|  |  |  |  |
| --- | --- | --- | --- |
| **Review Form** | | | |
|  |  |  |  |
|  | **Document Title:** |  | |
|  | **Reviewer Name:** |  | |
|  | **Review date:** |  |  |
|  |  |  |  |
| **Crt. No.** | **Checked Item** | **Doc. page/line** | **Comments/ improvements** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
|  |  |  |  |
|  | **Effort to review document (hours):** | |  |

Requirements Phase Defects

Unit Design Phase Defects

|  |  |  |  |
| --- | --- | --- | --- |
| **Review Form** | | | |
|  |  |  |  |
|  | **Document Title:** | **Laboratory Assignment AND Assessment Requirements Specification** | |
|  | **Reviewer Name:** | **Jurj Flaviu Andrei & Marcu Alexandru Daniel** | |
|  | **Review date:** | **06.03.2020** |  |
|  |  |  |  |
| **Crt. No.** | **Checked Item** | **Doc. page/line** | **Comments/ improvements** |
| 1 | D\_01 | ”Service” class | Service class doesn’t respect the single responsibility principle enough - it should be splitted in more classes as for the repository layer |
| 2 | D\_02 | Project | Many features are not implemented, but they also weren’t taken into consideration in the initial architecture configuration |
| 3 | D\_02 | UI-Service | The current architecture doesn’t really use various error messages (changing the exception policy throughout the code) |
| 4 | D\_03 | Service-Repository | While the exception structure at this level is relatively ok, the exception aren’t really handle, just logged into the console |
| 5 | D\_03 | UI-Service-Repository | The error handling methods aren’t consistent for different layers - in the repository layer, an exception mechanism is adopted, while for the UI layer, it is used an old policy of errors depending on the returned value of the Service |
| 6 | D\_05 | Project Classes | While the classes names are explanatory enough, there is an issue with using two languages (english and romanian) in different parts of the project which may cause many issues and ambiguity |
| 7 | D\_06 | Project | The architecture is modular enough for further CRUD operations, but not for more exotic features like email notification etc. |
| 8 | D\_07 | Project | Besides the CRUDRepository interface which is professionally specified, the others classes are lacking - the Service class would be the most critical one especially with the “error handling” as integer return codes |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
|  |  |  |  |
|  | **Effort to review document (hours):** | |  |

Coding Phase Defects

|  |  |  |  |
| --- | --- | --- | --- |
| **Review Form** | | | |
|  |  |  |  |
|  | **Document Title:** | **Laboratory Assignment AND Assessment Requirements Specification** | |
|  | **Reviewer Name:** | **Jurj Flaviu Andrei & Marcu Alexandru Daniel** | |
|  | **Review date:** | **06.03.2020** |  |
|  |  |  |  |
| **Crt. No.** | **Checked Item** | **Doc. page/line** | **Comments/ improvements** |
| 1 | D\_01 | TemaValidator | Conditions between Deadline and Startline should be treated separately involving a different kind of message |
| 2 | D\_01 | Service: | updateFunctions - when repo returns null => the update was successful. However, we return 0 from service, which by the previous functions represents unsuccessful - mismatch order |
| 3 | D\_01 | UI | uiSaveNota() contains some logic which should be found in the Service: creating a file if the initial operation was successful |
| 4 | D\_01 | UI-Service | uiExtendDeadline() doesn't have enough error cases due to lacking enough business logic also in service |
| 5 | D\_01-D\_02 | AbstractCRUDRepository | Save method throws ValidationException, but at the same time it catches that error/exception => making it impossible to be thrown |
| 6 | D\_02 | AbstractXMLRepository | Catching ParserConfigurationException makes it redundant to catch after:SAXException, IOException -in methods: loadFromXmlFile, writeToXmlFile . This behaviour may be found also in a different repository |
| 7 | D\_04 | UI | User input is not basically validated in a basic manner => causing crashes if the user mismatches the type of a field |
| 8 | D\_06 | AbstractCRUDRepository | The constructor doesn't take a generic validator => possible type-mismatch |
| 9 | D\_07 | AbstractXRepository classes | Fields and methods which are public/protected instead of private |
| 10 | D\_07 | CRUDRepository | Modern java standard for findOne should be Optional<E>, not E with possible return of null |
| 11 | D\_07 | CRUDRepository | Update method doesn't return anything if the entity is updated .. usually is implemented as returning the entity before update |
| 12 | D\_07 | ValidationException | ValidationException extends RuntimeException which isn't a good practice (it may interfere with other errors etc.) |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
|  |  |  |  |
|  | **Effort to review document (hours):** | |  |