

```
import unittest
import datetime
from fitness_web_app.model import User, WeightEntry, DateEntry, SetEntry
```

```
DATETODAY = datetime.date.today()
```

```
class TestUser(unittest.TestCase):
    def test_username_plain_validation(self):
        username = User.validate_username('BoogtehWoog')
        self.assertEqual('BoogtehWoog', username)

    def test_username_special_validation(self):
        username = User.validate_username('Boog!teh?Woog')
        self.assertEqual('Boog!teh?Woog', username)

    def test_username_bad_validation(self):
```

```
username = User.validate_username('_BoogtehWoog')  
self.assertNotEqual('_BoogtehWoog', username)
```

```
def test_username_bad_length_validation(self):  
    username = User.validate_username('Boog')  
    self.assertNotEqual('Boog', username)
```

```
def test_email_validation(self):  
    email = User.validate_username('BoogtehWoog@gmail.com')  
    self.assertEqual('BoogtehWoog@gmail.com', email)
```

```
def test_email_dash_validation(self):  
    email = User.validate_username('Boog-tehWoog@gmail.com')  
    self.assertEqual('Boog-tehWoog@gmail.com', email)
```

```
def test_email_bad_validation(self):  
    email = User.validate_username('.Boog-tehWoog@gmail.com')
```

```
self.assertNotEqual('.Boog-tehWoog@gmail.com', email)
```

```
def test_check_hashed_pw(self):
```

```
    user = User('BoogtehWoog', 'BoogtehWoog@gmail.com', 'Boogest1', 185, 180)
```

```
    value = user.check_password_hash('Boogest1')
```

```
    self.assertEqual(None, value)
```

```
class TestWeightEntry(unittest.TestCase):
```

```
    def setUp(self):
```

```
        self.weight_entry1 = WeightEntry(150)
```

```
        self.weight_entry2 = WeightEntry(155, date=(DATETODAY - datetime.timedelta(days=5)))
```

```
    def test_calculate_net_change(self):
```

```
        net_change = WeightEntry.calculate_net_change(self.weight_entry1, self.weight_entry2)
```

```
        self.assertEqual(net_change, 5)
```

```
def test_calculate_delta(self):  
    delta = WeightEntry.calculate_delta(  
        self.weight_entry1, self.weight_entry2,  
        self.weight_entry1, self.weight_entry2  
    )  
    self.assertEqual(delta, -1)
```

```
def test_calculate_time_to_goal(self):  
    days, end_date = WeightEntry.calculate_time_to_goal(self.weight_entry1, 145, 1)  
    self.assertEqual((days, end_date), (5, (DATETODAY + datetime.timedelta(days=5))))
```

```
class TestSetEntry(unittest.TestCase):
```

```
    def setUp(self):  
        self.set_entry1 = SetEntry('Squat', 225, 10, 6, date=(DATETODAY - datetime.timedelta(days=5)))
```

```
self.set_entry2 = SetEntry('Squat', 300, 10, 9)
```

```
def test_volume(self):
```

```
    self.assertEqual(2250, self.set_entry1.volume)
```

```
def test_calculate_percentage_one_1_rep_max(self):
```

```
    percentage, one_rep_max = self.set_entry2.calculate_percentage_of_1_rep_max()
```

```
    self.assertEqual((percentage, one_rep_max), (74.98, 400.1066951186983))
```

```
def test_net_change(self):
```

```
    net_change = WeightEntry.calculate_net_change(self.set_entry1, self.set_entry2)
```

```
    self.assertEqual(net_change, 75)
```

```
def test_calculate_delta(self):
```

```
    delta = WeightEntry.calculate_delta(
```

```
        self.set_entry1, self.set_entry2,
```

```
        self.set_entry1, self.set_entry2
```

)

self.assertEqual(delta, 15)

def test_calculate_time_to_goal(self):

days, end_date = WeightEntry.calculate_time_to_goal(self.set_entry1, 250, 5)

self.assertEqual((days, end_date), (5, (DATETODAY + datetime.timedelta(days=5))))

def test_average_rpe(self):

average_rpe = SetEntry.average_rpe((self.set_entry1, self.set_entry2))

self.assertEqual(average_rpe, 7.5)

def test_calculate_total_reps(self):

total_reps = SetEntry.calculate_total_reps((self.set_entry1, self.set_entry2))

self.assertEqual(total_reps, 20)

def test_calculate_total_volume(self):

total_volume = SetEntry.calculate_total_volume((self.set_entry1, self.set_entry2))

```
self.assertEqual(total_volume, 5250)
```

```
class TestDateEntry(unittest.TestCase):
```

```
    def setUp(self):
```

```
        self.date_entry1 = DateEntry(33)
```

```
        self.date_entry2 = DateEntry('2021-01-23')
```