LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6004NI Application Development

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2022 Autumn

Student Name: Achyut Adhikari

London Met ID: 20048811

College ID: NP01CP4S210055

Assignment Due Date:5th January 2023

Assignment Submission Date: 5th January 2023

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded

Table of Contents

Table OF Figure

Introduction

A shop that offers bike maintenance and repair is known as a bike service center. These services may consist of changing old out parts, repairing flat tires, adjusting brakes and gears, adjusting brake pads, and Others. In which using a paper-based system for stock and inventory can be extremely difficult to obtain using a manual paper-based technique. Additionally, there is a constant risk that the papers used for record keeping will lose or suffer damage consequently with outside factors The use of a manual, paper-based approach is therefore highly risky.

As a result, we are instructed to use C# and Visual Studio to develop the inventory management system for a bike servicing center in conformance with the requirements of this area of the module. The program was created using Visual Studio (2022).

Detailed Instruction to run and logging in:

Login

Logging in as administrator will allow the user to set inventory in the system. This can also allow the user to set the open and close time of the service center. The user can keep a backup of the rate file in the case they need require data.

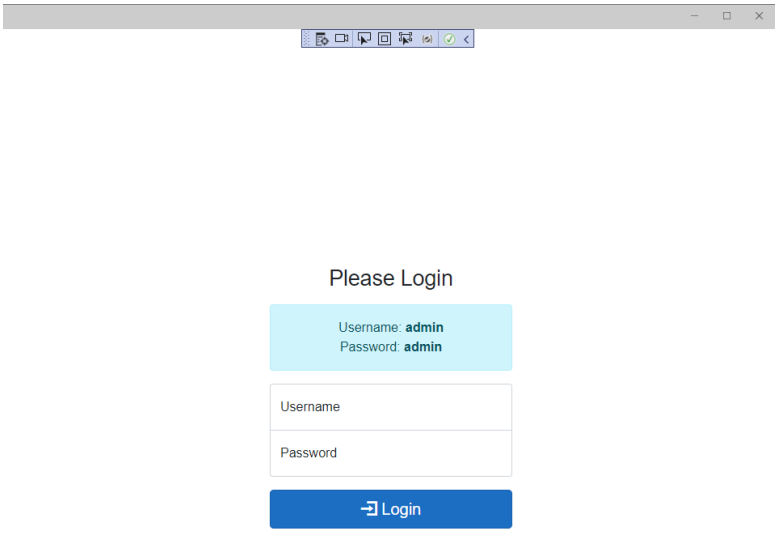

Figure 1 Login Page

Dashboard

After logging the system the user can see the dashboard in which users are shown Admin can add or delete the role but user cant.

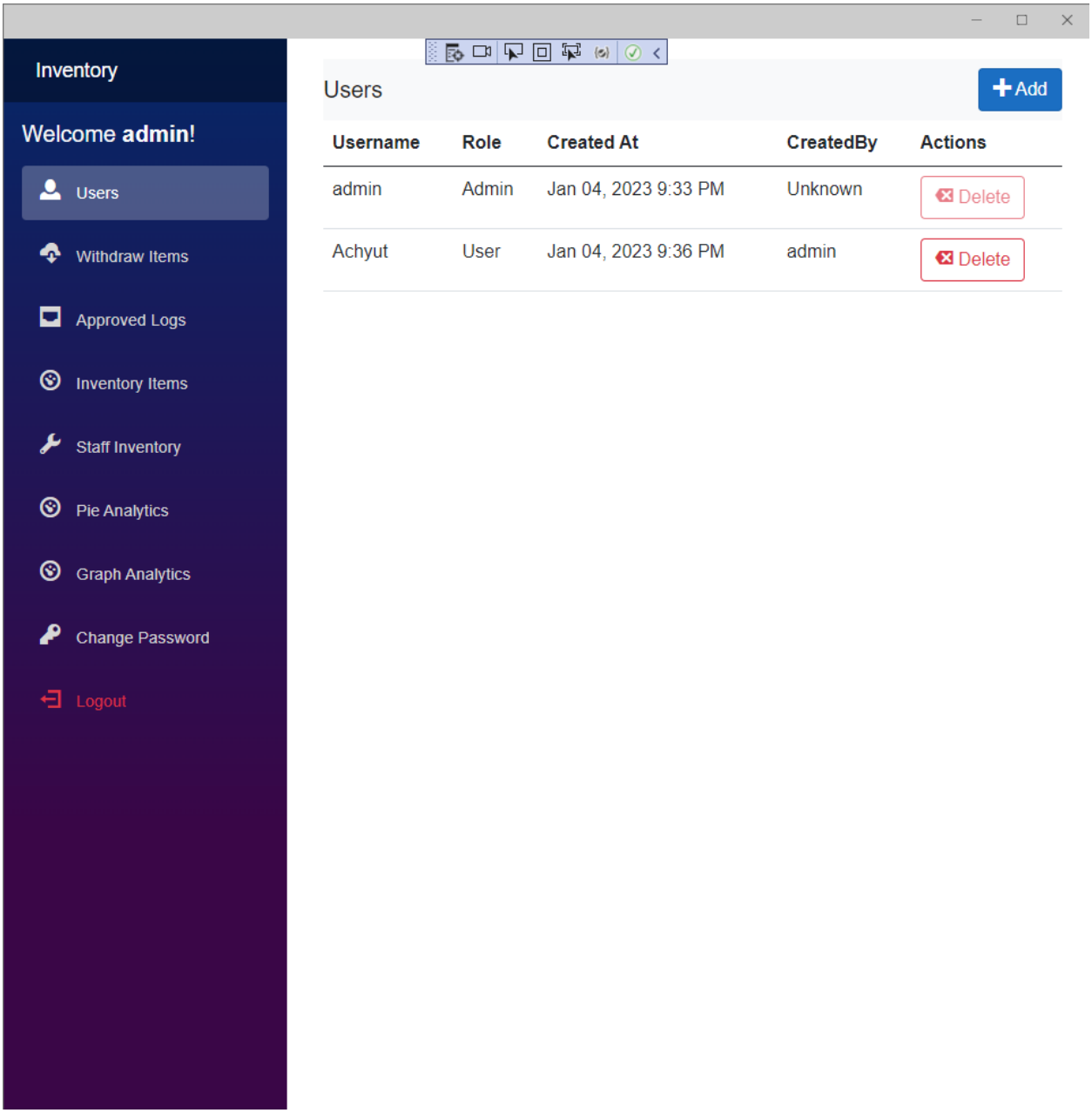

Figure 2 Admin Dashboard

Inventory Items

The inventory items is the items in inventory of bike service center in which user can add or delete the item kept in inventory.

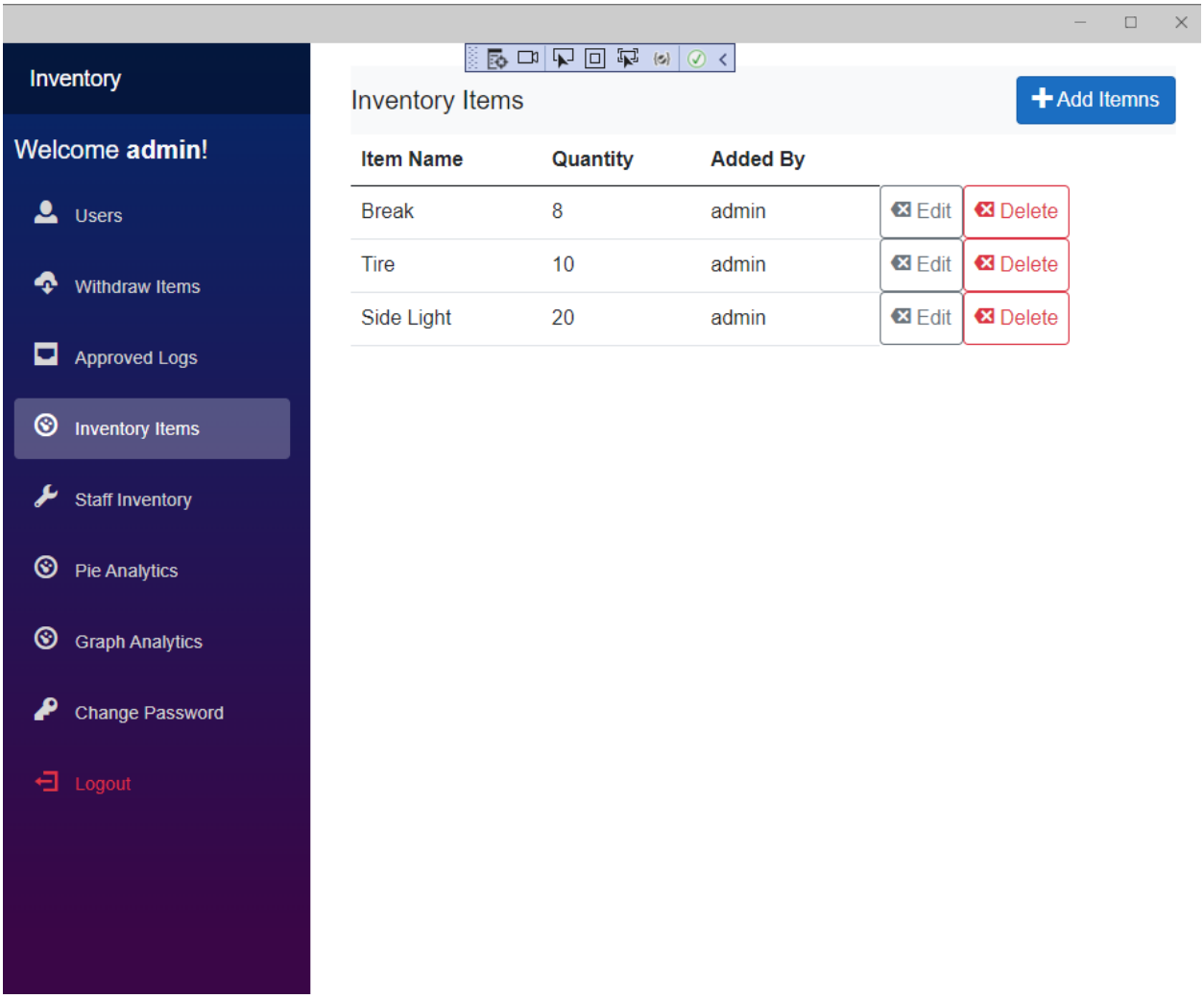

Figure 3 Inventory Items

Staff Inventory

This inventory is a inventory of the staff in which they are withdraw the items from the list and that can be tracked so that lost or any damaged caused by staff will be properly managed.

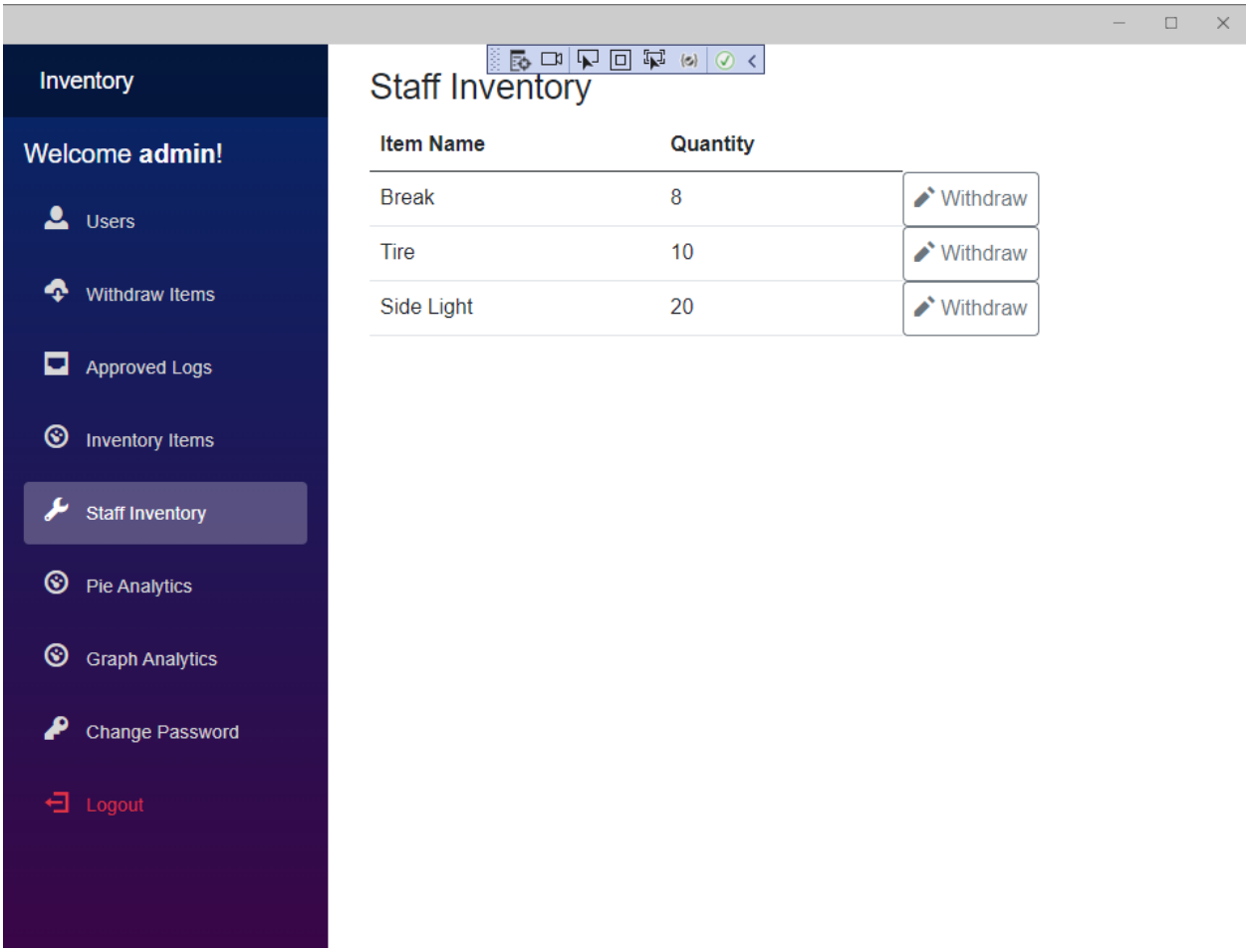

Figure 4 Staff Inventory

Pie Analytics

Pie Analytics is the real time pie chart which shows the data in pie format of inventory. Its shows the total number of Inventory items.

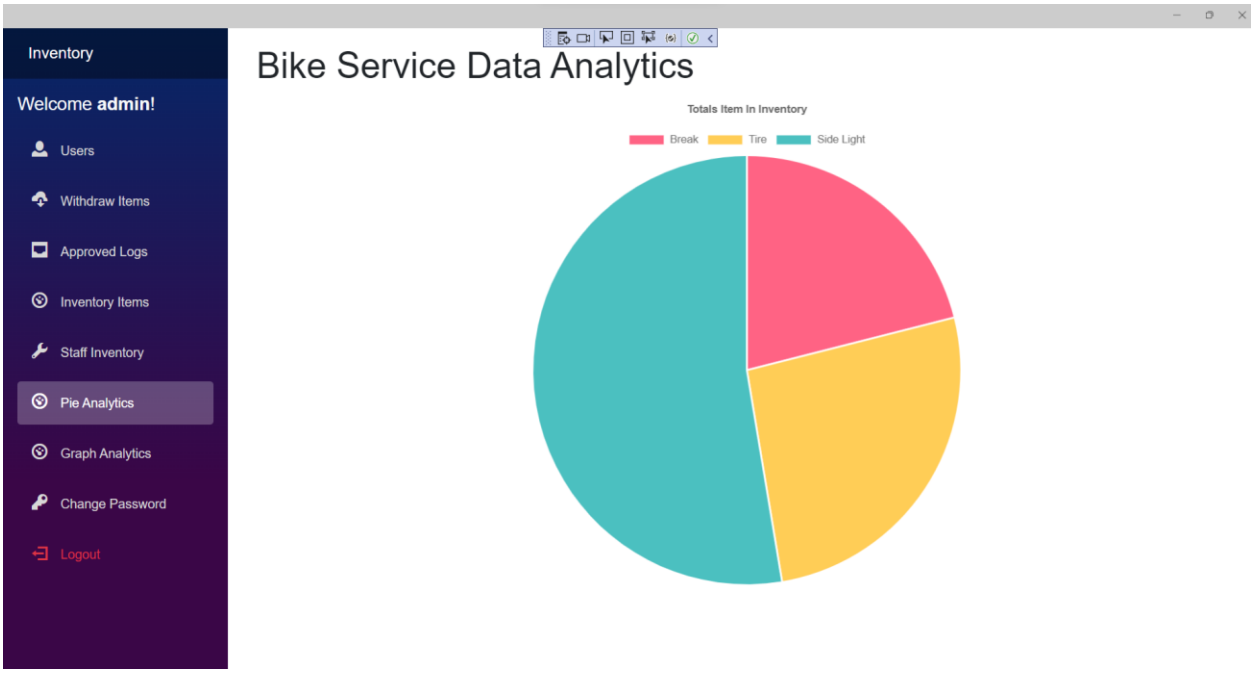

Figure 5 Pie analytics

Graph Analytics

The Graph Analytics is the graphical representation of the inventory. In this graph data is shown to analyze the inventory in graph.

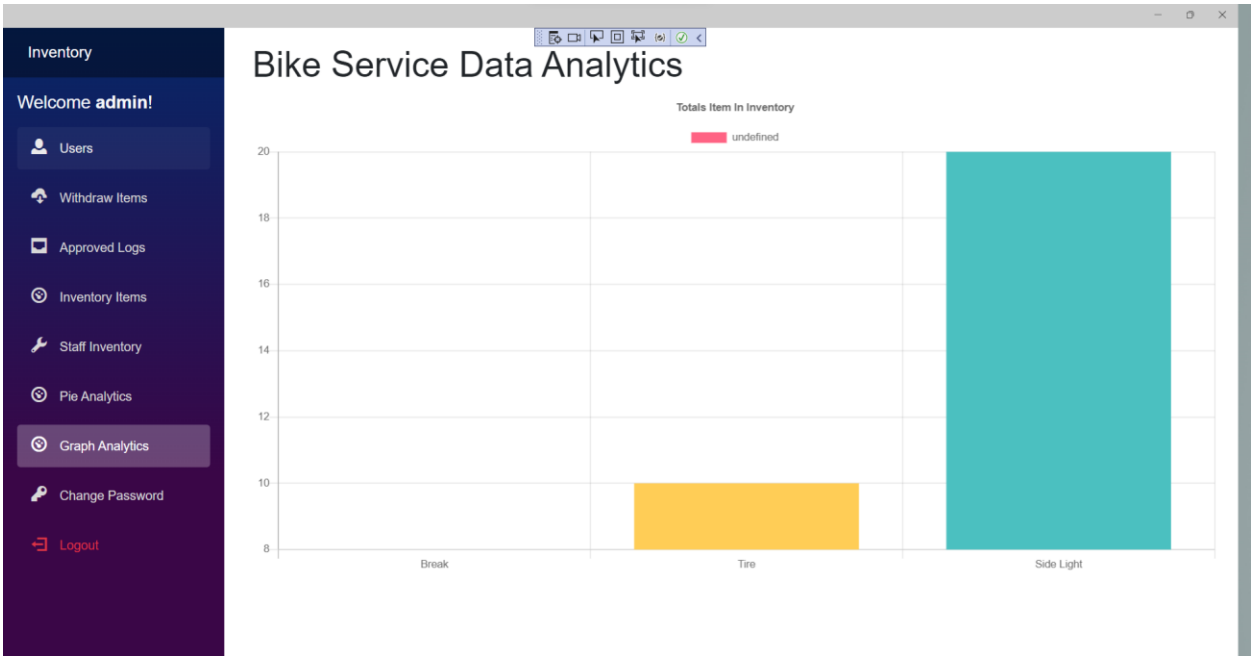

Figure 6 Graph Analytics

Change Password

The User can change password using the current password.

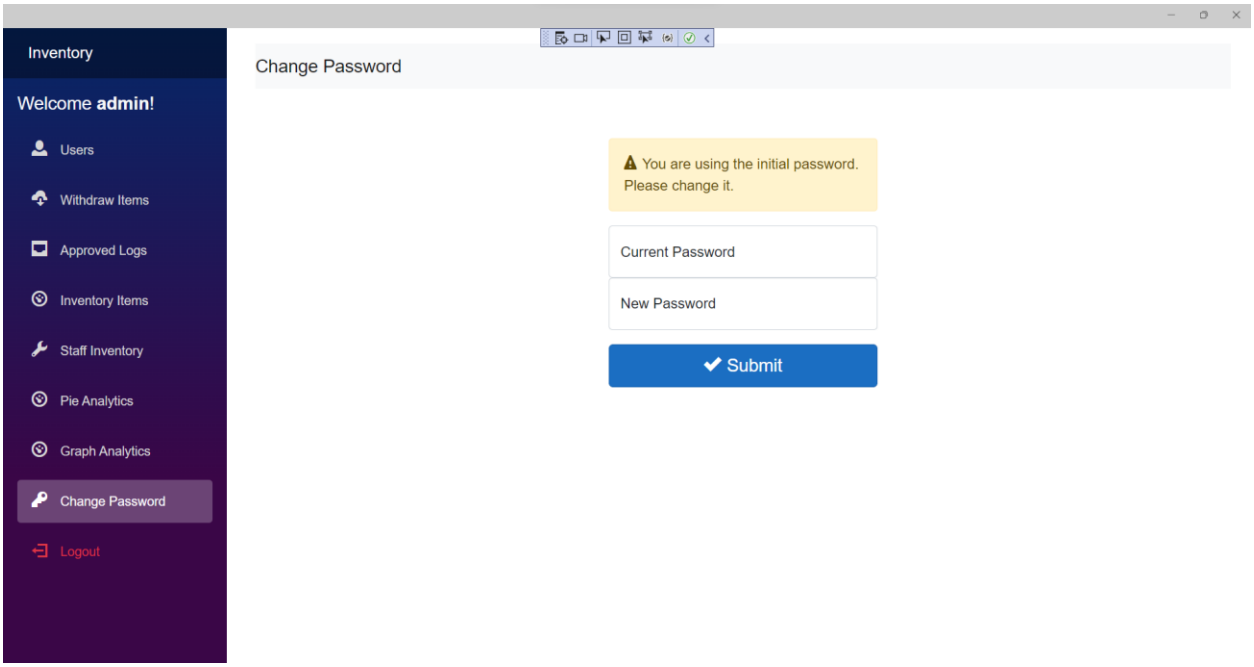

Figure 7 Change Password

Description of logical Solution

Merge Sort

A divide-and-conquer technique called merge sort sorts a list by continuously breaking it down into smaller sub - arrays and then putting the latter has back together in a certain order.

The algorithm operates as follows:

1. Split the list into two sub - arrays of equal size if it contains more than one element.

2. Sort the two following years repeatedly.

3. Reconcile the two two begin that were sorted in a certain order.

The merge step, which rejoins the two sorted complex mesh, is the most important step in the merge sort algorithm. To do this, the components of the sub - arrays are compared, and they are then added to a new list in the proper order.

Class and their purpose

1.Approved Items

Approveditems.Cs

Appr&veditem

Id: Guid & ItemName: string

Itemld: Guid & Quantity: int

IsApproved : bool

TakenBy: Guid TakerName: string

ApprovedBy: Guid

ApproverName : string

2. Approved Item service

SaveAll(Guid, List Approveditem >) : void

GetAII(Guid) : List ApprovedItem>

Create(Guid, string, Guid, int, string, Guid, string, bool) : List ‹Approveditems

3. Data Analysis

DataAnalysis.cs

DataAnalysisS

DataAnalysisDTO (Guid) : List «DataAnalysisDTO >

4. DataAnalysisDTO

TitleNam»: string

ValueCount: int

5. Global State

GlobalState.cs

GlobalState

CurrentUser: User

6. Inventory Items

Inventoryltems

Id: Guid

ItemName: string

Quantity: int

AddedBy: string

7. Inventory Service

InventoryService.c

InventoryService

SaveAll(Guid, List «Inventory/tems*) : void

GetAIlO : List < Inventorytems>

Create(Guid, string, string, int) : List «Inventoryltems>

Update(Guid, Guid, string, int) : List «Inventorytems>

Withdrawltem(Guid, Guid, string, int) : List « Inventory|tems>

RejectWithdrawltem(Guid, Guid, string, int) : List «Inventorytems>

CancelWithdrawltem(Guid, Guid, string, int) : List « Inventorytems>

Delete(Guid, Guid) : List < Inventory|tems*

1. **Role**

   Role.cs

   Role a User

   = Admin

2. **User**

   User

   Id: Guid

   Username: string

   PasswordHash: string

   Role: Role

   HasinitialPassword: bool

   CreatedAt: DateTime

   CreatedBy: Guid

3. **User Service**

   UsersService

   SeedUsername: string

   SeedPassword : string

   SaveAll(List <User») : void

   GetAllO : List «User»

   Create(Guid, string, string, Role) : List <User> MaxAdmin0 : bool

   SeedUsers0 : void

   GetByld(Guid) : User

   Delete(Guid) : List Users

   Login(string, string) : User ChargePassword(Guid, string, string) : User

4. **Utils**

   Utils

   _segmentDelimiter: char

   HashSecret(string) : string

   VerifyHash(string, string) : bool

GetAppDirectoryPath0 : string

GetAppUsersFilePath0 : string

GetTodosFilePath(Guid) : string

GetInventoryFilePathO: string

GetWithdrawlFilePath(Guid) : string

GetApprovedFilePath(Guid) : string

5. **With Draw Items**

Withdrawlltem

Id: Guid

ItemName: string

Quantity: int

Itemld: Guid

IsApproved: bool

TakenBy: Guid

TakerName: string

CreatedAt : DateTime

6. **With Draw Service**

Withdraw|Service

SaveAll(Guid, List Withdrawlltem»): void

GetAll(Guid) : List WithdrawIItem>

Create(Guid, Guid, string, int, string) : List Withdrawlitem> © Delete (Guid, Guid) : List <Withdrawlitem>

Reflection of own experience and conclusion

We covered the fundamentals of C# object-oriented programming, data structures, and algorithms throughout this section of our module. It also gave us a basic understanding of the Visual Studio IDE and the.NET Framework. The project makes use of some of the most well-liked internal and outside libraries, including JSON and Linq.

Creating a Windows program on Visual Studio was an entirely different experience, even if the graphical user interface application made as part of this module may correspond to the same concepts as GUIs in other languages and frameworks.

As I strongly believe that the tools used for this project will help me in near future to do a lot of works as I am planning to purse Artificial intelligence . net will be one of the parts of that journey. I want to go deeper into Microsoft's official documentation and develop several analogous apps as a strategy to improve my C# proficiency.