**IST 412 Team 5**

Sprint Document 1

**Project: Work Hour Tracker**

**Client: Jim Clarke**

**Developers: Dennis Smith**

**Sai Nithisha Guntamadugu**

**Ricky Zhao**

**Jackson Penning**

**Bijal Patel**

[**User Story Summary**](#_xnkkrfwij5z7) **3**

[**Deliverables**](#_9y0i82vyl7ak) **4**

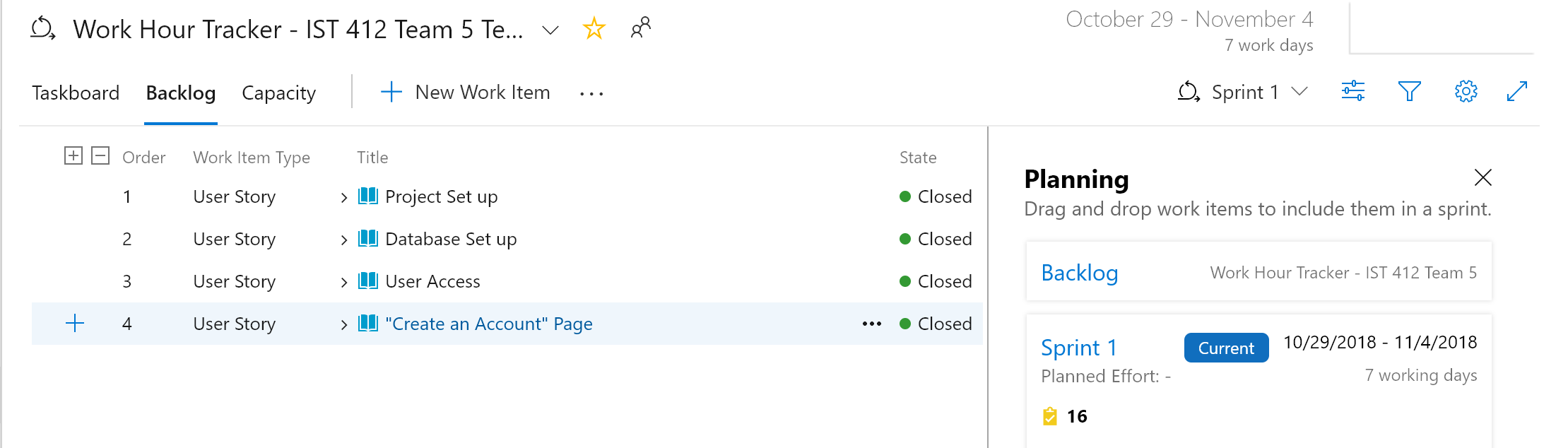
[**Retrospective**](#_4dwj4cksgw7g) **12**

[**Product Backlog Summary (Kanban board)**](#_9s92v06r7o7t) **13**

# User Story Summary

User Stories Completed

* Project Set up
* Database Set up
* User Access (Login)
* “Create an Account” Page

User Stories Started

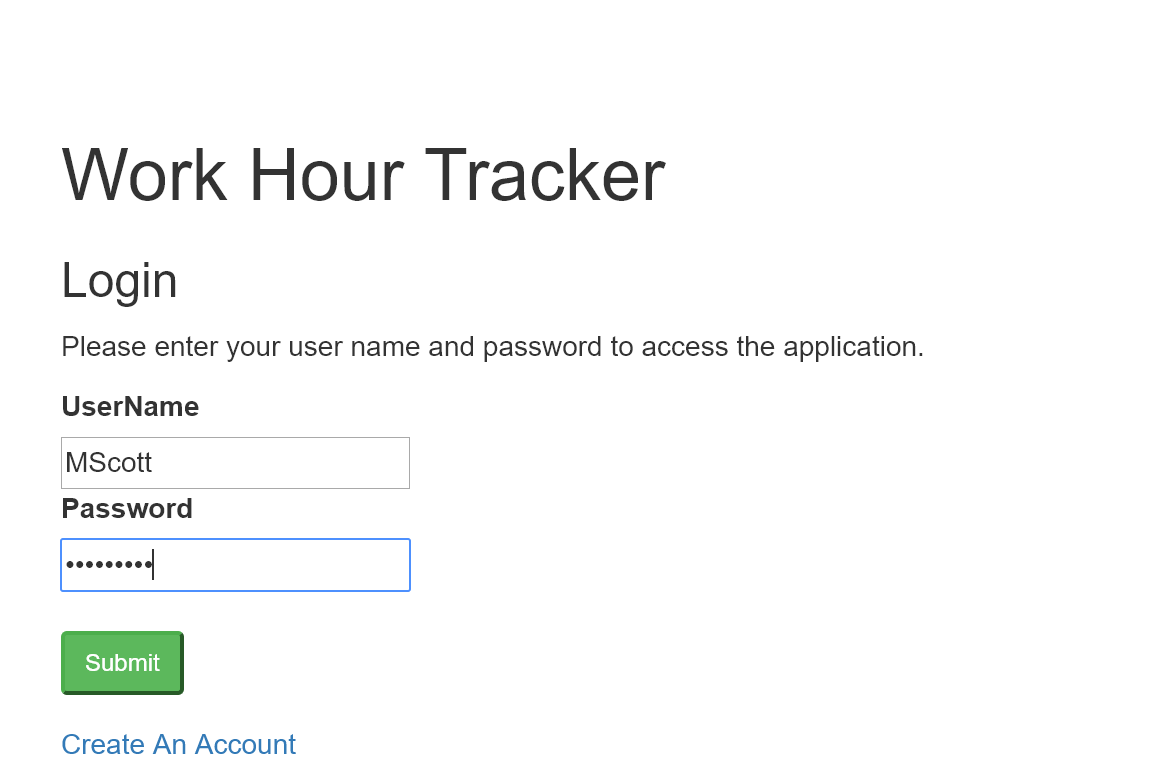
* “Create Project” Page

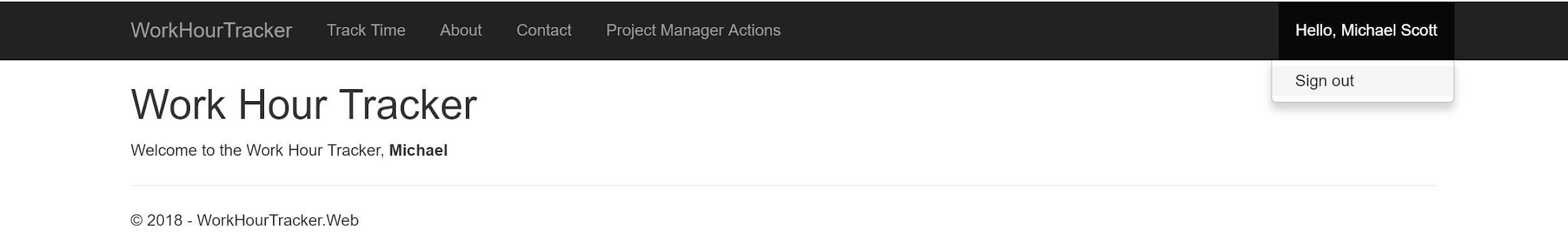
Number of Story Points Completed: 26

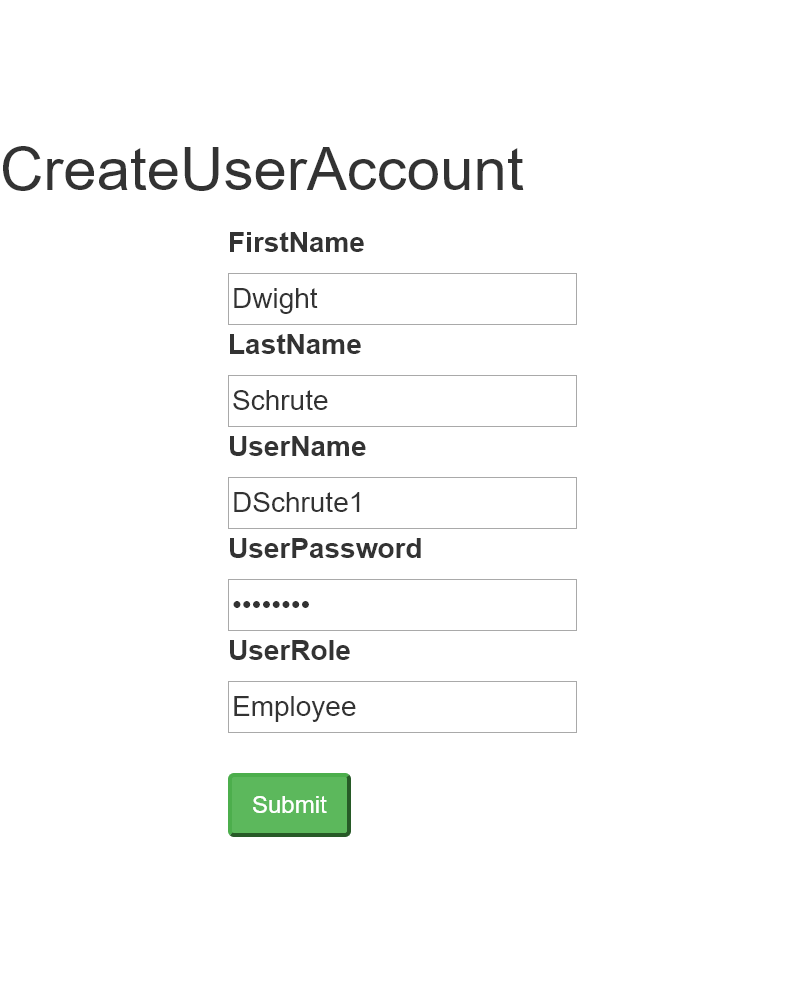
# 

# 

# Deliverables

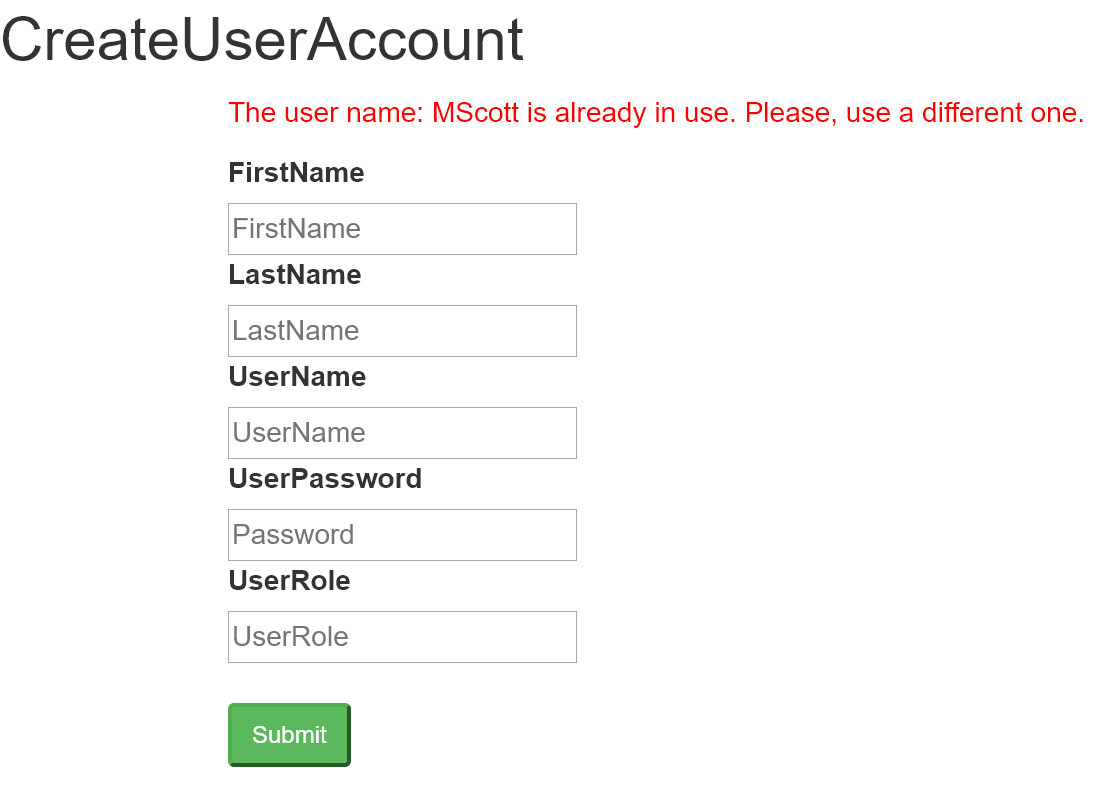
Created the login screen and developed code to allow the user to supply credentials to log into the system:

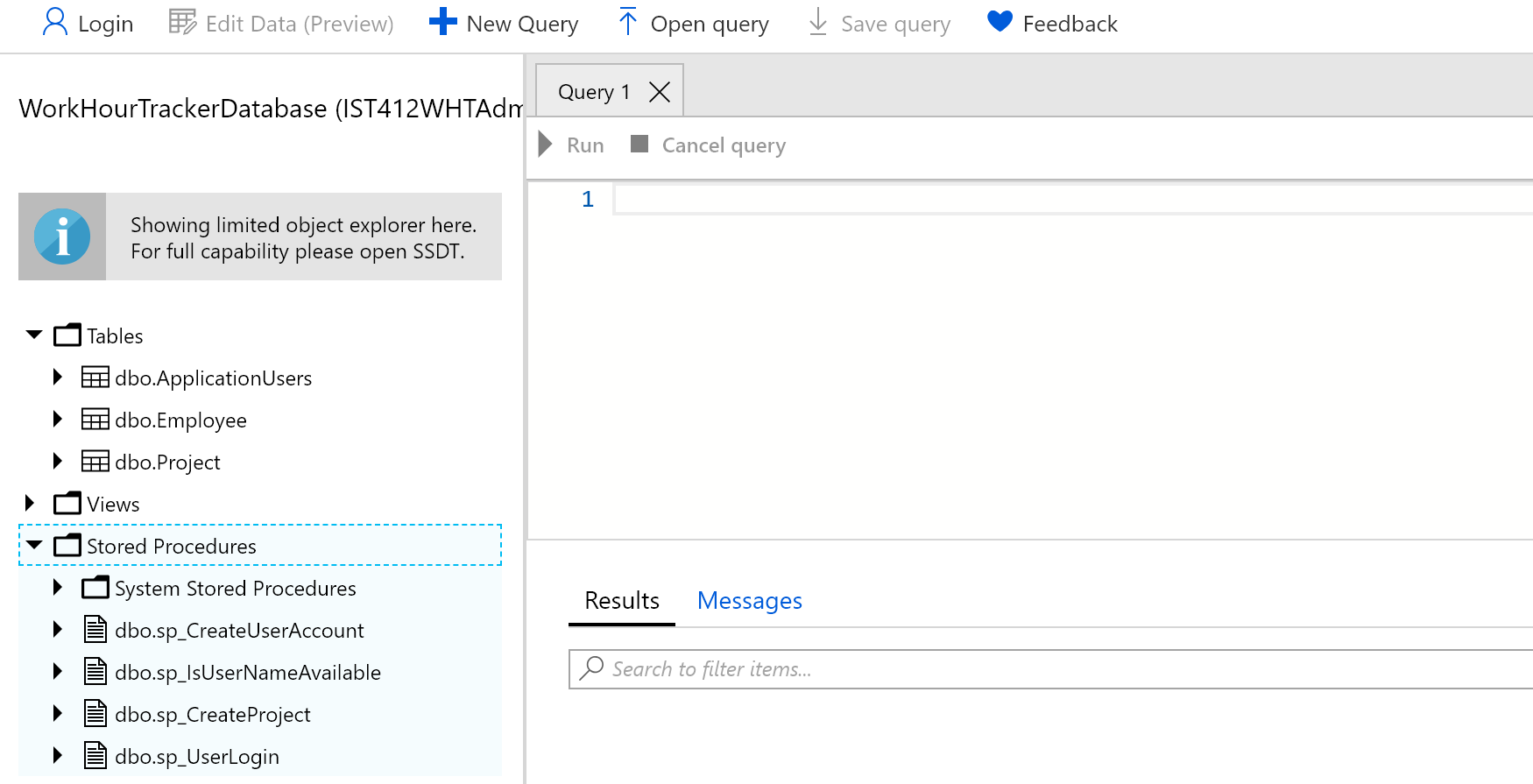
The user can successfully log into the system. The screenshot shows that the user can log out of the application and it shows pages that the team will be creating.

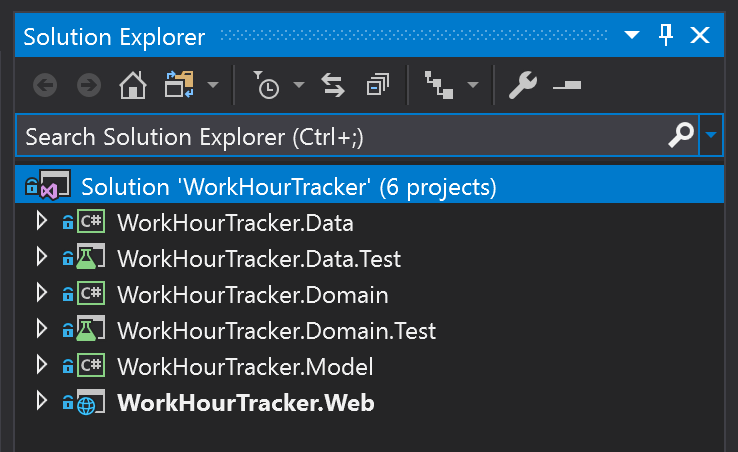
The user can create a new account:

.

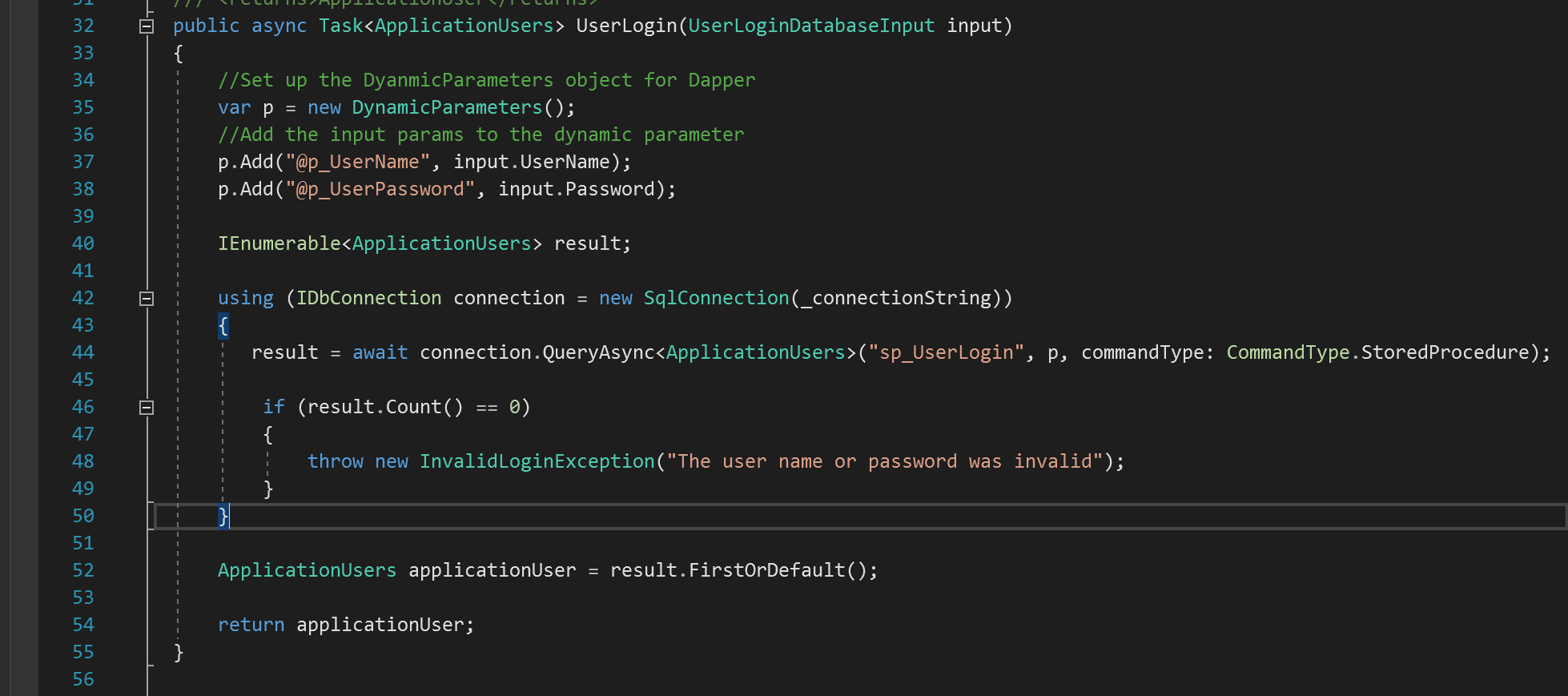
Validation on the “Create an Account” Page. The same username cannot be used in the system:

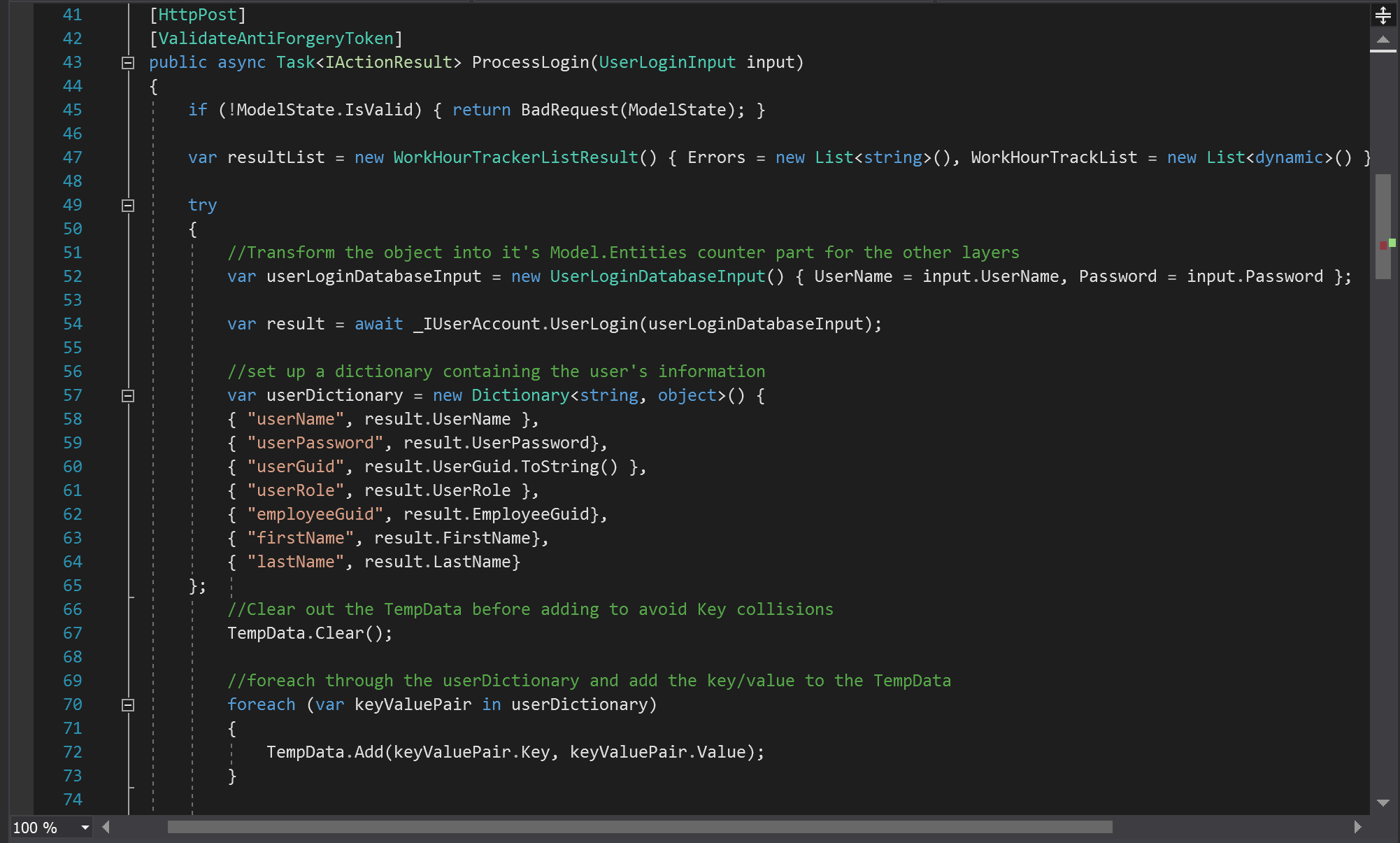


Team set up the Azure SQL Database:

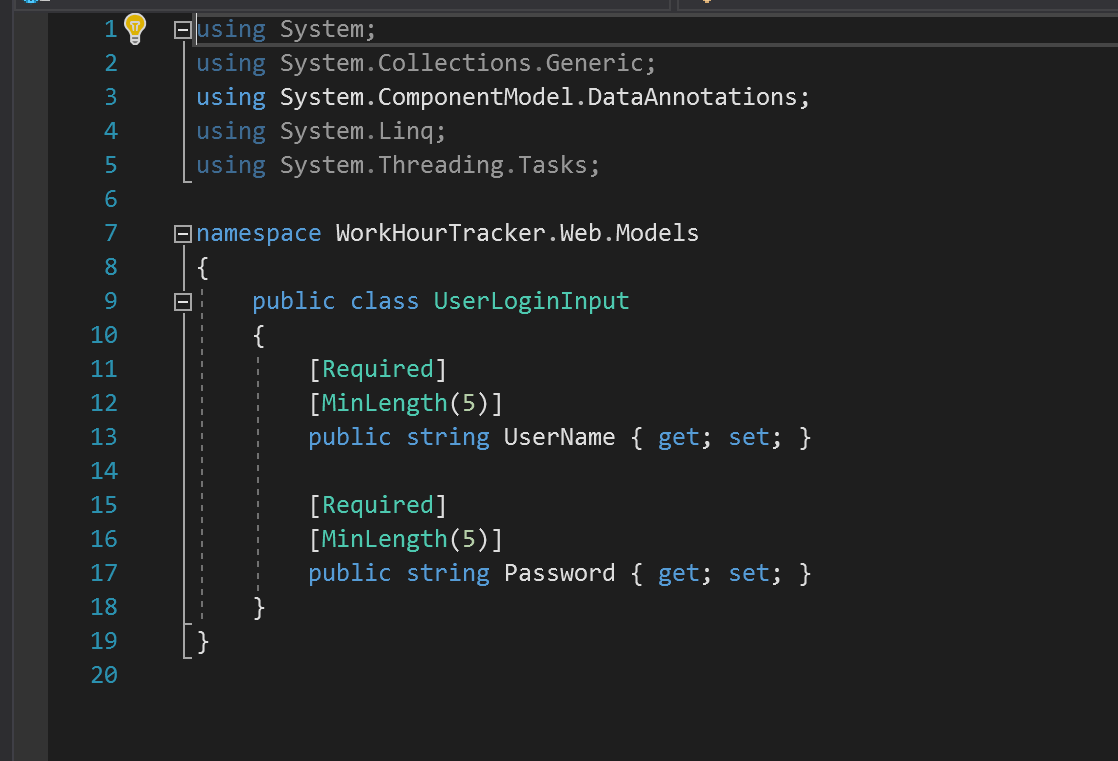
Project Framework Setup

Snippet of the Database layer Dapper Call to Azure SQL Database. This snippet shows the application using Dapper to make a call to the Azure SQL database.

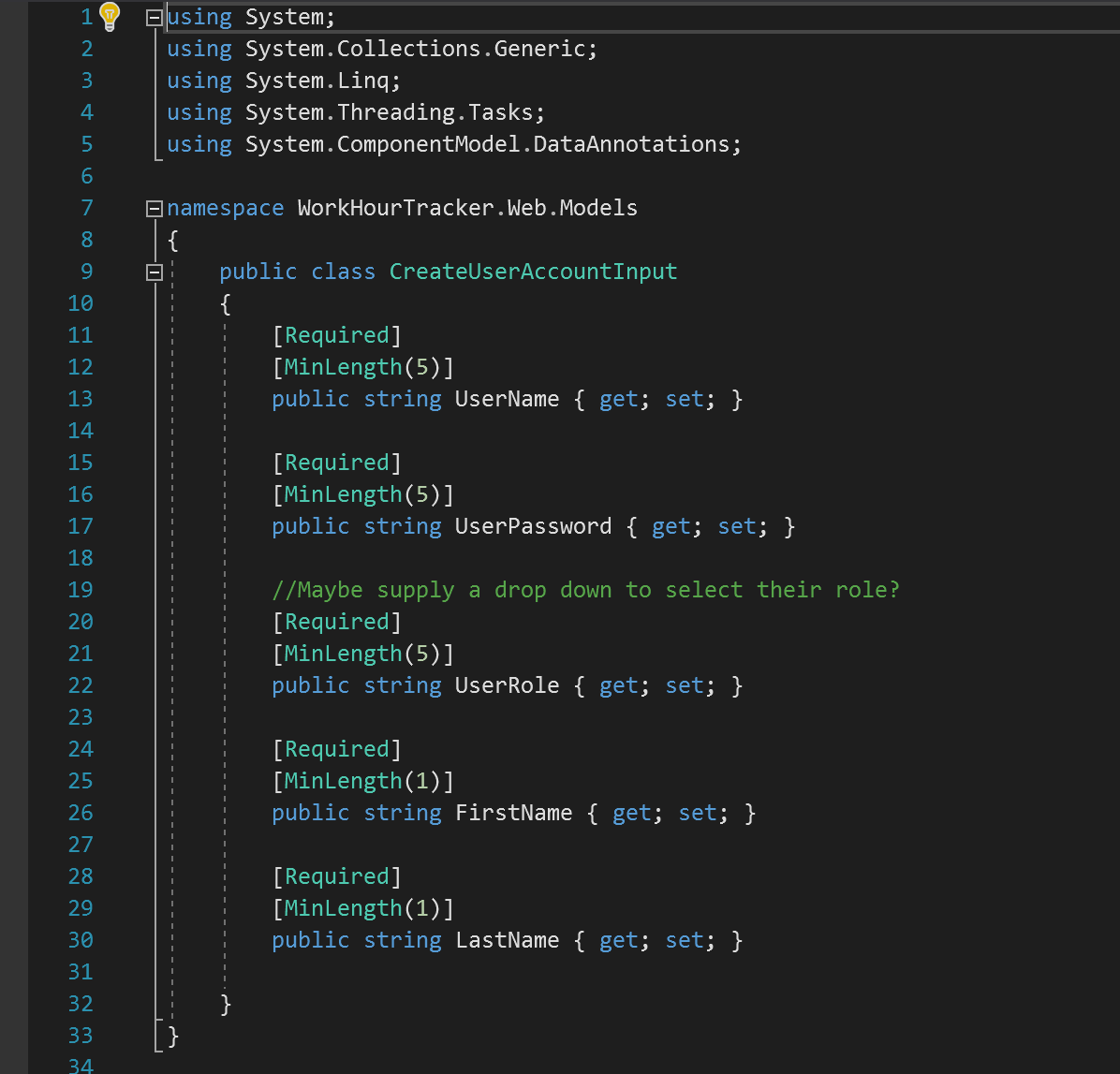


Snippet of how the code is logging a user in (using TempData). The application will take the ViewModel and if it is valid it will transform it into the UserLoginDatabaseInput class. This object gets passed to the Domain layer’s UserLogin method (by accessing the Domain layer’s IUserAccount interface) which gets passed to the data layer to send the data to the Azure SQL database. If this is successful then the application will store the user’s data in the TempData as Key/Value pairs.

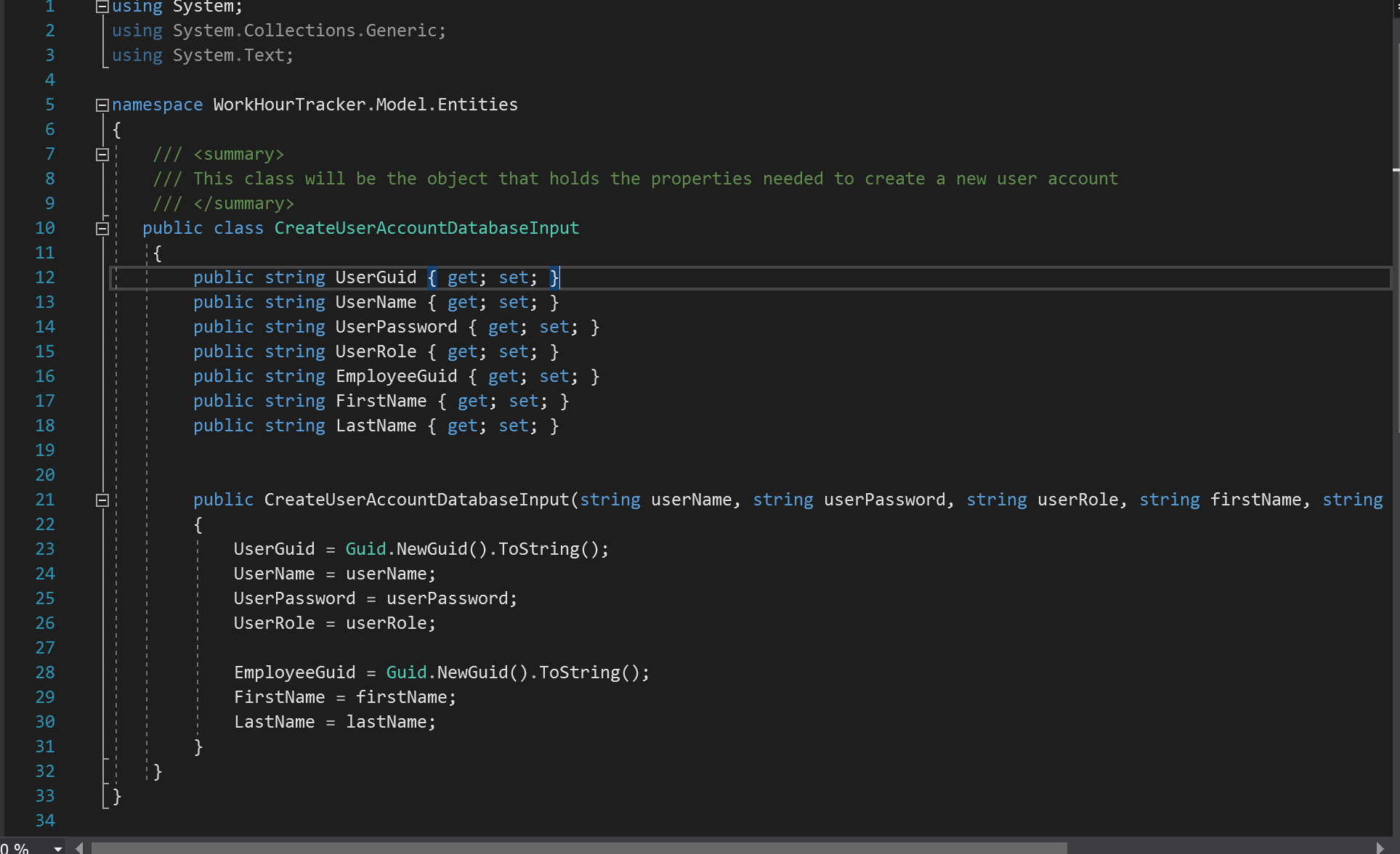
UserLoginInput ViewModel - This ViewModel is for the Login page. The ViewModel sets both fields as required and gives minLength to both fields.



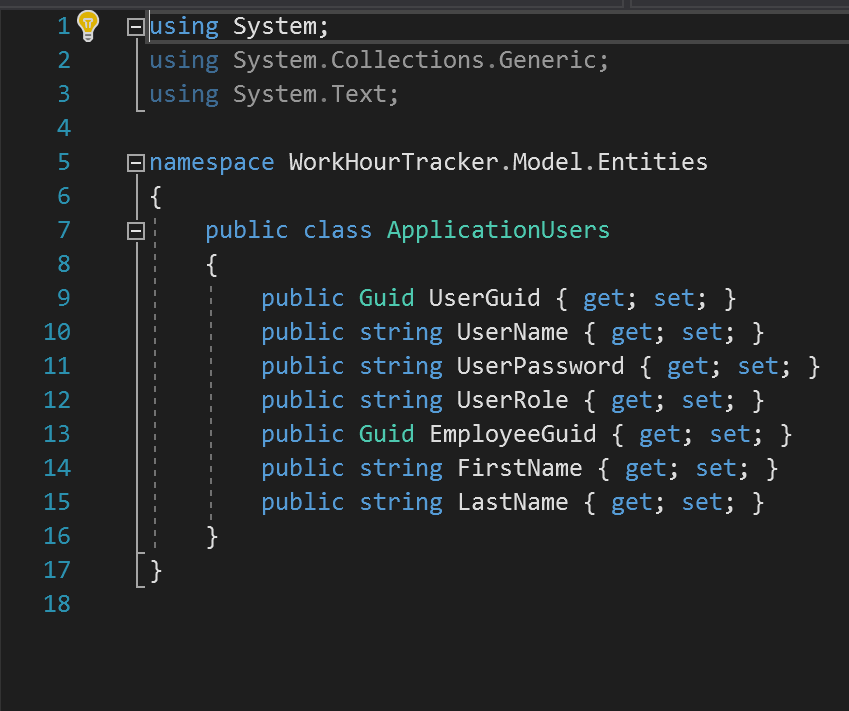
CreateUserAccountInput ViewModel - This class is the ViewModel for the “Create an Account” page. This ViewModel marks all fields as required and gives them a minLength that must be satisfied in order to be valid in the system.



CreateUserAccountDatabaseInput Model Entity - This class takes the validated ViewModel data and adds on additional fields like the UserGuid for storage in the database.



ApplcationUsers.cs - This class holds the user’s login information for use in other operations.



# Retrospective

What did the team do really well with?

The team did really well communicating with each other and making sure that we all had access to the GitHub repository.

What could the team improve upon in the next sprint?

The entire team needs to be able to clone the Git repository through Visual Studio onto their machines and push a simple update. The team is committed to getting this done early.

What lessons did you learn/observe in the last week?

The team is more familiar with the accessing the Azure Board in order to view the project backlog. Also, it was great to see the team picking up Git on Visual Studio quickly.

# Product Backlog Summary (Kanban board)

This screen shot displays the team’s Kanban board. The stories from Sprint 1 have been placed into the “Complete” column. The other stories are either in “Requirements Approval” (awaiting client’s approval) or “User Stories” (the team is taking the client’s need and translating it into a User Story).