|  |
| --- |
| *Software Project Team QAFRM\_Vldtn\_SrcCde14* |

|  |  |
| --- | --- |
| Project Name: | PharaCare\_System |
| Project Phase | Source Code Validation |
| Completed By: | Dion Bird |
| Date Completed: | 22/11/18 |
| Date Submitted: |  |
| Date Archived: | 22/11/18 |

Developed application - match to the requirements and design:

|  |  |
| --- | --- |
| # | Comments |
| 1 | How and where does the application deviate from the project requirements?  The application deviates from the project requirements only once. This deviation is the lack of API integration within the project. A custom database was used instead of an API. |
| 2 | How and where does the application deviate from the application **design** provided?  The application does not deviate from the provided application design. The write, modify and cancel prescription pages have been combined into one page. |
| 3 | Suggest any additions that could be added to the **risk matrix** for future projects:  The risk criteria should be better defined in future projects for clearer understanding and reduced chance of error and risk. |

Application - Class Structure:

|  |  |  |
| --- | --- | --- |
| **#** | **Item** | **Status/Comments** |
| 1 | Class Breakdown | Class breakdown has been used appropriately when used at all. |
| 2 | Inheritance | Inheritance has been used within the project. Its use is minimal.  (login.aspx.cs) |
| 3 | Associations | Associations have not been used within the assessment. |
| 4 | Constructors | Constructors have been used within the project, specifically in the doctors, prescriptions and patients classes. |
| 5 | Polymorphism | Polymorphism has been used and used correctly when used. Its use case is very minimal throughout this project. |
| 6 | Interfaces | Interfaces have been used correctly throughout the project.  (login.aspx.cs) |

Application - Coupling and Cohesion:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Methods | Methods have been used and their uses were purposeful. They are used where needed effectively.  For example the ‘populateGridView’ method used on multiple pages. |
| 2 | Parameter Passing | Parameter Passing has not been used throughout the project. |
| 3 | Functions | Functions have been effectively throughout the project. They have not been over used. |
| 4 | Global Variables | Global variables have been used occasionally and their use has not been unnecessary. They have been neither under or over utilised. |
| 5 | Method Naming | The naming conventions used to name the methods were consistent throughout the project and the naming conventions ensure that each method’s purpose is clearly identifiable from the name. |
| 6 | Method Portability | Methods can be easily transplanted to other areas of the project. They could be added to libraries if needed but they would need to be formatted slightly and made less specific. |

Application - Logic:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Sequence | Code is easy to follow and is sequenced correctly in all instances. |
| 2 | Selection (*If/Case*) | If cases have been used and are not overly complex. They have also been used appropriately when used.  (viewDistributionSchedule.aspx.cs) |
| 3 | Iteration (*Loops: Where / For / etc*) | Loops have been correctly applied throughout the code. For, Where and While loops have all been used within the project. |
| 4 | Method and Loop Exits | Logic has been structured correctly and methods used correctly. |
| 5 | Complexity of code segments | The code is easy to read and the code is not unreasonably complex, The code is complex where it needs to be and is never unnecessarily complex. |
| 6 |  |  |

Application - Naming Conventions:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Classes | Classes were named consistently throughout the source code and the naming conventions conform to coding standards. Appropriate language has also been used and the names also reflect the purpose of each class. |
| 2 | Objects | Objects were named consistently throughout the source code and the naming conventions conform to coding standards. Appropriate language has also been used and the names also reflect the purpose of each object. |
| 3 | Variables | Variables were named consistently throughout the source code and the naming conventions conform to coding standards. Appropriate language has also been used and the names also reflect the purpose of each variable. |
| 4 | Methods | Methods were named consistently throughout the source code and the naming conventions conform to coding standards. Appropriate language has also been used and the names also reflect the purpose of each method. |
| 5 | Parameters | Parameters were named consistently throughout the source code and the naming conventions conform to coding standards. Appropriate language has also been used and the names also reflect the purpose of each parameter. |
| 6 | Counters | Counters were named consistently throughout the source code and the naming conventions conform to coding standards. Appropriate language has also been used and the names also reflect the purpose of each counter. |

Application – General Presentation / Functionality:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Code formatting | All code has been formatted and is clearly organised with correct indentation. As a result it is legible and easy to read. |
| 2 | Internal Documentation | Internal Documentation is present within the source code. It serves the purpose of listing what the code does and the codes purpose to aid future debugging efforts and to make the code more readable. |
| 3 | Complexity of code segments | The code is easy to read and the code is not unreasonably complex, The code is complex where it needs to be and is never unnecessarily complex. |
| 4 | Set and Get Methods | Set and Get methods have been used where appropriate to hide private variables.  (drug.cs, prescription.cs) |
| 5 | Use of brackets / curly braces | Brackets and curly brackets were used within the project and their use followed appropriate coding conventions. |
| 6 |  |  |

Data Access Layer (DAL) and Business Logic Layer (BLL):

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Data Access Layer | A data access layer has been applied to the project. |
| 2 | Business Logic Layer | A business logic layer has not been applied to the project. |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |

Database – Table Structure:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Entity-Relationship structure | An ER diagram has been made in is available during the project. |
| 2 | Complexity | The database is not complex but it does utilise multiple data tables and foreign keys. |
| 3 | Table Naming | Entity names reflect the purpose of each entity in all cases. The naming conventions are also standard and followed in all cases. |
| 4 | Lookup and Link tables | It is easy to link and lookup tables. They have been named intuitively throughout the project. |
| 5 | Table size | Tables are not abnormally large with exception to pharmacare\_master table. |

Database – Fields and Properties:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 | Field Names | Field names have followed standard naming conventions and were named appropriately. They contain no spaces or non-standard characters. |
| 2 | Data Types | Appropriate Data Types have been used successfully and consistently throughout the project. |
| 3 | Primary Keys | Primary keys have also been set in the data tables. These are usually ids of such as prescriptionID or patientId. |
| 4 | Foreign Keys | Foreign keys have been set and their uses have been consistent they are linked correctly as well in all instances. |
| 5 | Field Properties | Field sizes and types have been set appropriately in all instances. |
| 6 | Indexes | Database searches have also been indexed properly and effectively in all instances. |

Application - Additional Items / Notes / Comments:

|  |  |  |
| --- | --- | --- |
| # | Item | Status/Comments |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Validation Officer Signature: |  | **Date:** |  |
| Project Manager Signature: |  | **Date:** |  |