**Group 15 – Rate My Dorm**

**Report 1 – User register**

PURPOSE

Execute when a new user is registering an account in our website.

OBJECTIVE

The purpose of automating the user registration process is to streamline user data entry, prevent duplicate registrations, and securely store user information, thereby improving overall system reliability and user satisfaction.

IMPLEMENTATION

The RegisterUser procedure performs the following key tasks:

**Email Verification**: Checks if the user's email address is already registered to prevent duplicate entries. A SELECT COUNT(\*) query checks for the existence of the email in the user\_table.

**User Data Insertion**: Adds new user details to the database if the email is not already registered. An INSERT INTO statement adds a new user record to user\_table if the email is not already registered.

**Transaction Management**: Ensures that changes are committed only if all operations are successful; otherwise, it rolls back the transaction to maintain data consistency.

TEST

**Test Scenario**

**Test Outcome**

**Report 2 – Review Post**

PURPOSE

Encapsulate the logic of the operation that a user posts a review on a dormitory

OBJECTIVE

In response to the need for more efficient and reliable review data management within our dormitory management system. This package is intended to streamline operations by automating review submissions and providing immediate insights into dormitory performance through average score calculations.

IMPLEMENTATION

The review\_pkg package consists of two main elements:

**Procedure add\_review**: This procedure facilitates the addition of new review entries to the review table. It accepts multiple parameters representing various aspects of a review and inserts them into the database, using a sequence to ensure unique identifiers for each entry.

**Function calculate\_average\_score**: This function calculates the average overall room score for a given dormitory based on existing review data, helping administrators quickly gauge overall resident satisfaction.

There’s also a trigger called ***trg\_update\_room\_score*** which will automatically calculate the average score of a dormitory and update it once there’s a new review was posted.

TEST

**Test Scenario**

**Test Outcome**

**Report 3 – Rent Lease**

PURPOSE

The package is designed to automate key aspects of lease management within our property management system. It encapsulates the logic of the operation that a user rents a lease on our website.

OBJECTIVE

It includes procedures and functions to add new leases, check the availability of properties, update rent values, and calculate the total amount due on a lease. It can streamline processes in adding, updating, and managing leases in a reliable and efficient manner, which streamline operations, enhance accuracy, and improve the overall efficiency of lease management.

IMPLEMENTATION

The package includes the following components:

**Procedure add\_lease**: This procedure will add a new lease record to the database.

**Function is\_available**: This function checks if a property is available for rent on a given date, make sure there’s no conflict occurs.

**Procedure update\_rent:** This procedure updates the monthly rent for an existing lease.

**Function calculate\_due\_amount**: This function calculates the total amount due for a lease, including any penalties for late payments

TEST

**Test Scenario**

**Test Outcome**

**Report 3 – Rent Lease**

PURPOSE

This report outlines the implementation of an automated system designed to calculate and continuously update the rankings of the top 5 dormitories based on their scores.

OBJECTIVE

The system comprises a database table, a function, a stored procedure, a trigger, and an encompassing package, all aimed at enhancing the transparency and competitiveness of dormitory ratings within a university housing system. It will ranking dormitories based on calculated scores from various criteria supports strategic improvements and provides clear metrics for potential residents. By providing a transparent, accurate, and up-to-date ranking of facilities, the system aids prospective and current residents in making informed decisions, while enabling administrators to pinpoint and prioritize areas for improvement.

IMPLEMENTATION

The system consists of several integrated components:

**TOP\_DORMITORIES Table**: Stores the top 5 dormitories with their calculated total scores and rankings.

**CalculateTotalScore Function**: Calculates the total score for a given dormitory. It will truncates the current data, then recalculates scores using the CalculateTotalScore function and inserts the new top 5 dormitories based on these scores.

**UpdateTopDormsTrigger Trigger:** Ensures the TOP\_DORMITORIES table is updated automatically after any update to the environment\_score, location\_score, or facility\_score in the DORMITORY table. Activates the UpdateTopDormitories procedure whenever relevant scores are updated.

**DormitoryManagement Package**: Encapsulates the CalculateTotalScore function and UpdateTopDormitories procedure. Simplifies management and updates of the functionalities related to dormitory scoring and ranking.

TEST

**Test Scenario**

**Test Outcome**

**Report 4 – Rent Lease**

PURPOSE

This is used to check the leases that is approaching the expire date, which can help the administrator to notify the user to extend lease or can publish the rent information on the website.

OBJECTIVE

The enhancements to the lease management system have significantly improved the ability to proactively manage lease expirations. This automation ensures that property management teams can focus on tenant relations and property maintenance without the burden of manually tracking lease terms. Future recommendations include integrating these notifications with email or SMS services to alert both staff and tenants directly.

IMPLEMENTATION

The system consists of several integrated components:

**TOP\_DORMITORIES Table**: Stores the top 5 dormitories with their calculated total scores and rankings.

**DaysUntilExpiration Function**: Calculates the number of days until a lease expires.

**GetExpiringLeases Procedure:** Identifies leases that are expiring within the next 30 days.

**NotifyOnLeaseUpdate Trigger**: Automatically triggers the GetExpiringLeases procedure whenever the lease\_end\_date is updated and falls within the 30-day window.

**LeaseManagement Package**: Encapsulates all related functions and procedures for managing lease expirations into a single package to streamline operations and maintenance.

TEST

**Test Scenario**

**Test Outcome**