**Assignment 2: Rent a Car with Intent**

*(COS30017 – Mobile Computing)*

**Student Name:** [Your Name]  
**Student ID:** [Your ID]  
**Date:** [Submission Date

Table of Contents

[**1.** **Introduction** 1](#_Toc210300650)

[**2.** **Planning and Research** 1](#_Toc210300651)

[2.1. Objectives 1](#_Toc210300652)

[2.2. Research Highlights 1](#_Toc210300653)

[2.3. Development Tools 1](#_Toc210300654)

[**3.** **Design Decisions** 1](#_Toc210300655)

[3.1. User Interface 1](#_Toc210300656)

[3.2. Widget Choices Table 5](#_Toc210300657)

[3.3. Parcelable Usage 5](#_Toc210300658)

[**4.** **Development Logs** 6](#_Toc210300659)

[4.1. Time Log Table 6](#_Toc210300660)

[**5.** **Testing** 7](#_Toc210300661)

[5.1. Test Strategy 7](#_Toc210300662)

[5.2. Test Results Table (Integrated Credit Balance Tracking) 7](#_Toc210300663)

[**6.** **Challenges, Explorations & Takeaways** 8](#_Toc210300664)

[6.1. Challenges 8](#_Toc210300665)

[6.2. Explorations 9](#_Toc210300666)

[6.3. Takeaways 9](#_Toc210300667)

[**7.** **Reflections on Assignment 1 & Changes for Assignment 2** 9](#_Toc210300668)

[**8.** **Conclusion** 9](#_Toc210300669)

[Figure 1 - Main Activity - Car List and Favorites 2](#_Toc210467573)

[Figure 2 - Rent Activity - Car Selection and Rental Duration 3](#_Toc210467574)

[Figure 3 - Rent Activity - Customer Information and Price Summary 4](#_Toc210467575)

[Figure 4 - Car Parcelable class with attributes and helper methods 5](#_Toc210467576)

[Figure 5 - Intent passing the selected car, credit balance, maximum rental cost, and dark mode state to BookingActivity using Parcelable objects. 6](#_Toc210467577)

[Table 1 - Widget Choices Table 5](#_Toc210467619)

[Table 2 - A summarised table of the development activities and durations 6](#_Toc210467620)

[Table 3 - The outcomes of the test cases executed for functional verification 7](#_Toc210467621)

# **Introduction**

This report documents the development of a proof-of-concept car rental mobile app that demonstrates multi-activity usage, intent-based data sharing, UI/UX considerations, and user interaction handling on Android. The app enables users to browse, rent, and manage cars while respecting credit limits, mark favourites, search, sort, and toggle between light and dark modes.

The report includes design decisions, development logs, testing outcomes, and reflections. All work adheres to Android best practices, with a special focus on memory and hardware limitations typical of mobile environments.

# **Planning and Research**

## Objectives

* Develop a two-activity mobile app using Intents for data sharing.
* Demonstrate Parcelable object usage for efficient inter-activity data transfer.
* Implement dynamic UI widgets such as RatingBar, Switch, and Slider.
* Ensure credit limit handling, favorites tracking, search, and sort functionalities.
* Integrate dark mode toggling across multiple screens.

## Research Highlights

* Studied Android documentation for Parcelable objects to optimize memory usage compared to Serializable.
* Reviewed best practices for RecyclerView, RatingBar, and EditText input validation.
* Explored light/dark mode themes using styles and resources.
* Investigated Espresso testing for automated UI testing and validation.

## Development Tools

* Android Studio Narwhal Feature Drop | 2025.1.2 Patch 2
* Kotlin 2.0.21
* Material Design Components
* GitHub Classroom for version control and code submission
* Emulated device Pixel 9 Pro API 36.0

# **Design Decisions**

## User Interface

* Main activity displays one car at a time with image and details.
* Next button cycles through the car list.
* Rent button opens second activity for booking.
* Credit balance displayed at the top with a TextView.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Figure 1 - Main Activity - Car List and Favorites

This screen displays the main activity of the app, where users can view one car at a time, browse using the “Next” button, and see their credit balance at the top. Users can search for cars by name or model, sort by rating, year, or cost, and toggle between Light and Dark mode. Favorite cars are listed in a separate section for quick access. The wireframe (left) illustrates layout placement, while the high-fidelity mockup (right) shows the styled app interface.

Screens screenshot of a phone

AI-generated content may be incorrect.

Figure 2 - Rent Activity - Car Selection and Rental Duration

This screen shows the Rent Activity where the user sees detailed information about the selected car, including name, year, cost, and special features (e.g., high rating, low mileage). The slider allows users to select rental duration from 1 to 7 days. The wireframe (left) shows input layout and widget placement, while the high-fidelity mockup (right) demonstrates the fully styled interface.

Screens screenshot of a phone

AI-generated content may be incorrect.

Figure 3 - Rent Activity - Customer Information and Price Summary

This screen allows users to input their personal information (full name, email, phone, driver license, and age) and choose additional options such as full insurance coverage. A price summary is displayed showing the base cost, insurance, tax, total cost, and updated credit balance. Wireframe (left) highlights layout and structure, while the high-fidelity mockup (right) shows final app styling.

## Widget Choices Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Widget** | **Purpose** | **Justification** | **Locations Used** |
| RatingBar | Display car rating | Interactive, visually intuitive | Main Activity |
| Switch | Toggle between light/dark mode | User-controlled theme change | Main & Rent Activity |
| Slider | Number of rental days (1–7) | Prevents invalid input | Rent Activity |
| Button | Navigation / Rent | Standard Android action trigger | Main & Rent Activity |
| Toast/Snackbar | Booking success/error | Immediate feedback | Both Activities |

Table 1 - Widget Choices Table

## Parcelable Usage

* All car details are encapsulated in a Car Parcelable object.

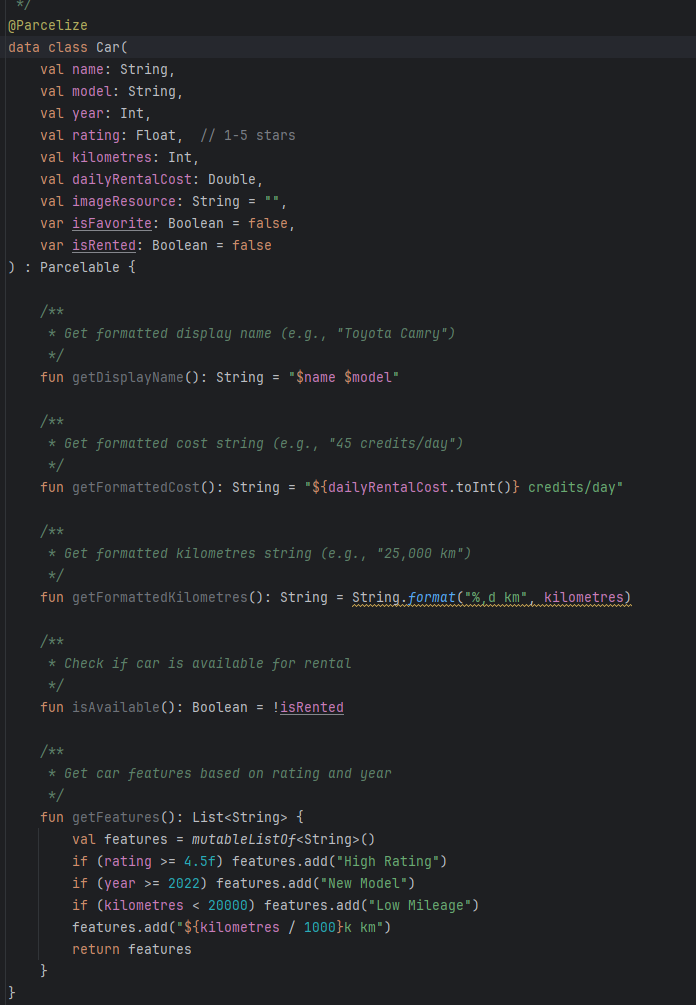


Figure 4 - Car Parcelable class with attributes and helper methods

The Car class encapsulates all car-related data and provides utility functions to format the display name, cost, and kilometres, check availability, and generate features based on rating, year, and mileage.

* Advantages: faster than Serializable, lower memory overhead, optimized for Android IPC.
* Intent example:

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 5 - Intent passing the selected car, credit balance, maximum rental cost, and dark mode state to BookingActivity using Parcelable objects.

# **Development Logs**

## Time Log Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Task** | **Duration** | **Notes** |
| 2025/09/26 | Project setup, Android Studio configuration | 2h | Created repository and initial project structure |
| 2025/09/27 | Car model & data class creation | 3h | Defined Parcelable objects and in-memory dataset |
| 2025/09/28 | UI design: main activity | 4h | Implemented car display, Next button, RatingBar |
| 2025/09/29 | Rent activity UI & Slider widget | 3h | Added rental days selection, input validation |
| 2025/09/30 | Favorites & Search functionality | 4h | Implemented heart icon, search filtering, sort menu |
| 2025/10/01 | Dark mode toggle & styling | 3h | Applied styles for multiple widgets/screens |
| 2025/10/02 | Testing, debugging, Espresso setup | 5h | Verified credit balance, bookings, and UI consistency |

Table 2 - A summarised table of the development activities and durations

# **Testing**

## Test Strategy

* Manual testing for navigation and UI consistency
* Automated testing using Espresso for text and button interactions
* Credit validation and error handling
* Favorites, search, and sort functionality

## Test Results Table (Integrated Credit Balance Tracking)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Outcome** | **Actual Outcome** | **Pass/Fail** |
| Rent car under balance | 200 credits | Booking succeeds, balance updated | Matches expectation | Pass |
| Rent car exceeding balance | 600 credits | Booking blocked, error message | Matches expectation | Pass |
| Cancel booking | Cancel pressed | Car available again, balance unchanged | Matches expectation | Pass |
| Mark favorite | Tap heart icon | Car appears in favorites list | Matches expectation | Pass |
| Search car by name | Enter “Civic” | Filtered list shows Civic | Matches expectation | Pass |
| Sort by rating | Select “High to Low” | Cars ordered by rating descending | Matches expectation | Pass |
| Dark mode toggle | Switch enabled | UI theme changes to dark mode | Matches expectation | Pass |
| Number of rental days | Slider 1–7 | Only valid selection allowed | Matches expectation | Pass |

Table 3 - The outcomes of the test cases executed for functional verification

# **Challenges, Explorations & Takeaways**

## Challenges

* Handling Parcelable objects efficiently while passing multiple fields.
* Integrating dynamic UI widgets with real-time data updates.
* Implementing credit limit validation and preventing invalid bookings.
* Testing RatingBars with Espresso required careful planning due to widget limitations.

## Explorations

* Explored multiple UI themes for light/dark mode.
* Experimented with search & sort algorithms to handle in-memory data efficiently.
* Investigated Toast vs Snackbar for user feedback.

## Takeaways

* Parcelable is faster and more memory-efficient than Serializable in Android.
* Planning UI widgets early saves significant refactoring time.
* Effective logging and commit messages improve traceability.
* Espresso testing is valuable for ensuring UI stability but requires widget-specific handling.

# **Reflections on Assignment 1 & Changes for Assignment 2**

* Assignment 1 focused on single-activity apps and simple interactions.
* Assignment 2 expands to multi-activity with advanced features:
  + Intent-based data transfer
  + Dynamic widgets (RatingBar, Switch, Slider)
  + Favorites, search, and sort
  + Dark mode
  + Credit tracking

Lessons Learned:

* Handling multi-activity data flow requires careful object modeling.
* User experience improves when errors are caught early and feedback is immediate.

# **Conclusion**

The app demonstrates a functional car rental experience with:

* Efficient multi-activity data sharing using Parcelable objects
* Responsive UI with RatingBar, Switch, and Slider
* Error checking and credit management
* Favorites, search, and sort functionality
* Light and dark mode themes

Development logs and test results confirm the app meets the specifications. Remaining improvements could include persistent data storage and more comprehensive Espresso coverage for all widgets.

**Appendices**

Appendix A – Acknowledgement of Generative AI Use

No Generative AI tools were used for this task.

Appendix B – Screenshots / Images

Booking Activity Test Results:

A screenshot of a computer

AI-generated content may be incorrect.

Main Activity Test Results:

A screenshot of a computer

AI-generated content may be incorrect.

App with the dark mode enabled:

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a car

AI-generated content may be incorrect.

Appendix C – Source Code Link

[Insert GitHub Classroom repository link]