# Use-Case

|  |  |
| --- | --- |
| Название | Вход в систему |
| Основное действующее лицо (актор) | Неавторизованный пользователь |
| Описание | Пользователь указывает логин-пароль и нажимает на кнопку вход. Система проверяет введённый логин-пароль и авторизует пользователя. |
| Триггер | Запуск программы или выход из аккаунта. |
| Предварительные условия | Пользователь не должен быть авторизирован в системе.  База данных должна быть доступна. |
| Выходные условия | Выходные условия отсутствуют. |
| Нормальное направление развития | Вход в систему:   1. Пользователь указывает логин и пароль и нажимает войти. 2. Система проверяет логин и пароль. 3. Пользователь может начать работать с системой. |
| Альтернативное направление развития | Вход в систему после 3 и более неудачных попыток:   1. Пользователь указывает логин и пароль и нажимает войти. 2. Система показывает окно с капчей и полем ввода 3. Пользователь вводит капчу. 4. Система проверяет логин, пароль и капчу. 5. Пользователь может начать работать с системой. |
| Исключения | Логин или пароль не совпадают:   1. Система показывает окно с сообщением об ошибке. 2. Работа начинается заново с шага 1. |

Требования ТЗ:

- Пароль более 8 символов

- Пароль должен содержать как минимум один специальный символ.

- Пароль должен содержать как минимум одну цифру.

- Пароль должен содержать как минимум одну букву.

- Появление Capcha после 3 неверных ввода.

# Классы-заглушки для тестов

class FakeShowMainWindowService : IShowMainWindowService

{

public int CountExecute { get; private set; } = 0;

public void ShowWindow(User user) =>

CountExecute++;

}

class FakeRequestCaptchaService : IRequestCaptchaService

{

public int CountExecute { get; private set; } = 0;

public bool Answer { get; set; } = true;

public bool RequestCaptcha()

{

CountExecute++;

return Answer;

}

}

class FakeCapthaGenerator : ICapthaGenerator

{

public bool IsValidCaptcha { get; set; } = true;

public int CountExecuteNewCapthca { get; set; } = 0;

public int WidhtImage { get; set; }

public int HeightImage { get; set; }

public ImageSource CapthaImage { get; }

public bool CheckCaptcha(string captcha)

{

return IsValidCaptcha;

}

public void NewCaptcha()

{

CountExecuteNewCapthca++;

}

}

# Тесты

## Test Case 1. Не введен логин.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь не ввел логин

Ожидаемый результат: Ошибка «Заполните все поля».

public async Task CheckEmptyLogin()

{

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

viewModel.Login = "";

viewModel.Password = "password";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

Assert.AreEqual(viewModel.ErrorMessage, "Заполните все поля");

}

## Test Case 2. Не введен пароль.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь не ввел логин

Ожидаемый результат: Ошибка «Заполните все поля».

public async Task CheckEmptyPassword()

{

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

viewModel.Login = "login";

viewModel.Password = "";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

Assert.AreEqual(viewModel.ErrorMessage, "Заполните все поля");

}

## Test Case 3. Верный логин и пароль.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит верный логин
2. Пользователь вводит верный пароль

Ожидаемый результат: Успешная авторизация

public async Task ValidCredentials()

{

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

viewModel.Login = "d.popova";

viewModel.Password = "password";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

Assert.IsTrue(string.IsNullOrEmpty(viewModel.ErrorMessage));

Assert.AreEqual(\_showMainWindowService.CountExecute, 1);

}

## Test Case 4. Неверный логин.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит неверный логин.
2. Пользователь вводит пароль.

Ожидаемый результат: Ошибка «Неверный логин или пароль».

public async Task InvalidLogin()

{

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

viewModel.Login = "invalid";

viewModel.Password = "password";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

Assert.AreEqual(viewModel.ErrorMessage, "Неверный логин или пароль");

Assert.IsTrue(string.IsNullOrEmpty(viewModel.Password));

Assert.AreEqual(\_requestCaptchaService.CountExecute, 0);

}

## Test Case 5. Неверный пароль.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит верный логин.
2. Пользователь вводит неверный пароль.

Ожидаемый результат: Ошибка «Неверный логин или пароль»

[Test]

public async Task InvalidPassword()

{

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

viewModel.Login = "d.popova";

viewModel.Password = "invalid";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

Assert.AreEqual(viewModel.ErrorMessage, "Неверный логин или пароль");

Assert.IsTrue(string.IsNullOrEmpty(viewModel.Password));

Assert.AreEqual(\_showMainWindowService.CountExecute, 0);

}

## Test Case 6. Пользователь ввёл неверный пароль 3 и более раз.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит неверный логин или пароль.
2. Повторить шаг первый 4 раза.

Ожидаемый результат: После 4 попытки показывается окно с предложением ввода капчи.

[Test]

public async Task InvalidPassword3AndMore()

{

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

for (int i = 0; i < 5; i++)

{

viewModel.Login = "d.popova";

viewModel.Password = $"invalidpassword{i}";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

}

Assert.AreEqual(\_requestCaptchaService.CountExecute, 2);

}

## Test Case 7. Пользователь отказался от ввода капчи.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит неверный логин или пароль.
2. Повторить шаг первый 4 раза.
3. При появлении окна с вводом капчи отказаться от ее ввода.

Ожидаемый результат: Ошибка «Необходим ввод капчи». Логин и пароль не проверяется на правильность.

[Test]

public async Task CaptchaCancel()

{

\_requestCaptchaService.Answer = false;

var viewModel = new AuthorizeViewModel(\_db, \_showMainWindowService, \_requestCaptchaService);

for (int i = 0; i < 4; i++)

{

viewModel.Login = "d.popova";

viewModel.Password = $"invalidpassword{i}";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

}

\_requestCaptchaService.Answer = false;

viewModel.Login = "d.popova";

viewModel.Password = $"password";

await viewModel.CheckCredentialsCommand.ExecuteAsync(null);

Assert.AreEqual(viewModel.ErrorMessage, "Ошибка капчи");

}

## Test Case 8. Пользователь ввел неправильную капчу.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит неверный логин или пароль.
2. Повторить шаг первый 4 раза.
3. При появлении окна с вводом капчи пользователь вводит неверную капчу.

Ожидаемый результат: Ошибка «Капча неверна». Логин и пароль не проверяется на правильность.

[Test]

public void InvalidCaptcha()

{

\_capthaGenerator.IsValidCaptcha = false;

var viewModel = new CaptchaViewModel(\_capthaGenerator);

viewModel.CheckCaptchaCommand.Execute(null);

Assert.IsFalse(viewModel.IsAccept);

}

## Test Case 9. Пользователь ввел правильную капчу.

Пошаговое описание действий, необходимых для проверки:

1. Пользователь вводит неверный логин или пароль.
2. Повторить шаг первый 4 раза.
3. При появлении окна с вводом капчи пользователь вводит верную капчу.

Ожидаемый результат: Логин и пароль проверяется на правильность.

[Test]

public void ValidCaptcha()

{

\_capthaGenerator.IsValidCaptcha = true;

var viewModel = new CaptchaViewModel(\_capthaGenerator);

viewModel.CheckCaptchaCommand.Execute(null);

Assert.IsTrue(viewModel.IsAccept);

}

# Матрица требований

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TC1 | TC2 | TC3 | TC4 | TC5 | TC6 | TC7 | TC8 | TC9 |
| Логин не введен | + |  |  |  |  |  |  |  |  |
| Пароль не введен |  | + |  |  |  |  |  |  |  |
| Верный логин и пароль |  |  | + |  |  |  |  |  |  |
| Неверный логин |  |  |  | + |  | + | + | + | + |
| Неверный пароль |  |  |  |  | + | + | + | + | + |
| Неверный логин или пароль введены 3 и более раз |  |  |  |  |  | + |  |  |  |
| Капча не введена |  |  |  |  |  | + | + |  |  |
| Капча введена верно |  |  |  |  |  | + |  | + |  |
| Капча введена не верно |  |  |  |  |  | + |  |  | + |

# Чек-лист

|  |  |  |
| --- | --- | --- |
| № | Заголовок | Ожидаемый результат |
| 1 | Не введен логин | Ошибка «Заполните все поля». |
| 2 | Не введен пароль | Ошибка «Заполните все поля». |
| 3 | Верный логин и пароль | Успешная авторизация |
| 4 | Неверный логин | Ошибка «Неверный логин или пароль» |
| 5 | Неверный пароль | Ошибка «Неверный логин или пароль» |
| 6 | Пользователь ввёл неверный пароль 3 и более раз | После 4 попытки показывается окно с предложением ввода капчи. |
| 7 | Пользователь отказался от ввода капчи | Ошибка «Необходим ввод капчи». Логин и пароль не проверяется на правильность. |
| 8 | Пользователь ввел неправильную капчу | Ошибка «Капча неверна». Логин и пароль не проверяется на правильность. |
| 9 | Пользователь ввел правильную капчу | Логин и пароль проверяется на правильность |