things faster and easier. We believe that the new tool will save users time because it will provide a standardized process to follow when refreshing an application endpoint which did not previously exist. With every instance that an application needs to be refreshed, this process will eliminate minutes of work—which add up quickly.



## 8.6 Business Learnings

This section details our team's business learnings in 7 different categories:

1. *Leadership*: What our team learned about how to effectively lead a team.
2. *Industry*: What we learned about the industries that our MQP was involved with—Fintech and Software Development.
3. *Operations*: What we learned about how the company runs and keeps track of data and information.
4. *Culture*: What we learned about the culture of the large financial company and what it was like to work with them. This knowledge allowed us to work and communicate more effectively with our sponsors and maximize our time with the MQP.
5. *Working Remotely in a Pandemic*: Due to the COVID-19 pandemic our team ended up having to do our project remotely. Working remotely gave us some business learnings that would have not occurred in a “normal” world.
6. *Time Management*: What we learned about managing our time so we could complete the project and achieve our goals.
7. *Team Management*: What we learned about how to manage a team.

*Leadership*. Leading a team is difficult and requires a lot of responsibility to keep the team together. Risk management is important, but communication is the most important thing when it comes to being a leader. Leadership becomes more difficult when there are multiple (or many) leaders for one team. It is important for team leaders to keep track of tasks that may be forgotten about or fall through the cracks, like following up on an email or setting up a meeting. It is also important for leaders to communicate with other leaders on other projects so they can learn some tips that can help them with projects they are working on. For example, our supervisor frequently consulted with other teams if we were having a problem that she could not help us resolve.

*Industry*. Fintech can be a very rewarding but difficult industry to work in. One may run into access issues, roadblocks, deadline delays, and project postponements—especially when doing part-time work or short consulting projects. Another key learning from this industry is that innovation—whether it is Fintech or any other form of technology—requires patience and takes time. It also requires the ideas and hard work of great teams with strong leaders who provide a clear path and clear goals for whatever project they are working on.

*Operations*. With a larger organization, it is harder to keep track of things. For example, the source code for one of the tools we worked on got lost throughout the years, so when it was time to update the tool to Angular, the UI had to be made from scratch. Also, it is difficult to know who uses what tools when they are shared across a number of departments. Larger companies are also at higher risk to security concerns, so they have a number of measures to protect themselves. These inevitably make certain tasks, such as getting access to tools/code repositories/APIs take more time. Our team learned some of the ways a large company navigates these barriers in order to improve operations such as Agile Scrum and Kanban. Our team also learned to always

maximize potential time to work when things are going perfectly because we did not know when the next access issue or roadblock would occur.

*Culture.* The formality of corporate culture (much like large operations in general) can make tasks take longer than they might in a school setting or different business. There may be multiple levels of approval for a project/design to go through before development even starts. It can take days or even weeks to get approval with multiple steps and this can also cause project delays. Our team also learned that these processes get harder the more formal an industry is. In our case the industry was very formal so it took a lot longer for our team to get things going due to the formality and sensitivity of the Finance, Fintech, and Technology industries.

*Working Remotely during a Pandemic.* Meeting remotely was convenient, but working remotely extended a lot of the access issues we had at the start of our project since we worked through virtual machines rather than in a company office. Additionally, remote work makes it much harder to form connections with teammates or coworkers especially since we never met any of our sponsors in person. Remote work can also make self management more difficult, since there are more distractions around at home than there might be in an office setting.

*Time Management.* Completing tasks and deliverables on time is vital in a business setting because failing to complete your work prevents other people in the company from moving forward. The team also had some other commitments outside of team work to schedule around and some members could not work a traditional 9-5 schedule. Our team had to plan around other meetings, work, and various extracurricular activities making time management more difficult.

*Team Management.* Organization and communication are very important parts of managing a team. Keeping tasks and goals organized ensures efficiency and communication allows all team members to be informed and allows for feedback and the best ideas to be put forward. Assigning clearly defined roles to individuals is important for a team to function. It makes it clear who is responsible for what tasks and allows team members to work independently. One thing our team did to make this work was to have flexible schedules aside from our daily morning meetings. Some of us did work early in the mornings, at nights, and on the weekends outside the usual schedule but our roles were clearly defined and the only thing that mattered was that work was completed on time. Our team also always made sure we were available to help each other out whether it was coding, fixing a part of the paper, or making user stories.

## 8.7 Key Business Takeaways

Some of our key business takeaways were that corporate projects like ours will always have some form of roadblocks whether it would be access issues, important people in the project being out of office for days or weeks at a time, or the changes in the project scope entirely. Since our team was faced with many setbacks and waiting periods, it was important for us to be as

prepared as we could and maximize the time we had. Our team did this by utilizing times of little to no access to the repositories to focus on the paper.

If we were to redo this project, we would recommend starting the onboarding process with the sponsors during the PQP term. This could have a number of benefits which include creating relationships with sponsors earlier and creating more time for development during the project term. Due to the security risks that financial firms face, getting access to software and code repositories takes time. Beyond that, once access is granted, it is still necessary to troubleshoot any problems which takes time. It is possible that these issues are caused due to remote work, however, we feel there would be a smoother transition if the onboarding process had begun earlier.

Starting the project early is important to allowing MQP teams to have sufficient time to complete the project. In addition, the initial onboarding and setup requires many people to complete small tasks and thus takes a great deal of time before MQP students can begin working on their project. Given the short time period, it is important to both the MQP team and the sponsor that they maximize the time spent working on the project rather than the setup.

## 9 Assessment

Our goal was to redesign and implement user interfaces for two internal company tools, an ID Cross Reference tool and a refresh cache tool. Our team was successfully able to redesign and implement the ID Cross Reference UI, but we were only able to redesign the Refresh Cache UI.

### 9.1 Achievements

Our team achieved all of the goals on the ID Cross Reference tool which included an intuitive information architecture and visual design. In addition to those goals, our team also analyzed how much time and money the company would save from this new UI. Our team was also able to modify the tool based on what users wanted in order to make their troubleshooting process easier.

We were able to partially complete our goals for the Refresh Cache tool which had a similar design process to that of the ID Cross Reference tool. Our team was able to complete a mock-up, get feedback, and make necessary adjustments. Due to the remote setting, our team had to go through additional steps to gain access to a development environment (i.e. setting up virtual machines, company accounts, obtaining access to codebase and necessary software) at the beginning of our project. After facing this delayed start to our development, we were ultimately not able to build or implement the Refresh Cache UI. Furthermore, our team was not able to analyze specific finance and time metrics for this tool although our team learned that the Refresh Cache project should save the company lots of time.

Even though our team was not able to completely finish the Refresh Cache project, we were still able to overcome many setbacks and delays. Despite our access issues, we still produced an effective internal UI for the ID Cross Reference tool and produced a mockup of the Refresh Cache UI for future developers to reference. Despite these setbacks, we learned it was important to continuously communicate with the project owners to learn what we should prioritize. We learned that sometimes teams do not accomplish their original goals, but it is important to complete one project in full before moving on to the next.

## 9.2 Major Takeaways

Our team learned a lot throughout this MQP through setbacks and successes. When encountering access issues, we worked on other tasks such as writing the paper, creating detailed mockups for our sponsor, choosing design patterns, gathering requirements, and writing user stories. This meant that once the development environment was set up, we were ready to start coding. Another major reason for our success was our team had great communication. We clearly communicated our objectives and goals and split up the work based on our strengths. Whenever an access issue occurred or the team got stuck, we immediately reached out to the necessary supervisor to get the issue solved as soon as possible. We also learned about the importance of being flexible to account for project setbacks and differing schedules.

## 9.3 Technical Learnings

Prior to the start of the project, the developers had only heard of the Angular framework. This project was a great opportunity to allow us to learn not only how to use Angular, but also how powerful it is and why it is considered an industry standard. Angular is perfect when there are components in a webpage that need to get repeated with slightly different information. The general structure of the framework helps out with organization. Often files tend to be short and the benefits of this grow exponentially with application size. Concise code allows new developers to read and understand existing code much quicker. This makes the process of adding features or changing existing features much easier for developers.

We also increased our understanding of styling principles. Angular uses .scss files for styling the HTML components. The developers were familiar with css, but had to learn how to use scss in order to be successful. Though both languages are very similar, the additions that scss has proved to be helpful, simplified our code, and made it easier for new developers to understand. It allowed us to store variables which were used throughout the app and allowed class names to be defined as children of other classes.

The entire team also learned how to operate using the Agile Kanban methodology. Kanban was helpful for organizing and tracking our progress, and keeping each team member busy and accountable for their work. We learned how to regulate our workload using pipelines and how to organize weekly workloads in the Kanban format.

## 9.4 Self Reflection for Industrial Engineering Majors

As we've discussed in Chapters 5 and 6, the visual design of the two UIs was a large part of our project. We found that making the design process collaborative not only as a team but also with our sponsor helped us get to the ideal design in a short amount of time. Starting with a demonstration of the tools from our sponsor and a clear overview of the requirements from the company helped us determine the most important features of the design. Understanding the tools' functionality assisted us in creating not only visually pleasing but very functional UI designs. We created each design as a team and worked through designing for functionality, required features, and then aesthetics, before immediately submitting the draft to our sponsor for a review. Our UI design for the ID Cross Reference tool was approved after three rounds of feedback from our sponsor, and our design for the Refresh Cache was approved after two. Only after we got full design approval did we begin full development of each tool.

The main constraints we considered in the design and impact of our UIs were human interaction and sustainability. Since both UIs would be used by employees, our team had to make usability a leading factor in our designs. Our work is also helping the company update its UIs onto a new framework, which will allow them to be easily edited by future developers. Due to this, our team had to consider the complexity of our designs as well as our code. We wanted to make sure that design elements and code could be easily edited in the future whenever the company needs to update these UIs again.

Our biggest learnings throughout our project were mainly centered around the culture and structure of the company. As we discussed in Section 8.6, the large scale of sensitive information the company handles led to many delays in our project. However, we learned a lot from our sponsor about different processes within the company and why and how they are the way they are. We were able to see interactions between smaller teams, learn about the hierarchy between different levels of management, and experience the company's full stack development process. Throughout our project meetings, our sponsors taught us a lot about Agile and software development processes specific to the business as well as differences between different companies and industries.

Throughout our project, our communication and teamwork skills grew immensely. A big contributor to this learning was when we created the designs for each UI. Since we worked online, we were on a voice call with each other as we all edited the same document. We quickly discovered the importance of voicing your actions (ex. "I'm going to work on creating button x now") so we didn't create the same elements multiple times or delete others' work. We also learned how hard it can be to express a visual design in words, so working together helped us make sure we were interpreting the design requirements in the same way. By the end of our project we were much more aware of and attentive to one another. Additionally, each Business School student took turns leading morning standup meetings, which taught us about team leadership within professional projects.

## 10 Future Development

There are many different areas of future development for the two tools we worked on as well as our project as a whole. Since we had such a slow start to our project, the main thing we felt we were lacking was time. If we had gotten more time to work on our project, we would have been able to polish our code for both tools more, develop the UI for the Refresh Cache tool, and also complete the third project for the company. Another part of our project that we would want to improve is user testing. If we had more time and resources we would conduct more formal user testing with company employees and have a more iterative process between testing and development to make sure that our designs were effective. Additionally, more time would have allowed our team to more fully analyze business and financial data and obtain specific metrics to better approximate the benefits of our project.

For the Cross Reference tool, the main thing our UI helped improve was employee efficiency. The cleaner look and condensed page was more straightforward than the original UI which we believed would save users time when using the tool. Unfortunately, we were unable to do much testing on our time-saving theory before the end of our project, so this is one area we wish we could have explored more. On a larger scale, we think further improvements could be made to the tool if our team and sponsor had more resources—perhaps some way of integrating the tool as a browser add-on for employees with access so it could be more easily accessed.

## 11 Conclusion

Our team was able to rebuild the ID Cross Reference tool using the Angular framework, getting the sponsor closer to achieving their goal of updating all their internal tools to use this framework and host their applications on the cloud. Our redesigned UI got positive feedback from users and will save users time. Although we did not develop the Refresh Cache application in Angular like we had hoped to, we did gather requirements and design a mockup UI for future developers to reference.

This project taught us how important time management is, especially within a large financial company. Due to the nature of its business, every component of the company has added security which makes processes such as setting up virtual machines and getting access to code repositories and applications take longer than expected. Our developers also learned how to use the Angular framework which is valuable since it is an industry standard tool for developing applications. Our team was able to practice working with the Agile Kanban methodology and two of our members also practiced being product owners and taking on the role of scrum master. Overall, our team gained valuable experience working on a real world project and learned many skills that we can take with us into our future projects and professions.

# Bibliography

- AgileThought. "Getting Started with Kanban." Accessed November 22, 2021.  
<https://agilethought.com/blogs/getting-started-with-kanban/>
- Beaubein, Sean. "8 Websites With Really Awesome User Interface (UI) Design." CareerFoundry. Last modified August 6, 2021.  
<https://careerfoundry.com/en/blog/ui-design/8-sites-with-great-ui/>
- Berezhoni, Roman. "What is UI Design and Why is it Important?" F5 Studio. Last modified April 12, 2019.  
<https://f5-studio.com/articles/what-is-user-interface-design-and-why-is-it-important/>
- Business News Daily. "What Is Agile Scrum Methodology?" Accessed November 8, 2021.  
<https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>
- Djamasbi, Soussan; Strong, Diane. "User-Experience-driven Innovation in Smart and Connected Worlds." *AIS Transactions on Human-Computer Interaction*. (2019): 215-230. .  
<https://doi.org/10.17705/1thci.00121>
- <Financial Company>. "About <Financial Company> - Our Company." Accessed November 12, 2021. <https://www.<Financial Company>.com/about-<Financial Company>/our-company>
- <Financial Company> Labs. "Featured Case Studies." <Financial Company>. Accessed November 12, 2021. <https://labs.<Financial Company>.com/work/overview>
- Gonzalez, Deborah. "Managing Online Risk." *Apps, Mobile, and Social Media Security* (2015): 53-77. <https://doi.org/10.1016/B978-0-12-420055-5.00003-7>
- Lashford, Richard. "Technology, Risk, and Change Management." Management Concepts. Last Modified July 1, 2016.  
<https://blogs.managementconcepts.com/2016/07/01/technology-risk-and-change-management/>
- Morgan, Lisa. "Operational Risk." TechTarget Network. Accessed November 18, 2021.  
<https://searchcompliance.techtarget.com/definition/operational-risk>
- NIBusinessInfo.co.uk. "Use Innovation to Start or Grow Your Business." NIBusinessInfo.co.uk. Accessed November 18, 2021.  
<https://www.nibusinessinfo.co.uk/content/advantages-innovation>

- Peterson, Amanda. "5 Leading Tech Companies Using Great UX to Win." Motivate. Last modified 2019.  
<https://www.motivatedesign.com/5-leading-tech-companies-using-great-ux-to-win/>
- Radigan, Dan. "What is kanban?" Atlassian. Accessed November 15, 2021.  
<https://www.atlassian.com/agile/kanban>
- Reviewpoint. "What is Reputation and why it is Necessary to Grow Your Business." Reviewpoint. Accessed November 18, 2021.  
<https://blog.reviewpoint.com/what-is-reputation-and-why-is-it-necessary-to-grow-your-business>
- Sa, Erika. "A preview of the new agility boards in Jira Software." Atlassian. Published May 30, 2018.  
<https://www.atlassian.com/blog/jira-software/preview-new-agility-boards-jira-software>
- ScienceDirect. "Reputational Risk - an overview." Accessed November 18, 2021.  
<https://www.sciencedirect.com/topics/computer-science/reputational-risk>
- Tutorialspoint.com. "Software User Interface Design." Tutorialspoint. Accessed November 12, 2021.  
[https://www.tutorialspoint.com/software\\_engineering/software\\_user\\_interface\\_design.htm](https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm)
- Usability.gov. "User Centered Design Basics." US General Services Administration. Accessed November 12, 2021.  
<https://www.usability.gov/what-and-why/user-centered-design.html>
- Usability.gov (2). "User Interface Design Basics." US General Services Administration. Accessed November 12, 2021.  
<https://www.usability.gov/what-and-why/user-interface-design.html>
- Wood, Urko. "The 2 Great Risks of Innovation." The Business Risks. Last Modified February 25, 2016.  
<https://www.bizjournals.com/bizjournals/how-to/growth-strategies/2016/02/the-two-paths-of-innovation.html>