Diploma in Software Engineering and Design
Assignment Cover Sheet

Course name: Diploma in Software Engineering and Design	**Student's name:**
Module Name /or number: Mobile Application Development (15 credits)	
Assignment title and/or number: DSED-03 Android Assessment (Hangman App)	
Assessment weighting	*6.25% of the overall programme.*
Due date:	**Date submitted**: (late submissions incur 10% penalty, after 7 days late, the assessment will not be marked)
Assessment conditions:	This is a resource-based assessment. This means that you may have access to any relevant resources to assist you. This could include, for example, your learning materials, information on the Internet, and so on. However, all work must be your own with no assistance from any other person.
Submission requirements:	You're required to submit the following into your assignment submission directory: • This document, completed where appropriate • Visual Studio / Xamarin project files • GitHub link below:
Passing Criteria:	Need to score 50 or more marks to pass the assessment. **Total Marks : 100**

Learning Outcomes:	• Demonstrate understanding of the architecture and components of a mobile platform • Use a development environment with software development kits appropriate to the mobile platform • Create and deploy simple applications for the mobile platform

Assignment Checklist:

Requirement	**Completed**
Database	✓✗
User interface	✓✗
Functionality	✓✗
Coding	✓✗
Testing	✓✗
Documentation	✓✗
Github Upload	✓✗

Disclaimer of Plagiarism and Collusion

I declare that, to the best of my knowledge, this assessment is my own work, and has not been copied from any other student's work or from any other source.
Enter your name here to indicate you agree to the above statement.

Hangman
Requirement

The assessment consists of designing a Hangman game for the Android platform.

Functionality Requirement

1) Simulate the Hangman game
2) Store the list of words in a SQLite database
3) Set up a scoring system
4) Store user high scores in the SQLite Database

User Interface Requirement

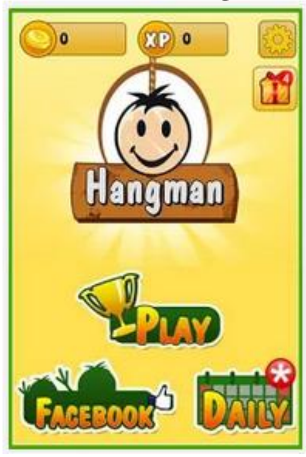
1. Design your UI for different screen sizes
2. Design a splash screen for your app
3. Design an Icon for your app

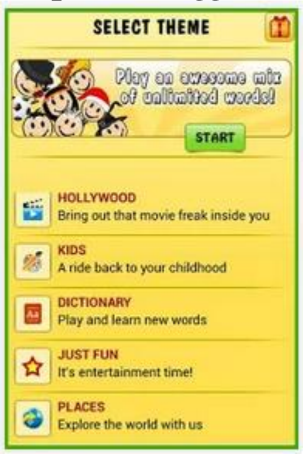
Platform Requirement

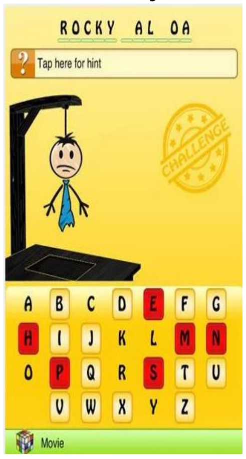
1. Your App needs to support Android 4.4 and up

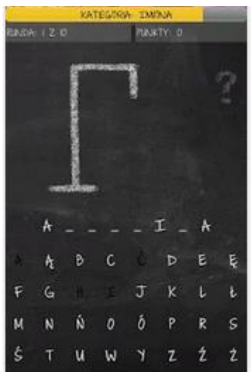
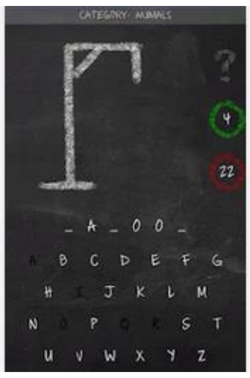
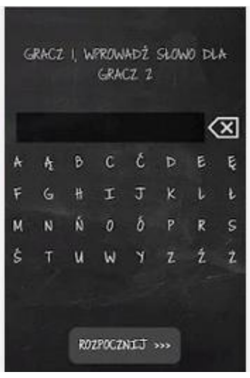
UI Design

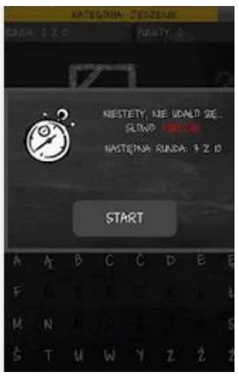
Below is the sample UI design for the app. You can choose your color combination and layout




SELECT THEME
Play an awesome mix of unlimited words!
START
HOLLYWOOD
Bring out that movie freak inside you
KIDS
A ride back to your childhood
DICTIONARY
Play and learn new words
JUST FUN
It's entertainment time!
PLACES
Explore the world with us





Testing Requirements

Kindly fill in the testing sheet below with your comments.

Below table lists, some common quality checks you need to perform before uploading your app on the Google play store.

	Test Condition	**Result (Pass/Fail)**	**Comments**
App Design	Navigate to all parts of the app — all screens, dialogs, settings, and all user flows.		
	Navigate to all parts of the app — all screens, dialogs, settings, and all user flows 1. App displays graphics, text, images, and other UI elements without noticeable distortion, blurring, or pixilation. 2. App provides high-quality graphics for all targeted screen sizes and form factors, including for larger-screen devices such as tablets. 3. No aliasing at the edges of menus, buttons, and other UI elements is visible. 4. Composition is acceptable in all supported form factors, including for larger-screen devices such as tablets. 5. No cut-off letters or words are visible. 6. No improper word wraps within buttons or icons are visible. 7. Sufficient spacing between text and surrounding elements.		
App State App preserves user or app state when leaving the foreground and prevents accidental data loss due to back-navigation and other state changes. When returning to the	From each app screen, press the device's Home key, then re-launch the app from the All Apps screen. When the app is re-launched from Home or All Apps, the app restores the app state as closely as possible to the previous state		
	From each app screen, switch to another running app and then return to the app under test using the Recent's app switcher.		

foreground, the app must restore the preserved state and any significant stateful transaction that was pending, such as changes to editable fields, game progress, menus, videos, and other sections of the app or game.	When the app is resumed from the Recent's app switcher, the app returns the user to the exact state in which it was last used.		
	From each app screen (and dialogs), press the Back button. On Back key presses, the app gives the user the option of saving any app or user state that would otherwise be lost on back-navigation.		
	From each app screen, rotate the device between landscape and portrait orientation. The App should correctly preserve and restore user or app state.		
Stability and Performance	App does not crash, force close, freeze, or otherwise function abnormally on any targeted device. App loads quickly or provides onscreen feedback to the user (a progress indicator or similar cue) if the app takes longer than two seconds to load.		

<u>Marking Schedule</u>

Parameters	
Program Functionality	**50**
App design and UI look	**10**
Coding standards	**20**
Creativity	**10**
Testing	**10**

Marking Criteria

% of Grade	Excellent 80-100%	Adequate 60-80%	Poor 50 - 60%	Not Met 0 - 50%
Program Functionality 50%	No errors, program always works correctly and meets the specification	Minor details of the program specification are violated, program functions incorrectly for some inputs.	Significant details of the specification are violated, program often exhibits incorrect behavior.	Program only functions correctly in very limited cases or not at all
App design and User Interface 10%	Simple easy to use intuitive UI, no errors spelling mistake and good color schemes used	Minor errors with the UI, minor layout issues	Major UI errors making it hard to understand	Significant UI errors with no logical sense and program crash
Coding Standards 20%	No errors, code uses the best approach in every case and follows the coding standards	Minor errors or repetition of code, coding and naming standards not followed in some occasions	Code uses poorly chosen approaches in some places. Naming standards not followed in some places.	Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion. Poor naming and coding standards
Creativity 10%	Creative approach used for the app design, layout and functionality	Some creative aspects in the app design, layout and functionality	Not much creativity, the app is basic in terms of design, layout and functionality	NA
Testing 10%	Program is well tested to identify and fix bugs and errors. No major bugs or defects in the program. Testing results matches the actual program.	Program is well tested to identify most of the bugs, but some bugs still exist. Some test cases marked pass which are false	Program has a lot of major bugs and is not tested. Testing sheet incorrect or incompletely filled out.	Program is not at all tested and testing sheet is not filled.