**Household Listing Design**

Table of Contents

**Section 1: Household Listing BRD Requirements3**

Household Listing Creation3

Household Listing Viewing3

Household Listing Editing4

**Section 2: Household Listing High Level Design5**

2.1 DAO and Model Requirements5

2.2 DAO Additional Information6

2.3 Class Diagrams (HHListingModel, HHListingDAO)6

2.4 Activity Diagram: Household Listing Creation 7

2.5 Activity Diagram: Household Listing Retrieval8

2.6 Activity Diagram: Household Listing Editing 9

2.7 Activity Diagram: Household Listing Deletion 10

**Section 3: HHListingService Design11**

3.1 Design Comments: HHListingService::Insert11

3.2 Design Comments HHListingService::Update11-12

3.3 Design Comments HHListingService::Retrieve13

3.4 Design Comments HHLIstingService::Delete14

3.5 Sequence Diagram: HHListingService::Insert15

3.6 Sequence Diagram: HHListingService::Update 16

3.7 Sequence Diagram: HHListingService::Retrieve 17

3.8 Sequence Diagram: HHListingService::Delete 18

**Section 4: HHListingController Design19**

4.1 Design Comments: HHListingController::Insert 16

4.2 Design Comments: HHListingController::Insert 16

4.3 Design Comments: HHListingController::Insert 16

4.4 Design Comments: HHListingController::Insert 16

4.5 Sequence Diagram: HHListingService::Insert14

4.6 Sequence Diagram: HHListingService::Update 14

4.7 Sequence Diagram: HHListingService::Retrieve 14

4.8 Sequence Diagram: HHListingService::Delete 14

**Section 5: Household Listing Viewing Low Level Design17**

5.1 WIP 17

5.2 WIP 17

5.3 WIP 17

5.4 WIP 17

**Section 6: Household Listing Editing Low Level Design18**

6.1 WIP 18

6.2 WIP 1

**Section 1: BRD Requirements taken from RoomAid BRD v5.0 (Descoped)**

A household listing is responsible for the additional details associated with a household for Household search purposes. These will be used for household search.

**Household Listing**

* **Making a Household Listing Available**
  + Preconditions:
    - User has created a household
    - User has selected to make their household available
  + Inputs:
    - Estimated monthly rent of available room(s)
    - A household description
      * Shall have a 2000 character limit
      * Shall allow alphanumeric and special characters
      * Shall be empty by default
    - A household type
    - Household images (optional)
  + Successful Postconditions:
    - A **household listing** is **available** and appears in household searches with the following information:
      * The household host’s name
      * A zip code retrieved from the household physical address
      * A monthly rent
      * A household type
      * A household description
        + Shall have a 2000 character limit
        + Shall allow alphanumeric and special characters
        + Shall be empty by default
      * An availability indication that is either **available** or **unavailable**.
    - A **household listing** may additionally have the following details:
      * Images of the household
* **Household Listing Viewing**: A user shall be able to view a household listing by selecting it from the list of households generated from a household search.
  + Preconditions:
    - **User** selects a household listing
  + Success Postconditions:
    - Shall display the following details to a non-tenant user:
      * The zip code
      * The estimated monthly rent of available room(s)
      * The household host’s name
      * The availability
      * The household description
      * Images of the household
  + Additional Conditions:
    - The correct details must be shown depending on the **user’s** authorizations.
  + Failure Postconditions:
    - If the household becomes **unavailable** and a **user** attempts to view the household listing through a household search, notify the **user** that the household is no longer available.

* **Household Listing Editing**: A user shall be able to edit a household listing’s information.
  + Preconditions:
    - **User** is the **host** of the household they are attempting to edit
  + Inputs:
    - A valid physical address
      * A street address with a character range of 1-200 characters
      * A city name with a character range of 1-200 characters
      * A numeric ZIP code
      * Optional: Apartment number and suite number
      * The entered address must exist in CA, USA
    - The estimated monthly rent of available room(s)
      * Shall be a nonnegative number with 2 decimal places.
    - The availability as either **available** or **unavailable**
    - The household description
      * Shall have a limit of 2000 characters
      * Shall allow alphanumeric and special characters
      * Shall be empty by default
  + Success Postconditions:
    - The following information regarding the household shall be edited in persistent storage if changes were made.
      * Physical address
    - The following information regarding the household listing shall be edited in persistent storage if changes were made:
      * Availability
      * Estimated monthly rent
      * Household Description
      * Household images
      * Household type
  + Additional condition:
    - Current values for the household listing must be provided as default input
  + Failure Postconditions:
    - If any of the above inputs are invalid, notify the user that one or more fields were invalid

**Section 2: Household Listing High Level Designs**

**2.1 DAO and Model Requirements**

**Household Listing Model** that holds the following details:

* **HID: Integer (Mandatory)**
* Household host’s First Name: String; 200 character maximum
* Zip Code: String; 5 character maximum
* StreetAddress; 200 character maximum
* Availability: SQLBoolean
* HouseholdType: String; 30 character maximum
* Price: Decimal; Non-negative, 2 decimal places
* ListingDescription: String; 2000 character maximum
* ImagePath: String[]; 260 character maximum per string
* DatePosted: DateTime in UTC (All other timestamps in application are in UTC)
  + Users are concerned with a general time period since posting rather than exact dates.

**Household Listing DAO** to communicate with the Database:

* Requires a database connection string.
* Functionality:
  + Requires reference Household Listing Model
    - **Insert**: Creates a Listing in database based on provided model details
    - **Update**: Edits a Listing in database based on provided model details
    - **Delete**: Deletes a Listing from database based on given HID
      * Note: All above functionality is meant to be initiated by a single user.
  + Retrieve information about 1 Listing in a database
    - Requires HID
    - Future scenarios may require a retrieval of multiple listings
* Must have access to the following database tables:
  + Users & Residents
    - Household Host name
  + Households
    - Zip Code
    - StreetAddress
    - IsAvailable
  + HouseholdListings
    - HouseholdType
    - Listing Description
    - Price
  + HouseholdImages
    - ImagePath

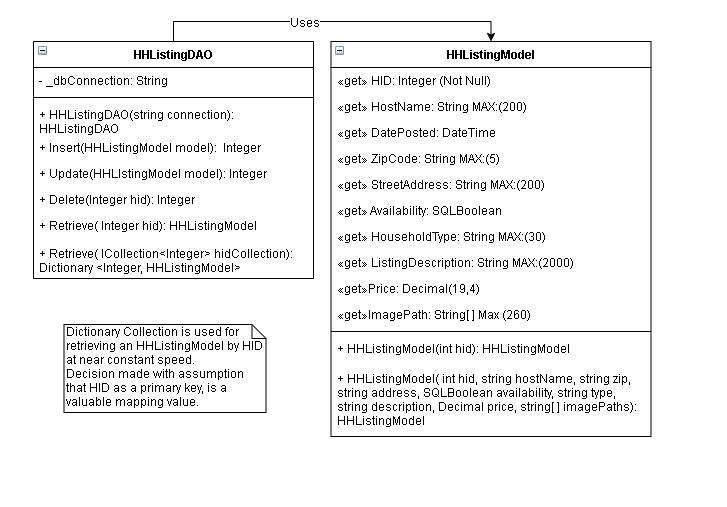
**2.2 Household Listing DAO – Additional Information**

**Return Values** shall be in the number of rows changed for error handling

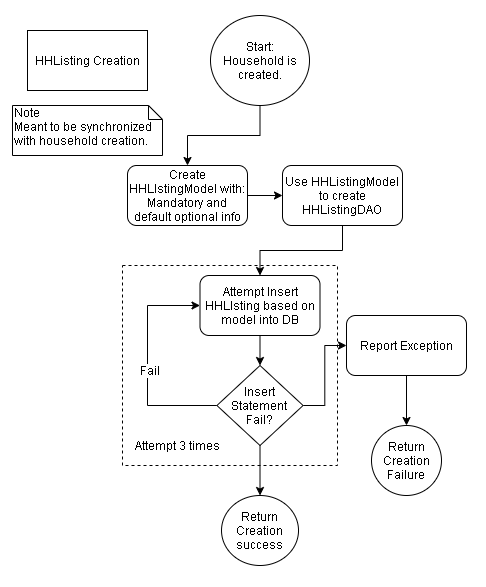
* Due to time constraints, an interface shall **not** be constructed at the current time
  + In the future, use an IDAO interface with the following required functions:
    - Insert
    - Retrieve
    - Update
    - Delete
      * Return values tentatively will be number of rows changed.

**2.3 Class Diagrams for DAO and Model**

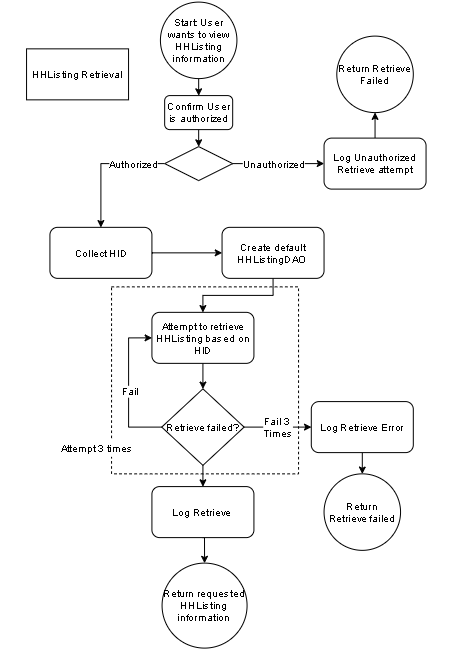
Note: 4/30/20: HHListingModel must be changed to reflect DatePosted



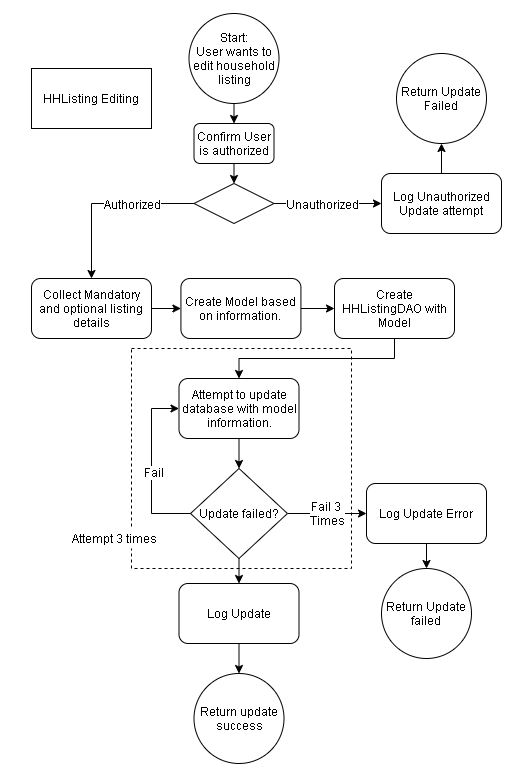
**2.4 Activity Diagram: Household Listing Creation**

****

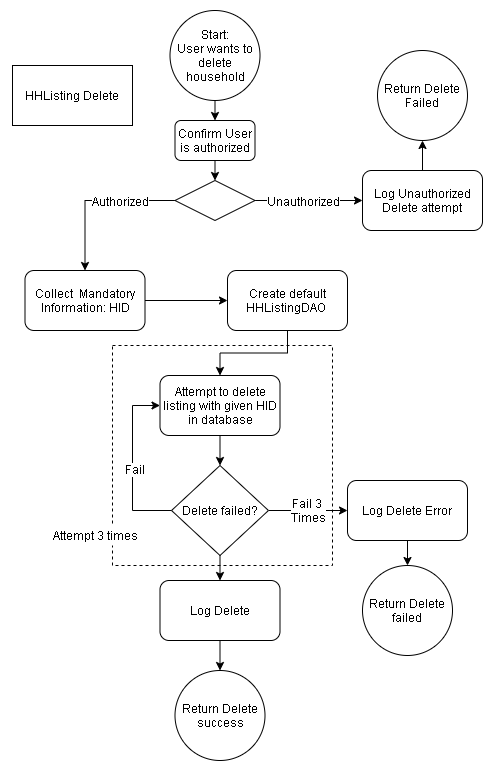
**2.5 Activity Diagram: Household Listing Retrieval**

****

**2.6 Activity Diagram: Household Listing Editing**



**2.7 Activity Diagram: Household Listing Deletion**



**Section 3: HHListingService Design**

**3.0 HHListingService Comments**

The HHListingService will be responsible for ensuring that Listing creation, edits, retrieval and deletions shall occur.

Three attempts shall be made for any command to be performed, in the case that any SQL command fails 3 times, we will return CustomExceptions with the appropriate message.

**3.1 Design Comments: HHListingService::Insert**

Last Edited: 4/29/2020

HHListingDAO::Insert uses the following model details:

* HID: Integer
* HouseholdType: String; 30 character maximum
* Price: Decimal; Non-negative, 2 decimal places
* ListingDescription: String; 2000 character maximum
* DatePosted: DateTime; In UTC

Upon creation of a household, these fields cannot be determined and thus shall be set to default values:

* Price: 0.00
* HouseholdType: “Not Specified”
* ListingDescription: “”

Inputs:

* HID: Integer
  + Used as a primary key for creating the household listing

Outputs:

* Integer
  + Represents the number of rows returned. Expected result is 1.

**3.2 Design Comments: HHListingService::Update**

Last Edited: 5/3/2020

The current edition of the BRD requires that we allow a user to edit the following details of a Household Listing:

* Inputs:
  + A valid physical address
  + The estimated monthly rent of available room(s)
  + The availability as either **available** or **unavailable**
  + The household description

However, the successful Postconditions require that the Household Type be included as part of the household editing process. As of writing, the HouseholdType is represented in the database as part of a household rather than a household listing. The designer is making a decision to add the household type as an input to allow this information to be edited.

* The following information regarding the household shall be edited in persistent storage if changes were made.
  + Physical address

**3.2 Design Comments: HHListingService::Update cont.**

* The following information regarding the household listing shall be edited in persistent storage if changes were made:
  + Availability
  + Estimated monthly rent
  + Household Description
  + Household images
  + Household type:

**Input(s):**

* **Mandatory:** HID: Integer
* **Optional**:
  + ZipCode: String
  + Address: String
  + HouseholdType: String
  + Price: Decimal
  + Availability: SQLBoolean
  + Description: String

**Output:**

* Integer: Expected value should be 1 row updated.
  + Throws CustomException otherwise.

**Additional Concernss:**

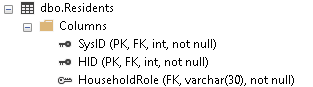
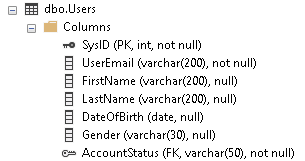
* Due to the optional nature of an update request, not all fields may be needed to update. There are two possible methods of resolving this issue.
  + Require the front end/input include all information to be included in an edit request. Such an example would include prepopulating an edit view with the current Household Listing information.
  + Using a StringBuilder to construct the query string with depending on whether optional arguments were included.
* As of writing, the 1st approach seems more feasible.
* StreetAddress is a field that is supposed to be editable, but currently not in use in HouseholdListing viewing. This is due to the initial idea of using the street address to create an obfuscated image using GoogleMap API, but implementation is not possible due to time constraints.

**3.3 Design Comments: HHListingService::Retrieve**

Last Edited: 5/3/2020

The Retrieve method is intended to be used in conjunction with the household Search function and the Household Listing edit function.

It is required to retrieve the following information about a **single** household and its listing:

* From dbo.Households:
  + zip code
  + availability
* From dbo.HouseholdListings:
  + estimated monthly rent/price
  + household descriptions
* From dbo.HouseholdImages:
  + images of the household
* From a join with dbo.Residents and dbo.Users on SysID where HID equal the selected household.
  + household host’s name (Decided to be a concatenation of First and Last Name)
  + 
  + 

**Input(s):**

* HID: Integer
  + Household ID shall be used to find the relevant information.

**Output:**

* Model: HHListingModel
  + zip code
  + availability
  + price
  + household listing description
  + household host first and last name
  + Date Posted
* Throws CustomException otherwise.

**Additional Concerns:**

* If this function is to be used for the case of editing household information, additional information will be required such as StreetAddress
* Additional permissions may require different queries or different information to be required. Such presets have not been specified yet.

**3.4 Design Comments: HHListingService::Delete**

Last edited: 5/3/2020

The delete method is intended to be used in the scenario where a **household is disbanded or deleted** and should be part of the process.

* As the functionality is unspecified, only HouseholdListings and HouseholdImages information shall be affected.

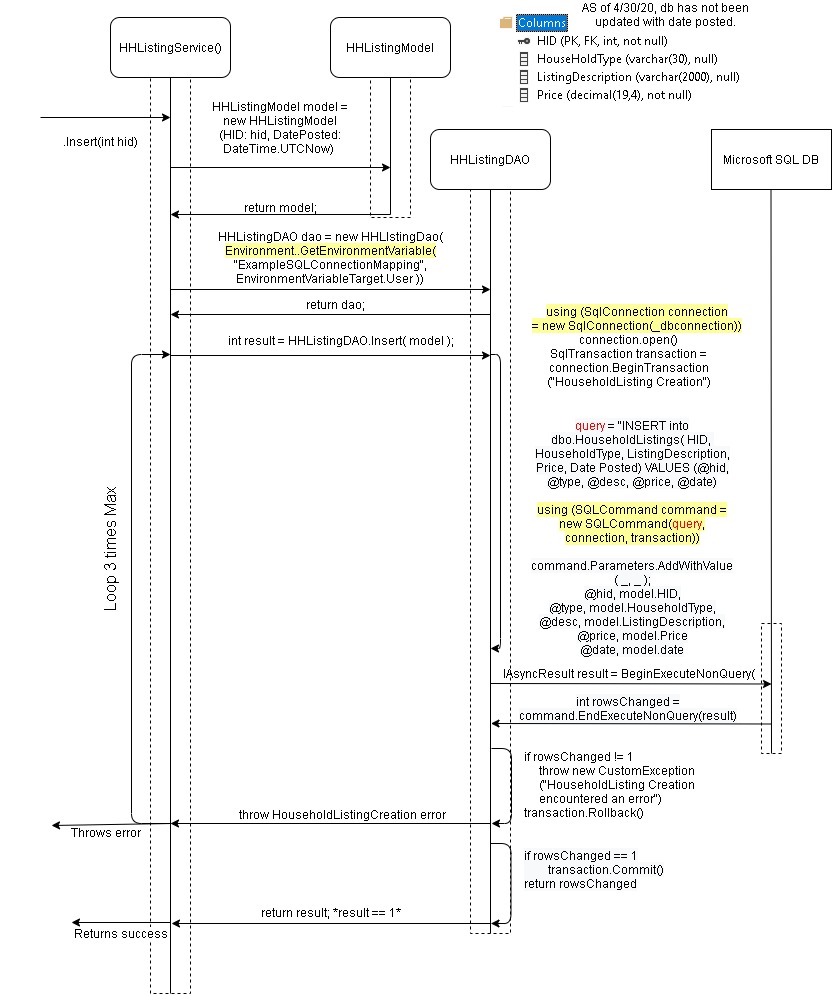
**Input(s):**

* HID: Integer

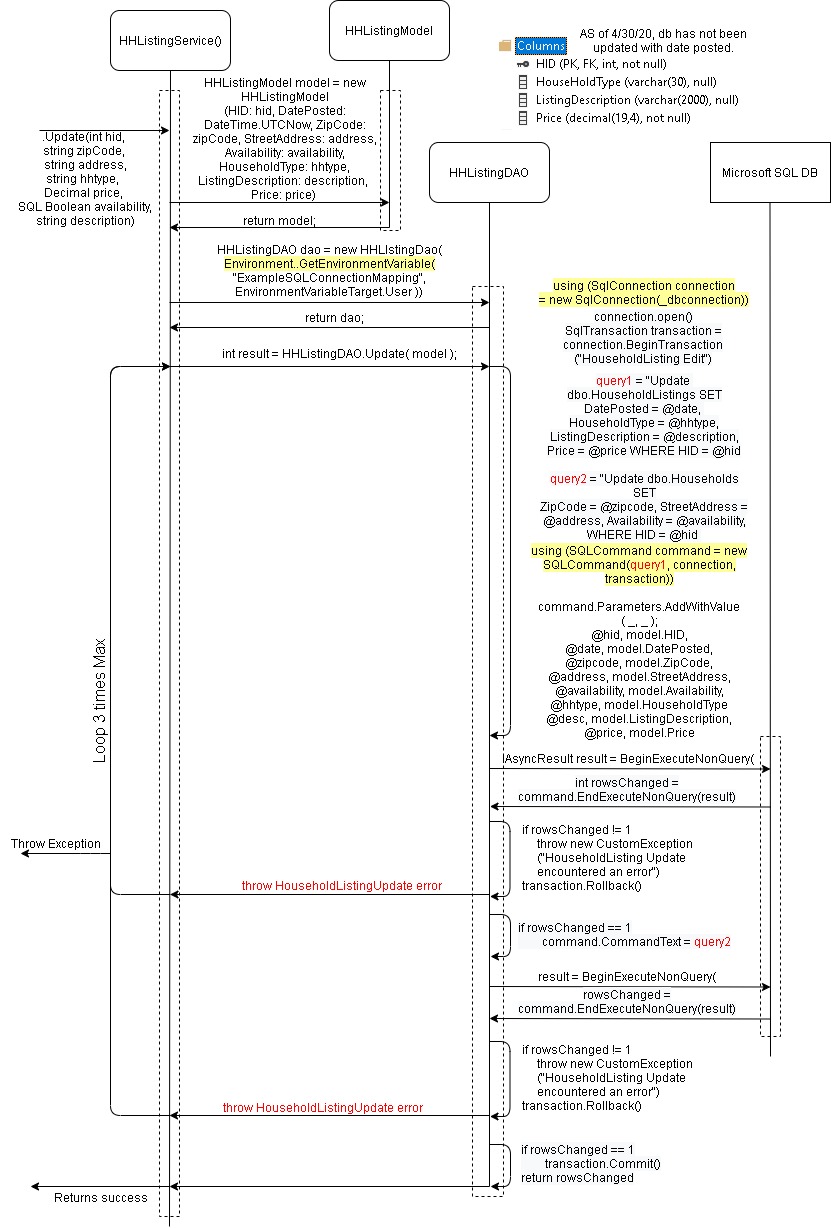
**Output:**

* Integer
  + Should return the number of rows Changed. Expected value is = 1.
* Throws CustomException otherwise.

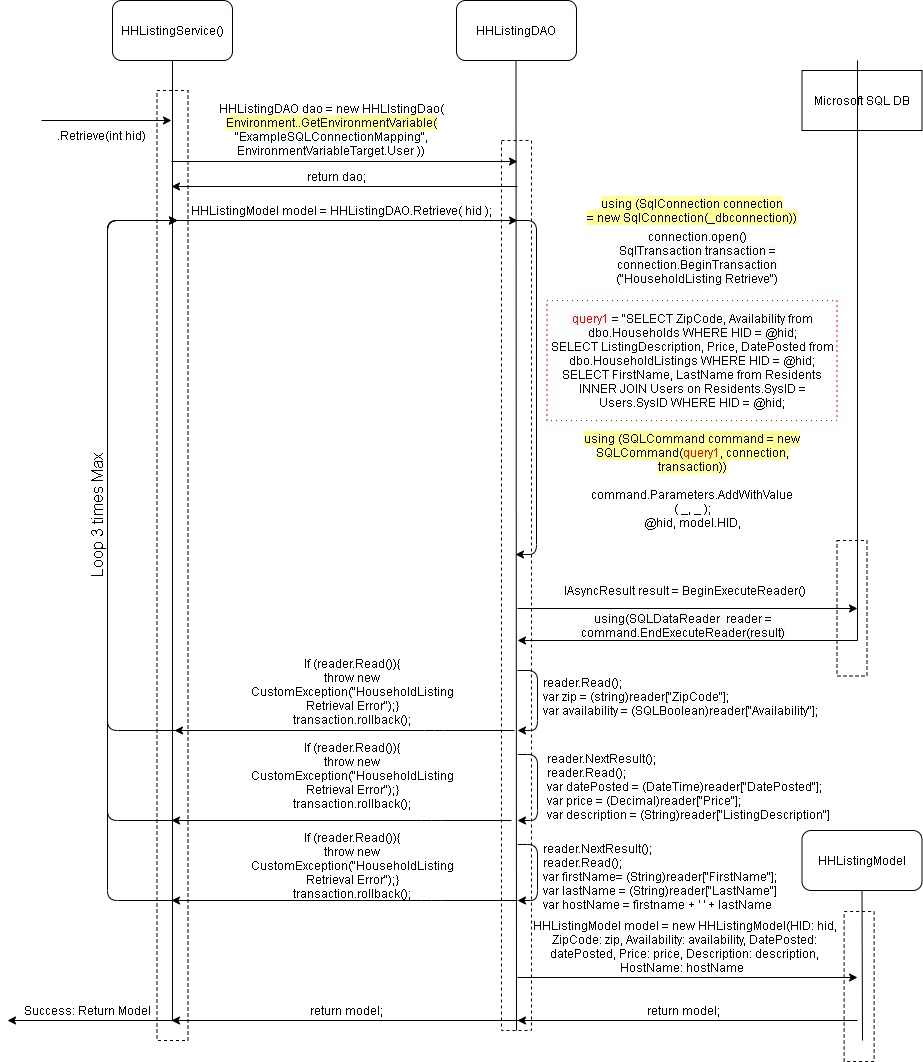
**3.5 Sequence Diagram: HHListingService::Insert ( )**

****

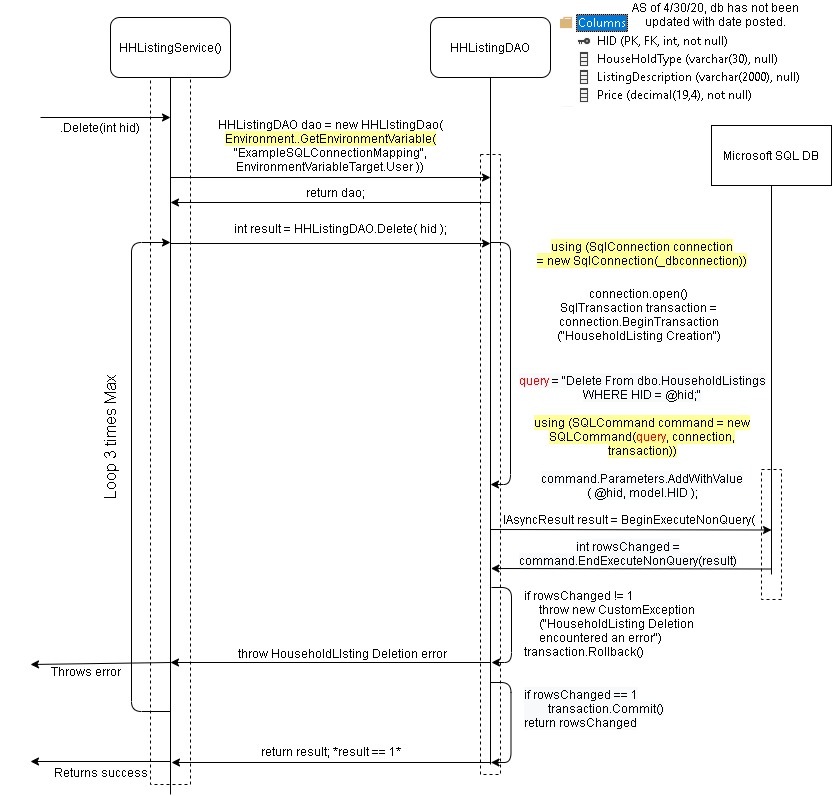
**3.6 Sequence Diagram: HHListingService::Update( )**

****

**3.7 Sequence Diagram: HHListingService::Retrieve( )**

****

**3.8 Sequence Diagram: HHLIstingService::Delete**

****