|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Name: HestiService system** | | | | | |
| **Author:** Anke Brits | **Date:** 21 July 2024 | | | **Version:** 1.0.0 | |
|  | | | | | |
| **Use Case Name:** | Generate customer invoice | | **Use Case Type** | | |
| **Use Case ID:** | 4.7 | | 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 generating a customer invoice.  The admin selects the quote that they would like to generate an invoice for. The system takes the relevant information from the quote and assigns them to the values in the new invoice and issued the invoice. The admin is then redirected to the Invoices screen to view the newly created invoice.  The use case concludes when the invoice has been successfully generated. | | | | |
| **Pre-condition:** | * The admin must be logged in * There must be a quote to generate an invoice from | | | | |
| **Trigger:** | * The admin wants to generate an invoice for a specific quote. The admin clicks the “Quotes” tab on the navigation bar. | | | | |
| **Typical Course**  **of Events:** | **Actor Action** | **System Response** | | | |
| **Manual Action** | | | **Automated Action** |
| Step 1: The admin wants to generate an invoice for a specific quote. The admin clicks the “Quotes” tab on the navigation bar. |  | | | Step 2: 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.  [ALT] |
| Step 3: The admin clicks the “Generate Invoice” button.  [ALT] |  | | | Step 4: 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 Insert query to create the invoice from the Invoice 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 retrieves the selected quote's information and creates a new invoice instance using Entity Framework Core and saves it in the Invoice table with the following attributes:   * Invoice\_Id (PK) * Customer\_Id (FK) * Quote\_Id (FK) * Invoice\_Status\_Id (FK) * Date\_Issued * Date\_Paid * Total\_Amount * Customer\_Reference\_Number * Discount\_Id (FK) * Description   The system links the Invoice 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 Invoice table to the Quote table using the foreign key Quote\_Id. The Quote table has 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 Invoice table to the Invoice\_Status table using the foreign key Invoice\_Status\_Id. The Invoice\_Status table has the following attributes:   * Invoice\_Status\_Id (PK) * Name * Description   The system sets the Invoice\_Status\_Id to 2 (“Unpaid”).  [ALT] |
|  |  | | | Step 5: After generating the invoice, the system redirects the admin to the "Invoices" screen. This screen contains the following elements:  A heading with the text “Invoices” at the top of the screen.  A table displaying the invoices with the following columns:   * Invoice ID * Customer Name * Quote ID * Status * Date Issued * Date Paid * Total Amount * Customer Reference Number * Discount Percentage * Description * Actions   If there is a service rating associated with the Invoice, then a “View Customer Rating” button will be displayed.  The system will send a request form the Angular frontend to the Invoices 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 Insert query to create the invoice from the Invoice 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 invoices by using Entity Framework Core to retrieve only the related data in the Invoice table with the following attributes:   * Invoice\_Id (PK) * Customer\_Id (FK) * Quote\_Id (FK) * Invoice\_Status\_Id (FK) * Date\_Issued * Date\_Paid * Total\_Amount * Customer\_Reference\_Number * Discount\_Id (FK) * Description   The system links the Invoice 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 Invoice table to the Quote table using the foreign key Quote\_Id. The Quote table has 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 Invoice table to the Invoice\_Status table using the foreign key Invoice\_Status\_Id. The Invoice\_Status table has the following attributes:   * Invoice\_Status\_Id (PK) * Name * Description   The system links the Invoice table to the Discount table using the foreign key Discount\_Id. The Discount table has the following attributes:   * Discount\_Id (PK) * Percentage * Description   The system logs the following when invoice is generated:   * 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 Audit\_trail\_Id is automatically incremented. |
| **Alternate Courses:** | [ALT] Step 2: There are no records in the Quotes table from the database. The system will display an error message with the text “No items were found” | | | | |
| [ALT] Step 3: There are no record in the quote table that don’t have an invoice associated with them. The use case terminates. | | | | |
|  | [ALT] Step 4: The system fails to add the new invoice to the database. The system will display a notification. This use case terminates. | | | | |
| **Conclusion:** | The new invoice is displayed on the “Invoices” screen and the Invoices screen is updated. | | | | |
| **Post-condition:** | A new invoice has been added to the database. | | | | |
| **Business Rues:** | * Only the admin can generate an invoice | | | | |
| **Implementation Constraints and Specifications:** | * None | | | | |
| **Assumptions:** | * None | | | | |
| **Open Issues:** | * None | | | | |