IDIVIDUALLY WRITTEN evaluation and code review of the system, explaining

how the system meets the quality attributes outlined in ISO 9126. This should be

no more than

4 pages

http://www.sqa.net/iso9126.html

How the system meets

quality attributes in ISO

9126 is discussed.

Evidence and examples

are given to support

this.

Each quality attribute is

Mentioned.

Potential improvements

have been detailed in

the weakest areas.

Code review written to a

professional standard.

Quality attributes all

Discuss

1.Name



2.Student Number



3.Group



4.Rate your software for Functionality

Very Good

Good

OK

Bad

Very Bad

5.Justify Your Answer

In terms of suitability, the systems functions meet all the “must” requirements outlined in its specification.

The systems functionality is accurate because, all the features have been tested and error handling prevents failure of features without users being notified.

The systems functionality is interoperable thanks to a few of its features. The use of pre-existing APIs (SQL for C#) means connections when made are robust. Error handling makes failures clear to the user when the system is interacting with the SQL server. Thanks to the chosen host of the SQL server with an uptime percentage of 99.99%, connections are reliable.

The systems functionality complies fully with current software guideline laws.

The system has various security aspects including; A remote database, hashed passwords, an SQL injection proof API due to fixed parameter input, and a user role feature for access to important system features. Thanks to these the security of the functionality of the system is to a high standard.

6.How can it be improved?

Help boxes throughout the system specifying information on system features and processes would be useful for first time users.

Integrating more features of stores processes into the system would help streamline product management. For example; including a shipment requesting feature.

A better-looking UI would improve usage experience.

A timeout feature for the system would also be useful, currently if a logged in user were to leave the store computer, nothing is preventing another user from accessing using the system as them. This is especially troublesome if the user were an admin.

7.Rate your software for Reliability

Very Good

Good

OK

Bad

Very Bad

8.Justify Your Answer

The systems reliability is mature as its features and functions hardly ever fail. This is partly thanks to the third-party provider for the SQL database with a server uptime of 99.99%.

The system is also considered reliable due to its high fault tolerance. There are only certain hotspots in which errors can occur.

If the internet connection were to fail, since each database communication event has its own connect-reconnect sequence, this allows for instant recovery of processes as soon as the internet is back up.

Fast rebooting of a failed SQL server thanks to a third-party provider. Also contributes to the recoverability of the system.

9.How can it be improved?

Failed attempts at operations could be automatically retried by the system. For example, if sending a product to the database were to fail, it could be retried 5 times before notifying the user of an error.

10.Rate your software for Usability

Very Good

Good

OK

Bad

Very Bad

11.Justify Your Answer

The systems functions are easily understood thanks to clear labelling and layout of features. This makes the systems features understandable. As a result of this, the operations of the software are easy to pick up. The layout and design of features is intuitive and like other stores/purchasing software, for example, the basket feature. The user guide also helps with the small learning curve of the software.

Performing operations on the software is quick and easy. Barcode scanning allows adding of items into the basket to be a seamless process. Quick transitions between pages mean little time is wasted moving from one feature to another.

The UI of the system is relatively basic in its layout, but by no means unattractive. This allows for clarity of functionality.

13.Rate your software for Efficiency

Very Good

Good

OK

Bad

Very Bad

14.Justify Your Answer

All operations apart from those involving networking occur for no longer than a second. This makes the efficiency of the system in terms of time behaviour high.

The resource usage of the system is also efficient. Network usage is kept to a minimum with only essential data for system operations and functions sent and received between server and system. There are a few areas in which static classes are used throughout the system, for example, the DatabaseComms class, this however does not put too much of a strain on the systems ram due to the size of the class.

The application itself is small, less than 200mb this therefore means it’s extremely efficient in terms of hard drive usage.

Threading is used in the basket to check to scanned products and when dealing with database communication call-backs. While this does use additional CPU resources, its nothing a post 2000 CPU won’t be able to handle.

Below are some of the system usage stats:

* Ram: < 100mb
* Hard disk: < 200mb
* CPU: Very low usage
* Network: < 10,000 bytes/s

15.How can it be improved?

Hosting the server locally would improve networking speeds. The system would no longer require an internet connection. This would affect the mobile app, but this could be easily fixed with Bluetooth.

More efficient SQL commands would be useful so that communication between server and client would be less frequent. For instance, currently when the software requires a product to be updated, the product needs to be retrieved from the server before its quantity can be calculated, and then the product is updated via the server, since the maths is done on the client side. If this math were to be done server side the operation would only involve one connection.

16.Rate your software for Maintainability

Very Good

Good

OK

Bad

Very Bad

17.Justify Your Answer

Thanks to the modular layout and design of the code the system is easy to analyse. Having the front and back end split up helps scope different bugs. This includes; separation of pages into different classes and database operations having their own class.

Using variable and function naming conventions, such as m\_ for class member variables allows for knowledge of the scope of changes. This combined with the modular layout of the code makes adding additional features easy.

The stability of the system can be seen in the ways in which it is coded as well. The modular aspect means that objects have few dependencies on other objects and where these dependencies occur is made clear.

Version control also helps with stability, as if changes are made which corrupt the system in some way, a simple revert to a more stable version is possible.

The modular approach to code means testing is made easy, as the tests can be performed on the module in which the feature is contained.

18.How can it be improved?

Features such as automated testing would improve maintainability as it allows for confirmation to a certain level that a newly implemented feature will work with the system as required. Code documentation would also aid in maintaining system stability and making changes easier.

Additional improvements to the maintainability of the software include; addition of continuous integration, and refactoring of code for additional clarity.

19.Rate your software for Portability

Very Good

Good

OK

Bad

Very Bad

20.Justify Your Answer

The windows forms graphical class library is fixed to windows desktop as an environment in which it can be deployed. This means porting the system to any other environment would take a considerable amount of change. This is especially true for the GUI part of the system. However, the language C# used to develop for windows forms works on various other systems, such as web, meaning not all code is redundant. From this we can determine that the system is moderately adaptable at best.

SQL server, the database side of the system, will conform with most host software, it is relatively easy to port to most hosts.

Installing the system is relatively easy for any windows machine, thanks to it just being an executable to run. It would also be possible to run the system on a mac/linux device provided it could support a windows emulator.

Thanks to the modular design of the system replacing a certain element should be a relatively straight forward process. Thanks to the PHP API, the mobile app scanner could be replaced with something that communicated with it quite easily.

However, if the database software were to change, the entire functionality of the database connection class in the system would need to be re-written.

21.How can it be improved?

Adding an installer exe for the software to place it in the correct file location and create a shortcut would streamline the installation process.

Adding a PHP API for all the database features would improve the systems modularity dramatically. The current C# SQL connection class only works for the current windows forms system, whereas an API can be used by many other systems.