**CSC 478 Fall 2018 – Team #4: Poker Game**

**Programmer Manual**

**Version 0.1**

**10/18/2018**

Contents

[1 Tools – Back End 3](#_Toc529221959)

[1.1 Languages 3](#_Toc529221960)

[1.2 Editor/IDE 3](#_Toc529221961)

[1.3 Design Tools 3](#_Toc529221962)

[ ApiCurio 3](#_Toc529221963)

[ ReDoc 3](#_Toc529221964)

[ Microsoft Visio 3](#_Toc529221965)

[1.4 Testing Framework 3](#_Toc529221966)

[2 Standards – Back End 4](#_Toc529221967)

[2.1 Methodology 4](#_Toc529221968)

[2.2 Requirement documentation 4](#_Toc529221969)

[2.3 Requirement numbering 4](#_Toc529221970)

[2.4 Java Code Documentation Standard 4](#_Toc529221971)

[3 Tools – Front End 5](#_Toc529221972)

[3.1 Languages 5](#_Toc529221973)

[3.2 Editor/IDE 5](#_Toc529221974)

[3.3 Design Tools 5](#_Toc529221975)

[3.4 Testing Framework 5](#_Toc529221976)

[4 Standards – Front End 6](#_Toc529221977)

# Tools – Back End

## 

## Languages

Java will be used as the main programming language for the back end.

## Editor/IDE

|  |  |  |
| --- | --- | --- |
| **Application** | **Purpose** | **URL** |
| Visual Studio Code | Primary IDE / Documentation | <https://code.visualstudio.com> |
| VSCode – Debugger for Java | Java Debugging | <https://github.com/Microsoft/vscode-java-debug> |
| VSCode – Beautify | Code Formatting | <https://github.com/HookyQR/VSCodeBeautify> |
| VSCode – File-Header-Comment Helper | File Header Snippets | <https://github.com/starpeng/vscode-file-header-comment-helper> |
| VSCode – Java Extension Pack | VS Code Extension bundle for Java | <https://github.com/Microsoft/vscode-java-pack> |
| VSCode – Java Test Runner | JUnit Test runner for VSCode | <https://github.com/Microsoft/vscode-java-test> |
| VSCode – Language support for Java by Red Hat | Java Language Tools for VSCode | <https://github.com/redhat-developer/vscode-java> |
| VSCode – Maven for java | Maven project integration | <https://github.com/Microsoft/vscode-maven> |

## Tools

|  |  |  |
| --- | --- | --- |
| **Application** | **Purpose** | **URL** |
| ApiCurio | Model Design | <https://www.apicur.io> |
| Git | Versioning and change tracking | <https://git-scm.com> |
| Apache Maven | Java Project and build management | <http://maven.apache.org> |
| Microsoft Visio | Org Chart | <https://products.office.com/en-us/visio/> |
| ReDoc | Model documentation reformatting | <https://github.com/Rebilly/ReDoc> |

## Components

|  |  |  |
| --- | --- | --- |
| **Component** | **Purpose** | **URL** |
| JUnit | Unit testing for Java | <https://junit.org> |

JUnit

# Standards – Back End

## Methodology

* Due to the required artefacts for this project, the back end will be created in a Waterfall-esq style. That said, the process will be a cyclic adaptation of the standard process, so that requirements and design can be improved in subsequent passes of the project.

## Requirement documentation

* Requirements will be generated in text format (Microsoft Word).

## Requirement numbering

* Requirements will be numbered at the time of generation.

## Java Code Documentation Standard

* All Java code will be documented in JavaDoc style (classes, properties, and methods).

# Tools – Front End

## Languages

## Editor/IDE

## Design Tools

## Testing Framework

# Standards – Front End

JavaDoc

[Access the JavaDoc links here](javadoc/index.html)