|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Name: HestiService system** | | | | | |
| **Author:** Anke Brits HELLO | **Date:** 21 July 2024 | | | **Version:** 1.0.0 | |
|  | | | | | |
| **Use Case Name:** | View quote | | **Use Case Type** | | |
| **Use Case ID:** | 2.15 | | Business Requirements: ◻ | | |
| **Priority:** | High | | System Analysis: ◻ | | |
| **Source:** | Client study (Hestico) | | System Design: ☒ | | |
| **Primary Business Actor (PBA):** | Customer | | | | |
| **Primary System Actor (PSA):** | None | | | | |
| **Other Participating Actors:** | None | | | | |
| **Other Interested Stakeholders:** | Admin | | | | |
| **Description:** | This use case describes the process of the customer viewing a quote.  The customer will be able to view their quotes. The customer will click on the Quotes link on the navigation bar and the system will redirect the customer to the “View Customer Quotes” screen.  The use case ends when the quote has been viewed. | | | | |
| **Pre-condition:** | * The customer must be logged in. * The quote must exist to be viewed. | | | | |
| **Trigger:** | * The customer wants to view their quotes. The customer clicks the “Quotes” tab on the Navigation bar. | | | | |
| **Typical Course**  **of Events:** | **Actor Action** | **System Response** | | | |
| **Manual Action** | | | **Automated Action** |
| Step 1: The customer wants to view their quotes. The customer clicks the “Quotes” tab on the Navigation bar. |  | | | Step 2: The system displays the “View Customer Quotes” screen which contains the following elements:  A heading with the text "Quotes" at the top of the screen.  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 (buttons to approve, decline, or request service based on the status and type of the quote)   If the quote is a preliminary quote and the quote status is “Approved”, then a “Request Service” button will be displayed in the Actions column in the table in the row of the relevant quote.  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 * Account\_No * Date\_of\_Account\_Request * Cell\_Phone\_No * Telephone\_No * Work\_No * Company\_Email\_Address * User\_ID (FK) * Flag\_ID (FK)   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 * Serial\_Number   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) * Employee\_Type\_Id (FK) * UserId * Name * Surname * Address * Phone\_No * Gender * Race * Preferred\_Name   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 customer enters the quote type or employee name of the Quote they want to search for. |  | | | Step 4: The system searches for the Quotes 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 Quotes table on the View Customer Quotes screen.  [ALT] |
| Step 5: The customer successfully views the quotes. |  | | |  |
| **Alternate Courses:** | [ALT] Step 2a: 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 2b:  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”. | | | | |
| [ALT] Step 4: There is no records in the Quote 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 3. | | | | |
| **Conclusion:** | The customer successfully views their quotes. | | | | |
| **Post-condition:** | The quotes were successfully retrieved from the database. | | | | |
| **Business Rues:** | * None | | | | |
| **Implementation Constraints and Specifications:** | * None | | | | |
| **Assumptions:** | * None | | | | |
| **Open Issues:** | * None | | | | |