

Test Case on Registration

Requirements for Doctor:

- Unique Username (at least 5 characters)
- At least 8 characters for password
- Valid email (@ a valid email)
- Age (18 and older)
- Contact Number (at least 8 numbers)

Black Box Testing (Registration) Table

Test Inputs					Expected Output	Actual Output
Username	Password	Email	Age	Contact Number		
abcd	password123	abcd@gmail.com	40	91234567	Username must be between 5 to 50 characters	Username must be between 5 to 50 characters
abcdefghijklm nopqrst vwxyz bcdefghijk lmnopqrst vwxyz	password123	abcd@gmail.com	40	91234567	Username must be between 5 to 50 characters	Username must be between 5 to 50 characters
abcde	p123	abcd@gmail.com	40	91234567	The password cannot be shorter than 8 characters	The password cannot be shorter than 8 characters
abcde	password123	aagmail.com	40	91234567	Please put a proper email. E.g. abc@gmail.com	Please put a proper email. E.g. abc@gmail.com
abcde	password123	abcd@gmail.com	17	91234567	You must be at least 18 years old to register.	You must be at least 18 years old to register.
abcde	password123	abcd@gmail.com	40	9999	Please enter a proper phone number. E.g. 91234567	Please enter a proper phone number. E.g. 91234567
abcde	password123	abcd@gmail.com	40	91234567	No error, successfully registers	No error, successfully registers

Black Box Testing

(Doctor) Password: Number of characters

Equivalence classes (Doctor):

- Valid range: Password ≥ 8
- Invalid range: Password ≤ 7

Valid Boundary values (Doctor):

- 8
- Value just below the boundary: 7

Invalid Boundary values (Doctor):

- 7
- Value just above the boundary: 8
- Value just below the boundary: 6

(Doctor) Username: Must be unique

Equivalence classes (Doctor):

- Valid range:
 - Username not in system database
 - Username ≥ 5 characters
 - Username ≤ 50 characters
- Invalid range:
 - Username in system database
 - Username ≤ 4 characters
 - Username ≥ 51 characters

Valid Boundary values (Doctor):

- Username not in system database
- 5, 50
- Value just below the boundary: 4
- Value just above the boundary: 51

Invalid Boundary values (Doctor):

- Username in system database
- 4, 51
- Value just above the boundary: 52
- Value just below the boundary: 3

(Doctor) Age: Older than 18-year old

Equivalence classes (Doctor):

- Valid range: Current Date - Date of Birth ≥ 18
- Invalid range: Current Date - Date of Birth ≤ 17

Valid Boundary values (Doctor):

- 18
- Value just below the boundary: 17

Invalid Boundary values (Doctor):

- 17
- Value just above the boundary: 18
- Value just below the boundary: 16

(Doctor) Contact Number: at least 8 numbers

Equivalence classes (Doctor):

- Valid range: Contact Number ≥ 8
- Invalid range:
 - Contact Number ≤ 7

Valid Boundary values (Doctor):

- 8
- Value just above the boundary: 9
- Value just below the boundary: 7

Invalid Boundary values (Doctor):

- 7
- Value just above the boundary: 8
- Value just below the boundary: 6

(Doctor) Email: Must have a proper email (@gmail.com / @hotmail.com / etc)

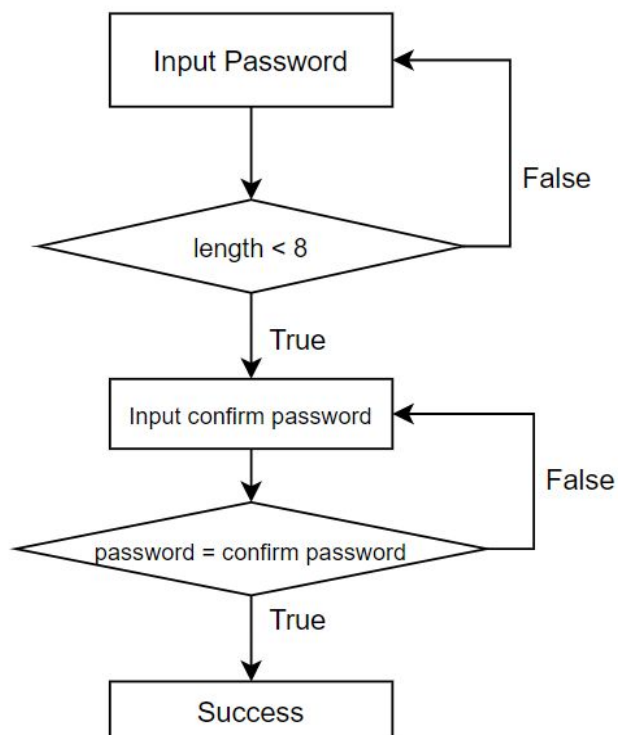
Equivalence classes (Doctor):

- Valid range: contain @something.com
- Invalid range: does not contain @something.com

White Box Testing (Registration)

(Doctor) Password: Number of characters

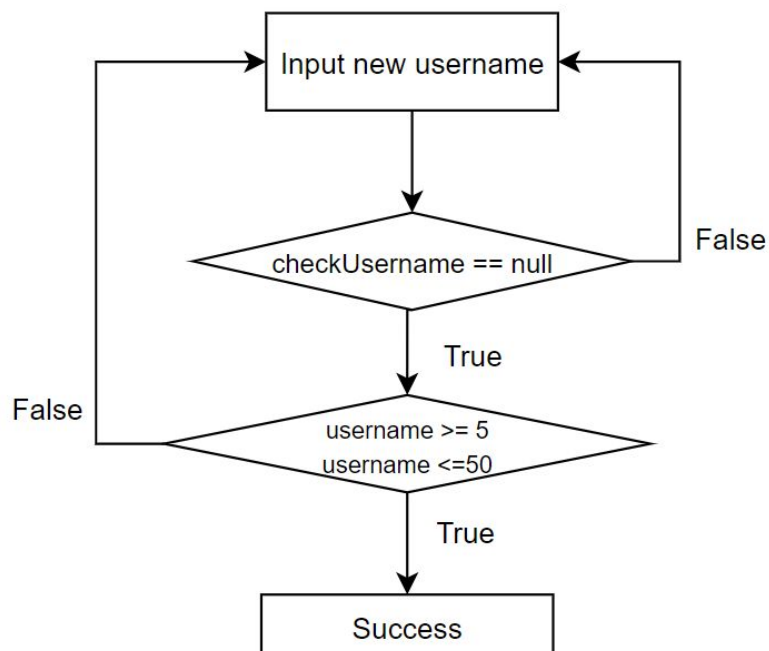
```
[Required]
[StringLength(100, MinimumLength = 8, ErrorMessage = "The password cannot be shorter than 8 characters")]
[DisplayName("Password")]
41 references
public string password { get; set; }
[NotMapped]
[Required]
[DisplayName("Confirm Password")]
[System.ComponentModel.DataAnnotations.Compare("password")]
34 references
public string confirmPassword { get; set; }
```



(Doctor) Username: Must be unique

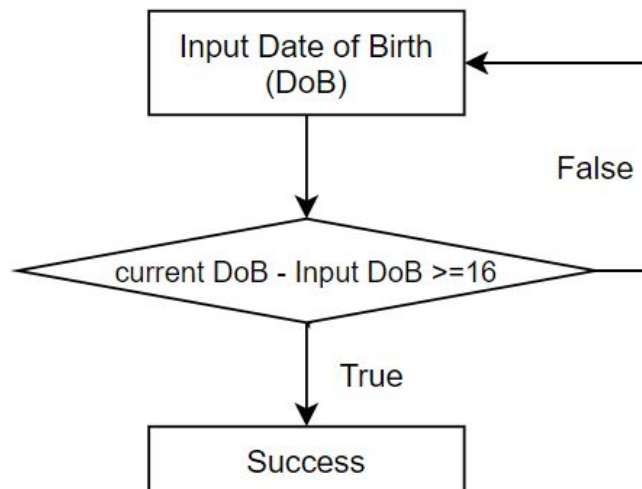
```
if (checkUsername == null)
{
    if (checkEmail == null)
    {
        //Set details
        accountInfo.GeneralUsers.password = GetMD5(accountInfo.GeneralUsers.password);
        accountInfo.GeneralUsers.profilePicture = img;
        accountInfo.InterestedParties.userName = accountInfo.GeneralUsers.userName;
        accountInfo.InterestedParties.bio = "";

        //Insert into database
        db.Configuration.ValidateOnSaveEnabled = false;
        db.GeneralUser.Add(accountInfo.GeneralUsers);
        db.InterestedParty.Add(accountInfo.InterestedParties);
        db.SaveChanges();
        return RedirectToAction("SignUpSuccessMember");
    }
    else
    {
        ViewData["EmailExists"] = "Email already exists";
    }
}
else
{
    ViewData["UsernameExists"] = "Username already exists";
}
```



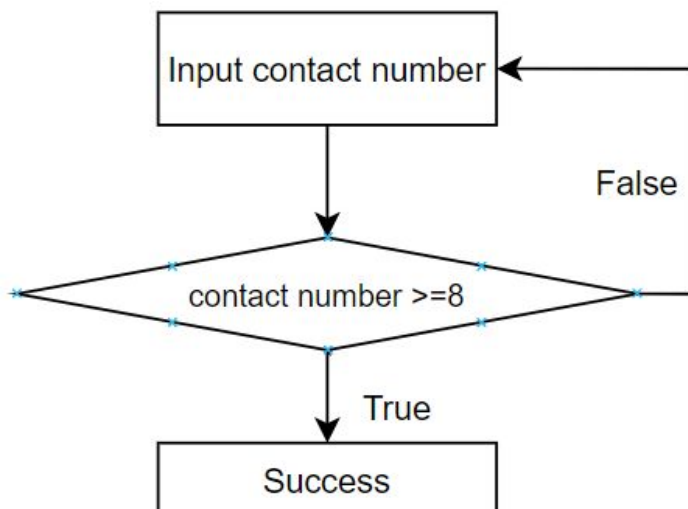
(Doctor) Age: Older than 16-year old

```
[Required]
[DisplayName("Date of Birth")]
[Range(typeof(DateTime), "1/1/1900", "31/12/2002",
    ErrorMessage = "You must be at least 16 years old to register.")]
[DataType(DataType.Date)]
```



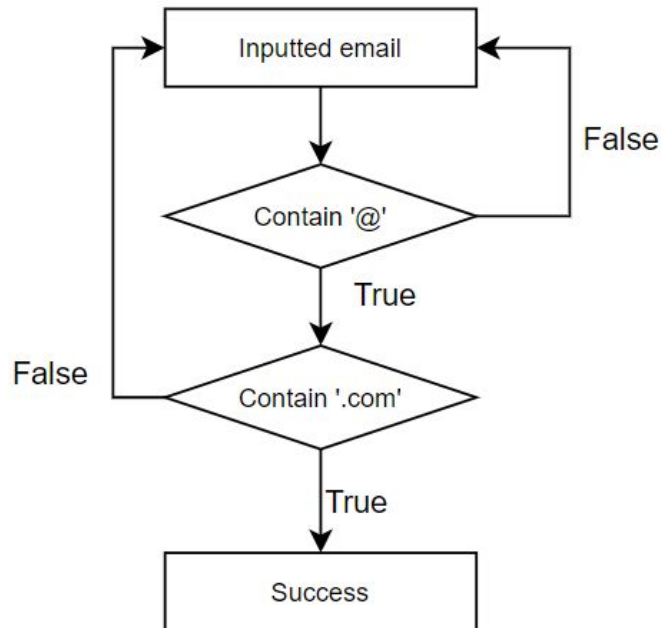
Contact Number: at least 8 numbers

```
[Required(ErrorMessage = "Please enter a proper phone number. E.g. 91234567")]
[DisplayName("Contact Number")]
[Range(10000000, 99999999, ErrorMessage = "Please enter a proper phone number. E.g. 91234567")]
20 references
public int contactNo { get; set; }
```

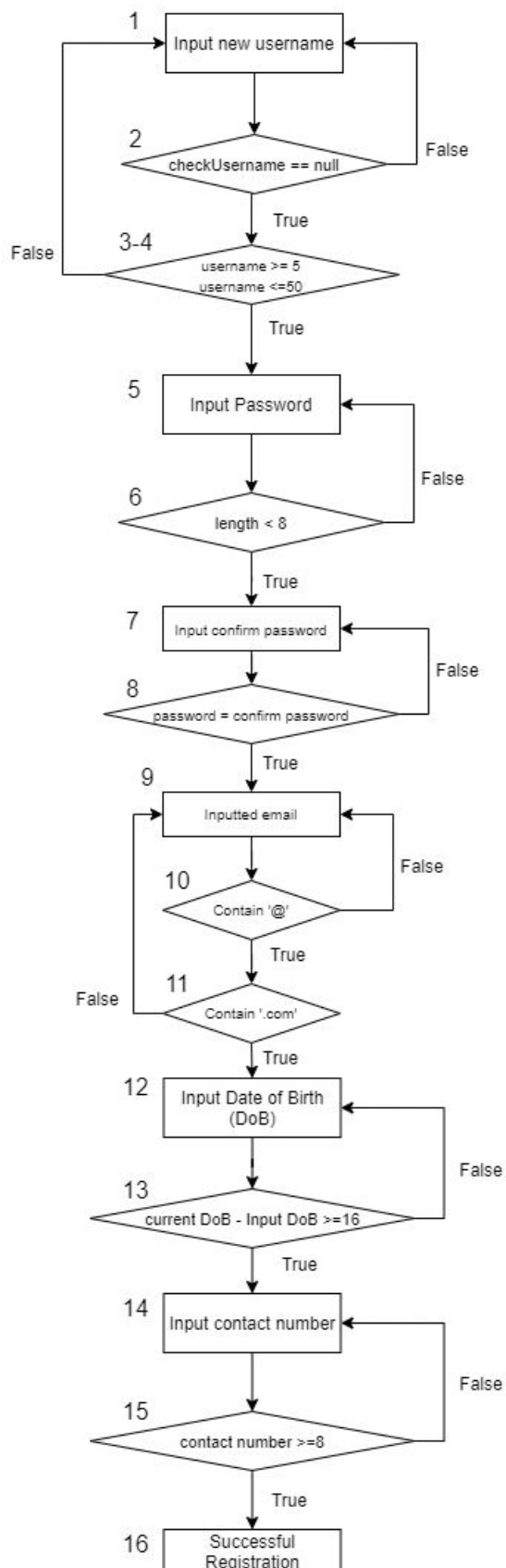


Email: contain '@gmail.com' / '@hotmail.com' / etc

```
[Required]
[RegularExpression(@"[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}", ErrorMessage = "Please put a proper email. E.g. abc@gmail.com")]
[DisplayName("Email")]
44 references
public string email { get; set; }
```



Overall Flow Chart



Baseline Path: 1,2,3-4,5,6,7,8,9,10,11,12,13,14,15,16 [one straight line downwards]

Requirements for login (Doctor):

- Correct username
- Correct password
- Approved by admin

Black Box Testing (Login) Table

Test Inputs		Expected Output	Actual Output
Username	Password		
		Please fill out this field	Please fill out this field
	password123	Please fill out this field	Please fill out this field
abcde		Please fill out this field	Please fill out this field
abcd	pass123	Wrong username/password, please try again	Wrong username/password, please try again
abcd	password123	Wrong username/password, please try again	Wrong username/password, please try again
abcde	passw0rd12	Wrong username/password, please try again	Wrong username/password, please try again
abcde	password123	No error. Successful Login	No error. Successful Login

(Doctor) Login: Verify username

Equivalence classes:

- Valid: username = data username
- Invalid: username != data username

(Doctor) Login: Verify password

Equivalence classes:

- Valid: password = data password
- Invalid: password != data password

(Doctor) Login: Verify account is approved by admin

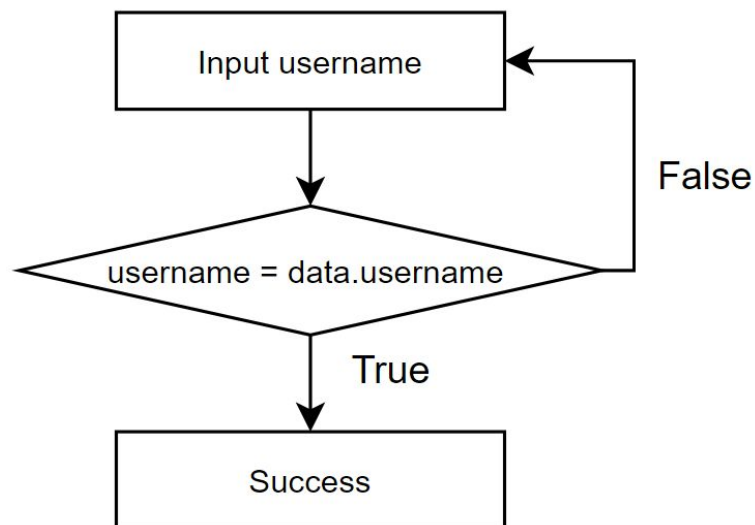
Equivalence classes:

- Valid: Admin Verify = True
- Invalid: Admin Verify = False

White Box Testing (Login)

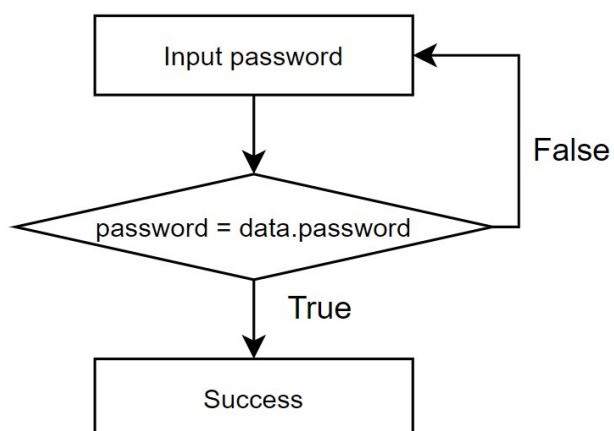
(Doctor) Login: Verify username

```
//Else if interested party account  
else  
{  
    //Store username in session  
    Session["username"] = data.FirstOrDefault().userName;  
    Session["docOrMember"] = "member";  
    //Set error messages to null  
    TempData["loginFailed"] = "";  
    return RedirectToAction("Stories", "Story");  
}
```



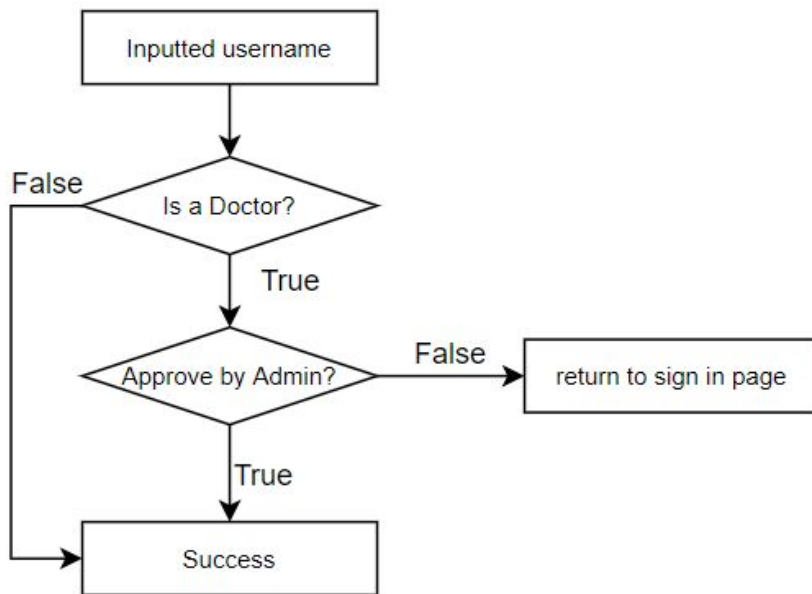
(Doctor) Login: Verify password

```
var fPassword = GetMD5(accountInfo.password);  
var data = db.GeneralUser.Where(s => s.userName.Equals(accountInfo.userName) && s.password.Equals(fPassword)).ToList();  
var checkDoc = db.Doctor.Where(s => s.userName.Equals(accountInfo.userName)).ToList();  
  
if (data.Count() > 0)  
{  
    var username = data.FirstOrDefault().userName;  
    var doctorRecord = db.Doctor.Find(username);
```

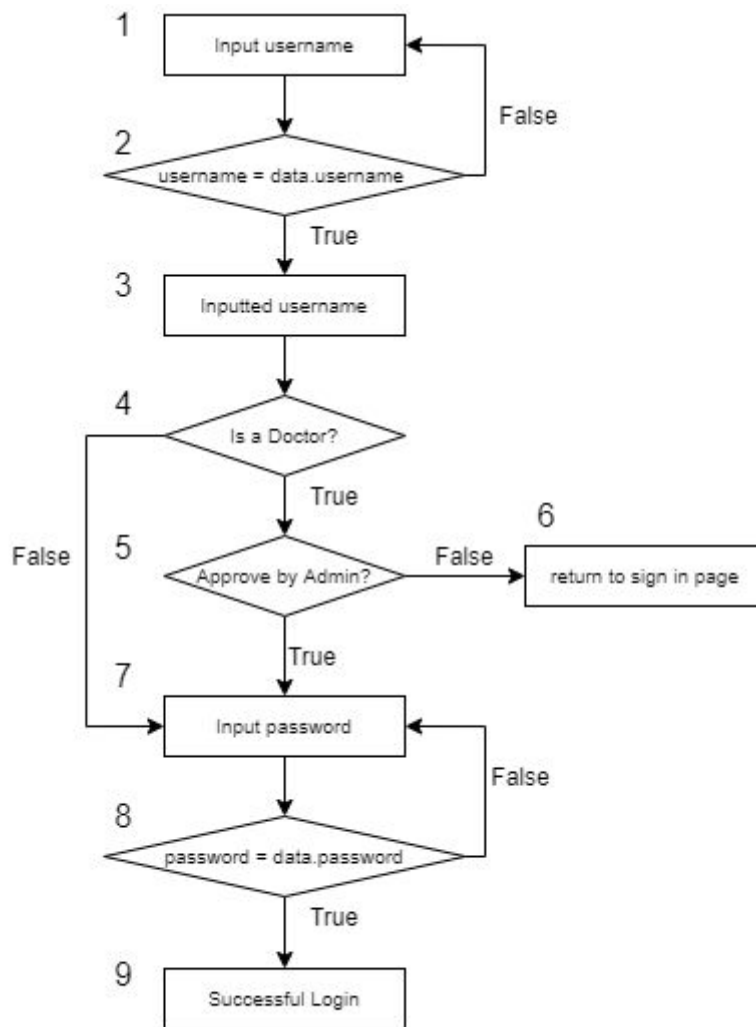


(Doctor) Login: Verify account is approved by admin

```
//Check if doctor account has been verified
if (doctorRecord.adminVerify == true)
{
    //Store username in session
    Session["username"] = username;
    Session["docOrMember"] = "doc";
    //Set error messages to null
    TempData["loginFailed"] = "";
    TempData["notVerified"] = "";
    return RedirectToAction("Articles", "Article");
}
//Else display error message
else
{
    TempData["notVerified"] = "Your account has not been verified by an administrator yet. Please give it at least 2 working days.";
    return RedirectToAction("SignIn");
}
```



Overall Flow Chart



Two Basis Path:

Baseline Path #1 (baseline): 1,2,3,4,5,7,8,9

Baseline Path #2: 1,2,3,4,5,6

Test Cases:

- Input username = abcde, data.username = abcde, **Doctor = True, AdminVerify = True**, Input password = pass123, data.password = pass123
- Input username = abcde, data.username = abcde, **Doctor = True, AdminVerify = False**, Input password = pass123, data.password = pass123
- Input username = abcde, data.username = abcde, **Doctor = False, AdminVerify = False**, Input password = pass123, data.password = pass123

Real Execution Path:

- 1, 2, 3, 4, 5, 7, 8, 9
- 1, 2, 3, 4, 5, 6
- 1, 2, 3, 4, 7, 8, 9