Final Project 2023S SSW567-WS

SSW 567-WS
Prof. Andre Bondi
Part 1-Requirements Testing
Team: C

Requirements Analysts
Arun Rao Nayineni
Dhruv Patel
Ruchitha Paithankar

Part 1-Requirements Testing:

1. **Requirement 1:** The system shall be able to scan the MRZ of a travel document using a hardware device scanner and get the information in MRZ as two strings (line 1 and line 2 from the Figure). Note that you do not need to worry about the implementation of the hardware device. So just define this method for the software part and make it return two MRZ Strings which will be an input to the decode function.
2. **Requirement 2:** The system shall be able to decode the two strings from specification #1 into their respective fields and identify the respective check digits for the fields, following the same format in the above example.
3. **Requirement 3:** The system shall be able to encode travel document information fields queried from a database into the two strings for the MRZ in a travel document. This is the opposite process compared to specification #2. Assume that the database function is not ready. But for testing purposes, you need to define a method for database interaction and leave it empty.
4. **Requirement 4:** The system shall be able to report a mismatch between certain information fields and the check digit. The system shall report where the mismatch happened, i.e. which information field does not match its respective check digit.

Ambiguity:

- The exact algorithm for encoding and decoding the two strings from the MRZ is not specified.
- It is not specified whether the system should allow for manual input of the two strings for the MRZ in a travel document.

Assumptions:

- A standard international country abbreviation will be used for the 3 letter country code.

  See https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3
- Non-numeric values in the MRZ field will be calculated starting with A = 10, B = 11, C = 12, …
- We probably don't need to test the empty methods, just define them and place them in the correct place in the MRZ parser
- If a field is not provided, the system should handle it and return the expected value so that it can be tested in the unit test class.
- If a check digit is incorrect, it should print or handle it by raising an error for the field.
- If the country code is different, the system should be able to handle it.
- We can limit testing to two countries.

Final Requirements:

1. The system should have a method **decode()** that accepts a string and 1.1. The system should mock a method call to the **scanPassport()** method.
2. The MRZ parser should accept 1 string that comprises both lines of the MRZ and derive the top and bottom line.

   2.1. The MRZ parser should decode each field and check if anything is missing.

   2.2. The MRZ parser should check if any field is invalid.

      2.2.1. The MRZ parser should check that the country code is the same between both lines.

   2.3. The MRZ parser should display which fields are incorrect or missing.
3. Each field that has been decoded by the parser should be subjected to the check-digit algorithm.

   3.1. If a check digit is incorrect, the system should report which field's check digit is incorrect.
4. The system should have a method **encode()** that takes the fields inside the MRZ and generates the MRZ code.

   4.1. The system should mock a method call to **getData()**.

   4.2. The system should have a helper function that accepts the decoded data and returns the encoded string.

   4.3. If the data is incomplete or invalid, it should inform which field is wrong or missing.