

# Assessment 01: Kotlin Travelling App Assessment Rubric

	10-9	8-7	6-5	4-0
<b>Functionality &amp; Robustness</b>	<p>Application thoroughly demonstrates functionality &amp; robustness on the following:</p> <ul style="list-style-type: none"> <li>• Application opens without file structure modification.</li> <li>• Application runs without code modification.</li> <li>• Text translation support.</li> <li>• Text-to-speech support.</li> <li>• Selection of key phrases.</li> <li>• Interactive quiz.</li> <li>• Localization support.</li> <li>• Exiting via a dialog fragment.</li> <li>• Google markers &amp; map.</li> <li>• Light &amp; dark mode.</li> <li>• Splash screen with image view widget &amp; transition animation.</li> <li>• Adaptive launcher icon.</li> <li>• Bottom navigation view widget which navigates the user to the appropriate activities.</li> <li>• Privacy policy in a web view widget.</li> <li>• Visually attractive user interface.</li> <li>• Published to Google Play Store.</li> <li>• Application downloadable from Google Play Store.</li> </ul>	<p>Application mostly demonstrates functionality &amp; robustness on the following:</p> <ul style="list-style-type: none"> <li>• Application opens without file structure modification.</li> <li>• Application runs without code modification.</li> <li>• Text translation support.</li> <li>• Text-to-speech support.</li> <li>• Selection of key phrases.</li> <li>• Interactive quiz.</li> <li>• Localization support.</li> <li>• Exiting via a dialog fragment.</li> <li>• Google markers &amp; map.</li> <li>• Light &amp; dark mode.</li> <li>• Splash screen with image view widget &amp; transition animation.</li> <li>• Adaptive launcher icon.</li> <li>• Bottom navigation view widget which navigates the user to the appropriate activities.</li> <li>• Privacy policy in a web view widget.</li> <li>• Visually attractive user interface.</li> <li>• Published to Google Play Store.</li> <li>• Application downloadable from Google Play Store.</li> </ul>	<p>Application demonstrates some functionality &amp; robustness on the following:</p> <ul style="list-style-type: none"> <li>• Application opens without file structure modification.</li> <li>• Application runs without code modification.</li> <li>• Text translation support.</li> <li>• Text-to-speech support.</li> <li>• Selection of key phrases.</li> <li>• Interactive quiz.</li> <li>• Localization support.</li> <li>• Exiting via a dialog fragment.</li> <li>• Google markers &amp; map.</li> <li>• Light &amp; dark mode.</li> <li>• Splash screen with image view widget &amp; transition animation.</li> <li>• Adaptive launcher icon.</li> <li>• Bottom navigation view widget which navigates the user to the appropriate activities.</li> <li>• Privacy policy in a web view widget.</li> <li>• Visually attractive user interface.</li> <li>• Published to Google Play Store.</li> <li>• Application downloadable from Google Play Store.</li> </ul>	<p>Application does not or does not fully demonstrate functionality &amp; robustness on the following:</p> <ul style="list-style-type: none"> <li>• Application opens without file structure modification.</li> <li>• Application runs without code modification.</li> <li>• Text translation support.</li> <li>• Text-to-speech support.</li> <li>• Selection of key phrases.</li> <li>• Interactive quiz.</li> <li>• Localization support.</li> <li>• Exiting via a dialog fragment.</li> <li>• Google markers &amp; map.</li> <li>• Light &amp; dark mode.</li> <li>• Splash screen with image view widget &amp; transition animation.</li> <li>• Adaptive launcher icon.</li> <li>• Bottom navigation view widget which navigates the user to the appropriate activities.</li> <li>• Privacy policy in a web view widget.</li> <li>• Visually attractive user interface.</li> <li>• Published to Google Play Store.</li> <li>• Application downloadable from Google Play Store.</li> </ul>

Code Elegance	<p>All Kotlin files contain no magic numbers/strings &amp; are stored in their appropriate XML files.</p> <p>All XML files contain no magic numbers/strings &amp; 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"> <li>• Correct use of intermediate variables, i.e., no method calls as arguments.</li> <li>• Idiomatic use of control flow, data structures &amp; other in-built functions.</li> <li>• Sufficient modularity, i.e., code adheres to various OO design principles.</li> <li>• Adhere to an OO architecture, i.e., classes, functions, concise naming &amp; functions assigned to the correct classes.</li> <li>• Efficient algorithmic approach.</li> </ul>	<p>Most Kotlin files contain no magic numbers/strings &amp; are stored in their appropriate XML files.</p> <p>Most XML files contain no magic numbers/strings &amp; are stored in their appropriate XML files.</p> <p>Application code mostly demonstrates code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Correct use of intermediate variables, i.e., no method calls as arguments.</li> <li>• Idiomatic use of control flow, data structures &amp; other in-built functions.</li> <li>• Sufficient modularity, i.e., code adheres to various OO design principles.</li> <li>• Adhere to an OO architecture, i.e., classes, functions, concise naming &amp; functions assigned to the correct classes.</li> <li>• Efficient algorithmic approach.</li> </ul>	<p>Some Kotlin files contain no magic numbers/strings &amp; are stored in their appropriate XML files.</p> <p>Some XML files contain no magic numbers/strings &amp; are stored in their appropriate XML files.</p> <p>Application code demonstrates some code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Correct use of intermediate variables, i.e., no method calls as arguments.</li> <li>• Idiomatic use of control flow, data structures &amp; other in-built functions.</li> <li>• Sufficient modularity, i.e., code adheres to various OO design principles.</li> <li>• Adhere to an OO architecture, i.e., classes, functions, concise naming &amp; functions assigned to the correct classes.</li> <li>• Efficient algorithmic approach.</li> </ul>	<p>Kotlin files contain frequent magic numbers/strings &amp; are not or are not fully stored in their appropriate XML files.</p> <p>XML files contain frequent magic numbers/strings &amp; are not or are not fully stored in their appropriate XML files.</p> <p>Application code does not does not fully demonstrate code elegance on the following:</p> <ul style="list-style-type: none"> <li>• Correct use of intermediate variables, i.e., no method calls as arguments.</li> <li>• Idiomatic use of control flow, data structures &amp; other in-built functions.</li> <li>• Sufficient modularity, i.e., code adheres to various OO design principles.</li> <li>• Adhere to an OO architecture, i.e., classes, functions, concise naming &amp; functions assigned to the correct classes.</li> <li>• Efficient algorithmic approach.</li> </ul>
---------------	---	---	---	---

<b>Documentation &amp; Git Usage</b>	Privacy policy thoroughly discloses user information collected by the application.	Privacy policy mostly discloses user information collected by the application.	Privacy policy briefly discloses user information collected by the application.	Privacy policy does not or does not fully disclose user information collected by the application.
	Wireframes thoroughly sketched & reflect the final application.	Application wireframes mostly sketched & reflect the final application.	Application wireframes briefly sketched & reflect the final application.	Application wireframes are not or are not fully sketched & do not or do not fully reflect the final application.
	Step-by-step user guide thoroughly describes each activity in detail.	Step-by-step user guide mostly describes each activity in detail.	Step-by-step user guide briefly describes each activity in detail.	Step-by-step guide does not or does not fully describe each activity in detail.
	Application code thoroughly commented with KDoc & generated with Dokka.	Application code mostly commented with KDoc & generated with Dokka.	Some application code commented with KDoc & generated with Dokka.	Application code is not or not fully commented with KDoc & generated with Dokka.
	Git branches thoroughly named with convention & contain the correct code relating to the functional requirement.	Git branches mostly named with convention & contain the correct code relating to the functional requirement.	Some git branches named with convention & contain the correct code relating to the functional requirement.	Git branches are not or are not fully named with convention & do not or do not fully contain the correct code relating to the functional requirement.
	Git commit messages thoroughly reflect the functional requirement changes.	Git commit messages mostly reflect the functional requirement changes.	Some git commit messages reflect the functional requirement changes.	Git commit messages do not or do not fully reflect the functional requirement changes.

# Assessment 01: Kotlin Travelling App

## Marking Cover Sheet

Name:

Date:

Learner ID:

Assessor's Name:

Assessor's Signature:

Criteria	Out Of	Weighting	Final Result
Functionality & Robustness	10	40	
Code Elegance	10	45	
Documentation & Git Usage	10	15	
Final Result			/100
This assessment is worth 35% of the final mark for the Design and Development of Application Mobile Devices course.			

Feedback: