LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

**Module Code & Module Title**
**CS6004NI – Application development**

**Assessment Weightage & Type**
**30% Individual Coursework**

**Year and Semester**
**2022-23**

**Student Name: Kishor Shrestha**
**London Met ID: 2008913**
**College ID: np01cp4s210161**
**Assignment Due Date: 5th January 2022**
**Assignment Submission Date: 5th January 2022**

*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 marks of zero will be awarded.*

## Table of Contents

1 Introduction ........ 6

2 Detailed Instruction to run the program ........ 6

2.1 Main system user interface (Login Form) ........ 7

2.2 Staff Dashboard ........ 8

2.3 Change password of Staff. ........ 10

2.4 Change Name of staff or update name ........ 11

2.5 Items list show on staff ........ 12

2.6 Items listed on staff. ........ 13

2.7 Requests Items to approve or delete ........ 13

2.9 Approve Takeout due to Time Table ........ 15

2.10 Add Items ........ 16

2.11 Record Approved by admin ........ 16

2.12 Record deletes by admin ........ 17

2.13 Record rejected by admin on table ........ 18

2.14 Record approved by admin on table ........ 18

2.15 Dashboard of Items listed on bar graph. ........ 19

2.16 Restriction on admin ........ 19

3 System Architecture ........ 20

4 Classes and their purpose ........ 22

4.1 Purpose of Login Staff ........ 22

4.2 Purpose of Staff Dashboard class ........ 23

4.3 Purpose of Admin Dashboard class ........ 24

4.4 Purpose of Daily Report class ........ 25

4.5 Purpose of Weekly Report class ........ 25

5 Reflection of the learning experience ........ 25

6 Conclusion ........ 26

6.1 Evaluation of work ........ 26

6.2 Learning Outcomes ........ 27

7 Bibliography ........ 28

# Table of Figure

- ✓ Figure 1: Main system user interface (login form). ........ 8
- ✓ Figure 2:Login Staff. ........ 9
- ✓ Figure 3:Update details of Staff. ........ 10
- ✓ Figure 4:Change Name of staff or update name. ........ 11
- ✓ Figure 5:Items Table of Staff. ........ 12
- ✓ Figure 6:Items name and there details of staff. ........ 13
- ✓ Figure 7:requesting items to admin. ........ 14
- ✓ Figure 8:requesting approve Takeout. ........ 15
- ✓ Figure 9:Figure of Add items. ........ 16
- ✓ Figure 10:Figure of Record is approved by admin ........ 17
- ✓ Figure 11:Figure of request reject by admin. ........ 18
- ✓ Figure 12:Figure of Bar graph of items. ........ 19

Figure 13: limitation in system when there is include of more than two admin ........ 20

- ✓ Figure 13:Figure of System Archicheture. ........ 21

# Table of Table

✓ Table 1:Table of Login Form. 22
✓ Table 2:Table of Staff Dashboard. 23
✓ Table 3:Table of Admin Dashboard. 24
✓ Table 4:Table of Daily repost. 25
✓ Table 5:Table of Weekly report. 25

# 1 Introduction

Using C# and the .NET Framework, you can create a desktop application that is able to store and manage inventory data for a recreation facility. The C# language is a popular choice for developing desktop applications, especially for Windows-based systems, due to its strong support for object-oriented programming and its easy integration with the .NET Framework. The .NET Framework provides a set of libraries and tools that allow developers to easily build and deploy a variety of applications, including web, mobile, and desktop applications.
This might include features such as:

1. Adding, modifying, and deleting inventory items
2. Tracking the quantity of each item in stock
3. Generating reports on inventory levels, usage patterns, and other trends
4. Alerting staff when items are running low or need to be restocked
5. Overall, the use of C# and the .NET Framework can help you create a powerful and efficient inventory management system for your recreation facility.

It sounds like the current paper-based system for managing inventory and tracking visitor information at The Bike Center has some issues, and that you are looking to develop a C# desktop application to address these problems.

Some of the key features of the proposed application may include:

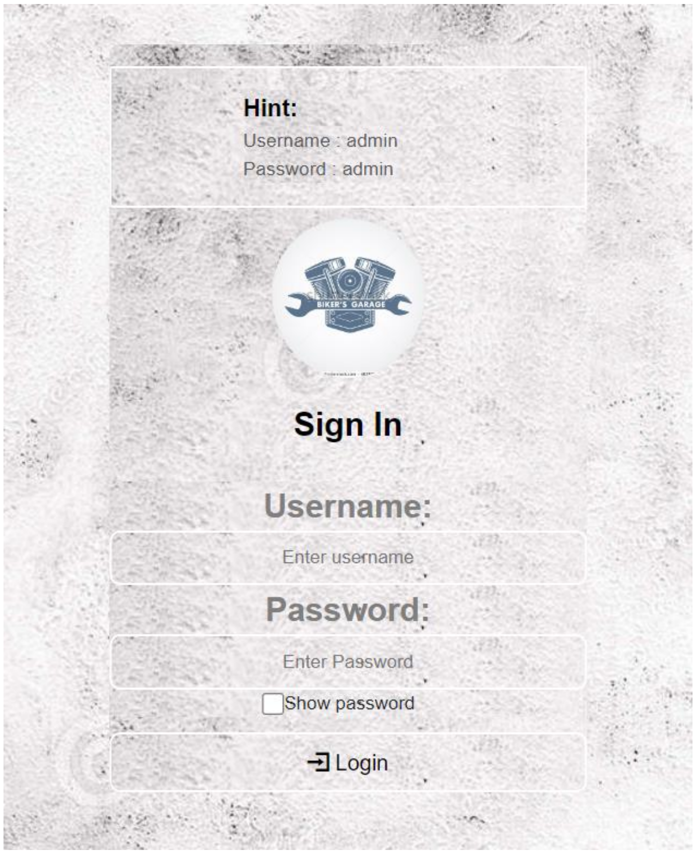
- Tracking visitor information: This might include storing details such as the visitor's name, age, and length of stay at the Bike Center.
- Maintaining ticket prices: The application could allow staff to set different ticket prices for different age groups or types of visitors, and to adjust these prices as needed.
- Managing inventory: The application could allow staff to add, modify, and delete inventory items, as well as track the quantity of each item in stock.
- Generating reports: The application could provide a range of reports on things like inventory levels, visitor trends, and revenue, to help staff better understand the performance of the Bike Center.
- Access hours: The application could be set to only be accessible during certain hours of the day, such as 9:00 AM to 4:00 PM.
- Overall, the proposed C# desktop application could provide a more efficient and effective way to manage inventory and track visitor information at The Bike Center, helping to improve operations and decision-making. (source code, 2023)

# 2 Detailed Instruction to run the program

A user guide or comprehensive instructions document can be a helpful resource for users of the Bike Center System, as it provides step-by-step instructions on how to use the different features of the application. Some key sections that might be included in the user guide could include:

- Getting started: This section could provide an overview of the main features of the Bike Center System, as well as information on how to install and launch the application.
- Managing inventory: This section could provide instructions on how to add, modify, and delete inventory items, as well as how to track the quantity of each item in stock.
- Tracking visitor information: This section could explain how to input and view details about visitors, including their name, age, and length of stay.
- Setting ticket prices: This section could describe how to set different ticket prices for different age groups or types of visitors, and how to adjust these prices as needed.
- Generating reports: This section could provide instructions on how to access and view different reports, such as those on inventory levels, visitor trends, and revenue.
- Troubleshooting: This section could provide solutions to common problems that users might encounter when using the Bike Center System, such as error messages or issues with data entry.
- Overall, a well-written user guide can help users of the Bike Center System to get up and running quickly and easily, and to make the most of the application's features.

## 2.1 Main system user interface (Login Form)



**Hint:**
Username : admin
Password : admin

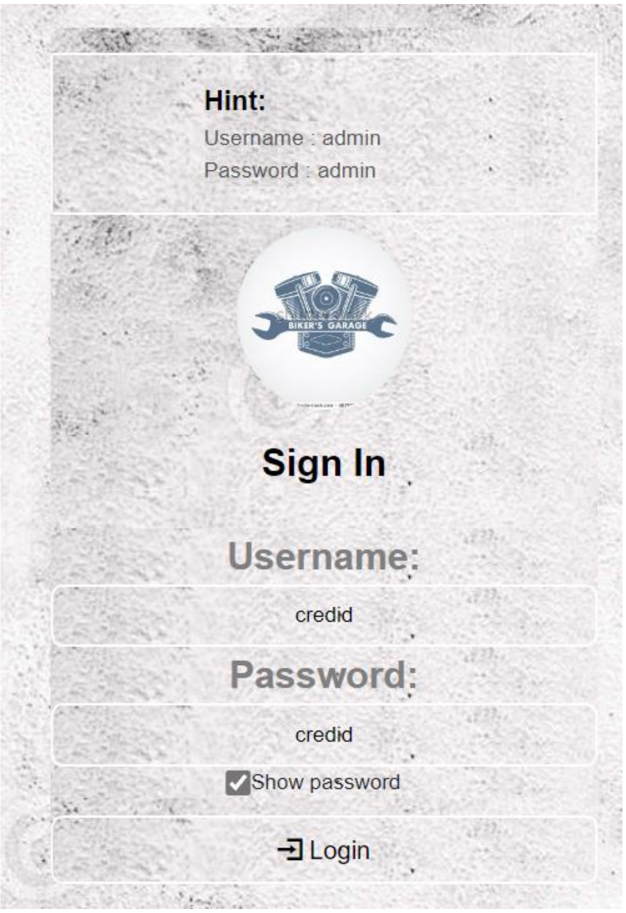
- ✓ Figure 1: Main system user interface (login form).

When the system is started, the primary form, which is a login form, loads.
Staff and administrators can access their own dashboards primarily through this method.
The username and password for both Staff and Admin are provided, so either one can log in from this page:

Username for Admin: admin
Password for Admin: admin Password

To sign in to the system, use the designated username and password.

## 2.2 Staff Dashboard



**Hint:**

Username : admin

Password : admin

Sign In

Username:

credid

Password:

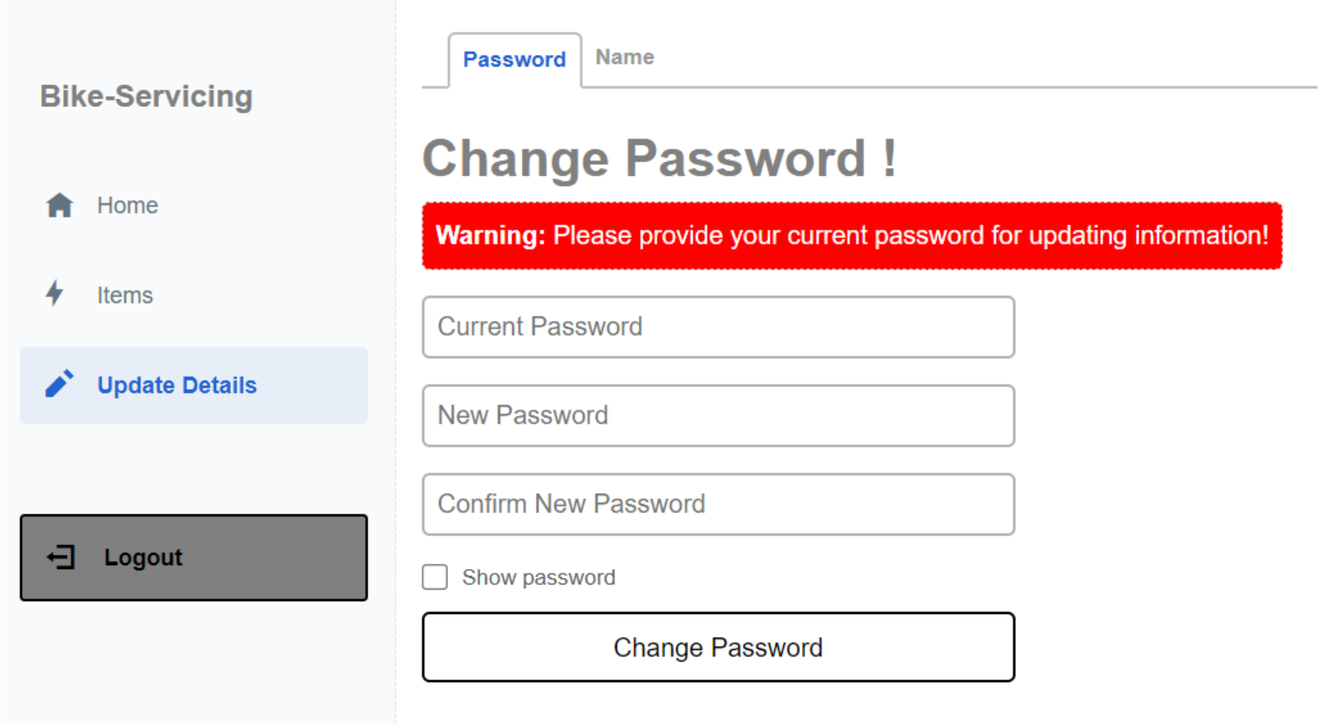
credid

☑ Show password

Login

✓ Figure 2:Login Staff.

## 2.3 Change password of Staff.



Change Password !

**Warning:** Please provide your current password for updating information!

Current Password

New Password

Confirm New Password

□ Show password

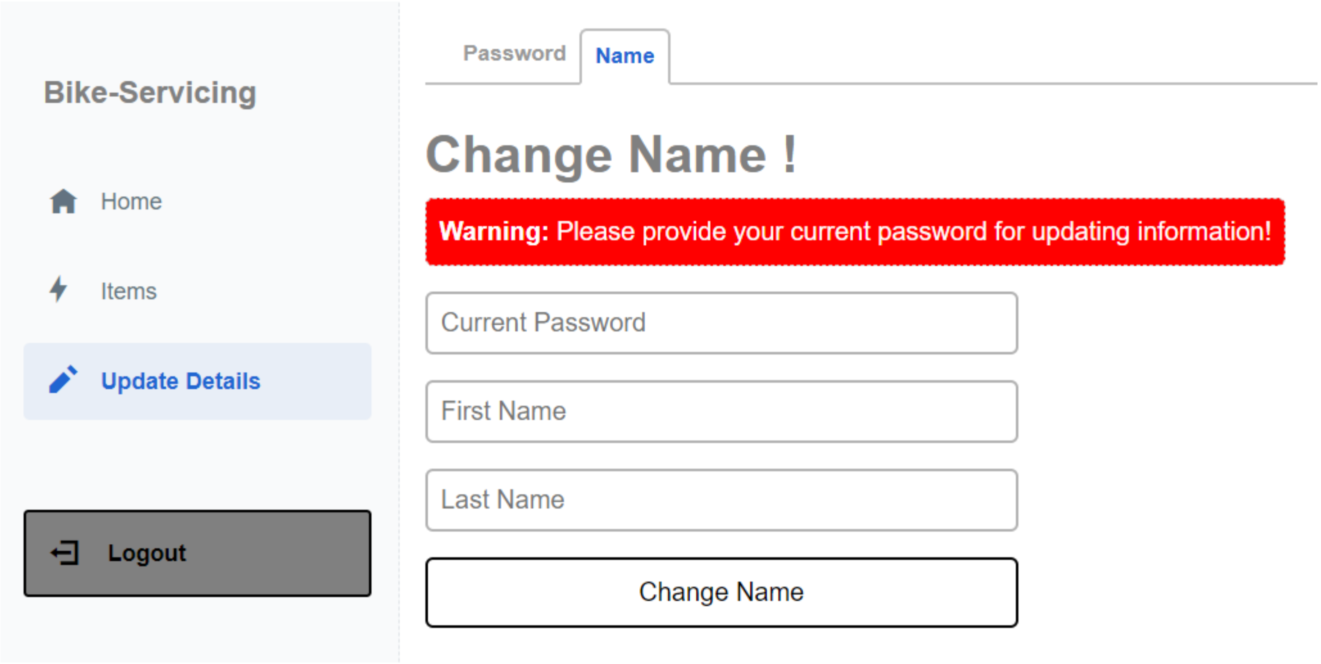
Change Password

✓ Figure 3:Update details of Staff.

To change the current password for the Bike-servicing Center System, you will need to follow the steps below:

- Click on the "New- password" and confirm new password.
- In the "Change Password" section, enter your current password in the first field.
- In the "New Password" field, enter the password that you would like to use going forward.
- Re-enter the new password in the "Confirm New Password" field to confirm it.
- Click the "Change Password" button to save the new password.

## 2.4 Change Name of staff or update name



Change Name !

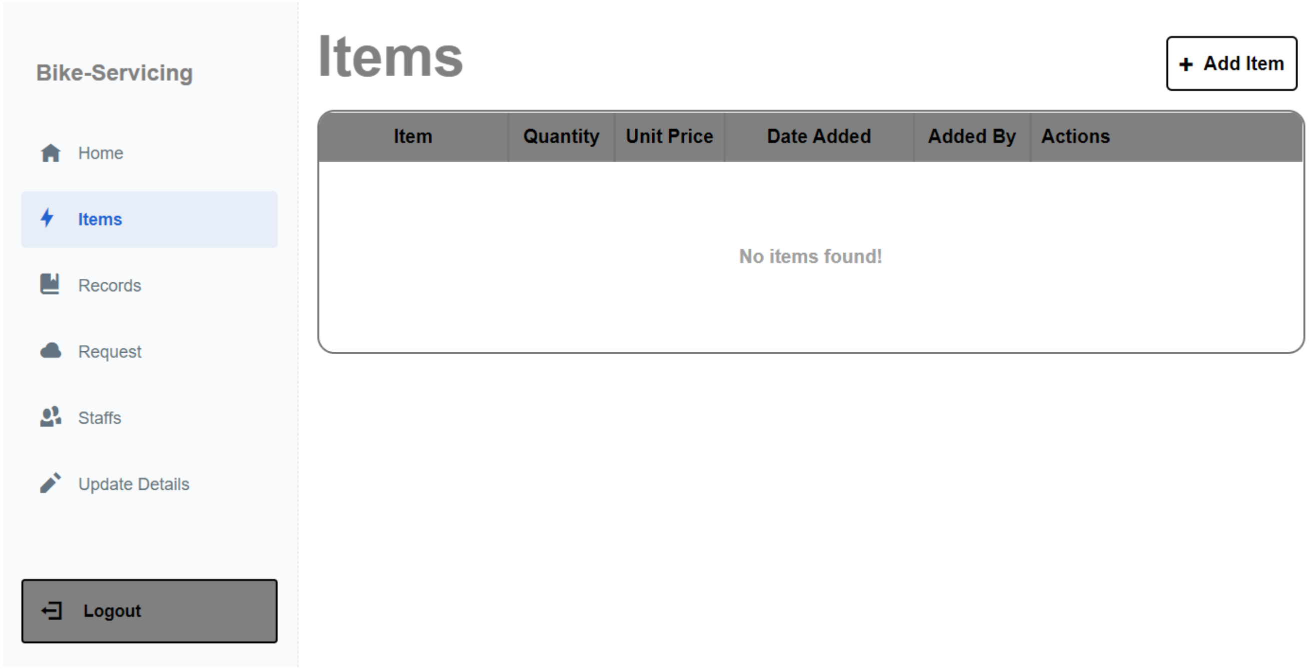
**Warning:** Please provide your current password for updating information!

- ✓ Figure 4:Change Name of staff or update name.

To add a new staff member or change the name of an existing staff member in the Bike-servicing System, you will need to follow the steps below:

- ✓ Click on the Update Details  and Name option in the main menu.
- ✓ In the "Change Name" section, enter the name of the new staff member in the "Name" field.
- ✓ (Optional) Enter any additional details about the staff member, such as their role or contact information.
- ✓ Click the "Change Name" button to add the new staff member to the system.

## 2.5 Items list show on staff



| Item | Quantity | Unit Price | Date Added | Added By | Actions |
|---|---|---|---|---|---|

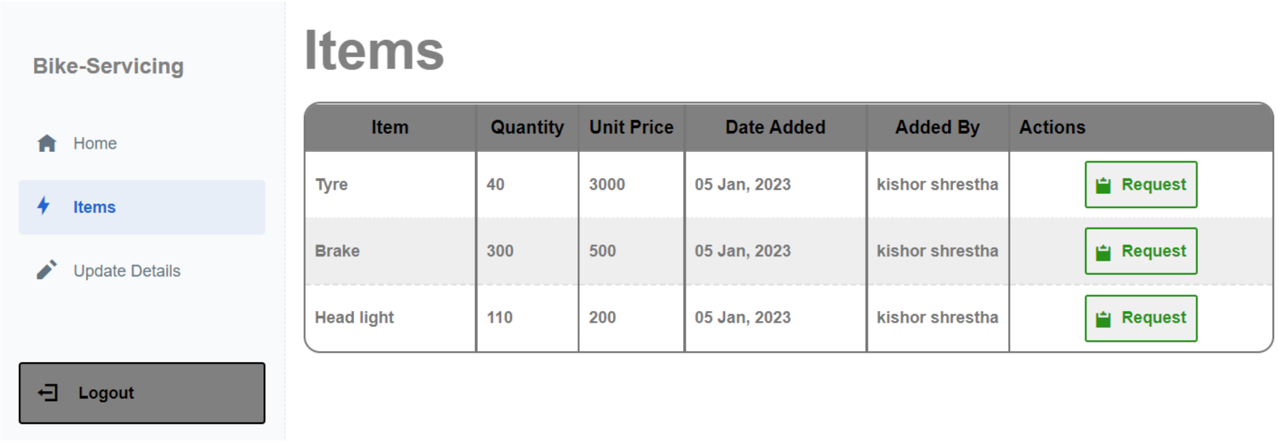
No items found!

✓ Figure 5:Items Table of Staff.

The Bike-servicing System likely has a feature that allows staff to view a list of all the inventory items in the system. To view this list, you will need to follow the steps below:

- ✓ Launch the Bike Servicing System.
- ✓ Click on the "Inventory" or "Items" option in the main menu.
- ✓ A list of all the inventory items in the system should be displayed on the screen. This list may include details such as the item name, quantity in stock, and any other relevant information.
- ✓ Depending on the specific design of the Bike Servicing System,you may also be able to filter or sort the list of items by different criteria, such as item category or stock level. Some systems may also allow you to search for specific items by name or other attributes.

## 2.6 Items listed on staff.



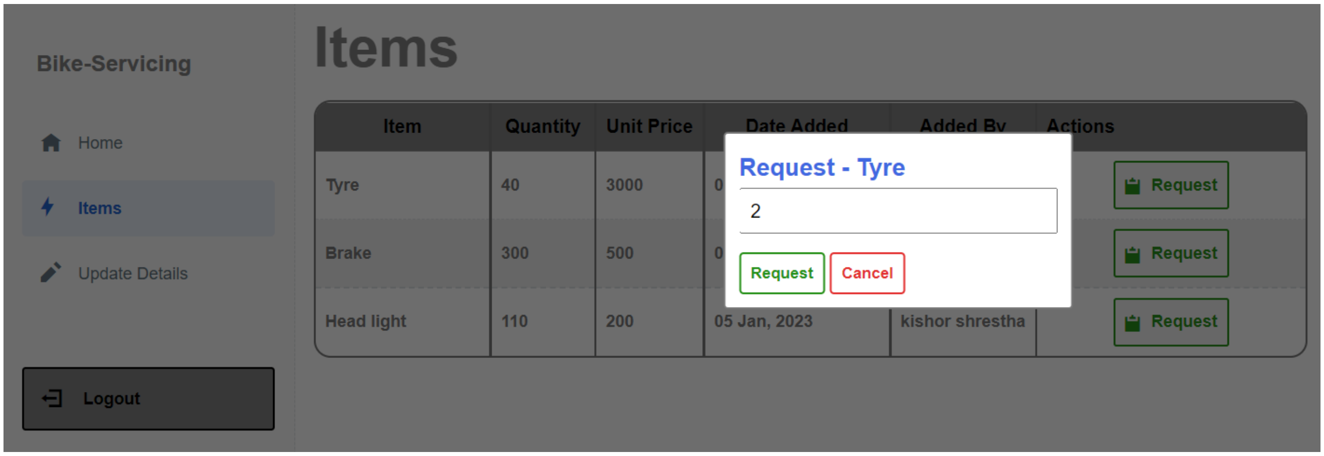
| Item | Quantity | Unit Price | Date Added | Added By | Actions |
|---|---|---|---|---|---|
| Tyre | 40 | 3000 | 05 Jan, 2023 | kishor shrestha | Request |
| Brake | 300 | 500 | 05 Jan, 2023 | kishor shrestha | Request |
| Head light | 110 | 200 | 05 Jan, 2023 | kishor shrestha | Request |

✓ Figure 6:Items name and there details of staff.

To add a new item to the list of inventory items in the Bike Center System, you will need to follow the steps below:

- ✓ Launch the Bike servicing center.
- ✓ Click on the "Inventory" or "Items" option in the main menu.
- ✓ In the "Add Item" section, enter the name of the new item in the "Name" field.
- ✓ (Optional) Enter any additional details about the item, such as its unit price, Date added, or action.
- ✓ Once you have approved by admin the new item to the list, it should be displayed in the staff table along with the other items. You may need to refresh the table or page in order to see the new item.

## 2.7 Requests Items to approve or delete

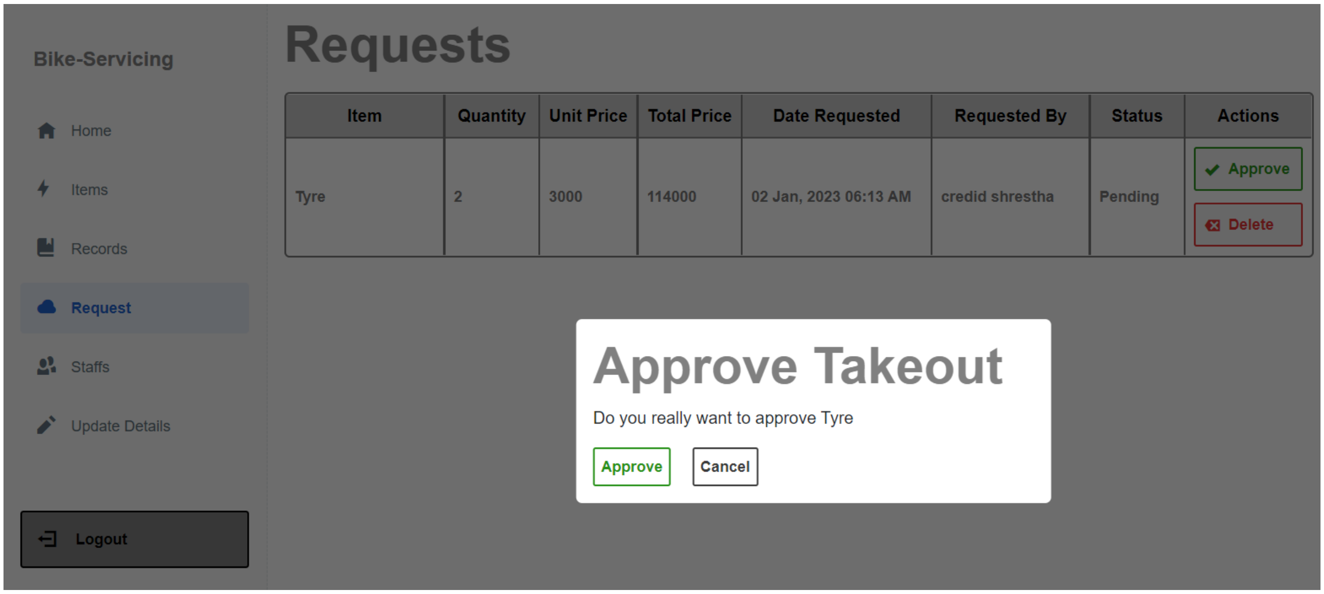


Items

| Item | Quantity | Unit Price | Date Added | Added By | Actions |
|---|---|---|---|---|---|
| Tyre | 40 | 3000 | 0 | | Request |
| Brake | 300 | 500 | 0 | | Request |
| Head light | 110 | 200 | 05 Jan, 2023 | kishor shrestha | Request |

Request - Tyre

2



Requests

| Item | Quantity | Unit Price | Total Price | Date Requested | Requested By | Status | Actions |
|---|---|---|---|---|---|---|---|
| Tyre | 2 | 3000 | 114000 | 02 Jan, 2023 06:13 AM | credid shrestha | Pending | Approve<br>Delete |

Approve Takeout

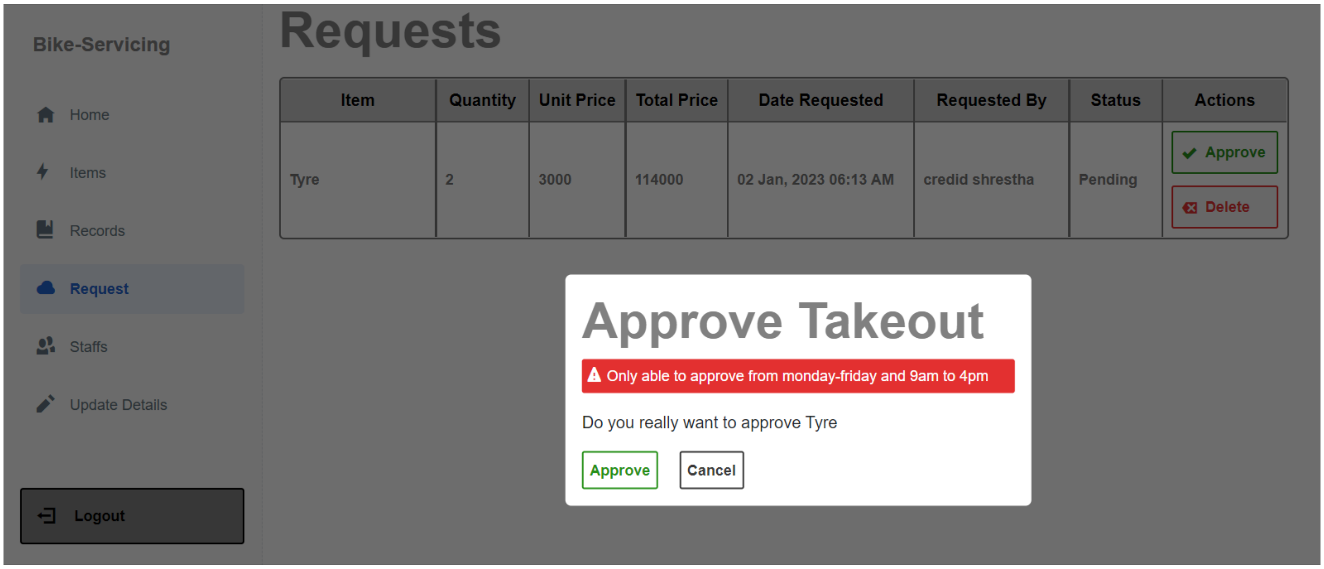
Do you really want to approve Tyre

✓ Figure 7:requesting items to admin.

In the Bike servicing, it like staff members are able to request certain items from the inventory, and that these requests need to be approved by an administrator before the items can be issued. To approve a staff request for an item, you will need to follow the steps below:

- ✓ Launch the Bike servicing System.
- ✓ Go to admin panel.
- ✓ Find the item that the staff member has requested in the list of inventory items.
- ✓ Click on the item to open its detail page or profile.
- ✓ In the "Approve Request" section, enter the quantity of the item that the staff member has requested.
- ✓ Click the "Approve" button to approve the request.
- ✓ Once you have approved the request, the requested quantity of the item should be removed from the inventory and made available for the staff member to use.

## 2.9 Approve Takeout due to Time Table



Requests

| Item | Quantity | Unit Price | Total Price | Date Requested | Requested By | Status | Actions |
|---|---|---|---|---|---|---|---|
| Tyre | 2 | 3000 | 114000 | 02 Jan, 2023 06:13 AM | credid shrestha | Pending | Approve<br>Delete |

Approve Takeout

⚠ Only able to approve from monday-friday and 9am to 4pm

Do you really want to approve Tyre

Approve Cancel

✓ Figure 8:requesting approve Takeout.

you have set certain hours during which staff members can request items from the inventory in the Bike servicing System, specifically from 9:00 AM to 4:00 PM on weekdays (Monday through Friday). To ensure that requests are only approved during these hours, you may want to consider setting up a system that checks the current time and only allows requests to be approved within the designated timeframe.

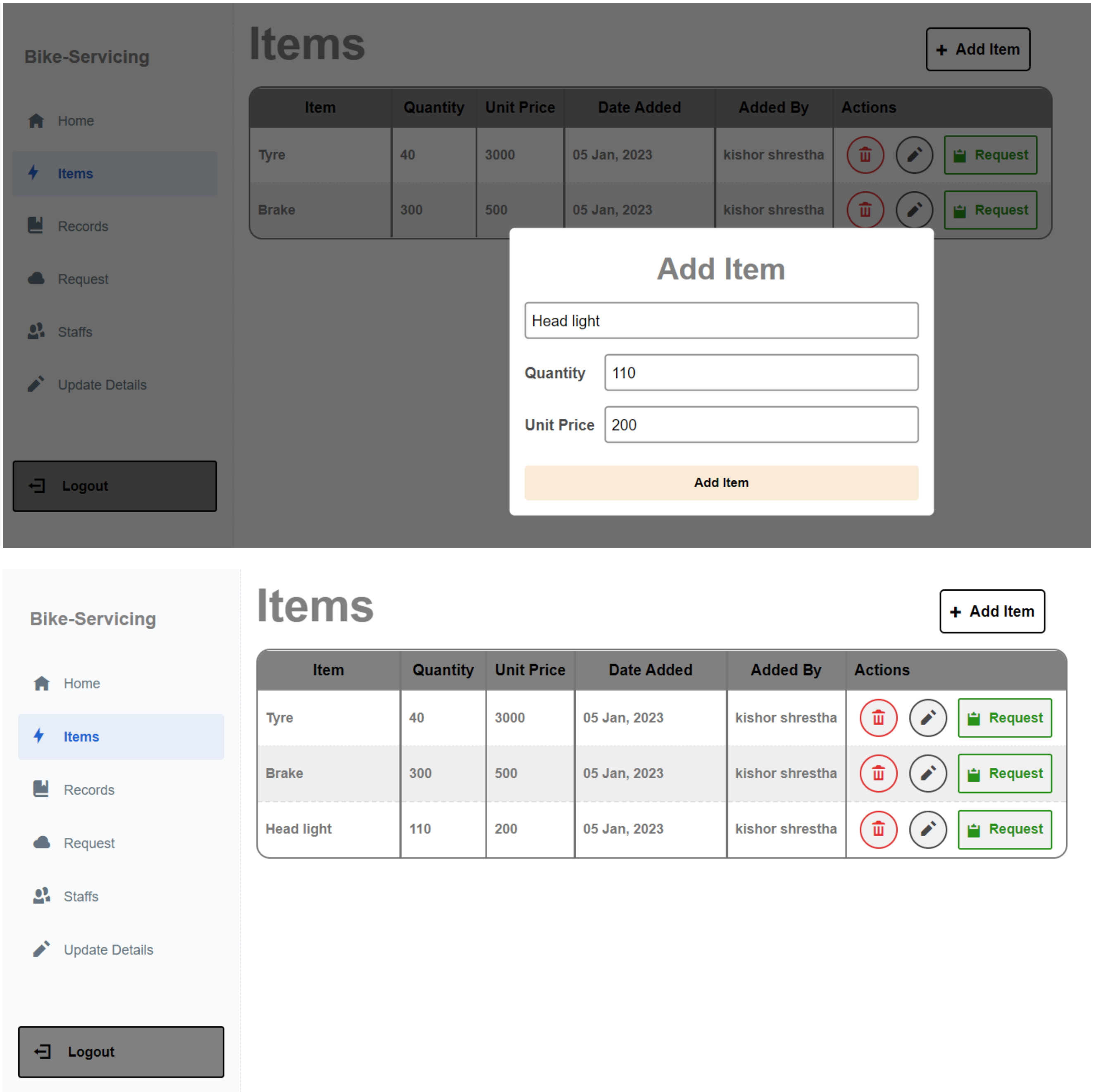
Here are some possible ways to implement this:

Use a built-in system clock or time function: Many programming languages, including C#, have built-in functions or libraries that allow you to retrieve the current time and date. You can use these functions to check the current time before approving a request, and to display an error message or notification if the request is being made outside of the designated hours.

Set up a timer or schedule: You could use a timer or scheduling system to automatically approve or deny requests based on the current time. For example, you might set up a timer that runs every minute and checks the current time. If the time is within the designated hours, the timer could automatically approve any pending requests.

Overall, by implementing a system to control the hours during which requests can be approved, you can help ensure that requests are only approved during the designated times and that the inventory is properly managed.

## 2.10 Add Items



| Item | Quantity | Unit Price | Date Added | Added By | Actions |
|---|---|---|---|---|---|
| Tyre | 40 | 3000 | 05 Jan, 2023 | kishor shrestha | Request |
| Brake | 300 | 500 | 05 Jan, 2023 | kishor shrestha | Request |

Add Item

Head light

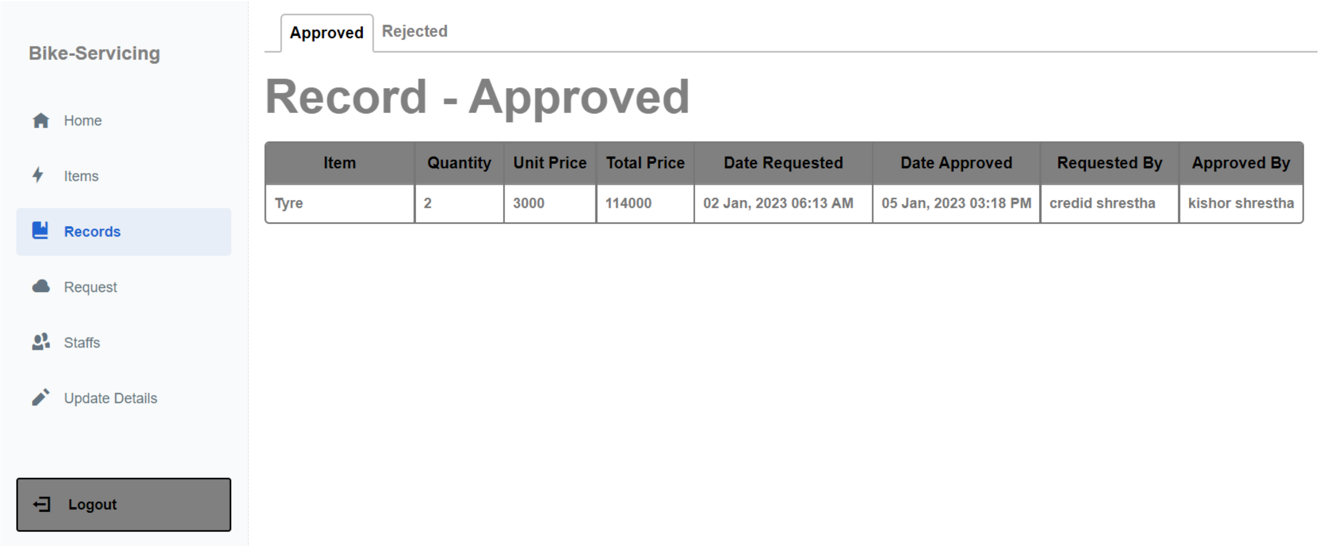
Quantity: 110

Unit Price: 200

| Item | Quantity | Unit Price | Date Added | Added By | Actions |
|---|---|---|---|---|---|
| Tyre | 40 | 3000 | 05 Jan, 2023 | kishor shrestha | Request |
| Brake | 300 | 500 | 05 Jan, 2023 | kishor shrestha | Request |
| Head light | 110 | 200 | 05 Jan, 2023 | kishor shrestha | Request |

✓ Figure 9:Figure of Add items.

## 2.11 Record Approved by admin



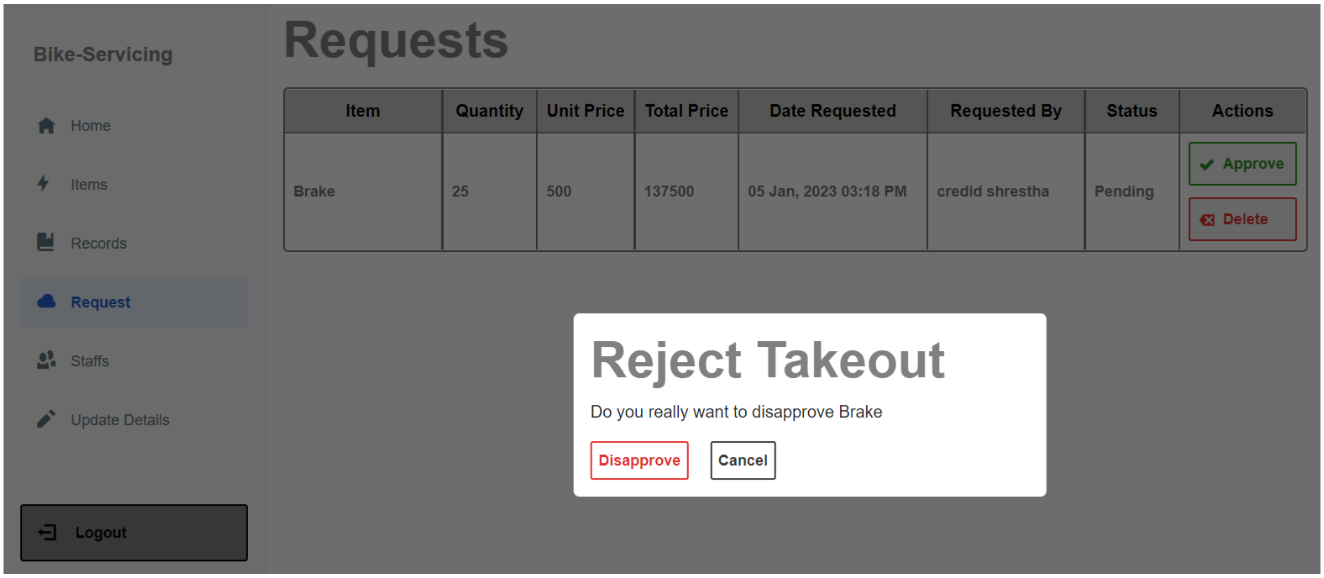
Record - Approved

| Item | Quantity | Unit Price | Total Price | Date Requested | Date Approved | Requested By | Approved By |
|---|---|---|---|---|---|---|---|
| Tyre | 2 | 3000 | 114000 | 02 Jan, 2023 06:13 AM | 05 Jan, 2023 03:18 PM | credid shrestha | kishor shrestha |

- ✓ Figure 10:Figure of Record is approved by admin

you are using the Bike servicing System to track and manage requests for items from the inventory, and that these requests need to be approved by an administrator before they can be fulfilled. When an administrator approves a request, it is likely that this action is recorded in the system for later reference.

## 2.12 Record deletes by admin



Requests

| Item | Quantity | Unit Price | Total Price | Date Requested | Requested By | Status | Actions |
|---|---|---|---|---|---|---|---|
| Brake | 25 | 500 | 137500 | 05 Jan, 2023 03:18 PM | credid shrestha | Pending | Approve<br>Delete |

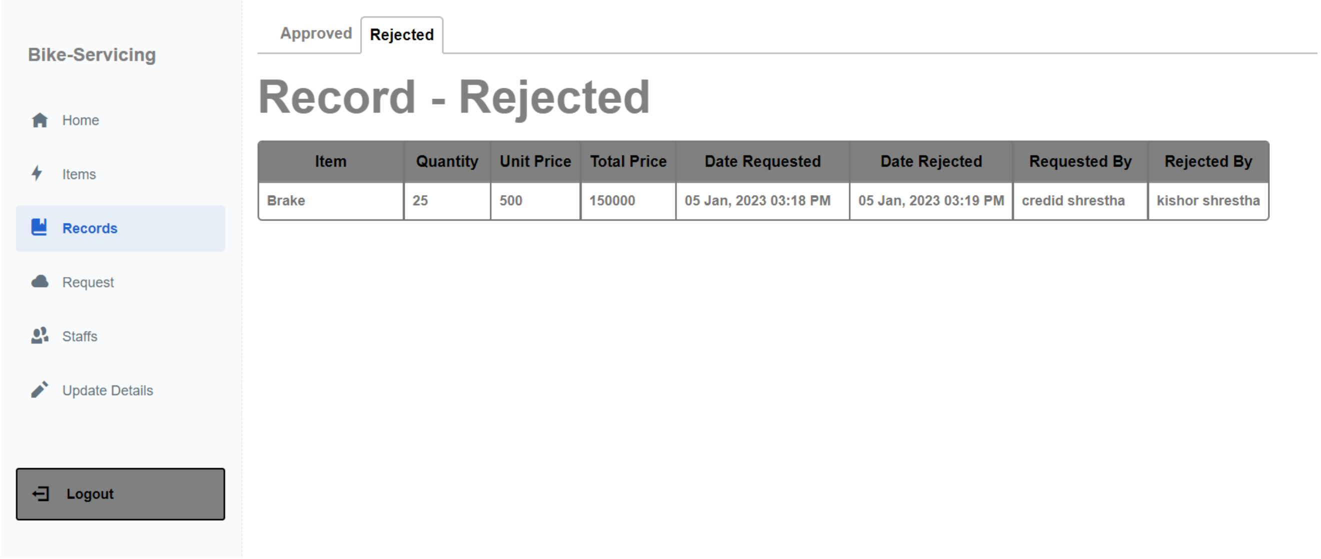
Reject Takeout

Do you really want to disapprove Brake

- ✓ Figure 11:Figure of request reject by admin.

In the Bike servicing System, it requests for items from the inventory can be either approved or rejected by an administrator. If a request is rejected, it is likely that this action is recorded in the system for later reference.

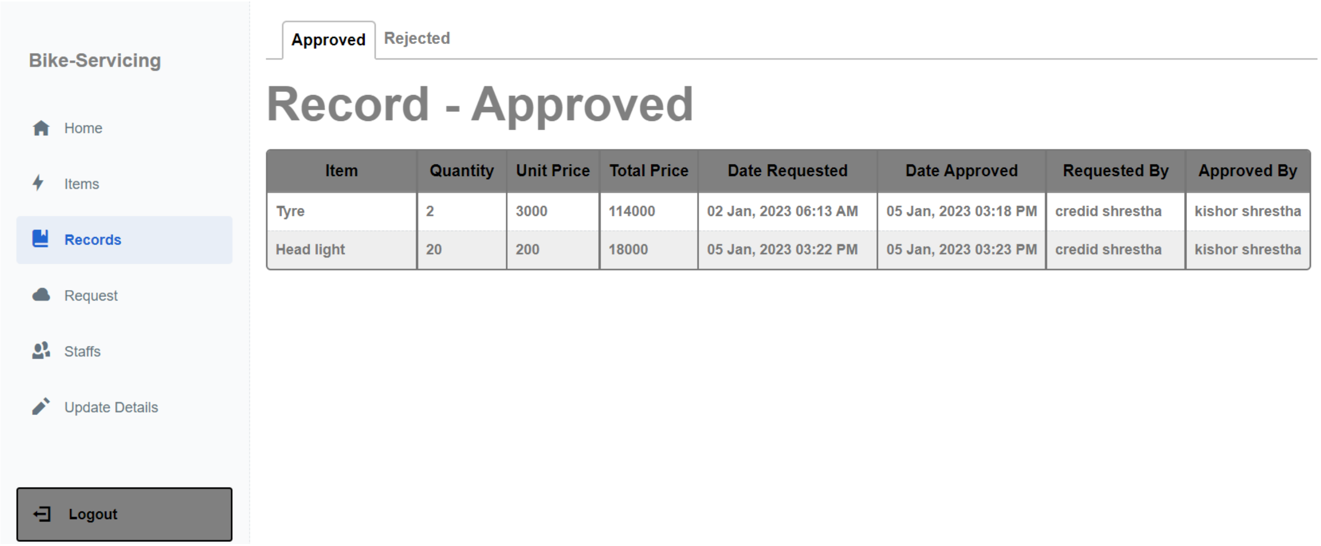
## 2.13 Record rejected by admin on table



Record - Rejected

| Item | Quantity | Unit Price | Total Price | Date Requested | Date Rejected | Requested By | Rejected By |
|---|---|---|---|---|---|---|---|
| Brake | 25 | 500 | 150000 | 05 Jan, 2023 03:18 PM | 05 Jan, 2023 03:19 PM | credid shrestha | kishor shrestha |

## 2.14 Record approved by admin on table

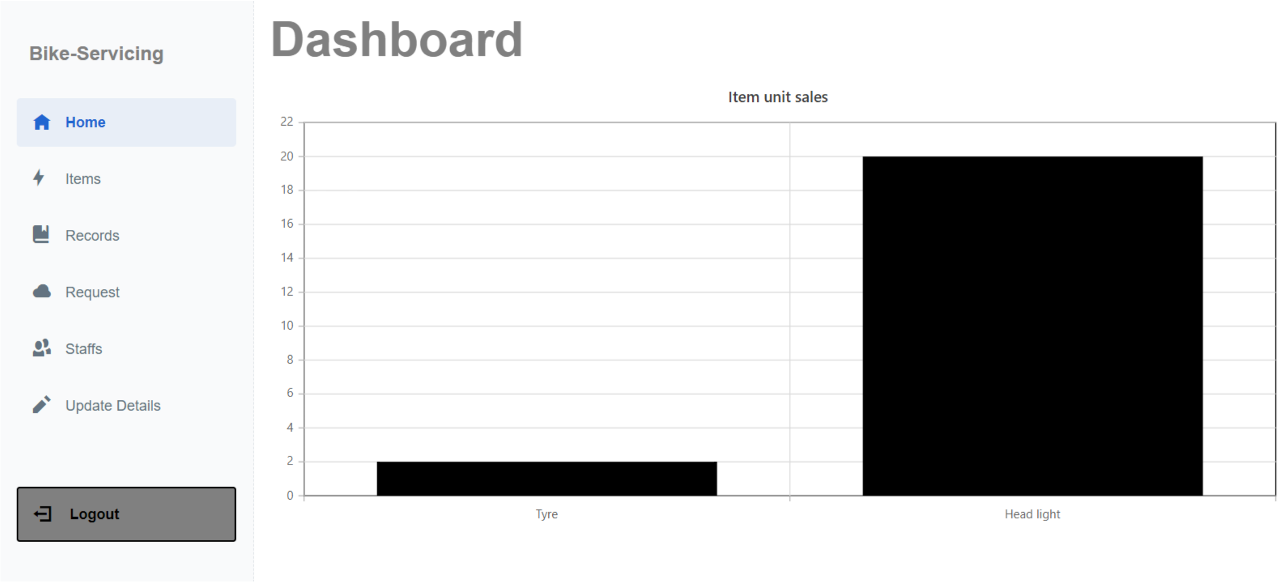


Record - Approved

| Item | Quantity | Unit Price | Total Price | Date Requested | Date Approved | Requested By | Approved By |
|---|---|---|---|---|---|---|---|
| Tyre | 2 | 3000 | 114000 | 02 Jan, 2023 06:13 AM | 05 Jan, 2023 03:18 PM | credid shrestha | kishor shrestha |
| Head light | 20 | 200 | 18000 | 05 Jan, 2023 03:22 PM | 05 Jan, 2023 03:23 PM | credid shrestha | kishor shrestha |

you are using the Bike servicing System to track and manage requests for items from the inventory, and that these requests need to be approved by an administrator before they can be fulfilled. When an administrator approves a request, it is likely that this action is recorded in the system for later reference.

## 2.15 Dashboard of Items listed on bar graph.

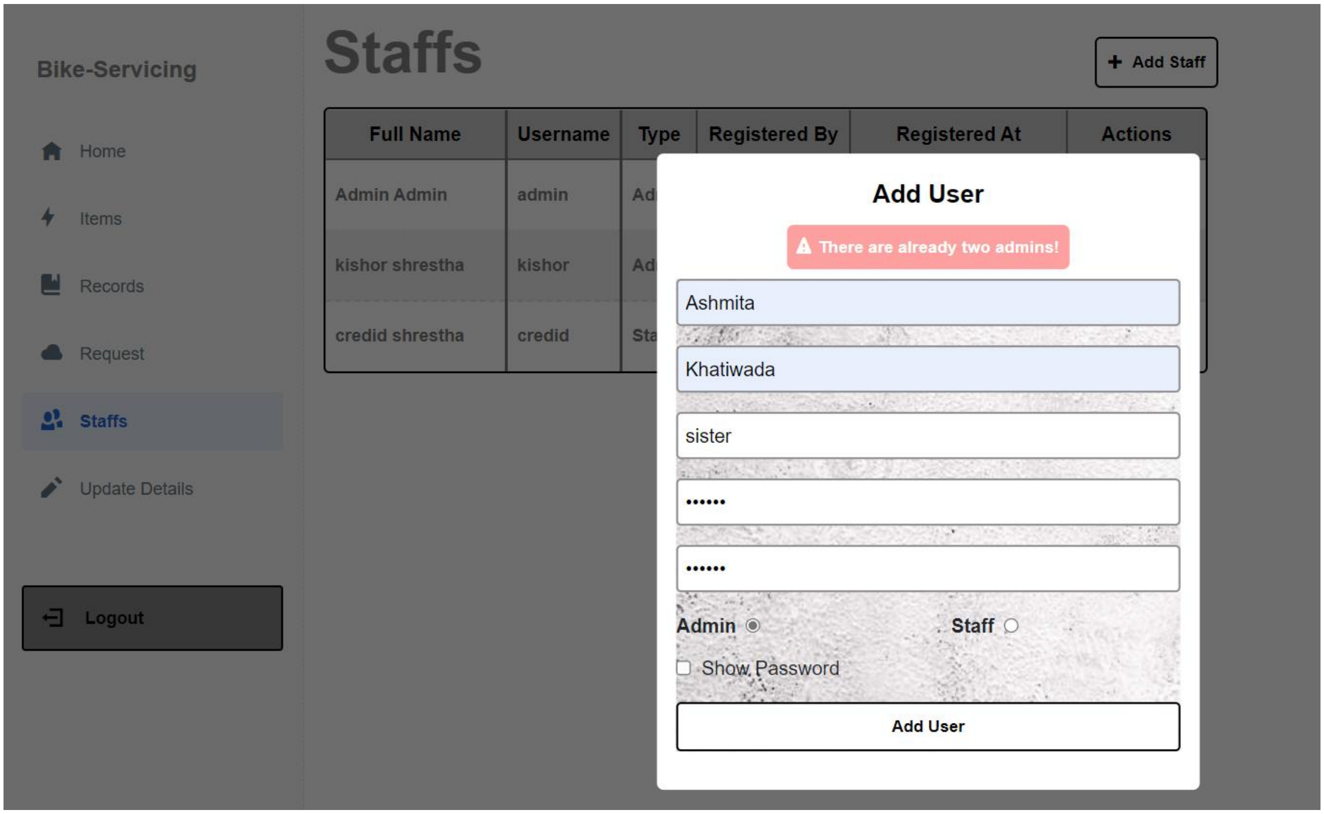


- ✓ Figure 12:Figure of Bar graph of items.

Bike Servicing System includes a dashboard feature that displays a bar graph showing the quantity of different items in the inventory. This type of graph can be a useful way to visualize and understand trends in the inventory over time, and can help administrators make informed decisions about restocking and resource allocation.

## 2.16 Restriction on admin

There is limitation in system when there is include of more than two admin.



Staffs

| Full Name | Username | Type | Registered By | Registered At | Actions |
|---|---|---|---|---|---|
| Admin Admin | admin | Ad | | | |
| kishor shrestha | kishor | Ad | | | |
| credid shrestha | credid | Sta | | | |

Add User

⚠ There are already two admins!

Ashmita

Khatiwada

sister

••••••

••••••

Admin ● Staff ○

□ Show Password

*Figure 13: limitation in system when there is include of more than two admin*

# 3 System Architecture

A bike servicing system is a computer-based system that is used to manage the maintenance and repair of bicycles. It is used to keep track of bike servicing schedules, maintenance and repair tasks, and customer information.

There are several different types of architecture that can be used for a bike servicing system, including:

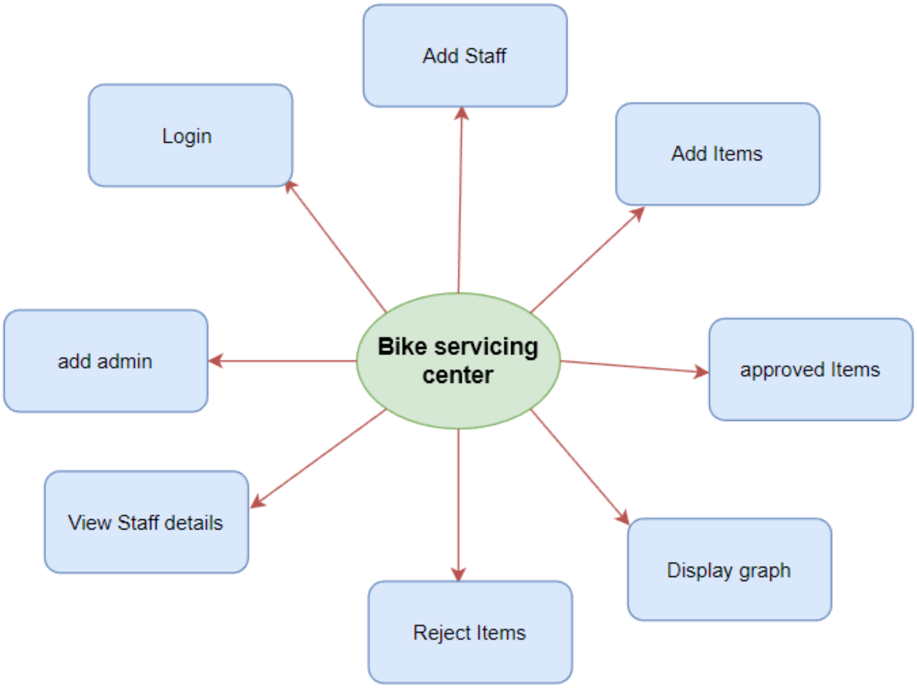
Client-server architecture: In this architecture, the system is divided into two main components: the client and the server. The client is typically a computer or device that is used to access the system, while the server is a computer that stores the database and other system components. The client sends requests to the server, which processes the requests and sends back the appropriate response.

Cloud-based architecture: In this architecture, the system is hosted on a cloud-based platform, such as Amazon Web Services or Microsoft Azure. This allows users to access the system from anywhere with an internet connection.

Standalone architecture: In this architecture, the system is installed on a single computer and does not require a connection to a network or the internet. This can be useful for small bike repair shops that do not need to share data with other systems.

Hybrid architecture: In this architecture, the system is made up of a combination of the above architectures. For example, the system may be hosted on a cloud-based platform, but also have a standalone component installed on individual computers.

The specific architecture of a bike servicing system will depend on the needs and resources of the business using it. (edrawmax, n.d.)



✓ Figure 14:Figure of System Archicheture.

# 4 Classes and their purpose

In a bike servicing system written in C#, the classes might include:

1. Customer: This class might represent a customer of the bike servicing center. It could include attributes such as name, address, and phone number, as well as methods for adding and retrieving customer information.

2. Bike: This class might represent a bike that is being serviced at the center. It could include attributes such as serial number, make, model, and year, as well as methods for adding and retrieving bike information.

3. ServiceOrder: This class might represent a service order for a bike at the center. It could include attributes such as order number, customer, bike, service type, and service date, as well as methods for creating and managing service orders.

4. Inventory: This class might represent the inventory of parts and supplies at the bike servicing center. It could include attributes such as item name, item number, and quantity, as well as methods for managing the inventory and checking stock levels.

5. Technician: This class might represent a technician who works at the bike servicing center. It could include attributes such as name, contact information, and skill level, as well as methods for managing technician information and assigning technicians to service orders.

These are just a few examples of the classes that might be included in a bike servicing system written in C#. The specific classes and attributes included in the system will depend on the needs and requirements of the system.

## 4.1 Purpose of Login Staff

| **Staff** | **IDE/Self made** | **Purpose** |
|---|---|---|
| Login.cs | IDE | This class's objective is to enable workers and administrators to access their individual dashboards. |
| Login.cs[Design] | Self-made | It is a login form created to accept the staff and admin's username and password to login. |

✓ Table 1:Table of Login Form.

## 4.2 Purpose of Staff Dashboard class

| Class | IDE/Self made | Purpose |
|---|---|---|
| StaffDashboard.cs | IDE | This class's goal is to collect client information for the creation of bike parts. The rate is shown to the staff in the admin-specified grid.<br>to produce daily and weekly reports.<br>to see the client information. |
| StaffDashboard.cs[Design] | Self-made | It also has a sign-out button that takes you back to the login page as well as buttons for entering customer entries, examining customer information, and creating weekly and daily reports. |

✓ Table 2:Table of Staff Dashboard.

## 4.3 Purpose of Admin Dashboard class

| **Class** | **IDE/Self made** | **Purpose** |
|---|---|---|
| AdminDashboard.cs | IDE | This class's function is to include the Items that can be imported into the staff dashboard for the purpose of entering client information.<br>To see the staff's daily report, click here (category of items). the weekly report to view. Sorting a weekly report by the quantity of items. |
| AdminDashboard.cs[Design] | Self-made | It has buttons for adding Items, reading the daily and weekly reports, sorting by total Items, and a sign-out link that takes users back to the login page. |

✓ Table 3:Table of Admin Dashboard.

## 4.4 Purpose of Daily Report class

| Class | IDE/Self made | Purpose |
|---|---|---|
| DailyReport.cs | Self-made | The total Items and category for the item are provided by this class, which was created to act as a datatype. |

- ✓ Table 4:Table of Daily repost.

## 4.5 Purpose of Weekly Report class

| Class | IDE/Self made | Purpose |
|---|---|---|
| WeeklyReport.cs | Self-made | This class was created to function as a datatype, offering methods to obtain the day, total number of things, and the time required to extract and approve items for an object. |

- ✓ Table 5:Table of Weekly report.

# 5 Reflection of the learning experience

C# was used as the programming language, while Visual Studio was used as the IDE, to create the Bike Servicing system. I had never worked on a project like it before, so the entire program was new. In the beginning, I had a hard time getting used to Visual Studio and the programming language. However, the issue was eventually resolved with more time and effort. The tutorial projects and lecture slides were both very helpful in resolving a number of issues. The project went from having little information at the beginning to having more knowledge at the finish, but overall it went well. I found that Visual Studio was the user-interactive IDE that offered every component needed during the system's development while I was working on it.

# 6 Conclusion

We have been working on a project to create a bike servicing system as part of a course or module. A desktop application is a type of software that is installed on a computer and can be accessed through the operating system.

To create a fully-functional desktop application for a bike servicing system, you will need to consider the various features and functionality that the application should have. This might include the ability to enter and manage customer and bike information, create and track service orders, manage inventory, and assign technicians to service orders.

You will also need to consider the user interface of the application, and how it will be used by technicians and other employees at the bike servicing center. This might include designing user-friendly forms and menus, and ensuring that the application is easy to navigate.

It is important to carefully plan and design your application before beginning development, as this will help to ensure that the final product meets the needs of the users and is of high quality.

## 6.1 Evaluation of work

There are several factors that you can consider when evaluating the work for a bike servicing system:

1. Functionality: Does the system have all of the necessary features and functionality to support the needs of the bike servicing center? Does it allow technicians to enter and manage customer and bike information, create and track service orders, manage inventory, and assign technicians to service orders?

2. User experience: Is the system easy to use and navigate? Are the forms and menus intuitive and user-friendly?

3. Performance: Does the system run smoothly and efficiently, with fast load times and no errors or bugs?

4. Data accuracy and integrity: Is the data entered into the system accurate and reliable? Is it stored securely and backed up regularly to prevent loss of data?

5. Security: Are appropriate measures in place to protect the system and data from unauthorized access or tampering?

6. Maintenance and support: Is there a plan in place for ongoing maintenance and support of the system? Are there processes in place to address any issues or bugs that arise?

By considering these factors, you can evaluate the work for your bike servicing system and determine whether it meets the needs of the users and is of high quality.

## 6.2 Learning Outcomes

There are several learning outcomes that you might expect to achieve by working on a bike servicing system:

Improved programming skills: You may develop or improve your programming skills, such as understanding how to design and implement classes, use data structures and algorithms, and write clean and efficient code.

Knowledge of software development process: You may learn about the software development process, including how to plan, design, implement, test, and deploy a software system.

Familiarity with software development tools and technologies: You may become familiar with various software development tools and technologies, such as programming languages, libraries, frameworks, and version control systems.

Understanding of user experience design: You may learn about user experience design and how to create user-friendly and intuitive interfaces for software applications.

Understanding of database design: You may learn about database design and how to create a database schema to store and manage data for a software system.

Understanding of software testing and debugging: You may learn about software testing and debugging, and how to identify and fix errors and bugs in your code.

By working on a bike servicing system, you may also develop problem-solving skills, communication skills, and the ability to work independently or as part of a team.

# 7 Bibliography

*edrawmax*. (n.d.). Retrieved from edrawmax: https://www.edrawmax.com/templates/1018832/

*source code*. (2023, 1 5). Retrieved from source code: https://code-projects.org/inventory-system-in-c-with-source-code/#:~:text=Inventory%20System%20is%20a%20simple,stock%20out%2C%20returns%2C%20etc.