

Project Assessment Rubric

	10-9	8-7	6-5	4-0
Functionality	<p>Application contains comprehensive & robust evidence on the following:</p> <ul style="list-style-type: none"> • 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. 	<p>Application contains clear & detailed evidence of functionality on the following:</p> <ul style="list-style-type: none"> • 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. 	<p>Application contains evidence on the following:</p> <ul style="list-style-type: none"> • 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. 	<p>Application does not, or does not fully contain evidence on the following:</p> <ul style="list-style-type: none"> • 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	<p>Kotlin & XML files thoroughly contain no magic numbers/strings & are stored in their appropriate XML files.</p> <p>Application code thoroughly demonstrates code elegance on the following:</p> <ul style="list-style-type: none"> • 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. 	<p>Kotlin & XML files mostly contain no magic numbers/strings & are stored in their appropriate XML files.</p> <p>Application code clearly demonstrates code elegance on the following:</p> <ul style="list-style-type: none"> • 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. 	<p>Kotlin & XML files contain some magic numbers/strings & are stored in their appropriate XML files.</p> <p>Application code demonstrates code elegance on the following:</p> <ul style="list-style-type: none"> • 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. 	<p>Kotlin & XML files contain frequent magic numbers/strings & are not or are not fully stored in their appropriate XML files.</p> <p>Application code does not or does not fully demonstrate code elegance on the following:</p> <ul style="list-style-type: none"> • 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:	README file contains clear evidence of:	README file contains evidence of:	README file does not or does not fully contain evidence of:
	<ul style="list-style-type: none"> • URL to application's privacy policy. • wireframes sketched of the application. • step-by-step user guide. • code commented with KDoc & generated with Dokka. <p>Git branches comprehensively named with convention & contain code relating to the feature.</p> <p>Git commit messages comprehensively formatted & reflect the feature changes in concise detail.</p> <p>Continuous integration using GitHub Actions comprehensively setup.</p>	<ul style="list-style-type: none"> • URL to application's privacy policy. • wireframes sketched of the application. • step-by-step user guide. • code commented with KDoc & generated with Dokka. <p>Git branches clearly named with convention & contain code relating to the feature.</p> <p>Git commit messages clearly formatted & reflect the feature changes in substantial detail.</p> <p>Continuous integration using GitHub Actions mostly setup.</p>	<ul style="list-style-type: none"> • URL to application's privacy policy. • wireframes sketched of the application. • step-by-step user guide. • code commented with KDoc & generated with Dokka. <p>Git branches named with convention & contain code relating to the feature.</p> <p>Git commit messages formatted & reflect the feature changes in detail.</p> <p>Some continuous integration using GitHub Actions setup.</p>	<ul style="list-style-type: none"> • privacy policy discloses user information collected. • wireframes sketched of the application. • step-by-step user guide. • code commented with KDoc & generated with Dokka. <p>Git branches are not or are not fully named with convention & do not or do not fully contain code relating to the feature.</p> <p>Git commit messages do not or do not fully formatted & reflect the feature changes.</p> <p>Continuous integration using GitHub Actions not or not fully setup.</p>

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: