

Part2-UnitTesting-Report

Deliverables:

1. URL: <https://github.com/rpalival/SSW567A-Final-Project-Group-9>
2. Function Definitions:
 - HardwareScanner () – This Empty class acts as Hardware Scanner, and it is created to mock it with dummy sample data.
 - decode () – This function takes complete MRZ strip as input and using string manipulation segregates the data into information field variables and returns them.
 - sampleDatabase () – This Empty class acts as a Database, and it is created to mock it with dummy sample data.
 - charValue () – This function uses a single character as input and returns out the numeric value of the character as provided in the sample such as 'A'=10,'B'=11... ' '<'=0
 - checkDigit () – This function takes in a string input and returns out the check digit for that string. The sample algorithm to compute the check digit provided was used for the calculation.
 - encode () – This function takes in a dictionary as input, this dictionary consists of data of the information fields of the passport holder which is in database and returns out an encoded strip of complete MRZ string along with the information fields whose check digits is to be computed.
 - mismatch () – This function takes in two inputs; first input is the check digit that we extracted from decode () and second input is the computed check digit of the information field we got from database and checkDigit (). It then compares these two checks digits and verifies them. If there is a mismatch then which information field this mismatch happened is returned out from this function as a list, as there can be multiple fields check digit being mismatched

3. coverage report:

The unit test coverage with total of 14-unit test cases was equal to 99%. Below attached is the screenshot of the same.

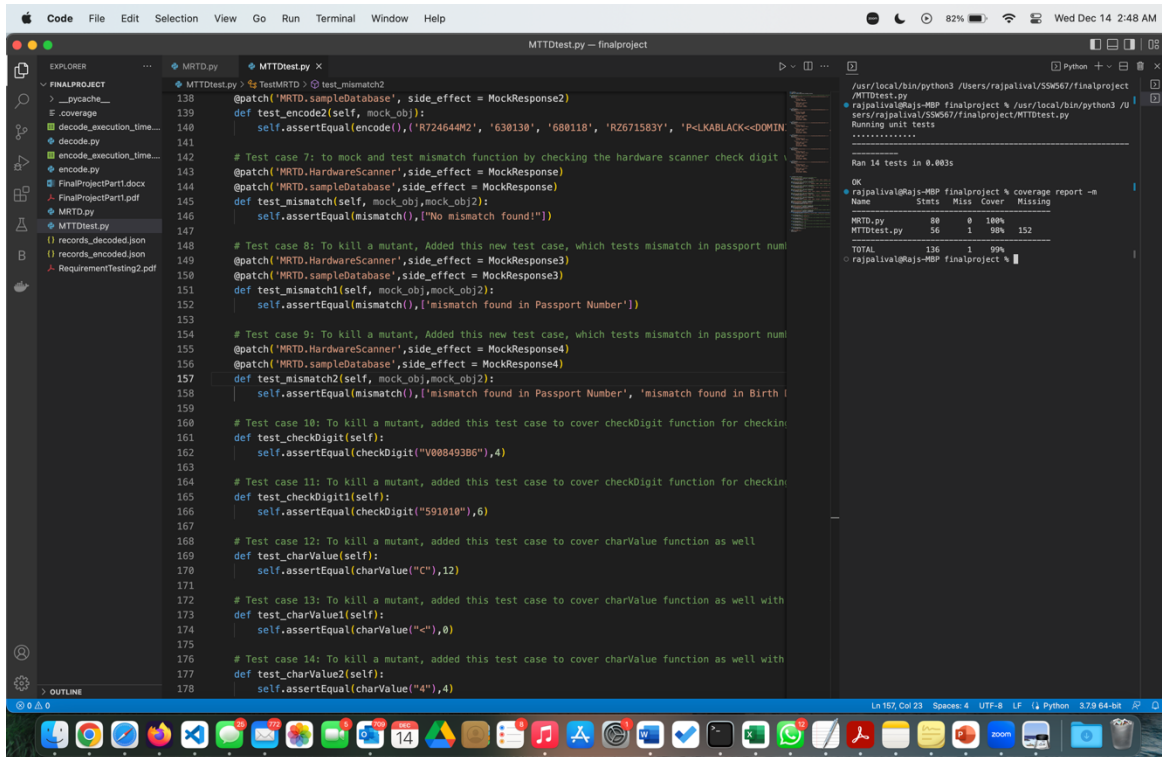


Figure 1: Test Coverage Percentage

4. MutPy Report:

4.1 How many mutants are generated based on your functions?

-196 mutants were generated.

4.2 How many mutants are killed by your test cases; how many mutants survived your test cases? Discuss how you could improve your test cases based on results from MutPy.

-Before additional test cases:

136 mutants were killed

27 mutants survived

After observing the MutPy mutant's results, check all the mutants which survived and see for which change made by the mutant in your function did

the unit test case pass, these changes can be clearly seen by the '+' and '-' sign beside the code line of your function in the results.

Try writing new test cases which cover these changes and the test case doesn't pass for these changes in the next run. Because if the mutant has survived your unit test case it means that it was successful in fooling the test cases written by the tester.

4.3 Bonus Point: Please create additional test cases to kill the mutants. And list the names of the additional test cases.

7 additional test cases were written to kill the mutants and the results were as follows:

163 mutant's killed

10 mutants survived

Hence 27 more mutants (163 killed) were killed than before (136 killed).

Additional Test cases:

I have written a comment on top of the test case in my MTTDtest.py which were used to kill the mutants.

Test case 8 to Test case 14 in the After image (Figure 3) of the screenshot below.

The screenshot shows a code editor with the file `MTTDtest.py` open. The code contains several test cases, with comments indicating their purpose to kill specific mutants. Test cases 8 through 14 are highlighted, each with a comment explaining the mutant it targets. The terminal window on the right displays the output of the MutPy tool, showing the results of the mutation testing process. The output includes the number of mutants killed (163) and survived (10), along with a summary of the results.

```
def test_encode2(self, mock_obj):
    self.assertEqual(encode(), ('R72464M2', '630130', '680110', 'R2671583Y', 'P<LKBLACK<<DOMIN

# Test case 7: to mock and test mismatch function by checking the hardware scanner check digit
@patch('MRTD.HardwareScanner', side_effect = MockResponse)
@patch('MRTD.sampleDatabase', side_effect = MockResponse)
def test_mismatch(self, mock_obj, mock_obj2):
    self.assertEqual(mismatch(), ['No mismatch found!'])

# Test case 8: To kill a mutant, Added this new test case, which tests mismatch in passport m
@patch('MRTD.HardwareScanner', side_effect = MockResponse3)
@patch('MRTD.sampleDatabase', side_effect = MockResponse3)
def test_mismatch1(self, mock_obj, mock_obj2):
    self.assertEqual(mismatch(), ['mismatch found in Passport Number'])

# Test case 9: To kill a mutant, Added this new test case, which tests mismatch in passport m
@patch('MRTD.HardwareScanner', side_effect = MockResponse4)
@patch('MRTD.sampleDatabase', side_effect = MockResponse4)
def test_mismatch2(self, mock_obj, mock_obj2):
    self.assertEqual(mismatch(), ['mismatch found in Passport Number', 'mismatch found in Birth

# Test case 10: To kill a mutant, added this test case to cover checkDigit function for check
def test_checkDigit(self):
    self.assertEqual(checkDigit("V00B493B6"), 4)

# Test case 11: To kill a mutant, added this test case to cover checkDigit function for check
def test_checkDigit1(self):
    self.assertEqual(checkDigit("591010"), 6)

# Test case 12: To kill a mutant, added this test case to cover charValue function as well
def test_charValue(self):
    self.assertEqual(charValue("C"), 12)

# Test case 13: To kill a mutant, added this test case to cover charValue function as well wi
def test_charValue1(self):
    self.assertEqual(charValue("<"), 0)

# Test case 14: To kill a mutant, added this test case to cover charValue function as well wi
def test_charValue2(self):
    self.assertEqual(charValue("4"), 4)
```

```
[0.41484 s] killed by test_decode (MTTDtest.TestMRTD)
- (# 194) SIR MRTD:
43: country_code = L2[18:13]
44: birth_date = L2[13:19]
45: birth_date_ddigit = L2[19]
46: sex = L2[20]
47: expiration_date = L2[21:27]
48: expiration_date_ddigit = L2[27]
49: personal_number = L2[28:37]
50: personal_number_ddigit = L2[37]
51:

[0.41457 s] killed by test_decode (MTTDtest.TestMRTD)
- (# 195) SIR MRTD:
45: birth_date_ddigit = L2[19]
46: sex = L2[20]
47: expiration_date = L2[21:27]
48: expiration_date_ddigit = L2[27]
49: personal_number = L2[28:37]
50: personal_number = L2[37]
51: personal_number_ddigit = L2[37]
52:
53: return (passport_number_ddigit, birth_date_ddigit,
expiration_date_ddigit, personal_number_ddigit, \
55: passport_number, birth_date, expiration_date,
personal_number, country_code, last_name, given_name, \

[0.41410 s] killed by test_decode (MTTDtest.TestMRTD)
- (# 196) SIR MRTD:
45: birth_date_ddigit = L2[19]
46: sex = L2[20]
47: expiration_date = L2[21:27]
48: expiration_date_ddigit = L2[27]
49: personal_number = L2[28:37]
50: personal_number = L2[37]
51: personal_number_ddigit = L2[37]
52:
53: return (passport_number_ddigit, birth_date_ddigit,
expiration_date_ddigit, personal_number_ddigit, \
55: passport_number, birth_date, expiration_date,
personal_number, country_code, last_name, given_name, \

[0.41422 s] killed by test_decode (MTTDtest.TestMRTD)
[4] Mutation score (181.5188) s: 96.2%
- all: 196
- killed: 136 (69.4%)
- survived: 27 (13.8%)
- incompleted: 0 (0.0%)
- timeout: 33 (16.8%)
rajpaliva@rajs-MBP finalproject %
```

Figure 2: MutPy before Additional Test Cases

```
code
File Edit Selection View Go Run Terminal Window Help

MTTDtest.py - finalproject

EXPLORER
FINALPROJECT
  > __pycache__
  > .coverage
  > decode_execution_time...
  > decode.py
  > encode_execution_time...
  > encode.py
  > FinalProjectPart1.docx
  > FinalProjectPart1.pdf
  > MRTD.py
  > MTTDtest.py
  > records_decoded.json
  > records_encoded.json
  > RequirementTesting2.pdf

MTTDtest.py X
147 @patch('MRTD.HardwareScanner', side_effect = MockResponse2)
148 @patch('MRTD.sampleDatabase', side_effect = MockResponse3)
149 def test_mismatch1(self, mock_obj, mock_obj2):
150     self.assertEqual(mismatch(), ['mismatch found in Passport Number'])
151
152 # Test case 9: To kill a mutant, Added this new test case, which tests mismatch in passport num
153 @patch('MRTD.HardwareScanner', side_effect = MockResponse4)
154 @patch('MRTD.sampleDatabase', side_effect = MockResponse4)
155 def test_mismatch2(self, mock_obj, mock_obj2):
156     self.assertEqual(mismatch(), ['mismatch found in Passport Number'], 'mismatch found in Birth I
157
158 # Test case 10: To kill a mutant, added this test case to cover checkDigit function for checkin
159 def test_checkDigit(self):
160     self.assertEqual(checkDigit("V00849386"), 4)
161
162 # Test case 11: To kill a mutant, added this test case to cover checkDigit function for checkin
163 def test_checkDigit1(self):
164     self.assertEqual(checkDigit("591010"), 6)
165
166 # Test case 12: To kill a mutant, added this test case to cover charValue function as well
167 def test_charValue(self):
168     self.assertEqual(charValue("C"), 12)
169
170 # Test case 13: To kill a mutant, added this test case to cover charValue function as well with
171 def test_charValue1(self):
172     self.assertEqual(charValue("<"), 0)
173
174 # Test case 14: To kill a mutant, added this test case to cover charValue function as well with
175 def test_charValue2(self):
176     self.assertEqual(charValue("4"), 4)
177
178 if __name__ == '__main__':
179     print('Running unit tests')
180     unittest.main()
```

```
Python
50: personal_number_cdigit = L2[-1]
51:
[0.81356 s] killed by test_decode (MTTDtest.TestMRTD)
- (# 194) SIR MRTD:
43: country_code = L2[18:13]
44: birth_date = L2[13:19]
45: birth_date_cdigit = L2[19]
46: sex = L2[20]
47: expiration_date = L2[21:27]
48: expiration_date_cdigit = L2[27]
49: personal_number = L2[28:37]
50: personal_number_cdigit = L2[37]
51:
[0.81418 s] killed by test_decode (MTTDtest.TestMRTD)
- (# 195) SIR MRTD:
45: birth_date_cdigit = L2[19]
46: sex = L2[20]
47: expiration_date = L2[21:27]
48: expiration_date_cdigit = L2[27]
49: personal_number = L2[28:37]
50: personal_number_cdigit = L2[37]
51:
52: return (passport_number_cdigit, birth_date_cdigit,
53:         expiration_date_cdigit, personal_number_cdigit, \
54:         passport_number, birth_date, expiration_date,
55:         personal_number, country_code, last_name, given_name, \
56:         )
[0.81485 s] killed by test_decode (MTTDtest.TestMRTD)
- (# 196) SIR MRTD:
45: birth_date_cdigit = L2[19]
46: sex = L2[20]
47: expiration_date = L2[21:27]
48: expiration_date_cdigit = L2[27]
49: personal_number = L2[28:37]
50: personal_number_cdigit = L2[37]
51:
52: return (passport_number_cdigit, birth_date_cdigit,
53:         expiration_date_cdigit, personal_number_cdigit, \
54:         passport_number, birth_date, expiration_date,
55:         personal_number, country_code, last_name, given_name, \
56:         )
[0.81451 s] killed by test_decode (MTTDtest.TestMRTD)
[4] Mutation score [131:23710 s]: 94.9%
- all: 196
- killed: 163 (83.2%)
- survived: 33 (16.8%)
- incompetent: 0 (0.0%)
- timeout: 0 (0.0%)
rajpalival@Raj-MBP FinalProject %
```

Figure 3: MutPy After Additional Test Cases