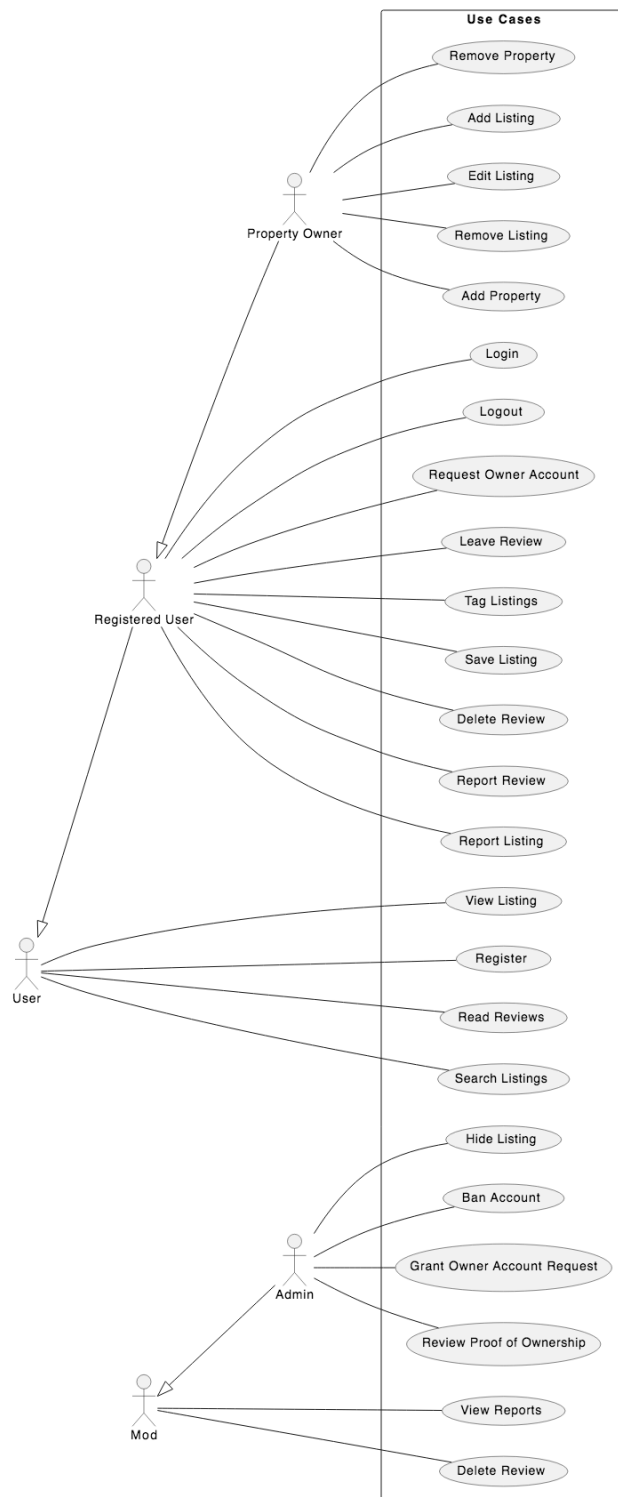


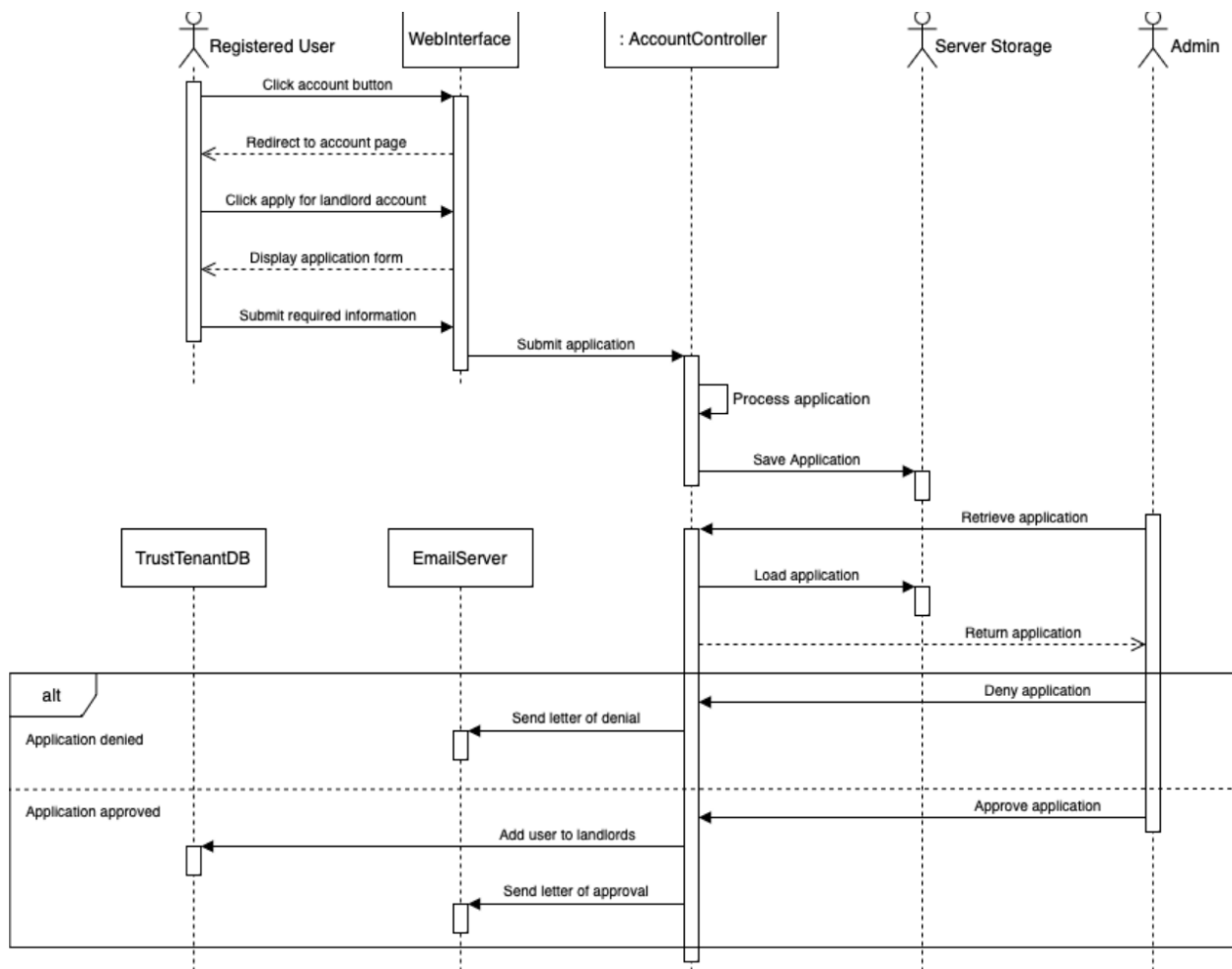
Final Project CPSC 471 Progress Report 3: Functional Design

UML Diagram: Use Case

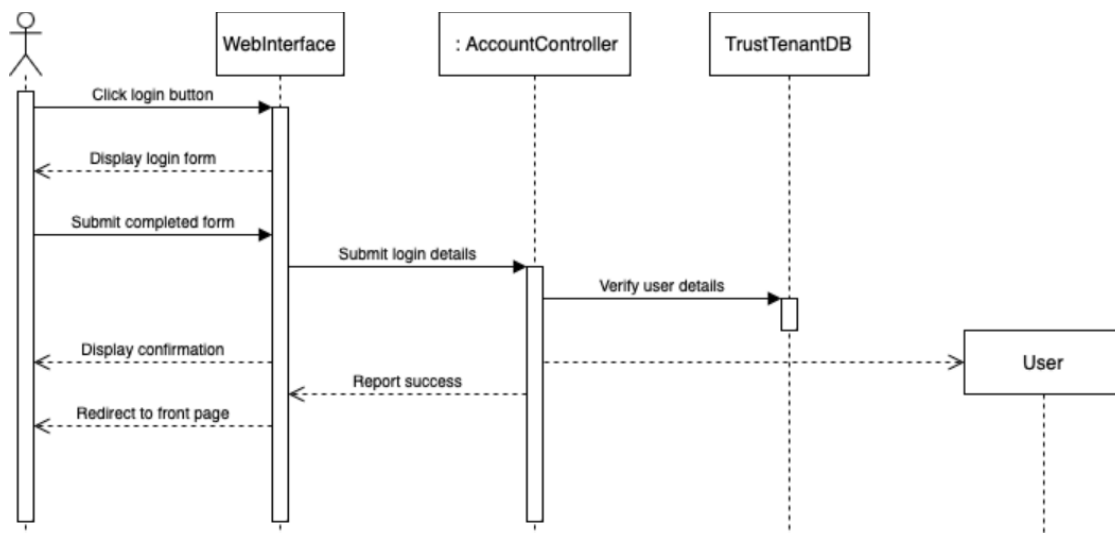


Sequence Diagram

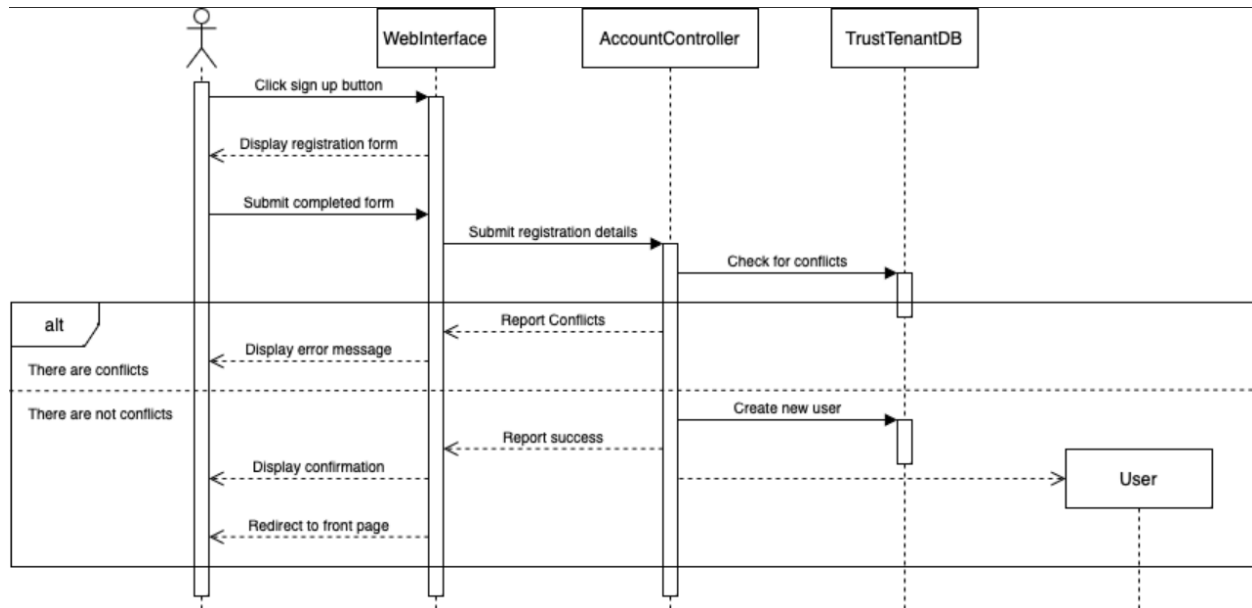
Registered User Requests to Become Property Owner



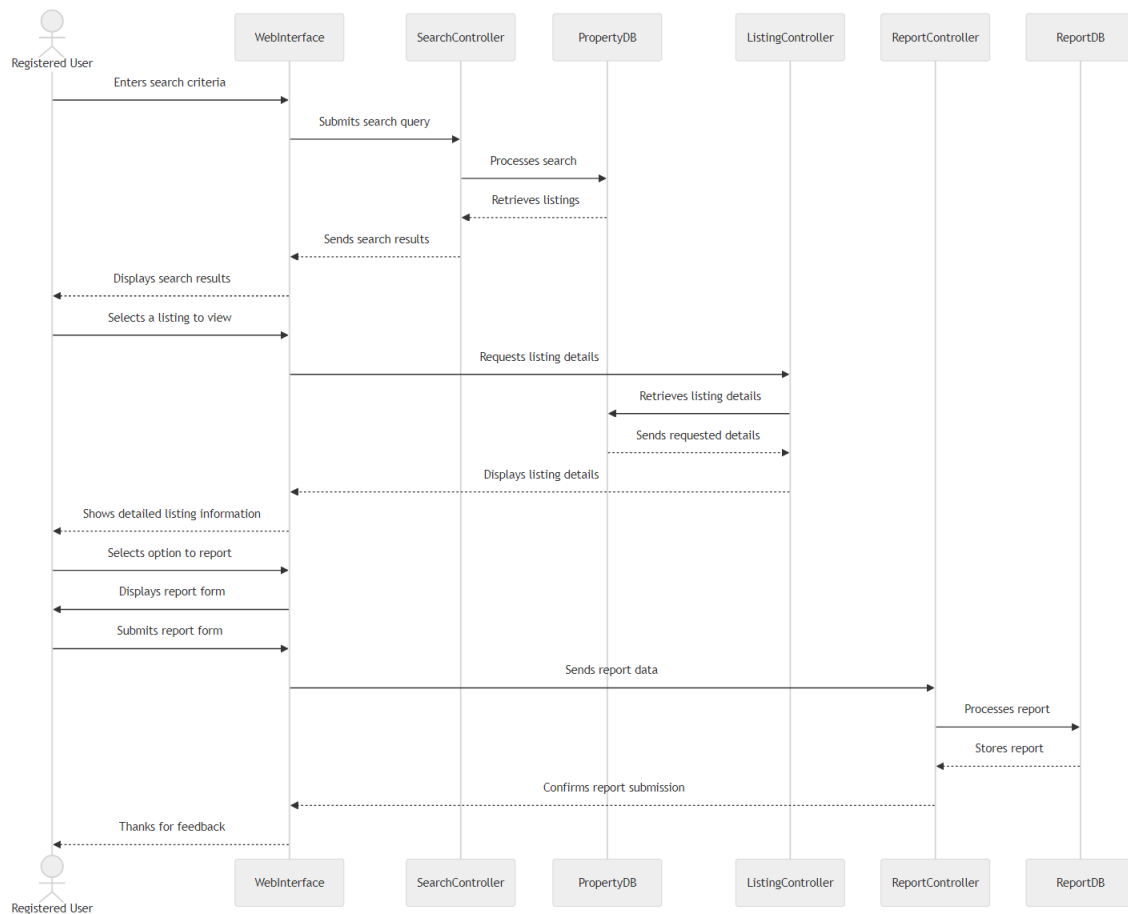
Registered User Logs In



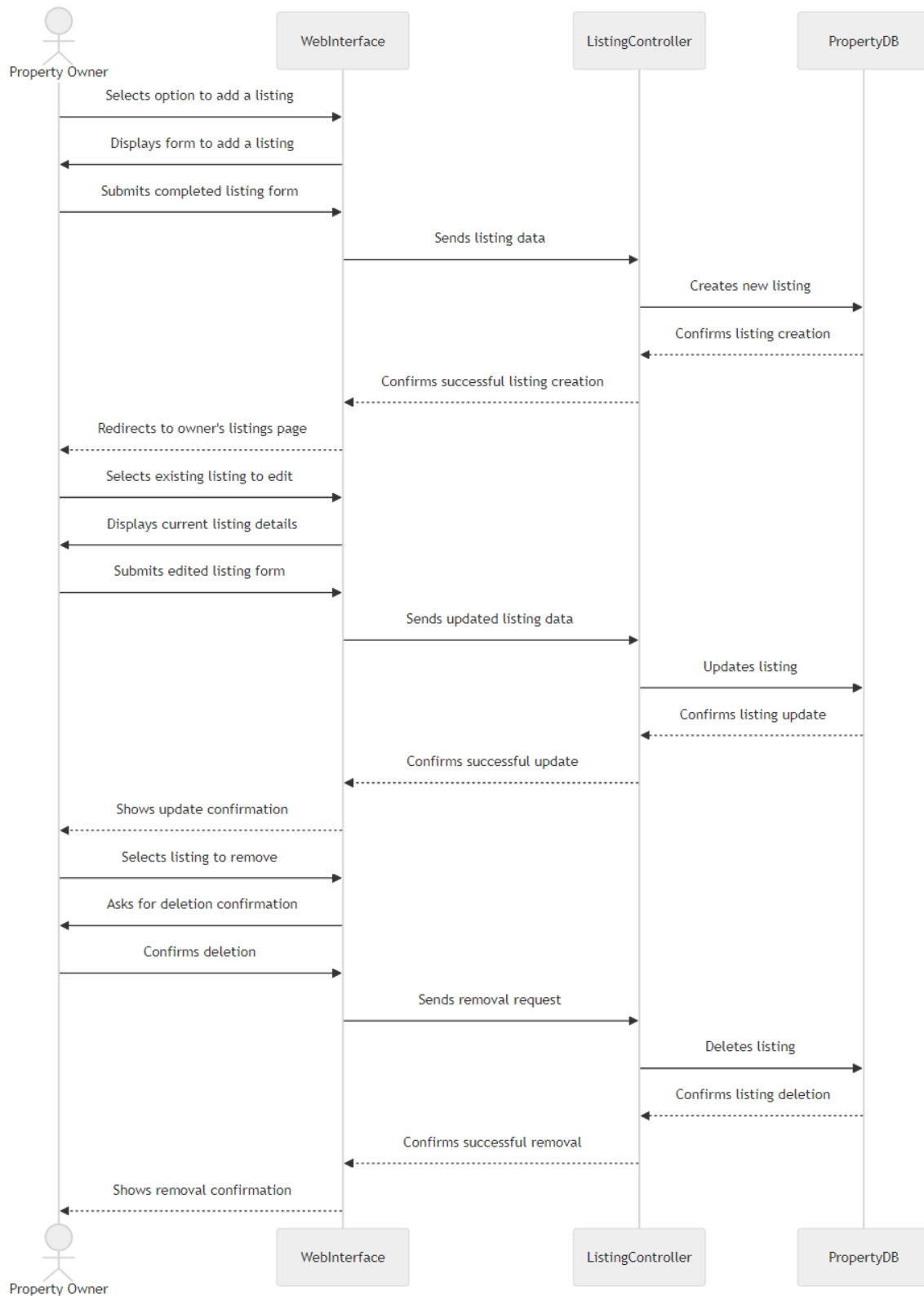
User Becomes Registered User



Registered User Searches Listings, Views Details, and Reports It



Property Owner Adds Listing, Edits and Deletes It



SQL Database Creation Statements

```
CREATE TABLE User (  
    Username VARCHAR,  
    FName VARCHAR,  
    LName VARCHAR,  
    Email VARCHAR UNIQUE,  
    Saved TEXT,  
    Password VARCHAR  
    PRIMARY KEY (Username)  
);
```

```
CREATE TABLE Landlord (  
    Email VARCHAR,  
    Verification BOOLEAN,  
    PRIMARY KEY (Email),  
    FOREIGN KEY (Email) REFERENCES User(Email)  
);
```

```
CREATE TABLE Property (  
    Address VARCHAR,  
    Type VARCHAR,  
    Tags TEXT  
    PRIMARY KEY (ADDRESS)  
);
```

```
CREATE TABLE Listing (  
    U_descriptors TEXT,  
    N_beds INT,  
    Sqr_footage INT,  
    Type VARCHAR,  
    N_baths INT,  
    Listingnum VARCHAR  
    PRIMARY KEY (Listingnum)  
);
```

```
CREATE TABLE Community (  
    Name VARCHAR,  
    Amenities TEXT  
    PRIMARY KEY(Name)  
);
```

```
CREATE TABLE Admin (  
    Admin_ID VARCHAR,  
    Password VARCHAR  
    PRIMARY KEY(Admin_ID)  
);
```

```
CREATE TABLE Listing_comp (  
    Listingnum VARCHAR,
```

```
OtherID VARCHAR,  
PRIMARY KEY (Listingnum, OtherID),  
FOREIGN KEY (Listingnum) REFERENCES Listing(Listingnum)  
  
);  
  
CREATE TABLE Review_comp (  
    ReviewID VARCHAR,  
    OtherID VARCHAR,  
    PRIMARY KEY (ReviewID, OtherID),  
    FOREIGN KEY (ReviewID) REFERENCES Review(ReviewID)  
  
);  
  
CREATE TABLE Complaint (  
    ComplaintID VARCHAR,  
    Body TEXT,  
    Status VARCHAR,  
    Header VARCHAR,  
    Listingnum VARCHAR,  
    PRIMARY KEY (ComplaintID)  
    FOREIGN KEY (Listingnum) REFERENCES Listing(Listingnum)  
  
);  
  
CREATE TABLE Moderator (  
    Username VARCHAR  
    Password VARCHAR,  
    PRIMARY KEY(Username)  
    FOREIGN KEY (Username) REFERENCES User(Username)  
  
);  
  
CREATE TABLE Review (  
    Writes VARCHAR,  
    Score INT,  
    ReviewID VARCHAR,  
    Body TEXT,  
    Header VARCHAR,  
    Listingnum VARCHAR,  
    PRIMARY KEY (ReviewID)  
    FOREIGN KEY (Listingnum) REFERENCES Listing(Listingnum),  
    FOREIGN KEY (Writes) REFERENCES User(Username)  
  
);
```

SQL Data Access and Manipulation Queries

For account login and registration:

```
SELECT *  
FROM USER  
WHERE Email = User supplied email
```

```
INSERT INTO USER (Email, Fname, Lname, Username, Password)  
VALUES (...)
```

```
INSERT INTO LANDLORD (Email)  
VALUES (...)
```

```
UPDATE User  
SET Fname = NewFname, Lname = NewLname, Email = NewEmail, Password = NewPassword  
WHERE Username = EnteredUsername
```

```
DELETE FROM User  
WHERE Username = SelectedUsername
```

```
UPDATE Landlord  
SET Email = EnteredEmail  
WHERE Email = EnteredEmail
```

```
SELECT *  
FROM Listings  
WHERE ListingID = SelectedListingID
```

```
SELECT *  
FROM Listing  
WHERE U_descriptors LIKE '%AddedDescriptor%'  
AND N_beds >= SelectedNumberOfBeds  
AND Sqr_footage >= SelectedSquareFootage  
AND Type = SelectedType  
AND N_baths >= SelectedNumberOfBaths
```

```
DELETE FROM Listing  
WHERE Listingnum = SelectedListingNum
```

```
SELECT *  
FROM Review  
WHERE Listingnum = EnteredListingNum
```

```
INSERT INTO Review (Score, ReviewID, body, Header, Listingnum)  
VALUES (EnteredScore, NewReviewID, EnteredBody, EnteredHeader, EnteredListingNum)
```

```
UPDATE Review  
SET body = NewBody, Header = NewHeader, Score = NewScore  
WHERE ReviewID = SelectedReviewID
```

```
DELETE FROM Review  
WHERE ReviewID = SelectedReviewID
```

```
SELECT *  
FROM Complaint  
WHERE Listingnum = EnteredListingNum
```

```
INSERT INTO Complaint (Body, ComplaintID, Status, Header)  
VALUES (SuppliedBody, NewComplaintID, SuppliedStatus, SuppliedHeader)
```

```
UPDATE Complaint  
SET Status = NewStatus  
WHERE ComplaintID = SelectedComplaintID
```

```
DELETE FROM Complaint  
WHERE ComplaintID = SelectedComplaintID
```