Instructor: Rakeshkumar Shukla
Course: CSCI 3432-01E
07 April 2021

Final Document
Inventory Management System

Team DB01E 1
Shane Casiano - Email: sc10196@georgiasouthern.edu - Eagle ID: 901014912
Joseph Dodd - Email: jd11123@georgiasouthern.edu, Eagle ID: 901001055
Duncan Hedden - Email: dh21003@georgiasouthern.edu, Eagle ID: 901250343
Dakarai Holcomb - Email: dh06770@georgiasouthern.edu, Eagle ID: 900976512
Jade Spahr- Email: Js21914@georgiasouthern.edu; Eagle ID: 901007178

Table of Contents

1. Achievements

The original plan for this project has remained mostly intact. Create an inventory management system with the capability to expand to eventually also hold a website where direct sales can be made to a possible customer. Unfortunately, while that part has remained mostly intact, gone is the premise of user differentiation. It was too much work in one small period of time to complete an entire application that contained the functionality we hoped for, the most complex of these features was user access control. There are two ways this could have been managed, either through users within the DBMS, or with a more specific user table. Choosing the latter also meant we would need to implement all of the access control for said users, and the host of different views you could receive depending on your particular access level. While this is one feature that got cut, every other intended feature made it into our application. This is mainly the managing of currently available inventory in comparison with incoming and outgoing inventory, as well as a user group to differentiate users from employees, and a simple budget system with the room to grow to the needs of any company using our software. This program was intended to be a framework, and modular enough that portions of it can be reused throughout the warehouse and sales industry, while being repurposed to the specific needs of the user group. This is why there are many tables within the IMS DB that appear to not be used to their full potential. This is because they are not, instead portions are implemented to simplify any future work that could be done on this program. Over time, this program has changed to more closely suit the needs of a possible employer for such a database solution, and while some of these requirements were more difficult to meet than others, we feel we have come as close as possible to the initial directives set forth for this application.

2. Quality Assessment

Overall, we had very little quality pitfalls within the application. Once you put into perspective that this entire application was planned, and coded in a matter of only two weeks, you begin to understand how exactly this program came together. While we have expanded majorly from the original wireframes and other directive diagrams, we were able to make a program that is an excellent indicator of our ability to merge and utilize databases for programs, while simultaneously simulating and meeting the possible needs of a company who could have hired us for this program. The biggest quality issue then is due to a lack of testing. Testing is especially difficult when a database can have hundreds, if not thousands of combinations of queries for only a single table. This multiplied by all of the tables we have and writing test cases for each would take even longer than actually creating the program. Thus, it was overwhelmingly decided to make the program as

useful as possible, without the major concern of testing. This means that certain issues exists, such as logging in to a database, which was never quite fully tested, and while fixes, like deleting a metadata file, are small, they can get inconvenient, and would need to be repaired for a more robust release version of the Inventory Management System.

3. Shane Casiano's Experiences

Which parts were the most fun?

The integration steps, and seeing a project that fully functions were probably the most relieving and rewarding, and the most fun to see.

Which parts were the most challenging?

The initial start, for me personally, of the code for integration of the database. It was very difficult to assume/guess the needs of our program. This gets better as the program is closer to implementation, but the initial steps are difficult.

How did you solve those challenges?

The simplest way to solve this was simply having more discussions regarding the directions of our project

Which parts were the easiest?

The simplest part, though most tedious, was creating the database.

What did you learn that you did not imagine you would have?

I learned how to package JavaFX within a .jar file, you would think this is a rather simple thing to do, but as of Java 8, the FX:Builder has not exists, as such, you need to create a new class to call your ingress function (main), and only then can you package JavaFX, as it has a calling function. No, you're not supposed to do this, but it works.

If you had to do it all over again, what would you have done differently?

More than likely I would have begun earlier, the problem with this was that myself, and my team members lacked the necessary information to move forward. Time management is usually not an issue, but it's a very difficult thing to ask to complete an entire database without the knowledge on how to do so, and to the point everyone did feel comfortable enough, we would've had about half a week to create the database.

What is your overall experience of working in a team?

My overall experience is positive, we tended to work quite well together due to the fact that we all had strong time management skills, so regardless of the pitfalls we ran into, solutions were often swift.

Final comments:

This project led me to question whether or not databases are truly the most useful method for data storage, sure, the processing on a database is quite excellent, but

for many implementations, I feel as though it just isn't the most useful. Regardless, we were lucky to find one of the few implementations that greatly benefitted from a database. Working with my team members, I've been more lucky than usual, and I am extremely grateful to them for all of the input they've had throughout the semester.

4. Joseph Dodd's Experiences

Which parts were the most fun?

The part where we actually coded the project was the most fun.

Which parts were the most challenging?

Coordinating back-end parts of the project with the front-end was difficult simply because we had different people working on both.

How did you solve those challenges?

Good communication and discussions during team meetings helped us solve that challenge.

Which parts were the easiest?

The SQL parts were relatively easy.

What did you learn that you did not imagine you would have?

I didn't imagine I would pick up on SQL that easily.

If you had to do it all over again, what would you have done differently?

I would've learned SQL sooner so I could've worked on the project a lot earlier, leaving more time for troubleshooting and fine-tuning the project. I also would've refreshed my knowledge of JavaFX so I could've helped with the GUI side of the project as well as the SQL side.

What is your overall experience of working in a team?

I had a very positive experience working with this specific team. Everybody was on top of their part, and were communicative every step of the way. Assignments were handled in a timely manner as well.

5. Duncan Hedden's Experiences

Which parts were the most fun?

I had the most fun getting to work with such a great group of people who all brought a great deal to the table. Being able to learn from them all over the past few weeks this semester was definitely the highlight of this assignment.

Which parts were the most challenging?

The most challenging portion of this project was definitely dealing with JavaFX and its communication with the database.

How did you solve those challenges?

Time and communication with my teammates. I was fortunate enough to have been in a group of people who all put in the time and effort to really understand exactly what it is that needed to be done and how exactly to accomplish this. Other than that, just taking the time to understand how everything interacts with one another and a lot of research.

Which parts were the easiest?

The easiest part was the building of the UI in the JavaFX. It has a very intuitive design system.

What did you learn that you did not imagine you would have?

I never truly understood just how much time and resources goes into the maintenance and creation of a server. Moving forward I will definitely look for alternative options, especially since I was very limited by being unable to run this assignment on my own personal laptop.

If you had to do it all over again, what would you have done differently?

I would have started earlier on the project. I underestimated just how intricate this assignment was and could have begun research and laying foundations earlier.

What is your overall experience of working in a team?

My experience was very positive. Everyone was always on top of their weekly/semester goals and were always willing to help in their own time. I am very lucky to have been able to work with this group.

Final comments:

Overall, I am happy to have had this experience, but moving forward I believe I will look for alternatives for data storage.

6. Dakarai Holcomb's Experiences

Which parts were the most fun?

Implementing the front end UI and seeing the whole project come together and work.

Which parts were the most challenging?

Figuring out the direction of our project and just how we all wanted it to be. Initialing starting the project was the hardest part and once we all figured out just how we wanted things, implementing our ideas became much easier.

How did you solve those challenges?

A weekly meeting where we discussed how we each wanted things till we were all happy with the ideas presented.

Which parts were the easiest?

Using scene builder inorder to set up some of the GUI.

What did you learn that you did not imagine you would have?

How to access the mySQL server.

If you had to do it all over again, what would you have done differently?

Freshened up my JavaFX earlier than I did.

What is your overall experience of working in a team?

My overall experience was positive. All of my team members worked well together and were able to help each other when one had an issue.

Final comments:

Overall this project was an experience that I'm glad I've had. While parts of it were tedious and or frustrating, I believe that what I have learned while working on this project will come in handy in the future.

7. Jade Spahr's Experiences

Which parts were the most fun?

Initially setting up the JavaFX was definitely the most fun as the code practically wrote itself.

Which parts were the most challenging?

The initialization of the project as it took the longest since making all the parts of the GUI connect and work properly was quite tedious.

How did you solve those challenges?

I set aside time to work with my team and be able to do research to find reasonable solutions.

Which parts were the easiest?

Initially setting up the JavaFX and playing with the functions as they started working.

What did you learn that you did not imagine you would have?

JavaFX has similar functionality to WPF, in the sense that FXML and controller files are very similar to XAML and XAML.cs files. I also learned some SQL statements and the steps on how to set up a database.

If you had to do it all over again, what would you have done differently?

Given myself more time as well as not going between two different languages as I also had an OOD project.

What is your overall experience of working in a team?

My overall experience with my team was positive. Coordination was key while working with everyone which affected the time management throughout the project. This coordination is something we could have improved upon.

Final comments:

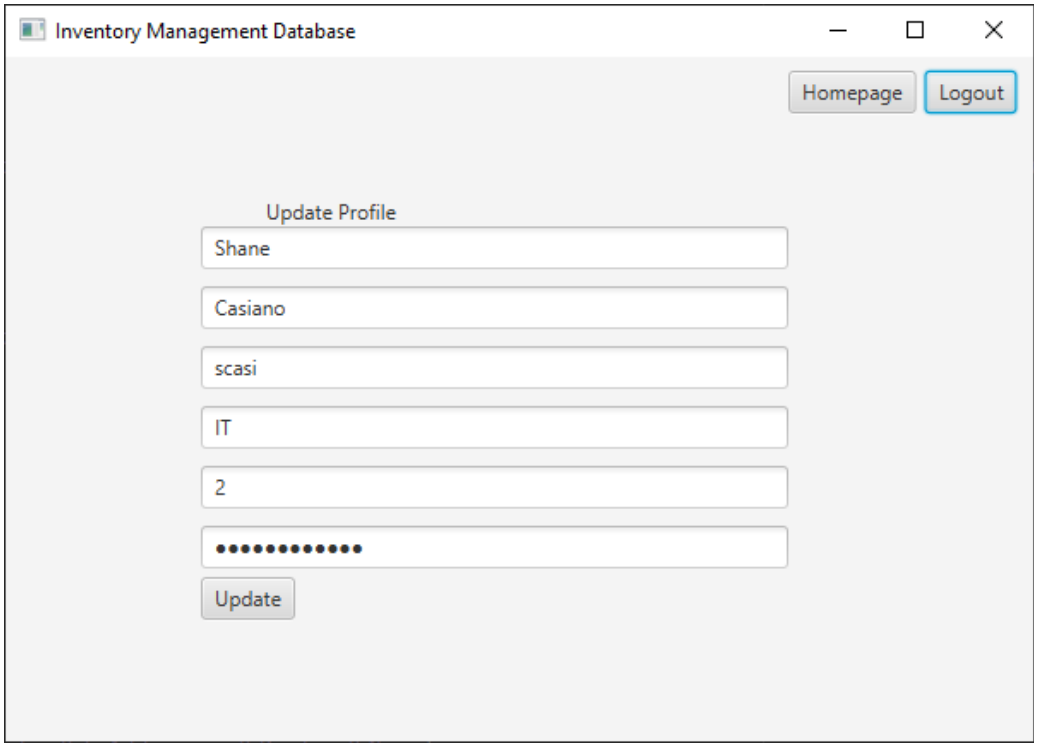
We need more time to learn SQL.

8. Future Expandability

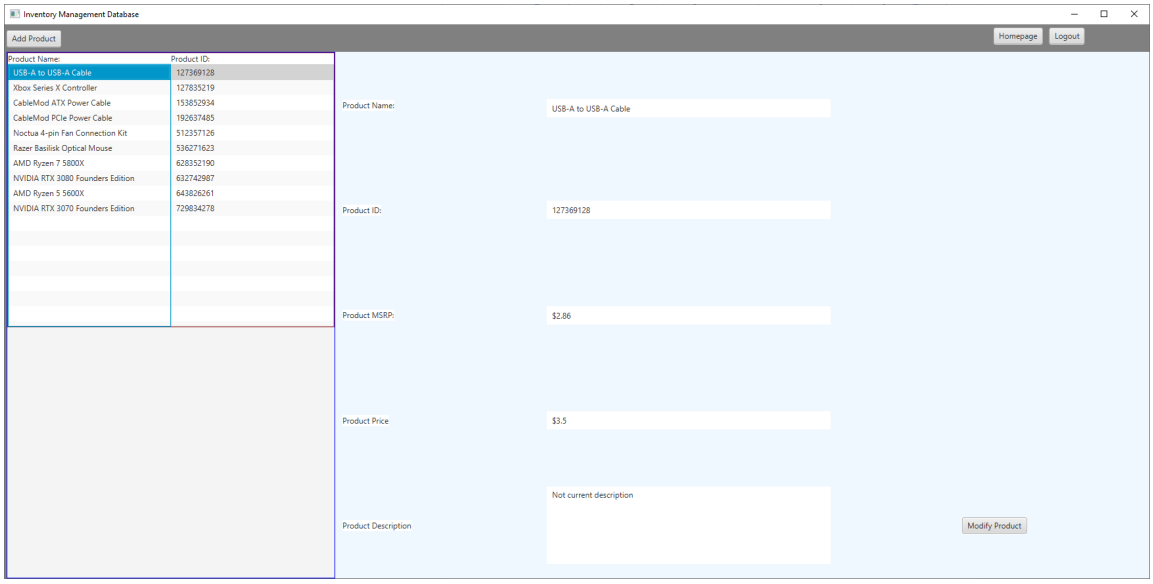
One of the major points of expandability for this project is an accompanying website. This costs money, and will likely never be implemented unless we were hired and paid to be able to eat the cost of hosting the site and database. Along with this major feature that could be built in, several smaller features could also be included, such as a more robust database, specifically splitting it into several databases, to show separation. Adding images to products, and matching descriptions would also be a good feature to add. Furthermore, more tightly controlled access to the program, and likewise, full implementation of employee access would create almost six months worth of work alone. These features would take our program from a simple proof of concept, to a fully fledged inventory management system that can be rapidly deployed around the world to businesses that may require such a program. More than likely though, these programs already exist, and instead, we will take the knowledge of creating this program to better understand the design choices that these companies have made, in the pursuit of improving already implemented solutions, rather than building entirely new ones that may cause more issues than it is worth.

9. Views / Forms

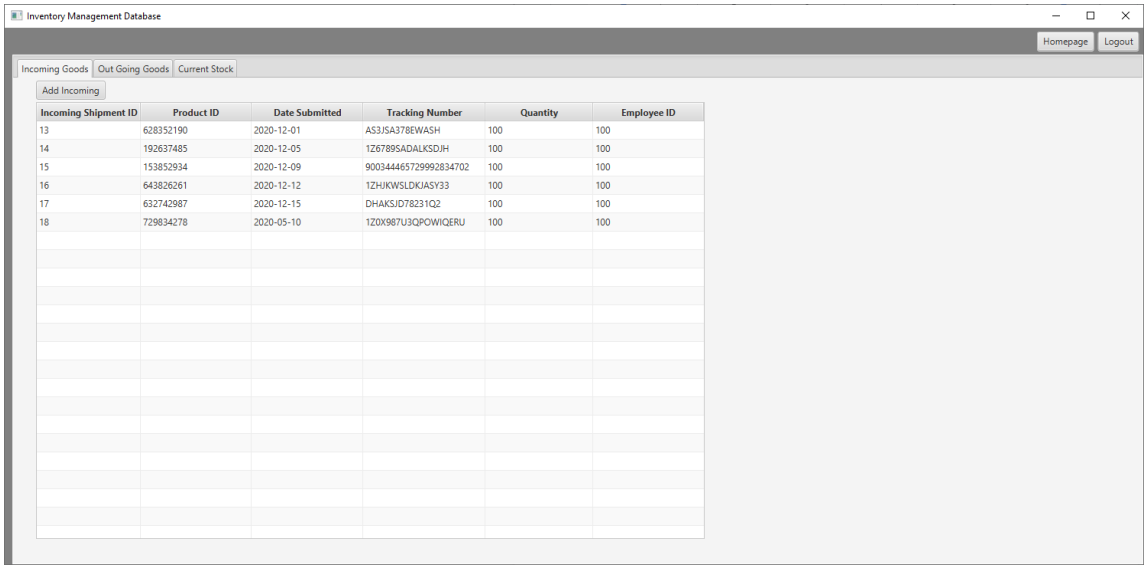
Profile information: A page containing the users current information, as well as methods to modify said information, this information comes from the users table.



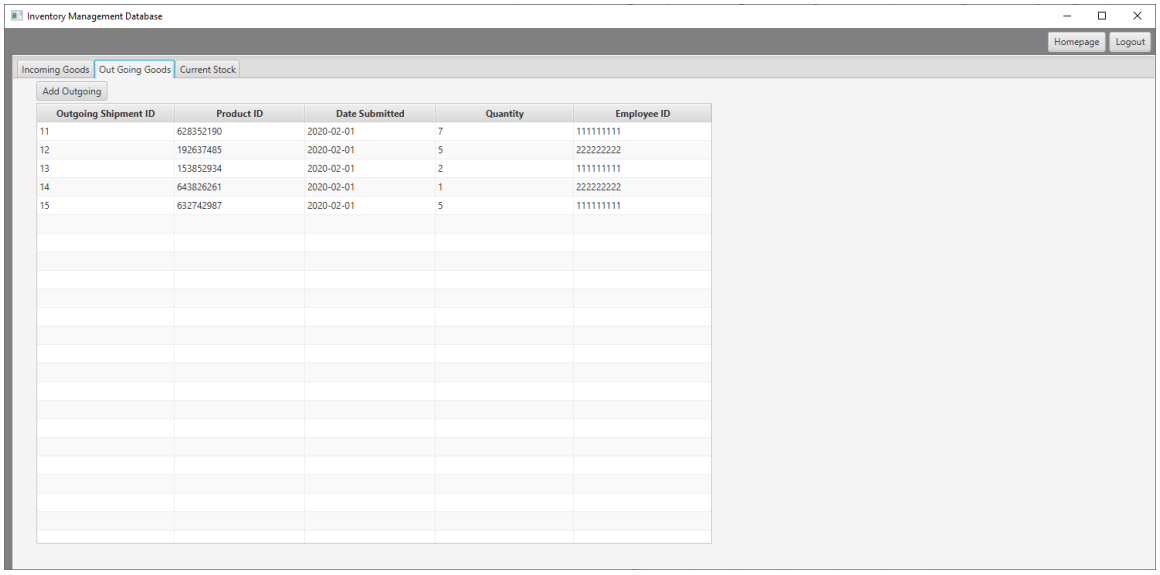
Products: A page containing the current information in the products table, as well as methods to modify said information. This information comes from the products table.



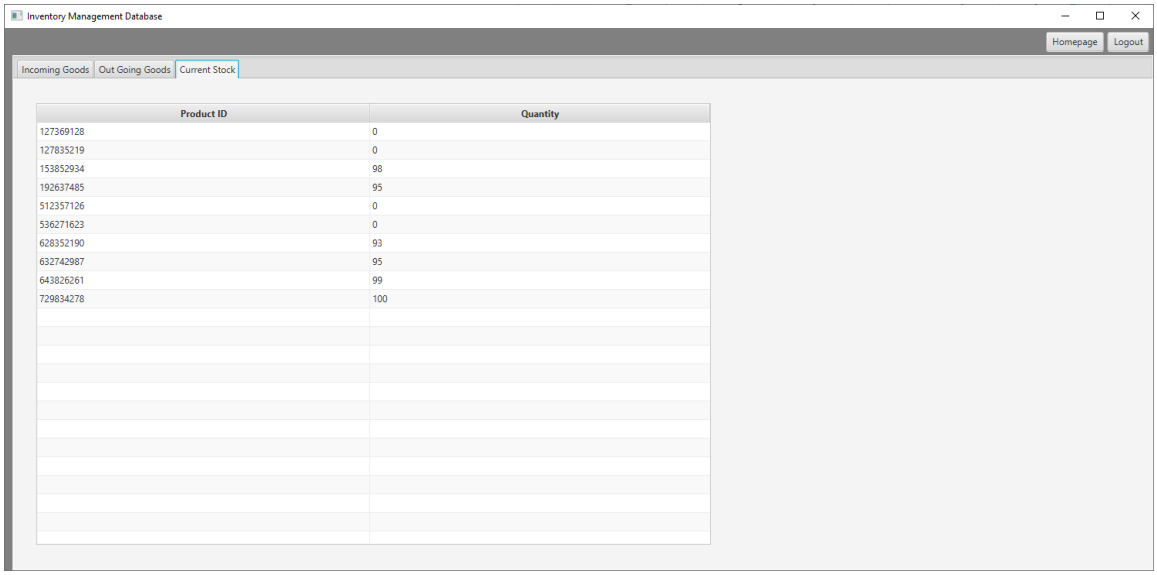
Product-Status: Incoming Goods: A page containing information from the incoming_goods table, it also includes methods to modify and delete the information.



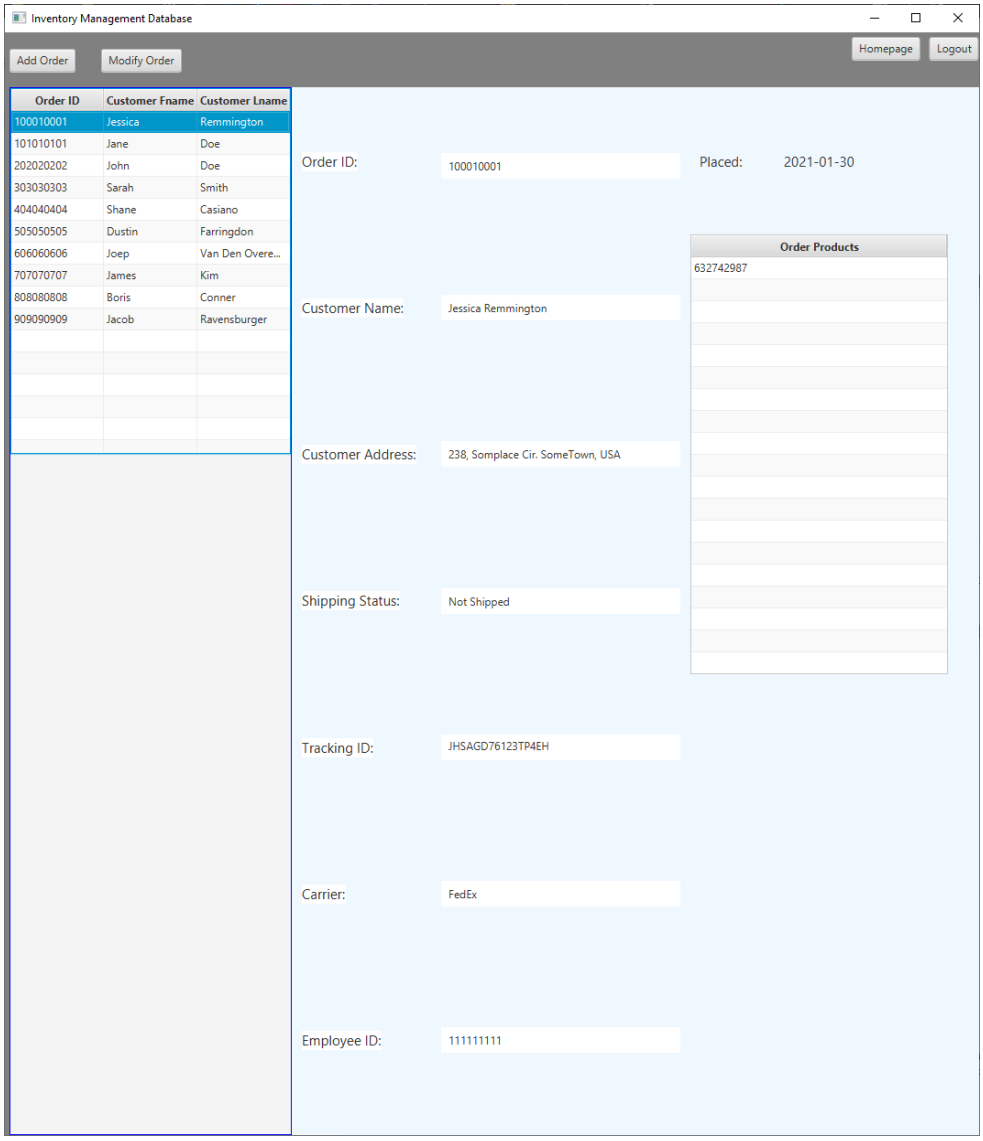
Product Status: Outgoing Goods: A page containing information from the outgoing_goods table, it also includes methods to modify and delete the information.



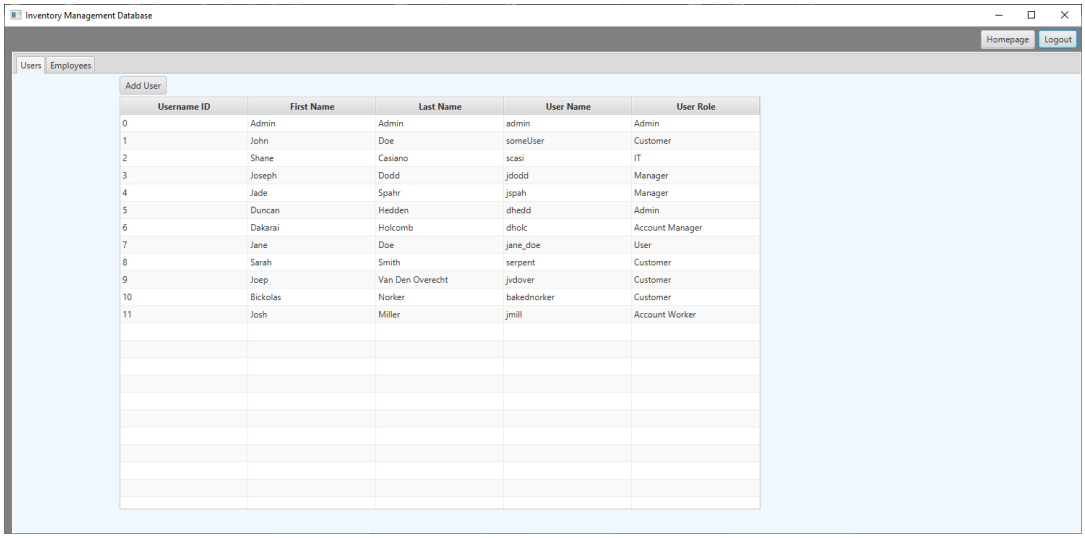
Product Status: Current Stock: A report indicating the current products that are in stock. Information here is generated from incoming and outgoing goods.



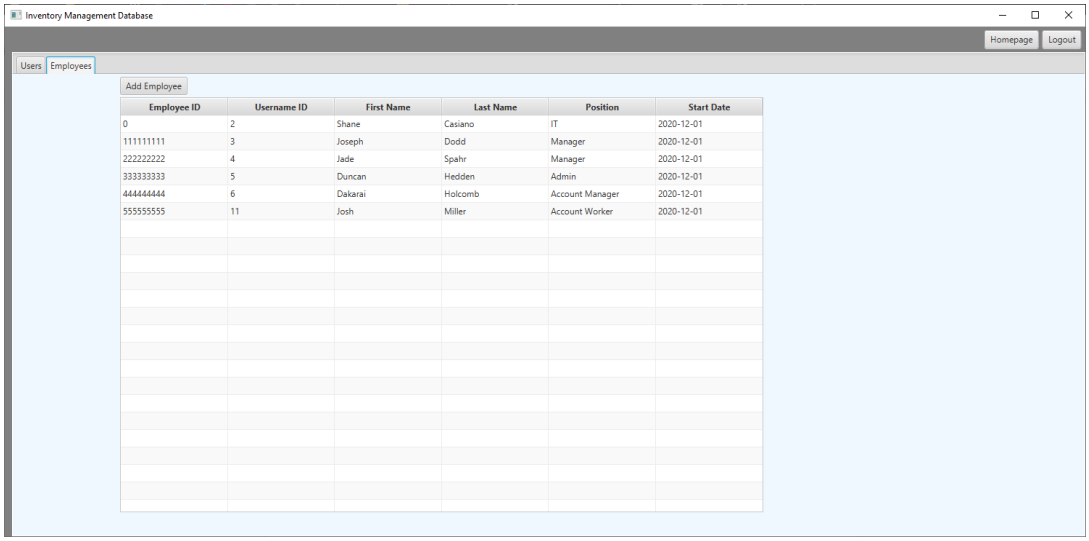
Orders: A page containing information directly related to an order, some of this information is modifiable. Information is retrieved from multiple tables, including the tracking table, orders, and order_items.



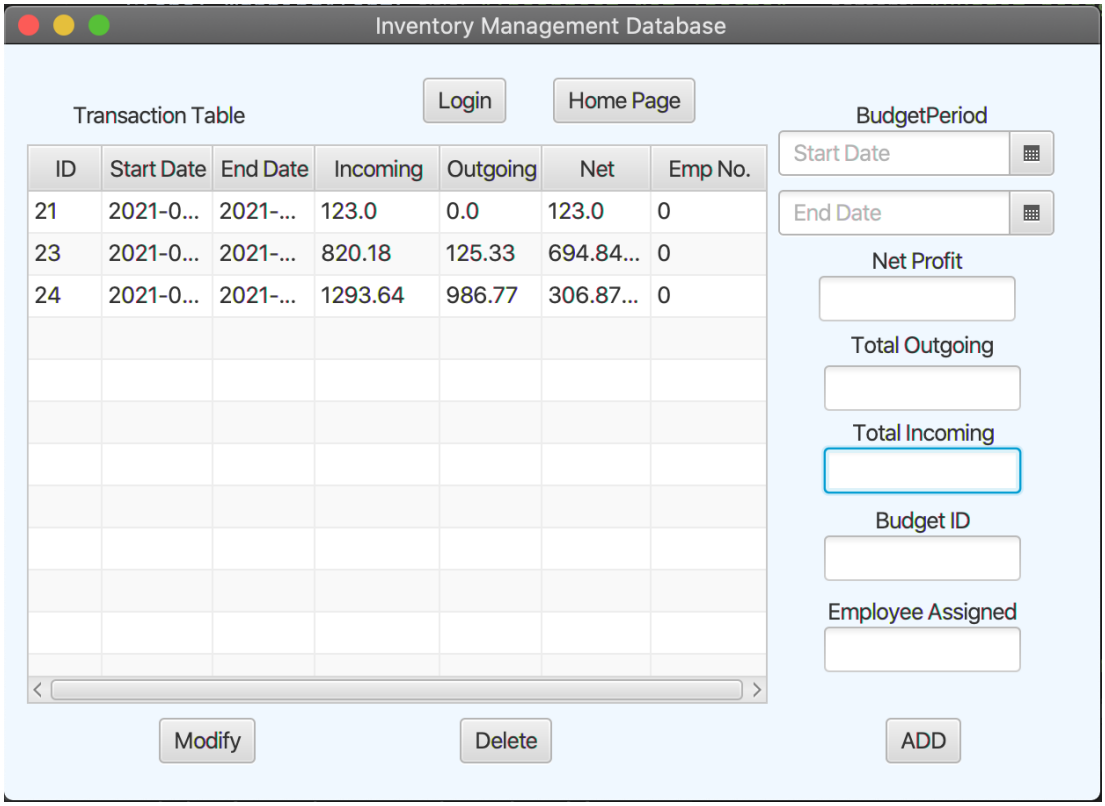
Users: A page containing all users in the database, ideally for an admin to change user information, or create a new user



Employees: A page containing all employees in the database, ideally for an admin to change employee information, or create a new employee.



Budget: This view contains information pertaining to current and previous budgets, which are manually inserted based on invoice and spending information at the company level.


Inventory Management Database
Transaction Table
Login
Home Page
BudgetPeriod
Start Date
End Date
Net Profit
Total Outgoing
Total Incoming
Budget ID
Employee Assigned
ID
Start Date
End Date
Incoming
Outgoing
Net
Emp No.
21
2021-0...
2021-...
123.0
0.0
123.0
0
23
2021-0...
2021-...
820.18
125.33
694.84...
0
24
2021-0...
2021-...
1293.64
986.77
306.87...
0
Modify
Delete
ADD