Who’s Joe? Joe Coffee

Project Team: Matthew Malzahn, Henry Reichard, Mikael Williams

**1. Project Definition**

The general purpose of this project is to create a successful system that allows the users to quickly and efficiently clock in/out, view their schedules, request days off, manage schedules, manage inventory, and update data such as: employees, inventory counts, schedules, and pay. This allows for easy manipulation of data within a work place that helps move things smoothly on the business side of things.

Many stores across the country still rely of ancient systems to perform these functions. While some of these systems might still work like they always have, it may pose a challenge in continued maintenance over a long period of time and with training new users to the systems. Thus, this project’s goal is to bring store’s management system back into the 21st century by creating a simple, efficient, and elegant way to handle employee data and inventory.

The way the project will be handled is by using a web-based interface created with HTML5/Bootstrap and CSS to create an elegant way for users to navigate the data that will be stored in a database using phpMyAdmin that is linked with Amazon AWS to create a server that may be accessed remotely. This allows for versatility in where the project can be used, rather than have it limited to one local computer.

The project scope will include, but will not be limited to:

* Creating an efficient means for management users to access and adjust employee data
* Allow managers to access store’s inventory and manipulate data in efficient manner
* Allow employee users to perform basic work functions such as clocking in/out, view schedules, and request days off

**2. Project Requirements**

The functional requirement of the system is that is must be able to handle the requests of the users, either manager or employee. It must be able to retrieve inventory, employee, and schedule data and allow the singed-in manager to be able to view, manipulate, and create data as needed. The system must also be able to retrieve schedule times, have ability to log clock in and clock out times, and to log requests off to prevent management to schedule an employee on a particular day for a general employee.

The user interface will be a web-based GUI built using HTML/Bootstrap and CSS to create an elegant and easily navigated interface that allows the employee or manager to efficiently and intuitively conduct their work needs in a smooth manner.

The system will be versatile, able to be accessed by most modern operating systems on most modern hardware. There are no specific requirements of hardware needed to access the project. The computer being used only needs internet access to be able to access the server which holds the data through a specific secure link.

The database is running from phpMyAdmin on a server through Amazon AWS.

The security of the system includes using php to sanitize inputs to prevent SQL injection attacks, and the users will be required to input correct credentials to access the system. There will also be hierarchical privileges that separate what managers and employees can do without allowing manager functions to be accessed by general employees that do not have clearance.

**3. Project Specification**

The domain for this system retail businesses and its employees. More specifically, this system enables a business’s employees to perform work related tasks with ease and with a small learning curve. The purpose of the domain analysis is ensuring our application provides a quick, efficient, and reliable service to a business (or in this case, a hypothetical popular coffee shop) and to relieve the workload of its employees.

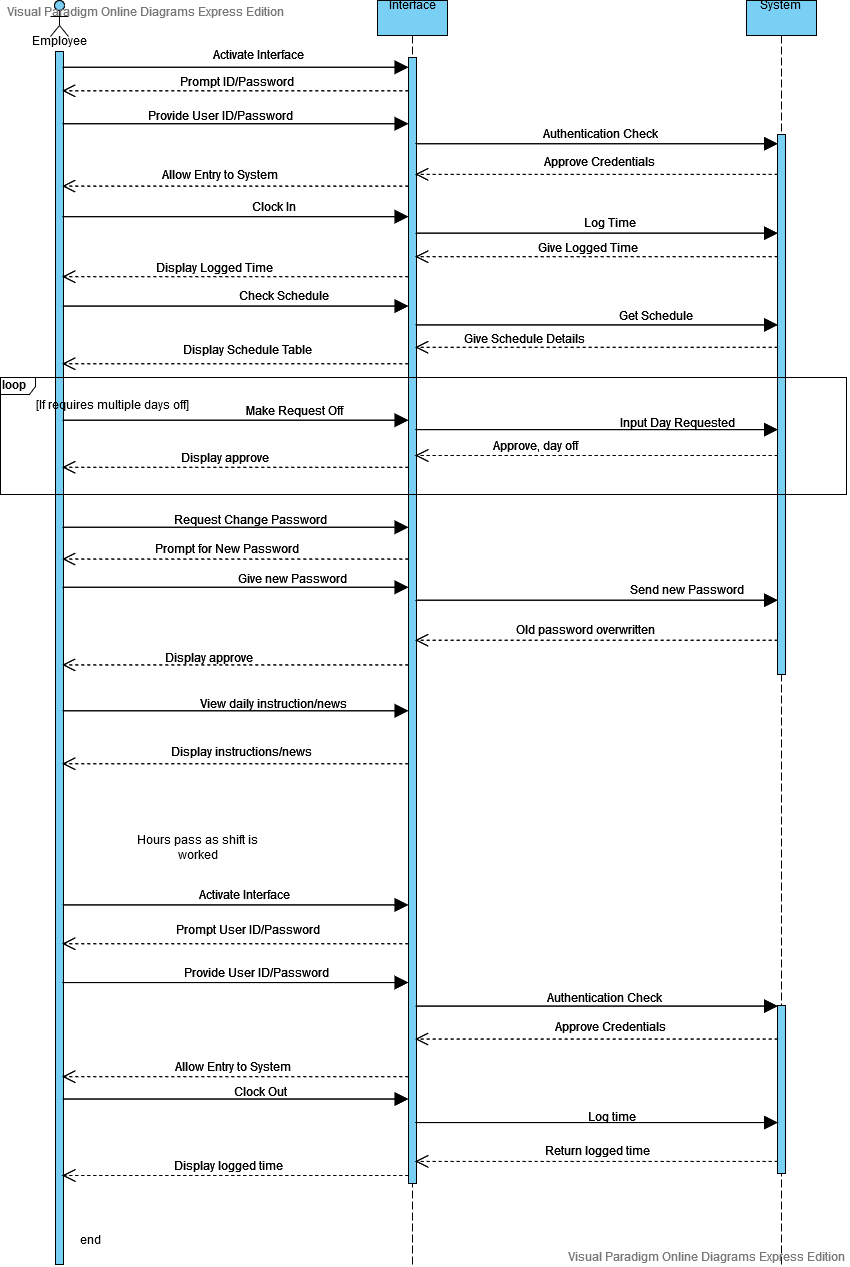
The project will use HTML/Bootstrap/CSS as the framework for the web-based interface, the development environment is LAMPS with Maria SQL for the database, Amazon EC2 Instance on Linux 2 for the server, and phpStorm and Notepad++ as development environments.

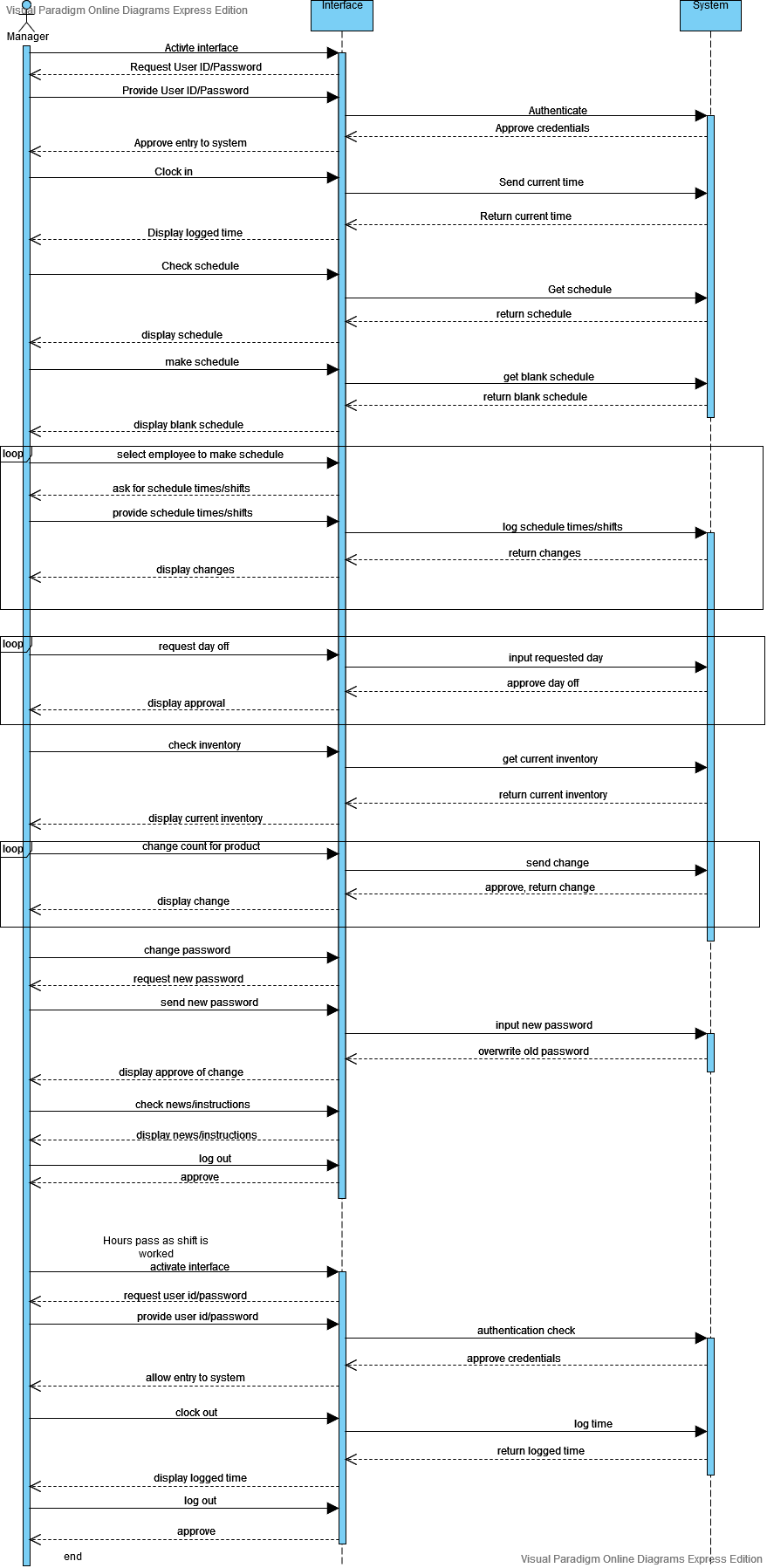
This project is meant to be a desktop application.

This is a business application.

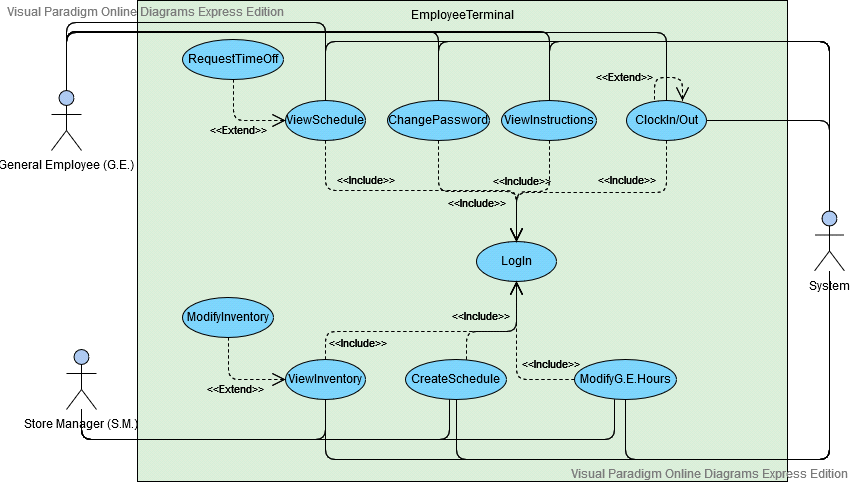
**4. System – Design Perspective**

The subsystems of the project are: the employee features, manager features, and the database management.

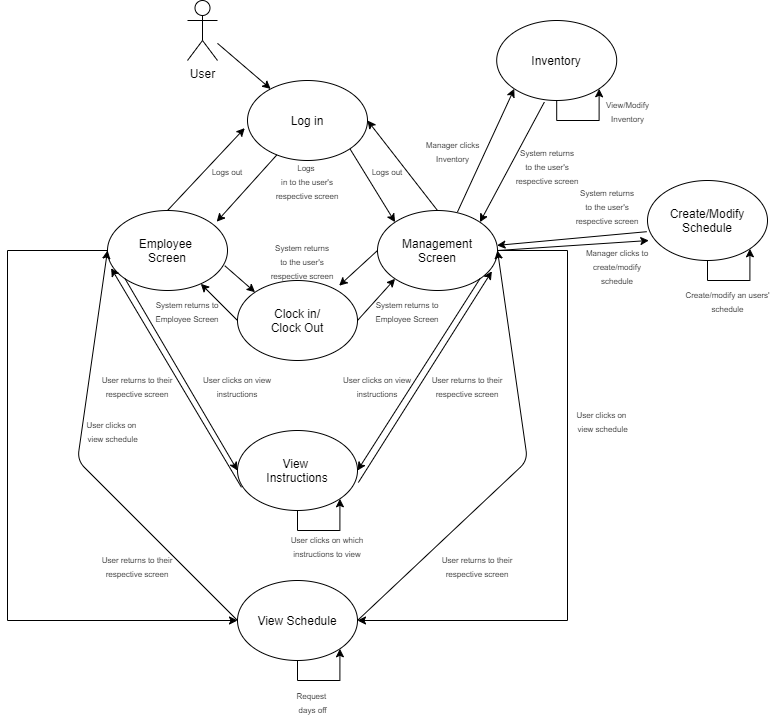
**Employee SequenceDiagram:**

**Manager Sequence Diagram:** 

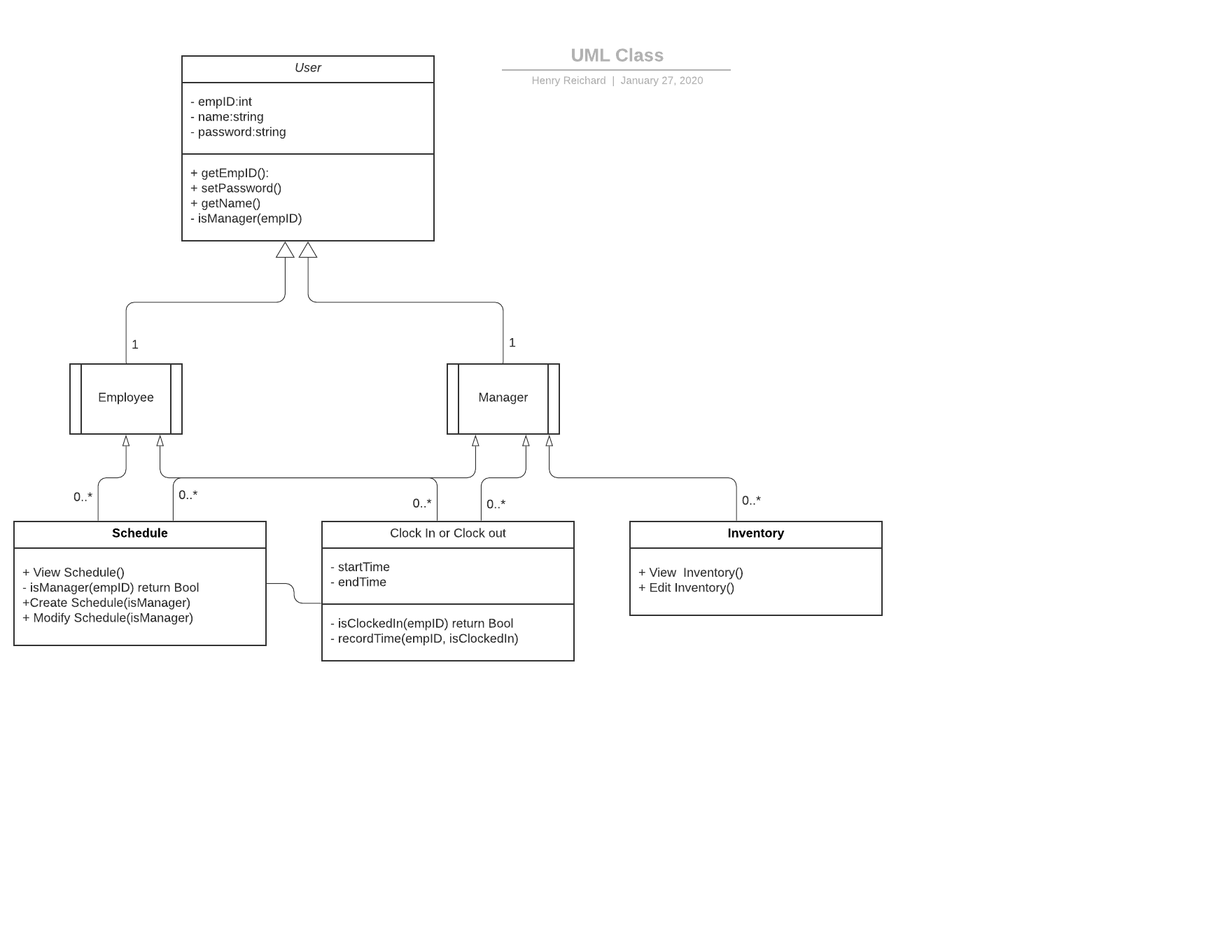
**Use Case Diagram:**



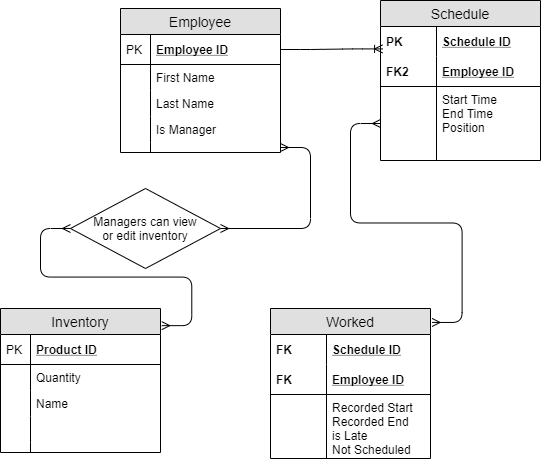
**State Diagram**



**Class Diagram:**



**ER Diagram for the database:**



**5. System – Analysis Perspective**

**Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Data Format | Field Size | Description | Example | Required? |
| EmployeID | Int | xxxx | 4 | Unique ID for employees | 4609 | Yes |
| ScheduleID | Int | xxxxxxxxxx | 11 | Unique ID for Schedule. It’s in Unix time for the first day of the work week at midnight | 1580706000 | Yes |
| First Name | varchar |  | 12 | First name of employee | Brody | Yes |
| Last Name | Varchar |  | 16 | Last name of employee | Gore | Yes |
| Position | Char |  | 10 | Position of employee | Cashier | Yes |
| isManager | Boolean |  | 1 | Checks if employee is manager | True | Yes |
| Start\_time | Int | xxxxxxxxxx | 11 | Unix time for schedule start time | 1580706000 | Yes |
| End time | Int | xxxxxxxxxx | 11 | Unix time for scheduled end time | 1580706000 | Yes |
| Recorded Start | Int | xxxxxxxxxx | 11 | Records employee time when the employee clocks in. Its in Unix time. | 1580706000 | Yes |
| Recorded End | int | xxxxxxxxxx | 11 | Records employee time when the employee clocks out. It’s in Unix time. | 1580706000 | Yes |
| isLate | Boolean |  | 1 | Marks True if employee is late | True | No |
| isScheduled | Boolean |  | 1 | Marks True if employee is scheduled to work at the time of clocking in/out | True | No |
| ProductID | Int | Xxxxxx | 6 | Unique product ID | 123456 | Yes |
| Product name | Char |  | 8 | Product’s name | Joe’s Coffee | No |
| Quantity | Int | Xxxx | 4 | Quantity of the product | 200 | Yes |

**Time Complexity**

Overall, the whole system will have an O(n) time complexity. For some functionalities as in looking up a user’s schedule, requesting day off, and clocking in and out it will be O(1). This is because the system can directly index to the employeeID and scheduleID. For sorting through inventory, all schedules, and all employees then it will be O(N). Thus, the whole system time complexity will be O(n), since the data is small for this project and probably will be small in a small business application then this won’t matter. With applications with big data, a sorting algorithm will be needed.

**6. Project Scrum Report**

**7. Subsystems**

**7.1. Employee Usage**

**7.2. Manager Usage**

**7.3. System Usage**