|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Name: HestiService system** | | | | | |
| **Author:** Anke Brits | **Date:** 21 July 2024 | | | **Version:** 1.0.0 | |
|  | | | | | |
| **Use Case Name:** | Accept parts quote | | **Use Case Type** | | |
| **Use Case ID:** | 2.16 | | 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 accepting a parts quote.  The customer wants to accept a parts quote. The system invokes use case 2.15 View Quote. The customer selects the accept quote button and the quote status is changed to accepted. The system updates the quote in the database and generates an invoice.  The use case concludes when the parts quote is accepted, and the new invoice is generated. | | | | |
| **Pre-condition:** | * The customer must be logged in. | | | | |
| **Typical Course**  **of Events:** | **Actor Action** | **System Response** | | | |
| **Manual Action** | | | **Automated Action** |
| Step 1: The customer wants to accept a parts quote. The customer clicks the “Quotes” tab in the navigation bar. |  | | | Step 2: The system invokes use case 2.15 View Quote. |
|  |  | | | Step 3: The system successfully displays the quotes that the customer was searching for. |
| Step 4: The customer selects the “Accept” button of the quote they would like to accept. |  | | | Step 5: 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 Update query to retrieve the quote from the Quote Entity and the corresponding information from tables that are referenced by the foreign keys.  The system updates the status of the quote to "Approved” (Quote\_Status\_Id = 2) using Entity Framework Core and saves the updated status 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 |
|  |  | | | Step 6: 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 Create query to create the Invoice in 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 links the Quote 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   The system links the Invoice table to the Customer table using the foreign key Customer\_Reference\_Number. The Customer table has the following attributes:   * Customer\_Reference\_Number (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 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 links the Invoice 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 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 logs the following when quote is updated:   * user performing the operation * Transaction Type: * The description which contains the quote id and the invoice id.   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. |
|  |  | | | Step 7: The system removes the buttons from the Actions column. |
| **Alternate Courses:** |  | | | | |
| **Conclusion:** | The customer has successfully accepted the parts quote. | | | | |
| **Post-condition:** | The status of the parts quote was successfully updated in the quote entity in the database. | | | | |
| **Business Rues:** | * None | | | | |
| **Implementation Constraints and Specifications:** | * None | | | | |
| **Assumptions:** | * None | | | | |
| **Open Issues:** | * None | | | | |