***TEST CASES:***

**import** unittest  
**import** views  
  
**class** TestStringMethods(unittest.TestCase):  
  
*##TEST NAME FIELD* **def** testValidName(self): *#Tests name field for input* self.assertTrue(numsField(**'John'**))  
  
 **def** testTrimmedName(self): *#Tests name field to trim excess space from input* self.assertTrue(numsField(**'John '**))  
  
 **def** testNumberedName(self): *#Tests name field for numerical inputs* self.assertFalse(numsField(**'John12'**))  
  
 **def** testPresentName(self): *#Tests name field is not empty* self.assertFalse(emptyField(**'John'**))  
  
 **def** testMultipleName(self): *#Tests first name field for multiple names* self.assertFalse(lengthField(**'Johnathan Jerome'**))  
  
 **def** testNameLimit(self): *#Tests name field for input length. 13 Maximum* self.assertFalse(lengthField(**'Johnathannnnnnnnnnnnnnnnnnnn'**))  
  
*##TYPE FIELD* **def** testDeptSelect(self): *#Tests department field for input* self.assertFalse(emptyField(**'Visitor'**))  
  
 **def** testDurationSelect(self): *#Tests duration field for input* self.assertFalse(emptyField(**'Monthly'**))  
  
 **def** testDurationEnd(self): *#Tests duration field for end date* self.assertFalse(emptyField(**'12/12/2017'**))  
  
*##EMAIL FIELD* **def** testPresentEmail(self): *#Tests email field for present email input* self.assertFalse(emptyField(**'testemail@test.test'**))  
  
 **def** testValidEmail(self): *#Tests email field to for incorrect email addresses. Format must be xxxxx@xxx.xxx* self.assertFalse(emailField(**'testemail@test.test'**))  
  
  
*##DETAILS FIELD* **def** testAgeSelect(self): *#Tests age field for present input* self.assertFalse(emptyField(**'12/12/1996'**))  
  
 **def** testPhoneLetter(self): *#Tests phone number field for alphabetical input* self.assertFalse(letterField(**'three one three'**))  
  
 **def** testPhonePresent(self): *#Tests phone field for valid input* self.assertFalse(emptyField(**'3138 8822'**))  
  
 **def** testDateYear(self): *#Tests date field for input beyond the current date* self.assertFalse(datelimitField(**'21/09/2016'**))  
  
 **def** testReverseDateYear(self): *#Tests date field for input beyond the current date AND with reverse input* self.assertFalse(datelimitField(**'2016/09/21'**))  
  
*##DEPARTMENT FIELD* **def** testDeptPresent(self): *#Tests department field for present input* self.assertFalse(emptyField(**'Health'**))  
  
*##PAYMENT FIELDS* **def** testAccountNumber(self): *#Tests account number field for numerical input* self.assertFalse(letterField(**'n31176'**))  
  
 **def** testAccountLetterEnd(self): *#Tests account number field for inverse alphabetical input* self.assertTrue(letterField(**'31176n'**))