Project Deliverable Report Instructions

This document is a template for the project deliverable submissions. Your group will edit this document all semester and submit it each time you have an updated section (new deliverable is done). The goal is to keep everything in one place to be able to refer back to previous work and save you some time formatting a new document for every deliverable. Please keep track changes ON so that the TA can see what has been changed every time it gets submitted. So that it does not become laggy when the document is too large, 2 templates are provided (PD B-D and E-I).

Template conventions:

- Remove all red text, it is only there to guide you
- Remove this page (instructions)
- Replace all instances of <xxx> with the appropriate information for your group, for example you could replace <GROUP NUMBER> by 'B1.3'

GNG2101

Design Project Progress Update

B1.1

Submitted by:

Ryan Langley, 300365966

Matthew Hadjis, 300356139

Jamie Watson, 300356613

Nick Mesquita, 300361908

Leonardo Atalla, 300352589

October 20, 2024

University of Ottawa

Table of Contents

List of Figures

List of Tables

List of Acronyms and Glossary

Provide a list of acronyms and associated literal translations used within the document. List the acronyms in alphabetical order using a tabular format as depicted below.

Table 1. Acronyms

Acronym	Definition
AIS	Asset Inventory System
BCHC	Bethany Children's Health Center
CSS	Cascading Style Sheets
DFX	Design for X
LCA	Life Cycle Analysis

Provide clear and concise definitions for terms used in this document that may be unfamiliar to readers of the document. Terms are to be listed in alphabetical order.

Table 2. Glossary

Term	Acronym	Definition

1 Introduction

The goal of this project is to develop an improvement for Bethany Children's Health Center's (BCHC) current asset inventory system (AIS). BCHC's current AIS operates too slowly for the health professionals and takes from the time that they can be helping patients. The design of this project will expedite the check in/check out process of the various equipment that is used at BCHC as well as simplify the inventory management of the entire system.

This document contains the work for project deliverables E through I. Each section encompasses its own deliverable.

2 Prototype 1, Project Progress Presentation, Peer Feedback and Team Dynamics

Abstract

This deliverable is focused on the development and testing of the first prototype for the project. The prototype was developed with a focus on the admin capabilities of the software as well as the overall handling of high usage in both users and assets. Tests were developed for this prototype to ensure that the target metrics and specifications are met.

2.1 Prototype 1

1.) Our critical assumption for this prototype was that the prototype be able to handle 10,000+ assets in the database, as well as over 2000 users. This relates to scalability, reliability, and testability. If the prototype can handle having 10,000 assets and still run, this handles the scalability component of our design. If the prototype can still function reliably with the amount of assets, and not lag or stutter significantly, then it satisfies the reliability requirement. For testability, our prototype satisfies the test plan that was outlined previously. In addition, the prototype is built very compartmentally, in doing so errors are easy to isolate and resolve.

2.) Given the complexity and number of files and code in the prototype, we've elected to add links to a demo and to our Github as opposed to adding many screenshots of code or UI elements. The diagrams below show the flow of how the full prototype would work.

https://drive.google.com/file/d/1YRaT4lhlIbHpK0W4qu_8pZvEGA8mO5b2/view?usp=sharing

Github Link: https://github.com/Jamie-Watson/GNG2101-AssetInventory

Figure 2.1.1 - Flow Chart of Program Design

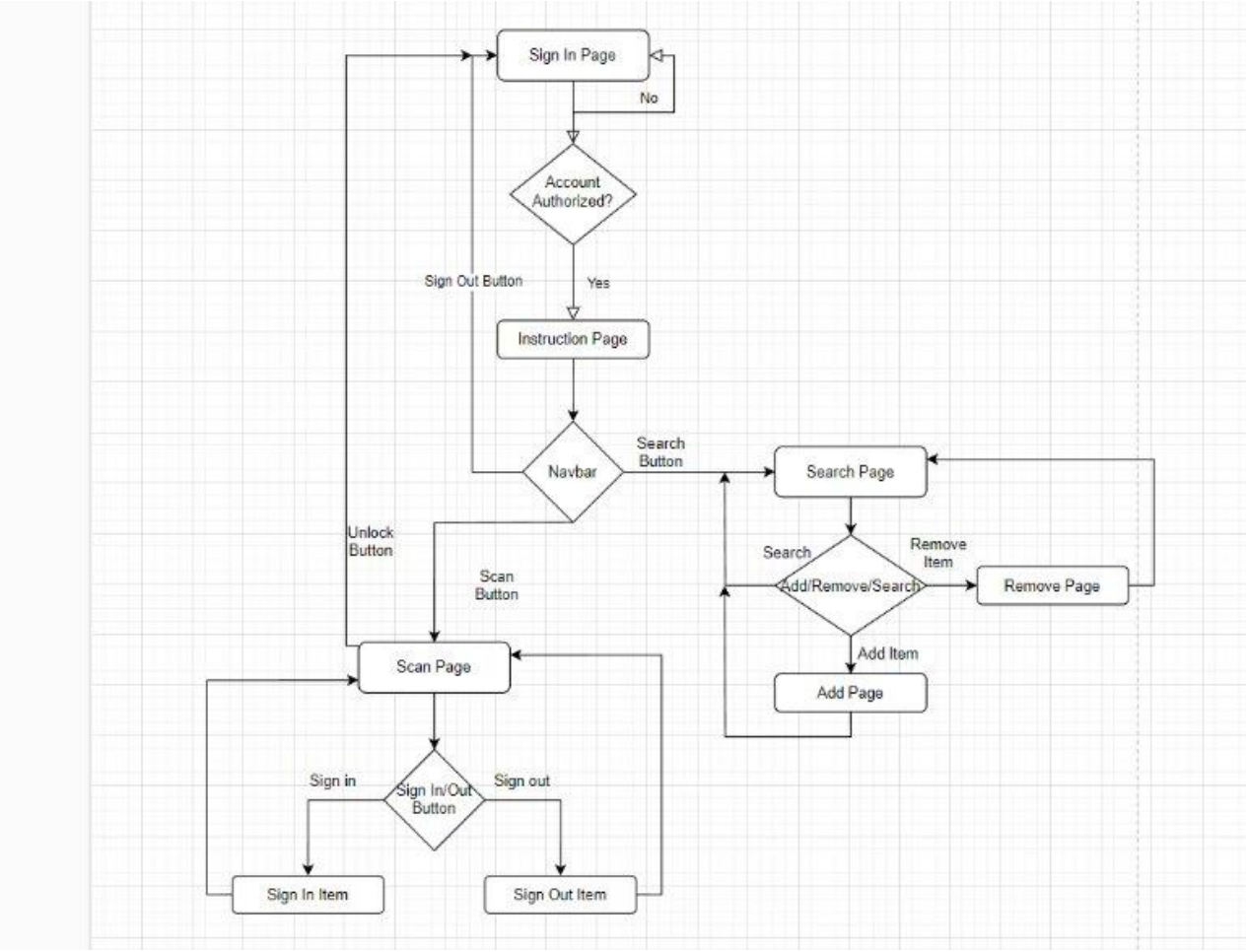
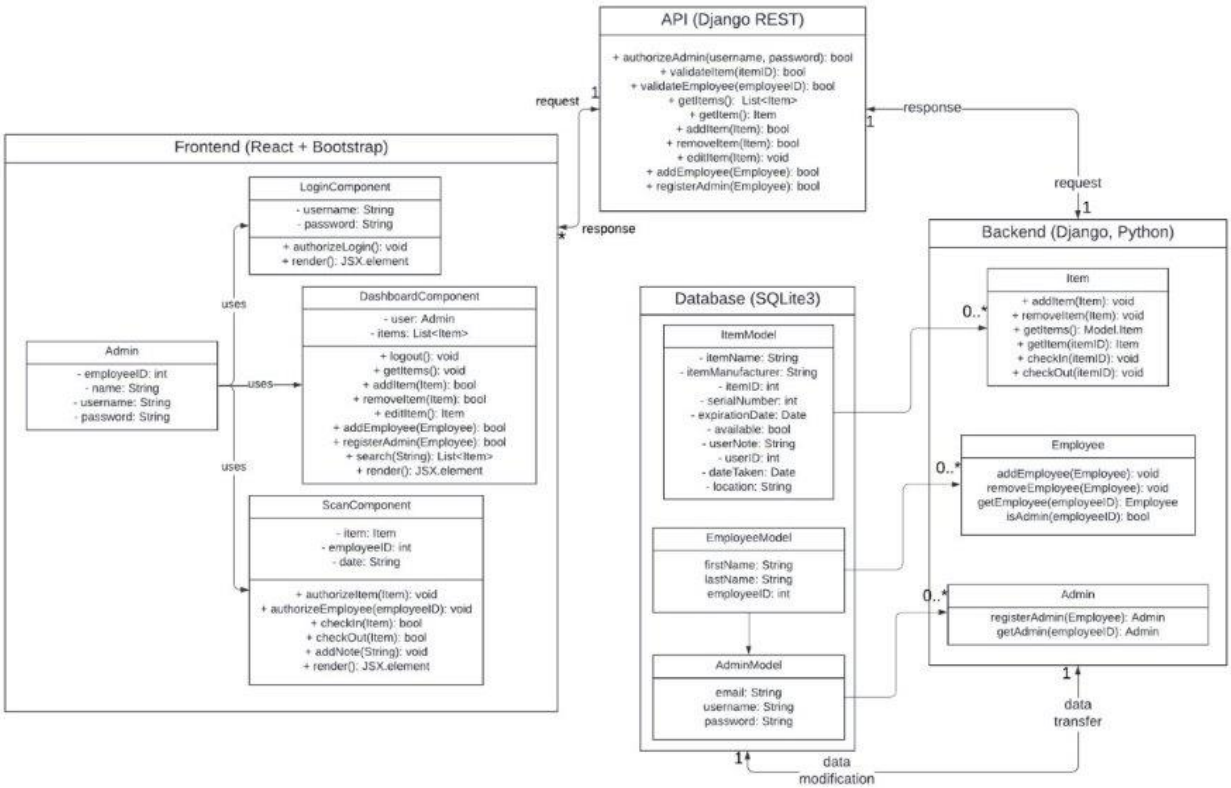


Figure 2.1.2 UML Design



3.) In comparison to our target specifications from deliverable C (desired assets >=2000, preferably infinitely scalable, desired users >=2000, preferably infinitely scalable, cost = $0 and check in/out time being <60 seconds), our prototype can test the first two of three of these. As stated above, the prototype can handle up to 10,000 assets without noticeable lag or stuttering. The same can be said about the number of users, minimal lag is present with large numbers of users. The check in/out time is unable to be tested with this prototype, as it currently lacks the physical component necessary to test this aspect.

Test Number	Reason for Prototype	Evaluation Criteria/Determine Measurables	Level of Prototype

1	Performance measurement	System Functionality (Number of Assets)	Medium/High Fidelity, Comprehensive
2	Performance measurement	System Functionality (Number of Users)	Medium/High Fidelity, Comprehensive

Kind of Prototype	Metrics	Test Description
Analytical	Assets: # before system failure	How many assets the prototype can support before the system fails
Analytical	Users: # before system failure	How many users the prototype can support before the system fails
Analytical	Cost	How much did it cost to produce this prototype; will additional costs be accured during further development

	Analysis Method	Minimum	Expected Physical Maximum	System Breaking Point	Interpretation

# of Assets	Add assets until the system fails	2000	4000	30,000	This passes easily, the number of assets will not be an issue.
# of Users	Add users until the system fails	2000	3000	5000	This passes easily, the number of users will not be an issue.
Cost	Cost of prototype	0	0	0	This passes easily as all aspects of this prototype were developed using free tools available to anyone

Given that the current prototype meets the minimum expectations for number of assets and users, in addition to the expected maximum of assets and users, we elected to test the system to see what the breaking point of the system would be (e.g. how many assets or users can the system handle before it becomes non-functional). The expected maximums were created based off of realistic

limits for how many assets and user the BCHC could handle. Given the physical constraints of the building, it is unlikely that the number of assets will exceed around 4000. Additionally, due to the same physical constraints that limit the amount of assets, the number of users will likely not exceed 3000 at any given time. These satisfy the DFX requirements we set out for initially.
Since we want the final system to be free or as financially available as possible to BCHC we have developed our prototype using free tools such as Bootstrap and Django in addition to python in order to maintain a development cost of 0 dollars. The only expected cost in further prototyping is the physical scanner required to scan the barcodes of staff and on the equipment itself. This connects to our DFX of designing for affordability as the lower the cost of development, the less that B.

Conclusion:

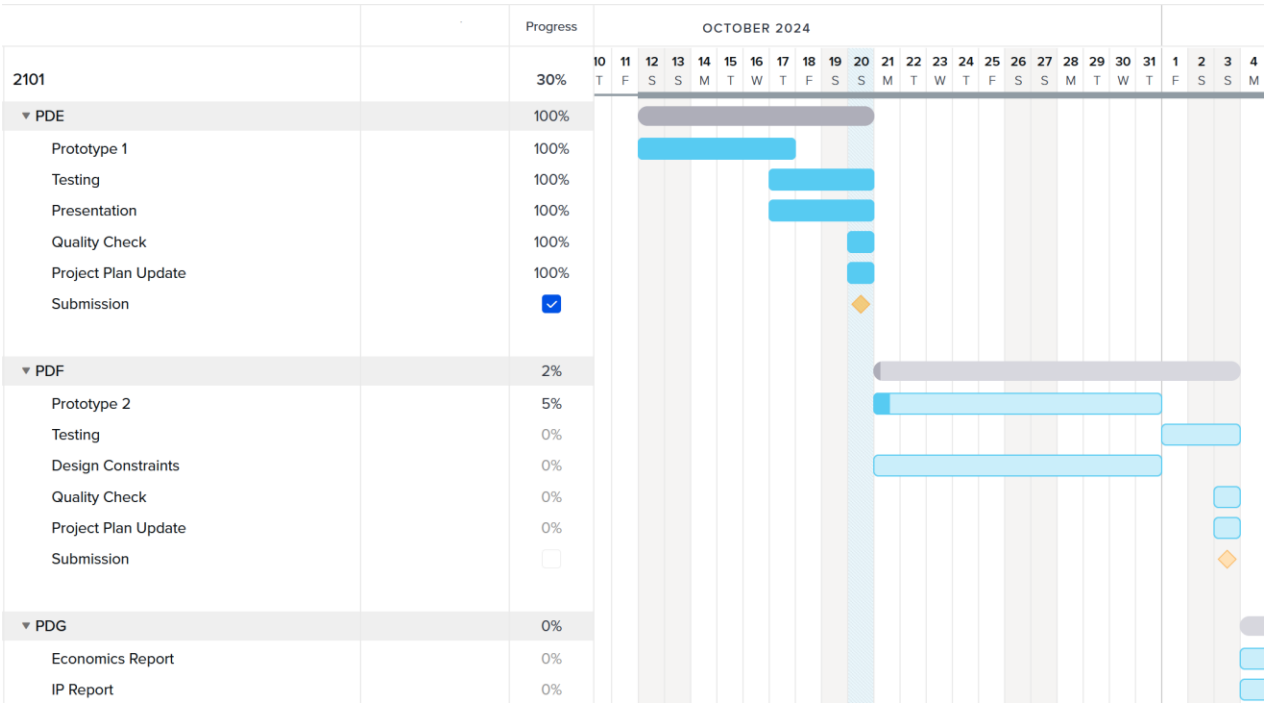
In conclusion, after completing the prototype development and testing, we successfully validated our critical assumption. The prototype performed as expected, demonstrating that it could handle a large number of items. Our testing results, documented and compared to target specifications, show that the system meets the necessary requirements.

2.2 Project Progress Presentation

PDE Group B1.1.pptx

2.3 Add a link to your presentation slides.Project plan update

Figure 2.3.1


Progress
OCTOBER 2024
2101
30%
10 T 11 F 12 S 13 S 14 M 15 T 16 W 17 T 18 F 19 S 20 S 21 M 22 T 23 W 24 T 25 F 26 S 27 S 28 M 29 T 30 W 31 T 1 F 2 S 3 S 4 M
PDE 100%
Prototype 1 100%
Testing 100%
Presentation 100%
Quality Check 100%
Project Plan Update 100%
Submission
PDF 2%
Prototype 2 5%
Testing 0%
Design Constraints 0%
Quality Check 0%
Project Plan Update 0%
Submission
PDG 0%
Economics Report 0%
IP Report 0%

3 Design Constraints and Prototype 2

3.1 Design constraints

The two most important DFXs that influence our prototypes are Design for Scalability and Deisgn for Reliability. Design for Scalability is important because the client's most pressing need is that the AIS must be able to handle the high number of items and users that BCHC has. Therefore, by designing for scalability we ensure that the AIS we provide can scale to handle any fluctuations in assets or users that may occur. Design for Reliability is an important design constraint because the majority of the non-admin users of the AIS will be health care professionals who may not be as capable of troubleshooting errors that may occur in the day-to-day usage of the system. By designing for reliability, we aim to minimize any downtimes that may occur. This will then assist in maximizing the time that the healthcare professionals can spend doing patient treatment.

For our design constraint related to designing for reliability the current design is capable of handling 30,000 assets and 5000 users which easily surpasses the marginal values and expected physical maximums of our design and the centre respectively. Therefore, no issues are expected to come up in the day-to-day operations because off volume of assets or users. Our current detailed design addresses potential issues in the check in/out process by ensuring that items exist and are not currently signed out before permitting a user to check out an item. Further testing in prototype 2 and our final prototype for design day will ensure that the program runs reliably during the check in and check out process. Design constraints relating to scalability are similar to the constraints for reliability. As stated previously, the prototype as of current is able to handle 30,000 assets and 5000 users. Given that the maximum expected physical amount for these is 4000 and 3000 respectively, this satisfies any scalability requirement with little effort

In terms of proof to demonstrate the effectiveness of our design in satisfying the constraints, the conduction of previous testing has demonstrated how our design satisfies the Design for Reliability and Scalability constraints. For these constraints, two functionality tests were conducted where we inserted as many assets and users into the system as possible and observed at which point the system failed. This is an accurate measure for reliability because if our system can run effectively, and not crash or stutter, while handling a larger number of users/assets than the health center requires, then it can therefore be considered reliable enough to handle our target specifications. In addition, for scalability, the system being able to handle a larger number of assets/users and still running effectively demonstrates that our system would be able to accommodate for any potential scaling that the health center might require in the future. The proof of these declarations can be seen in the testing table below:

Test Number	Reason for Prototype	Evaluation Criteria/Determine Measurables	Level of Prototype
1	Performance measurement	System Functionality (Number of Assets)	Medium/High Fidelity, Comprehensive
2	Performance measurement	System Functionality (Number of Users)	Medium/High Fidelity, Comprehensive

Kind of Prototype	Metrics	Test Description
Analytical	Assets: # before system failure	How many assets the prototype can support before the system fails
Analytical	Users: # before system failure	How many users the prototype can support before the system fails
Analytical	Cost	How much did it cost to produce this prototype; will additional costs be accured during further development

	Analysis Method	Minimum	Expected Physical Maximum	System Breaking Point	Interpretation
# of Assets	Add assets until the system fails	2000	4000	30,000	This passes easily, the number of assets will not be an issue.
# of Users	Add users until the system fails	2000	3000	5000	This passes easily, the number of users will not be an issue.
Cost	Cost of prototype	0	0	0	This passes easily as all aspects of this prototype were developed using free tools available to anyone

3.2 Prototype 2

3.2.1- Client Feedback/ Testing Results

There is no new client feedback, as client meeting 3 has not yet occurred. However, during the presentation we received feedback from the professor that we should incorporate Nielson's Heuristics into our UI design.

3.2.2 - Critical Product Assumptions

Our critical assumption is that our barcode scanner works for all barcodes, and that our system can accurately read the information from the scanner. This critical assumption relates back to two out of our five DFX's that we outlined: Design for Reliability and Design for Testability. If the barcode scanner works as intended and can accurately scan the sample barcodes that we give it, it would therefore be considered reliable, as it would theoretically work for any barcode at the health center.

Additionally, if our system can incorporate the information received from the barcode scanner during each test, then it will have successfully achieved our goals from the test plan, ensuring that it has been effectively implemented for testing.

3.2.3 - Prototype Development

Our second set of prototypes will incorporate more of the physical integration into the system, specifically the use of the barcode scanner. In the final prototype, the system should be able to automatically sign out assets by just scanning them. As such, in this prototype, the more integration we have with the barcode scanner the better.

3.2.4 - Prototype Documentation

Our prototype 2 hasn't had significant visual changes, as it mostly is trying to progress the backend implementation of the barcode scanner. Due to this, the documentation of the prototype is effectively the same as prototype 1, and we'll link the same items we had previously below.

Demo link: https://drive.google.com/file/d/1YRaT4lhlIbHpK0W4qu_8pZvEGA8mO5b2/view?usp=sharing

Github Link: https://github.com/Jamie-Watson/GNG2101-AssetInventory

3.2.5 - Prototype Testing and Analysis

Test Number	Reason for Prototype	Evaluation Criteria/Determine Measurables	Level of Prototype
1	Performance measurement	System/Barcode Functionality	Medium/High Fidelity, Comprehensiv

Kind of Prototype	Metrics	Test Description
Analytical	Yes or No Value	Can the barcode scanner read the barcode and have this

		information sent to our system?

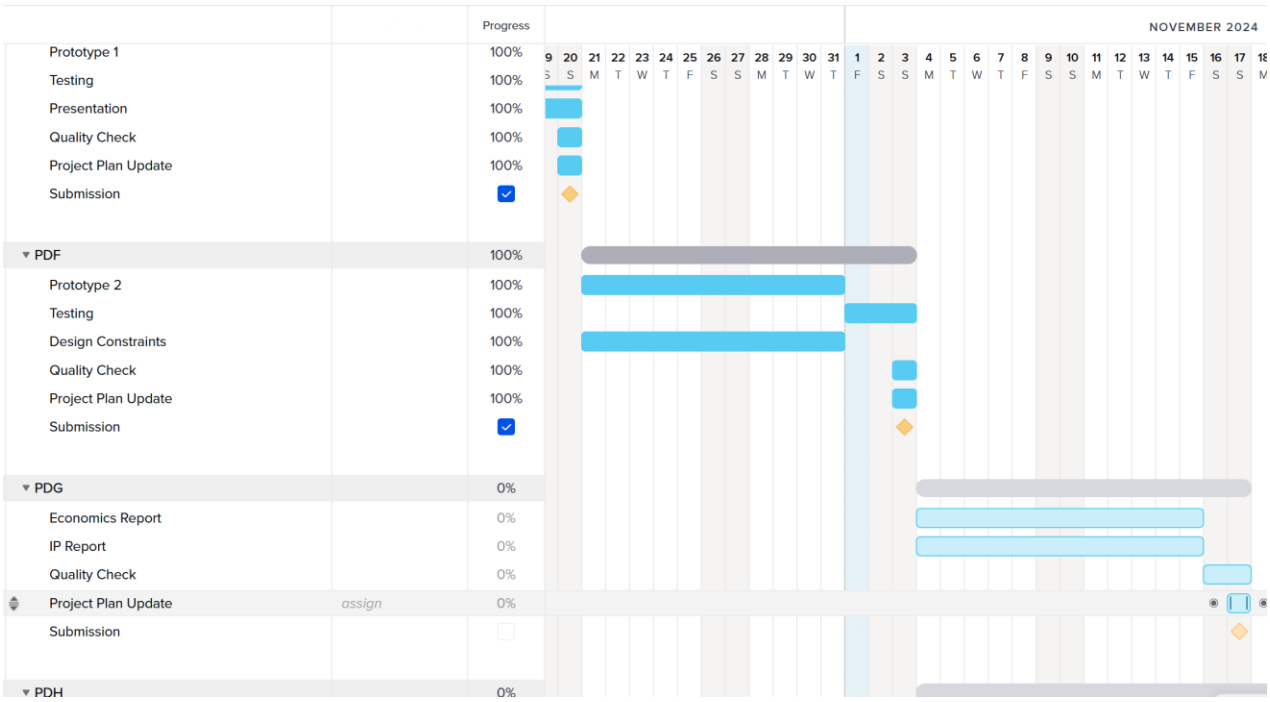
Analysis Method	Minimum	Interpretation
Scan the barcode and observe whether it outputs the text/value.	2	Our barcode scanner can read the barcodes that we give it, and it accurately outputs a value from the barcode.

3.2.6 - Client Presentation

We plan to present a demo of our current system at the client meeting, as well as the barcode scanner in use. We'd like to gather more feedback from the client, as up to this point the major points of feedback have been features we were already planning to integrate, or the client expressing that they liked what we had so far. As much as the encouragement is appreciated, it leaves us with a lack of refinement in the direction that is desired for the design. Preferable feedback would be different specific fields that the admin would be able to enter for items, such as expiry date, etc.

3.3 Project plan update

Figure 3.3.1


Progress
NOVEMBER 2024
Prototype 1 100%
Testing 100%
Presentation 100%
Quality Check 100%
Project Plan Update 100%
Submission
PDF 100%
Prototype 2 100%
Testing 100%
Design Constraints 100%
Quality Check 100%
Project Plan Update 100%
Submission
PDG 0%
Economics Report 0%
IP Report 0%
Quality Check 0%
Project Plan Update assign 0%
Submission
PDH 0%

4 Economic and IP Considerations

4.1 Economics report

Cost Classification Table:

	Variable	Fixed	Direct	Indirect
Material	$9.65 per unit (Barcode Scanner)	$0	$9.65	$0
Labour	$0	$800,000	$0	$800,000
Overhead	$0	$1000 (Business Insurance) $20,000(Marketing) $5,000 (Admin Fees)	$0	$0

1. Cost Analysis:

- **Variable Costs**:
  - **Material Cost (COGS)**: $9.65 per user, applied once per new subscription.
- **Fixed Costs**:
  - **Salaries**: $800,000 per year for the team.
  - **Marketing Expenses**: Estimated at $20,000 annually to establish brand visibility and generate leads.
- **Indirect Costs**:
  - **Overhead**: Minimal due to remote operations.
  - **Miscellaneous Administrative Expenses**: $5,000 per year.

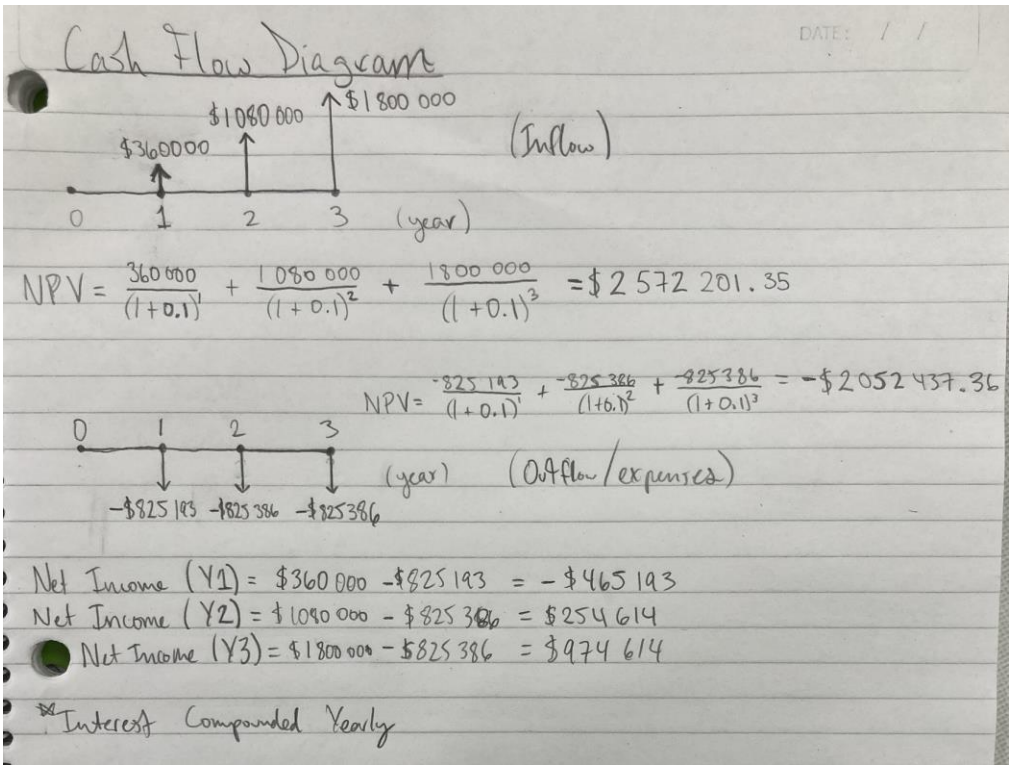
2. 3-Year Income Statement

Year	**Expected Users**	**Revenue**	**COGS for New User**	**Gross Profit**	**Operating Expense**	**Operating Income**

	(End of Year					
1	20	$360,000	$193	$359,807	$825,000	-$465,193
2	60	$1,080,000	$386	$1,079,614	$825,000	$254,614
3	100	$1,800,000	$386	$1,799,614	$825,000	$974,614

3. NPV Analysis and Break-Even Point

Using NPV analysis for cumulative cash flows over three years, with an assumed discount rate of 10%:



- **Year 1**: Negative cash flow of approximately $465,193.
- **Year 2 and 3**: Positive cash flows begin to cover the initial deficit, with cumulative breakeven expected towards the end of Year 3.

The break-even point occurs at roughly 46 users gained, or 17 months in.

4. Assumptions Justification

- **Market Demand**: Assumes demand growth aligned with organizations adopting AIS gradually, reaching full capacity by Year 3.
- **Pricing Strategy**: Monthly subscription of $1,500 fits the high-value, enterprise software market.

Assumptions:

- Instead of manufacturing the barcode scanners, we will purchase them in bulk
    - The distribution of the barcode scanners will be done through drop shipping rather than through a large warehouse
    - This decision was made to reduce the cost of manufacturing the barcode scanners, as the amount of them required does not justify making them
- Our company is fully online, we don't need an actual building to run our operation
    - This reduces maintenance costs of having to keep a building open every day when the majority of the work can be done online
- Users increase linearly, from 0 to 100 by Year 3
- This is just a typical increase of users; we decided not to choose exponential as we did not think our product would expand this fastRevenue calculated based on cumulative monthly subscription.

The one-time material cost applies only to new subscriptions.

References:

- https://fortune.com/2023/08/16/atlassian-airbnb-remote-work-return-to-office/
- https://www.alibaba.com/product-detail/2D-Wired-Barcode-Scanner-Handheld-High_1601195005711.html?spm=a2700.galleryofferlist.normal_offer.d_title.17a613a0N108BR&selectedCarrierCode=SEMI_MANAGED_STANDARD@@STANDARD
- https://www.zensurance.com/
- https://snipeitapp.com/pricing
- https://www.brahmin-solutions.com/pricing

4.2 Intellectual property report

https://patents.google.com/patent/US20090195384A1/en?q=(asset+inventory)&oq=asset+inventory

This is a design for a system and method for inventory management that is very similar to our design. It utilizes a similar methodology to our design where a scanner scans a label on an asset. The system, which knows where the scanner is located then adds the asset to the inventory in the location of the scanner.

While the system is very similar to our design, the patent was filed in the United States, therefore it is not applicable to our product in Canada. Furthermore, the assignee has let the patent expire, which means that they no longer have the exclusive rights to the design.

The design in the patent is similar to our product, but there are several distinguishing features that make our product unique. Our product allows for users to sign in and sign out assets, as well as providing additional features related to the overall inventory. This includes the admin being able to monitor who has what assets and the function to add and remove assets. Another difference between the two designs is that our program can be utilized from any location as the assets are assigned a location based on the object instead of where it was scanned. Therefore, our program will track who has the asset instead of it simply being “signed-out.””

https://patents.google.com/patent/US20180068264A1/en

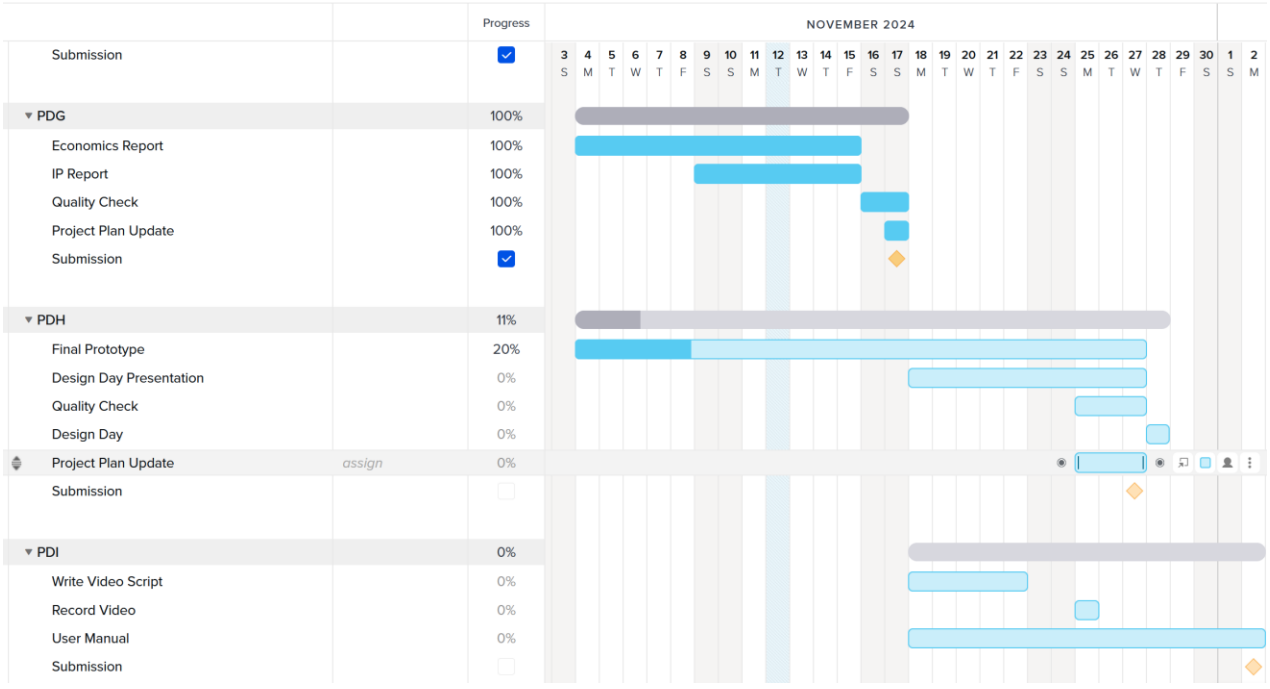
This design outlines a device by which a wrist-mounted barcode scanner is used to manage inventory. This design many elements that are similar to ours, such as the product ID/code, the location, date, when it was signed out, as well as general data regarding the inventory item. In the process of scanning, the device can determine the location of the inventory item. This is done through requesting the data from the server, and if authorised to, receive a response detailing the inventory data.

Although the system operates very similarly to our design, it's intended use case is in large warehouse/ department stores in which large amounts of items move in and out frequently. Our design is stationary, and involves the assets being present at the scanning station, as opposed to the opposite. Additionally, the patent is only in use in the US and Great Britain, for Walmart and ASDA respectively. As such, it isn't applicable to our product.

Furthermore, the design varies from ours in that a camera is used to scan barcodes, as opposed to the optical laser that our design incorporates. The patent also makes use of touch-screen capabilities on the scanning device, as well as a guiding function that gives feedback to the user on what to do. Both features are not present in our design. In sum, this patent, while similar, should pose no legal restraints on our product.

4.3 Project plan update

Add a screenshot of your gantt chart.



5 Design Day Pitch and Final Prototype Evaluation

Hi folks!

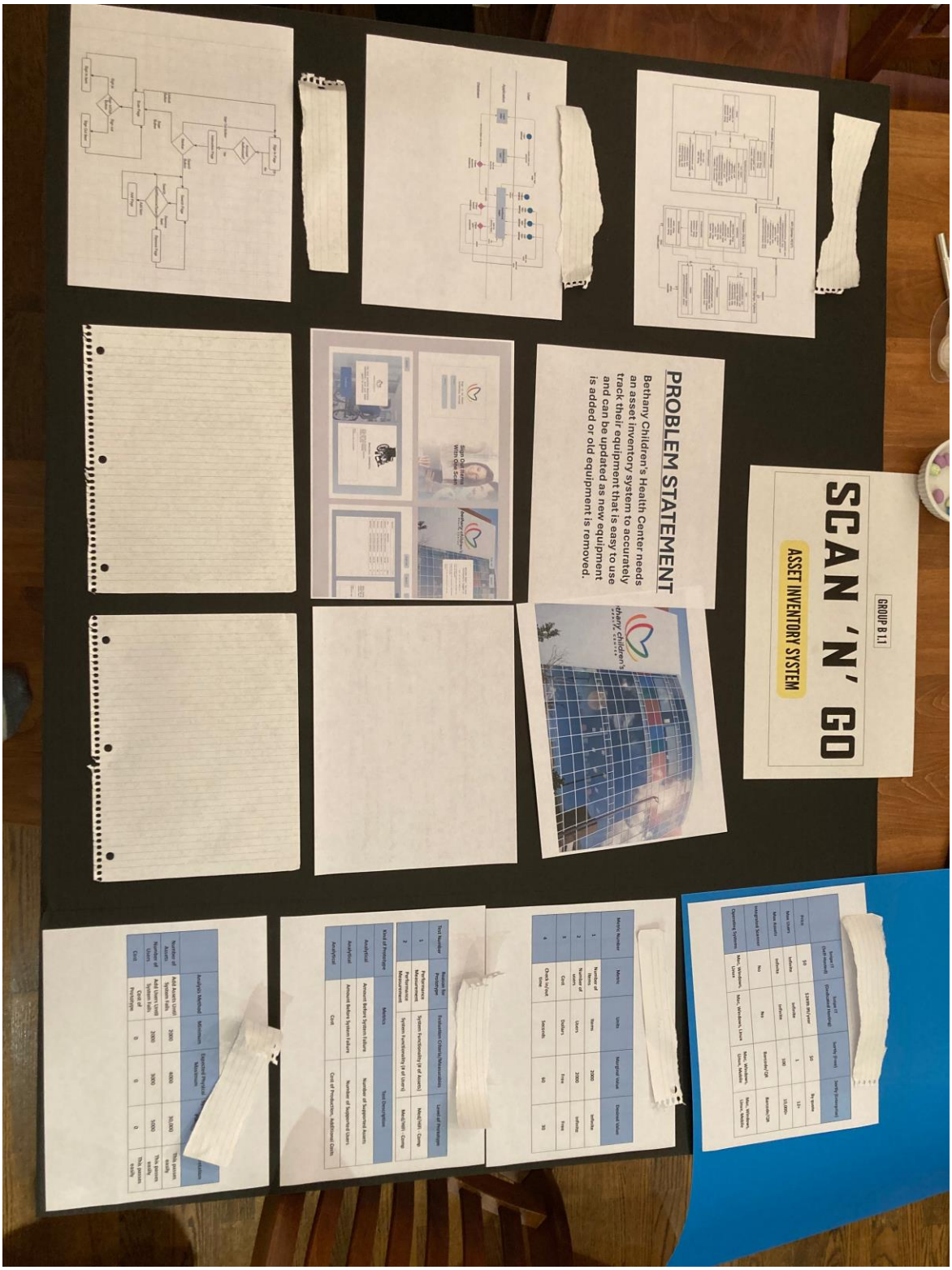
We're Scan-N-Go, and this is our design. The problem set before us is that the Bethany Children's Health Center in Bethany, Oklahoma needs an Asset Inventory System to effectively manage their equipment. Their current system is rather slow and ineffective, and we've set out to change that. This is important because of the rapid turnover between patients in the day. For example, using their current system, it can take up to 5 minutes to sign out an asset for a physical therapist to use in an appointment. Given that the typical appointment slot is 30 minutes, that's 1/6 of the appointment that a child doesn't receive care for. Add this up over an 8-hour workday, and that's over two and a half appointments that are lost due to asset sign in/out time. What we've done is reduce the average sign in time to 10 seconds, increasing asset uptime drastically.

In order to make use of our design, the therapists need to:

- Select sign in/out
- Scan their personal barcode identifier to log in
- Scan the desired asset
- Scan their personal barcode again to assign/unassign an asset

And then they're done. This is important as it allows the physical therapists to see up to 2 more patients per day or see them for longer without having to stress about dealing with the equipment they need for a given patient.

Now, why are we better than current solutions on the market? Well for one, we're free. By comparison, their current system costs at least $2,500 USD, and more likely around $5000 USD. We're faster, with sign in/out being only 10 seconds as opposed to 5 minutes. This is due to two key factors in our design; simplicity in the scanning process and being locally hosted. Our scanning process is only 4 steps long, all of which require very little action from the user. Their current system requires the physical therapists to use iPads to scan a QR code, manually type their login to a website, then sign out the desired asset to themselves. The problem there is that frequently the website can take several minutes to respond to the user's request. This is where being locally hosted comes in. We don't need to request information from some far-off cloud server to sign out an asset, it's all locally hosted so the process is much faster. Additionally, the same slow process needs to be repeated to check equipment back in, adding even more time. Now, a live demo of our design in action:



Preliminary version of poster board for design day:

Link to github: https://github.com/Jamie-Watson/GNG2101-AssetInventory

6 Video and User Manual

6.1 Video pitch

Add link to video.

6.2 User manual

See separate template for the user manual. Do not write the content here.

7 Conclusions

Summarize your lessons learned and your work related to your project. Discuss any outstanding issues or implications for the project.

8 Bibliography

Insert your list of references here.