



College of Engineering, Construction and Living Sciences  
Bachelor of Information Technology  
IN721: Mobile Application Development  
Level 7, Credits 15  
**Practical 04: NASA API**

## Assessment Overview

In this assessment, you will design, develop & UI test an application which requests data from the **NASA API** & displays the data using a **RecyclerView**. This assessment contributes **3%** towards your final mark in **IN721: Mobile Application Development**.

## Learning Outcomes

At the successful completion of this course, learners will be able to:

1. Implement & publish complete, non-trivial, industry-standard mobile applications following sound architectural & code-quality standards.
2. Identify relevant use cases for a mobile computing scenario & incorporate them into an effective user experience design.
3. Follow industry standard software engineering practice in the design of mobile applications.

## Assessment Table

Assessment Activity	Weighting	Learning Outcomes	Assessment Grading Scheme	Completion Requirements
Practical	20%	2, 3	CRA	Cumulative
Project	80%	1, 2, 3	CRA	Cumulative

## Conditions of Assessment

You will complete this individual assessment inside & outside timetabled class time. This assessment will need to be completed by **Friday, 16 April 2021 at 5:00 PM**.

## Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **IN721: Mobile Application Development**.

## Authenticity

All parts of your submitted assessment must be completely your work & any references must be cited appropriately including, externally-sourced graphic elements. Provide your references in a **README.md** file. All media must be royalty free (or legally purchased) for educational use. Failure to do this will result in a mark of **zero** for this assessment.

## Policy on Submissions, Extensions, Resubmissions & Resits

The school's process concerning submissions, extensions, resubmissions & resits complies with **Otago Polytechnic** policies. Learners can view policies on the **Otago Polytechnic** website located at <https://www.op.ac.nz/about-us/governance-and-management/policies>.

## Submissions

You must submit all program files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – <https://classroom.github.com/a/VJlq7Ae0>. Create a new branch called **04-nasa-api** from the **main** branch by running the command - **git checkout -b 04-nasa-api**. This branch will be your development branch for this assessment. Once you have completed this assessment, create a pull request & assign the **GitHub** user **grayson-orr** to a reviewer. **Do not** merge your own pull request. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**.

## Extensions

Familiarise yourself with the assessment due date. If you need an extension, contact the course lecturer before the due date. If you require more than a week's extension, a medical certificate or support letter from your manager may be needed.

## Resubmissions

Learners may be requested to resubmit an assessment following a rework of part/s of the original assessment. Resubmissions are to be completed within a negotiable short time frame & usually must be completed within the timing of the course to which the assessment relates. Resubmissions will be available to learners who have made a genuine attempt at the first assessment opportunity & achieved a **D grade (40-49%)**. The maximum grade awarded for resubmission will be **C-**.

## Resits

Resits & reassessments are not applicable in **IN721: Mobile Application Development**.

## Instructions - Learning Outcomes 2, 3

### Task One (2%):

Create a new project with the following configurations:

- Template - Empty activity

- Name - NASA
- Package name - op.mobile.app.dev.nasa
- Save location - /path to your practical GitHub repository/04-nasa
- Language - Kotlin
- Minimum SDK - API 28: Android 9.0 (Pie)

The application structure is similar to the code examples from the **13-recycler-view** teaching session. Instead, you will request data from the **NASA API**.

To use the **NASA API**, you must generate an **API key**. To do this, go to <https://api.nasa.gov> & sign up. This **API key** allows you to access & use **web services** available on the **Data.gov** developer network.

Familiarise yourself with the following **API URL**:

[https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&page=1&api\\_key=DEMO\\_KEY](https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?sol=1000&page=1&api_key=DEMO_KEY)

**Note:** you will replace **DEMO\_KEY** with your generated **API key**.

It is important that you **do not** expose your **API key** publicly, i.e., in a public GitHub repository. To store your **API key** safely, declare it in **local.properties**. **Note:** pay careful attention to the comments.

In **recycler\_view\_item.xml**, add two **TextViews** & an **ImageView**. This will display the camera's name & full name as well as the image source. Set **recycler\_view\_item.xml** to the **RecyclerView's listitem** attribute in **fragment\_nasa.xml**.

Run your application on either an **Android Emulator** or **connect device**.

¡ADD IMAGE HERE!

### Task Three (1%):

Create a new test file called **NASATest**. To do this, right-click on **op.mobile.app.dev.nasa (androidTest)** > **Kotlin Class/File**. In **NASATest.kt**, write three UI tests. To run your test file, right-click **NASATest.kt** > **'Run NASATest'**.