**GoalManager**

*User manual*

**Abstract**

GoalManager is an enterprise resource management tool that supplies an internal management interface for employee goals. The software features a web-based interactive front-end that serves as a shell for interacting with a rigorous multi-user role-based back-end. User roles include Employee, Supervisor, and Administrator. Employees work within a Department and can create, view, edit, and update Goals that align with Department Categories and Quarter dates as deadlines. Supervisors also share Employee functionality regarding Goals, and include two methods for analyzing reports: viewing Department-level reports, and individual Employee reports. Administrators handle the creation and modification of Employees, Supervisors, Departments, Categories, and Quarters. GoalManager uses a SQL database for all business data, and an Identity database for role-based authentication.

This document describes the user manual for operating GoalManager. The document is ordered as follows: Chapter 1 describes the program and its system requirements. Chapter 2 details user roles. Chapter 3 covers user role functions. Chapter 4 illustrates the main database schema and the ASP.NET Identity database.

This document describes functionality of GoalManager v.1.0, dated 28 April 2017.

Table of Contents

[1 System overview 1](#_Toc385946325)

[1.1 Program interface 1](#_Toc385946326)

[1.2 System requirements 1](#_Toc385946327)

[2 User Roles 2](#_Toc385946328)

[2.1 Roles 2](#_Toc385946329)

[2.1.1 Employee 2](#_Toc385946330)

[2.1.2 Supervisor 3](#_Toc385946331)

[2.1.3 Administrator 4](#_Toc385946332)

[3 Role-based Functions 5](#_Toc385946341)

[3.1 Employee Functions 5](#_Toc385946342)

[3.1.1 Create Goal 5](#_Toc385946344)

[3.1.2 View Goal 6](#_Toc385946344)

[3.1.3 Update Goal 7](#_Toc385946344)

[3.2 Supervisor Functions 8](#_Toc385946343)

[3.2.1 Create Goal 8](#_Toc385946344)

[3.2.2 View Goal 9](#_Toc385946345)

[3.2.3 Update Goal 10](#_Toc385946346)

[3.2.4 Add Category 11](#_Toc385946346)

[3.2.5 View Employee Report 12](#_Toc385946346)

[3.2.6 View Department Report 15](#_Toc385946346)

[3.3 Administrator Functions 16](#_Toc385946347)

[3.3.1 Create Employee 16](#_Toc385946348)

[3.3.2 Create Department 16](#_Toc385946349)

[3.3.3 Modify Employee 18](#_Toc385946348)

[3.3.4 Modify Department 19](#_Toc385946348)

[4 Database Schema 20](#_Toc385946368)

[4.1 UserDB 20](#_Toc385946369)

[4.2 IdentityDB 22](#_Toc385946370)

# System overview

## Program interface

GoalManager includes an interactive user interface as displayed in Figure 1.1. All system functionality is accessible through this main interface through the Login portal. Role-based use cases are restricted to logged in users with the appropriate role.

Users can click the “Log in” button to access the log in portal. Only authorized users can log in. The TaxSlayer logo redirects to the /Home/Index page from any place on the website.

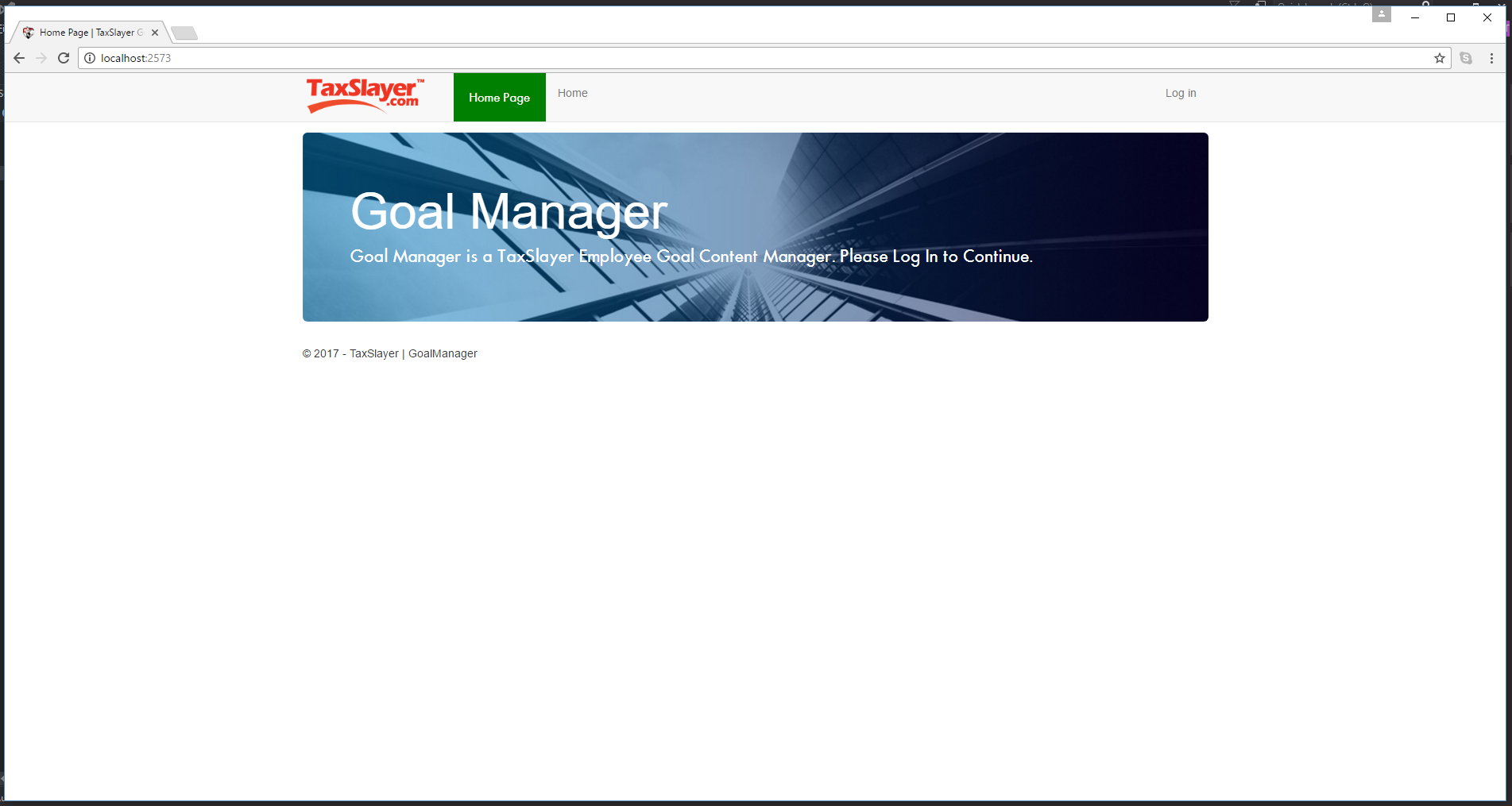


Figure 1.1. /Home/Index

## System requirements

GoalManager is written in C# 7.0 within the .NET framework. GoalManager is implemented on ASP.NET MVC architecture. As a web-based application, GoalManager must be hosted on an internal web server for all operations.

# User Roles

## Roles

GoalManager provides three user roles: Employees, Supervisors, and Administrators.

### Employee

Employees are presented with EmployeeHome, featured in Figure 2.1. EmployeeHome displays any Pending Goals awaiting Supervisor approval, followed by all current Active Goals and each associated Update. Next are all Denied Goals, which have been denied by a Supervisor. Last are all Failed Goals, which detail all Goals that have a due date past the system time.

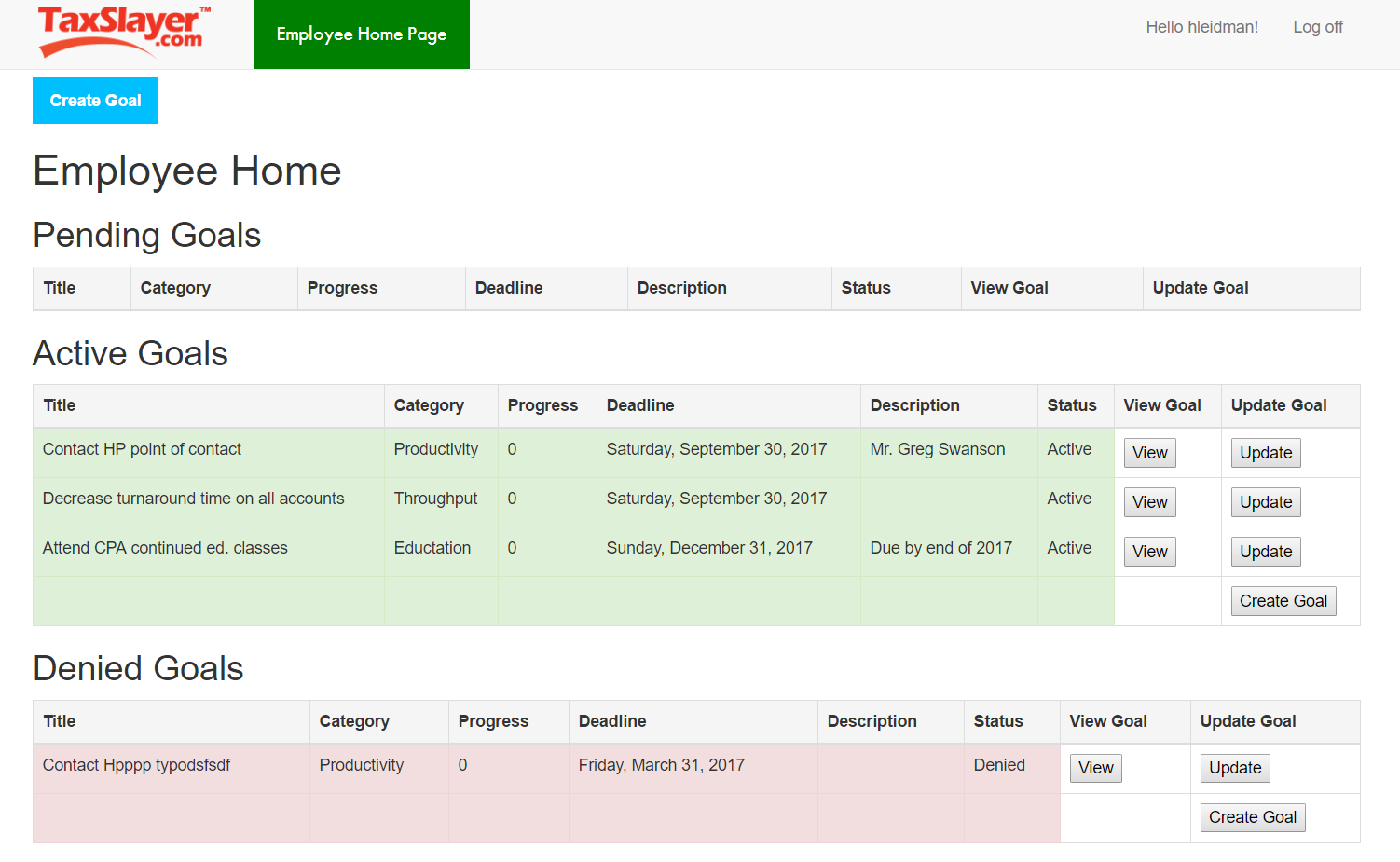


Figure 2.1. /Home/EmployeeHome

### Supervisor

Supervisors are delivered to SupervisorHome, which is shown in Figure 2.2. Supervisors differ from Employees primarily by the management role that every Supervisor has. A Supervisor manages a Department, and by extension, all Employees in that Department. Much like EmployeeHome, Supervisors have the same Goal tables as Employees, as detailed in 2.1.1. Furthermore, Supervisors have a Pending Goals queue not for their own goals, but for all managed Employees in their Department, where they can either Approve or Deny queued Goals from managed Employees.

Like Employees, Supervisors can click the “Create Goal” button to create a personal Goal, and view all their Goals via the “View” button.

Supervisors can add Department Categories through the “Add Category” button.

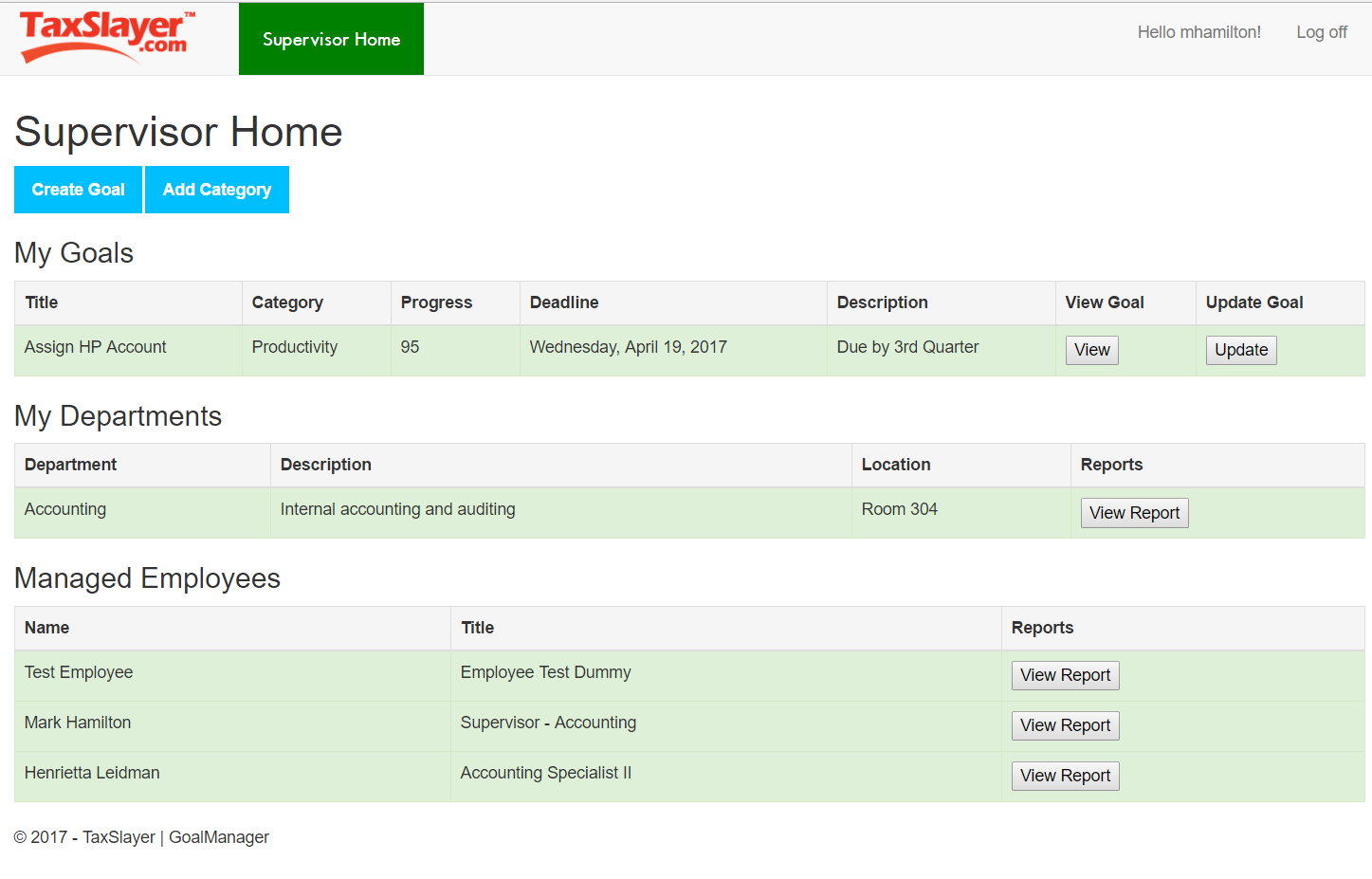


Figure 2.2. SupervisorHome

### Administrators

Administrators differ from Employees and Supervisors in that they do not have any Goal functionality, nor participate in conventional business Departments. All valid Administrators are restricted to the Administrators Department. Administrators are presented with AdminHome, which list a table of all current Administrators, a table of all Departments and their Supervisors, and a table for all non-Administrator users.

Administrators can modify a selected Department by clicking the “Modify” button attached to that Department. Modifying a Department includes adding Quarters, Categories, and changing the Supervisor of that Department. They can also modify Employees and Supervisors by clicking the “Modify” button attached to that Employee/Supervisor. Administrators can also modify other Administrators via the same interaction as modifying a Department or Employee/Supervisor.

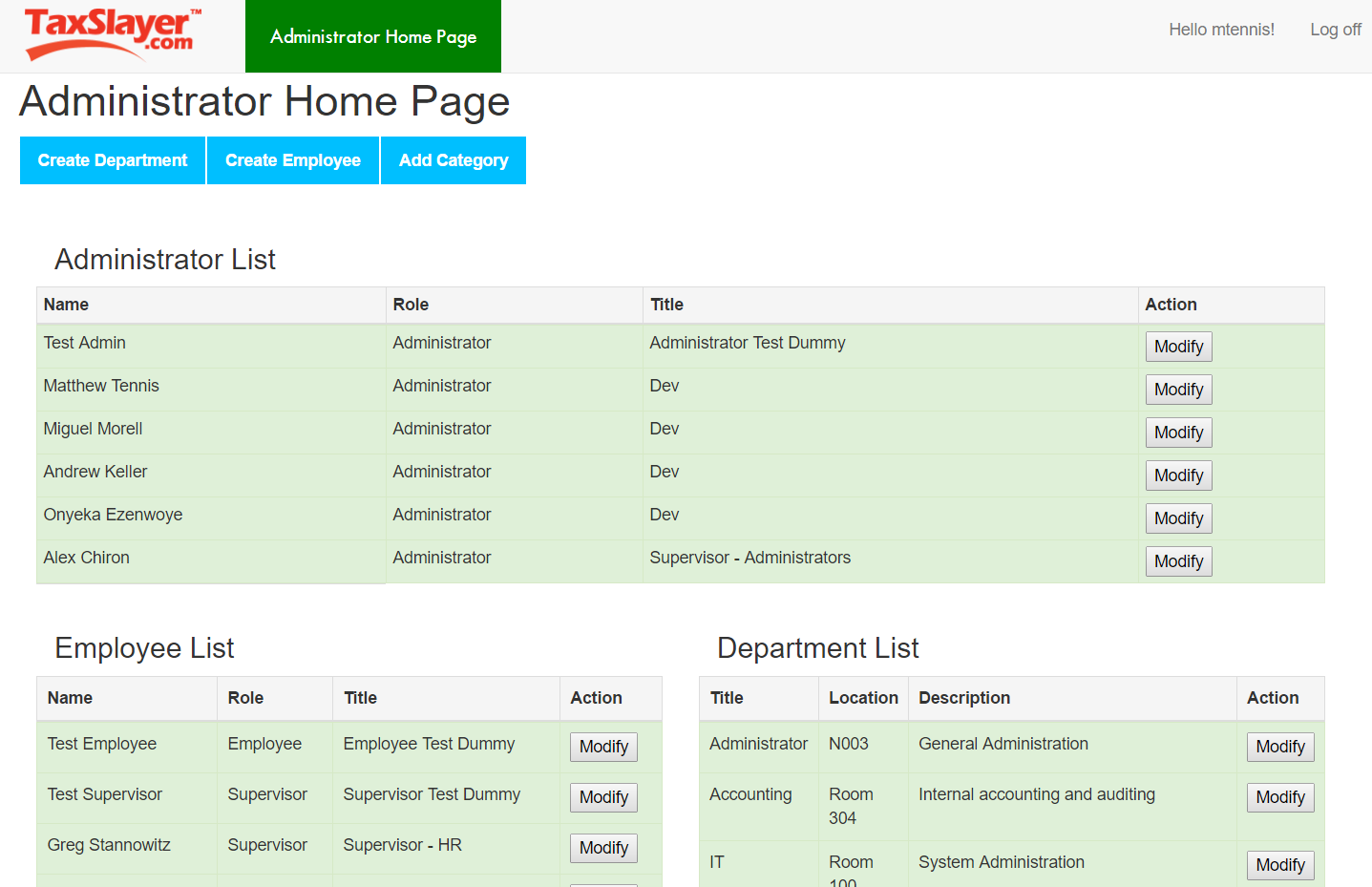


Figure 2.3. /Home/AdminHome

# Role-based Functions

## Employee Functions

Employees interfacing with GoalManager do so for its titular purpose: to manage Goals. Employees can first view any Goals currently assigned to them. Create Goal is displayed below as Figure 3.1. Employees can create Goals under Department Categories and Quarter dates. Employees can click the “Create Goal” button to create a new goal pending Supervisor approval.

### Create Goal

Employees submit a valid Title, an optional Description, and a selected Department Category and Quarter date. Clicking the “Submit” button checks validation and creates a new Goal with the Status of “Pending”. Pending Goals can either be approved or denied by the Employee’s Supervisor.

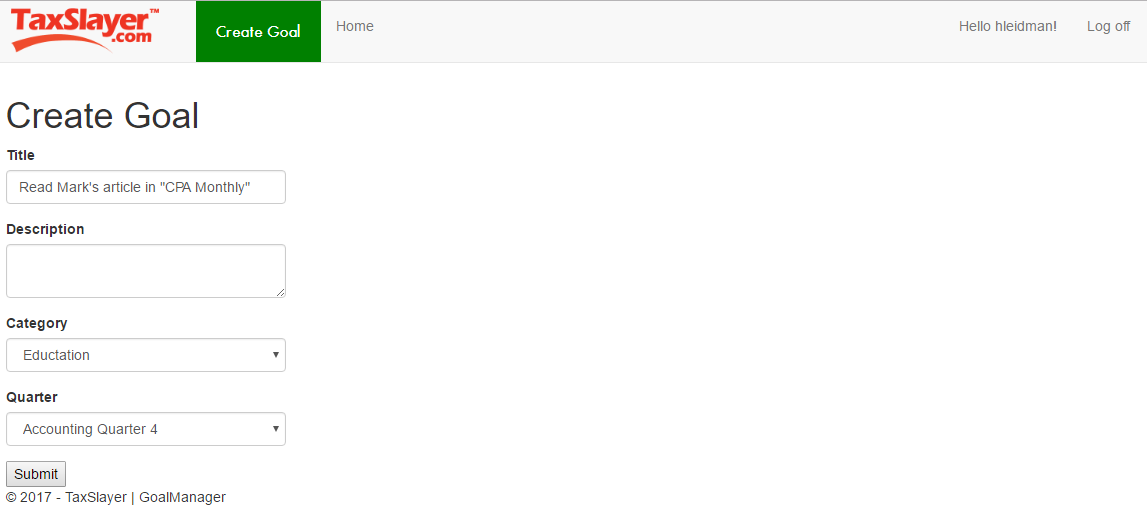


Figure 3.1. CreateGoal

### View Goal

Employees can also click the “View” button to view the history of that Goal, which includes all Updates applied to that Goal. Viewing detailed Goal history is featured in Figure 3.2.

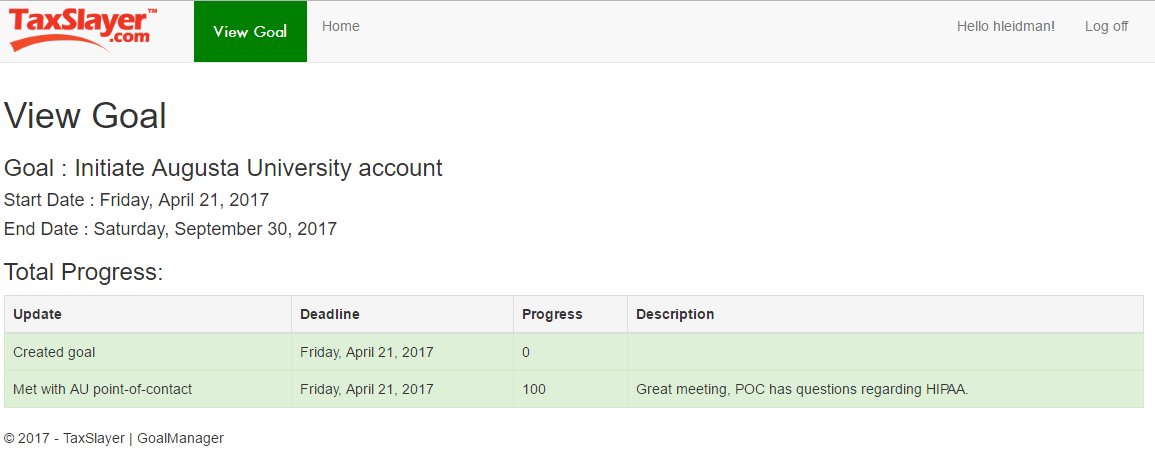


Figure 3.2. ViewGoal

### Update Goal

Employees can update a Goal by clicking the “Update” button attached to the Goal’s list element. The UpdateGoal page is featured in Figure 3.3.

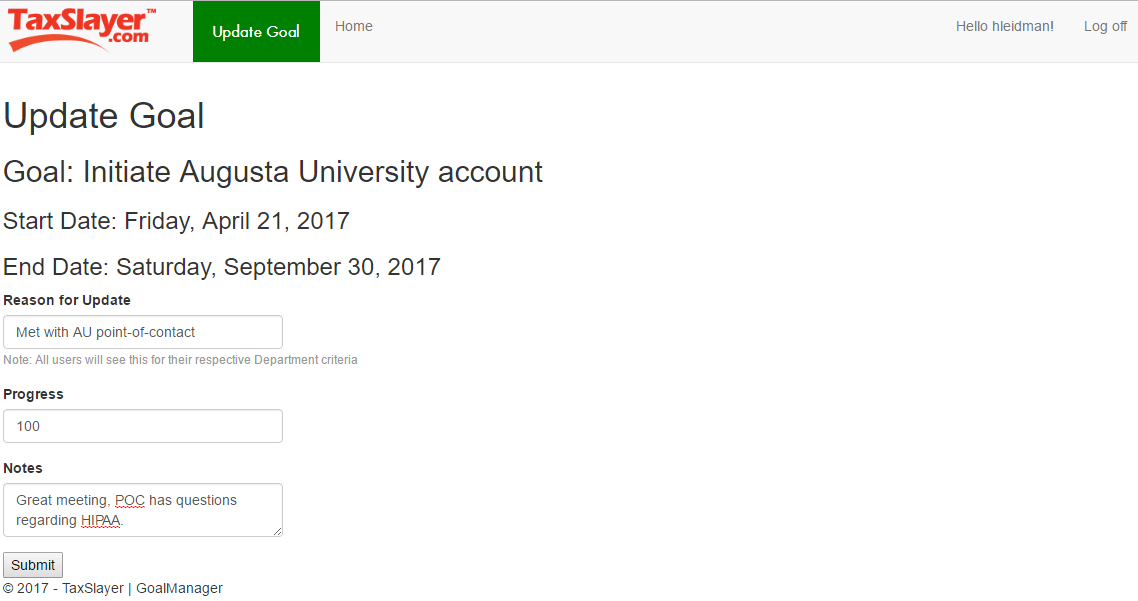


Figure 3.3. UpdateGoal

## Supervisor Functions

### Create Goal

Supervisors can create Goals alongside the Quarter dates and Categories of the Department that they manage. However, Supervisor also have the ability to push a Goal out to every Employee in his or her Department. This is accomplished by the “Push to Department?” checkbox on the Create Goal page accessible only to Supervisors. Supervisor created Goals and Department-wide Goals are automatically given a status of “Active”. Figure 3.4 displays this functionality.

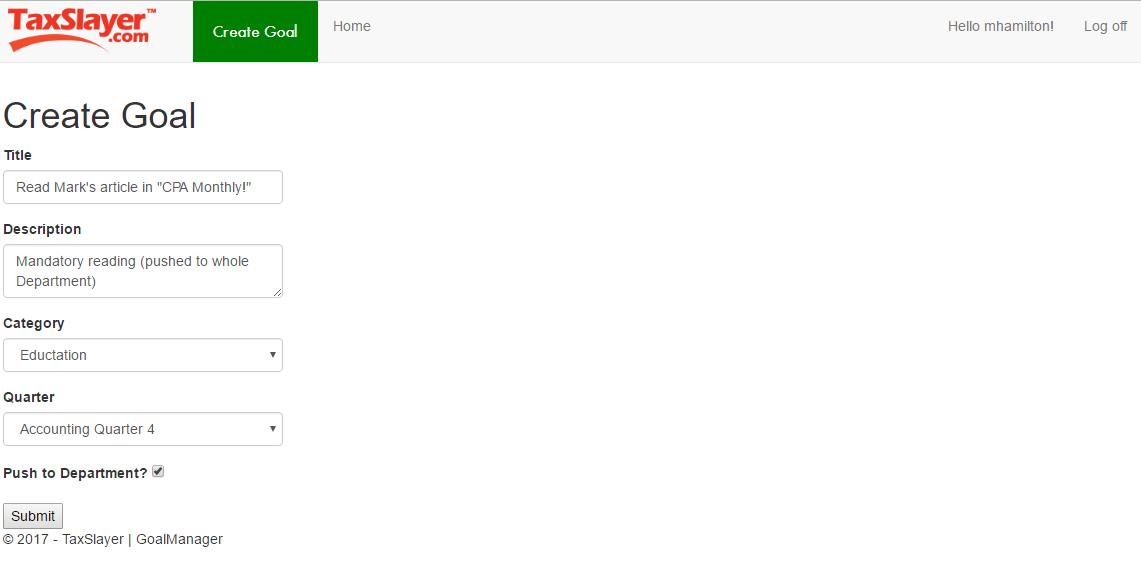


Figure 3.4. CreateGoal (Supervisor)

### View Goal

Like Employees, Supervisors can View one of their own Goals by clicking the View button. Figure 3.5 displays ViewGoal for Supervisors.

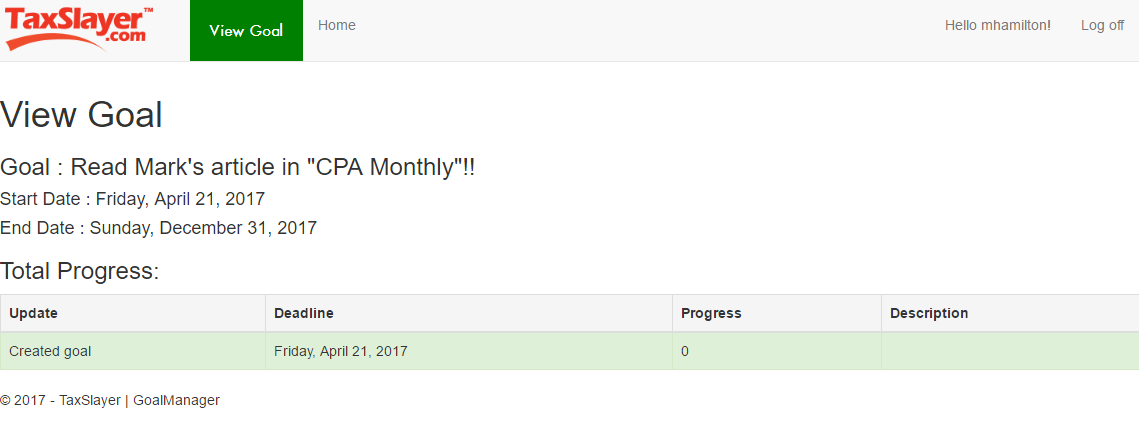


Figure 3.5. ViewGoal (Supervisor)

### Update Goal

Supervisors can also Update any of their own Goals with the same mechanic as Employees, as featured in Figure 3.6.

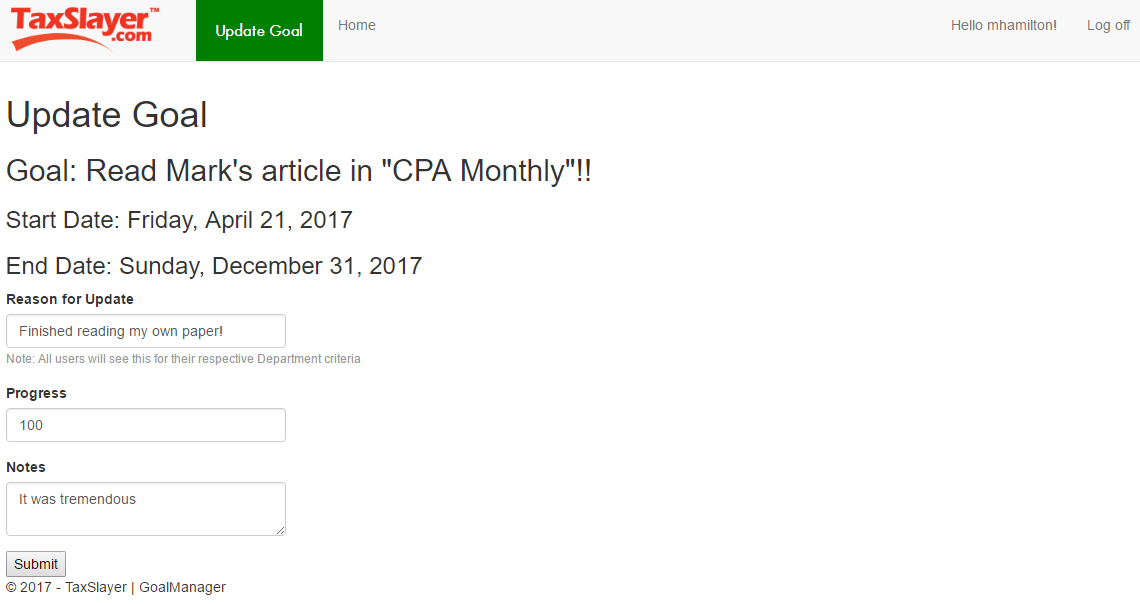


Figure 3.6. UpdateGoal (Supervisor)

### Add Category

Supervisors can add Goal Categories for their Department. By clicking “Add Category” on SupervisorHome, a list of current Department Categories will be presented, alongside a form field for entering a new Category. The “Submit” button adds the new Category to the Department. AddCategory is featured in Figure 3.7.

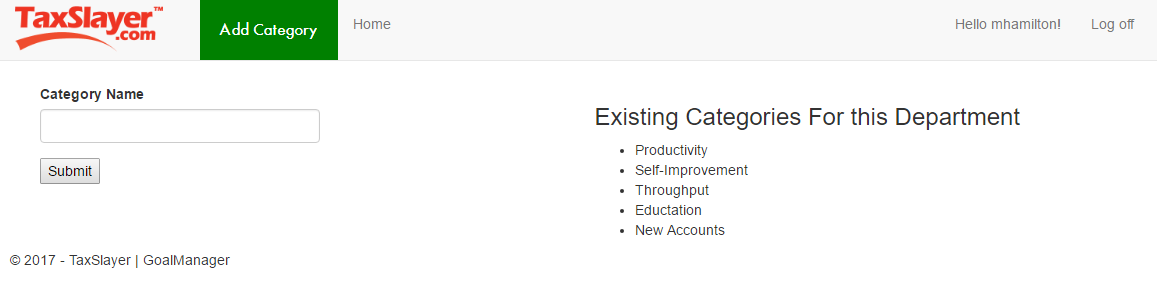


Figure 3.7. AddCategory

### View Department Report

Supervisors can view a report of the entire Department that he or she manages. On SupervisorHome, the Supervisor can click “View Report” on the Department list item. The departmental report presents aggregated Goal data across the entire Department. Figure 3.8 features the ViewDepartmentReport pie chart infograph for Department-wide statistics, Figure 3.9 features the bar graphs for each Employee, and Figure 3.10 displays the tabular report for each Employee.

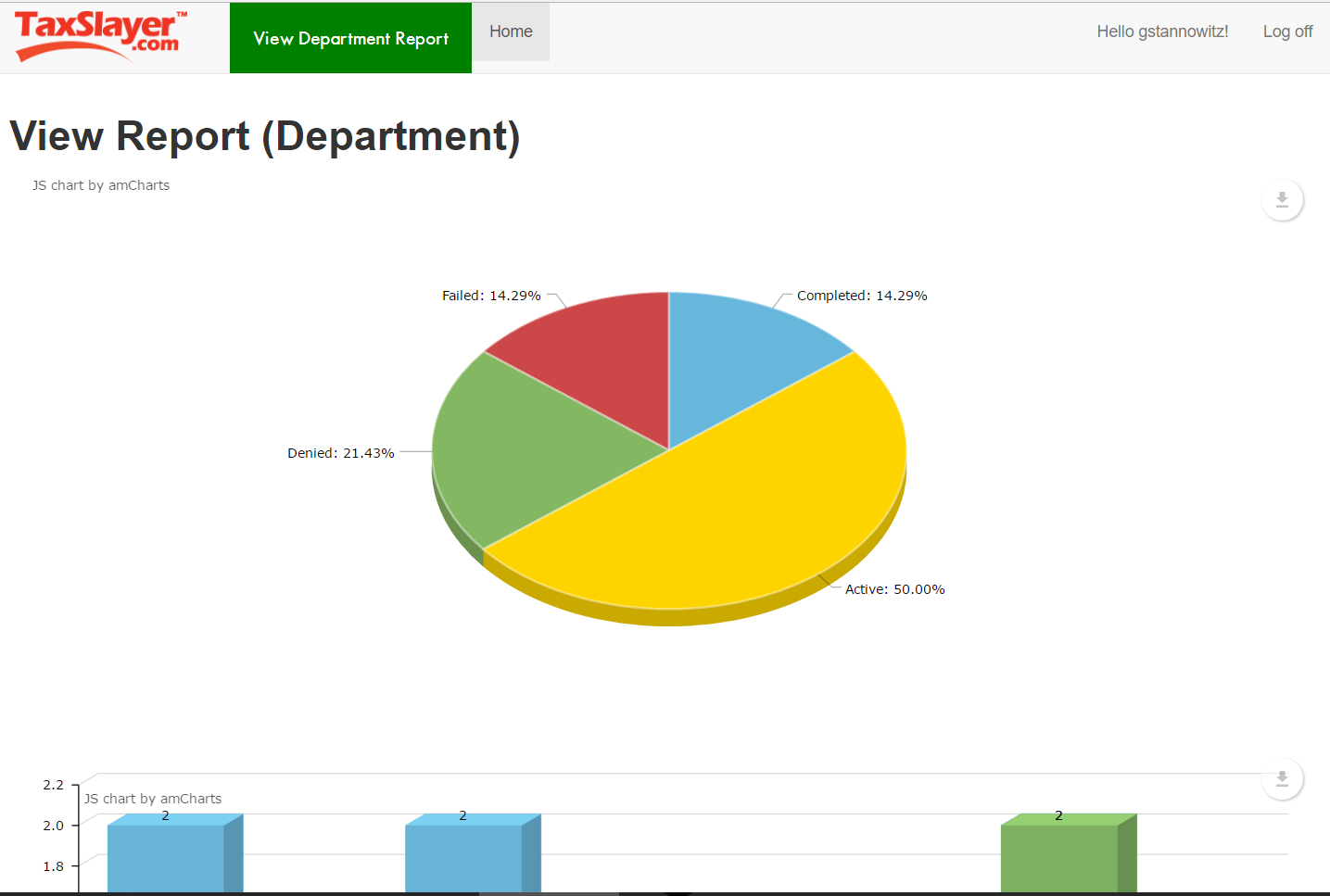


Figure 3.8. ViewReport (Department) Pie Chart

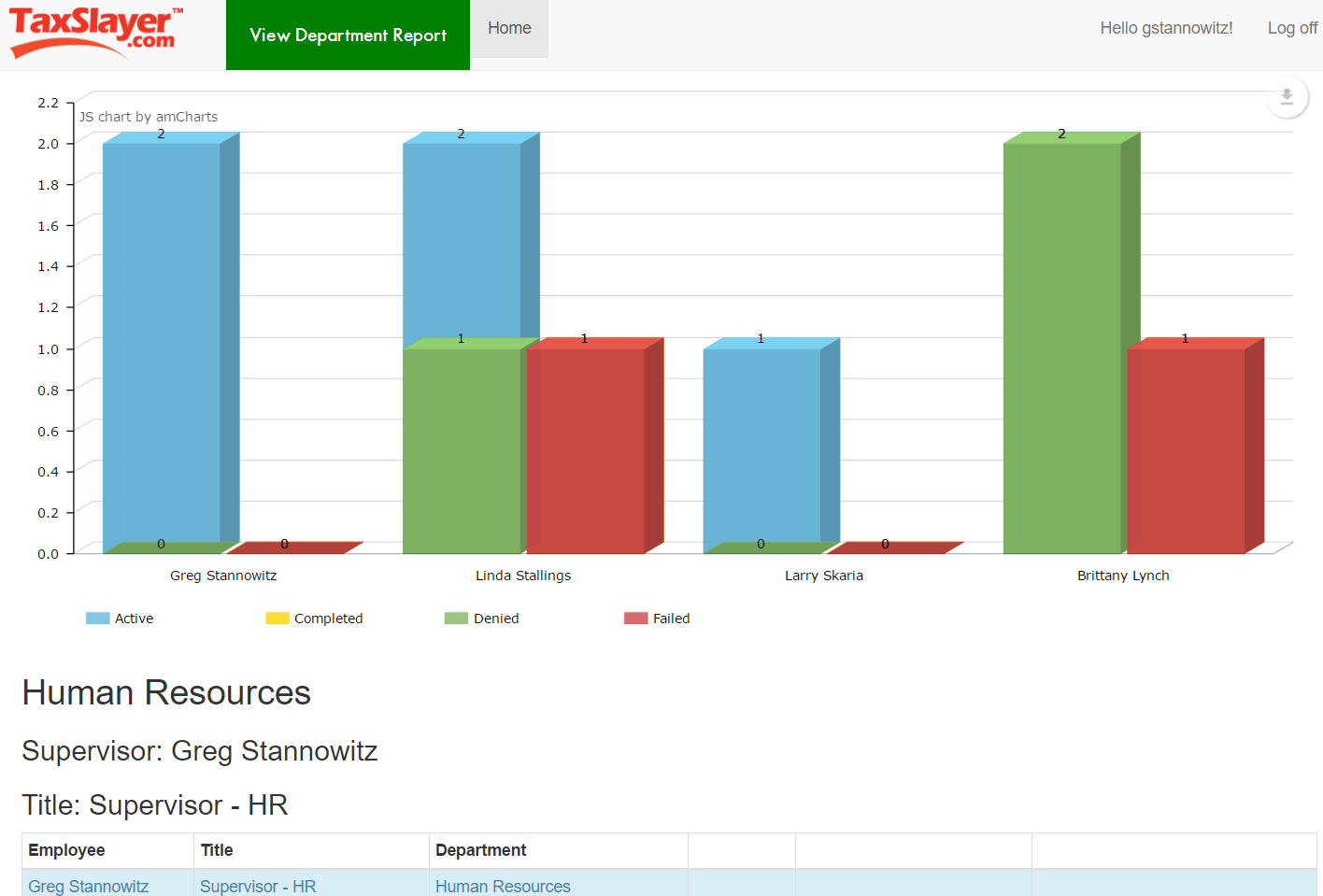


Figure 3.9. ViewReport (Department) Bar Graphs

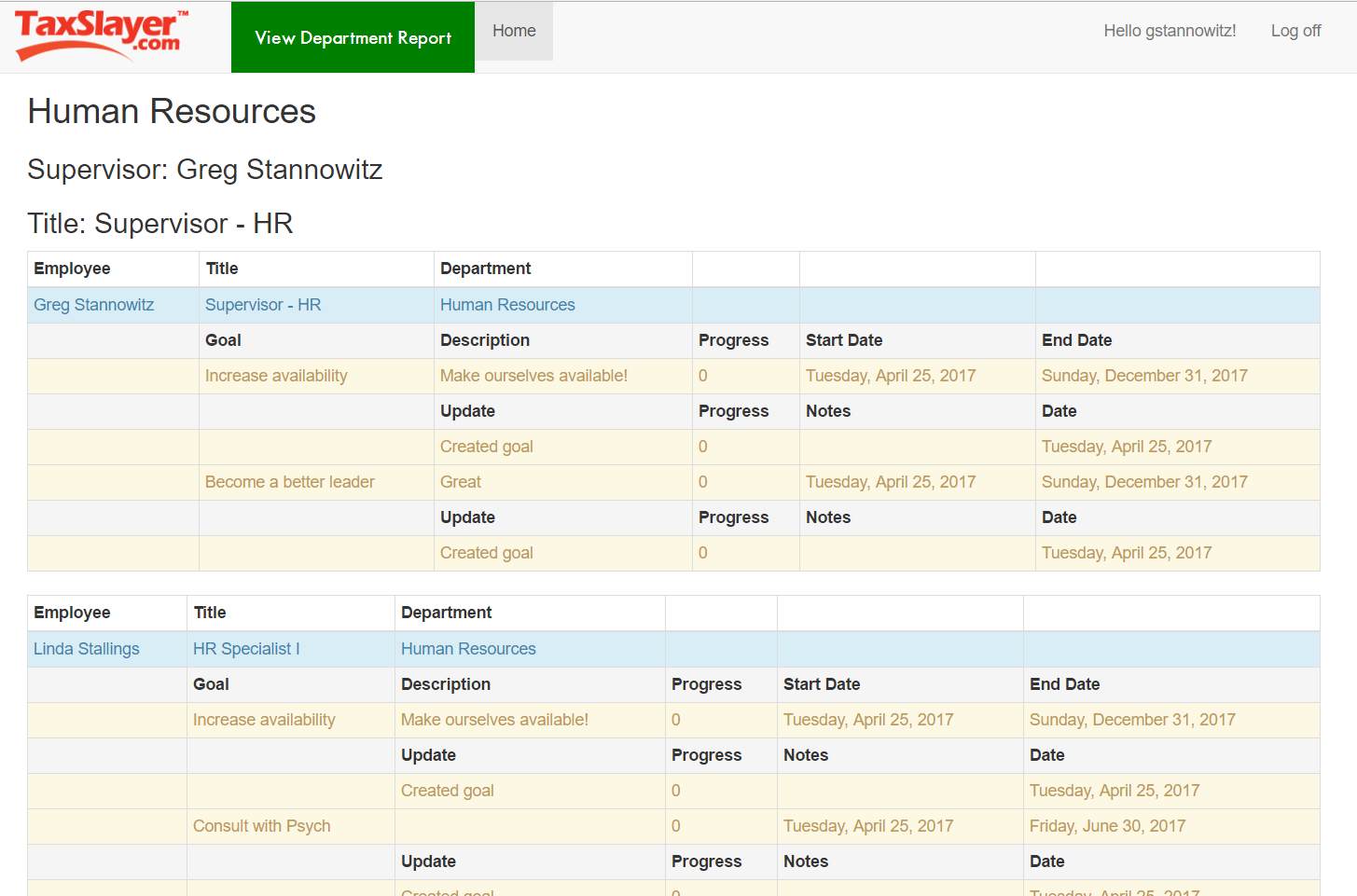


Figure 3.10. ViewReport (Department) Tabular Display

### View Employee Report

Similar to viewing a departmental report, a Supervisor can review any managed Employees by clicking the “View Report” button on an Employee’s list entry. The Employee report displays specific Goal data for that Employee, presenting those statistics in a graphical format. Figure 3.11 shows an individual Employee report.

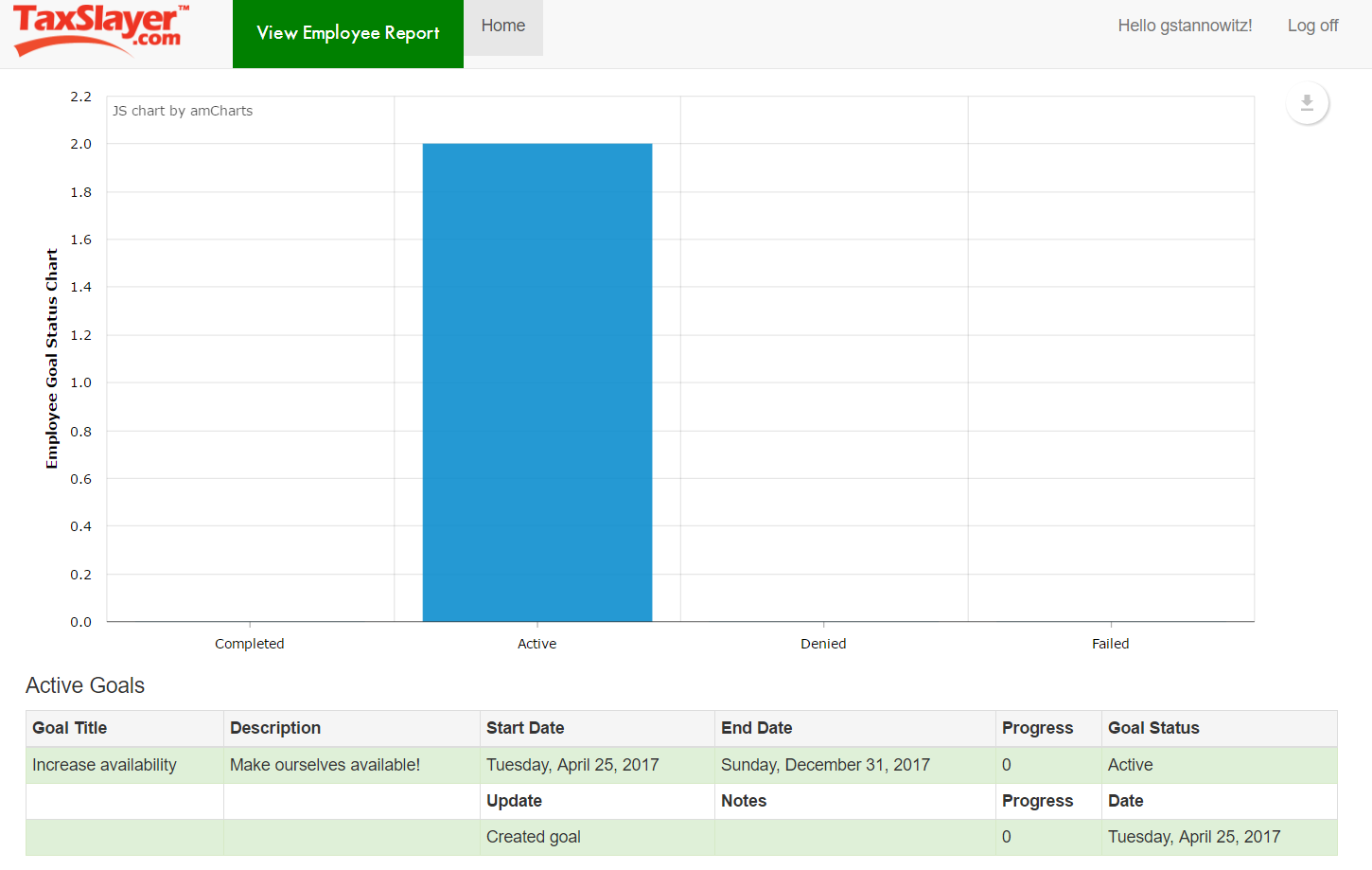


Figure 3.11. ViewReport (Employee)

## Administrator Functions

### Create Employee

Administrator functionality is limited to the creation and modification of Departments and all Users. Functions include CreateEmployee, CreateDepartment, ModifyEmployee, ModifyDepartment, and AddCategory. Figure 3.12 displays the view for creating new Employees.

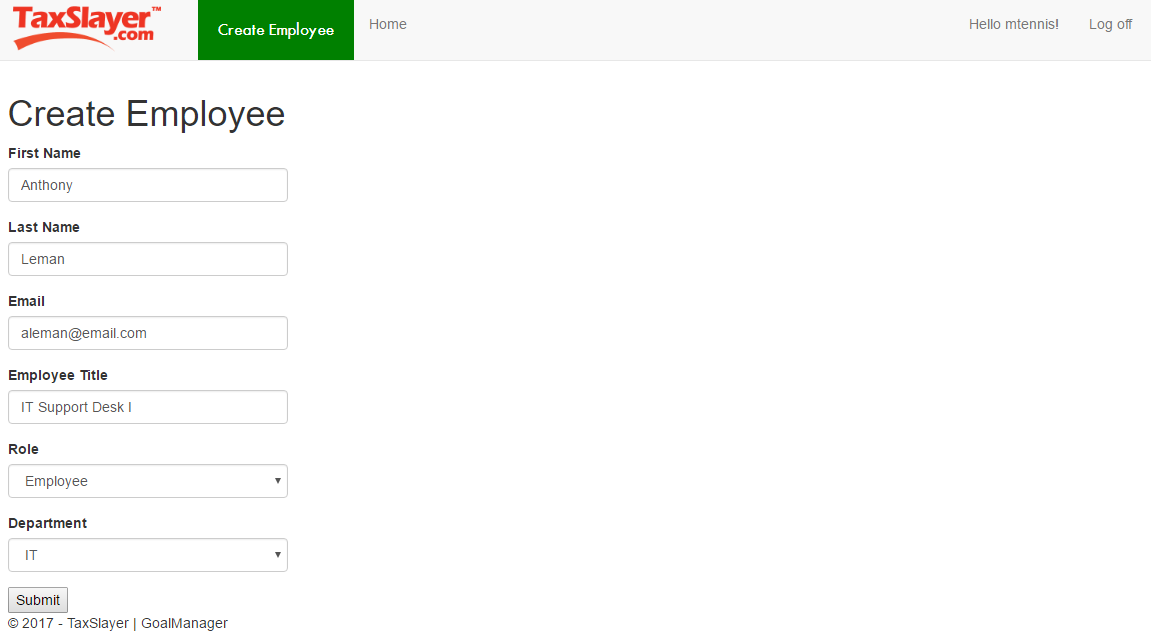


Figure 3.12. CreateEmployee

### Create Department

Administrators can navigate to CreateDepartment by clicking on the “Create Department” button at the top of AdminHome. The Administrator must provide a name for the Department, a physical location, an optional description, select a Supervisor from the pool of Supervisors, provide at least one Category (in the Category 1 field), and four Quarters with valid, chronological dates. CreateDepartment is featured Figure 3.11.

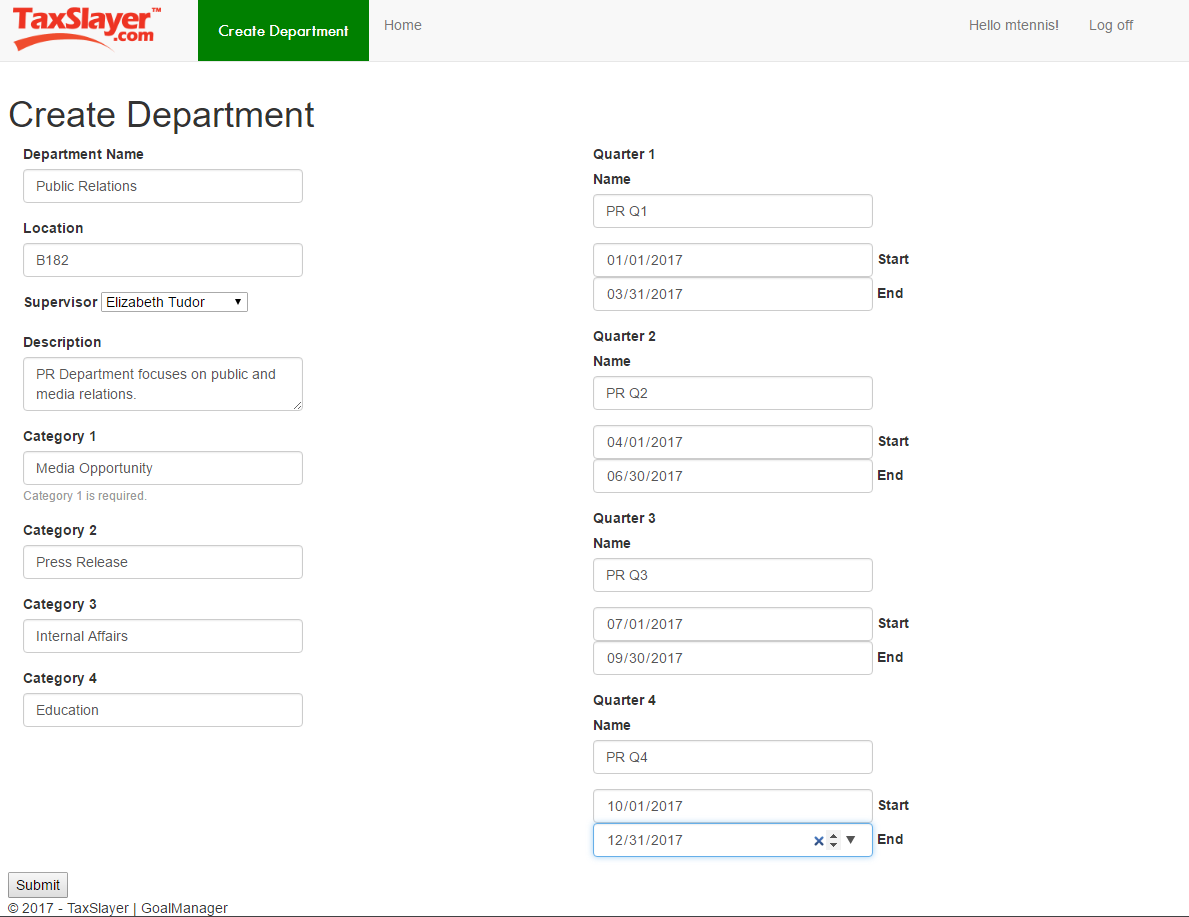


Figure 3.13. CreateDepartment

### Modify Employee

Administrators can navigate to ModifyEmployee by clicking on the “Modify” button attached to any User on AdminHome. The Administrator must provide a valid first and last name for the Employee, an email address, a role, the Employee’s active status, and optionally, change the Employee’s Department. Clicking the “Submit” button will change the Employee’s record in the database. ModifyEmployee is featured in Figure 3.14.

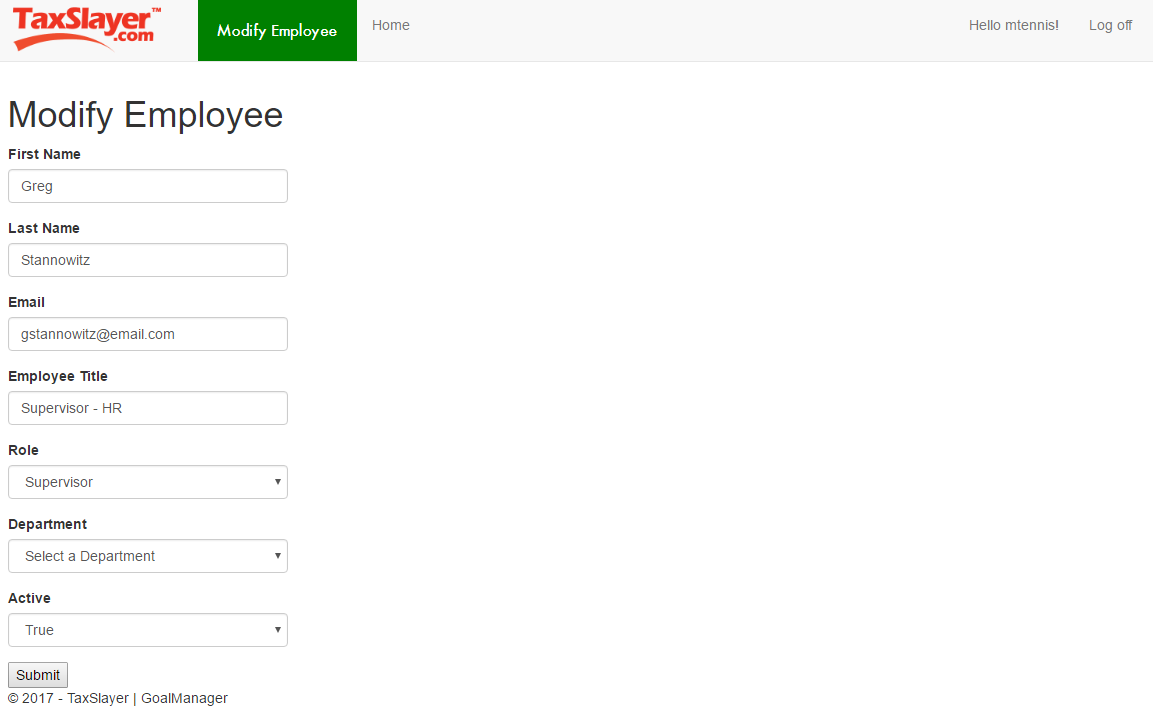


Figure 3.14. ModifyEmployee

### Modify Department

Administrators can modify existent Departments by clicking the “Modify” button attached to all Departments on AdminHome. The Department’s name, location, and description fields must be filled out. Likewise, updated Quarter data must be filled out.

ModifyDepartment is displayed in Figure 3.15.

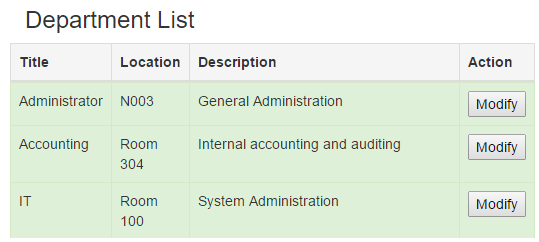
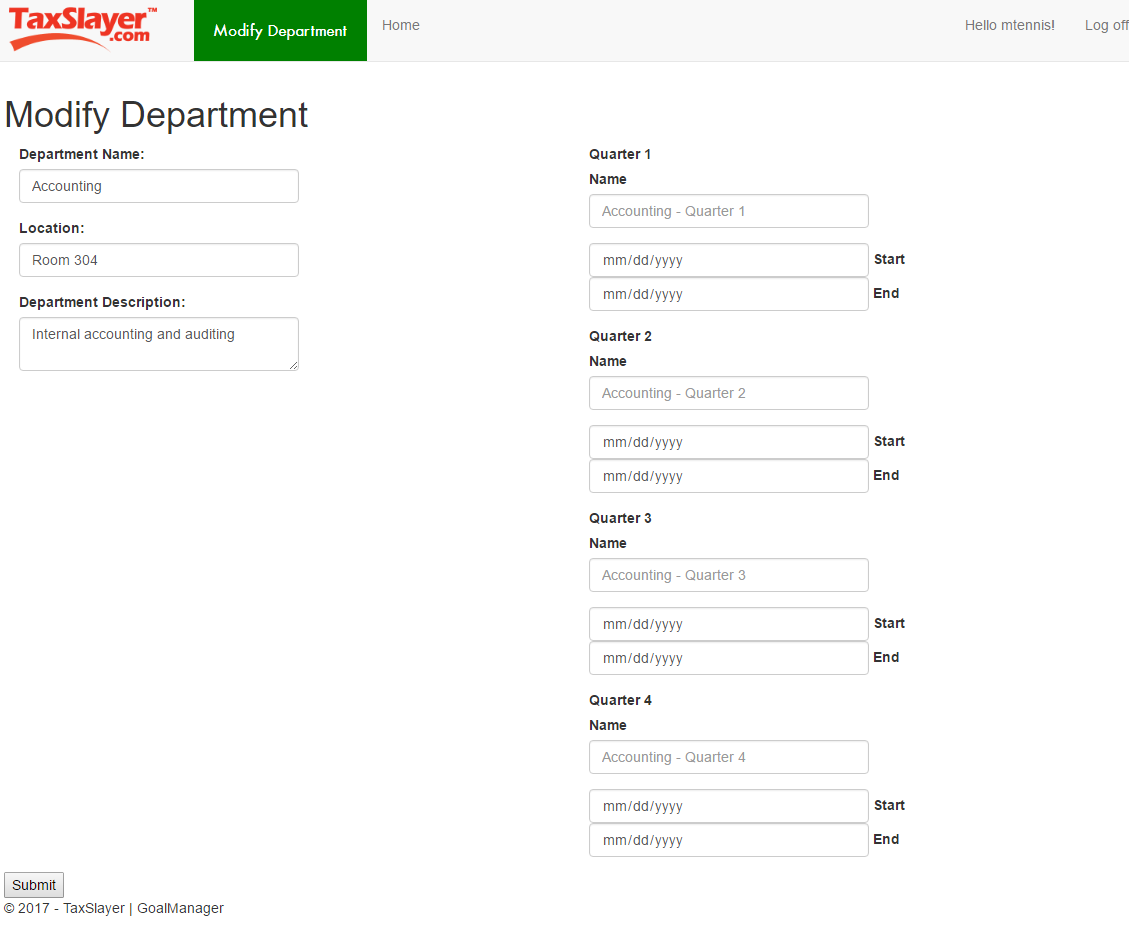


Figure 3.15. ModifyDepartment



# Database Schema

## UserDB

UserDB contains all logic relevant to the operation of GoalManager, save for login authentication. The login system and account database is handled by the C#/ASP.NET Identity solution, called IdentityDB. UserDB is implemented with Entity Framework 6.1.3 Database First model. Figure 4.1 below depicts the database schema of UserDB. The database is implemented as an embedded local SQL database with scripts written in Transact-SQL.

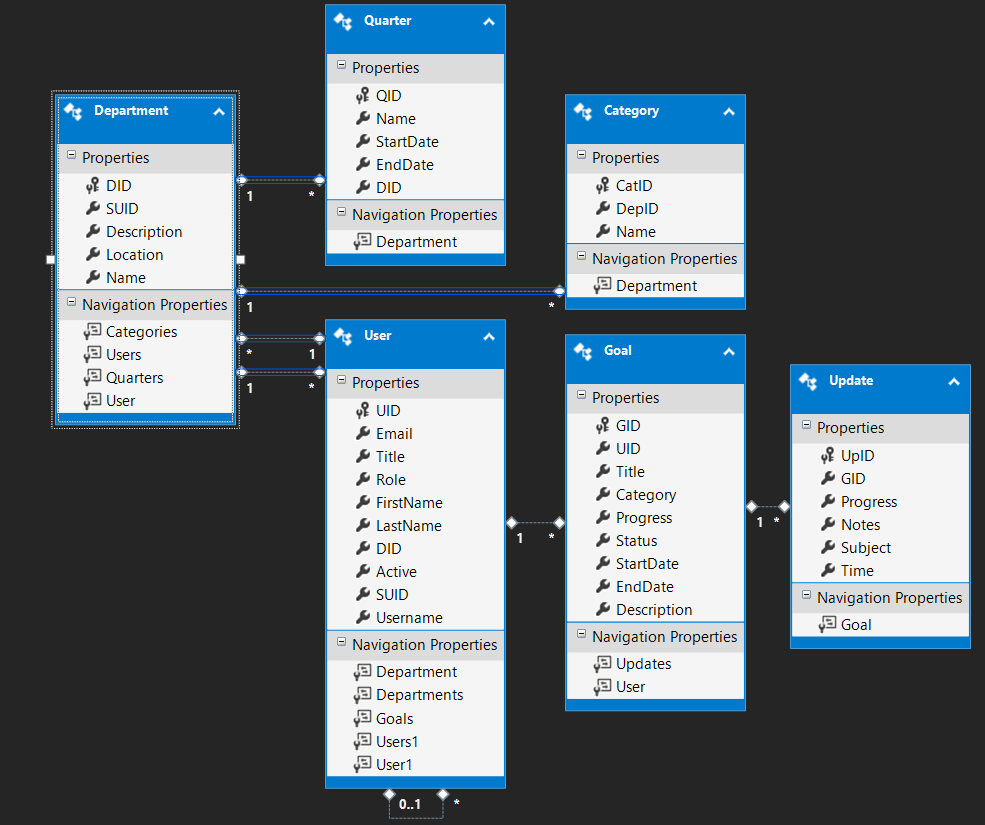


Figure 4.1. UserDB Database Schema

* 1. Departments – Associated with a User (1..n)
     1. **DID** : int – Primary key. Department’s unique identifier
     2. SUID : int – Foreign key to Users, Department Supervisor’s ID
     3. Description : nvarchar(256) – Department’s description
     4. Location : nvarchar(50) – Department’s physical location
     5. Name : nvarchar(50) – Department’s name
  2. Users – Associated with a Department (n..1), and User (0..1)
     1. **UID** : int – Primary key. User’s unique identifier
     2. Email : nvarchar(64) – Email address, used in IdentityDB.
     3. Title : nvarchar(64) – User’s title in his or her Department
     4. Role : nvarchar(16) – User’s role (Employee, Administrator, Supervisor)
     5. FirstName : nvarchar(50) – User’s first name
     6. LastName : nvarchar(50) – User’s last name
     7. DID : int – Foreign key to Departments, Department’s unique identifier
     8. Active : bit – Employee’s employment status (true or false)
     9. SUID : int – Foreign key to Users. Nullable (cascade rules). User’s Supervisor’s unique identifier.
     10. Username : nvarchar(50) – User’s username. Used in IdentityDB.
  3. Goals – Associated with a User (n..1)
     1. **GID** : int – Primary key. Goal’s unique identifier.
     2. UID : int – Foreign key to Users. This User owns this Goal.
     3. Title : nvarchar(50) – Goal’s title
     4. Category : nvarchar(50) – Goal’s Department Category (non key)
     5. Progress : int – Goal’s progress, business rule bound [0,100]
     6. Status : nvarchar(15) – Active, Completed, Denied, Failed, Pending
     7. StartDate : DateTime – Goal’s start date
     8. EndDate : DateTime – Goal’s end date
     9. Description : nvarchar(256) – Goal’s description
  4. Updates – Associated with a Goal (n..1)
     1. **UpID** : int – Primary key. Update’s unique identifier.
     2. GID : int – Foreign key to Goals. The Update’s associated Goal
     3. Progress : int – Update’s progress.
     4. Notes : nvarchar(256) – Update’s notes
     5. Subject : nvarchar(50) – Update’s subject line
     6. Time : datetime – Date of that Update posted.
  5. Quarters – Associated with a Department (n..1)
     1. **QID** : int - Primary key. Quarter’s unique identifier
     2. Name : nvarchar(50) – Quarter’s name
     3. StartDate : DateTime – Start date bound of Quarter
     4. EndDate : DateTime – End date bound of Quarter
     5. DID : Foreign key to Department
  6. Categories – Associated with a Department (n..1)
     1. **CatID** : int – Primary key. Category’s unique identifier
     2. DepID : int – Foreign key to Department
     3. Name : nvarchar(50) – Category’s name

## IdentityDB

IdentityDB is the C#/ASP.NET MVC solution for native authentication of Users. GoalManager utilizes IdentityDB as the role-based authentication mechanism of the web application. The Administrator’s CreateEmployee function (see 3.3.1) creates entries in IdentityDB utilizing the new User’s generated username and email address. These credentials are used for log in authentication and View authorization.