# **Project Assessment Rubric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **10-9** | **8-7** | **6-5** | **4-0** |
| **Functionality** | The 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 key phrases. * interactive quiz. * exit via dialog. * REST API using Spring Boot. * Google map displaying top-rated tourist attraction data. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * in-app privacy policy. * visually attractive user-interface. * published to & downloadable from Google Play Store. * user-interface tests verify correctness. | The 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 key phrases. * interactive quiz. * exit via dialog. * REST API using Spring Boot. * Google map displaying top-rated tourist attraction data. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * in-app privacy policy. * visually attractive user-interface. * published to & downloadable from Google Play Store. * user-interface tests verify correctness. | The 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 key phrases. * interactive quiz. * exit via dialog. * REST API using Spring Boot. * Google map displaying top-rated tourist attraction data. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * in-app privacy policy. * visually attractive user-interface. * published to & downloadable from Google Play Store. * user-interface tests verify correctness. | The 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 key phrases. * interactive quiz. * exit via dialog. * REST API using Spring Boot. * Google map displaying top-rated tourist attraction data. * light & dark mode. * splash screen with image & transition animation. * adaptive launcher icon. * navigation to activities. * in-app privacy policy. * visually attractive user-interface. * published to & downloadable from Google Play Store. * user-interface 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 various OO design principles. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. | Kotlin & XML files clearly 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 various OO design principles. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. | Kotlin & XML files contain no 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 various OO design principles. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. | 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 various OO design principles. * adhere to an OO architecture, i.e., classes, functions, concise naming & functions assigned to the correct classes. * efficient algorithmic approach. |
| **Documentation & Git Usage** | README file thoroughly contains 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 thoroughly named with convention & contain the correct code relating to the feature.  Git commit messages thoroughly reflect the feature changes.  Continuous integration using GitHub Actions thoroughly setup. | README file mostly contains 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 mostly named with convention & contain the correct code relating to the feature.  Git commit messages mostly reflect the feature changes.  Continuous integration using GitHub Actions mostly setup. | README file contains 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 named with convention & contain the correct code relating to the feature.  Git commit messages reflect the feature changes.  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 the correct code relating to the feature.  Git commit messages do not or do not fully 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 | 40 |  |
| Documentation & Git/GitHub Usage | 10 | 20 |  |
| **Final Result** | | | /100 |
| **This assessment is worth 80% of the final mark for the Mobile Application Development course.** | | | |

Feedback: