Requirement Model Update

Overview – I have made some changes to the domain classes and the uses cases since the analysis report. I have had to add some domain classes to the application. I got rid of a domain class receipt and replaced it with purchase. Then I added PurchaseLineItem, Payment, ReOrderItems and then I renamed bike service to ServiceTicket.

Event Use Case List

|  |  |
| --- | --- |
| Event | Use Cases |
| New Employee | Create Employee |
| Employee Info Changes | Update Employee |
| Employee Arrives At work | Employee Clock In |
| Employee gets off work | Employee Clock out |
| Find Customer | Customer lookup |
| Customer Is not in system | Create New Customer |
| Customer Needs Bike Serviced | Enter Service Ticket |
| Service Started, Completed, Or Paid For | Update Service Ticket Status |
| Application Starts | Load Service Queue |
| New Service Created | Create New Service Item |
| Service info changes | Update Service Item |
| New Product not in system | Create Product |
| Product Information Changes | Update Product |
| New Bike not in system | Create New Bike |
| Bike Information Changes | Update new Bike |
| Employee wants to know what products to reorder | ReOrderReport |
| Customer Wants to buys something | Create Purchase |
| Purchase entered | UpdateQtyOH |
| Customer brings Bike in that isn’t in system | Create Customer Bike |
|  |  |
|  |  |

Domain Model Class Diagram



Software Design Approach

I used a three layer Object oriented design. So I have a presentation layer, domain layer, and then a data access layer. The Presentation layer doesn’t talk directly to the data access layer. The Domain Layer Can Communicate with the presentation layer and the data access layer and then the data access layer can communicate with the domain layer.

Package diagram



Enter Service Ticket System Sequence Diagram



Enter Purchase Sequence Diagram

Database Design

I am using access for the dbms. As it was easily accessible to me and didn’t require any special licenses or software to setup. The database consists of 13 tables. I set up relationships between the tables so that

List of Tables

Service(ServiceID, ServiceName, ServiceType, ServiceDescrip, ServiceCost)

ServiceTicketLine(ID, *TicketID*, *ServiceID*)

ServiceTicket(TicketID, Descrip, Cost, Status, *CustID*, *BikeID*)

ServiceTicketEmpployees(ID, EmployeeAction, *TicketID*, *EmpID*)

Employee(EmpID, EmpFirstName, EmpLastName, EmpAddress1, EmpAddress2, EmpCity, EmpState, EmpZip, EmpPhone, EmpPayRate)

LoginCredintials(ID, LoginKey, *EmpID*)

Purchase(PurchaseID, SubTotal, Tax, Total, *CustID*, *EmpID*)

Payment(ID, PaymentType, Amount, CardType, CardNumber, *PurchaseID*)

PurchaseLineItem(ID, ItemName, ItemCost, ItemQty, TotalCost, ItemType, *PurchaseID*, *ItemID*)

Customer(CustID, CustFirstName, CustLastName, CustAddress1, CustAddress2, CustCity, CustState, CustZip, CustPhone)

CustomerBike(CBikeID, CBikeBrand, CBikeModel, CBikeSerial, *CustID*)

Prod(prodID, prodSKU, prodName, prodType, prodCost,qtyOH, reOrderPt)

NBike(nBikeID nBikeBrand, nBikeModel, nBikeType, nBikeCost, qtyOH, reOrder)

Database ERD On following 2 pages

List Constraints

Customer - None

Service – none

Prod – none

Employee none

NBike – none

CustomerBike - CustID

Purchase – CustID, EmpID

ServiceTicketLine – ServiceID, TIcketID

PurchaseLineItem - PurchaseID

ServiceTicket – CustID

SeviceTicketEmployee – TicketID, EmpID

LoginCredintials - EmpID

Printouts of each table

System Controls Design

I have incorporated system controls into all the database access controls, database backup/recovery, error trapping and handling, and then I have added some text boxes that are masked and will only allow certain number and type of characters to be entered. and validation of entered information. During the database access there is error handling to handle any database errors that may arise. When a database exception is thrown the application will show an error message saying that there was a database error and then will write the error message out to the console. Then Most of the methods within the application have error handling to handle various exceptions. When an exception is thrown it will output a msg. Then some of the text have been formatted so that the text that is displayed is formatted as currency. As for backing up and recovering the database I have added two menu options that will allow the user to backup the database and then restore it, also on application close the database gets backed up. For text validation I have created a class called Validation, this class has methods that will evaluate the entered string against a regular expression statement. There is a method to test:

AlphaNumeric – allows a-z, A-Z, 0-9 and Spaces, also checks to make sure string isn’t blank

Alpha - allows a-z, A-Z and Spaces, also checks to make sure string isn’t blank

Alpha Period – allows a-z, A-z, spaces, and Periods, also checks to make sure string isn’t blank

AlphaNumericPerid – allows a-z, A-Z, 0-9, spaces, and periods, also checks to make sure string isn’t blank

Address 2 –For Address Line 2 allows a-z, A-Z, 0-9, spaces. This Allows Blanks

Numeric – allows 0-9 only, also checks to make sure string isn’t blank

Cost – 0-9, period, and comma, also checks to make sure string isn’t blank

NotBlank – validates string isn’t left blank

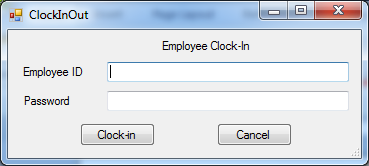
NotSelected – validates If credit card type is selected

Dialog Design

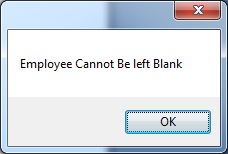
List of user cases and Dialogs

|  |
| --- |
| Create Employee |
| Update Employee |
| Employee Clock In |
| Employee Clock out |
| Customer lookup |
| Create New Customer |
| Enter Service Ticket |
| Update Service Ticket Status |
| Load Service Queue |
| Create New Service Item |
| Update Service Item |
| Create Product |
| Update Product |
| Create New Bike |
| Update new Bike |
| ReOrderReport |
| Create Purchase |
| Update QOH |
| Create Customer Bike  Backup Database  Restore Database |

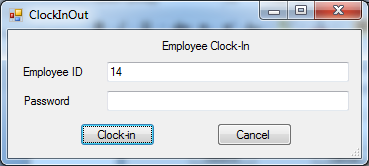
Enter Service Ticket Storyboards

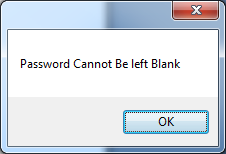


Initial Employee Clock-in on application start.

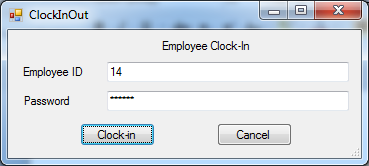


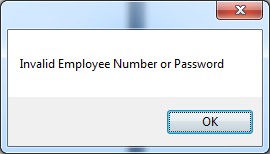
Trying to Clock-in without entering employee number



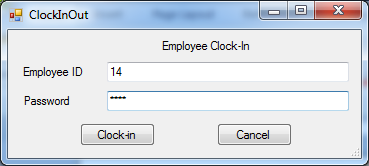


Employee Left Password Blank and tried to Clock-in

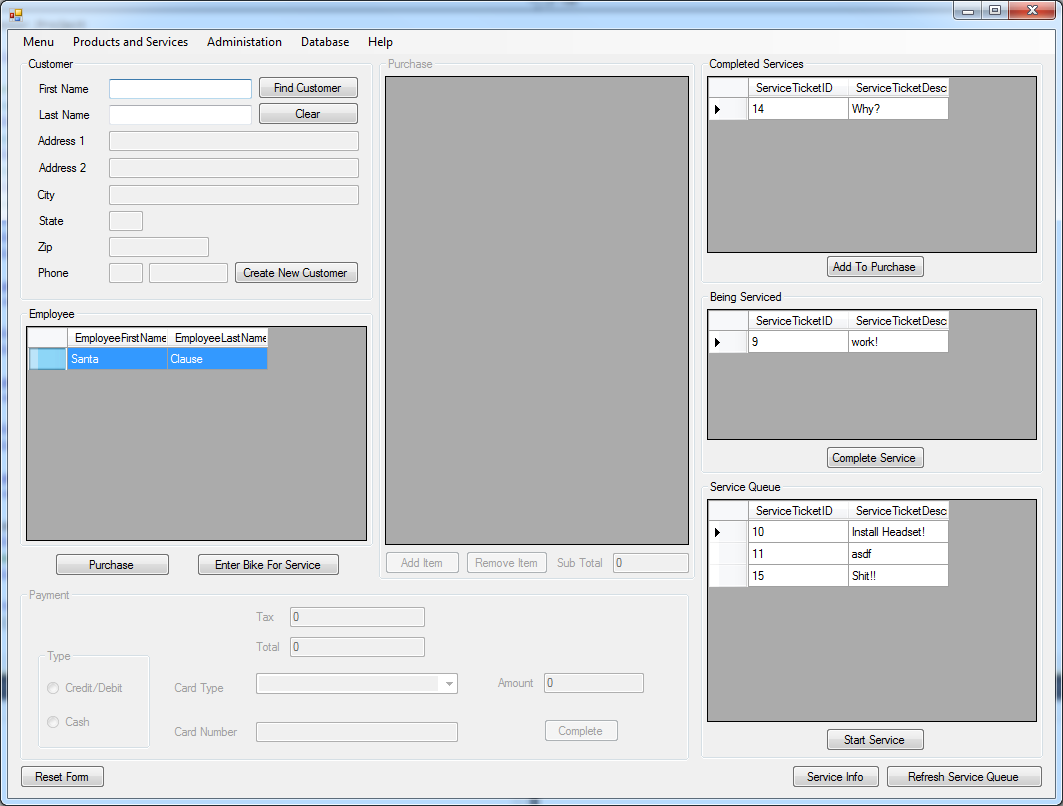




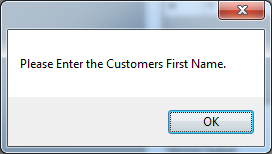
Employee Entered Wrong Password



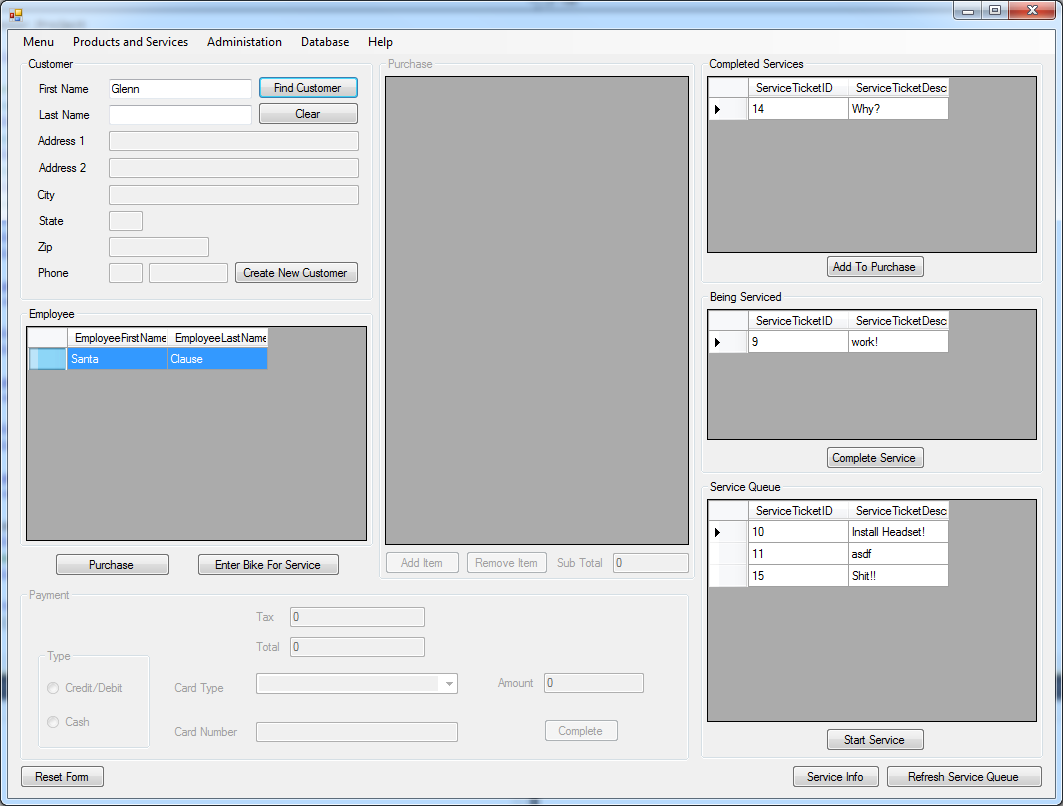
Entered Employee Number and Password



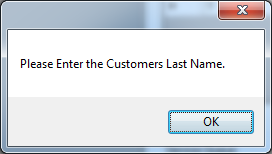
Employee Appears In Employee List



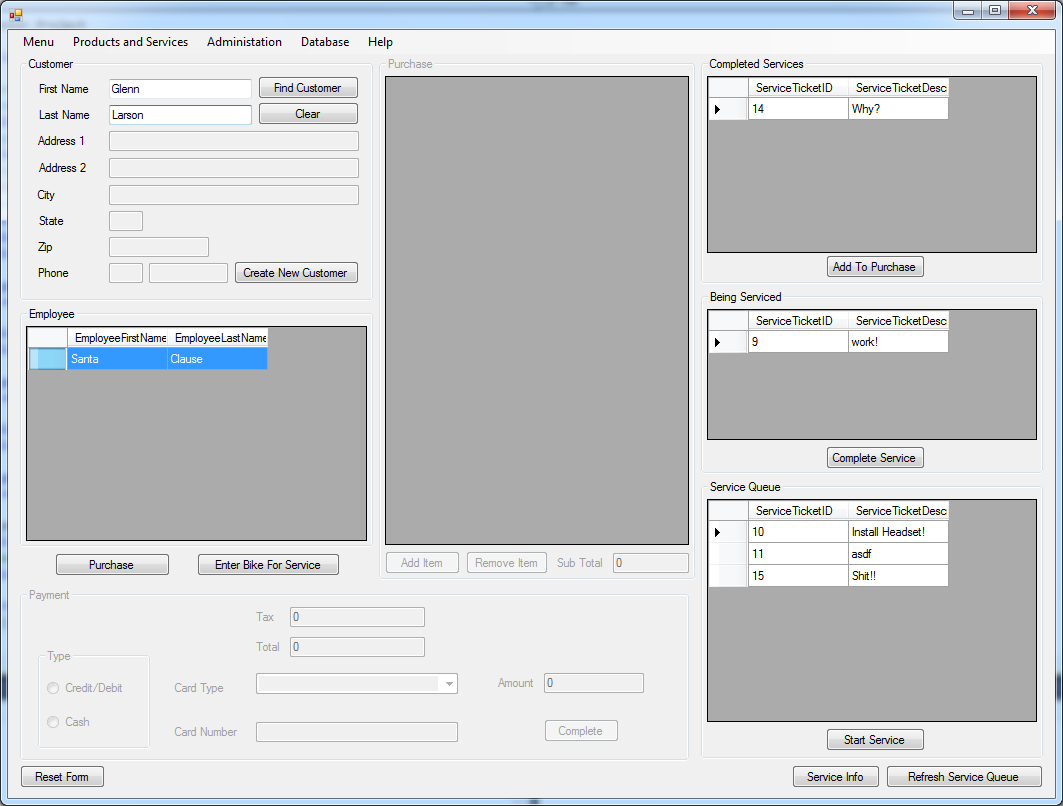
Employee Tries to lookup customer without entering customer first name



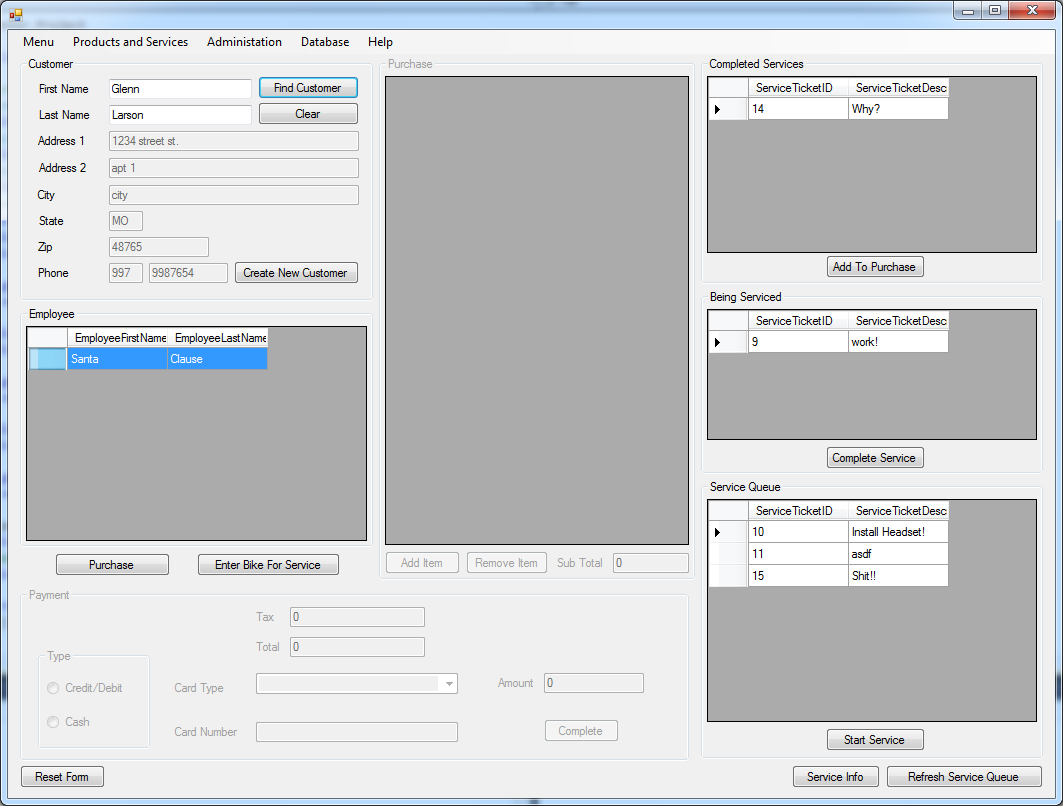
Employee Entered Customer First name but no last name



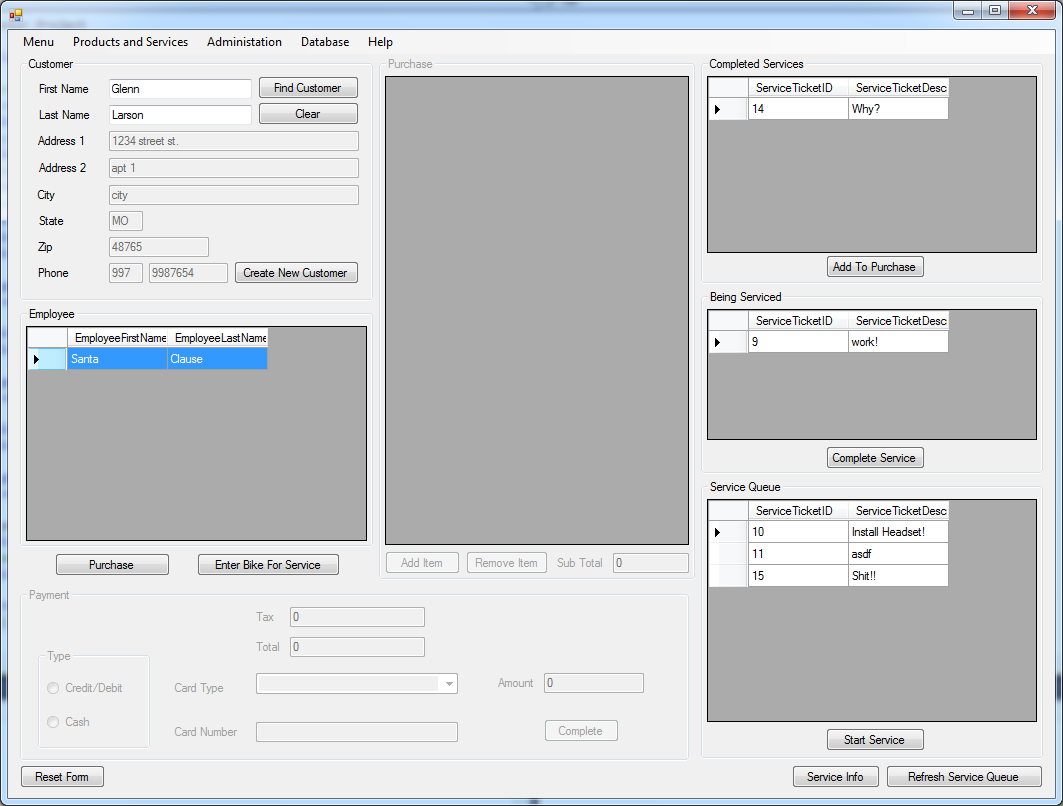
Error message



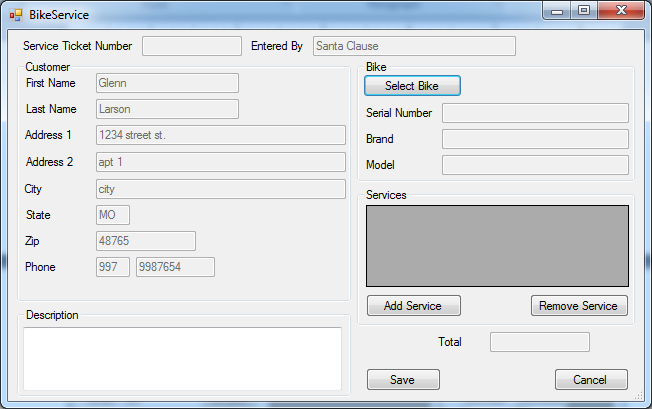
Employee Entered Customers first and last name and clicks find customer



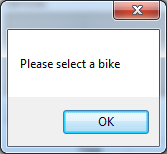
Customer Information is populated



Employee Selects their name from employee working list then Clicks Enter Bike For Service

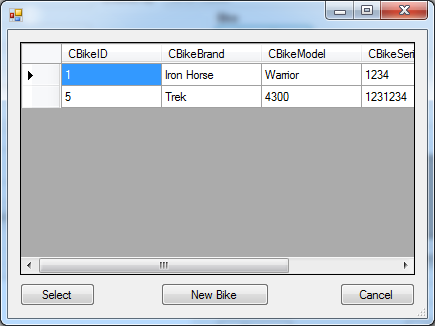


Bike Service Window Appears Populated With Customer Information and The Employees Name who entered it

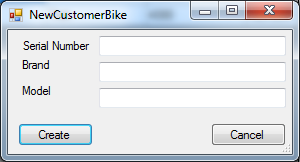


Error when employee tries to add service without selecting bike

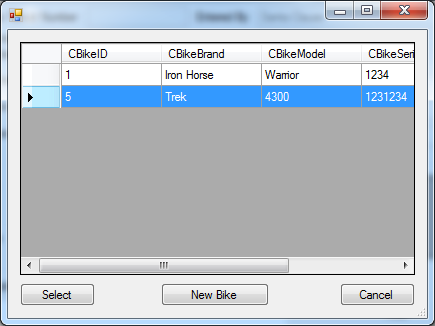
The Employee then clicks select bike button



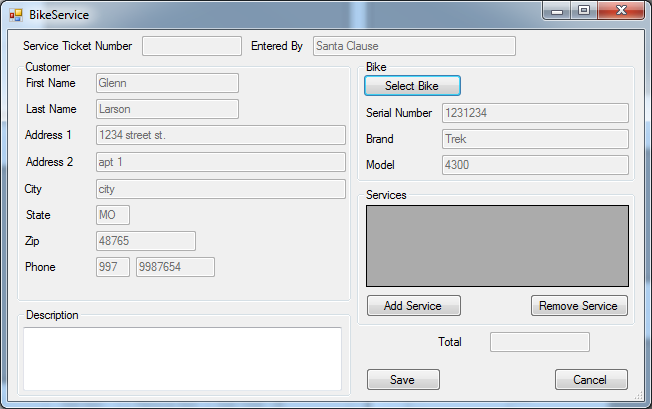
The Customer Bike List appears, if the customer bike isn’t listed then click new bike and enter bike information(Optional)



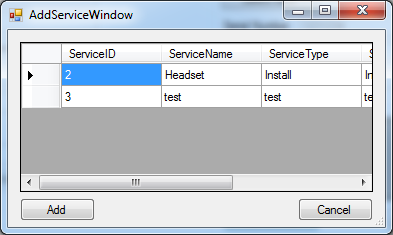
New customer bike Screen appears



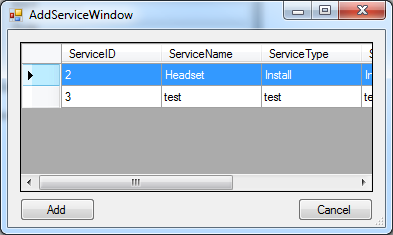
Other wise Employee Selects Bike and clicks select



Bike information populates in service ticket window Employee Then Clicks Add Service Btn



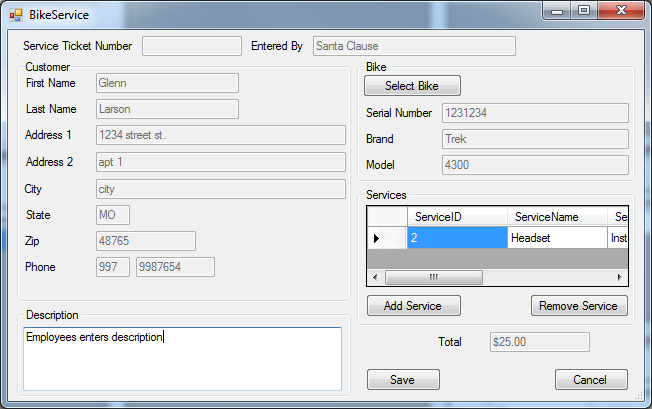
Add service window Appears



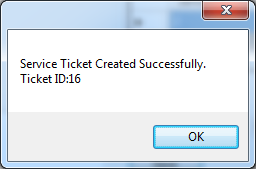
Employee Selects Service



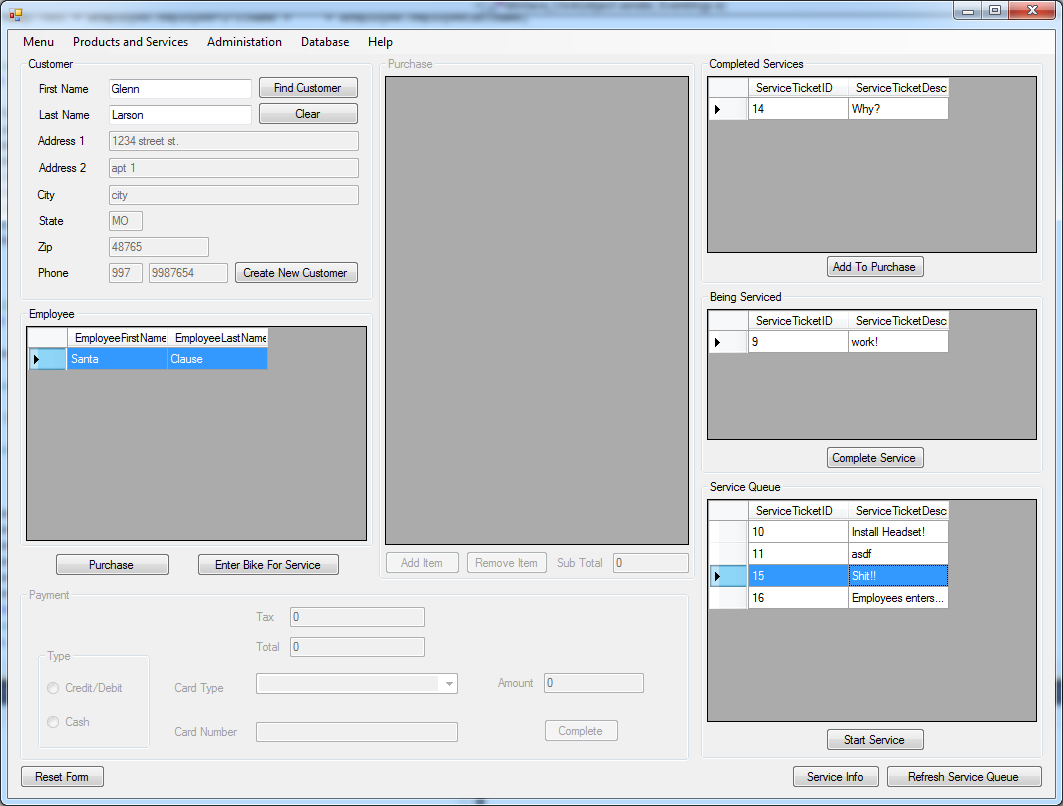
Service appears in List



Employee Enters Description and then clicks save

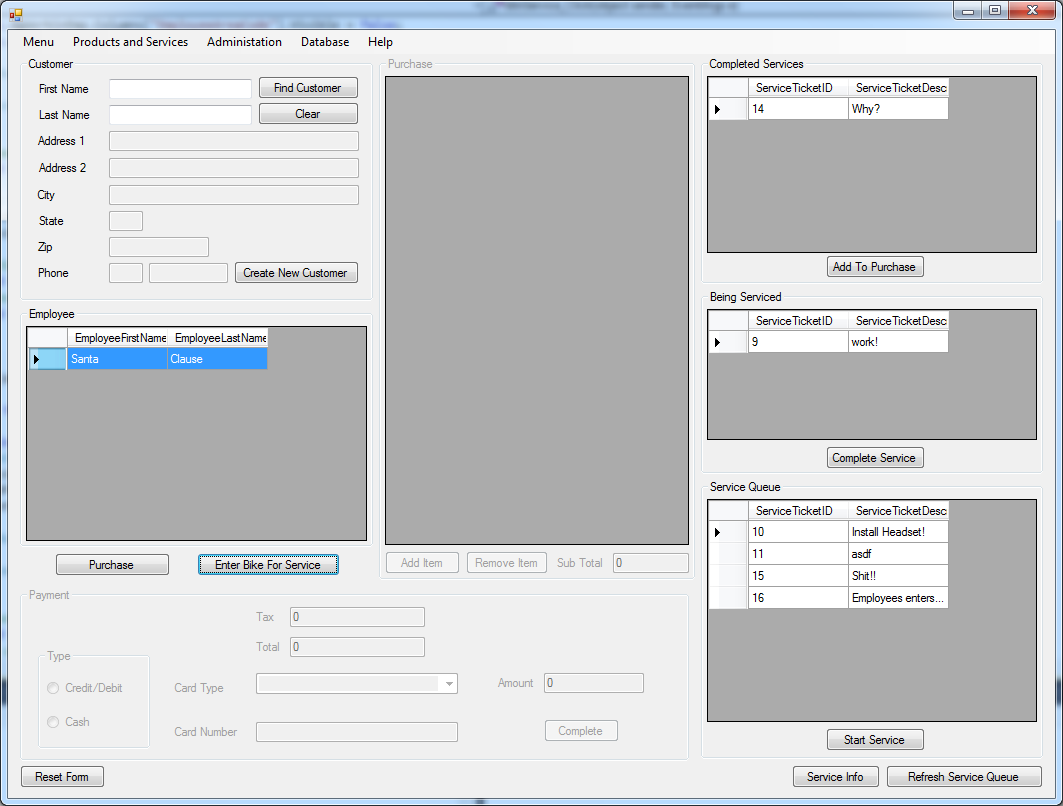


Message saying ticket created successfully and then gives the ticket id

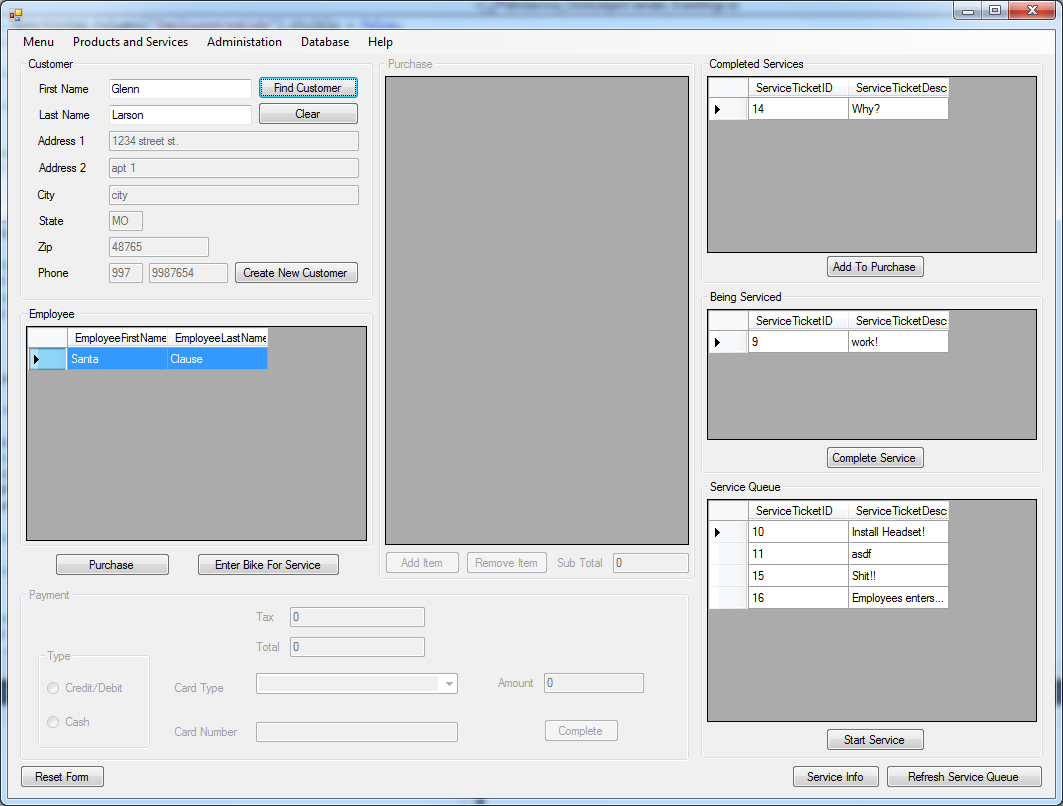


Service ticket appears in service queue

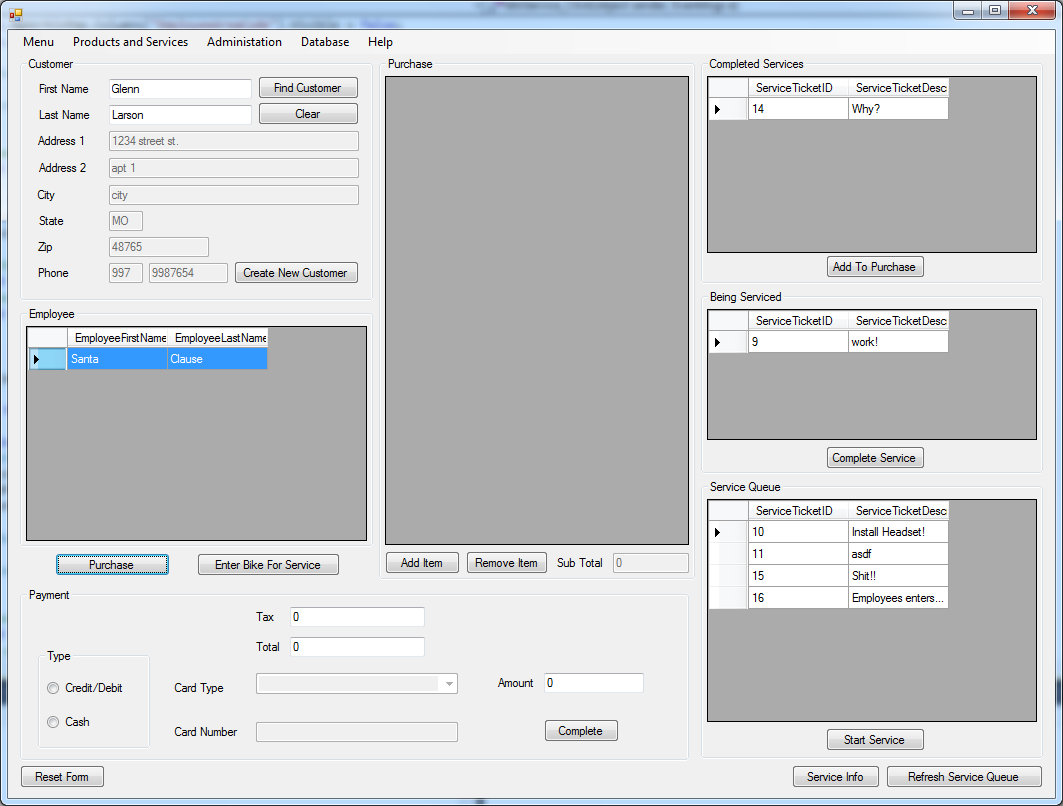
Enter Purchase StoryBoard



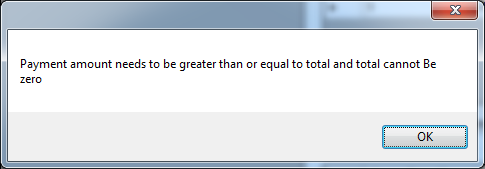
Application is already running and employee clocked in



Employee finds customer information then selects their name from working list and clicks purchase

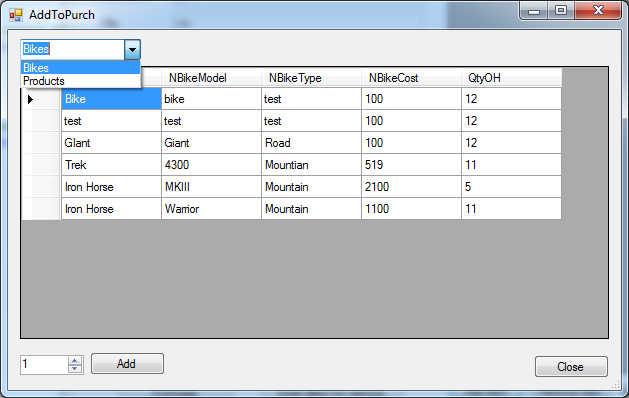


Purchase group box and payment group box become active. Employee tries to complete purchase without adding any items

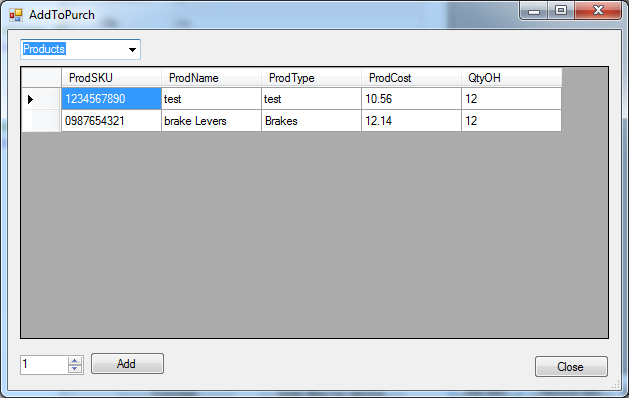


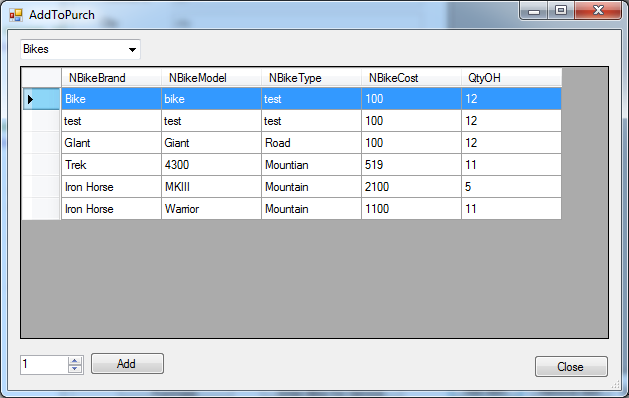
Message for not having any items added to purchase

Employee clicks add item button.

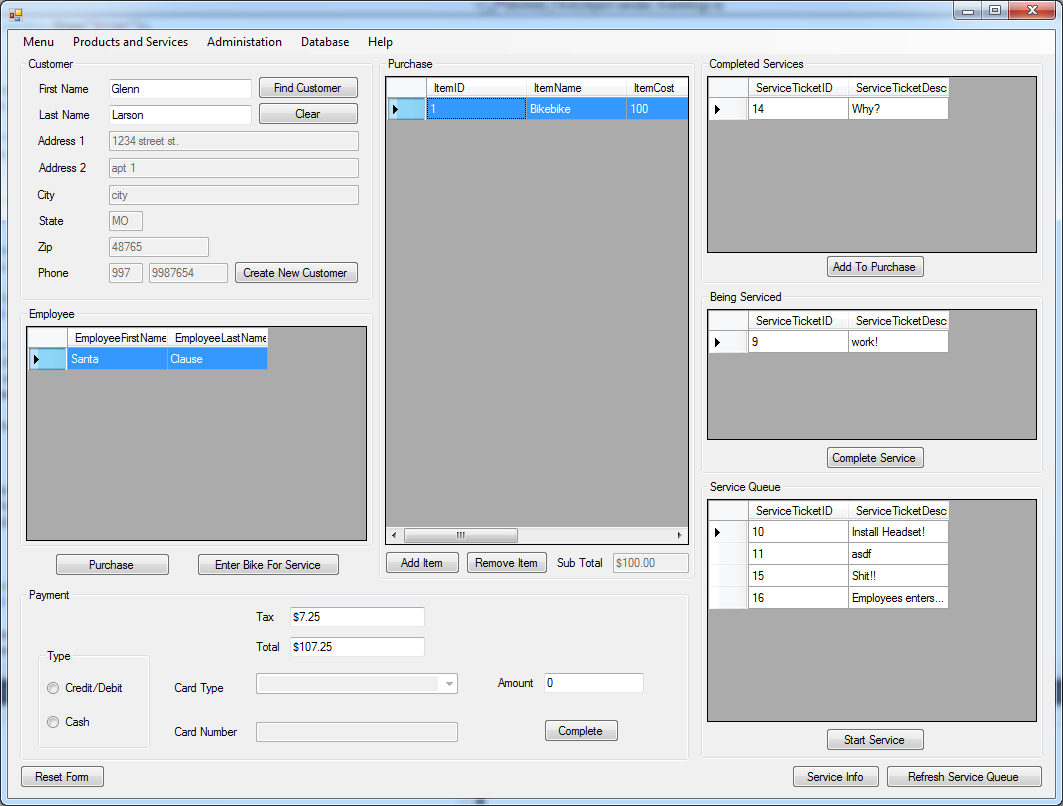


Window with items appears. There is a combo box at the top that can be used to switch between Bikes and Products.

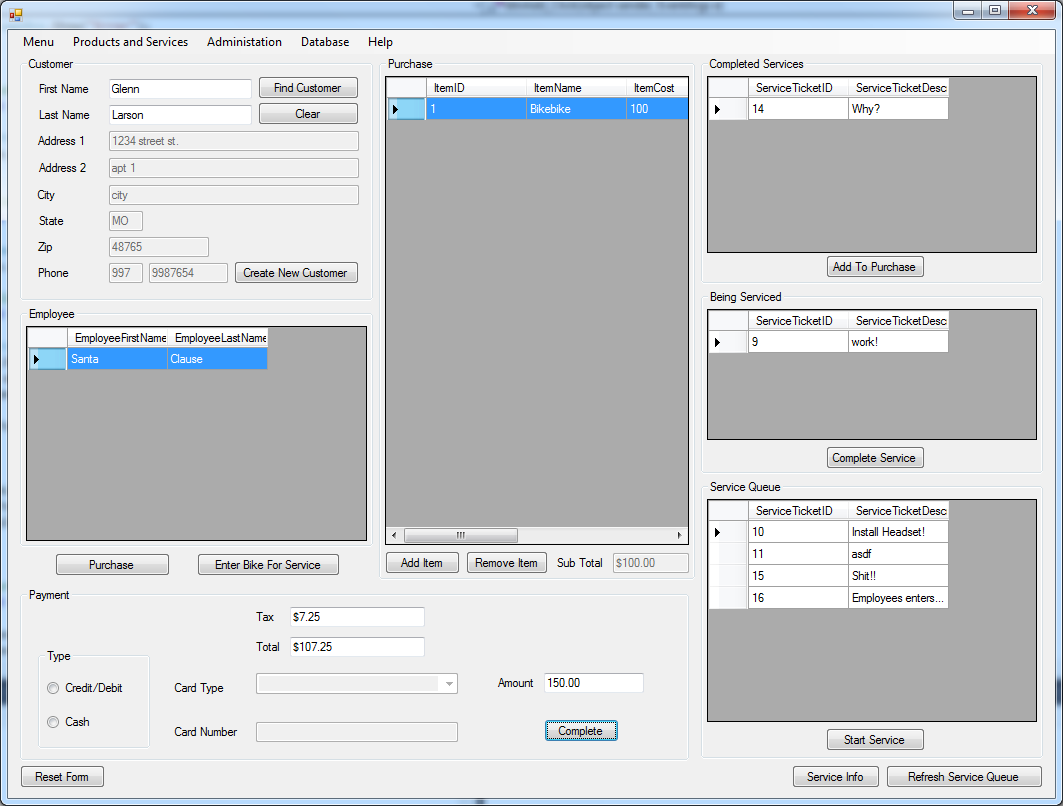




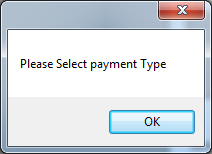
Employee then selects qty of 1 and hits add



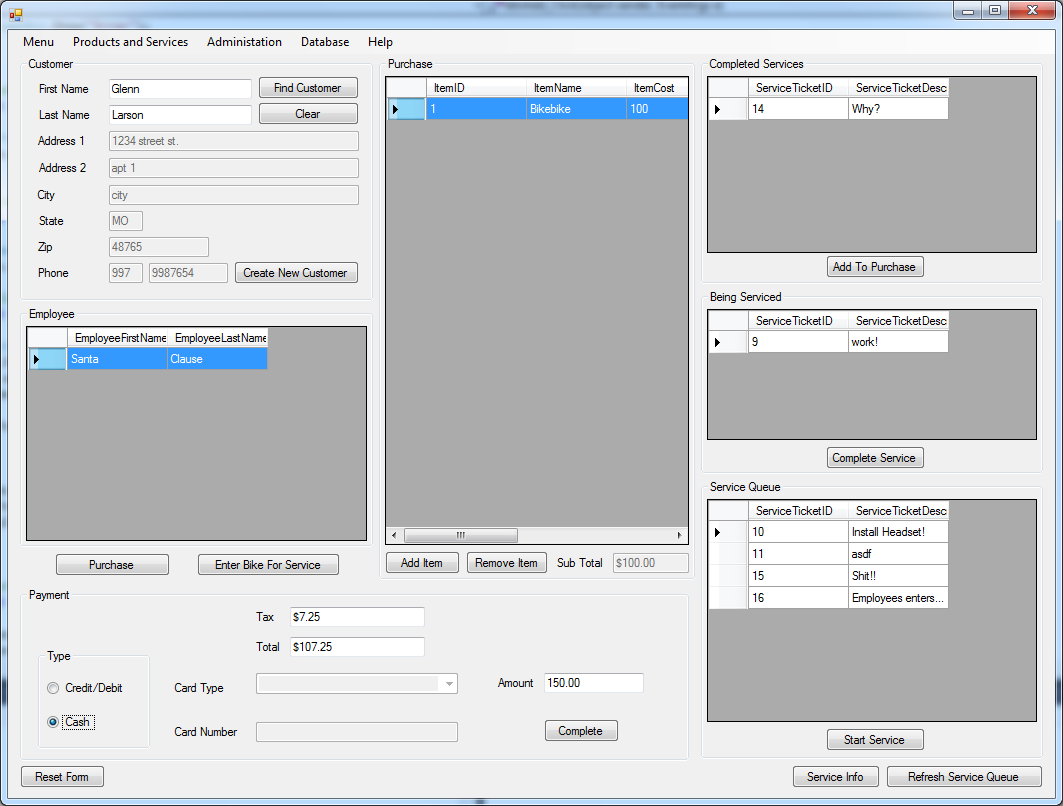
Items Appears In purchase list. Subtotal Gets filled in with subtotal of list and then tax gets calculated and then the total get calculated.



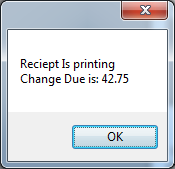
Employee Enters Payment amount but doesn’t select payment type

****

Message telling employee to select payment type.

****

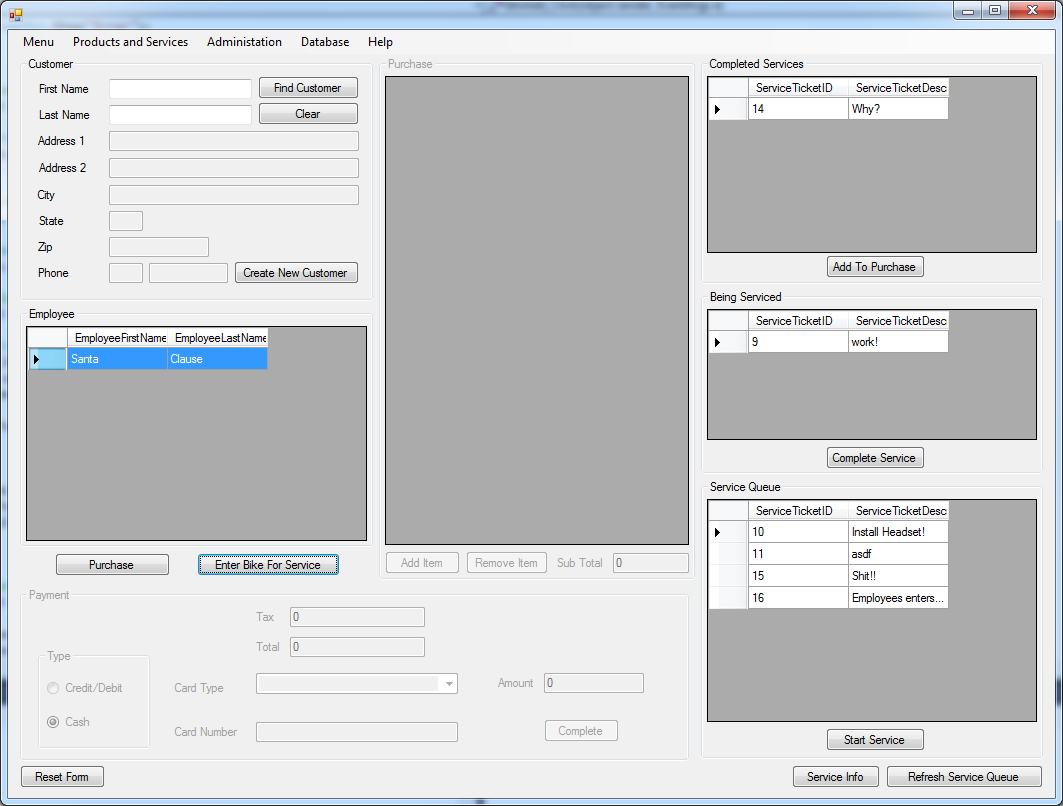
Employee chooses payment type of cash. Then Clicks Complete Button

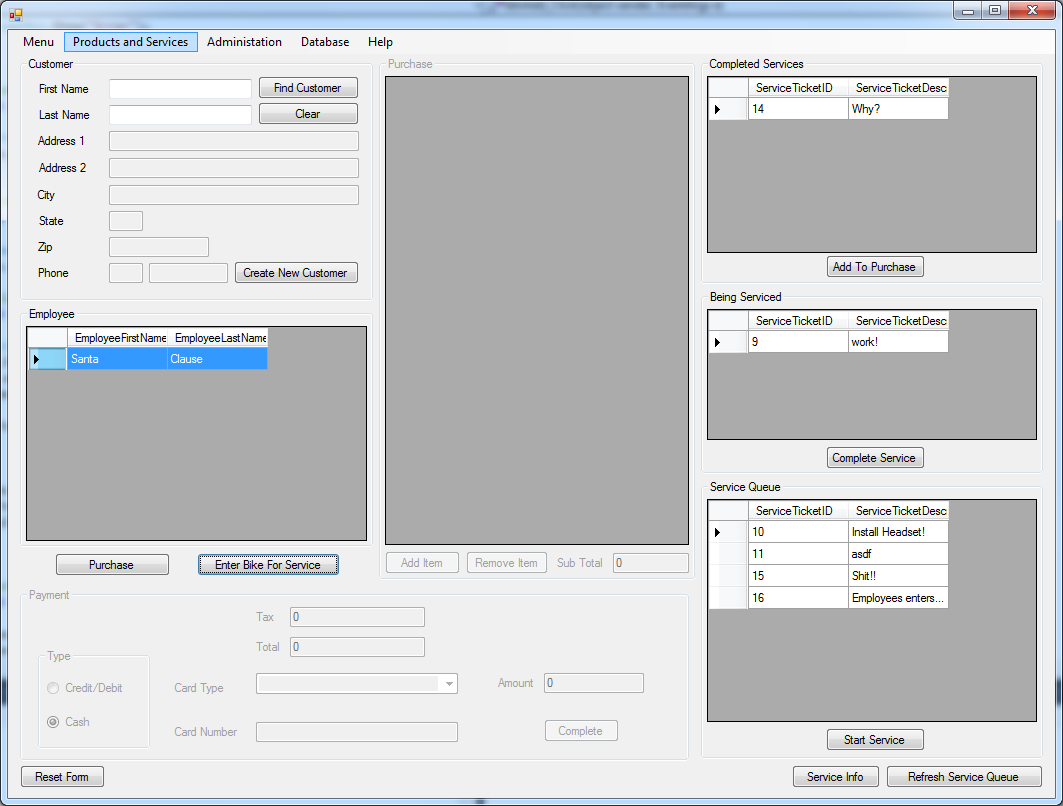


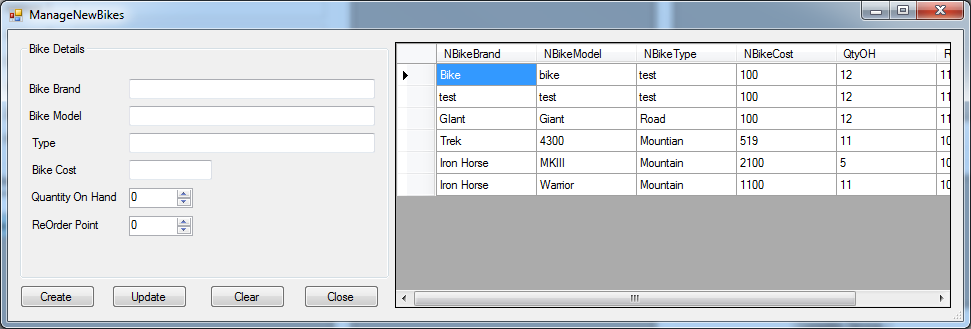
Message Saying receipt is printing and the amount of change that is due

Enter New Bike Story Board

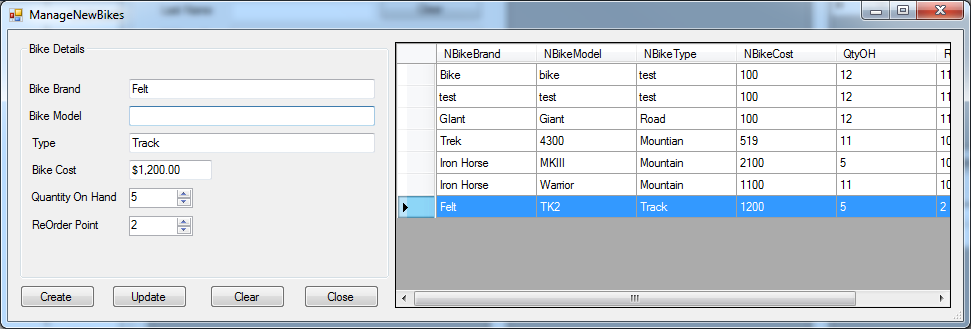
Add new Bike use case



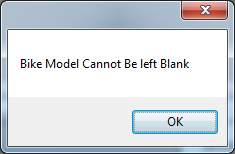


Employee Selects Products and services menu then selects manage New bikes(I tried to get picture with menu open but it wouldn’t stay open) 

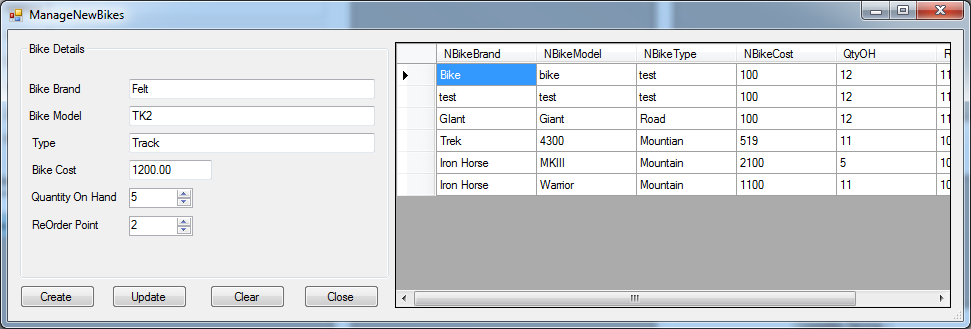
Manage bike windows appears



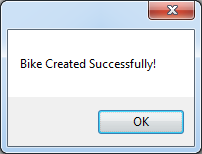
Employee Bike Information But Leaves Model Empty then hits create



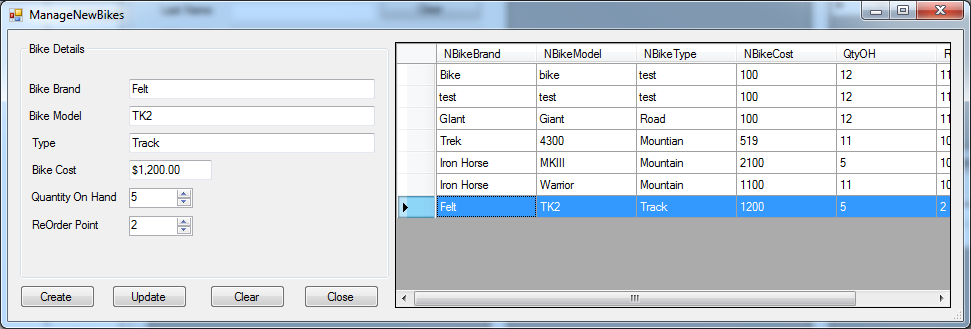
Message for when Bike Model is left blank



Employee Fixes Bike Model and clicks create button



Message saying bike was created successfully



Bike List Reloads and the new bike is visible in bike List

Incomplete Components

The Only Requirement that I havn’t completed is having 2 reports. I only have 1 report. I didn’t even try to implement a second report, because I am just ready to be done with this project.

Installation Evaluators

For installing the program there doesn’t need to be anything special installed. The Exe File and the database need to be located in the same directory.

|  |  |
| --- | --- |
| 1 | Test |
| 2 | pass |
| 3 | pass |
| 4 | pass |
| 9 | pass |
| 14 | pass |