

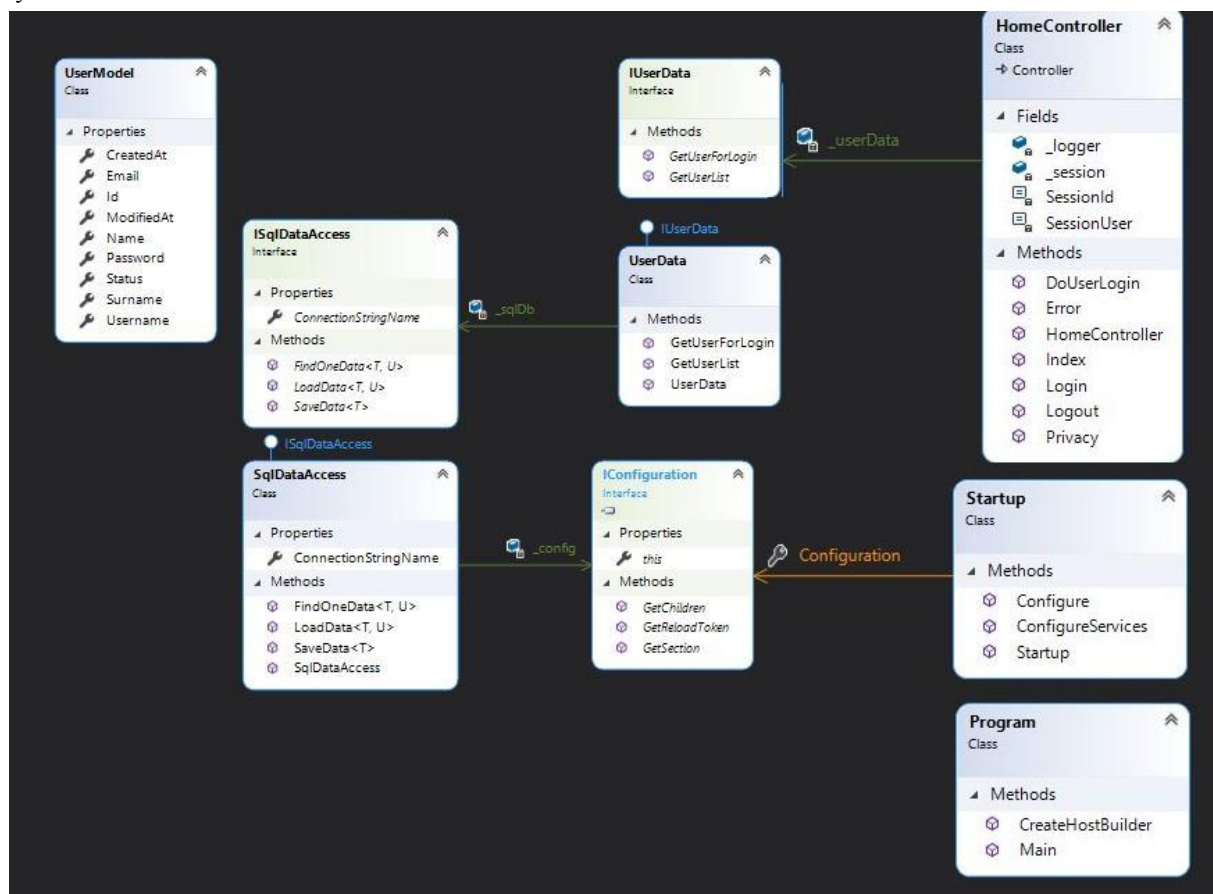
# Design Document

## 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

For example in our project the HomeController is the connection between the UserData model and the Index and Login views. It listens for events prompted by the Actor from views and responds appropriately. Login process occurs by getting data from view to controller then sending it to properly by model to the Data Access layer.

### 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

In our project we create ISqlDataAccess and IUserData as abstract interfaces. SqlDataAccess and UserData classes are the concrete classes.

## 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

