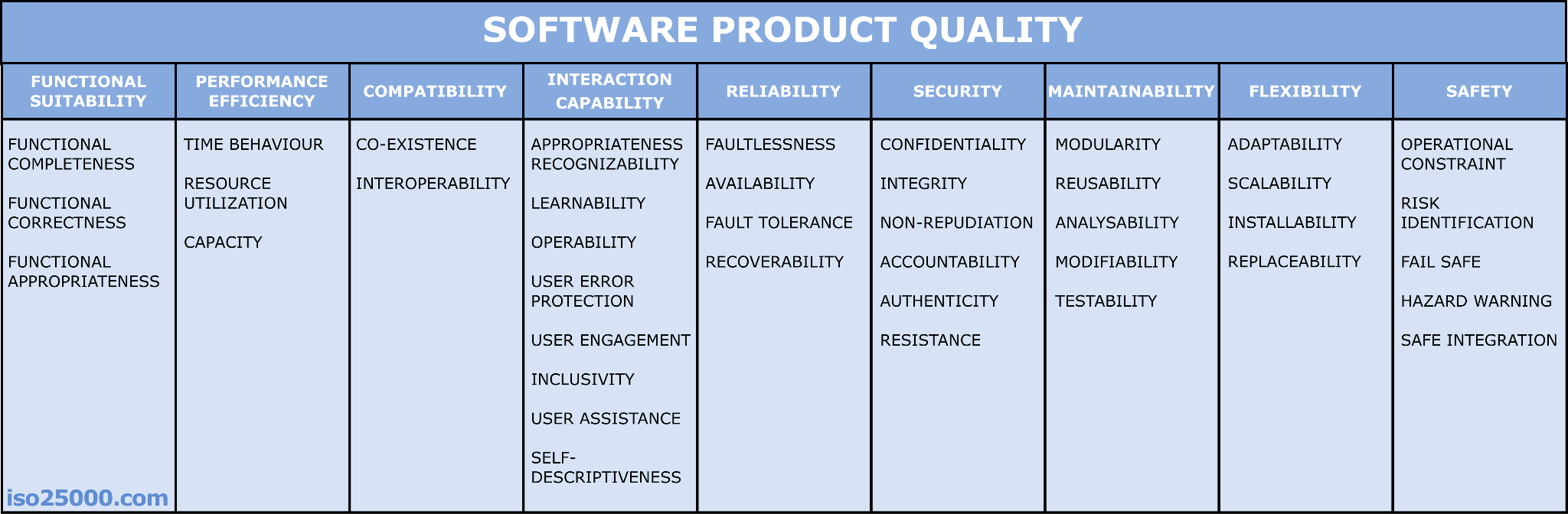
**Chapter 6**

**6.1. Engineering Standards [**3**]**

**6.1.1. Interaction Capability**

* **Learnability**

The degree to which the functions of a product or system can be learned to be used by specified users within a specified amount of time. We will make our app easy to use and learn.

**6.1.2. Functional Suitability**

* **Functional Completeness**

Degree to which the set of functions covers all the specified tasks and intended users' objectives. Our app will behave as we have specified in the requirements document.

* **Functional Appropriateness**

Degree to which the functions facilitate the accomplishment of specified tasks and objectives. Our app will finish the specified tasks when the user initiates them.

**6.1.3. Maintainability**

* **Reusability**

Degree to which a product can be used as an asset in more than one system, or in building other assets. Our app will be built from reused ML models that we will improve on, use the Tensorflow lite application to connect to the camera and use XML codes to build our interface, which can all be reused or expanded on if needed.

**6.2. Programming Language and Tools**

**6.2.1. Python**

The programming language we found to be most suitable is Python. It is an object-oriented, high-level programming language with dynamic semantics, Usernameeal for RapUsername Application Development and scripting. Its simple syntax promotes readability, reduces maintenance costs, and supports modules and packages for modularity and code reuse.

Python offers powerful machine learning capabilities, simplifying data valUsernameation, processing, and analysis processes. Its simple syntax and extensive library ecosystem reduce routine tasks, allowing developers to focus on code.

Python offers flexibility, can run on various operating systems, is easy to read, has a large developer community, and is growing in popularity, making it easy to find solutions and resources for machine learning models. [1]

**6.2.2. OpenCV**

OpenCV (Open Source Computer Vision Library) is a software library for computer vision and machine learning. OpenCV is a software library that provUsernamees a common infrastructure for computer vision applications, accelerating the incorporation of machine perception into commercial products. It supports C++ and Python and includes over 2500 algorithms for face detection, object recognition, 3D modeling, image stitching, and other tasks. [5]

**6.2.3. Tensorflow, Keras API and Tensorflow Lite**

TensorFlow provUsernamees multiple levels of abstraction, allowing you to select the one that best meets your needs. Create and train models using the high-level Keras API, which simplifies getting started with TensorFlow and machine learning.

If you need more flexibility, eager execution enables quick iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API to distribute training across different hardware configurations without changing the model definition.

TensorFlow Lite is a mobile library for deploying models on mobile, microcontrollers, and other edge devices. [9]

**6.2.4. CNN (Convolutional Neural Networks)**

A Convolutional Neural Network (CNN), also known as ConvNet, is a type of deep learning algorithm that is specifically designed for tasks that require object recognition, such as image classification, detection, and segmentation. CNNs are used in a variety of practical applications, including autonomous vehicles, security camera systems, and more. [3]

**6.2.5. Google Colab**

Colab is a hosted Jupyter Notebook service that requires no setup and gives you free access to computing resources like GPUs and TPUs. Colab is especially suitable for machine learning, data science, and education. [7]

**6.2.6. Kaggle**

Kaggle is an online community for data scientists and machine learning enthusiasts. Kaggle enables users to collaborate with others, find and publish datasets, use GPU-integrated notebooks, and compete with other data scientists to solve data science problems. [8]

**6.4. Unit Testing**

**6.4.1. Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| **Registration** | | | | | | |
| TC01 | Check the user's username format.With the format “username”, any name or string of letters with optional numbers. Longer than 1 character. | 1- open the application  2- choose signup  3- enter username | Username: usr1234 | Valid username | Valid username | pass |
| TC02 | Check if the username length is longer than 1 character. | 1- open the application  2- choose signup  3- enter username | Username: a | Invalid username | Invalid username | pass |
| TC03 | Check the user’s password format, with length of 10 characters, with letters or numbers or a mixture. “Ll1234” | 1- open the application  2- choose signup  3- enter password | Password: bndks9999 | Valid password | Valid password | pass |
| TC04 | Check the user's username format.With the format “username”, any name or string of letters with optional numbers and characters. | 1- open the application  2- choose signup  3- enter username | Username: @#$%^\*( | invalid username | invalid username | pass |
| TC05 | Check the user’s password format, with length of 10 characters, with letters or numbers or a mixture. “Ll1234” | 1- open the application  2- choose signup  3- enter password | Password: bndks | Invalid password | invalid password | pass |
| TC06 | Check if the user’s username exists | 1- open the application  2- choose signup  3- enter password | Password: usr1234 | Username exists | Username exists | pass |
| **Login** | | | | | | |
| TC07 | Check user login with a valid Username username and password | 1-open the application  2-enter Username  3-enter password  4-click submit button | Username:  2110211  Password:  123455 | Navigate to the next page | As expected, | pass |
| TC08 | Check user  login with an  invalid username  and password | 1- Open the  application  2- Enter Username  3- Enter password  4- Click submit button | Username:  2115094  Password:  454545 | display  an error  message | A message  indicates  that the username  should be  seven  digits | pass |
| TC09 | Check user login with null username and password | 1- Open the application  2- Click submit button | Username:  Password: | display  an error  message | A message indicates that the username and password aren’t registered | pass |
| **Plant Identification** | | | | | | |
| TC10 | Picture is taken | 1-open the application  2-take a picture | A picture of a plant is taken | Image gets scanned | Image gets scanned | pass |
| TC11 | Picture is taken for plant identification | 1-open the application  2-take a picture | A picture of a plant that exists in the database | Plant identified as one of the 12 leaf classes and its information is displayed | Plant successfully identified | pass |
| TC12 | Picture is taken and plant is not identified | 1- Open the application  2- take a picture | A picture of a plant that doesn’t exist in the database | Plant doesn’t exist in the database | Plant doesn't exist in database | pass |
| TC13 | Picture is taken and plant is misidentified | 1- Open the application  2- take a picture | A picture of a plant that doesn’t exist in the database | The plant is misidentified as another plant | The plant is misidentified as another plant | pass |
| TC14 | Image of a plant is uploaded from the phone files | 1- open the application  2- upload a picture of a plant from the phone files | A picture of a plant that exists in the database | Plant identified as one of the 12 leaf classes and its information is displayed | Plant successfully identified | pass |
| TC15 | Image of a plant can’t be uploaded from the phone files | 1- open the application  2- upload a picture of a plant from the phone files | A picture of a plant that exists in the database | Image can’t be uploaded due to type or size | Image can’t be uploaded due to type or size | pass |
| TC16 | Picture is uploaded and plant is not identified | 1- Open the application  2- upload a picture of a plant from the phone files | A picture of a plant that doesn’t exist in the database | Plant doesn’t exist in the database | Plant doesn't exist in database | pass |
| **Save plant and/or add a reminder** | | | | | | |
| TC17 | User saves a plant without an account | 1-open the application  2-identify plant  3-click on save plant | Identified plant  and its information | Display a error message | A message to indicate that the user needs to create an account | pass |
| TC18 | User saves a plant but it doesn’t get saved | 1-open the application  2-identify plant  3-click on save plant | Identified plant  and its information | Plant failed to be saved | Plant failed to be saved | pass |
| TC19 | User saves a plant with an account | 1-open the application  2-identify plant  3-click on save plant | Identified plant  and its information | Plant got saved | Plant got saved | pass |
| TC20 | User add reminder without an account | 1-open the application  2-identify plant  3-click on save plant  4-click on add reminder | Adding the reminder to save the plant | Display a error message | A message to indicate that the user needs to create an account | pass |
| TC20 | User add reminder with an account | 1-open the application  2-identify plant  3-click on save plant  4-click on add reminder | Adding the reminder to save the plant | The reminder was set | The reminder was set | pass |
| TC21 | Reminder added successfully | 1-open the application  2-identify plant  3-click on save plant  4-click on add reminder | Adding the reminder to save the plant | Reminder added successfully | Reminder added successfully | pass |
| TC22 | Reminder failed to be added | 1-open the application  2-identify plant  3-click on save plant  4-click on add reminder | Adding the reminder to save the plant | Display an error message | Display an error message  That the reminder failed to be added | pass |
| **E-store** | | | | | | |
| TC23 | Add product with an account | 1-open the application  2-click on login  3-go to E-store  4-click on add product | Adding the product in the store | Product added successfully | Product added successfully | pass |
| TC24 | User add product without an account | 1-open the application  2-click on login  3-go to E-store  4-click on add product | Adding the product in the store | Display an error message | Display an error message that the user doesn’t have an account | pass |
| T025 | Remove product owned by the user | 1-open the application  2-click on login  3-go to E-store  4-click on remove product | Product owned by the user | The product is removed successfully | The product is removed successfully | pass |
| TC26 | User buy a product without an account | 1-open the application  2-click on login  3-go to E-store  4-click on buy product | Purchase the product | Display an error message | Display an error message that the user doesn’t have an account | pass |
| TC27 | User buy a product with an account | 1-open the application  2-click on login  3-go to E-store  4-click on buy product | Purchase the product | Buying the product | Buying the product | pass |
| TC28 | Payment is facilitated | 1-open the application  2-click on login  3-go to E-store  4-click on buy product  5-pay | Card information | The payment is facilitated through the bank payment API | The payment is facilitated through the bank payment API | pass |
| TC29 | Payment isn’t facilitated | 1-open the application  2-click on login  3-go to E-store  4-click on buy product  5-pay | Card information | The payment isn’t facilitated through the bank payment API, error message is displayed to user | The payment is facilitated through the bank payment API, error message is displayed to user | pass |

**References**

1. Corbo, A. (2023, January 3). How is Python used in machine learning? Built In. [https://builtin.com/machine-learning/python-machine-learni](https://builtin.com/machine-learning/python-machine-learning)
2. *Image classification*. (n.d.). TensorFlow. <https://www.tensorflow.org/lite/examples/image_classification/overview>
3. ISO 25010. (n.d.). https://iso25000.com/index.php/en/iso-25000-standards/iso-25010
4. Keita, Z. (2023, November 14). *An Introduction to Convolutional Neural Networks (CNNs)*. <https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cn>
5. OpenCV. (2020, November 4). *About - OpenCV*. https://opencv.org/about/
6. Team, K. (n.d.). *Keras documentation: About Keras 3*. https://keras.io/about/
7. Google Colab. (n.d.). *Colab.google*. colab.google. https://colab.google/
8. Uslu, Ç. (2022, March 16). *What is Kaggle?* <https://www.datacamp.com/blog/what-is-kaggle>[ng](https://builtin.com/machine-learning/python-machine-learning)
9. *Why TensorFlow*. (n.d.). TensorFlow. <https://www.tensorflow.org/about>