

Task

This is an individual task with a fixed deadline -- no submissions will be accepted after the closing date.

If Turnitin indicates 25% or higher similarity (or a teacher detects a similar level of similarity) with other published work and/or work submitted by other students, the submission will be reviewed and a mark potentially withheld or delayed. Similar sections will **not** be assessed, i.e., if the introduction is copied from elsewhere, the introduction criteria will be awarded 0.

- PDF: A completed report for this task -- the report should set out your development plan/time logs, the key design decisions made, issues encountered and the solutions considered/used, and a reflection on what worked well/what could be improved. Please ensure that explanation/decisions relating to the rubric elements are covered. Any use of genAI must be acknowledged, with prompts and outputs included in an appendix. If genAI was not used, the statement "No Generative AI tools were used for this task" must be included in the acknowledgement. Please refer to the Unit Outline for more details around resubmission/redo of work where the teaching staff feel that genAI has been inappropriately used.
- The report **must** contain a link to code in a repository visible to teaching staff on GitHub Classroom (see [GitHub Classroom links](#) for the invite link). Repos are not to be public, nor should tutors be added manually -- **the task will not be assessed until the repo is in a suitable location.**
- Before or after submission, a demo in class is required, otherwise a penalty will be applied. The demo will be assessed according to the [Demo guidelines](#). Note that these will need to be spread over 2-3 weeks of classes; it is expected that students aiming for high marks will demo in the first weeks, as this will allow you to take on feedback ahead of submission.

Instructions

1. Start with planning and researching. This will include working through the tutorial exercises and quizzes. Outline and describe the major things you will need to do to create your app.
2. Develop an app that meets criteria as shown below.
3. The report needs to cover key knowledge gaps and process that was undertaken in this task, however note that this should be pitched at a fellow student (e.g., "Go to the Android website and download Android Studio" is not required). A reflection on Assignment 1 and any changes made for Assignment 2 would also be appropriate.
4. Submissions must be uploaded as a single PDF.

The app

A small music studio is branching out into monthly rentals of musical instruments and equipment. They are seeking a simple proof of concept app for clients to use.

- The app needs to show 3-4 items that can be rented by the studio. Note you can deal in "credit", not dollars necessarily. Each item should be shown one at a time; a "next" button is needed for

progressing through items. Assume that the app allows for immediate pickup only; there is no concept of "future" bookings.

- The data kept for each item should include the name, a rating from 0-5, a multi-choice attribute, and the "price" per month.
 - You will need to create a model for this -- note data should not be stored to disk, rather kept in memory.
 - The name and other details should be shown under each image in the first activity.
 - At least one of your attributes should use a non-TextView widget, such as a RatingBar, a Switch, or a RadioGroup; students looking for a challenge should make use of a multi-choice widget, such as a Chip.
- In module 5 we looked at UX. Please include 2 user stories and use cases for your app, along with **sketches** of two possible layouts for your app screens, noting the required widgets. Please note that this does not mean "pretty" designs should be made, nor full prototypes -- it is simply testing whether you can use sketching tools to think about and communicate ideas about your app's UI. Consider different UI elements for each of the two layouts and justify your choices.
- Use at least two styles in two locations -- that is, don't make a style just for one view, use it in multiple locations. This could be for defining text size, style, colour or something else.
- When the "borrow" button is clicked, a new screen with further details needs to be shown, potentially along with the image. The data needs to be passed as a Parcelable object. In your report, explain the components of your Intent and the advantages of using Parcelable objects on Android.
- The new details should be saved on pressing save (pressing back should be considered a cancellation), and should be shown to the user in the first activity in some way. Do not use persistent data for this task.
- Error checking should be included for required/incorrect fields -- this could be checking required details, not going over credit etc. This must stop the user from returning to the first activity until the error is fixed or they cancel the booking.
- This is a good app for UI testing. Note that RatingBars are not easy to test with Espresso so you should focus on text and buttons. If you use Espresso then it is expected that the code be edited to remove redundancy.
- When an item is booked or not booked (e.g., the booking action is cancelled), a Toast or Snackbar is required to denote this. Discuss your design choice in your report.
- Note that while some concepts from A1 are not explicitly required for A2, they may still be useful for this task and could be implemented with justification.

Disallowed concepts for this task: any form of persistent data, any form of RecyclerView. TabLayout may only be used if the tabs contain a *category* of instruments (i.e., not a single instrument per tab) and the instruments are booked in a way that meets the specified requirements.

Resources

Weeks 1-6 have set you up by isolating different parts of app development needed for this task.

Some further resources that might be useful include:

- For more information on RatingBar, for those who want to use it, see <https://developer.android.com/reference/android/widget/RatingBar>. In order to constrain the number of stars shown, check the layout_width setting.
- For using a Slider, see <https://m3.material.io/components/sliders/overview>
- [Links to an external site.](#).
- For information on Toasts, see <https://developer.android.com/guide/topics/ui/notifiers/toasts>. For Snackbars, see <https://developer.android.com/develop/ui/views/notifications/snackbar>
- [Links to an external site.](#).
- See <https://developer.android.com/guide/topics/ui/look-and-feel/themes> for hints on how to use styles.
- Advanced students might wish to explore the use of Fragments, however please note that demonstration of communication between fragments/activities is required along with a written comparison to activity-activity communication. You may wish to use Intents for a feature not listed above (e.g., email confirmation).