UNIBOOK
EMPLOYEE MANAGEMENT SYSTEM

GROUP 8

Reuben Reji
Praharsh Deep Singh
Shashank Shri Venkatesh
Parika Rawat
Hridyansh Gupta
Rituparna Desai

Index

Executive Summary

This document outlines the details of the proposed Employee Management System (EMS) called UniBook EMS. The system is intended to enhance the existing processes of employee shift scheduling, task tracking, employee discount verification, on-shift employee tracking, and inventory management for the University Book Center, a retail bookstore located on the campus of the University of Maryland in College Park, Maryland, USA. With the implementation of UniBook EMS, the University Book Center expects to see significant improvements in employee productivity, customer service, and overall store operations.

The team responsible for the development of UniBook EMS will use a parallel waterfall project methodology to ensure that the project is completed within the expected timeline of 89 days. This methodology involves dividing the project into multiple phases and working on some phases simultaneously. The parallel approach will enable the team to complete the project quickly and efficiently while ensuring that each phase is fully developed and meets all project requirements.

The UniBook EMS will be developed with the aim of improving the employee management processes of the University Book Center. The system will automate many of the manual processes currently in use and provide an easy-to-use interface for employees, supervisors, and managers. UniBook EMS will also provide real-time data analysis and reporting, allowing management to make informed decisions based on accurate and up-to-date inventory information. With the implementation of UniBook EMS, the University Book Center will be better equipped to manage employee schedules, track tasks, and monitor inventory levels, authenticate employee discounts and track on-shift employee attendance.

Introduction

The creation of UniBook EMS, an employee management system, is the main objective of this project. Its purpose is to provide managers, supervisors, and employees of the University Book Center with a comprehensive tool for scheduling, task tracking, inventory management, employee discounts verification, and on-shift employee tracking. The system will streamline the current processes and enhance the efficiency of the bookstore's operations.

System Planning Phase Report

1.0 Client and Industry Background

The University Book Center is a retail bookstore located on the campus of the University of Maryland in College Park, Maryland, USA. The bookstore offers a wide selection of textbooks,

course materials, and other academic supplies, as well as a variety of general interest books, apparel, and gifts. It serves the students, faculty, and staff of the University of Maryland who are interested in purchasing academic materials or supporting the university through the purchase of branded merchandise. The bookstore is operated by Follett Higher Education Group, which is a leading provider of educational products and services to institutions of higher learning throughout the United States and Canada.

The University Book Center employs a mix of full-time and part-time workers to fulfill its operational needs. Full-time employees typically hold managerial or supervisor positions. Part-time employees, typically college students who work on flexible schedules, are responsible for customer service and sales, and inventory management.

1.1 Opportunity

One of the main objectives of any bookstore is to provide high-quality products and services at competitive prices and maximize profits. The University Book Center, as the sole bookstore on campus, has a unique opportunity to serve the needs of students, faculty, and staff of the University of Maryland.

However, there are various areas where the bookstore can improve its current processes to ensure customer satisfaction, streamline employee scheduling, and enhance inventory management. Furthermore, implementing an employee discount verification system would be beneficial to minimize the risk of loss due to fraudulent transactions.

With this in mind, it was decided that a centralized management system for all University Book Center employees should be pursued.

1.2 Project Objective

The main objective is to develop a mobile app and a computer-based application for the university bookstore that streamlines employee scheduling, task tracking, employee discounts verification, on-shift employee tracking and inventory management.

1.3 Project Scope

This application will provide a user-friendly interface for bookstore employees to easily view and manage their work schedule as well as allow managers/supervisors to efficiently assign shifts and track employee’s task progress. Additionally, the app will integrate with the bookstore's Billing System to provide employee discounts on selected items and reduce fraudulent transactions. Overall, this project objectively aims to increase operational efficiency and customer satisfaction at the university bookstore through the development of a customized mobile app for both iOS and Android platforms as well as a Windows/ Mac application for ease of access for the Manager/Supervisors.

2.0 Project Methodology: Parallel Development

Based on the project requirements and the Work Breakdown Structure (WBS), parallel waterfall methodology would be the most suitable approach for this project.

The WBS is a visual representation of the project's tasks, deliverables, and dependencies. It highlights how the tasks are interrelated and the order in which they must be completed. By examining the WBS, it becomes evident that certain tasks must be completed before others can commence. For example, before development can begin, the requirements gathering phase must be completed. Similarly, before testing can be performed, the development phase must have been finished.

In this project, the parallel waterfall methodology is the most appropriate choice because it enables the team to work on multiple tasks concurrently without sacrificing quality. For example, after gathering requirements, the application development - front-end and back-end will happen concurrently. This approach ensures that there is a continuous flow of work and reduces the risk of delays caused by completing each task one after the other.

Additionally, the parallel waterfall methodology is a suitable choice because of the project's rigid requirements and scope. The client has specified precise requirements that must be met, and there is little room for error or deviation. The parallel waterfall methodology's structured approach ensures that the project's goals are met within the required timeline and budget.

In conclusion, based on the WBS analysis and the project's requirements, the parallel waterfall methodology is the best approach. Its ability to work on multiple tasks simultaneously without sacrificing quality and its structured approach are well-suited to the project's rigid requirements and scope.

Ability to develop systems with…	**Waterfall**	**Agile**	**Parallel Development**
Clear User Requirements	Poor	Good	Excellent
Customer Feedback	Poor	Good	Excellent
Collaborative Approach	Poor	Good	Excellent
A Short Time Constraint	Poor	Poor	Excellent

3.0 Project Plan

		WBS	Name	Duration	Start	Finish
1		**1**	**UniBook Employee Management System**	**89 days**	**4/5/23 8:00 AM**	**8/7/23 5:00 PM**
2		**1.1**	**Planning**	**14 days**	**4/5/23 8:00 AM**	**4/24/23 5:00 PM**
3		1.1.1	Plan Project Initiation Meeting	2 days	4/5/23 8:00 AM	4/6/23 5:00 PM
4		1.1.2	Communication Plans	2 days	4/7/23 8:00 AM	4/10/23 5:00 PM
5		1.1.3	Establish Project Steering Committee	2 days	4/11/23 8:00 AM	4/12/23 5:00 PM
6		1.1.4	Identify Primary Stakeholders	3 days	4/13/23 8:00 AM	4/17/23 5:00 PM
7		1.1.5	Get Charter Approved	5 days	4/18/23 8:00 AM	4/24/23 5:00 PM
8		**1.2**	**Analysis**	**12 days**	**4/18/23 8:00 AM**	**5/3/23 5:00 PM**
9		**1.2.1**	**Establish Requirements**	**2 days**	**4/18/23 8:00 AM**	**4/19/23 5:00 PM**
10		1.2.1.1	Employee Requirements	2 days	4/18/23 8:00 AM	4/19/23 5:00 PM
11		1.2.1.2	Manager Requirements	2 days	4/18/23 8:00 AM	4/19/23 5:00 PM
12		1.2.1.3	Supervisor Requirements	2 days	4/18/23 8:00 AM	4/19/23 5:00 PM
13		1.2.2	Integrations Analysis	2 days	4/20/23 8:00 AM	4/21/23 5:00 PM
14		1.2.3	Develop Use Cases	4 days	4/24/23 8:00 AM	4/27/23 5:00 PM
15		1.2.4	Review Use Cases	2 days	4/28/23 8:00 AM	5/1/23 5:00 PM
16		1.2.5	Approve Use Cases	2 days	5/2/23 8:00 AM	5/3/23 5:00 PM
17		**1.3**	**Design**	**20 days**	**5/4/23 8:00 AM**	**5/31/23 5:00 PM**
18		1.3.1	Develop Database Design	5 days	5/4/23 8:00 AM	5/10/23 5:00 PM
19		**1.3.2**	**Application Design**	**10 days**	**5/11/23 8:00 AM**	**5/24/23 5:00 PM**
20		1.3.2.1	Develop Front-end Design	10 days	5/11/23 8:00 AM	5/24/23 5:00 PM
21		1.3.2.2	Develop Back-end Design	10 days	5/11/23 8:00 AM	5/24/23 5:00 PM
22		1.3.3	Testing	5 days	5/25/23 8:00 AM	5/31/23 5:00 PM
23		**1.4**	**Implementation**	**48 days**	**6/1/23 8:00 AM**	**8/7/23 5:00 PM**
24		**1.4.1**	**Training**	**5 days**	**6/1/23 8:00 AM**	**6/7/23 5:00 PM**
25		1.4.1.1	User Training	5 days	6/1/23 8:00 AM	6/7/23 5:00 PM
26		1.4.2	Deployment	3 days	6/8/23 8:00 AM	6/12/23 5:00 PM
27		1.4.3	Maintenance	5 days	6/13/23 8:00 AM	6/19/23 5:00 PM
28		1.4.4	Operations	5 days	6/20/23 8:00 AM	6/26/23 5:00 PM
29		1.4.5	Track Performance	5 days	6/27/23 8:00 AM	7/3/23 5:00 PM
30		1.4.6	Feedback Analysis	10 days	7/4/23 8:00 AM	7/17/23 5:00 PM
31		1.4.7	Contingency Plan (Roll Back Plan)	10 days	7/18/23 8:00 AM	7/31/23 5:00 PM
32		1.4.8	Conversion Strategies	5 days	8/1/23 8:00 AM	8/7/23 5:00 PM

System Analysis Phase Report

4.0 Requirements

The processes that we will be improving/integrating in our system are the following:

1. Manage Shift Scheduling
2. Manage Task Assignment
3. Employee Discount Verification
4. On-Shift Employee Tracking
5. Inventory Management System

The Requirements for the above processes are mentioned below:

4.1 Business Requirements:

- System should allow the manager to assign shifts according to the employee schedule.
- System should allow the employee to view all the available shifts.
- System should allow the employee to enter the preferences of their shifts.
- System should be able to store information regarding employee work authorization.
- System should allow the manager to put up task details.
- System should generate reports of employee attendance.
- System should ensure that employee discounts are easy to apply and prevent misuse.
- System should allow employees to exchange shifts with other employees.
- System should alert the manager/supervisor if a product count is low.

4.2 User Requirements:

4.2.1 Employees:

- User should be able to view the tasks assigned to them.
- User should be able to view the shifts assigned to them after approval.
- User should be able to choose the time slots when they want to work.
- User should be able to send/ receive messages from other employees/ supervisors/ manager.
- User should be able to update the status of the task assigned to them.
- User should be able to view the task assignment details.
- User should be able to avail employee discounts.
- User should be able to request a discount, and authenticate the transaction.
- User should receive an alert if they are outside the store for more than 15 minutes during their shift.

- User should be able to view their break timings.

4.2.2 Supervisors:

- User should be alerted when employees drop/exchange shifts.
- User should have the ability to mark a task as completed.
- User should be able to assign a particular task to an employee.
- User should get an alert when the stock of an item on the floor is below threshold.
- User should be able to assign break timings.
- User should be able to assign tasks and shifts.
- User should be able to view the task completion updates and the task list.
- User should receive an alert if the employee assigned to them are outside the store for more than 15 minutes during their shift.

4.2.3 Manager:

- User should be able to assign a particular task to a supervisor.
- User should be able to approve the timeslot chosen by a particular employee.
- User should be able to see the attendance records of the employee.
- User should be able to approve the exchange of shifts initiated by employees.
- User should be able to send/receive messages from employees/supervisors.
- User should be able to customize the low stock alert threshold for each product.
- User should get an alert when the stock of an item on the floor is below threshold.
- User should be able to assign tasks and shifts.
- User should be able to view the task completion updates and the task list.
- User should receive an alert if an employee is outside the store for more than 15 minutes during their shift.

4.3 Functional Requirements:

4.3.1 Process Requirements

- The system should be able to integrate employee work schedules and availability.
- The system should be able to send work schedules to employees and track their progress.
- The inventory management system should alert the supervisor and manager if the stock quantity of any particular product goes below the threshold on the floor.
- The system should be able to track employee hours.
- The system should restrict the number of hours scheduled by each employee based on their work authorization.
- The system should send an alert to the manager 15 mins after an employee's shift has started and they are not present/signed in.

4.3.2 Information Requirements:

- The system should be able to store employee data.
- The system should be able to store employee work hours and location.
- The system should be able to store employee details and verify discounts.

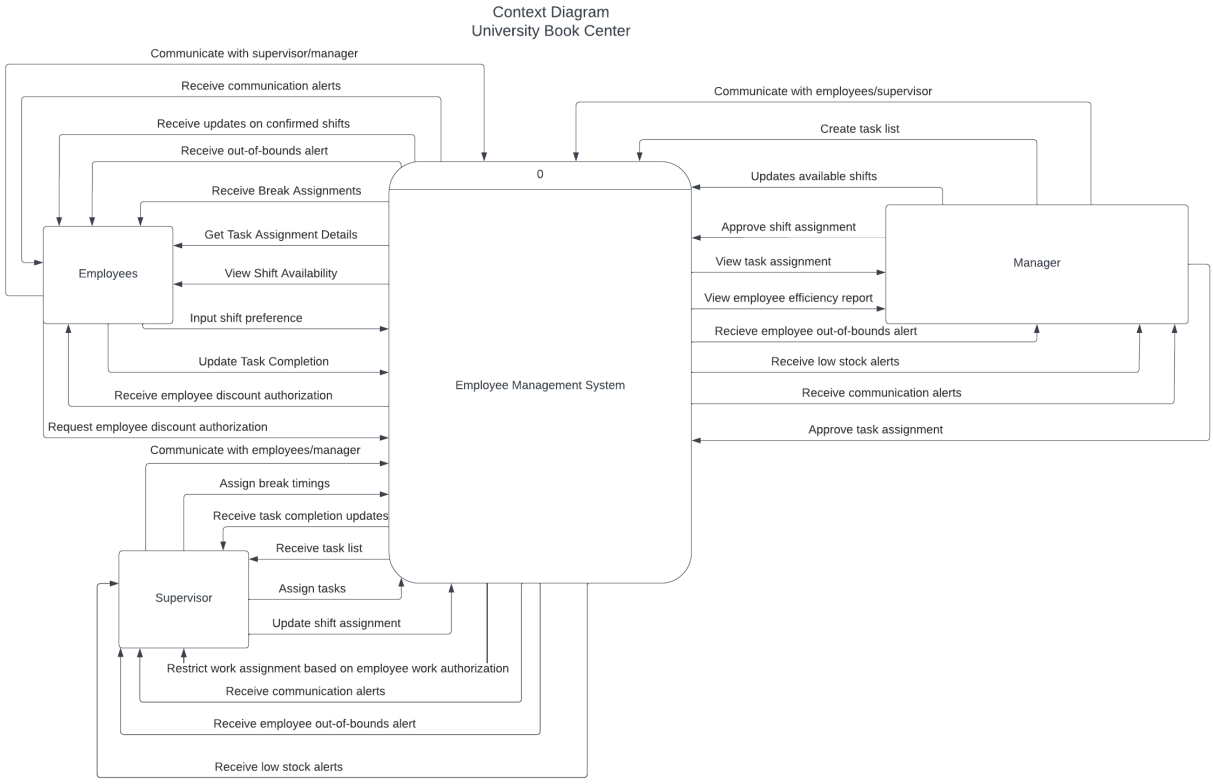
- The system should be able to provide real-time inventory data.
- The system should be able to store package data and track its location.
- The system should be able to store messages between the employees.

4.4 Non-Functional Requirements:

- Security & Privacy: Different levels of access based on specific permissions and roles.
- Personal information is protected in compliance with existing laws and regulations.
- The system should be fast and responsive to user requests.
- The system should be compatible with existing hardware and software at the University Book Center.
- The system should be available 24/7 and should always be able to handle at least 100 users.
- The system should store the employee's location only when they are on their shifts.

5.0 Data Flow Diagrams

5.1 Context Diagram



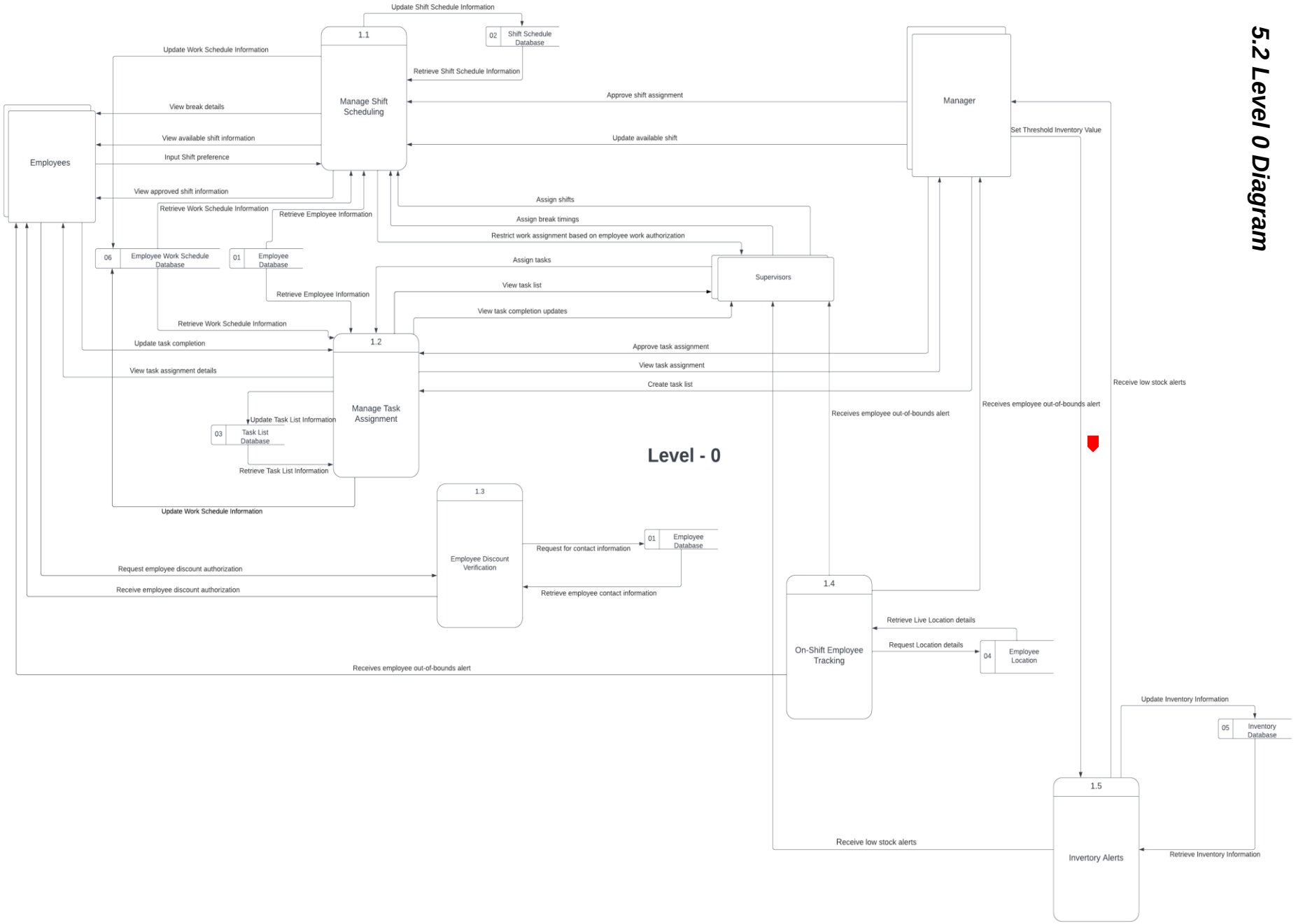
External Entities:

Employees(Staff): Represents a current part-time (student) employee working at the University Book Center.

Supervisors: Represents higher-level employees who manage different sections and staff.

Manager: Represents our beloved manager Doug Bunk who is the heart and soul of the University Book Center.

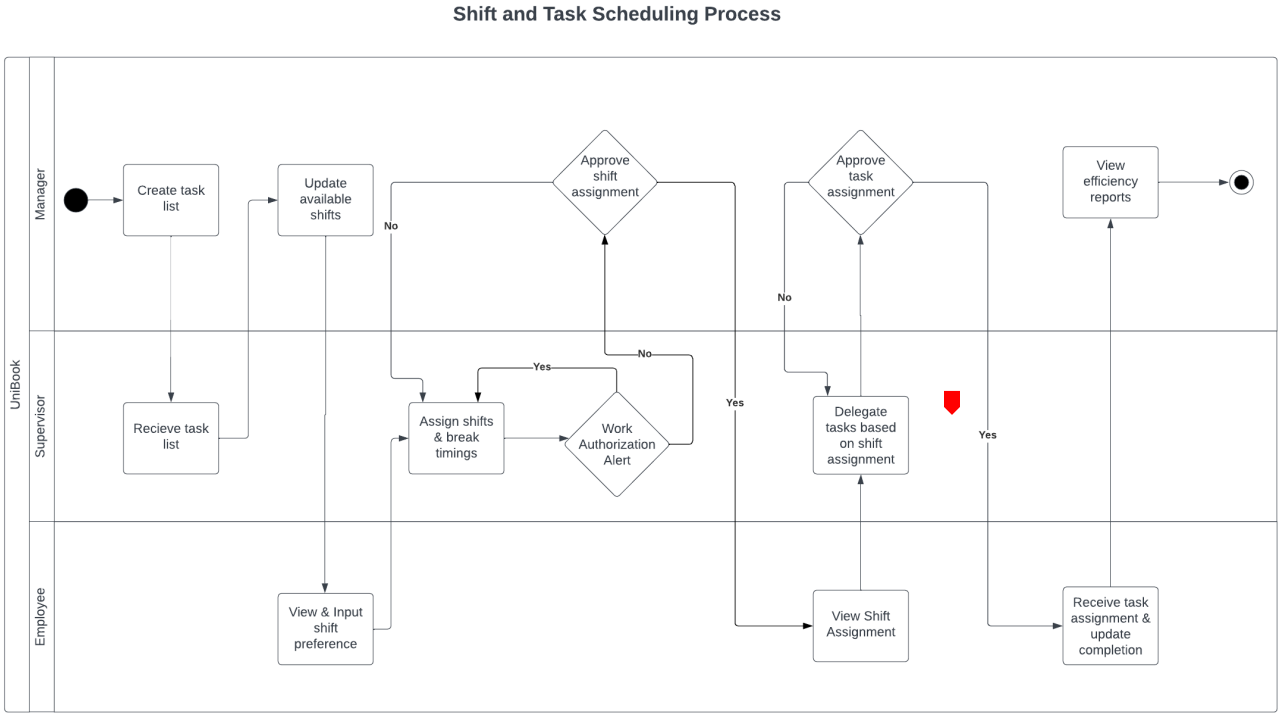
5.2 Level 0 Diagram



Description: There are a total of 5 processes in the level 0 diagram, namely Manage Shift Scheduling, Manage Task Assignment, Employee Discount Verification, On-Shift Employee Tracking, and Inventory Alerts on which actions are performed inside the system. There are 3 external entities namely Employee, Supervisor and Manager which receive data to and from the system and actions are performed outside the system. There are 6 data stores in the system namely Employee database, Shift Schedule database, Task list database, Employee location, Employee Work Schedule database and Inventory database where most of the information systems capture data for later use.

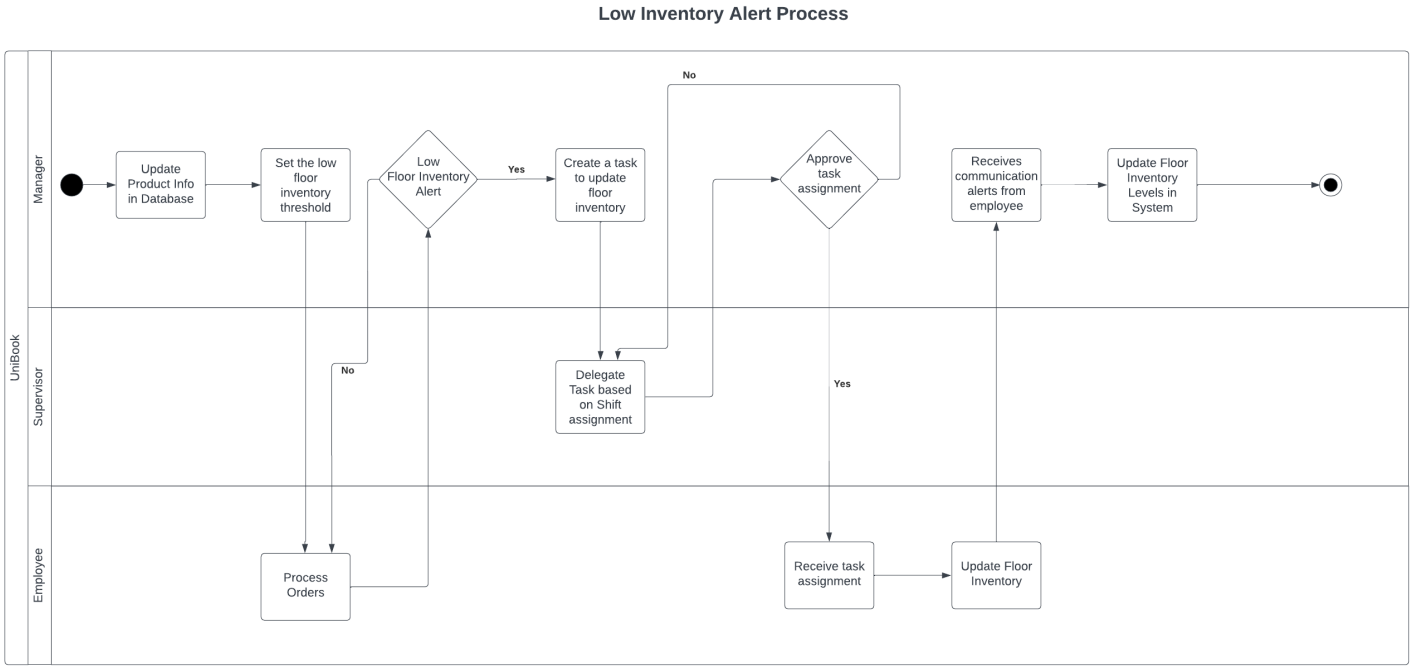
6.0 Swimlane Diagrams:

6.1 Employee Shift and Task Scheduling Process:



Description: The swimlane process flow diagram follows the process of a manager & supervisor who are scheduling and assigning shifts and tasks for the employees working at the University Book Center. The Manager can create shifts, employees can choose their preferred shifts, and supervisors can assign tasks. This process involves the manager, the supervisors, and the floor employee staff.

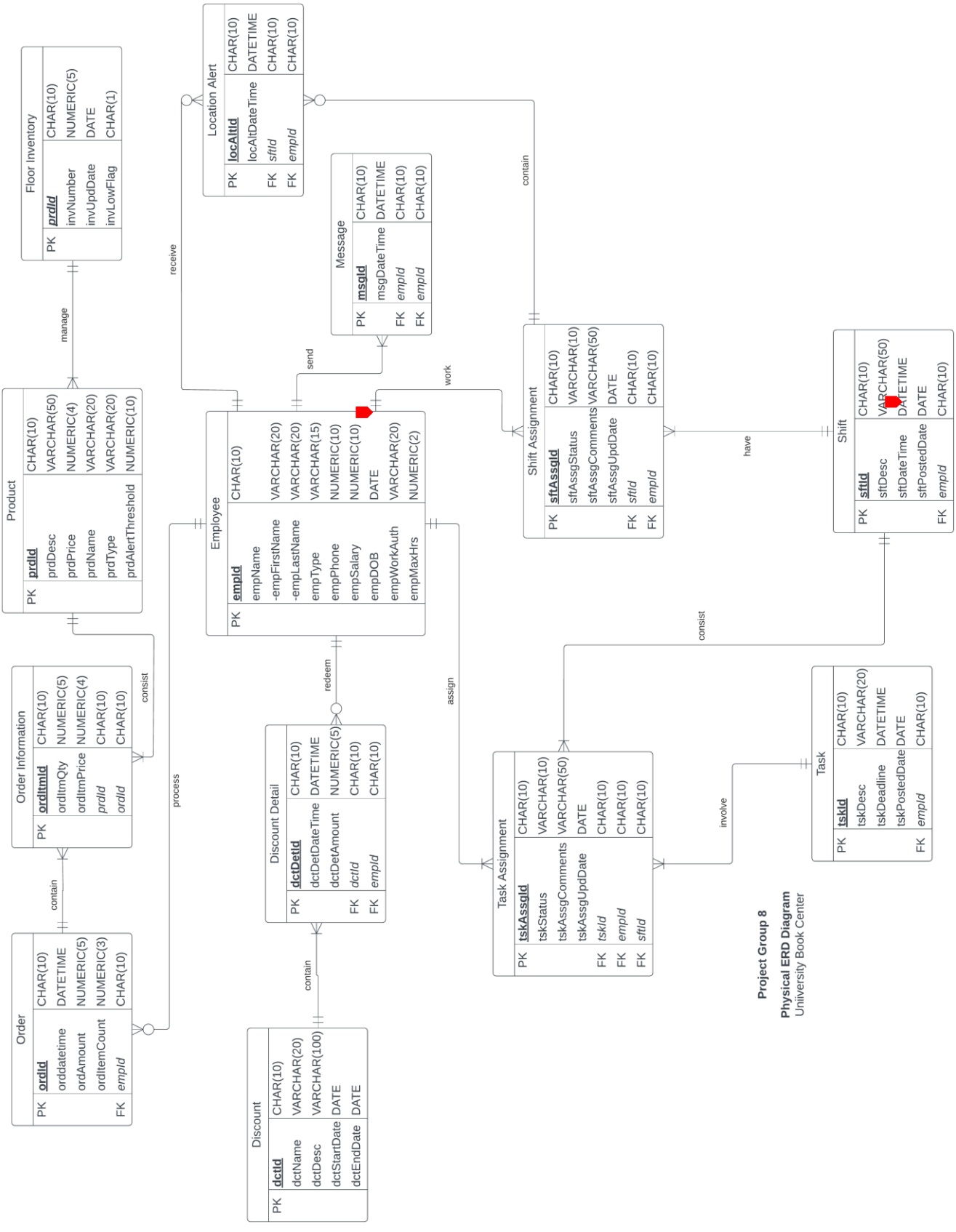
6.2 Low Inventory Alert Process:



Description: The swimlane process flow diagram follows the process of a manager setting up low thresholds for each product and receiving an alert when the inventory of a particular product goes below the threshold value at the University Book Center. This process involves the manager, the supervisors, and the floor employee staff.

System Design Phase Report

7.0 Physical ER Diagram



Description: There are a total of 13 entities in the physical ER diagram, namely Employee, Task, Shift, Task Assignment, Shift Assignment, Discount, Discount Detail, Message, Location Alert, Order, Order Item, Product, Floor Inventory. The relationship between all entities is indicated by the cardinality.

7.1 Entities

● Employee (Primary Key = empId) Contains information regarding an employee working at the University Book Center like name, Type, contact information and work authorization details.
● Task (Primary Key = tskId) Contains information regarding tasks like description, deadline, employee who assigned the task, and the date it was posted.
● Shift (Primary Key = sftId) Contains information regarding shifts like description, time, date, employee who assigned the shift, and the date it was posted.
● Task Assignment (Primary Key = tskAssgId) Contains information regarding task assignment like which task was assigned, the shift in which the task is to be completed, status, comments, and the date it was updated.
● Shift Assignment (Primary Key = sftAssgId) Contains information regarding shifts like which shift was assigned, the employee working in that shift, status, comments, and the date it was updated.
● Discount (Primary Key = dctId) Contains information regarding discounts like name, description, and the start and end date.
● Discount Detail (Primary Key = dctDetId) Contains information regarding discounts like the discount which was applied, employee who availed the discount, date, time and the amount.
● Message (Primary Key = msgId) Contains information regarding messages like the messages between the employees, date, and time.
● Location Alert (Primary Key = locAltId) Contains information regarding Location Alerts like the employee whose location is being tracked, shift, date, and time.
● Floor Inventory (Primary Key = prdId) Contains information regarding the floor inventory like the product, number of units, updated date and low threshold flag.
● Order (Primary Key = ordId) Contains information regarding the Orders like the order date, time, amount, item count and employee who processed the order.
● Order Item (Primary Key = ordItmId) Contains information regarding the Order Items like the product, order ID, quantity, and price.
● Product (Primary Key = prdId) Contains information regarding Products like the product ID, product description, price, name, type, and product alert threshold.

8.0 Design Architecture

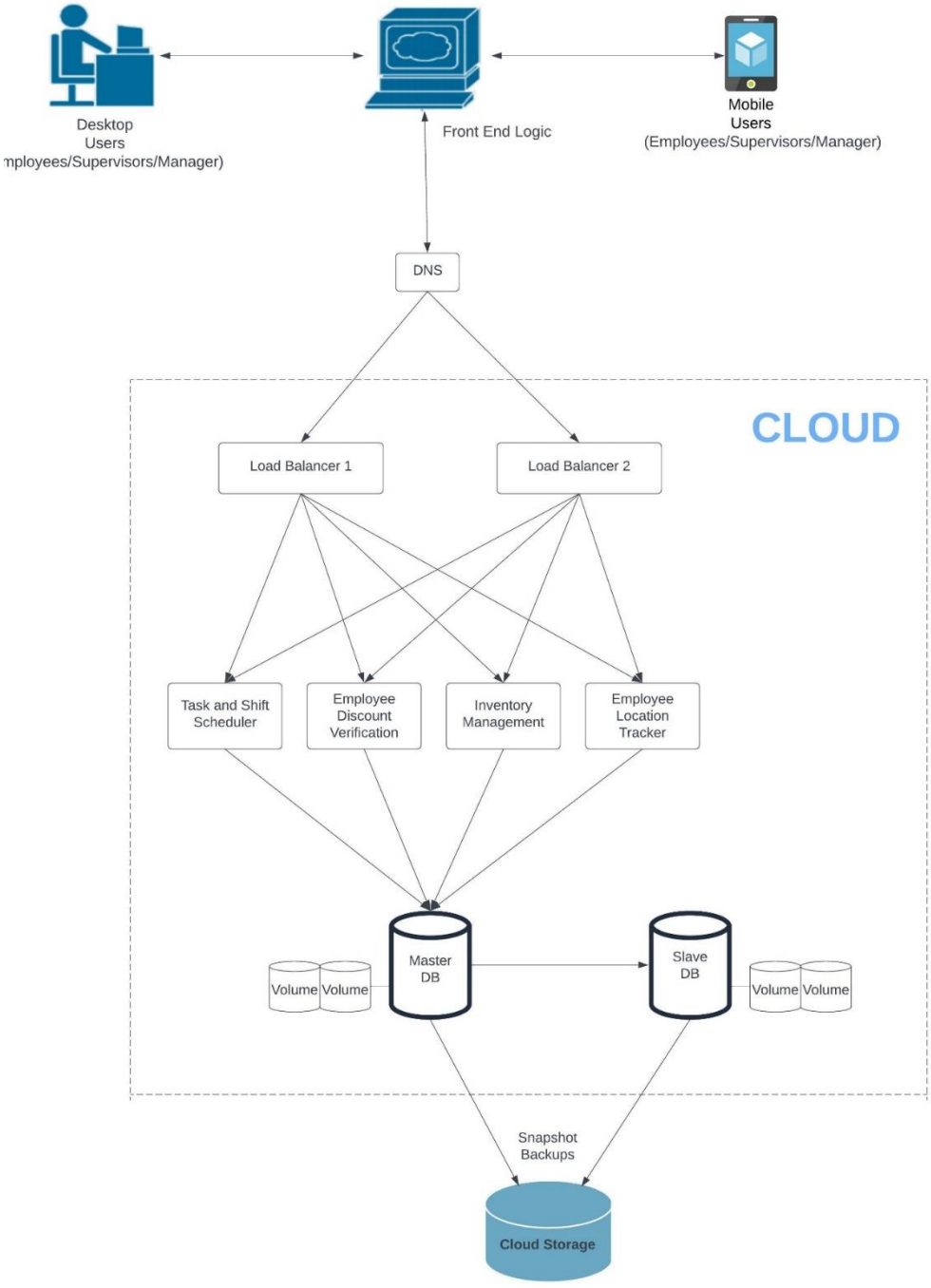
Our requirements for the architecture design matches very closely with the thick-client server based architecture. Also, as the design would involve a focus on multiple applications and data storage of all the employees, we thought that the best design would be a three tiered Server based architecture. It would have the client computer, application database, and a storage database.

Requirements	Our System
Operational Requirements	
System Integration Requirements	✓
Portability Requirements	✓
Maintainability Requirements	
Performance Requirements	
Speed Requirements	✓
Capacity Requirements	✓
Availability/Reliability Requirements	✓
Security Requirements	
High System Value	
Access Control Requirements	✓
Encryption/Authentication Requirements	✓
Virus Control Requirements	
Cultural/Political Requirements	
Multilingual Requirements	
Customization Requirements	
Making Unstated Norms Explicit	
Legal Requirements	✓

Table showing system requirements that we would need to address while designing our system

However, we realized that as we would be monitoring the location of each of the employees, a cloud based architecture would be more efficient. It would help us to track the information and give suggestions in real time. Our major concerns while designing the system were:

1. Scalability
2. Availability
3. Down-time
4. Security
5. Cost



Cloud Based Architecture for University Book Center

The primary concern for the University Book Center that our architecture wants to address is that the system should minimize downtime and data loss. This can be done easily by introducing multiple levels of redundancy, with duplicating systems and resources. The redundant single-site architecture is designed to provide high availability and fault tolerance for critical workloads and applications, while still maintaining a single point of presence. It provides high levels of reliability and performance for our applications without the added complexity or cost of a multi-site architecture.

The key components of the infrastructure are duplicated, including servers, storage, and networking equipment. Redundancy is achieved by having multiple components working in parallel, such that if one component fails, another can take over without any disruption to the service.

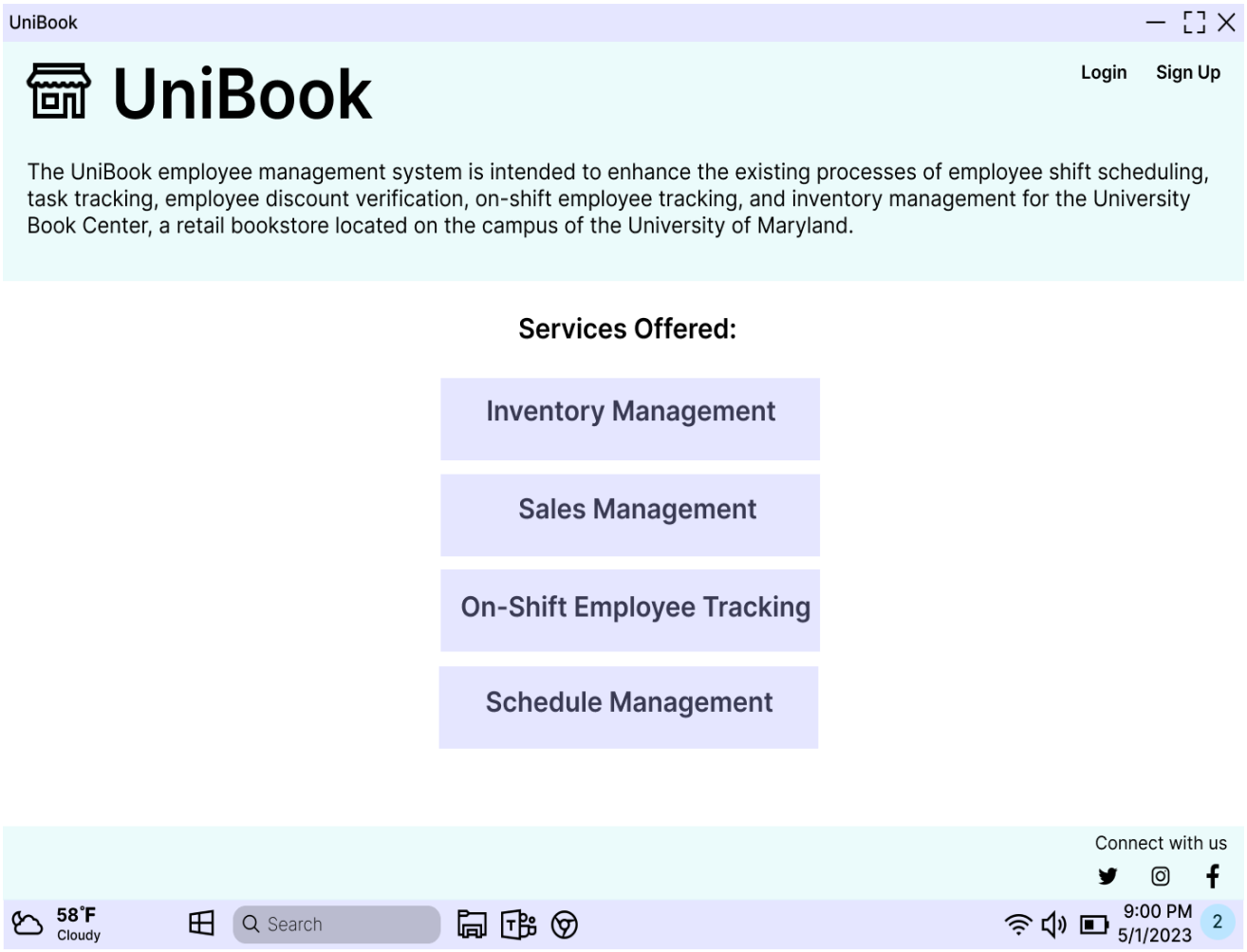
The use of load balancers and failover mechanisms can automatically redirect traffic to a redundant component in case of a failure, thereby, ensuring applications and services are always available. For example, if something were to happen to the traffic coming to load balancer 1, load balancer 2 will take over, so that the access to applications like Task, and Shift Shceduler, Employee Discount Verification, Inventory Management, and Employee Location Tracker is not disturbed.

Front End Logic takes care of the visualization and UI features of our system by providing the required content to the web browsers and the Android/iOS applications. All the data will be stored at the corporate data centers owned by University Book Center. Only one data center is required currently but can be scaled up, if necessary, in the future.
Overall, the redundant single-site cloud architecture for the University Book Center is a highly reliable and cost-effective approach that provides high levels of availability, fault tolerance, and scalability.

9.0 User Interface

9.1 Home Screen

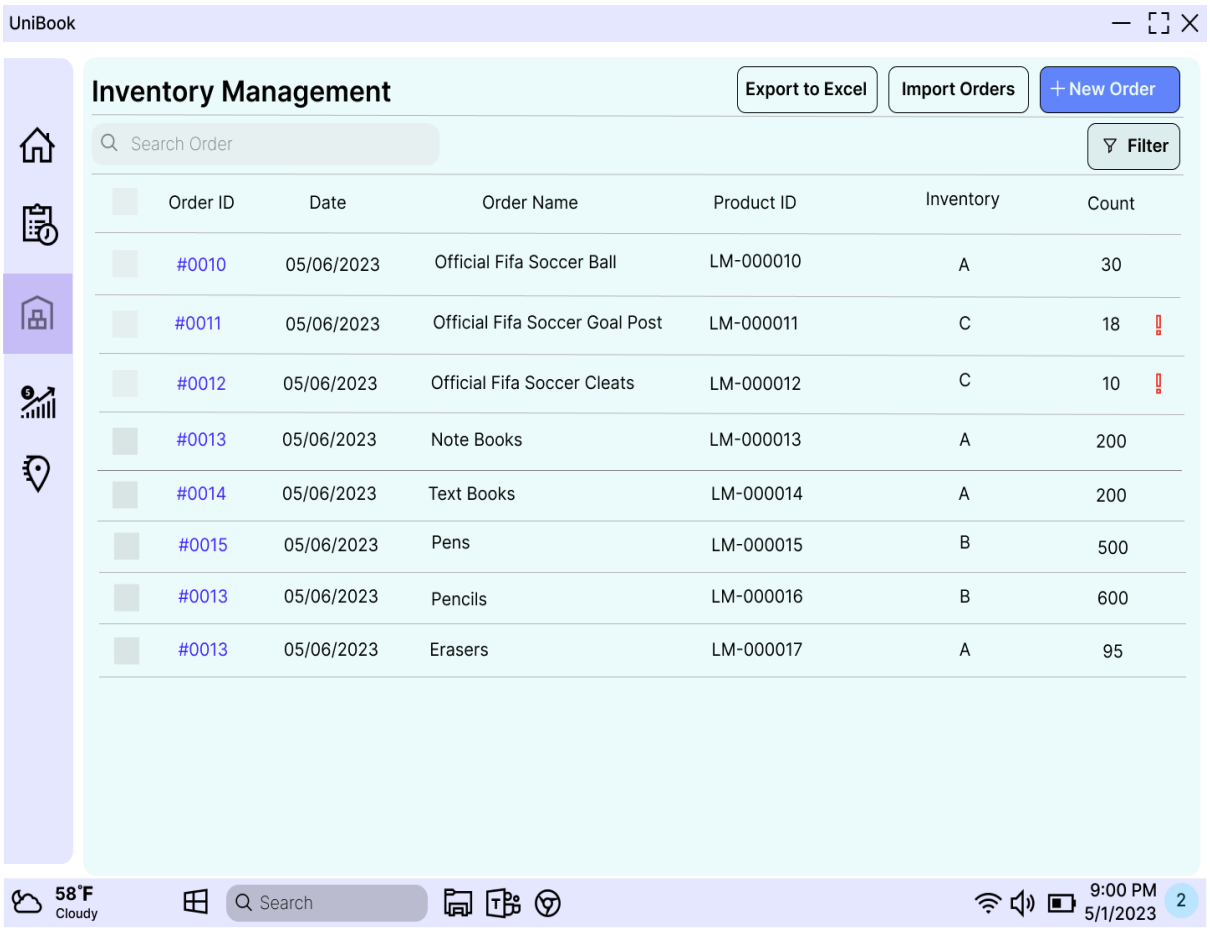
The screen below offers a brief description of UniBook and its perks. The application provides features such as Inventory Management, Sales Management, On-Shift Employee Tracking, and Schedule Management. These services will be made available to the user once they sign up for or log in to their account and become a member of the University Book Center staff group.



9.2 Inventory Management Screen

The following screen provides detailed information about the products currently available in the inventory. It offers information such as:

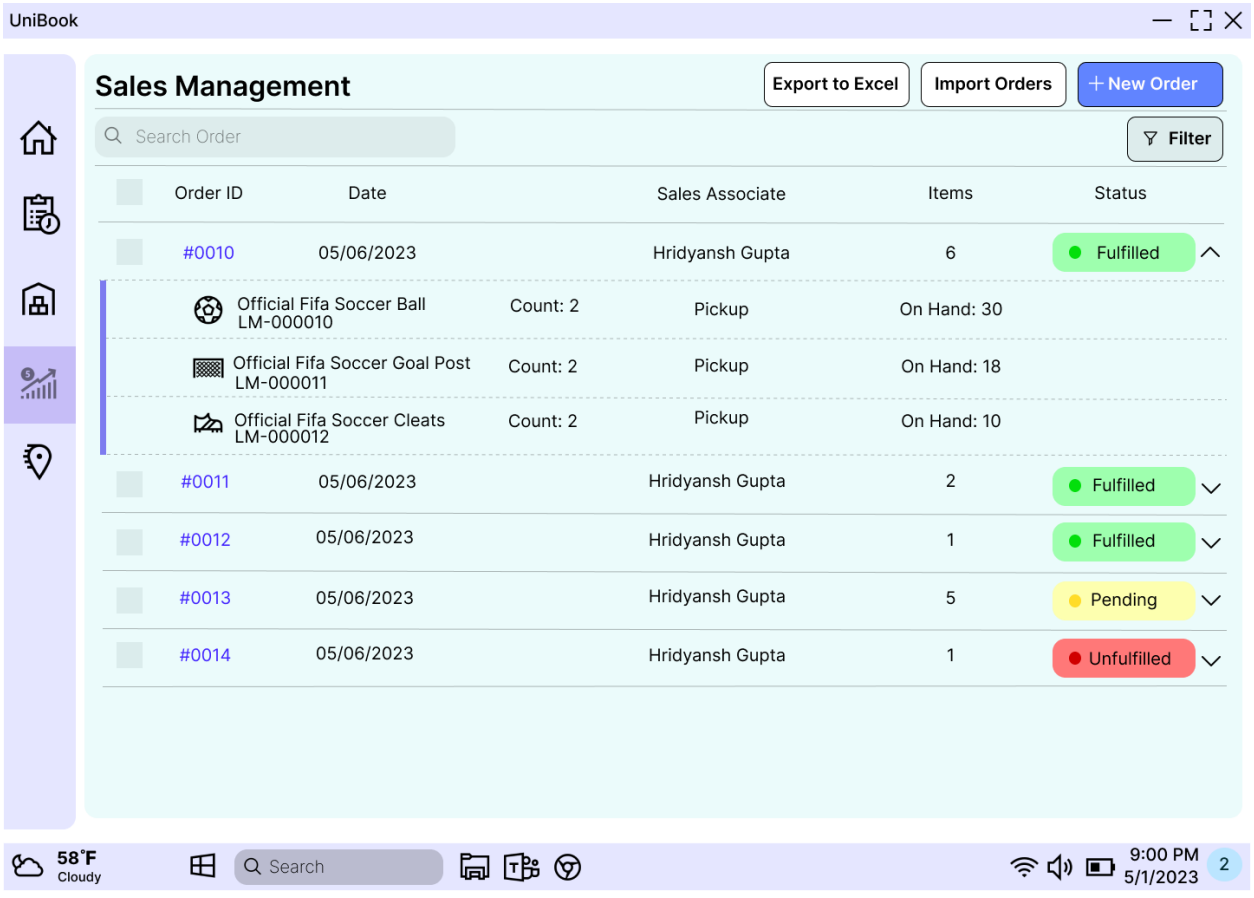
- The Order ID is the product's transaction ID when it is sold to a customer.
- The Date indicates when the product was available in stock.
- The Order Name indicates the product name.
- The Product ID is the ID assigned by the manufacturer during production.
- Inventory indicates where the merchandise is held.
- Count informs us of the current stock of a certain product.
- If the count of any product is less than 20, the user will be notified with an exclamation mark.
- There is an option to search using the search box or select certain orders using the checkbox.
- The user can filter the products they want to use according to date or inventory.
- The user may simply enter a new order into the program or import or export orders via an excel file.
- The user can navigate to other services by clicking on options on the left side of the screen.



9.3 Sales Management Screen

The screen below provides information about the products sold and the status of orders. It offers information such as:

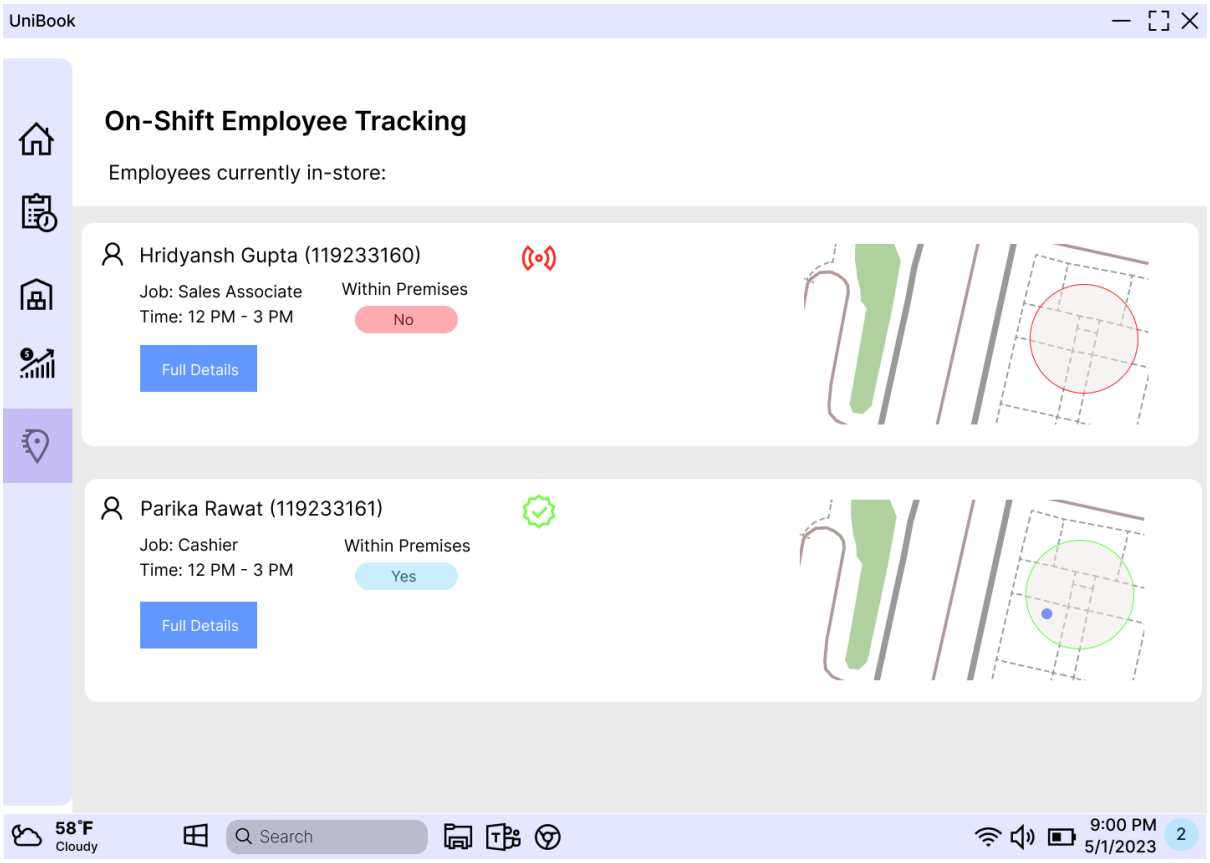
- The Order ID is the product's transaction ID when it is sold to a customer.
- The Date indicates when the product was sold.
- The Customer notifies us who purchased the goods.
- Sales Associate identifies the Sales Associate who made the sale.
- Items represent the total number of products purchased by the customer.
- Status tells us if the order is fulfilled, pending or unfulfilled. Fulfilled indicates that the customer has received the order. Pending indicates that the customer is yet to receive the order. Unfulfilled indicates that the order has not been processed by the University Book Center.
- There is an option to search using the search box or select certain orders using the checkbox.
- The user can filter the products they want to use according to date or inventory.
- The user may simply enter a new order into the program or import or export orders via an excel file.
- The user can navigate to other services by clicking on options on the left side of the screen.



9.4 On-Shift Employee Tracking Screen

The following screen tracks employees during work hours. It offers information such as:

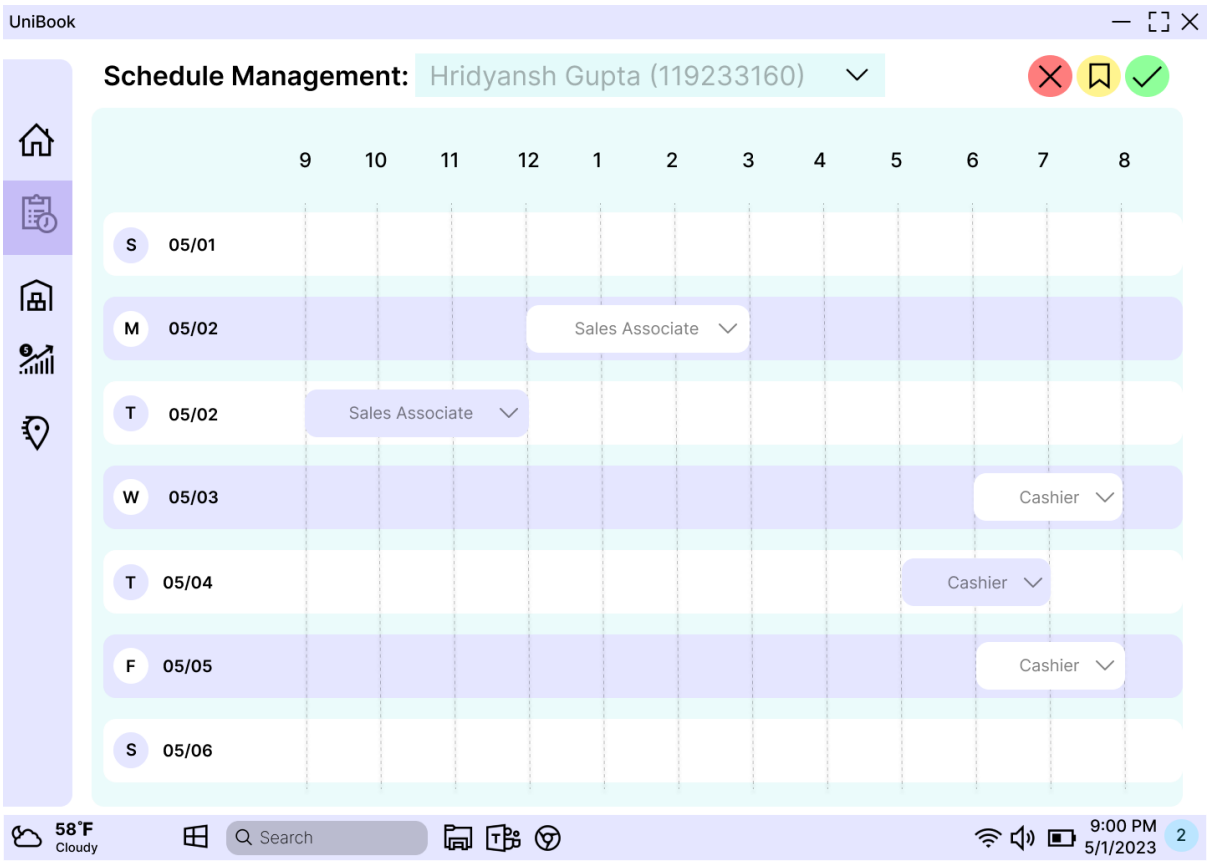
- The name of the employees currently working and their location within store premises.
- It will alert the user when an employee is outside the store premises during work hours.
- The user will see information regarding the jobs employees are assigned to.
- The user can navigate to other services by clicking on options on the left side of the screen.



9.5 Schedule Management Screen

The screen below provides information about the schedule of the employees. It offers information such as:

- The user will be able to create schedules for each employee by selecting their names from the drop-down menu.
- The user will be able to add work hours, edit them and place the assigned jobs and hours for an employee each day. The list of jobs can be accessed by a drop-down menu as well.
- The user can publish the schedule for the employee by clicking on the green tick mark button.
- The user can keep the schedule changes on hold by clicking on the yellow bookmark button.
- The user can clear the schedule by clicking on the red cancel button.
- The user can navigate to other services by clicking on options on the left side of the screen.



10.0 Test Scripts

Now that we have a basic understanding of the requirements, design, and development of the application, let's proceed further to the next phase i.e., Testing. This is to ensure whether the system fulfils all the requirements as outlined in the specifications or not. To make sure that this is done systematically, and results are documented carefully, we begin with developing Test Scripts.

Test Scripts define the actions that a tester/developer must perform to verify whether a specific function or feature of the application is working correctly or not. At this stage, the testing team will identify any defects, bugs or errors and report them back to the development team for permanent resolution.

In this section, we will be going through 3 test types namely, Unit testing, Stress testing and User Scenario testing.

Unit Testing – This is conducted by the developers to ensure that all the individual components or modules of the application are working correctly. It is further divided into two subcategories – Black Box Unit Testing and White Box Unit Testing

Black Box Unit Testing – is which verifies if the unit meets the requirements stated in the specification.

White Box Unit Testing – is which verifies if the actual code such as calculated field/hidden field/automated fields are working correctly or not.

Stress Testing – This ensures if the system runs smoothly with an increase in the load.

User Scenario Testing – This ensures if the system provides a seamless and user-friendly experience.

The below table has four test scripts of each of these test types in the following Test Plan Format: Test Scenario, User Input and Expected Result.

TEST TYPE	TEST SCENARIO	USER INPUT	EXPECTED RESULT
UNIT TESTING – BLACK BOX	Test that the supervisor can assign tasks to employees	The supervisor logs into the app and navigates to the 'Task Management' section. The supervisor selects an employee from the list and assigns him/her tasks for the entire week, based on their work preference. The supervisor saves the tasks and submits them.	The supervisor should be able to log in successfully and navigate to the 'Task Management' section. The supervisor should be able to select an employee from the list and assign him/her task for the entire week, based on their work preference. The supervisor should be able to save the tasks and submit them without any errors.
UNIT TESTING – BLACK BOX	Test if the 'Sales Management' section of the	Import the set of orders using the 'Import Orders' button.	The app should be able to store, receive and manage orders in the table accurately and consistently.

	app can accurately and consistently store, retrieve, and manage order data in the table.	Filter the table by different columns – e.g., Order ID, Date, Customer, Sales Associate, Items, Status using ‘Filter’ button. Search for specific orders using the ‘Search Order’ search box. Manually add a new order using the ‘ New Order’ button Edit or delete an existing order in the table. View more details about the history of the Order ID Export the entire table to an Excel worksheet using the ‘Export’ button	The set of orders should be accurately and consistently added to the table. The filter function should accurately and consistently retreive and display the filtered result. The search function should accurately and consistently retrieve and display the search results. The ’New Order’ button should accurately and consistently add a new order to the table. The editing or deletion function should accurately and consistently update or remove the order details in the table. The Order ID should expand further so that more details about the history of the Order ID can be seen. The exported file should contain all the orders and data in the correct format.
UNIT TESTING – BLACK BOX	Test if the ‘Inventory Management' section of the app can accurately and consistently store, retrieve, and manage order data in the table.	Import the set of orders using the ‘Import Orders’ button. Filter the table by different columns – e.g., Order ID, Date, Order Name, Product ID, Inventory, Count using ‘Filter’ button. Search for specific orders using the ‘Search Order’ search box. Manually add a new order using the ‘ New Order’ button	The app should be able to store, receive and manage orders in the table accurately and consistently. The set of orders should be accurately and consistently added to the table. The filter function should accurately and consistently retrieve and display the filtered result. The search function should accurately and consistently retrieve and display the search results.

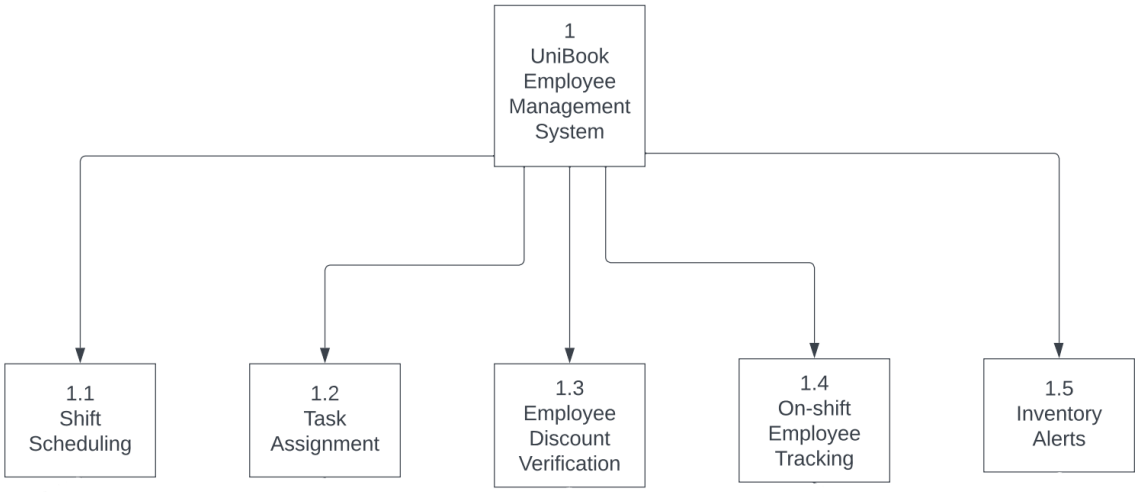
		Edit or delete an existing order in the table. Export the entire table to an Excel worksheet using the 'Export' button	The 'New Order' button should accurately and consistently add a new order to the table. The editing or deletion function should accurately and consistently update or remove the order details in the table. The exported file should contain all the orders and data in the correct format.
UNIT TESTING – WHITE BOX	Test that the 'Task Assignment' logic correctly updates the database or not	The developer checks if the database is updated with the new task list information. The developer checks if the system is able to retrieve task list information from the database. The developer simulates the email notification sent to the employees.	The database should contain a list of employees and their assigned tasks based on their work preference. The code should call the task assignment logic correctly, passing in the correct information to the database and email server and return the correct result. The email notification should be sent to the employee with the correct tasks and due dates
STRESS TESTING	Test that the "Task Management" section can handle a high volume of task assignments without crashing or slowing down.	Multiple supervisors' login with their own login credentials and navigate to the 'Task Management ' section of the app simultaneously. Each assigning a huge volume of tasks and verifying the same for different employees. Simulating a huge volume of simultaneous requests from different supervisors for multiple employees at once	The system should be able to handle the process of logging in, assigning tasks and verifying them through multiple supervisors without crashing or slowing down and the system resources should not be excessively consumed
STRESS TESTING	Test if the app can handle a large number of orders without slowing down or crashing.	Navigate to the 'Sales Management' section of the app. Import a large number of orders e.g., 10,000 orders into the app using the 'Import Order' button.	The app should be able to import a large number of orders e.g., 10,000 orders without crashing or encountering any errors The table should display all the imported orders without any issues, and the order details

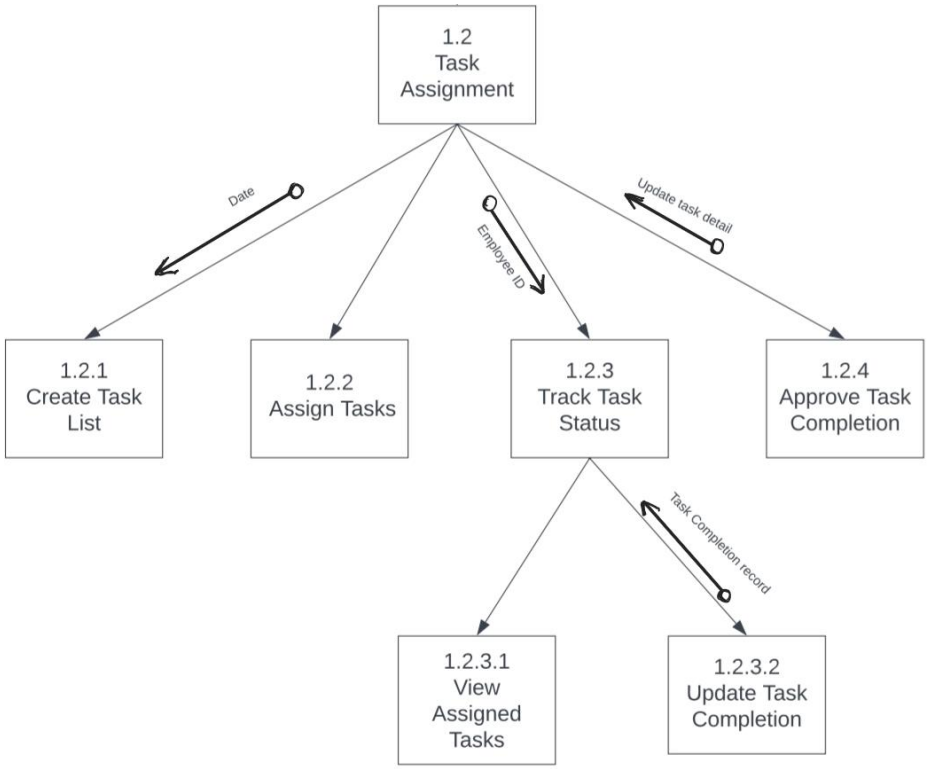
		Verify that the table displays all the imported orders without any issues. Try filtering the table using different columns and check if the app can handle the load without slowing down or crashing. Try seaching for a specific order ID and check if the app can retrieve the details without any issues. Try exporting the entire table to an Excel worksheet and verify that the exported file contains all the orders and data in the correct format	should be accurate and consistent with the imported data. Filtering the table using different columns should not cause any significant slowdowns or errors. Searching for a specific order ID should be fast and accurate, and the app should be able to retrieve the details of the orders without any issues. Exporting the entire table to an Excel worksheet should not cause any significant slowdowns or errors, and the exported file should contain all the orders and data in the correct format
STRESS TESTING	Stress tests the 'On Shift Employee Tracking' screen with a high number of concurrent users e.g., 500 or more accessing the app simultaneously during a busy shift	Any user logs in to the app with his/her own unique credentials. he/she navigates to the 'On shift Employee Tracking' screen. He/she randomly selects a different employee to track. He/she periodically refreshes the screen to update the employee locations and status. He/she may attempt to access the app from locations with poor network connectivity	Retrieve and display accurate and up-to-date employee location details in real-time. Send out-of-bounds alerts to the appropriate manager, supervisor, or employee, if an employee is detected outside the store during working hours. Update employee location details and status in real-time, without delays or errors Provide a seamless user experience for all 500 or more users, without any significant delays or performance issues
STRESS TESTING	Stress test Home Screen under heavy load	Simulate a heavy load of 500 concurrent users accessing the Home Screen at the same time	The Home Screen should be able to handle this heavy load without any significant delay or downtime. The page load time should not exceed 5 seconds, and the system should not crash or freeze. The description of University Book Store and its perks should be

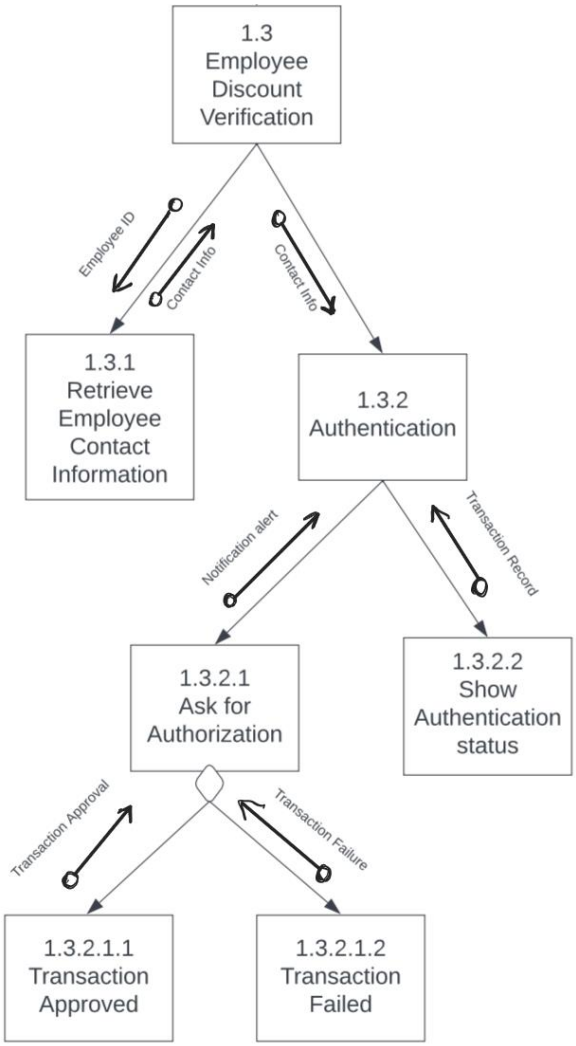
			displayed correctly, with all the images and text appearing as intended. The services such as 'Inventory Management', 'Sales Management', 'On-Shift Employee Tracking, 'Schedule Management' etc should be made able to the user
USER SCENARIO TESTING	Test that the 'Task Management' section provides a user-friendly and intuitive experience for everyone	User logs in as a supervisor and navigates to the 'Task Management' screen. Performs task assignment, verification and approving them	The supervisor should be able to perform the common use cases without encountering any significant issues or challenges, and the user interface should be intuitive and easy to navigate. The supervisor should be able to assign task, verify and approve them without encountering any errors.
USER SCENARIO TESTING	Test if the 'Sales Management' section of the app provides a user-friendly and intuitive experience for managing orders	Navigate to the 'Sales Management' section of the app. Use the 'Filter' button to filter the table by different columns – e.g., date, customer, status. Use the 'Search Order ID' search box for specific orders. Use the 'New Order' button to manually create a new order in the table. Use the 'Export' button to export the table to an excel worksheet. Use the 'Import Order' button to import a set of orders.	The app should provide a seamless and intuitive user experience for managing orders. The filter function should be accurate and consistent in retrieving and displaying the filtered results. The search function should be accurate and consistent in retrieving and displaying the search results. The 'New Order' button should add a new order accurately and consistently to the table. The 'Export button should generate an excel worksheet that contains all the orders and data in the correct format. The table and its columns should be well-organized, easy to read, and user-friendly.

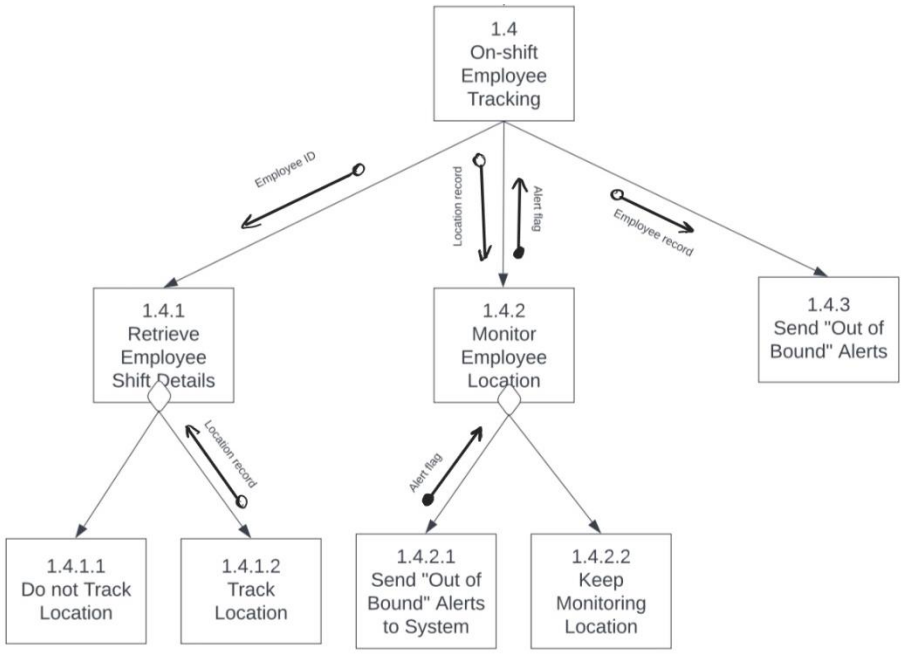
			The button and features are labelled appropriately and provide clear instructions to the user.
USER SCENARIO TESTING	A manager needs to quickly locate an employee who is not at their designated work area	The manager logs in to the app with his/her credentials. The manager navigates to the 'On Shift Employee Tracking' screen. The manager selects the employee that he/she needs to locate from the list of employees on the screen. The application displays a map showing the	The app should load quickly and present a user-friendly login screen. The manager should be able to easily navigate to the 'On Shift Employee Tracking' screen and see a list of all employees on shift. The manager should be able to quickly locate the desired employee from the list of employees on the screen. The map should show the employee's location accurately and in real-time. The manager should be able to easily identify the employee's location and the other details associated with it.
USER SCENARIO TESTING	Checking low stock alert	User opens the 'Inventory Management' section of the app. Navigates to the 'inventory' column of the table. Locates a product with low stock count. Verifies if that cell is highlighted in red with a low stock icon. He clicks on the 'Order ID' to view more details about the product and its order history	The user should be able to easily identify which products are low in stock and receive a clear alert message. The user should be able to view the order history of a product

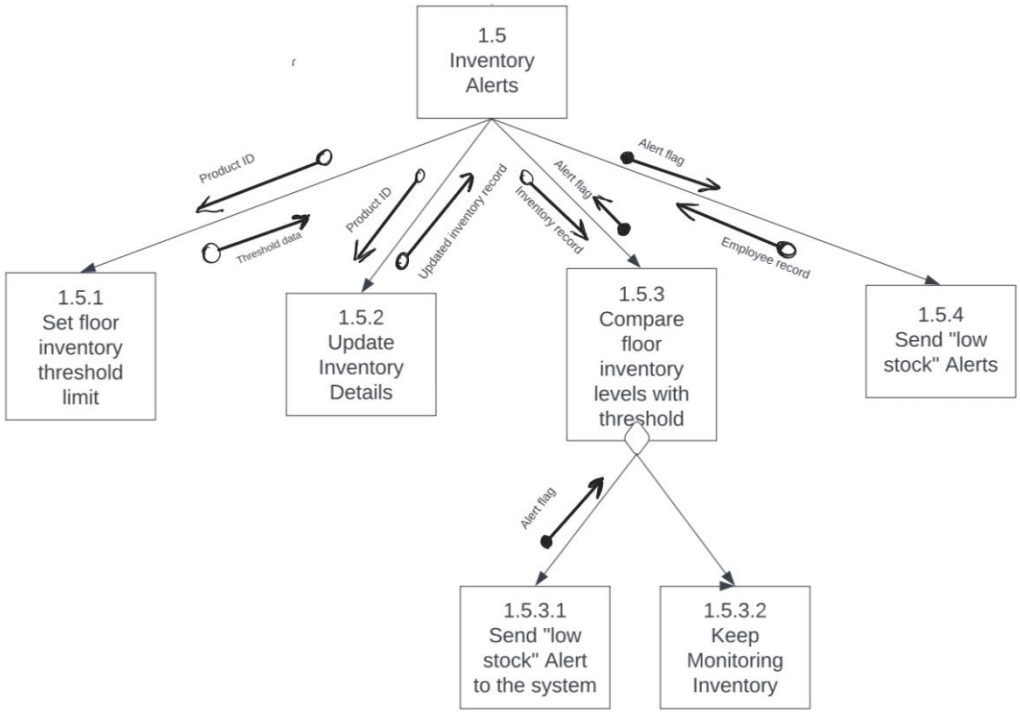
11.0 Structure Chart:




1.2
Task
Assignment
Date
Employee ID
Update task detail
1.2.1
Create Task
List
1.2.2
Assign Tasks
1.2.3
Track Task
Status
1.2.4
Approve Task
Completion
Task Completion record
1.2.3.1
View
Assigned
Tasks
1.2.3.2
Update Task
Completion


1.3
Employee
Discount
Verification
Employee ID
Contact Info
Contact Info
1.3.1
Retrieve
Employee
Contact
Information
1.3.2
Authentication
Notification alert
Transaction Record
1.3.2.1
Ask for
Authorization
1.3.2.2
Show
Authentication
status
Transaction Approval
Transaction Failure
1.3.2.1.1
Transaction
Approved
1.3.2.1.2
Transaction
Failed


1.4
On-shift
Employee
Tracking
Employee ID
Location record
Alert flag
Employee record
1.4.1
Retrieve
Employee
Shift Details
1.4.2
Monitor
Employee
Location
1.4.3
Send "Out of
Bound" Alerts
Location record
Alert flag
1.4.1.1
Do not Track
Location
1.4.1.2
Track
Location
1.4.2.1
Send "Out of
Bound" Alerts
to System
1.4.2.2
Keep
Monitoring
Location


1.5
Inventory
Alerts
Product ID
Threshold data
Product ID
Updated inventory record
Inventory record
Alert flag
Alert flag
Employee record
1.5.1
Set floor
inventory
threshold
limit
1.5.2
Update
Inventory
Details
1.5.3
Compare
floor
inventory
levels with
threshold
1.5.4
Send "low
stock" Alerts
Alert flag
1.5.3.1
Send "low
stock" Alert
to the system
1.5.3.2
Keep
Monitoring
Inventory

This structure chart describes the flow of tasks required for employee management at the University Book Center. The chart has one main module of the UniBook Employee Management System which primarily focuses on employee work and schedule. Below the topmost module, we have five different modules. The modules are as follows:

1. Shift Scheduling: Collects employee shift preferences and shift timings and assigns shifts to employees according to their work authorization and abilities. If the schedule is approved, employees can view it, if not the supervisor has to change it according to the manager's requirements.
2. Task Assignment: The module collects employee and schedule information and accordingly task creation and assignment take place. The next step shows updating the task status as and when the task progresses. The process ends with the task completion approval.
3. Employee Discount Verification: It starts with collecting employee contact information for authentication. If the details are correct the discount is approved and the system shows an authentication, otherwise the discount is declined.
4. On-Shift Employee Tracking: This module takes in the employee shift and break information. The system tracks the employee only when they are on shift. If the employee is off the store premises an employee out-of-bound alert is sent out.
5. Inventory Alerts: This module collects product IDs and thresholds for a low stock for each product and updates the inventory. If the threshold limit and floor inventory level for any product is the same, a "Low Stock" alert is sent out.

12.0 Conclusion:

The different requirements for the Employee Management Application have been identified to create the UniBook management application which aspires to be a one-stop solution for all operations related to a medium scale Book Center. A clear understanding of the project objectives and scope was delineated before outlining both the system analysis phase and the system design phase. An overall robust information system was designed for the University Book Center so that they can improve the efficiency, productivity, and streamline their daily processes allowing for the overall success and growth of the store.