

This assignment does not count toward the final grade.

Assignment 2

Start Assignment

- Due 23 Sep by 23:59
- Points 25
- Submitting a file upload
- File types pdf
- Available until 1 Oct at 23:59

New task for 2024

Please note that assessment has changed for this unit in 2024, and as such revisions might need to be made to the spec. Students are required to be checking this page, the discussion forums and lecture discussions in case of changes.

While this task states "This assignment does not count toward the final grade", it actually will, however students are only permitted to be awarded a single overall mark in portfolio units.

Assessment overview

- **Title:** Assignment 2: Rent with Intent
- This task addresses the following Unit Learning Outcomes:
 - ULO1 Explain the key differences between development of systems to run on mobile devices and on typical personal computing or internet-based environments, and apply this knowledge in the design of mobile device software.
 - ULO2 Design effective applications for a mobile device by taking into consideration the underlying hardware-imposed restrictions such as screen size, memory size and processor capability.
 - ULO3 Build, test and debug graphical applications for mobile devices by using the standard libraries that are bundled as part of the developers' toolkit for the mobile device.
- **Individual assessment**
- **Assessment type:** Written report and code
- **Weighting:** 25%
- **Word count (or equivalent):** no limit
- **Due date:** Please see Canvas submission space.
- **Submission procedure:** For this assignment you may submit as many times as you like before the due date. Your assignment will be checked for plagiarism with a similarity report generated through Turnitin. You will be able to review the Similarity report after the due date. The teaching staff can see each of your submissions, and the associated Similarity Report. Should you choose to submit more than once, the teaching staff will be marking the final uploaded file.

- **Feedback date:** 2 weeks after due date.
- **Referencing style:** APA 7
- **Who will see your work?** This work will be assessed by your tutor.

Assessment explanation

This app is a two-activity app that requires the use of intents.

On completion of this task, you will demonstrate that you are able to create a multi-activity app, share data between activities using intents, use images and resources, and work with more complicated UI widgets. You will also demonstrate that you are able to work with sharing data between activities in an advanced manner and take input from the user.

! Important Academic integrity

It is your responsibility to make sure your work is your own. If you provide work to another student you will be penalised the same as the student who uses your work. This includes failure of the assignment, an official penalty notice, and possible exclusion from the University.

Paying someone to do your assignment for you is one of the highest forms of Academic Misconduct and it will not be tolerated. Possible penalties for this include exclusion from the university and revocation of your degree.

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](https://swinburne.instructure.com/courses/61570/pages/github-classroom-links) (<https://swinburne.instructure.com/courses/61570/pages/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](https://swinburne.instructure.com/courses/61570/pages/demo-guidelines) (<https://swinburne.instructure.com/courses/61570/pages/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> (<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> (<https://m3.material.io/components/sliders/overview>).
- For information on Toasts, see <https://developer.android.com/guide/topics/ui/notifiers/toasts> (<https://developer.android.com/guide/topics/ui/notifiers/toasts>). For Snackbars, see <https://developer.android.com/develop/ui/views/notifications/snackbar> (<https://developer.android.com/develop/ui/views/notifications/snackbar>).
- See <https://developer.android.com/guide/topics/ui/look-and-feel/themes> (<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).

Late submissions and extensions

Please refer to the unit outline for late penalty calculations. Note for this assignment, **0%** is being applied per late day up to 5 working days -- this means a request for an extension up to 5 working days is **not** required. However, no further extensions beyond 5 working days will be granted without special consideration. **Submitting after the due date will take away time from later assessments, risk not being able to demo in class (which will incur a penalty), and delay your feedback -- tutors will be asked to prioritise marking timely submissions.**

COS30017 A2

Criteria	Ratings				Pts
UI/UX Does the app look good? Does it provide a good experience for the user?	7 to >5.0 Pts Excellent Evidence of thought (incl. research/references) incorporated into UI/UX. App provides excellent experience for mobile users.	5 to >3.5 Pts Good Thought put into UI/UX. App provides good experience for mobile users, however some minor	3.5 to >1.5 Pts Needs improvement A solid start, however UI/UX require major improvement, in particular thinking about mobile	1.5 to >0 Pts No or minimal attempt UI/UX require major improvement, in particular thinking about mobile users.	7 pts
Functionality Is the expected functionality implemented? Have debugging and testing been used to confirm correct behaviour?	7 to >5.0 Pts Excellent All required functionality implemented correctly, with thorough testing and debugging to confirm this. Extra functionality (commensurate with course progress and	5 to >3.5 Pts Good All required functionality implemented correctly, with some testing and debugging to confirm this. Testing and debugging need	3.5 to >1.5 Pts Needs improvement Most required functionality implemented correctly, with some testing and debugging used. Testing and debugging need	1.5 to >0 Pts No or minimal attempt Little to no functionality implemented correctly. No use of testing or debugging.	7 pts
Code quality Has care been taken with the code itself? Is the structure of the project sensible?	7 to >5.0 Pts Excellent Excellent attention to code style, including sufficient comments and documentation. Good use of Kotlin-specific features to produce an elegant solution. Project structure conforms to Android style in all aspects.	5 to >3.5 Pts Good Good attention to code style, including some comments and documentation, however minor improvements could be made. Some elegant code however inconsistently	3.5 to >1.5 Pts Needs improvement Some attention to code style, however major improvements could be made wrt e.g., documentation, formatting and variable naming.	1.5 to >0 Pts No or minimal attempt Little to no attention to code style. Project structure does not conform to Android style.	3 pts
Process and demo Is there evidence of a structured software development process? Has a demo been undertaken with teaching staff?	7 to >5.0 Pts Excellent Evidence of a sensible and structured process, including time logs and commits with succinct descriptions. Report matches product and commit history. Demo is assessed at A/B level.	5 to >3.5 Pts Good Some evidence of a sensible and structured process, including time logs and commits, however with minor improvements required. Report matches product and commit history. Demo is assessed at B/C level.	3.5 to >1.5 Pts Needs improvement Little evidence of a sensible and structured process, possibly including a lack of commit history or poor commit messages. Report mostly matches product and commit history. Demo is assessed at C/D level.	1.5 to >0 Pts No or minimal attempt Little to no evidence of a sensible and structured process, e.g., zip file submitted. Report does not match product and commit history. Demo is	8 pts
Total points: 25					

assessed at
D/E/F level.