



UNIVERSITY OF  
MPUMALANGA

Creating Opportunities

**DICT 311: Application Programming 311**

**Lecturer: Dr. Olalekan Samuel Ogunleye**

**Moderator: Dr Blessings Maoneke**

Total Marks: 100

**Due Date: May 19<sup>th</sup>, 2025**

## Instructions

1. This is not a group assignment project and not an individual project.
2. You are required to conduct further research to be able to perform the task of this assignment accordingly. However, plagiarism is not allowed in any form. Your code will be subjected to a plagiarism check. All the group members will be awarded zero if it is found that they have plagiarised.
3. This assignment could be time-consuming and require dedicated effort. You only have 5 days to work on this project. It is about building a simple functional system while incorporating the fundamental knowledge and understanding of the Android Kotlin framework.
4. You must add a plagiarism declaration to your submission, where you will declare that your submission is not plagiarized. You must sign the declaration.
5. The project deadline is sacrosanct. This deadline will not be extended.
6. This project will contribute 17.5%, tests and other formative assessments will contribute 33.5%, and the examination will carry 50% toward the final Mark. Therefore, you are advised to take the project seriously.
7. Project documentation of not less than two pages should accompany the submission. This should be submitted online.
8. You will not need to present this assignment. However, a short video of how your system works is needed.
9. Should you require further clarification on this assignment, a physical class is required and can be scheduled.

## Project Brief

As an enterprise software developer, you just joined Dudu and Masuku (D&M) Software Development Company. The University of Mpumalanga has contracted your company to manage the institution's information system implementation. In your first task on your first day of work in your company, you are required to develop a simple Task Management System that the institution's students can use to manage their simple daily tasks. You are required to develop a simple Task Management App using Android and Kotlin. The app should allow users to add, display, and delete tasks. The application should incorporate object-oriented programming, event handling, networking, and multimedia features.

The table below Highlights what must be developed based on the above brief.

1.1)	<b>Project Setup and UI Design</b> <ul style="list-style-type: none"><li>- Design a basic UI in Android Kotlin with:</li><li>- An EditText for entering tasks.</li><li>- A RecyclerView for displaying tasks.</li><li>- A Button to add new tasks.</li><li>- A Button to delete completed tasks.</li></ul>	(10)
1.2)	<b>Implementing an Object-Oriented Approach</b> <ul style="list-style-type: none"><li>- Create a Task class with properties (title, description, and completion status).</li><li>- Implement a TaskManager class that manages task storage and retrieval.</li><li>- Use a ViewModel for data persistence</li></ul>	(15)
1.3)	<b>Implementing Event Handling in Android</b> <ul style="list-style-type: none"><li>- Add a click event listener to handle task addition.</li><li>- Add an event listener for marking a task as completed.</li><li>- Implement a long-press listener to delete tasks.</li></ul>	(15)
1.4)	<b>Basic Graphics and Multimedia Support</b> <ul style="list-style-type: none"><li>- Add an image next to each task to indicate completion status.</li><li>- Play a notification sound when a task is added.</li><li>- Display a toast message when a task is deleted.</li></ul>	(15)
1.5)	<b>Developing a Network-Supported Feature</b> <ul style="list-style-type: none"><li>- Implement AsyncTask or Kotlin Coroutines to fetch task suggestions from an API.</li><li>- Use HttpURLConnection or Retrofit to send a request and display the response.</li></ul>	(15)
1.6)	<b>Sending Emails and SMS Notifications</b> <ul style="list-style-type: none"><li>- Implement a feature to send an email when a task is overdue.</li><li>- Implement a feature to send an SMS reminder for incomplete tasks. (4)</li></ul>	(15)
1.7)	<b>Implementing Database Features</b>	(15)

	<ul style="list-style-type: none"><li>- Store tasks using the Room Database for persistence. (3)</li><li>- Implement Dark Mode support for the app. (3)</li><li>- Add a search bar to filter tasks by title. (3)</li></ul>	
--	--	--

## Assignment Rubric

Your Project assignments will be graded according to the following Rubric.

Focus Area	Description	Marks
<b>Completeness</b>	This score will be determined by the result of running a series of automated tests on your code. The tests check whether your code implements the tasks you are asked to implement in the project and, thus, are a good measure of how <i>complete</i> your code is. You can run these tests yourself and, therefore, know your Completeness score before you submit your assignment.	30%
<b>Correctness</b>	This score encompasses issues with your code that, while not explicitly captured by the tests, could lead to incorrect behaviour (or simply neglect to implement something you were told to implement). I will usually provide hints on what is being looked out for in this rubric component. Please note that this rubric section will never re-penalize you for a failure already captured by the tests. So, if your code fails to implement something we told you to implement, but the tests already capture that, we would not re-penalize you here.	25%
<b>Design</b>	This score will be determined by "qualities, many of which are intangible, that don't have to do with (and exist to some extent independently of) the correct operation of your code." For example, you will have to decompose your code into multiple functions, where that decomposition is	25%

	up to you. Whether your decomposition makes sense and is well-designed will be evaluated in this score.	
<b>Style</b>	<p>The following will be looked at generally in awarding the marks:</p> <p><i>Documenting functions:</i> Every function you write must document what the function performs and what it is doing.</p> <p><i>Clarity:</i> Your code must be easy to read and understand. This is a fairly subjective aspect, but common deductions for lack of clarity include using variables without descriptive names, writing convoluted code that would be easier to understand with more code comments, using magic numbers, etc.</p>	20%
	<b>Total</b>	<b>100%</b>
<p><b>Note that submitting broken code (i.e., code with syntax errors or, generally, code that cannot be run) will receive an automatic 40-Marks penalty (out of 100 possible points). Please note that if your code is so broken that it cannot work with a few simple fixes, it will not get graded, and you will receive a zero on the project assignment.</b></p>		