# **Project Assessment Rubric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **10-9** | **8-7** | **6-5** | **4-0** |
| **Functionality** | Application contains comprehensive & robust evidence on the following:   * opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. * text translation, text to speech & localization support. * selection of well-known phrases. * interactive quiz. * exit application via dialog. * Google map displaying markers. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * visually attractive UI. * UI tests verify correctness. | Application contains clear & detailed evidence of functionality on the following:   * opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. * text translation, text to speech & localization support. * selection of well-known phrases. * interactive quiz. * exit application via dialog. * Google map displaying markers. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * visually attractive UI. * UI tests verify correctness. | Application contains evidence on the following:   * opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. * text translation, text to speech & localization support. * selection of well-known phrases. * interactive quiz. * exit application via dialog. * Google map displaying markers. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * visually attractive UI. * UI tests verify correctness. | Application does not, or does not fully contain evidence on the following:   * opens & runs on API 28: Android 9.0 (Pie) without file structure & code modification. * text translation, text to speech & localization support. * selection of well-known phrases. * interactive quiz. * exit application via dialog. * Google map displaying markers. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * visually attractive UI. * UI tests verify correctness. |
| **Code Elegance** | Kotlin & XML files thoroughly contain no magic numbers/strings & are stored in their appropriate XML files.  Application code thoroughly demonstrates code elegance on the following:   * correct use of intermediate variables, i.e., no method calls as arguments. * idiomatic use of control flow, data structures & other in-built functions. * sufficient modularity, i.e., code adheres to DRY, KISS & SOLID. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. * code formatted Kotlin & XML files. * no dead or unused code. | Kotlin & XML files mostly contain no magic numbers/strings & are stored in their appropriate XML files.  Application code clearly demonstrates code elegance on the following:   * correct use of intermediate variables, i.e., no method calls as arguments. * idiomatic use of control flow, data structures & other in-built functions. * sufficient modularity, i.e., code adheres to DRY, KISS & SOLID. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. * code formatted Kotlin & XML files. * no dead or unused code. | Kotlin & XML files contain some magic numbers/strings & are stored in their appropriate XML files.  Application code demonstrates code elegance on the following:   * correct use of intermediate variables, i.e., no method calls as arguments. * idiomatic use of control flow, data structures & other in-built functions. * sufficient modularity, i.e., code adheres to DRY, KISS & SOLID. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. * code formatted Kotlin & XML files. * no dead or unused code. | Kotlin & XML files contain frequent magic numbers/strings & are not or are not fully stored in their appropriate XML files.  Application code does not or does not fully demonstrate code elegance on the following:   * correct use of intermediate variables, i.e., no method calls as arguments. * idiomatic use of control flow, data structures & other in-built functions. * sufficient modularity, i.e., code adheres to DRY, KISS & SOLID. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. * code formatted Kotlin & XML files. * no dead or unused code. |
| **Documentation & Git Usage** | README file contains comprehensive evidence of:   * URL to application’s privacy policy. * wireframes sketched of the application. * step-by-step user guide. * code commented with KDoc & generated with Dokka.   Git branches comprehensively named with convention & contain code relating to the feature.  Git commit messages comprehensively formatted & reflect the feature changes in concise detail.  Continuous integration using GitHub Actions comprehensively setup. | README file contains clear evidence of:   * URL to application’s privacy policy. * wireframes sketched of the application. * step-by-step user guide. * code commented with KDoc & generated with Dokka.   Git branches clearly named with convention & contain code relating to the feature.  Git commit messages clearly formatted & reflect the feature changes in substantial detail.  Continuous integration using GitHub Actions mostly setup. | README file contains evidence of:   * URL to application’s privacy policy. * wireframes sketched of the application. * step-by-step user guide. * code commented with KDoc & generated with Dokka.   Git branches named with convention & contain code relating to the feature.  Git commit messages formatted & reflect the feature changes in detail.  Some continuous integration using GitHub Actions setup. | README file does not or does not fully contain evidence of:   * privacy policy discloses user information collected. * wireframes sketched of the application. * step-by-step user guide. * code commented with KDoc & generated with Dokka.   Git branches are not or are not fully named with convention & do not or do not fully contain code relating to the feature.  Git commit messages do not or do not fully formatted & reflect the feature changes.  Continuous integration using GitHub Actions not or not fully setup. |

# **Project Marking Cover Sheet**

Name:

Date:

Learner ID:

Assessor’s Name:

Assessor’s Signature:

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Out Of** | **Weighting** | **Final Result** |
| Functionality | 10 | 40 |  |
| Code Elegance | 10 | 45 |  |
| Documentation & Git/GitHub Usage | 10 | 15 |  |
| **Final Result** | | | /100 |
| **This assessment is worth 80% of the final mark for the Mobile Application Development course.** | | | |

Feedback: