**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



| **Capstone Proposal Project Name:** | Wave Solutions Warehouse Tracking Software |
| --- | --- |
| **Student Name:** | Jaime Vargas |

**Table of Contents**

BUSINESS PROBLEM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

The Customer. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

Business Case . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

Fulfillment. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

EXISTING GAPS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

SDLC METHODOLOGY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

DELIVERABLES. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Project Deliverables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

Product Deliverables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

IMPLEMENTATION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

VALIDATION AND VERIFICATION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

ENVIRONMENT AND COSTS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

Programming Environment. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

Environment Costs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

Human Resource Requirements. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

PROJECT TIMELINE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

**Business Problem**

**The Customer**

Wave Solutions is a mechanical manufacturing and distribution company. Specializing in tools, automotive, engine, and machinery hardware, Wave Solutions has been operating for over 100 years. Founded in 1903 by Aaron Phillips and Mark Bolton. The company initially started with only one Facility in Milan, Illinois. Since then, it has expanded across the midwest to states including Missouri, Iowa, Wisconsin, Minnesota, and Nebraska. Currently, there are 23 facilities in operation, with plans of opening many more. This project will focus on the warehouse aspect of this company, specifically the warehouse inventory and order tracking. The company is doing very well and has plans to expand even further. To keep up with higher demand, the current software system must be updated. While there is an inventory tracking system in place, its functions aren’t enough to track the relationships between new types of products being developed. The company has been keeping track of orders manually with pen and paper. The same goes for inventory locations. Order picking times have been far behind expected goals, and warehouse workers have been stressed trying to keep up with demands. There is a large backlog of inventory orders. After investigating bottlenecks, Wave Solutions has decided that an improved inventory management system is needed.

**Business Case**

With almost all inventory management being done with paper, keeping track of current inventory has been very difficult. Multiple processes are being done simultaneously. Trucks are unloaded, and their freight is verified and counted. Then warehouse workers take each part and place them in warehouse shelf locations. A separate team of warehouse workers focuses on order picking. They travel throughout the warehouse, searching shelves for items to fulfill their order. There are also warehouse transfers that need to be loaded and tracked. Each of these processes is manually tracked. When finished, their papers are submitted to shift supervisors to enter into the current system. A large bottleneck occurs at the end of shift times when supervisors are all entering their data. An hour per shift per day is spent just by entering these sheets into the system. There have also been a large number of inaccuracies being entered. This is caused because of the unreliability of manual tracking and by human error from the supervisors trying to rush the end of their shift. With the current database being so simple, inventory adjustments still have to be done one by one. There is also no tracking system for inventory editing. If something is changed, it is impossible to find who did it. The updated system will address all these issues by adding many more functionalities relevant to these inventory processes. It will be a scalable and quicker application-based software system.

**Fulfillment**

The new inventory management system will also come with an updated user interface. The goal of the new interface is to be more appealing as well as easier to understand. More people are going to be expected to be using the application, so understandability will be important. The application will be using an embedded database called SQLite for inventory tracking. Upon loading, the first page shown will be a log-in page. Every user of the application will be required to log in. This is done to ensure secure use of the system as well as to track interactions. After entering an authorized username and password, the user will be brought to the main page. Here there will be seven buttons, each leading to its own section.

The first button will lead to the bulk adding page. This is where many lines of inventory items could be added all at once. The intended purpose of this page is for when the warehouse is unloading large trucks but could be used for other situations where large quantities of inventory additions are needed. There will be a main table that displays the items that are going to be added, along with their quantities. Then there will be entries for related item data that will be needed to be filled out by the user. Entries will auto-fill if the ID is recognized. Buttons will be used to add, edit, and delete from the table. Upon clicking submit, the information in the table will be added to the database.

The second button leads to the storage page. Here the user can edit the storage location of inventory. Similar to the bulk add page, there will be a table and buttons to add, edit, and delete. This table will be auto-filled with information once a storage location is selected. Users could also move the selected items to a different location on this page.

The third button leads to the order-picking page. On load, the page will search the database for orders assigned to the current user. Assigned orders will then be displayed. The user will be able to select an order, and the inventory items to be picked will be displayed. After the item is picked, the user can mark the item as completed. If the item is unable to be picked, it could be marked as incomplete. Then once all items are marked, the user can click submit. This will then remove the completed items from the inventory in the database.

The fourth button leads to the bulk remove page. This page will be similar to the bulk add page but used for removing instead. It will be used for removing multiple items from the inventory at once. The intended use of this function is for warehouse transfers. Although, it can be used any time many items are needed to be removed.

The fifth button leads to the search page. Here there will be multiple ways to search the database. The user could search by items, parts, storage location, and order. A table will be auto-filled with the results of the search.

The sixth button will lead to the manager functions page. This button will only be visible to users who are signed in as managers. There are three main functionalities to the management functions. First, managers will be able to edit users. This includes creating, editing, and deleting users and their passwords. The second function will be order management. This is where orders are created, edited, or deleted. Items will be added to the orders, and users will be assigned to pick the orders. Lastly, managers will be able to generate, view, and print different types of reports for the database.

**Existing Gaps**

The current system has many deficiencies in functionality. There is no ability to add or remove many items at once. Items must be added one at a time, which is why there is such a large bottleneck at the end of shifts. There aren’t any storage locations shown. Workers must either memorize where items are typically stored or look them up in a logbook. This is especially hard for new employees. Each new employee needs to be taught where the warehouse items are located. It takes extra time to memorize these locations, which slows down productivity. It is especially hard on order pickers because they are expected to meet picking goals. There is no report generation. To get an idea of the current inventory, managers would need to either skim through log books or walk around the warehouse and look in person. The current search function is only able to search by part or by product. Adding more search options could be useful for finding various inventory information. There is no current order tracking functionality. With multiple orders being done at once, tracking a specific order would be troublesome. One would need to find the worker that the order is assigned to and ask about the status. Finding a completed order would need to be done by sifting through a stack of order sheets. The current system doesn’t have any users or required log-ins. Not having a log-in system leaves the database vulnerable to being accessed by anyone. Not having any user tracking causes inventory edits to be untraceable.

**SDLC Methodology**

The software development lifecycle methodology selected for this project will be the waterfall methodology because the requirements are clear and unlikely to change. The Waterfall method is a linear development method where each stage is fully completed before moving on to the next. It is used for projects that are clearly defined from the start. Wave Solutions has provided a clear understanding of the problem, and business requirements are not expected to change throughout the development of the new application. Specific needs for the inventory management system could be talked out with the stakeholder.

The first phase of the waterfall method is called the **Requirements Phase**. This is where all requirements for the project are gathered. This could be done in a variety of ways such as interviews, brainstorming, and analysis. Use case analysis can help clear up expectations for system behavior. Developing user stories clarifies the needs of the project from the user’s perspective. A clear understanding of what the project should accomplish is necessary before moving on to the next phase.

The next phase is the **Design Phase** where a technical solution is laid out. Using the established requirements, the system begins being designed. Data models, layouts, and diagrams are created to display the logic of the application. Finer details such as programming language, hardware, database relationships, and user interface design are planned out.

The **Implementation Phase** is next. This is where the actual construction of the product begins. Code is being written, databases are being deployed, and user interfaces are being created. All the information from the previous stage is turned into a usable product.

After implementation is the **Testing Phase**. Once all the coding is finished, the application is ready to be tested. Program tests are created by the programmers and sent to the testers to find any bugs, errors, exceptions, or errors in logic. If any of them are found, they are sent to the programmers to be fixed. This repeats until all known flaws are fixed.

After that is the **Deployment Phase**. The product is complete at this phase. This is where the product is delivered to the customer. The environment is set up, and the program is loaded onto machines. Users are shown how to use the application. If old software is being replaced, then this is where the transition to the new software is executed.

The final phase is the **Maintenance Phase**. At this phase, the product is fully delivered and used by the client. As issues arise, the product is patched. Updates are also sent during this phase. This phase could last for as long as the product exists, or be a predetermined amount of time before support is ended.

**Deliverables**

The two types of deliverables created using the Waterfall method are project and product deliverables. Project deliverables are related to the development process of the project. Project deliverables are given and managed by the project manager. Product deliverables consist of actual software that can be given to the customer. Some examples of these types of deliverables are given in the next section.

**Project Deliverables**

* Scope Statement
* Describes what the project is planning to develop as well as what is not included in development
* Shows the customer’s expectations for the product and its expected uses
* Defines project deliverables
* Shows the constraints of the project such as cost, time, and resources
* Requirements Specification
* Defines the purpose of the product
* Details the functional and non-functional requirements of the software application
* Specifies performance requirements required by the customer such as response time and efficiency.
* Anything the software is expected to accomplish by the customer and agreed upon by the development team.
* Development Schedule
* Shows the expected time needed for phases of development, milestones, and goals. Created through estimations and analysis of requirements.

Product Deliverables

* An interactive prototype that allows the customer to navigate through the application
* Graphical user interface that is user-friendly and aligns with the company’s image
* Functioning deployed database that interfaces with the application for data storage
* Fully functioning application delivered to the customer aligned with the requirements document
* Secure username verification that only allows verified users into the system

**Implementation**

The updated application will be replacing the old version, so it will need to be deployed when the current one is not being used. Given that Wave Solutions does not operate on the weekends, that is when our development team will deploy the new application. Total deployment time is expected to be very short, and easily done within the two off days. Luckily the current system is very simple. Current data in the system needs to be transferred to the new database. This will be done by loading scripts into workstations. The updated application is using the same platform, so changing the application will be as easy as pushing an update to workstations. Validation and verification will be conducted after implementation.

As for hardware, 5 new workstations are to be set up for the use of the application. Since more users are expected to be interacting with the system, this will prevent bottlenecking. Since the computers are expected to do simple tasks such as document creation, database interactions, and report generation, lower-end PCs will be used. Workstations are expected to be placed throughout the warehouse, so Wi-Fi connectivity needs to be expanded. More routers and Wi-Fi extenders are going to be needed. Adding hardware will not interfere with business processes, so it will be implemented whenever the machines are ready to be set up.

The new application will have an abundance of new functionality as well as a new look. This may confuse users. To have a smooth transition, there will be a thorough user training session one day before full deployment. All processes done with the application will be explained step by step. Special attention will be given to those who are more computer illiterate.

**Validation and Verification**

A testing plan will be created to validate and verify all aspects of the application. It will be checked to see if it aligns with the requirements document. It will also be checked by the customer to verify that it does what is expected. Multiple types of tests will be conducted, including unit testing, integration testing, system testing, and acceptance testing. More details for each of these are further provided below.

Unit tests are written to test the smallest parts of a system. Usually functions or methods. The system will have its smaller parts isolated and individually tested to see if they complete their intended function properly. Something as simple as verifying log-in information could be unit tested. Each component of the application will be checked this way.

After unit testing is integration testing. The units are then tested to see if they interact with each other properly. The purpose of this is to expose defects in the interaction of modules. Is data being sent properly between modules? Are modules receiving and loading the data that is sent? Verify that all units are communicating with each other properly.

System testing is done after unit testing. This is where the entire system is tested by the quality assurance team. The team will test the system’s interaction with other systems, its processing efficiency, and its functionality. They will also go through every possible data input to check for desired outputs. Only correct data types for entries should be enforced programmatically. Lastly, QA will evaluate the user experience of the system. Systems should have an organized and efficient logical flow to them that is easy to understand by the user.

The last type of testing is acceptance testing. This is where the application is set up in a similar environment to where it will be deployed, and end users use it to verify that it does what is expected of it. The business logic flow is the main thing being tested at this point. Does the application fulfill its desired business needs? Any feedback given by the user testers will be taken into consideration.

During any testing phase, if the product doesn’t pass its test, it will be revised and edited. This process will be repeated until it can pass its tests.

**Environments and Costs**

**Programming Environment**

Since the product is going to be an updated version of an existing application, it will be created using the same language and framework. It will be developed using C# code and the Windows Forms .NET framework. Windows Forms has many features such as controls, graphics, and data binding. It will be developed using the Visual Studio 2019 Integrated Development Environment. The database engine used will be SQLite. It is an embedded database library, so it will be hosted in the application itself. There is no need for a separate database management system or database administrator.

**Environment Costs**

The operational costs of the system will be very low. There is no need to pay for database hosting since it will be a locally embedded database system. There is also no need for application hosting because the application will be installed on each workstation. However, more workstations and networking hardware need to be purchased to create a wireless local area network. For each location of the company (currently at 23), the following will be required to purchase:

* Wi-Fi router $150 x2
* Wi-Fi extender $50 x3
* HP Elite Tower 600 G9 $800 x5
* HP 125 Wired Keyboard $15 x5
* HP 128 Laser Wired Mouse $26 x5
* HP V223ve FHD Monitor $110 x5

This equates to an estimated cost of...

(150\*2 + 50\*3 + 800\*5 + 15\*5 + 26\*5 + 110\*5) = $5,205 per facility OR $119,715 company-wide.

**Human Resource Requirements**

There are 3 main contributors to the cost of developing a software project. The first is the type of software being developed. This project is considered a software modification since it is going to be upgrading software that already exists. Second, is the size of the project. This project isn’t very large but isn’t very small either. Third is the number of people required for the project and how much they are being paid. This project will require a project manager, business analyst, designer, two developers, and a QA tester. The project manager will only be working part-time on this project since they have other projects to work with, so they are only clocking in 40 hours over the four weeks needed for development. The business analyst will be working the least of all other workers because they are only needed during the requirements-gathering phase. The two developers are working 130 hours each throughout the 4 weeks needed for development. They are actively working through most of the SDLC. The QA tester is only needed during the testing phase so they are estimated to work only 30 hours. Lastly, the designer is only needed during the design & analysis phases so they are estimated to get 40 hours.

| **Role** | **Wage \* Hours** | **Total** |
| --- | --- | --- |
| **Project Manager** | $50 \* 40 | $2,000 |
| **Business Analyst** | $45 \* 20 | $900 |
| **Developers** | ($85 \* 130) \* 2 | $22,100 |
| **QA Tester** | $50 \* 30 | $1,500 |
| **Designer** | $45 \*40 | $1,800 |
| **Grand Total** | | $28,300 |

**Project Timeline**

| **Phase** | **Milestone/Task** | **Deliverable** | **Description** | **Dates** |
| --- | --- | --- | --- | --- |
| **Pre-development** | Setup development environment, research API | Product backlog | Prepare for development of the product | 07/01/23 - 07/02/23 |
| **Planning** | Create project schedule | Project schedule | Plans out timing for tasks, deliverables, milestones. Used to track the overall progress of the project | 07/03/23 - 07/05/23 |
| **Planning** | Crate work breakdown structure (WBS) | WBS Diagram | A Hierarchical document that breaks work into smaller tasks. Shows the relationships and dependency of tasks. Includes scope, cost, and schedule baselines. | 07/06/23 - 07/08/23 |
| **Analysis** | Create Software Requirement Specification Document (SRS) | SRS Document | Details the agreed-upon requirements that will be included for the development of the project. Includes business, functional, and nonfunctional requirements. | 07/09/23 - 07/13/23 |
| **Design** | Design the user interface | Graphical User Interface | Design the digital interface that users will use to interact with the software system | 07/14/23 - 07/16/23 |
| **Design** | Database Design | Entity Relationship Diagram, Use case diagram | Database logic and flow is displayed through Unified Modeling Language Diagrams | 07/16/23 - 07/18/23 |
| **Implementation** | Fully construct the application | Final pre-tested version of the software | Referencing documents from previous phases, developers will begin and finish coding the software system. The system will align with agreed upon requirements. | 07/19/23 - 07/26/23 |
| **Testing** | Develop tests, test the software, fix any errors | Product successfully passes all tests | Developers will create multiple tests for the system. They will be sent to quality assurance who will then run the tests. If the system fails, then developers must update the system until all tests are passed. | 07/26/23 - 07/28/23 |
| **Deployment** | Install new hardware environment | Fully functioning hardware environment that is to be used with upcoming software | Installation of routers, Wi-Fi extenders, and workstations. (Can be done before software is finished) | 07/15/23 - 07/16/23 |
| **Deployment** | Upload software to machines, move current data to new database system | Customer accepts the software and verifies that it accomplishes what is expected | Full transition from previous software to the new one. Check that workstations are communicating with the database correctly. | 07/29/23 - 07/30/23 |
| **Maintenance** | Develop maintenance plan | Maintenance service contract | Reach a maintenance service level agreement with the customer. | 07/30/23 -  TBD |

**C868 – Software Capstone Project Summary**

**Task 2 – Section C**

| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Wave Solutions Database |
| --- | --- |
| **Student Name:** | Jaime Vargas |

DESIGN DOCUMENTS …………………………………………………………………21

Class Design ………………………………………………………………………….21

UI Design ……………………………………………………………………………..22

Login Page ……………………………………………………………………………22

Main Menu Page ……………………………………………………………………..24

Bulk Add Page ………………………………………………………………………..25

Storage Page …………………………………………………………………………26

orders Page …………………………………………………………………………..28

Bulk Remove Page ……………………………………………………………………30

Search Page ………………………………………………………………………….31

Manager Functions Page …………………………………………………………….33

Edit Users Page ………………………………………………………………………34

Edit Orders Page ……………………………………………………………………..36

Reports Page ………………………………………………………………………….37

Add User Page ………………………………………………………………………..39

Add Order Page ………………………………………………………………………40

Edit Inventory Page …………………………………………………………………..41

Add Inventory Page …………………………………………………………………..43

UNIT TEST PLAN

Purpose ………………………………………………………………………………..44

Overview ……………………………………………………………………………….44

Add Inventory Page ……………………………………………………………..45

Add Order Page …………………………………………………………………46

Add User Page …………………………………………………………………..47

Bulk Add Page …………………………………………………………………...47

Bulk Remove Page ………………………………………………………………48

Edit Inventory Page ……………………………………………………………..49

Edit Order Page ………………………………………………………………….50

Edit Users Page …………………………………………………………………51

Login Page ………………………………………………………………………52

Main Menu Mage ………………………………………………………………..52

Manager Functions Page ………………………………………………………53

Orders Page …………………………………………………………………….54

Reports Page ……………………………………………………………………55

Search Page ……………………………………………………………………55

Storage Page ……………………………………………………………………56

Test Plan……………………………………………………………………………….57

Items ……………………………………………………………………………...57

Features ………………………………………………………………………….57

Deliverables ………………………………………………………………………57

Tasks ……………………………………………………………………………..57

Needs …………………………………………………………………………….58

Pass/Fail Criteria ………………………………………………………………..58

Results ……………………………………………………………………………58

Source Code ………………………………………………………………………….59

User guide …………………………………………………………………………….59

Introduction ………………………………………………………………………59

Installing the Application ………………………………………………………...59

Login ……………………………………………………………………………...60

Add Stock to Inventory ………………………………………………………….60

Remove Stock From Inventory ………………………………………………….61

Move Storage Location of Inventory Item ……………………………………...61

View Items in Storage Location ………………………………………………...62

Search for Inventory Item ……………………………………………………….62

Search for Order ………………………………………………………………..62

View Assigned and Current Orders …………………………………………….63

Set Current Order ……………………………………………………………….63

Mark Current Order as Complete ………………………………………………63

Edit Existing Order ………………………………………………………………63

Remove Existing User …………………………………………………………...64

Add New User ……………………………………………………………………64

Edit Existing Order ………………………………………………………………65

Remove Existing Order ………………………………………………………….65

Add New Order …………………………………………………………………..65

Edit Existing Inventory Item ……………………………………………………..66

Remove Existing Inventory Item ………………………………………………...67

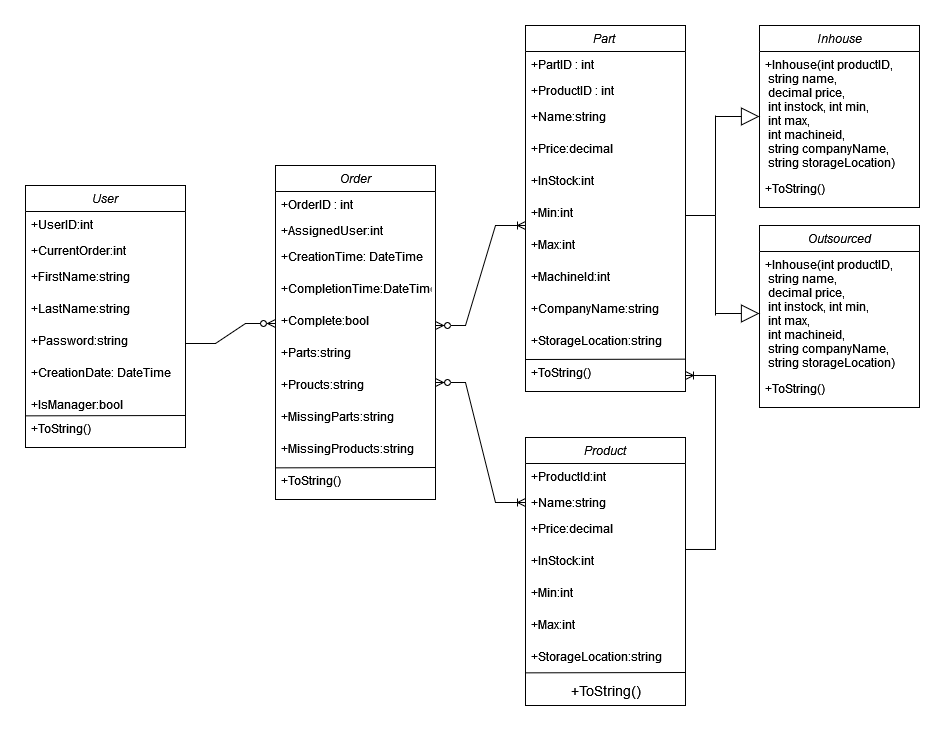
Add New Inventory Item …………………………………………………………67

View and Print Reports ………………………………………………………….67

# Application Design and Testing

# Design Documents

## Class Design

The main part of the database is documented in the following UML database diagram. These six classes are the backbone of all functionality of the database. This Entity Relationship Diagram (ERD) shows how the classes are related to one-another. The User class represents one user of the database. One user is able to have zero to many orders assigned to them at one time. Each order must have at least one Part or Product assigned to it. The Inhouse and Outsourced classes are Part classes that inherit all part data. An inhouse part is still a part, the same goes for an outsourced part. One product has one or many related parts.

A database service class is used to interface with these classes. It uses 33 methods to modify data. Buttons in the user interface utilize these methods. The database class is called DatabaseService. Methods created in it consist of adding, searching, getting, removing, updating, and adjusting data. This class uses sqlite syntax to interact with the database. Since the database is loaded into this class, it’s referenced by all other classes. For organization, all database methods are grouped together here.

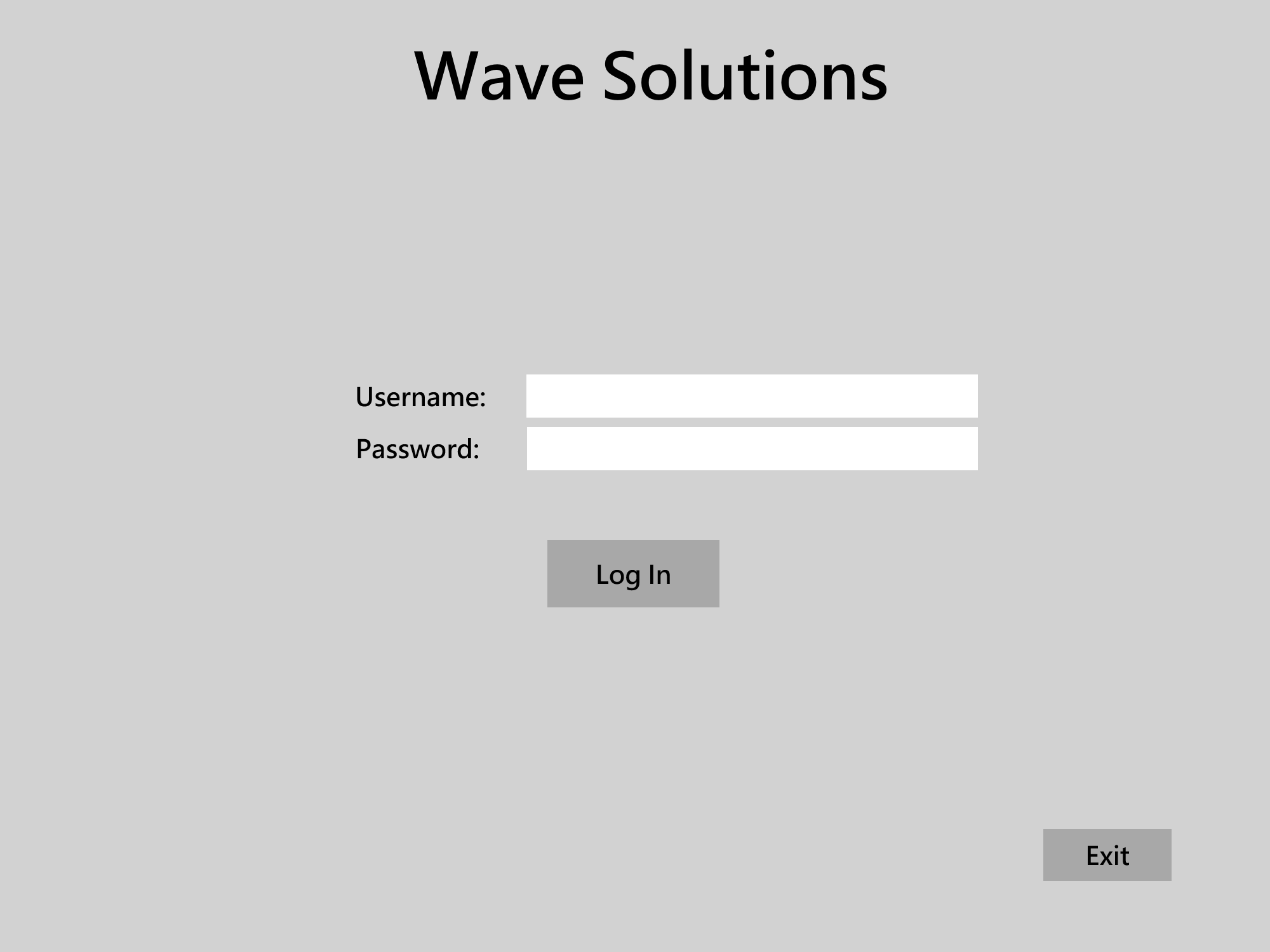
## Fifteen related pages are created from these six classes. These pages are AddInventory, AddOrder, AddUser, BulkAdd, BulkRemove, EditInventory, EditOrder, EditUser, Login, Main, Manager, Orders, Reports, Search, And Storage. The design of these pages will be shown in the next section.

## UI Design

Here the 15 pages will be shown in more detail. Their relations to one-another and their functionality is explored with low and high fidelity wireframes.

**Log-in Page**

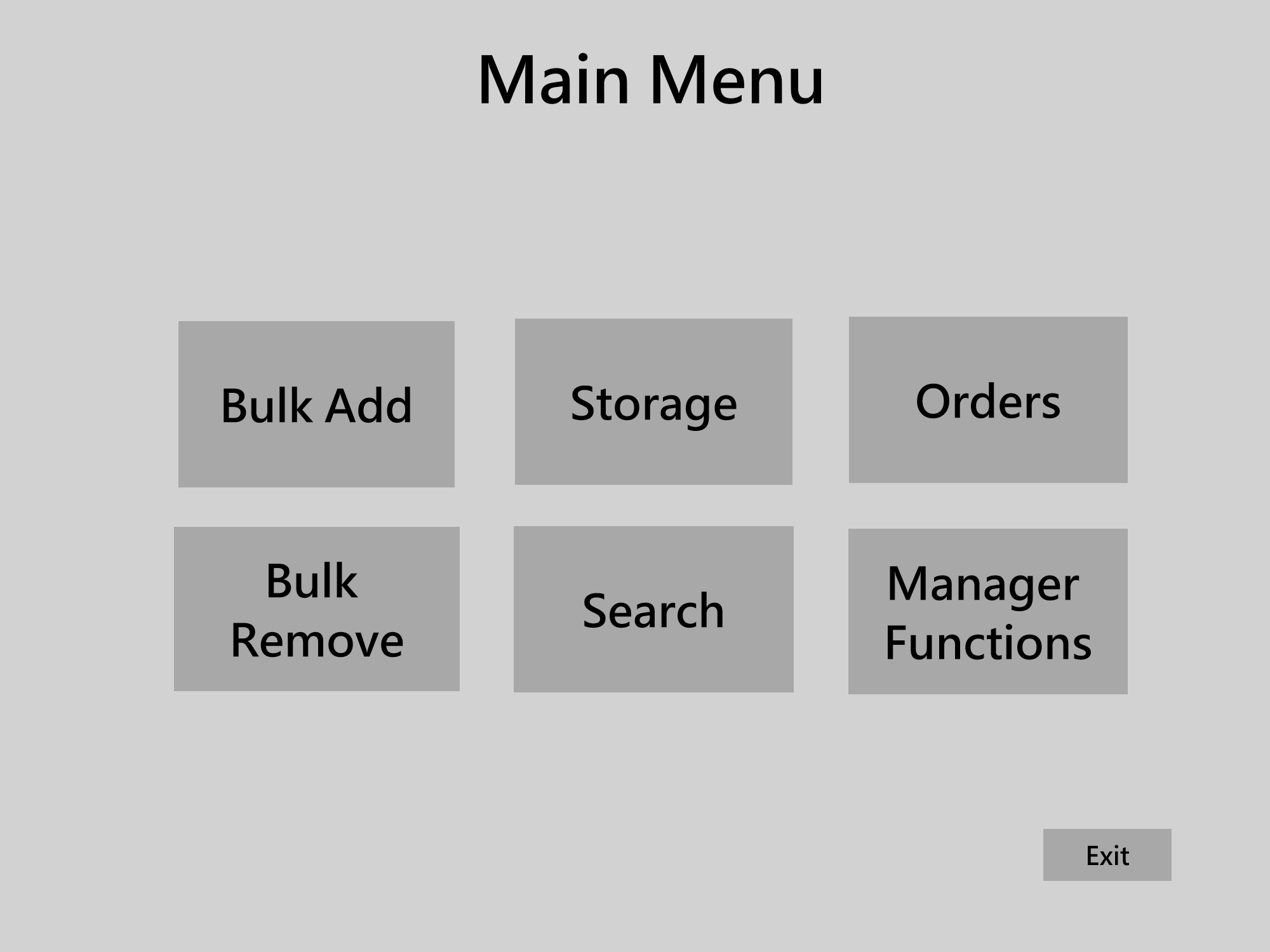
Upon running the program, the user login page will be displayed. An existing and correctly matching username(UserID) and password must be entered in order to access the system. Clicking the “Exit” button on the Login page will close the program. Pressing the “Exit” button on any other page will bring the user back to the Login page.

*Low Fidelity Login Page Wireframe*

*High Fidelity Login Page Wireframe*

**Main Menu Page**

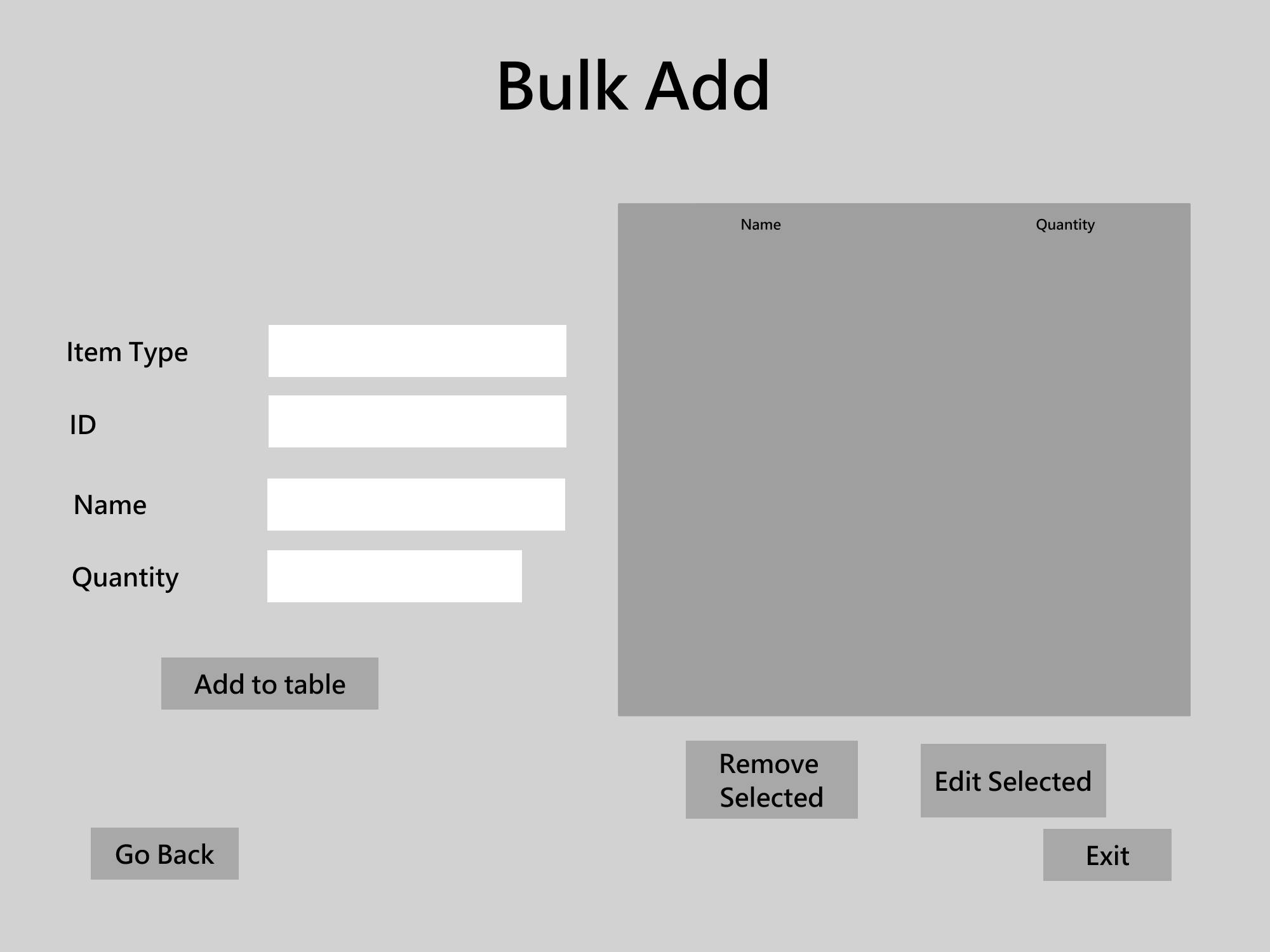
After correctly entering an existing username and password combination and clicking the “Log In” button, the Main Menu page appears. Here the user is presented with 6 buttons. They are Bulk Add, Storage, Orders, Bulk Remove, Search, and Manager Functions. The “Manager Functions” button is only visible if the logged-in user is marked as a manager. Moving forward, all pages that have a “Go Back” button will take the user to the previous page visited.

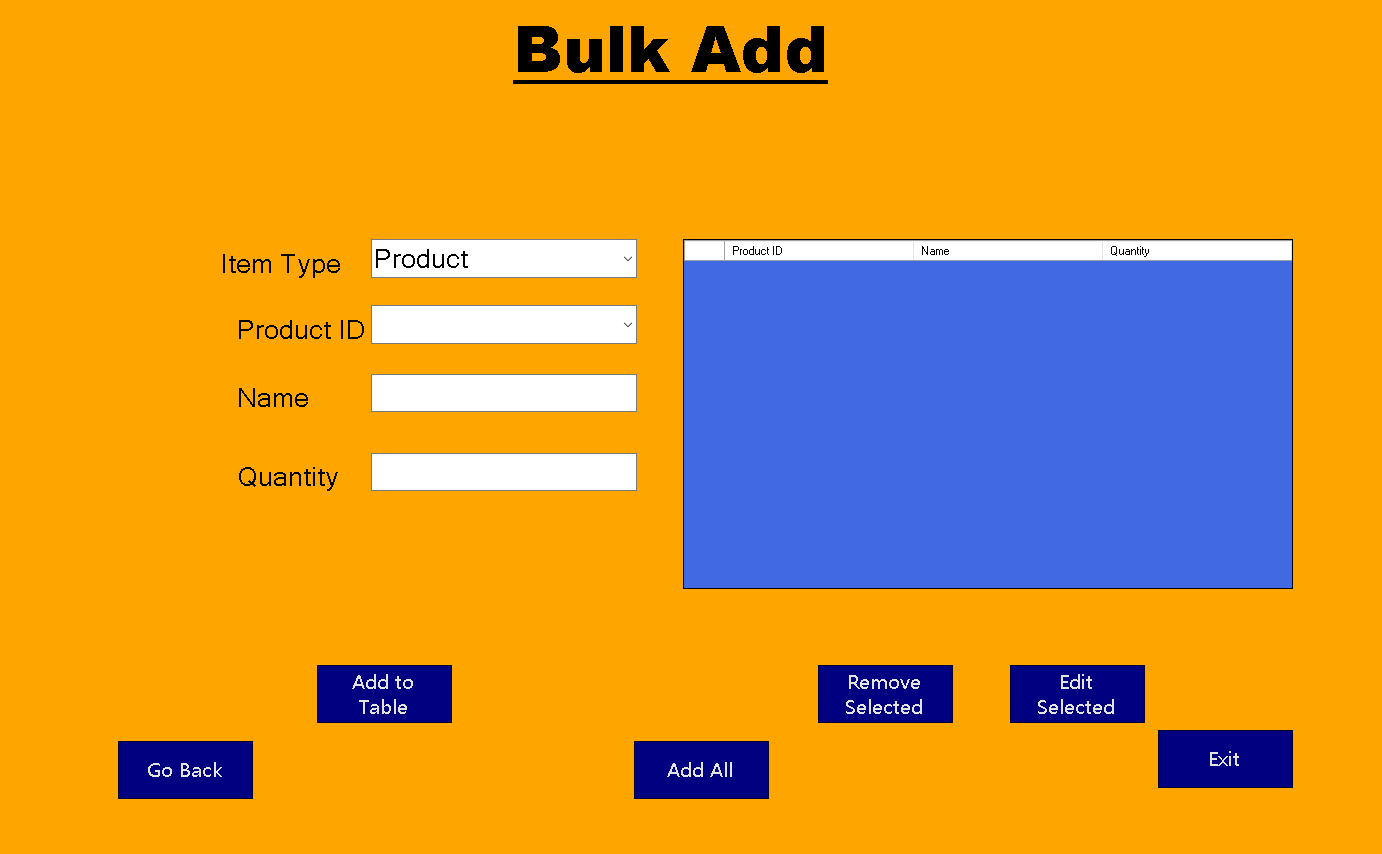
*Low Fidelity Main Menu Page Wireframe*

*High Fidelity Main Menu Page Wireframe*

**Bulk Add Page**

After clicking the “Bulk Add” button from the Main Menu Page, the user is taken to the Bulk Add page. This is where multiple different items could be added to stock in the database. Item type consists of products and parts. Selecting part from the Item Type dropbox fills the ID dropbox with part IDs. Selecting product from the Item Type dropbox fills the ID dropbox with product IDs. After selecting an ID, the name of that item is autofilled into the Name textbox. Then the user enters a number as the quantity and clicks the “Add to Table” button. This will add a new row to the data table consisting of the ID, Name, and quantity. Clicking the “Remove Selected” button will remove the selected row from the table. Clicking the “Edit Selected” button will populate the entries with the selected row’s data. Clicking the “Add All” button will add stock to all the items in the database that correspond to the items in the data table.

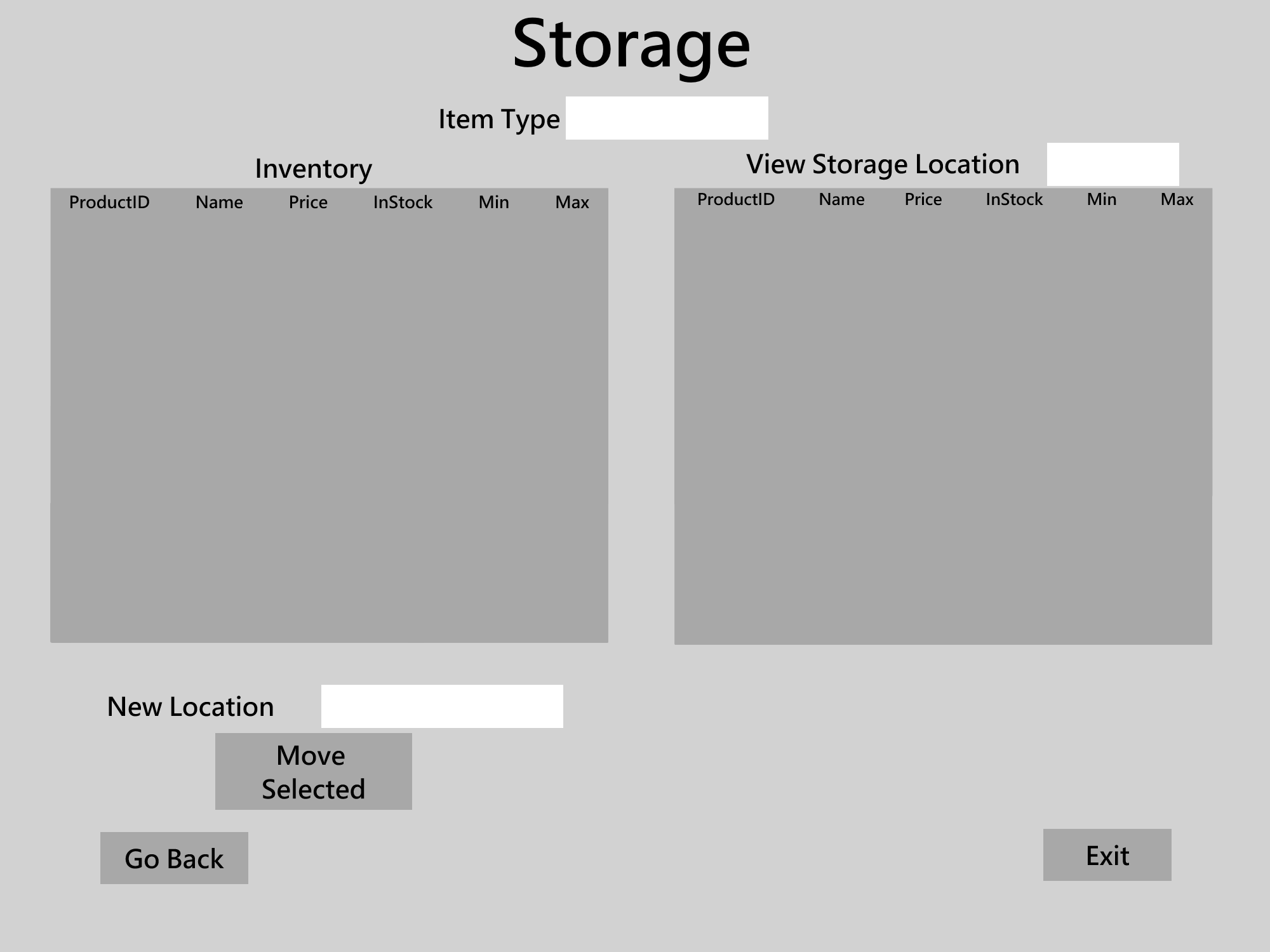
*Low Fidelity Bulk Add Page Wireframe-/*

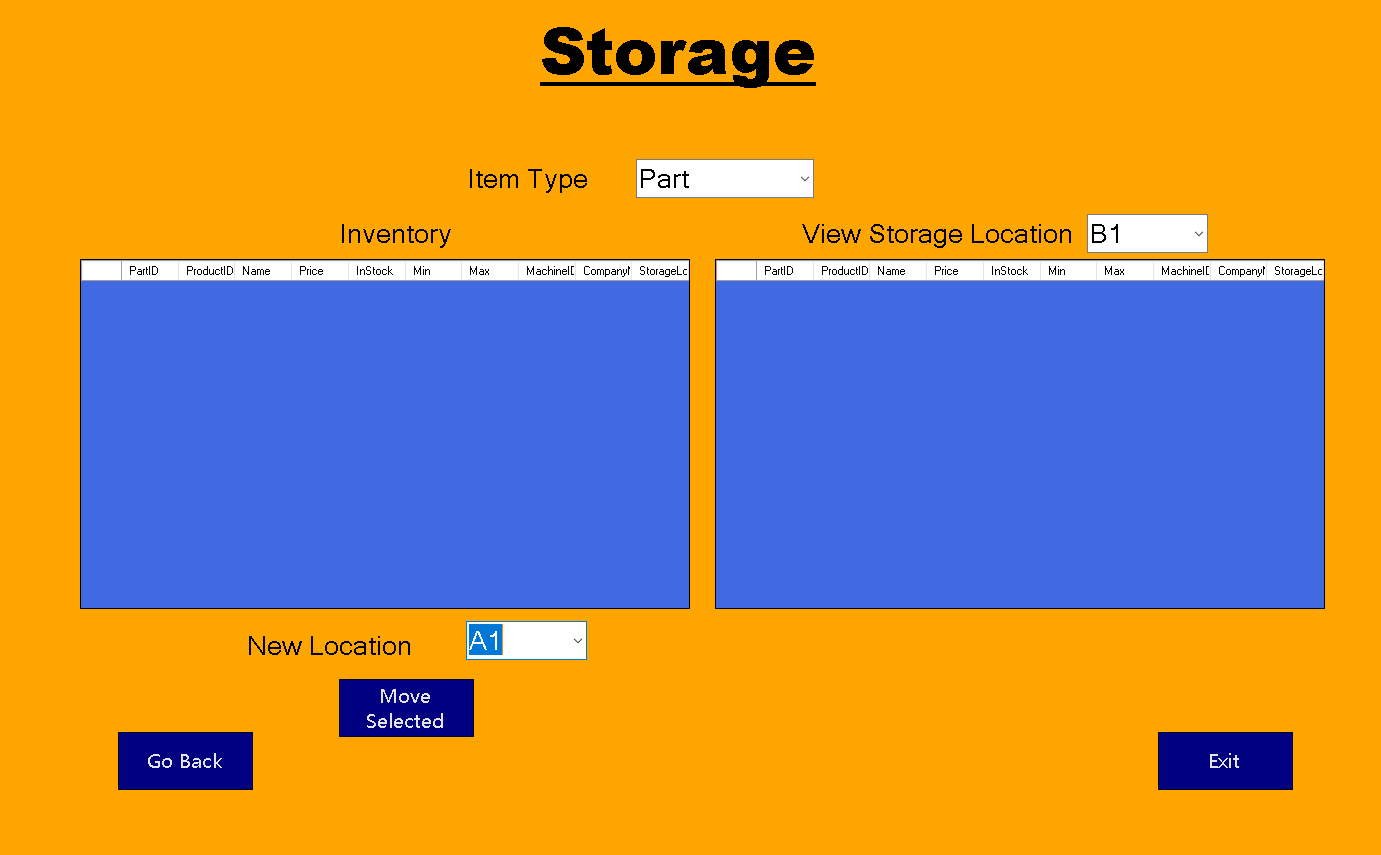
*High Fidelity Bulk Add Page Wireframe*

**Storage Page**

After clicking the “Storage” button from the Main Menu Page, the storage page appears. Here there are two data tables. The one on the left is used to display the inventory of the current selected item type. The one on the right is used to view items by selected storage location. Upon selecting a row in the left data table, the user can use the “New Location” drop box to select a new location then click the “Move Selected” button to change the storage location of the selected item. Selecting from the “View Storage Location” drop box changes what is shown in the data table on the right.

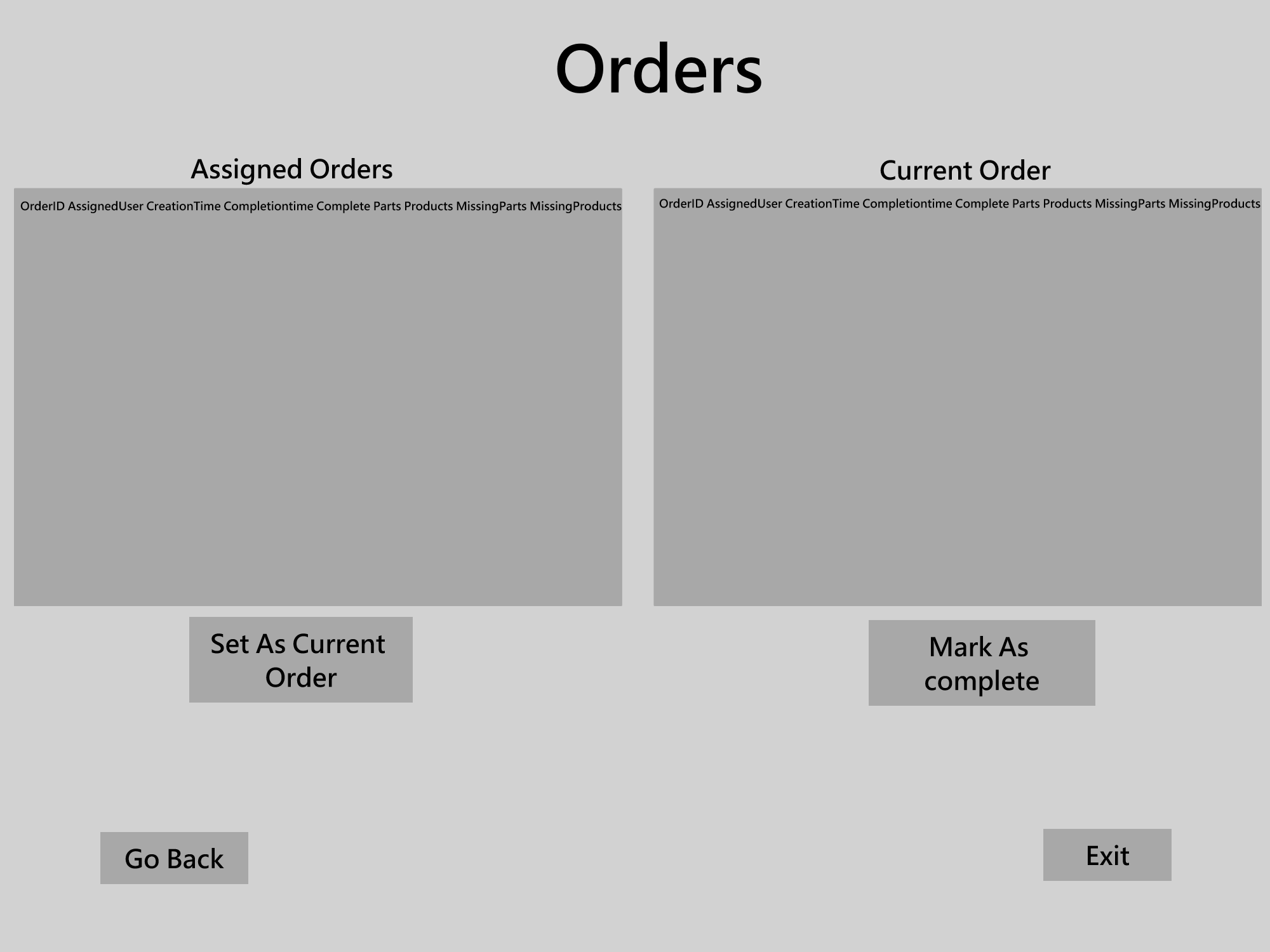
*Low Fidelity Storage Page Wireframe*



*High Fidelity Storage Page Wireframe*

**Orders Page**

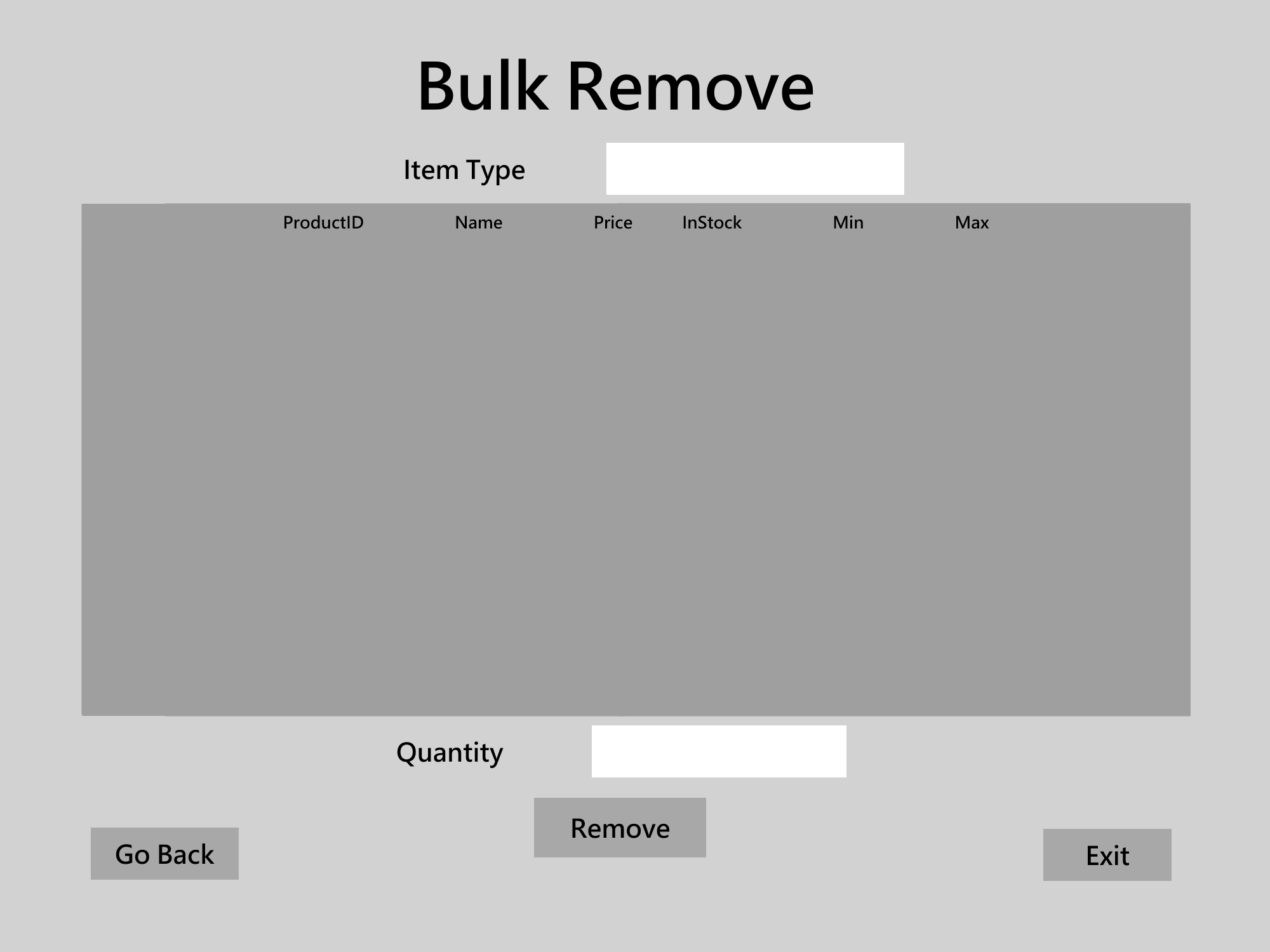
After clicking the “Orders” button on the Main Menu Page, the Orders Page appears. Two data grids are on this page. The left one lists all assigned orders for the current user that is logged in. The right one details the current active order for the current user. By selecting a row in the left data table and clicking the “Set As Current Order” button, that order will be set as the user’s current order which will then update on the right table. By clicking the “Mark As Complete” button, the current order is marked as complete, and the In Stock amounts for each item in the order will be reduced. If an item is moved to the “MissingParts” or “MissingProducts” column, then the stock amount will not be adjusted.

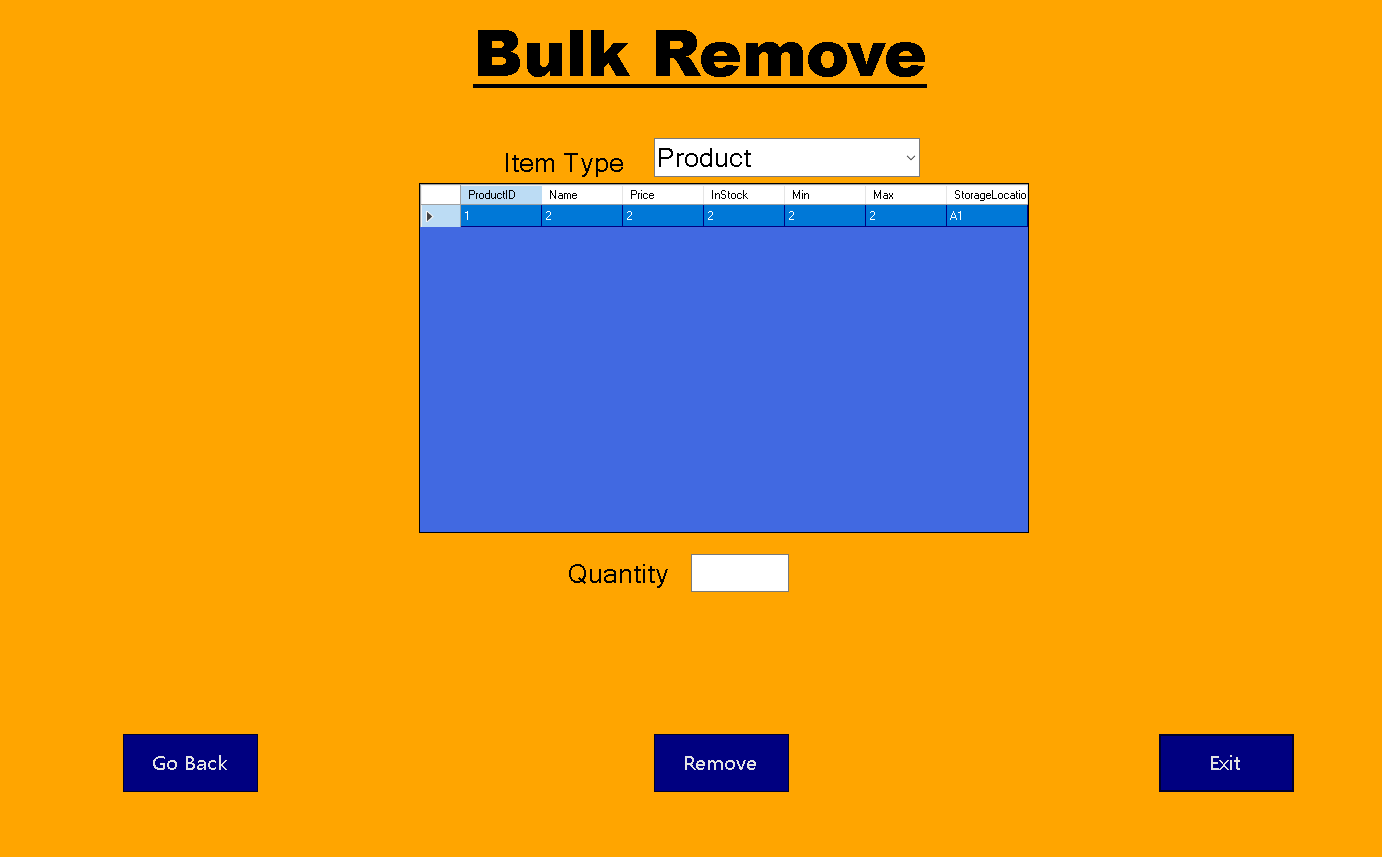
*Low Fidelity Orders Page Wireframe*

*High Fidelity Orders Page Wireframe*

**Bulk Remove Page**

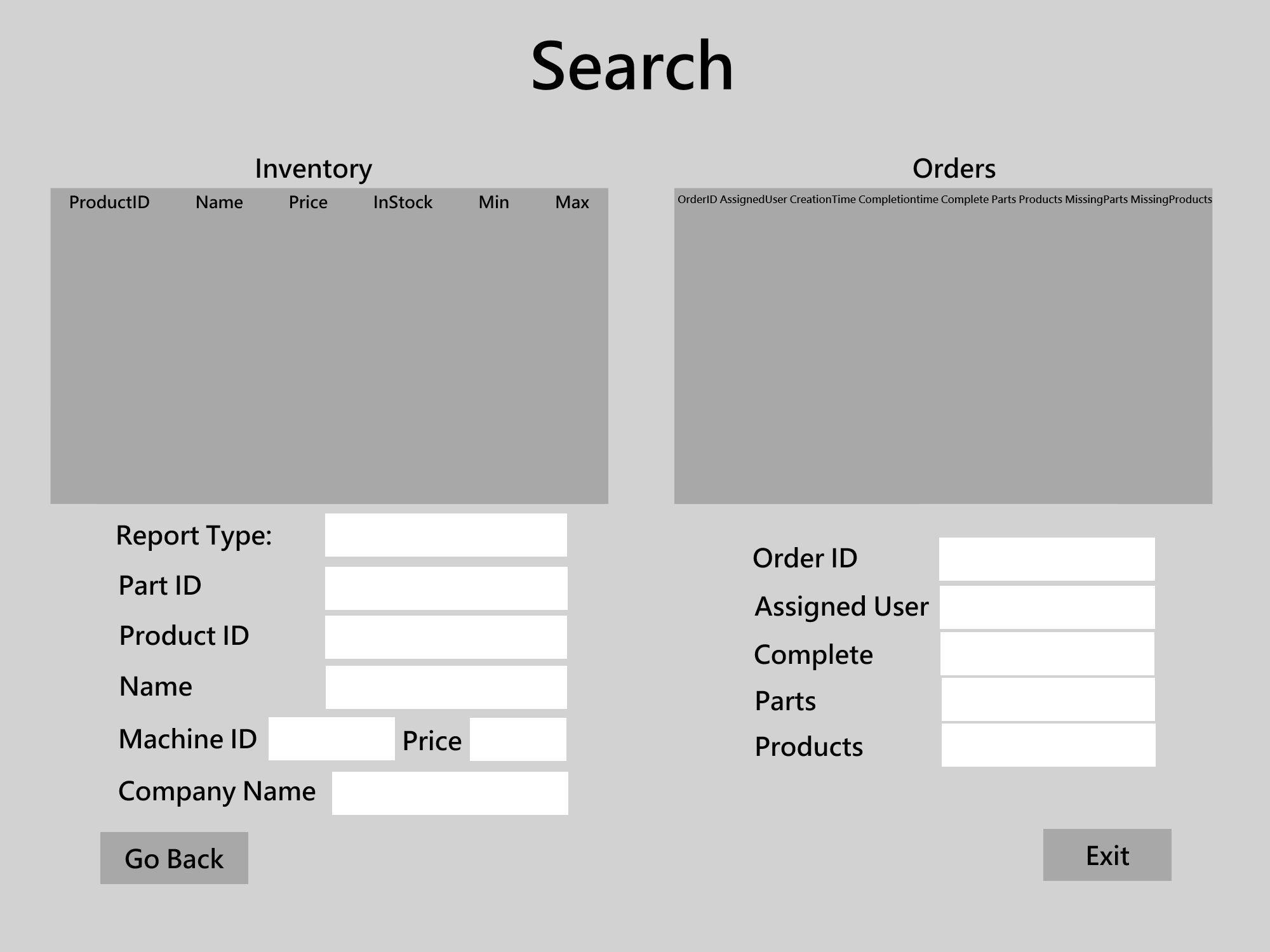
After clicking the “Bulk Remove” button on the Main Menu Page, the Bulk Remove Page appears. The ItemType dropbox contains Product, Inhouse Parts, and Outsourced Parts. By selecting one of those items, the data grid is loaded with all inventory data of the item type that is selected. The user then selects a row from the data grid, enters a number into the quantity box, and clicks the “Remove” button to reduce the in-stock amount of that item by the quantity entered.

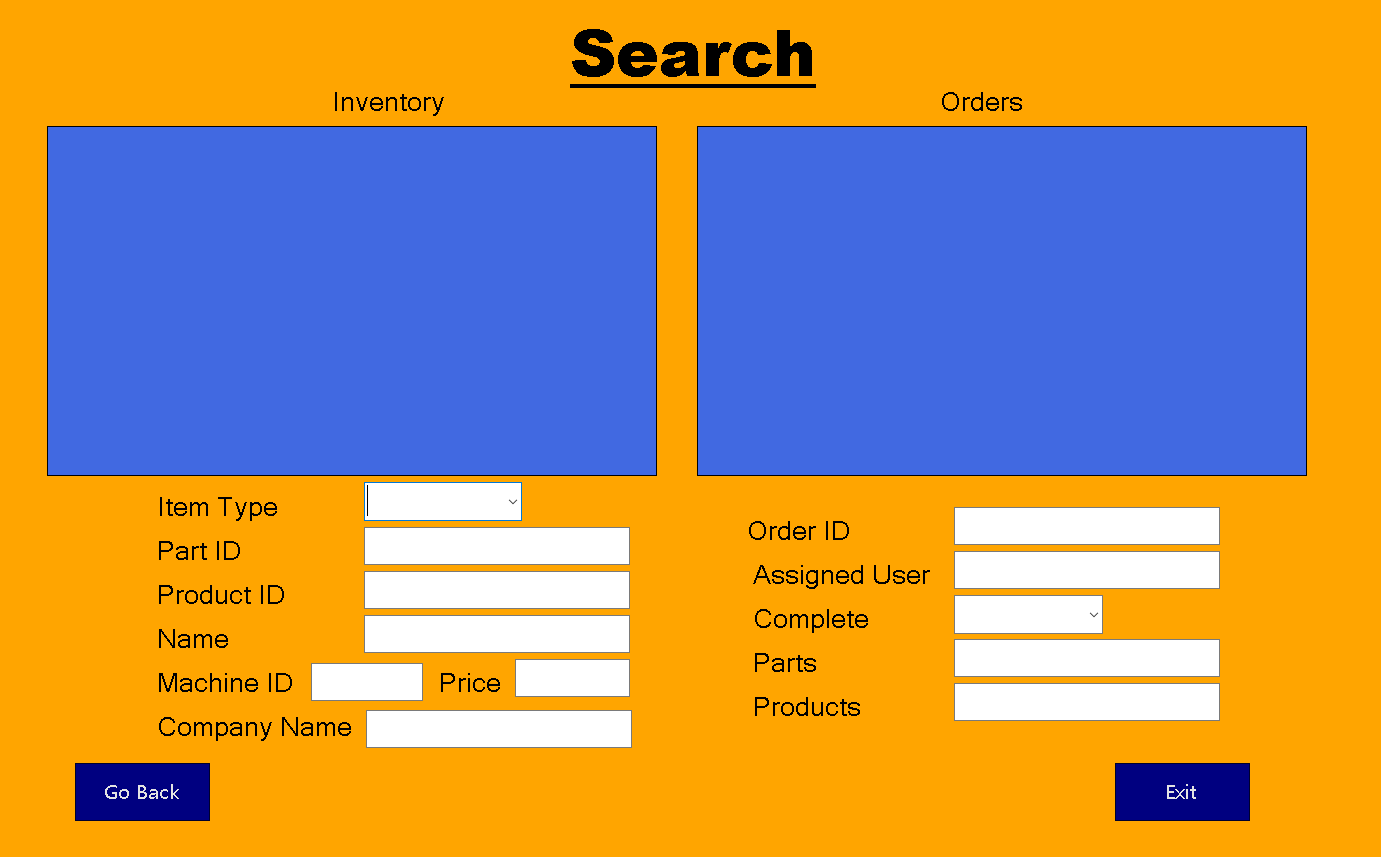
*Low Fidelity Bulk Remove Page*

*High Fidelity Bulk Remove Page*

**Search Page**

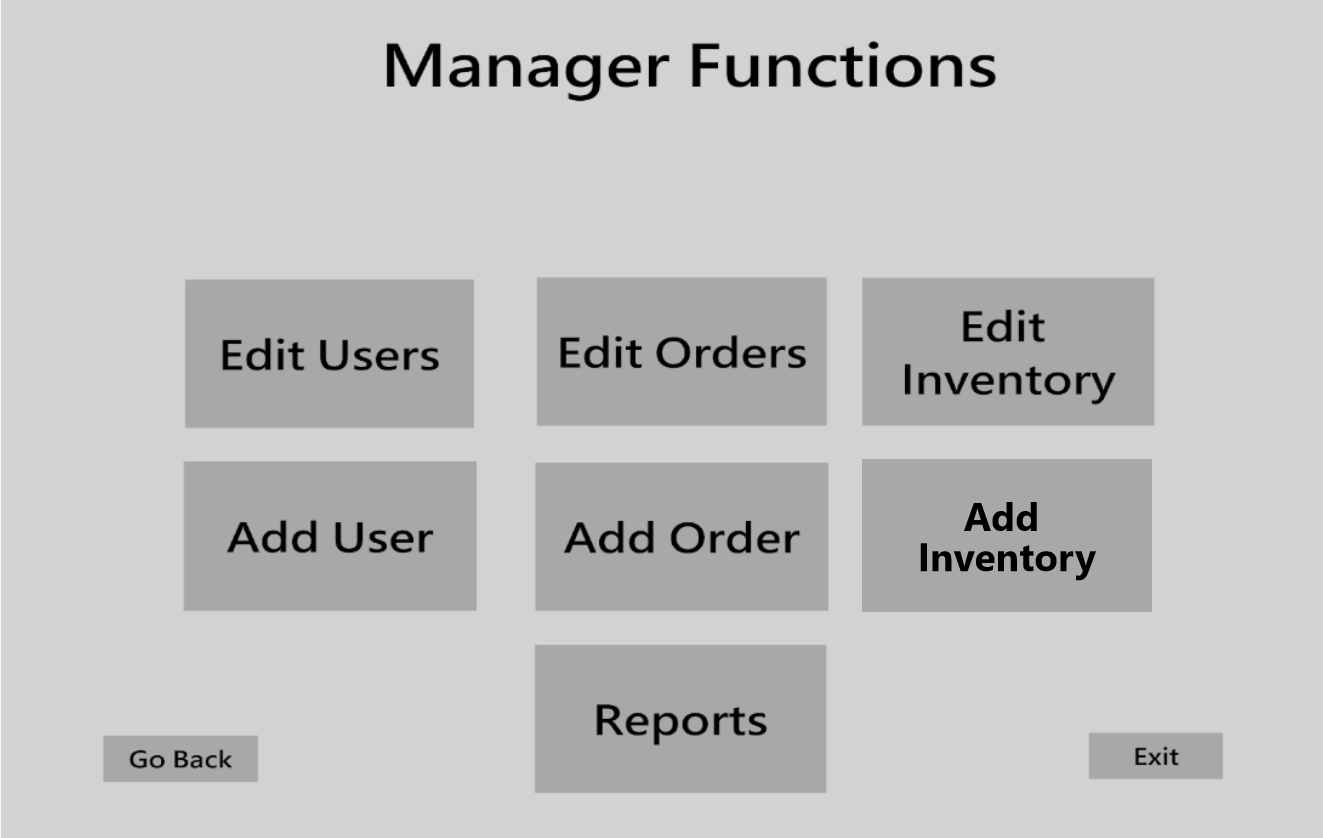
After clicking the “Search” button on the Main Menu Page, the Search Page appears. Here the user is able to search through the inventory by parts, and products. Relevant part or product data could be entered into the text boxes on the left. The left data grid will automatically update with any matching items. The data grid on the right will do the same for orders saved in the system, and for order data entered into the text boxes on the right.

*Low Fidelity Search Page Wireframe*

*High Fidelity Search Page Wireframe*

**Manager Functions Page**

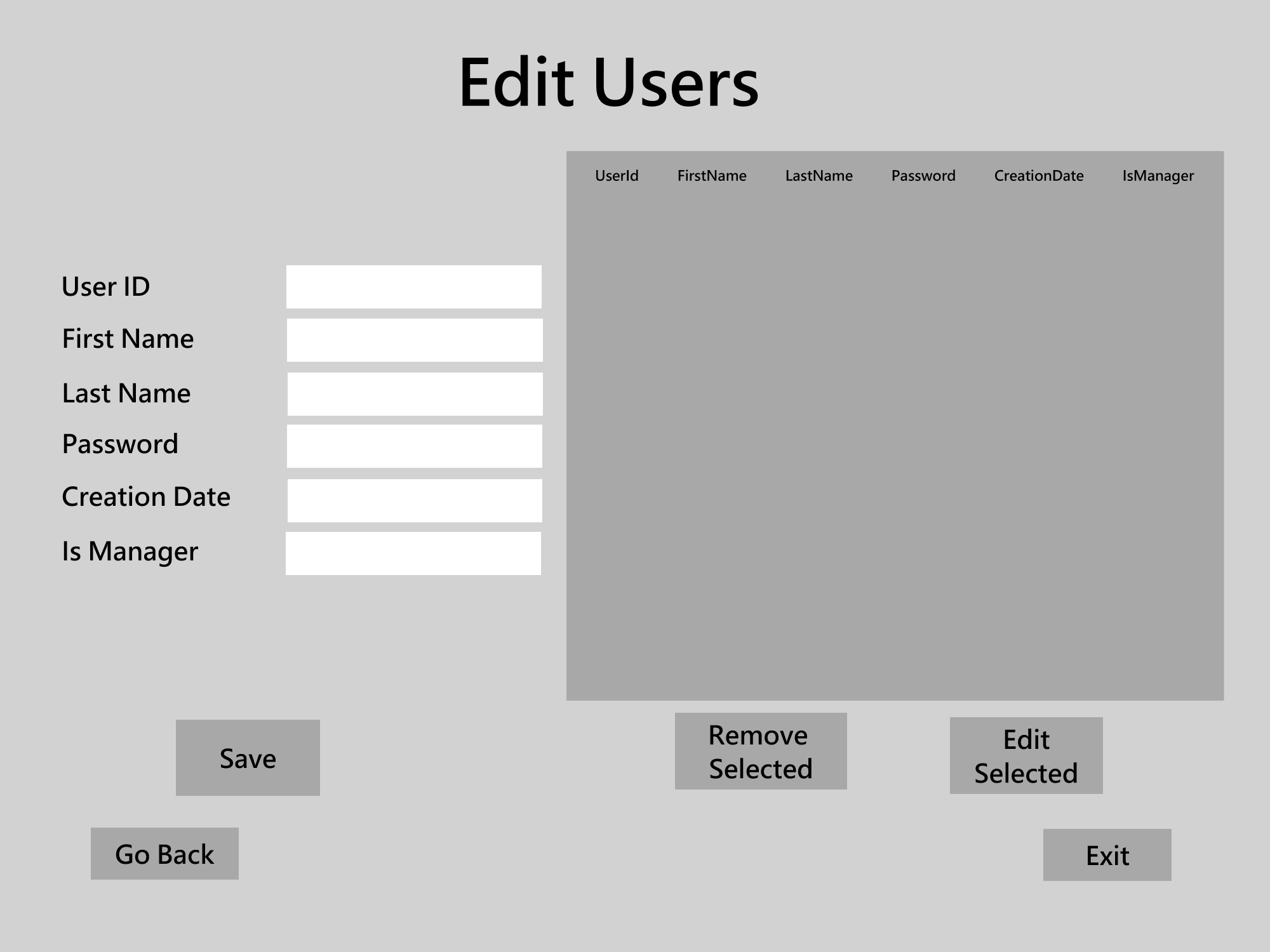
After clicking the “Manager Functions” button on the Main Menu Page, the Manager Functions Page appears. This page is only accessible to logged in users who are marked as managers. Here there are 7 buttons which include Edit Users, Edit Orders, Edit Inventory, Add User, Add Order, Add Inventory, and Reports..

*Low Fidelity Manager Functions Page*

*High Fidelity Manager Functions Page*

**Edit Users Page**

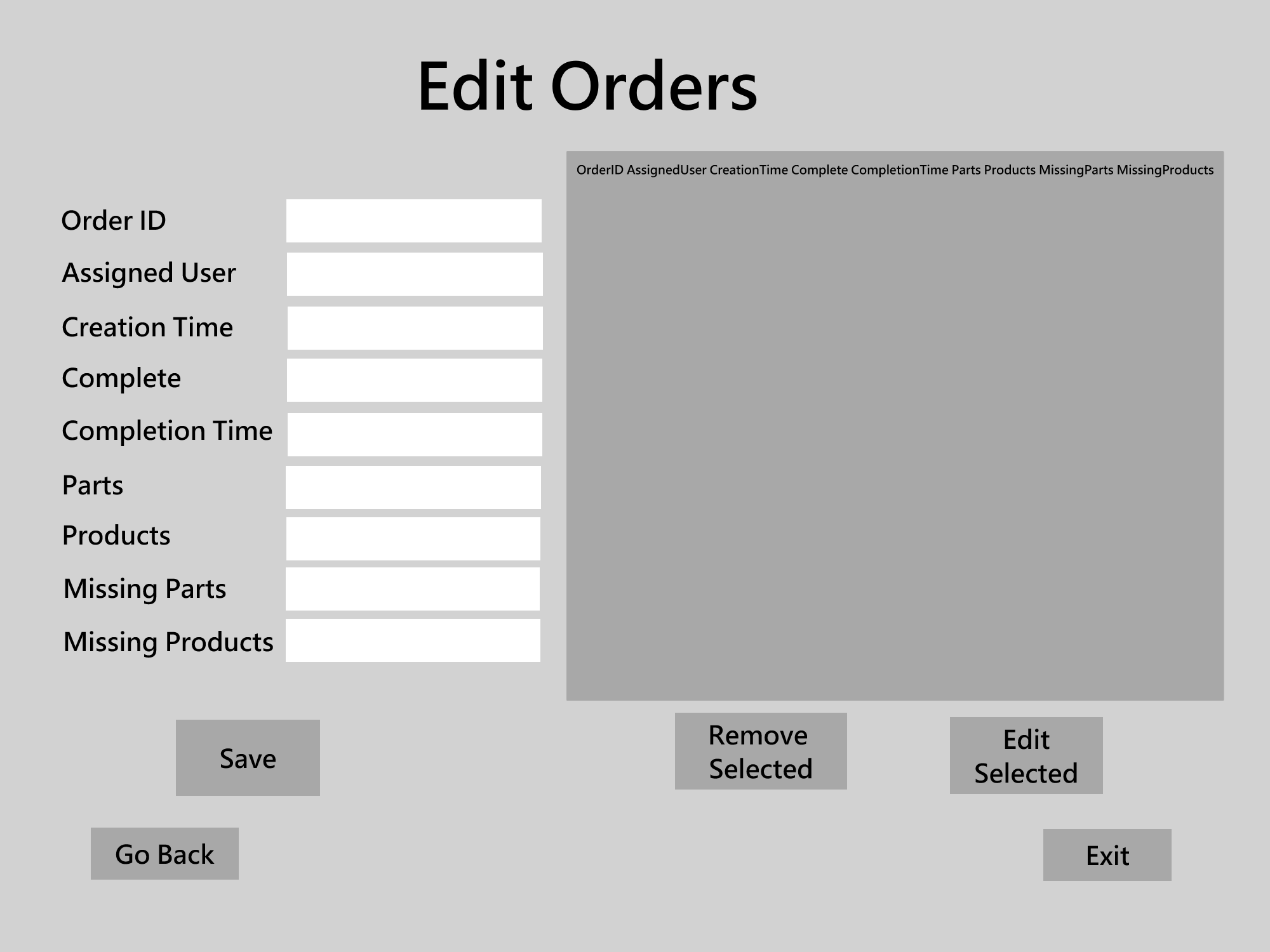
After clicking the “Edit Users” button on the Manager functions Page, The Edit Users Page appears. The data grid fills with data for each user in the database. The user is able to select a User ID from the drop box that contains all User IDs. Doing this will fill all the textboxes with the data from the selected user’s profile. Clicking the “Save” button will save any changes made to the selected user’s information. Clicking the “Remove Selected” button will remove the user that is selected in the data grid from the data grid. Clicking the “Edit Selected” button will select a matching User ID from the drop box that matches the user currently selected in the data grid. This will then fill the text boxes with the user’s information.

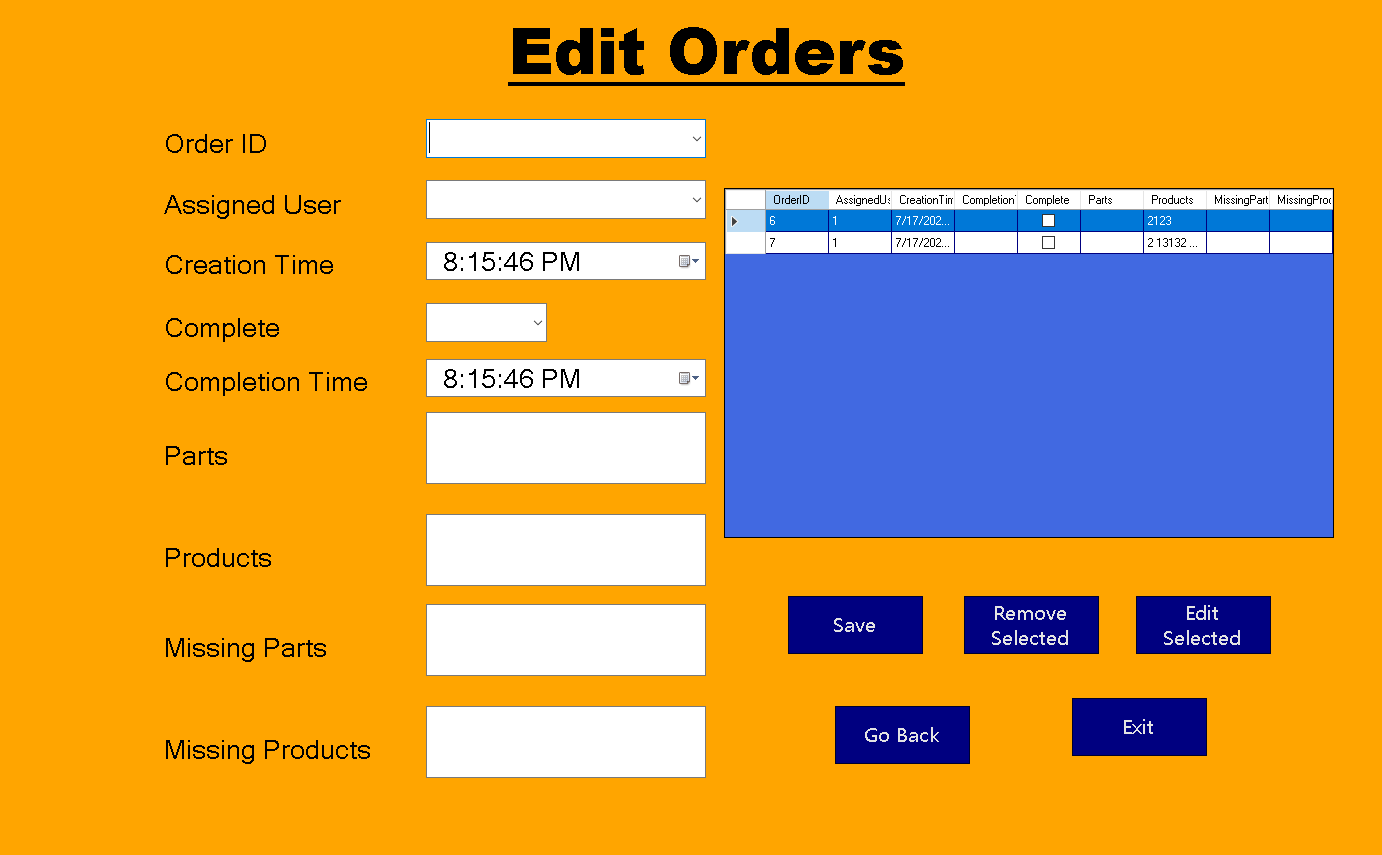
*Edit Users Page Low Fidelity Wireframe*

*Edit Users Page High Fidelity Wireframe*

**Edit Orders Page**

After clicking the “Edit Orders” button in the Manager Functions Page, the Edit Orders Page appears. The data grid is filled with all orders that are currently in the database. After selecting an Order ID from the dropbox, all text boxes are filled with that order’s relevant information. After clicking the “Save” button, any changes to the selected order will be saved to the database. By clicking the “Remove Selected” button, the user is able to remove an order from the database. It removes the corresponding order that relates to the selected row in the data grid. Clicking the “Edit Selected” button fills all text boxes with the selected order relating to the selected row’s information.

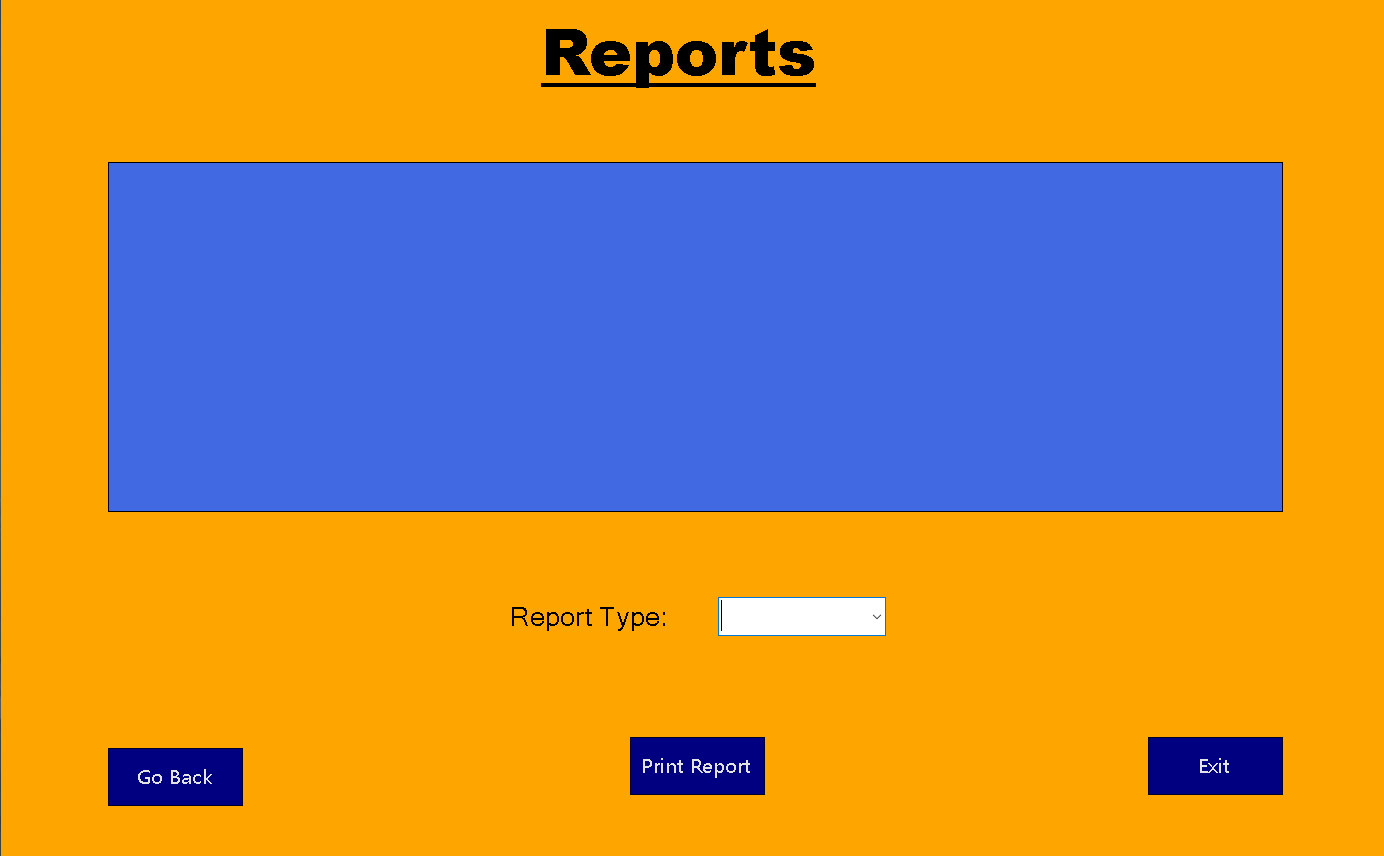
*Edit Orders Page Low Fidelity Wireframe*

*Edit Orders Page High Fidelity Wireframe*

**Reports Page**

After clicking the “Reports” button on the Manager Functions Page, the Reports Page appears. Here is a data grid that fills with the selected report. The selected reports are chosen from the drop box which include Orders, Products, Inhouse Parts, Outsourced Parts, All Parts, and Users. Clicking the “Print Report” button sends the report to a connected printer to be printed.

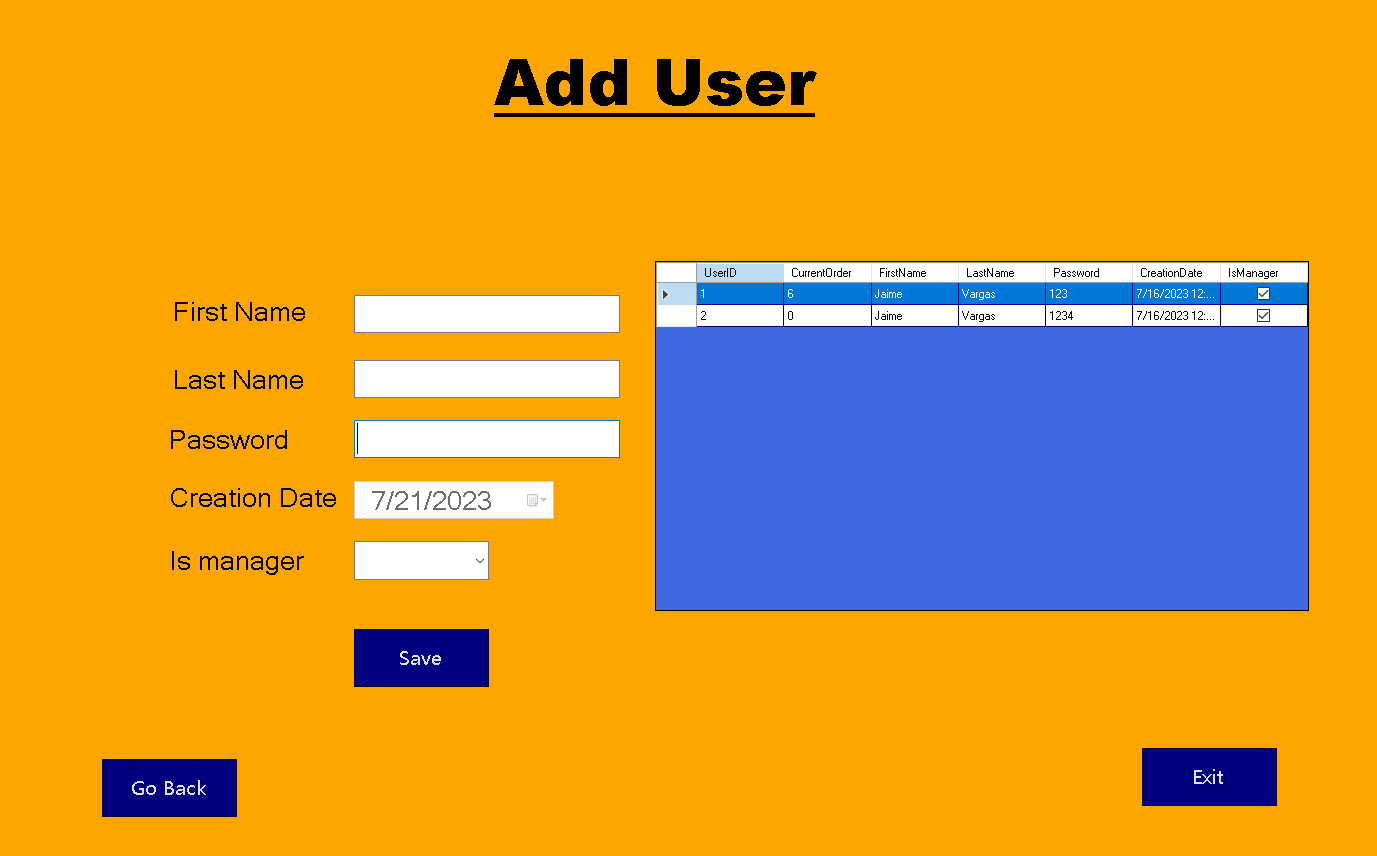
*Low Fidelity Reports Page Wireframe*

*High Fidelity Reports Page Wireframe*

**Add User Page**

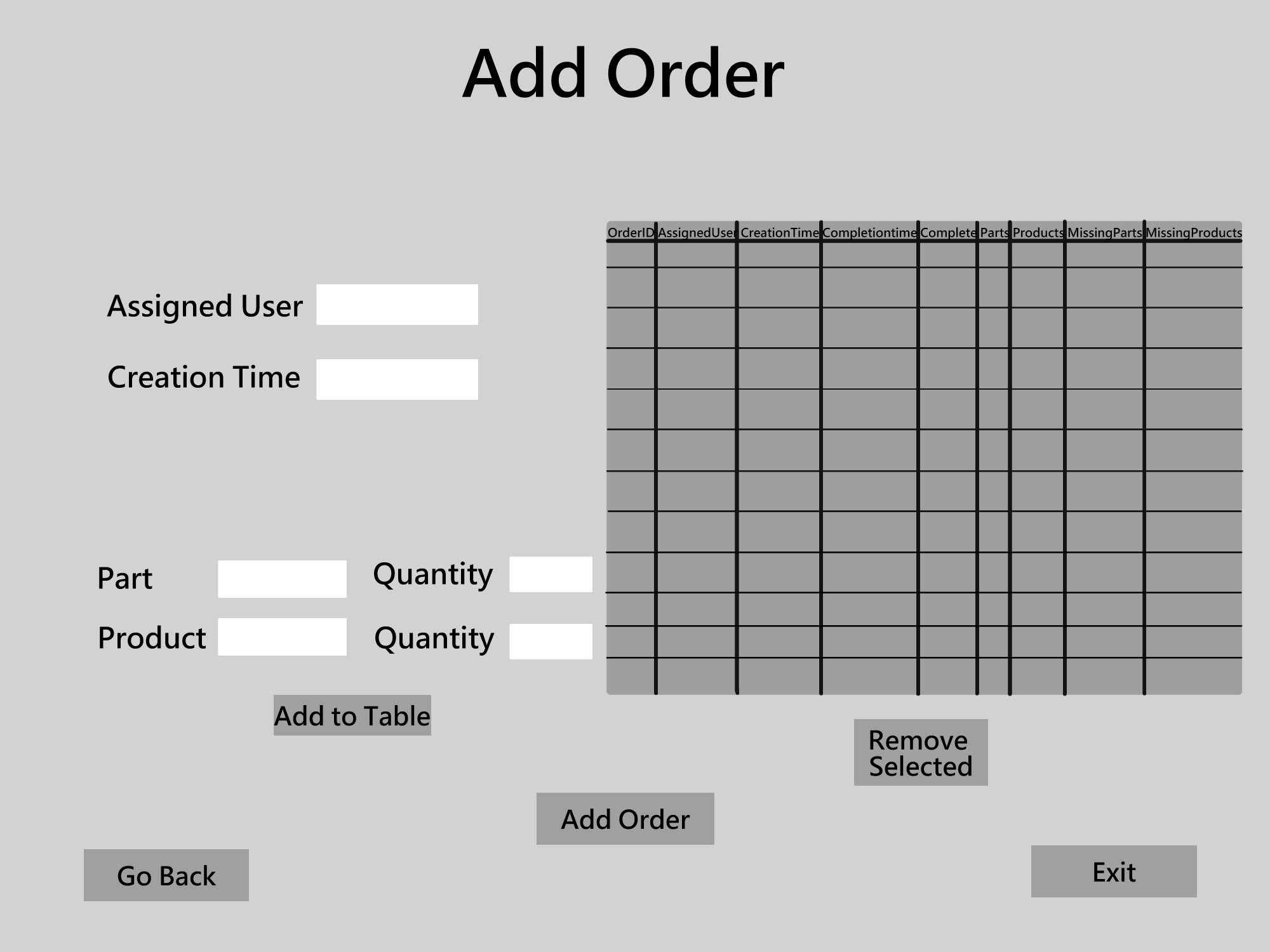
After clicking the “Add User” button in the Manager Functions Page, the Add User Page appears. The data grid fills with all users that are in the database. After filling in the textboxes for First Name, Last Name, Password, and selecting true or false for Is manager, the user can add a new user to the database by clicking the “Save” button.

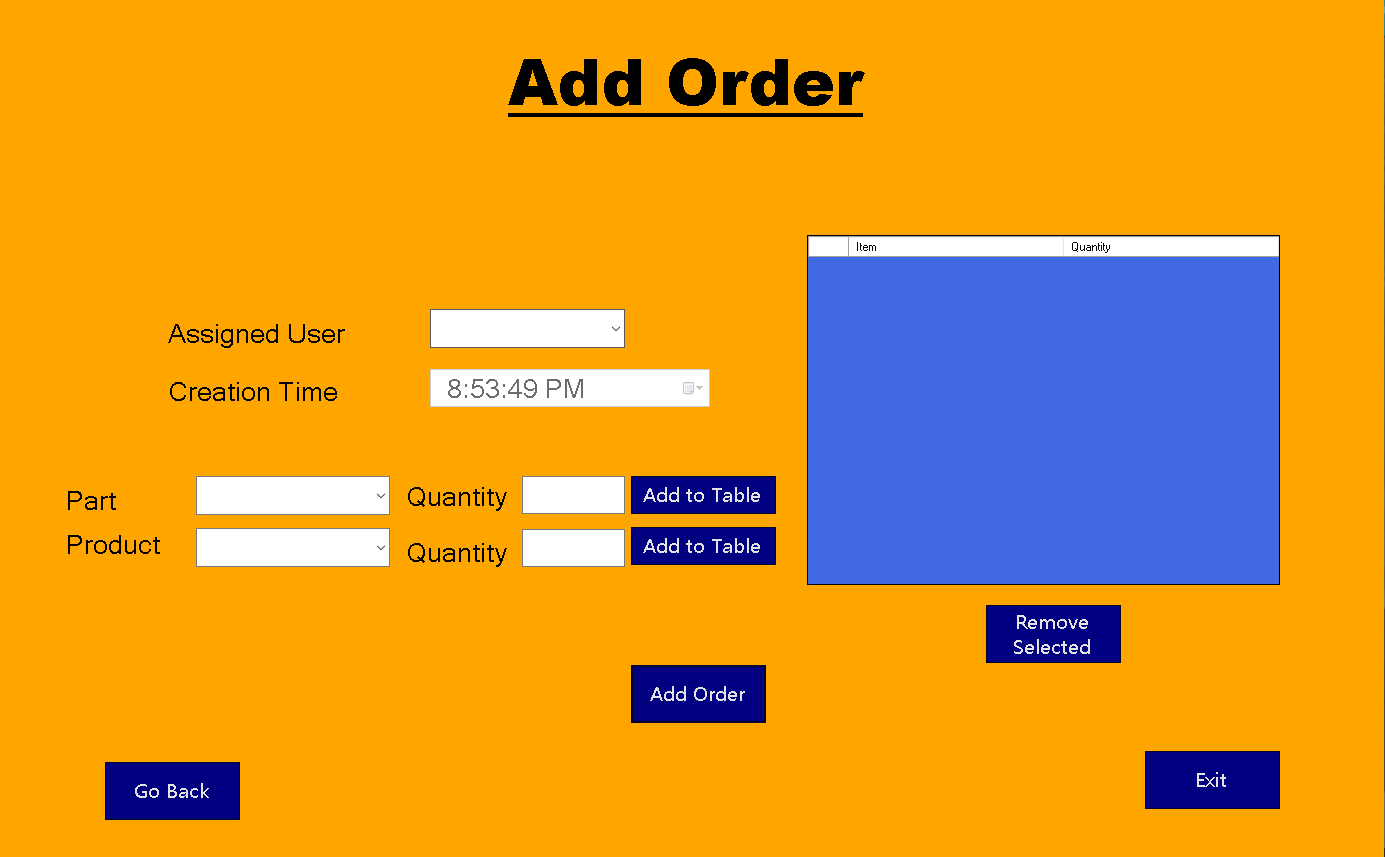
*Low Fidelity Add User Page Wireframe*

*High Fidelity Add User Page Wireframe*

**Add Order Page**

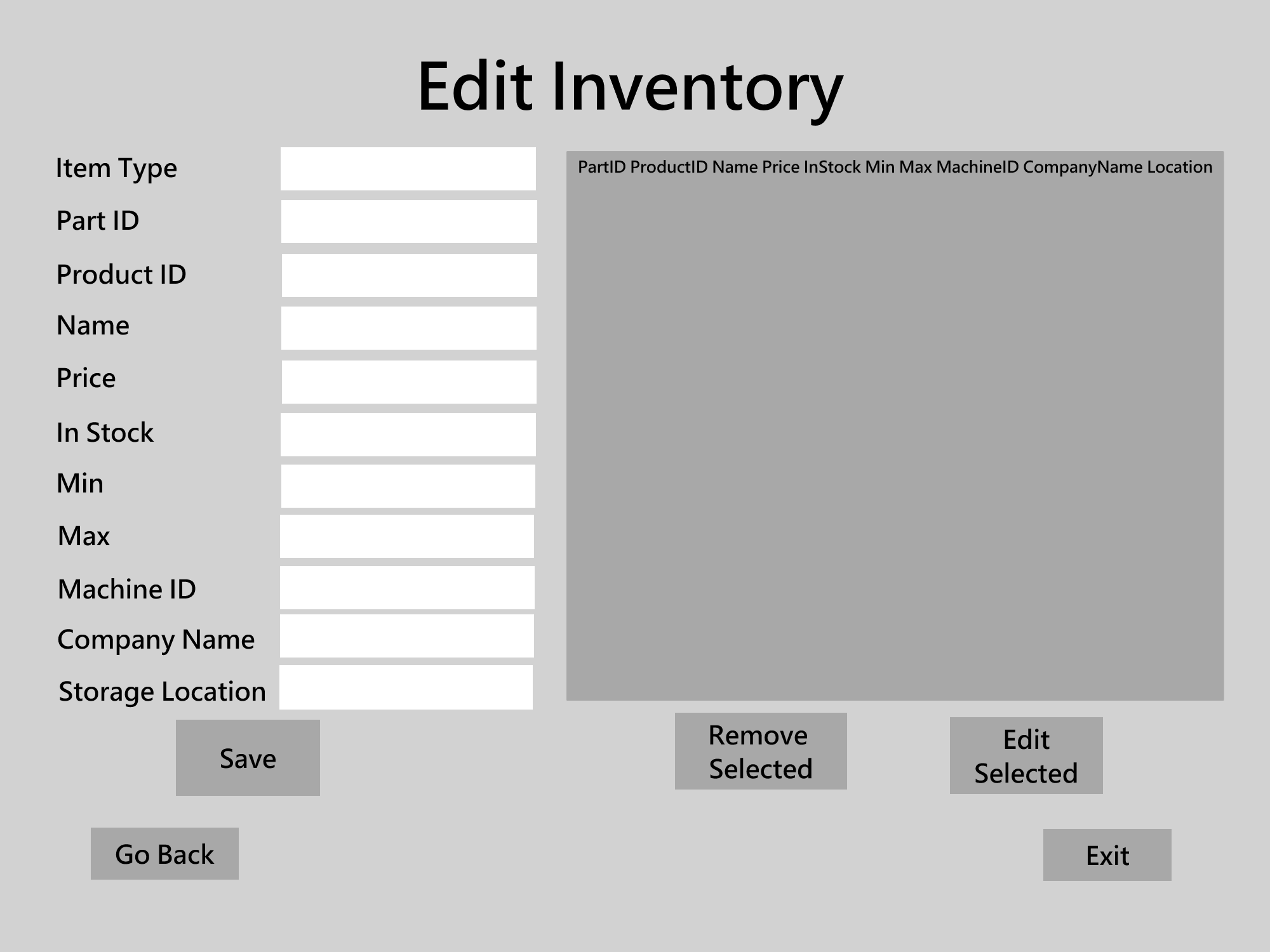
After clicking the “Add Order” button from the Manager Functions Page, the Add Order Page appears. Here the user is able to add new orders to the database. All User IDs are loaded into the Assigned User drop box. A user needs to be selected from this drop box to assign the order to them. Then, a part or product is selected, and their corresponding quantity needs to be filled, and the “Add to Table” button in the same row will add the item name along with its quantity to the data table. Clicking the “Remove Selected” button will remove the selected row from the data grid. Clicking the “Add Order” button will create a new order and add it to the database using the assigned user, creation time, and items and quantities from the data grid.

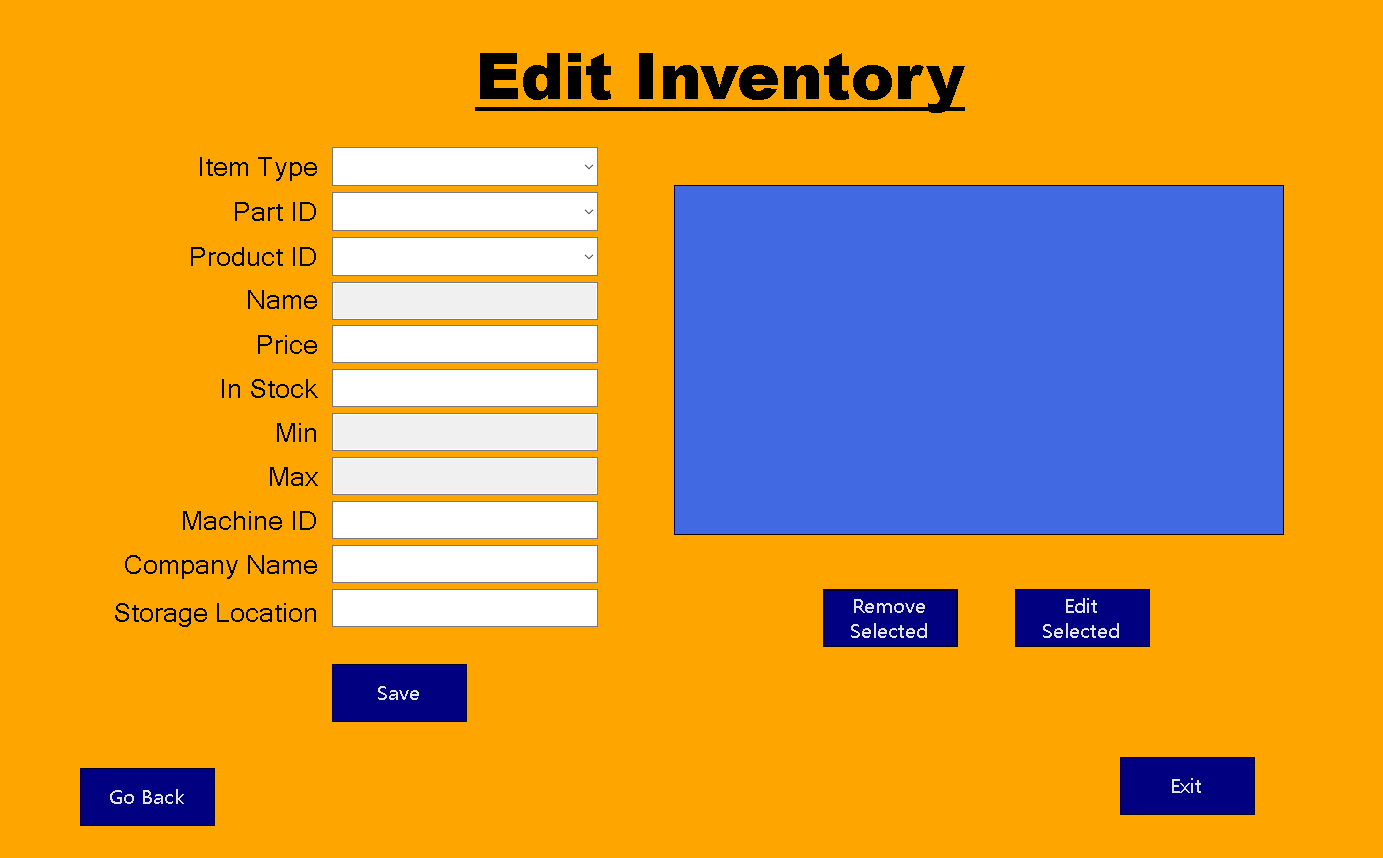
*Low Fidelity Add Order Page Wireframe*

*High Fidelity Add Order Page Wireframe*

**Edit Inventory Page**

After clicking the “Edit Inventory” button, the Edit Inventory Page Appears. Here the user is able to select item types and IDs from drop boxes. Once a type and ID is selected, the rest of the textboxes will autofill with corresponding data matching the main ID of the item. The data grid fills with data relevant to the type of item that is selected from the drop box. For example if parts are selected from the drop box, the data grid will fill with all parts in the database. Clicking the “Save” button will update the selected item in the database with the information that was entered into the text boxes. Clicking the “Remove Selected” button will remove the item that matches the selected item’s row from the database. Clicking the “Edit Selected” button will fill all text boxes with the selected row’s relevant information.

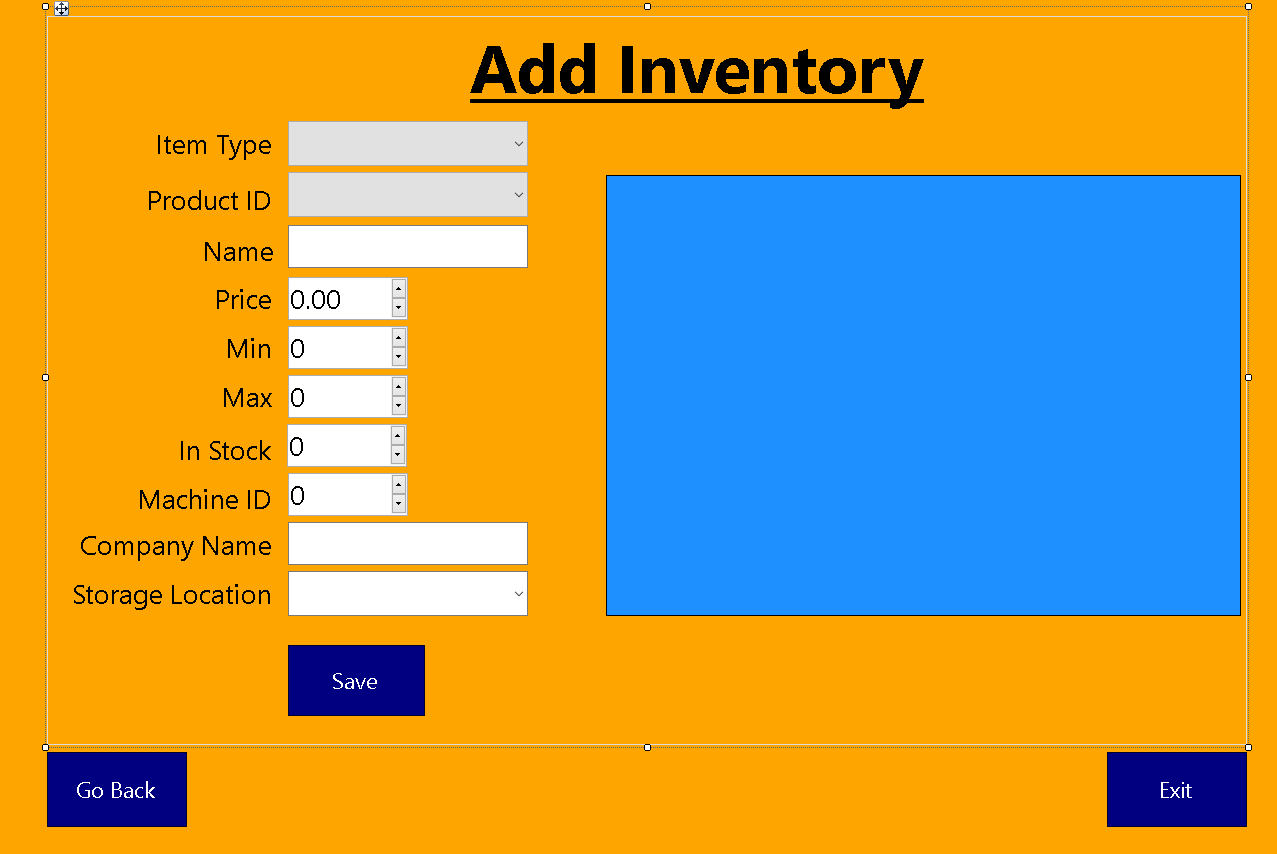
*Low Fidelity Edit Inventory Page Wireframe*

*High Fidelity Edit Inventory Page Wireframe*

**Add Inventory Page**

After clicking the “Add Inventory” button, the Edit Inventory Page Appears. Here the user is able to add new inventory items. The item type could be either an in-house part, outsourced part, or product. Once all entries are filled, clicking the “Save” button will add the item to the database and update the data table.

*Low Fidelity Add Inventory Page Wireframe*

*High Fidelity Add Inventory Page*

# Unit Test Plan

### Purpose

The main type of testing done for the application is manual testing. Testing was done through the user interface. Each page is considered an individual unit. The functionality, data handling, error handling, and navigation of the pages were tested. All unit tests must be passed before the unit is integrated into the main system. The results of the unit tests are displayed in the upcoming tables. Each table represents a page, or unit, that is being tested. Any errors, bugs, or issues found during testing are documented and assigned to a developer to fix. Then the tests are run again and the process repeats until all unit tests are passed. After all unit tests pass, full regression testing is done to ensure that the software still performs as expected.

### Overview

The main functionality of this application is inventory management. Being able to efficiently manage inventory for the distribution centers are key to their operation. That is why checking for proper inventory editing is very important to the success of this project. The handling of data transfer through the application could become complicated, it is important to verify that the data is being processed correctly.

Data being represented in the user interface was checked against the data that is actually shown in the database. Accurate relationships between products, parts, orders, and users is vital for proper functioning of the application. Editing, adding, removing, and updating objects of the database through the user interface was thoroughly tested to ensure that their functions are being executed correctly. Certain values such as IDs are autogenerated, which means that the user should not be able to change this value. Correct data for entries was tested throughout the application. For example, no empty entries or white space entries are allowed. Many forms autofill data based on selections, this was tested to see if it was done correctly. Buttons were tested to see if they handle invalid arguments, such as clicking a “Delete” button when no row is selected. All combo boxes, text boxes, data grids, and buttons were thoroughly tested for proper functionality, error handling, and data manipulation.

**Add Inventory Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 1 | Check for blank field validation | 1. Do not select or enter any information 2. Click the “Add” button | Clicking the “Add” button should display a relevant error message and no new item should be added. | Clicking the “Add” button should display a relevant error message and no new item should be added. | Pass |
| 2 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 3 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Manager Functions page | The application returns to the Manager Functions page | Pass |
| 4 | Add new item | 1. Select an item type from the drop box 2. If the item type is product, fill in Name, Price, In-Stock, Min, Max, and Storage Location.   Else Fill in Name, Price, In-Stock, Min, Max, Machine ID, Company Name, and Storage Location entries.   1. Click the “Add” button | A confirmation message is shown, then a new item is added to the database which is shown in the data grid. | A confirmation message is shown, then a new item is added to the database which is shown in the data grid. | Pass |

**Add Order Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 5 | Add part to table | 1. Select a user from the “assigned user” combo box if not done already 2. Select a part form the “part” combo box 3. Enter number in the quantity text box in the part row 4. Click the “Add to Table” button in the part row | The part and quantity are added to the page’s table | The part and quantity are added to the page’s table | Pass |
| 6 | Add product to table | 1. Select a user from the “assigned user” combo box if not done already 2. Select a product from the “product” combo box 3. Enter a number in the quantity textbox in the product row 4. Click the “Add to Table” button in the product row | The product and quantity are added to the page’s table | The product and quantity are added to the page’s table | Pass |
| 7 | Remove selected | 1. Select a row in the table 2. Click the “Remove Selected” button | The selected row is removed from the table | The selected row is removed from the table | Pass |
| 8 | Add order | 1. Add at least one part or product to the table 2. Click the “Add Order” button | The order is added to the database and a confirmation message is displayed | The order is added to the database and a confirmation message is displayed | Pass |
| 9 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Manager Functions page | The application returns to the Manager Functions page | Pass |
| 10 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 11 | Check for blank field validation | 1. Do not select or enter any information 2. Click the “Add Order” button 3. Click each “Add to Table” button | The “Add to Table” buttons should display a relevant error message and not add to the table.  The “Add Order” button should display a relevant error message and not add the order to the database | The “Add to Table” buttons should display a relevant error message and not add to the table.  The “Add Order” button should display a relevant error message and not add the order to the database | Pass |
| 12 | Check for invalid part detection | 1. Enter a part that is not in the “Part” combo box list  2. Enter a nonnegative integer into the part quantity box  3. Click the “Add to Table'' button in the part row | An error message shows, and the part is not added to the table | An error message shows, and the part is not added to the table | Pass |

**Add User Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 13 | Check for blank field validation | 1. Do not select or enter any information 2. Click the “Save” button | Clicking the “Save” button should display a relevant error message and no new user should be added. | Clicking the “Save” button should display a relevant error message and no new user should be added. | Pass |
| 14 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 15 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Manager Functions page | The application returns to the Manager Functions page | Pass |
| 16 | Add new user | 1. Enter values for First Name, Last Name, and Password 2. Select True or False in the Is manager drop box. 3. Click the “Save” button | A confirmation message is shown, then a new user is added to the database which is shown in the data grid. | A confirmation message is shown, then a new user is added to the database which is shown in the data grid. | Pass |

**Bulk Add Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 17 | Check for blank field validation | 1. Do not select or enter any information 2. Click the “Add to Table” button | Clicking the “Add to Table” button should display a relevant error message and nothing is added to the table | Clicking the “Add to Table” button should display a relevant error message and nothing is added to the table | Pass |
| 18 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 19 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |
| 20 | Add item to table | 1. Select values for the Item Type and ID drop boxes. 2. Enter a nonnegative integer into the quantity textbox 3. Click the “Add to Table” button | A new item is added to the data grid as a row. | A new item is added to the datagrid as a row. | Pass |
| 21 | Remove from table | 1. Add an item to the table 2. Select the row that is in the table 3. Click the “Remove Selected” button | The selected row is removed from the table | The selected row is removed from the table | Pass |
| 22 | Detect nothing to remove | 1. Do not add anything to the data table 2. Click the “Remove Selected” button | Nothing happens. The program should not crash because it catches the invalid operation | Nothing happens. The program should not crash because it catches the invalid operation | Pass |
| 23 | Edit selected row | 1. Add an item to the table 2. Select the row that is in the table 3. Click the “Edit Selected” button | The item type, ID, name, and quantity entries auto-fill with the row’s information | The item type, ID, name, and quantity entries auto-fill with the row’s information | Pass |
| 24 | Detect nothing to edit | 1. Do not add anything to the data table 2. Click the “Edit Selected” button | Nothing happens. The program should not crash because it catches the invalid operation | Nothing happens. The program should not crash because it catches the invalid operation | Pass |

**Bulk Remove Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 25 | Check for selected item type | 1. Do not select an item type from the drop box 2. Click the “Remove” button | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Pass |
| 26 | Check for entered quantity | 1. Select an item type from the drop box 2. Delete the number from the quantity entry box 3. Click the “Remove” button | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Pass |
| 27 | Check for selected item from data table | 1. Select an item type from the drop box 2. Enter a quantity or leave the default quantity of 1 3. Do not click any row in the data grid 4. Click the “Remove” button | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Pass |
| 28 | Check for enough stock | 1. Select an item type from the drop box 2. Select a row from the data grid 3. Enter a quantity that is larger than the selected row’s quantity | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Clicking the “Remove” button should display a relevant error message and no item quantity is changed | Pass |
| 29 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 30 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |

**Edit Inventory Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 31 | Check for empty values | 1. Leave any of the entry boxes empty 2. Click the “Save” button | A relevant error message appears and no inventory item is edited | A relevant error message appears and no inventory item is edited | Pass |
| 32 | Edit product | 1. Select “Product” from the item type drop box 2. Select a product ID from the drop box 3. Enter new information or leave the information the same for the Name, price, in-stock, min, max, and storage location boxes 4. Click the “Save” button | A confirmation is shown and the selected item’s details are updated, the updates are reflected in the data grid. | A confirmation is shown and the selected item’s details are updated, the updates are reflected in the data grid. | Pass |
| 33 | Test remove selected | 1. Select any item type from the “Item Type” drop box 2. Select a row from the data table 3. Click the “Remove Selected” button | The selected row is removed from the data table | The selected row is removed from the data table | Pass |
| 34 | Test empty remove | 1. Do not select any row in the data table 2. Click the “Remove Selected” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |
| 35 | Test edit selected | 1. Select an item type from the “Item Type” drop box 2. Select a row from the data table 3. Click the “Edit Selected” button | The information from the selected row populates its relevant information into the text boxes | The information from the selected row populates its relevant information into the text boxes | Pass |
| 36 | Test edit empty | 1. Do not select any row in the data table 2. Click the “Edit Selected” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |
| 37 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 38 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Manager Functions Page | The application returns to the Manager Functions Page | Pass |

**Edit Order Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 39 | Check for empty values | 1. Leave any of the entry boxes empty 2. Click the “Save” button | A relevant error message appears and no order is edited | A relevant error message appears and no order is edited | Pass |
| 40 | Edit order | 1. Select an Order ID from the drop box 2. Select a user from the Assigned User drop box 3. Select a date for the Creation time, and completion time boxes 4. Mark the order as true or false for the Complete drop box 5. For parts, products, missing parts, and missing products, enter the names of existing items followed by a number to represent quantity, and then press enter for each item in each text box.   ex: “Rebar 23  Rods 12  Shingles 7 “ for Parts   1. Click the “Save” button | The selected order’s information is updated and the changes are shown in the data table. | The selected order’s information is updated and the changes are shown in the data table. | Pass |
| 41 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 42 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Manager Functions Page | The application returns to the Manager Functions Page | Pass |
| 43 | Test remove selected | 1. Select a row from the data table 2. Click the “Remove Selected” button | The selected row is removed from the data table | The selected row is removed from the data table | Pass |
| 44 | Test empty remove | 1. Do not select any row in the data table 2. Click the “Remove Selected” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |
| 45 | Test edit selected | 1. Select a row from the data table 2. Click the “Edit Selected” button | The information from the selected row populates its relevant information into the text boxes | The information from the selected row populates its relevant information into the text boxes | Pass |
| 46 | Test edit empty | 1. Do not select any row in the data table 2. Click the “Edit Selected” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |

**Edit Users Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 47 | Check for empty values | 1. Leave any of the entry boxes empty 2. Click the “Save” button | A relevant error message appears and no order is edited | A relevant error message appears and no order is edited | Pass |
| 48 | Edit order | 1. Select a User ID from the drop box 2. Type in the user’s first name, last name, and password in the corresponding text boxes 3. Select a date for the creation date in the date picker 4. Select true or false from the is manager drop box 5. click the “Save” button | The selected user’s information is updated and the changes are shown in the data table. | The selected user’s information is updated and the changes are shown in the data table. | Pass |
| 49 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 50 | Go back to previous page | 1. Click the “Go Back” button | The application returns to the Manager Functions Page | The application returns to the Manager Functions Page | Pass |
| 51 | Test remove selected | 1. Select a row from the data table 2. Click the “Remove Selected” button | The selected row is removed from the data table | The selected row is removed from the data table | Pass |
| 52 | Test empty remove | 1. Do not select any row in the data table 2. Click the “Remove Selected” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |
| 53 | Test edit selected | 1. Select a row from the data table 2. Click the “Edit Selected” button | The information from the selected row populates its relevant information into the text boxes | The information from the selected row populates its relevant information into the text boxes | Pass |
| 54 | Test edit empty | 1. Do not select any row in the data table 2. Click the “Edit Selected” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |

**Login Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 55 | Test incorrect username and password | 1. Enter a username and password combination that does not exist in the database OR leave one of the text boxes empty 2. Click the “Log In” button | A relevant error message appears and the user is not logged in to the system | A relevant error message appears and the user is not logged in to the system | Pass |
| 56 | Exit application | 1. Click the “Exit” button | The application closes | The application closes | Pass |

**Main Menu Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 57 | Test “Bulk Add” button | 1. Click the “Bulk Add” button | The Main Menu Page disappears and the Bulk Add Page appears. | The Main Menu Page disappears and the Bulk Add Page appears. | Pass |
| 58 | Test “Storage” button | 1. Click the “Storage” button | The Main Menu Page disappears and the Storage Page appears. | The Main Menu Page disappears and the Storage Page appears. | Pass |
| 59 | Test “Orders” button | 1. Click the “Orders” button | The Main Menu Page disappears and the Orders Page appears. | The Main Menu Page disappears and the Orders Page appears. | Pass |
| 60 | Test “Bulk Remove” button | 1. Click the “Bulk Remove” button | The Main Menu Page disappears and the Bulk Remove Page appears. | The Main Menu page disappears and the Bulk Remove Page appears. | Pass |
| 61 | Test “Search” button | 1. Click the “Search” button | The Main Menu Page disappears and the Search Page appears. | The Main Menu Page disappears and the Search Page appears. | Pass |
| 62 | Test “Manager Functions” | 1. Click the “Manager Functions” button | The Main Menu Page disappears and the Manager Functions Page appears | The Main Menu Page disappears and the Manager Functions Page appears |  |
| 63 | Exit application | 1. Click the “Exit” button | The application returns to the Login Page | The application returns to the Login page | Pass |

**Manager Functions Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 64 | Test “Edit Users” button | 1. Click the “Edit Users” button | The Manager Functions Page disappears and the Edit Users Page appears. | The Manager Functions Page disappears and the Edit Users Page appears. | Pass |
| 65 | Test “Edit Orders” button | 1. Click the “Edit Orders” button | The Manager Functions Page disappears and the Edit Orders Page appears. | The Manager Functions Page disappears and the Edit Orders Page appears. | Pass |
| 66 | Test “Reports” button | 1. Click the “Reports” button | The Manager Functions Page disappears and the Reports Page appears. | The Manager Functions Page disappears and the Reports Page appears. | Pass |
| 67 | Test “Add User” button | 1. Click the “Add User” button | The Manager Functions Page disappears and the Add User Page appears. | The Manager Functions Page disappears and the Add User Page appears. | Pass |
| 68 | Test “Add Order” button | 1. Click the “Add Order” button | The Manager Functions Page disappears and the Add Order Page appears. | The Manager Functions Page disappears and the Add Order Page appears. | Pass |
| 69 | Test “Edit Inventory” button | 1. Click the “Edit Inventory” button | The Manager Functions Page disappears and the Edit Inventory Page appears. | The Manager Functions Page disappears and the Edit Inventory Page appears. | Pass |
| 70 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 71 | Go back to the previous page | 1. Click the “Go Back” button. | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |

**Orders Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 72 | Go back to the previous page | 1. Click the “Go Back” button. | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |
| 73 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 74 | Test setting current order | 1. Select a row from the data grid labeled “Assigned Orders” 2. Click the “Set As Current Order” button | The selected order is set as the current order, and the data grid labeled “Current Order” fills with the new current order’s information | The selected order is set as the current order, and the data grid labeled “Current Order” fills with the new current order’s information | Pass |
| 75 | Test set empty order | 1. Do not select any row from the data table labeled “Assigned Orders” 2. Click the “Set As Current Order” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |
| 76 | Test order complete | 1. Check that there is a current order for the logged in user, if not then set an order as the current order 2. Click the “Mark As Complete” button | The order is removed from the “Current Order” data table. The in-stock amount for each item in the order is decreased for each item not marked missing. | The order is removed from the “Current Order” data table. The in-stock amount for each item in the order is decreased for each item not marked missing. | Pass |
| 77 | Test set empty order | 1. After completing an order, the “Current Order” data table should be empty. 2. With the “Current Order” data table empty, click the “Mark As Complete” button | Nothing happens. The system doesn’t crash because it detects no row is selected | Nothing happens. The system doesn’t crash because it detects no row is selected | Pass |

**Reports Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 78 | Go back to the previous page | 1. Click the “Go Back” button. | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |
| 79 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 80 | Print Report | 1. Select a report type from the drop box 2. Click the “Print Report” button. | The selected report type is sent to the printer queue | The selected report type is sent to the printer queue | Pass |
| 81 | View report | 1. Select a report type from the drop box | The data table is filled with information based on the selected report type | The data table is filled with information based on the selected report type | Pass |

**Search Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 82 | Go back to the previous page | 1. Click the “Go Back” button. | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |
| 83 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 84 | Check for inventory search match | 1. Select an item type from the drop box 2. Choose a row from the inventory data grid 3. Fill in all text boxes underneath the inventory grid with matching information to the row | As data is filled, the grid filters rows until an exact match is found | As data is filled, the grid filters rows until an exact match is found | Pass |
| 85 | Check for order search match | 1. Choose a row from the orders data grid 2. Fill in all text boxes below the orders grid with matching information to the row | As data is filled, the grid filters rows until an exact match is found | As data is filled, the grid filters rows until an exact match is found | Pass |

**Storage Page**

| **Test Case** | **Description** | **Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 86 | Go back to the previous page | 1. Click the “Go Back” button. | The application returns to the Main Menu Page | The application returns to the Main Menu Page | Pass |
| 87 | Exit application | 1. Click the “Exit” button | The application returns to the Login page | The application returns to the Login page | Pass |
| 88 | Test item type change | 1. Select “Part” from the Item Type drop box 2. Select “Product” from the Item Type drop box | Upon selecting “Part” the inventory data grid is loaded with all Parts in the database. Upon selecting “Product” the inventory data grid is loaded with all Products in the database. | Upon selecting “Part” the inventory data grid is loaded with all Parts in the database. Upon selecting “Product” the inventory data grid is loaded with all Products in the database. | Pass |
| 89 | Test view storage location | 1. Select a storage location from the View Storage Location drop box | Upon selecting a storage location, the data grid view on the right fills with items that are stored at the selected location | Upon selecting a storage location, the data grid view on the right fills with items that are stored at the selected location | Pass |
| 90 | Test move location | 1. Select an item type from the drop box 2. Select a row from the inventory data table 3. Select a different location in the New Location drop box 4. Click the “Move Selected” button 5. Select the new location in the storage location drop box 6. Check to see if the item moved to the new location | The location of the item changes to the new selected location and is able to be seen in the storage location data table | The location of the item changes to the new selected location and is able to be seen in the storage location data table | Pass |

## Test Plan

### Items

### Items required to perform unit testing:

1. A computer workstation for each tester
2. Correct login information
3. Test results documentation
4. Developer(s) that are able to fix any issues found during testing
5. Quality assurance personnel (at least 1)

### Features

All tests are done through the user interface of the application. Validation is done by checking for matching data between the user interface and the database. Each unit must pass all unit tests before being integrated with other units.

### Deliverables

### Test Summary Report to document an overview of what happened during testing.

* Bug reports
* Unit test results

### Tasks

### Complete unit tests by following the steps written for each test

1. Report all bugs, unexpected results, and issues to developers
2. Developers fix reported problems
3. Retest application
4. Repeat until units pass all tests

### Needs

* Access to the test environment
* Visual studio 2019 or newer
* Access to the project solution file

### Pass/Fail Criteria

The test plan documents the expected results of each unit test. If the results of a unit test are different than the expected result, then that is considered a failure. If the results of a unit test are the same as the expected result, then that is considered passing. Bugs, discrepancies, issues, and failures are sent to the developers by using the Bug Report Document. The quality assurance unit tester fills out this form. It is then sent to the development team which assigns the bug to be fixed by a developer.

The contents of a bug report consist of

1. Bug ID
2. Environment
3. Steps to reproduce the bug
4. Expected result
5. Actual result
6. Visual proof of the bug
7. Severity level

After the bug is fixed by the developer, the QA tester is cleared to run the unit test again.

## Results

The unit tests were all done using manual testing with the user interface. After all unit tests are passed, a regression test was

**Source Code**

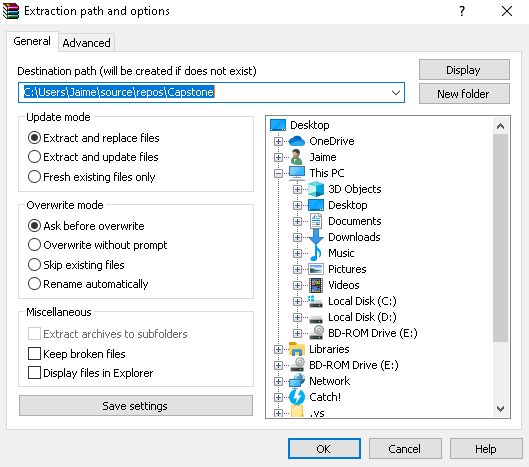
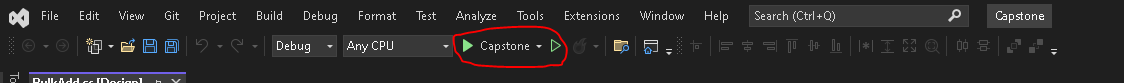
The source code for the software system will be included in an attached zip file. The following user guide details setup and instructions for using the application.

# User Guide

## Introduction

This guide is intended to inform all users of the application about how to use it correctly. Managers and regular users have different functionalities allowed to them. This guide will detail how to properly use the application for both types of users. Each function is labeled as a header and has step by step instructions in its relative section. To begin, the guide describes how to install the application.

## Installing the Application

1. Download the .zip file titled “Capstone”
2. Right-click Capstone.zip and select “Extract files”
3. Using the form that appears, extract all files to the default location
4. Navigate to the file that contains the extracted files by using the computers File Explorer
5. Double click the Capstone.sln file, and open it using Visual Studio
6. Click the green arrow at the top of Visual Studio or press F5 to begin running the program

## *Login*

*After opening the application, the user is presented with the Login Screen.*

*The default Username is 7 and the password is 123. Enter “7” into the username textbox and “123” into the password text box. Then click the “Log In” button.*

***Add Stock to Inventory***

1. *Navigate to the “Bulk Add” page.*

*Login ▶ Main Menu ▶ Bulk Add*

*For each item being added to inventory…*

1. *Select either “Inhouse Part”, “Outsourced Part” or “Product” from the Item Type dropbox.*
2. *Select the ID of the item you are adding from the second drop box, the name will auto-fill*
3. *Enter the quantity to add in the number box*
4. *Click the “Add to Table” button*

* *If an item added to the table needs to be removed, click on the item’s row to highlight it and then click the “Remove Selected” button.*
* *If an item added to the table needs to be edited, click on the item’s row to highlight it and then click the “Edit Selected” button.*

*After adding all items to the data grid, click the “Add All” button to increase the stock amount for all items that are in the data grid.*

***Remove Stock From Inventory***

1. *Navigate to the “Bulk Remove” page*

*Login ▶ Main Menu ▶ Bulk Remove*

1. *Select an item type from the “Item Type” drop box*
2. *Select the item from the data grid by clicking on it’s row*
3. *Enter the quantity to remove from stock in the “Quantity” number box*
4. *Click the “Remove” button*

*(Stock entered to be removed must be less than current amount in stock)*

***Move Storage Location of Inventory Item***

1. *Navigate to the “Storage” page*

*Login ▶ Main Menu ▶ Storage*

1. *Select an item type from the “Item Type” drop box*

*The “Inventory” data grid fills with all items of the selected type*

1. *Select an item from the “Inventory: data grid by clicking on the item’s row*
2. *Either enter a new location or select an existing location in the “New Location” combo box*
3. *Click the “Move Selected” button*

***View Items in Storage Location***

1. *Navigate to the “Storage” page*

*Login ▶ Main Menu ▶ Storage*

1. *Select a location from the “View Storage Location” drop box*

*The data grid on the right side of the page fills with all items that are stored at the selected storage location.*

***Search for Inventory Item***

1. *Navigate to the “Search” page*

*Login ▶ Main Menu ▶ Search*

1. *Select an item type from the “Item Type” drop box*
2. *Enter information into any of the “Part Id”, “Product ID”, “Name”, “Machine ID”, “Price”, or “Company Name” entry fields.*
3. *As information is entered, the “Inventory” data grid on the left side of the page will fill with items that match all entered information.*
4. *If the data grid is empty, then there are no matches found*

***Search for Order***

1. *Navigate to the “Search” page*

*Login ▶ Main Menu ▶ Search*

1. *Enter information into any of the “Order ID”, “Assigned User”, “Complete”, “Parts”, and “Products” entry fields*
2. *As information is entered, the “Orders” data grid on the right side of the page will fill with orders that match all the entered information*
3. *If the data grid is empty, then there are no matches found*

***View Assigned and Current Orders***

1. *Navigate to the “Orders” page*

*Login ▶ Main Menu ▶ Orders*

1. *Assigned orders to the current logged-in user will appear in the left data grid labeled “Assigned Orders”*
2. *The current order to the current logged-in user will appear in the right data grid labeled “Current Order”*

***Set Current Order***

1. *Navigate to the “Orders” page*

*Login ▶ Main Menu ▶ Orders*

1. *Select an order from the “Assigned Orders” data grid by selecting the order’s row*
2. *Click the “Set As Current Order” button*
3. *The “Current Order” data grid will update to contain the information for the new current order*

***Mark Current Order as Complete***

1. *Navigate to the “Orders” page*

*Login ▶ Main Menu ▶ Orders*

1. *Click the “Mark As Complete” button that is underneath the “Current Order” data grid*
2. *The “Current Order” data grid will clear and the stock for each item in the order will be adjusted*

***Edit Existing User***

1. *Navigate to the “Edit Users” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Edit Users*

1. *Select a user from the “User ID” drop box OR click a user’s row in the data grid then click the “Edit Selected” button*
2. *That user’s information will auto fill into the “First Name”, “Last Name”, “Password”, “Creation Date”, and “Is Manager” entries.*
3. *Change the selected user’s information in their corresponding entries.*
4. *Ensure that there are no missing or empty entries*
5. *Click the “Save” button*

*The user’s information will be updated and the updates will be reflected in the data grid.*

***Remove Existing User***

1. *Navigate to the “Edit Users” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Edit Users*

1. *Select a user from the data grid by clicking on that user’s row*
2. *Click the “Remove Selected” button*

*The user will be removed from the database and the changes will be reflected in the data grid.*

***Add New User***

1. *Navigate to the “Add Users” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Add Users*

1. *Enter information into the “First Name”, “Last Name”, “Password”, “Creation Date”, and “Is Manager” entries*
2. *Ensure that there are no missing or empty entries*
3. *Click the “Save” button*

*The new user will be added to the database and the changes will be reflected*

***Edit Existing Order***

1. *Navigate to the “Edit Orders” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Edit Orders*

1. *Select an order from the “Order ID” drop box OR select an order from the data grid by clicking on the order’s row then clicking the “Edit Selected” button*

*The order’s information will autofill into the “User ID”, “Creation Time”, “Complete”, “Completion Time”, “Parts”, “Products”, “Missing Parts” and “Missing Products” fields.*

1. *Edit the order’s information in their corresponding fields*
2. *Click the “Save” button*

*The order’s information is saved to the database and the changes are reflected in the data grid.*

***Remove Existing Order***

1. *Navigate to the “Edit Orders” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Edit Orders*

1. *Select an order from the data grid by clicking on the order’s row*
2. *Click the “Remove Selected” button*

*The order is removed from the database and the changes are reflected in the data grid.*

***Add New Order***

1. *Navigate to the “Add Order” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Add Order*

1. *Select an existing user from the “User ID” drop box*

*For each item to be added to the order…*

1. *Select a part or product name from their drop boxes*
2. *Enter the quantity for the selected item in the item’s row*
3. *Click the “Add to Table” button in the same row as the item*

*The item’s name and quantity will be added to the table.*

*If an item needs to be removed, click the item’s row in the data table, then click the “Remove Selected” button.*

1. *Click the “Add Order” button*

*A new order will be added to the database with the data grid’s information to the selected User ID and the data grid will clear.*

***Edit Existing Inventory Item***

1. *Navigate to the “Edit Inventory” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Edit Inventory*

1. *Select an item type from the “Item Type” drop box*
2. *If the item type is a part, select a part ID from the “Part ID” drop box*

*If the item is a product, select a product ID from the “Product ID” drop box*

*OR click on an item’s row from the data grid view then click the “Edit Selected” button.*

*The selected item’s information will fill all entries.*

1. *Edit the item’s information in it’s relevant entries*
2. *Ensure that no entries are missing or empty*
3. *Click the “Save” button*

*The item’s information is changed in the database and the data grid will reflect the changes.*

***Remove Existing Inventory Item***

1. *Navigate to the “Edit Inventory” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Edit Inventory*

1. *Select the type of item that needs to be removed from the “Item Type” drop box*
2. *Select the item from the data grid by clicking the item’s row*
3. *Click the “Remove Selected” button*
4. *A confirmation message will appear, Click the“Yes” option to remove the item*

*The item will be removed from the database and the data grid will reflect the changes.*

***Add New Inventory Item***

1. *Navigate to the “Add Inventory” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Add Inventory*

1. *Select the type of item to be added from the “Item Type” drop box*
2. *Fill in all fields for the item*
3. *Ensure that no entries are missing or empty*
4. *Click the “Save” button*

*The item is added to the database and the data grid will reflect the changes.*

***View and Print Reports***

1. *Navigate to the “Reports” page*

*Login ▶ Main Menu ▶ Manager Functions ▶ Reports*

1. *Select a report type from the “Report Type” drop box*

*The data grid will fill with the report’s data*

1. *To print the report, click the “Print Report” button*