|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Name: HestiService system** | | | | | |
| **Author:** Anke Brits | **Date:** 21 July 2024 | | | **Version:** 1.0.0 | |
|  | | | | | |
| **Use Case Name:** | Create parts quote | | **Use Case Type** | | |
| **Use Case ID:** | 3.34 | | Business Requirements: ◻ | | |
| **Priority:** | High | | System Analysis: ◻ | | |
| **Source:** | Client study (Hestico) | | System Design: ☒ | | |
| **Primary Business Actor (PBA):** | Admin | | | | |
| **Primary System Actor (PSA):** | None | | | | |
| **Other Participating Actors:** | None | | | | |
| **Other Interested Stakeholders:** | None | | | | |
| **Description:** | This use case describes the process of creating a parts quote.  The admin navigates to a specific Work Order and chooses to view the parts associated with the work order. The admin selects the generate quote button and the system generates the work | | | | |
| **Pre-condition:** | * The admin must be logged in. | | | | |
| **Trigger:** | * The admin wants to generate a preliminary quote. The admin clicks the “Work Order” tab on the navigation bar. | | | | |
| **Typical Course**  **of Events:** | **Actor Action** | **System Response** | | | |
| **Manual Action** | | | **Automated Action** |
| Step 1: The admin wants to generate a preliminary quote. The admin clicks the “Work Order” tab on the navigation bar. |  | | | Step 2: The system loads the “View Work Orders” screen that contains the following elements  A heading with the text “Work Orders” top of the screen to the left.  An input textbox for the admin to input the information they want to search for below the “Work Orders” heading to the left.   * A placeholder within the input textbox with the label “Search..”   A card for each work order that contain the following information:   * Work Order ID * Service Request ID * Machine Type * Status * Employee * Reason * Date Started * Date Completed * View Updates button * View Parts button   The attributes Work Order ID, Service Request ID, Machine Type, Status, Employee, Reason, Date Started, and Date Completed will be displayed in the attribute’s respective places in the card.  The system will send a request form the Angular frontend to the Work Order service where the service will make a http get request to the .NET Core backend which makes use of a Lambda LINQ Query which creates a SQL Select query to retrieve the work order from the Work Order Entity and the corresponding information from tables that are referenced by the foreign keys. The tables referenced by the foreign keys are described below.  The system displays the work order details by using Entity Framework Core to retrieve only the related data in the Work\_Order table with the following attributes:   * Work\_Order\_Id (PK) * Service\_Request\_Id (FK) * Machine\_Type\_Id (FK) * Work\_Order\_Status\_Id (FK) * Employee\_Id (FK) * Reason * Date\_Started * Date\_Completed * Invoice\_Id (FK)   The system links the Work\_Order table to the Service\_Request table using the foreign key Service\_Request\_Id. The Service\_Request table has the following attributes:   * Service\_Request\_Id (PK) * Service\_Request\_Status\_Id (FK) * Service\_Type\_Id (FK) * Technician\_Id (FK) * Machine\_Type\_Id (FK) * Representative\_Id (FK) * Preferred\_Date * Alternative\_Date\_1 * Alternative\_Date\_2 * Alternative\_Date\_3 * Problem\_Description * Finalised\_Date * Work\_Order\_Id (FK)   The system links the Work\_Order table to the Machine\_Type table using the foreign key Machine\_Type\_Id. The Machine\_Type table has the following attributes:   * Machine\_Type\_Id (PK) * Name * Description * Serial\_Number   The system links the Work\_Order table to the Work\_Order\_Status table using the foreign key Work\_Order\_Status\_Id. The Work\_Order\_Status table has the following attributes:   * Work\_Order\_Status\_Id (PK) * Name * Description   The system links the Work\_Order table to the Employee table using the foreign key Employee\_Id. The Employee table has the following attributes:   * Employee\_Id (PK) * Employee\_Type\_Id (FK) * UserId * Name * Surname * Address * Phone\_No * Gender * Race * Preferred\_Name   The system links the Work\_Order table to the Invoice table using the foreign key Invoice\_Id. The Invoice table has the following attributes:   * Invoice\_Id (PK) * Customer\_Reference\_Number * Discount\_Id (FK) * Work\_Order\_Id (FK) * Quote\_Id (FK) * Invoice\_Status\_Id (FK) * Date\_Issued * Date\_Paid * Total\_Amount * Description   If there is no Invoice associated with the Work Order that is displayed, then the Invoice\_Id is not displayed.  If the Date\_Completed value is set to Null, then the Date\_Completed will not be displayed on the card.  [ALT] |
| Step 3: The technician enters the customer’s name or employee name of the Work Order they want to search for. |  | | | Step 4: The system searches for the Work Orders in the database from the Work Orders table using ASP.Net 7 Web API controller by using a LINQ query and displays the records that match the inputted information in the Work Order cards on the View Work Order screen.  [ALT] |
| Step 5: The technician clicks the “View Parts” button of the specific Work Order. |  | | | Step 6: The system loads the “View Parts” screen that contains the following elements:  A heading with the text “Inventory Work Orders for Work Order ID” at the top of the screen.  A table displaying the inventory work orders with the following columns:   * Inventory ID * Inventory Name * Description * Price * Quantity * Status * Action   If the InventoryWorkOrder doesn’t have a quote associated with it then a “Generate Parts Quote” button is displayed in the Action column.  The system displays the inventory work orders by using Entity Framework Core to retrieve only the related data in the InventoryWorkOrder table with the following attributes:   * InventoryWorkOrder\_Id (PK) * Work\_Order\_Id (FK) * Inventory\_Id (FK) * Quantity * Inventory\_Work\_Order\_Status\_Id (FK)   The system links the InventoryWorkOrder table to the Work\_Order table using the foreign key Work\_Order\_Id. The Work\_Order table has the following attributes:   * Work\_Order\_Id (PK) * Service\_Request\_Id (FK) * Machine\_Type\_Id (FK) * Work\_Order\_Status\_Id (FK) * Employee\_Id (FK) * Reason * Date\_Started * Date\_Completed * Invoice\_Id (FK)   The system links the InventoryWorkOrder table to the Inventory table using the foreign key Inventory\_Id. The Inventory table has the following attributes:   * Inventory\_Id (PK) * Inventory\_Status\_Id (FK) * Name * Inventory\_Description * Price * Quantity   The system links the InventoryWorkOrder table to the InventoryWorkOrderStatus table using the foreign key Inventory\_Work\_Order\_Status\_Id. The InventoryWorkOrderStatus table has the following attributes:   * Inventory\_Work\_Order\_Status\_Id (PK) * Name * Description   [ALT] |
|  | Step 7: The admin clicks the “Generate Parts Quote” button. |  | | | Step 8: The system calculates the quote fee as the inventory price multiplied with the quantity used or requested.  The system uses Entity Framework Core to save the entered information in the Quote table with the following ibutes:   * Quote\_Id (PK) * Customer\_Id (FK) * Service\_Type\_Id (FK) * Machine\_Type\_Id (FK) * Employee\_Id (FK) * Quote\_Fee * Date * Description * Quote\_Status\_Id (FK) * Quote\_Type\_Id (FK)   The system links the Quote table to the Customer table using the foreign key Customer\_Id. The Customer table has the following attributes:   * Customer\_Id (PK) * Company\_Name * Email * Phone\_Number   The system links the Quote table to the Service\_Type table using the foreign key Service\_Type\_Id. The Service\_Type table has the following attributes:   * Service\_Type\_Id (PK) * Name * Description   The system links the Quote table to the Machine\_Type table using the foreign key Machine\_Type\_Id. The Machine\_Type table has the following attributes:   * Machine\_Type\_Id (PK) * Name * Description   The system links the Quote table to the Employee table using the foreign key Employee\_Id. The Employee table has the following attributes:   * Employee\_Id (PK) * Name * Surname * Phone\_Number * Email   The system links the Quote table to the Quote\_Status table using the foreign key Quote\_Status\_Id. The Quote\_Status table has the following attributes:   * Quote\_Status\_Id (PK) * Name * Description   The system links the Quote table to the Quote\_Type table using the foreign key Quote\_Type\_Id. The Quote\_Type table has the following attributes:   * Quote\_Type\_Id (PK) * Name * Description   The system logs the following when quote is viewed:   * user performing the operation * Transaction Type: * The description which contains the quote id.   In the following Audit\_Trail entity has the following attributes:   * Audit\_Trail\_Id (PK) * Date\_Time * User\_Name * Transaction\_Type * Description. * The Autdit\_trail\_Id is automatically incremented. |
|  |  |  | | | Step 9: The system redirects the admin to the “Quote List” screen which contains the following elements:  A heading with the text “Quotes” at the top of the screen.  A “Generate Preliminary Quote” button.  A table displaying the quotes with the following columns:   * Quote ID * Customer Name * Service Type * Machine Type * Employee Name * Quote Fee * Date * Description * Status * Type * Actions   If there is no invoice associated with the quote and the quote is not a preliminary quote, then a button called “Generate Invoice” will be displayed.  The system will send a request form the Angular frontend to the Quote service where the service will make a http get request to the .NET Core backend which makes use of a Lambda LINQ Query which creates a SQL Select query to retrieve the quote from the Quote Entity and the corresponding information from tables that are referenced by the foreign keys. The tables referenced by the foreign keys are described below.  The system displays the quotes by using Entity Framework Core to retrieve only the related data in the Quote table with the following attributes:   * Quote\_Id (PK) * Customer\_Id (FK) * Service\_Type\_Id (FK) * Machine\_Type\_Id (FK) * Employee\_Id (FK) * Quote\_Fee * Date * Description * Quote\_Status\_Id (FK) * Quote\_Type\_Id (FK)   The system links the Quote table to the Customer table using the foreign key Customer\_Id. The Customer table has the following attributes:   * Customer\_Id (PK) * Company\_Name * Email * Phone\_Number   The system links the Quote table to the Service\_Type table using the foreign key Service\_Type\_Id. The Service\_Type table has the following attributes:   * Service\_Type\_Id (PK) * Name * Description   The system links the Quote table to the Machine\_Type table using the foreign key Machine\_Type\_Id. The Machine\_Type table has the following attributes:   * Machine\_Type\_Id (PK) * Name * Description   The system links the Quote table to the Employee table using the foreign key Employee\_Id. The Employee table has the following attributes:   * Employee\_Id (PK) * Name * Surname * Phone\_Number * Email   The system links the Quote table to the Quote\_Status table using the foreign key Quote\_Status\_Id. The Quote\_Status table has the following attributes:   * Quote\_Status\_Id (PK) * Name * Description   The system links the Quote table to the Quote\_Type table using the foreign key Quote\_Type\_Id. The Quote\_Type table has the following attributes:   * Quote\_Type\_Id (PK) * Name * Description   The system logs the following when quote is viewed:   * user performing the operation * Transaction Type: * The description which contains the Customer\_Id.   In the following Audit\_Trail entity has the following attributes:   * Audit\_Trail\_Id (PK) * Date\_Time * User\_Name * Transaction\_Type * Description.   The Autdit\_trail\_Id is automatically incremented. |
|  |  |  | | |  |
|  |  |  | | |  |
| **Alternate Courses:** |  | | | | |
| [ALT] Step 2: There are no records in the Work Order table from the database. The system will display an error message with the text “No items were found” | | | | |
| [ALT] Step 2:  There is an error in retrieving the information from the Work Order table in the database. The system displays an error message. A label with the text "There is an error with the system, please try Again Later”. | | | | |
| [ALT] Step 4: There is no records in the Work Order table from the database that matches the input search criteria. The system will display a notification to state “Data not found”. Admin clears search criteria, go to step 2. | | | | |
| [ALT] Step 6a: There are no records in the Inventory Work Order table from the database. The system will display an error message with the text “No items were found”. | | | | |
| [ALT] Step 6b: There is an error in retrieving the information from the Inventory Work Order table in the database. The system displays an error message. A label with the text "There is an error with the system, please try Again Later”. | | | | |
| [ALT] Step 9a: There are no records in the Quote table from the database. The system will display an error message with the text “No items were found” | | | | |
| [ALT] Step 9b: There is an error in retrieving the information from the Quote table in the database. The system displays an error message. A label with the text "There is an error with the system, please try Again Later”. | | | | |
| **Conclusion:** | The admin successfully creates a parts quote. | | | | |
| **Post-condition:** | The new quote is successfully saved to the database. | | | | |
| **Business Rues:** | * Only the admin can create parts quotes | | | | |
| **Implementation Constraints and Specifications:** | * None | | | | |
| **Assumptions:** | * None | | | | |
| **Open Issues:** | * None | | | | |