

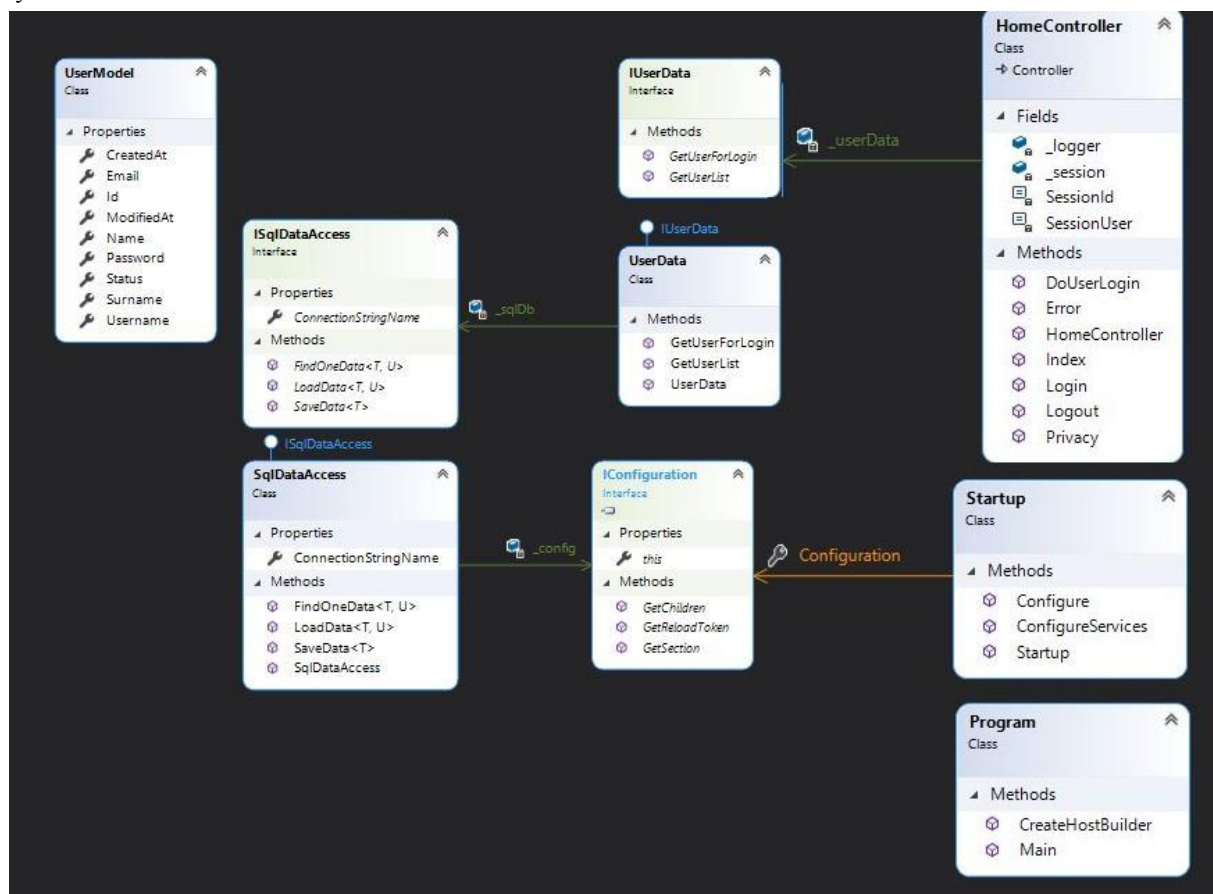
Design Document

Revision History

Date	Author(s)	Changes	Version
03/05/2021	Özgün Şen Berfu Anıl Esra Ateş Nehir Erdem	initial document	v1.0

Design structure

The design of the software should follow the decisions defined in the Architecture notebook document. The system's structure and the model classes are shown below.



Stock Management	v1.0.1
Design Document	Date: 03/05/2021

Patterns

Controller Pattern

Overview

A controller handles all incoming requests from the user interface and controls the data flow.

Structure

The controller creates any models needed by request, does necessary work, then prepares the response data for the view. Views are generated using this response data. The output of the view is returned to the client to be shown on the browser to the User.

Singleton Pattern

Overview

The singleton pattern is a software design pattern that restricts the instantiation of a class to one "single" instance. This is useful when exactly one object is needed to coordinate actions across the system.

Structure

The singleton pattern has a lot of advantages for common artefacts that can be used in the system. When it comes to Database Connection classes, the singleton pattern can be very useful. We can monitor load balancing, unnecessary connections, and shared db connection management by managing Db Connections through a single case.

Abstract Factory Pattern

Overview

Abstract Factory Pattern says that just define an interface or abstract class for creating families of related objects but without specifying their concrete sub-classes.

Structure

User data and SQL database access classes have their own interfaces that create a factory pattern of their own.

Requirement realizations

Verification of all input data will be made with the data annotations (property attributes written on the head of each property of each class) written on the domain classes.

Stock Management	v1.0.1
Design Document	Date: 03/05/2021

UC1.2: System Sequence Diagram

