UNIVERSITY OF RWANDA

COLLEGE OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER ENGINEERING YEAR 3

Mobile Application Systems and Design

Flutter Lab3

Prepared by: Ibyishaka Ruth 223014217

Tuyisenge Ange Aurele 223015798

25th/2/2026

1. Introduction

This project is a multi-screen Travel Application UI developed using the Flutter framework. The application was designed as a user-interface–focused mobile system that demonstrates advanced layout structuring, reusable widget composition, responsive design techniques, and structured navigation across multiple screens.

The primary objective of the application is to showcase the practical implementation of complex Flutter UI components without relying on external APIs or databases. All data used within the system is hard-coded, allowing the emphasis to remain on front-end architecture, state management, and user experience design rather than backend integration.

The system demonstrates:

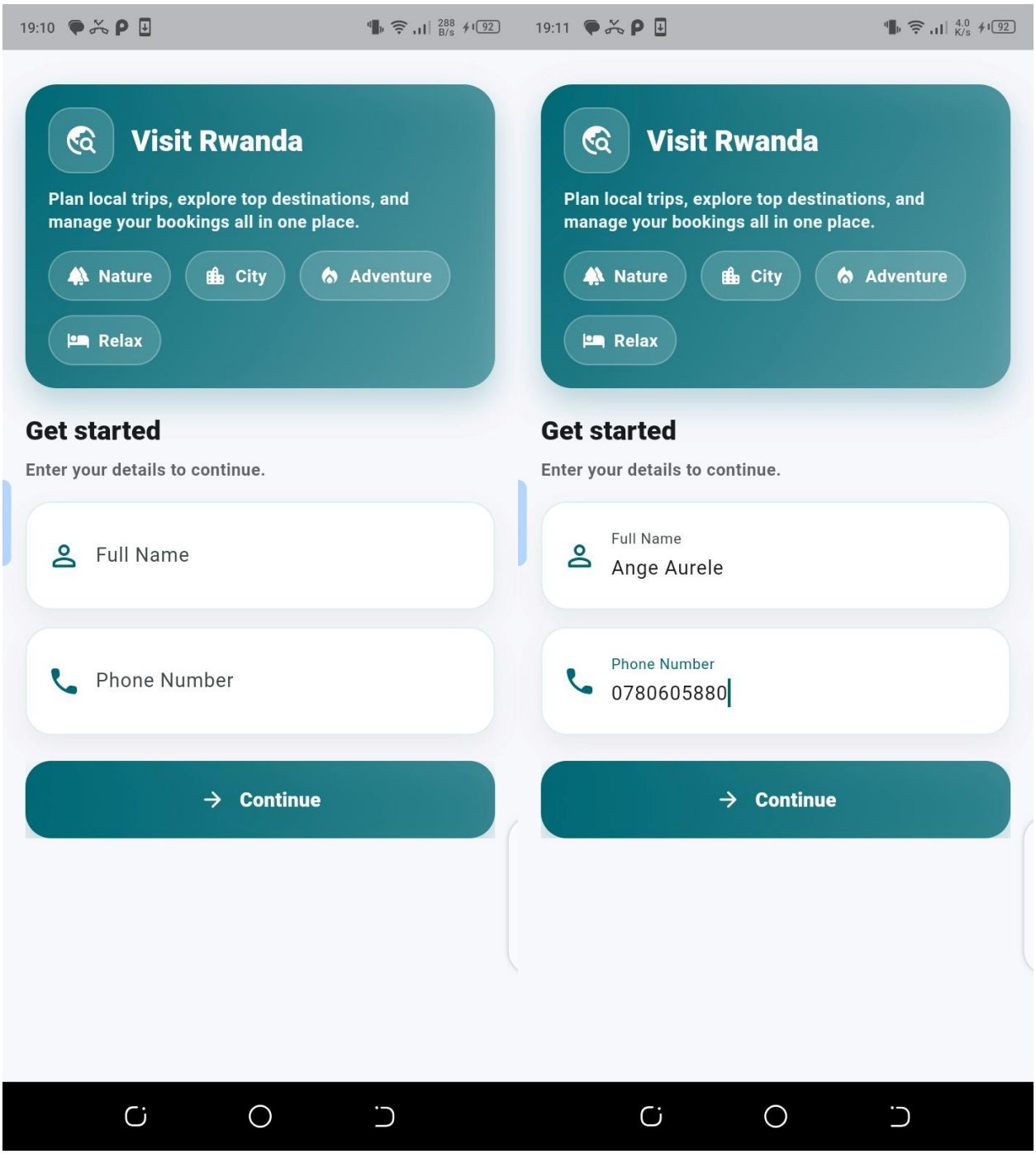
- Multi-screen navigation using Navigator.push and Navigator.pop
- Data passing between screens
- Stateful and stateless widget interaction
- UI state management using ValueNotifier
- Responsive layout handling using LayoutBuilder
- Custom reusable components to improve maintainability and modularity
- Structured separation of concerns through organized folders (screens, widgets, data, models)

The application simulates a local Rwanda travel platform where users can:

- Register basic information (UI-only session management)
- Browse destinations
- Filter destinations by category
- Search dynamically using keyword matching
- View detailed destination information
- Mark destinations as favorites
- Book trips using selectable options (transport type, number of travelers)
- View booked trips in a personalized profile section
- Logout and reset session data

2. Screens Overview

2.1 Welcome Screen



Description

The Welcome Screen appears when the application starts. It collects user information (Name and Phone Number) before accessing the main app.

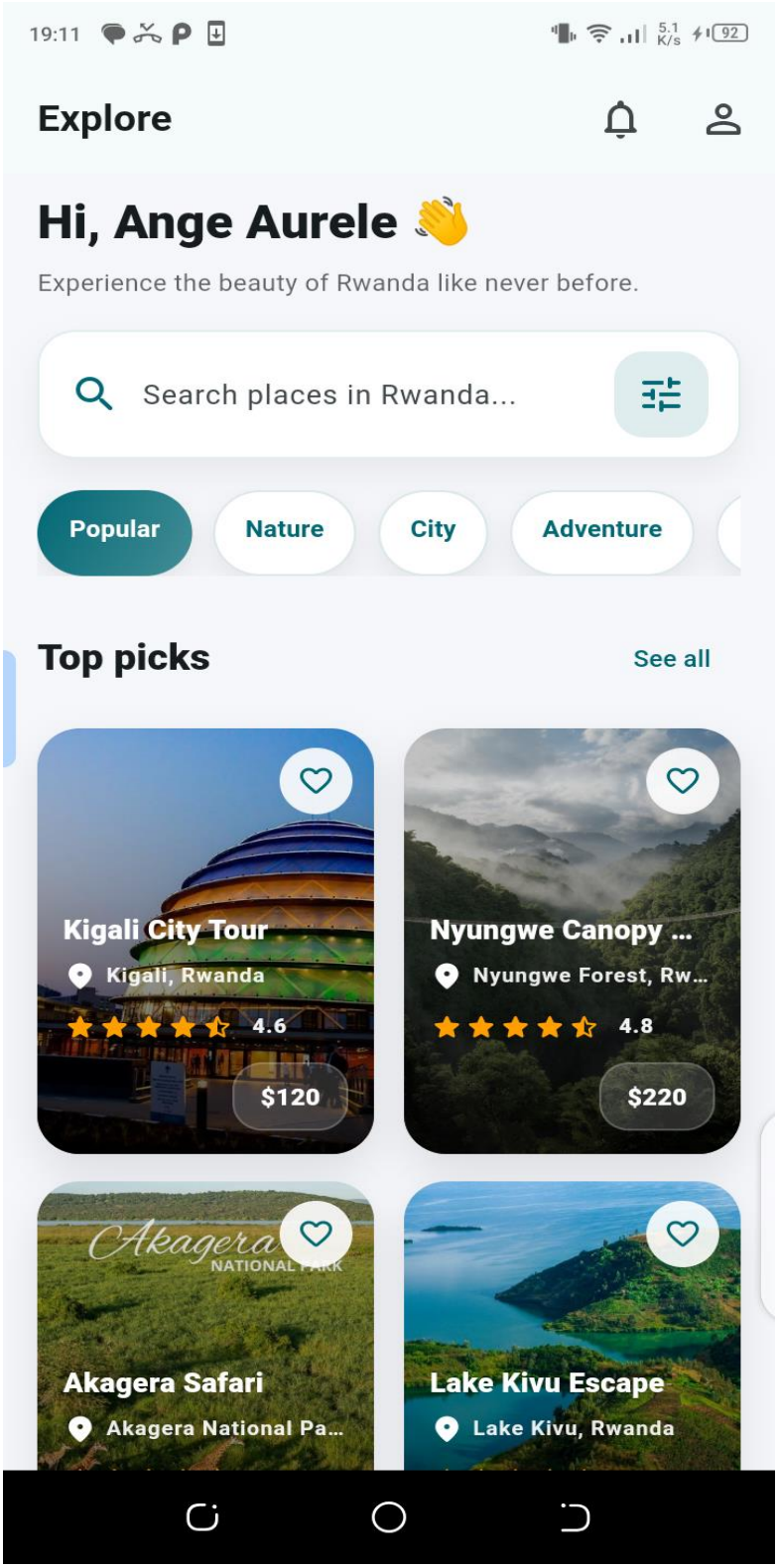
Layout Choices

- **Gradient hero card:** A gradient background is used to make the welcome section stand out while maintaining a modern and attractive design.
- **Custom reusable field containers:** Input fields are wrapped in styled containers to ensure consistent design, better alignment, and improved reusability across the application.
- **Consistent rounded borders and shadows:** Rounded corners and subtle shadows provide a modern UI appearance and create visual depth between sections.
- **Clean vertical spacing using SizedBox:** SizedBox is used to maintain consistent spacing between elements, improving readability and preventing clutter.
- **Uses ListView to prevent overflow on small devices:** Wrapping the layout in a ListView ensures the screen scrolls properly on smaller devices and avoids overflow errors.

Widgets Used

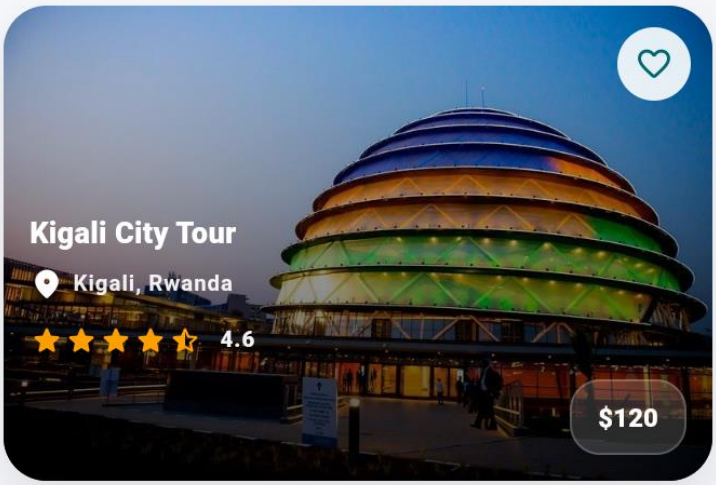
- Scaffold
- SafeArea
- ListView
- Container
- Column
- Row
- Text
- TextField
- Icon
- SizedBox
- BoxDecoration
- LinearGradient
- PrimaryButton (custom widget)

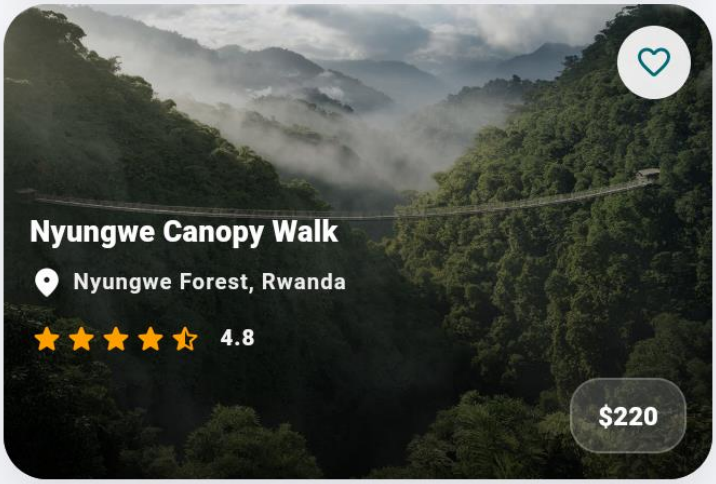
2.2 Home Screen (Dashboard)

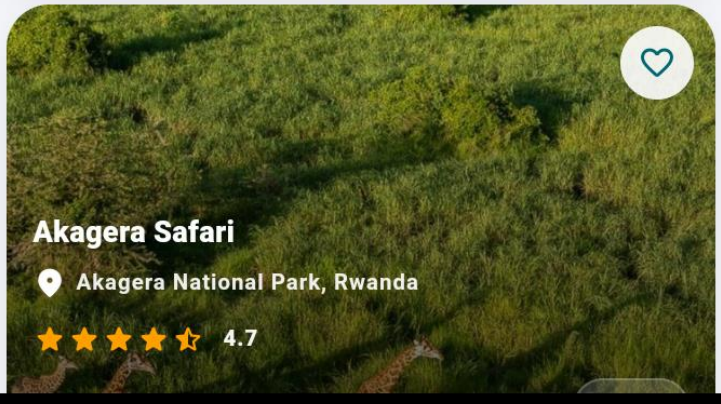


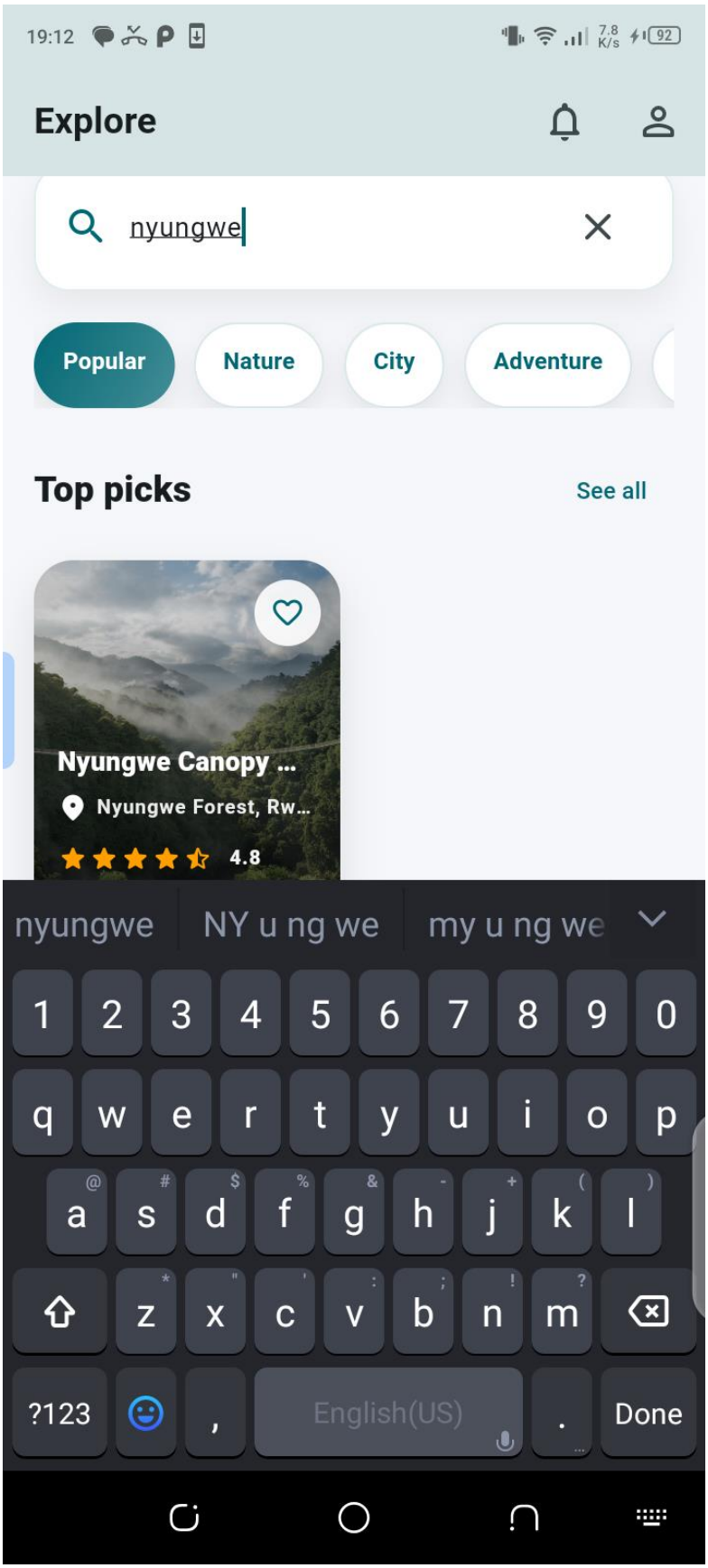
19:11

All Places








19:12
Explore
nyungwe
Popular
Nature
City
Adventure
Top picks
See all
Nyungwe Canopy ...
Nyungwe Forest, Rw...
4.8
nyungwe
NY u ng we
my u ng we
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
English(US)
Done

Description

The Home Screen serves as the main dashboard of the application, providing users with an overview of available travel destinations. It includes a personalized greeting message, a functional search bar for filtering destinations, a horizontally scrollable category selector, and a responsive grid of destination cards. The screen also features a “See All” button for expanded viewing, along with quick access icons for profile and notifications, ensuring smooth and intuitive navigation.

Layout Choices

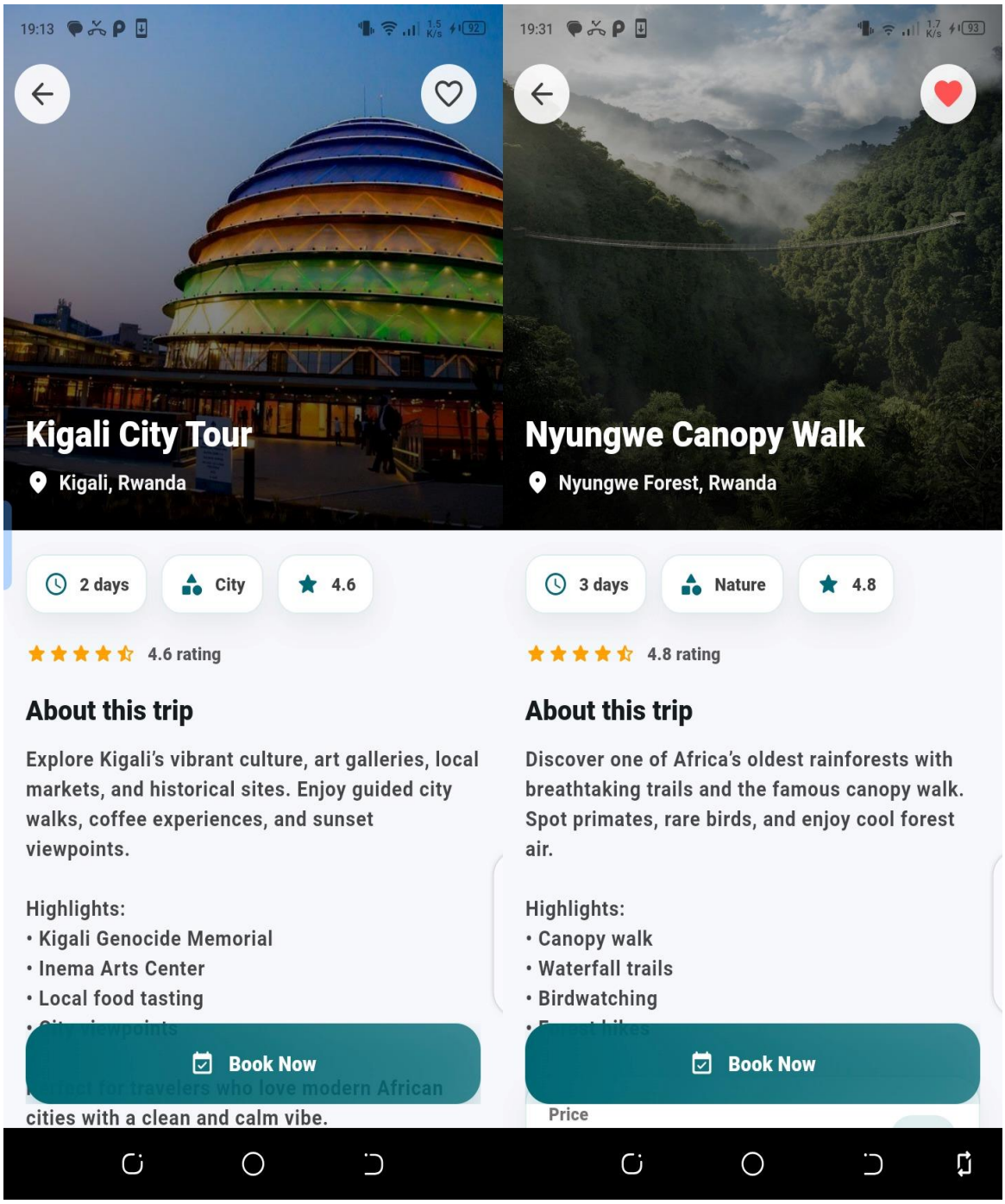
- **ListView for vertical scrolling:** The main layout is wrapped in a ListView to allow smooth vertical scrolling and ensure all content remains accessible on different screen sizes.
- **GridView.builder for dynamic destination cards:** GridView.builder is used to efficiently generate destination cards dynamically based on the available data.
- **LayoutBuilder for responsive grid columns:** LayoutBuilder adjusts the number of grid columns depending on screen width, ensuring responsive design across devices.
- **ValueListenableBuilder for user session updates:** ValueListenableBuilder updates the UI in real time when user session data changes, such as login or logout.
- **Custom reusable DestinationCard widget:** A reusable DestinationCard component promotes consistency, modularity, and cleaner code organization.
- **Dynamic search filtering logic:** The search feature filters destinations in real time based on user input, improving usability and interactivity.

Widgets Used

- AppBar
- IconButton
- ListView
- GridView.builder
- LayoutBuilder
- CategoryChip (custom)
- DestinationCard (custom)
- Text

- ValueListenableBuilder
- Row
- Expanded
- InkWell

2.3 Detail Screen



Description

The Detail Screen presents information about a selected destination in a visually engaging layout. It features a large header image with a gradient overlay for improved readability, a clickable favorite icon for user interaction, and a scrollable description section. The screen also includes a styled price card and a sticky “Book Now” button positioned at the bottom to encourage quick and convenient booking actions.

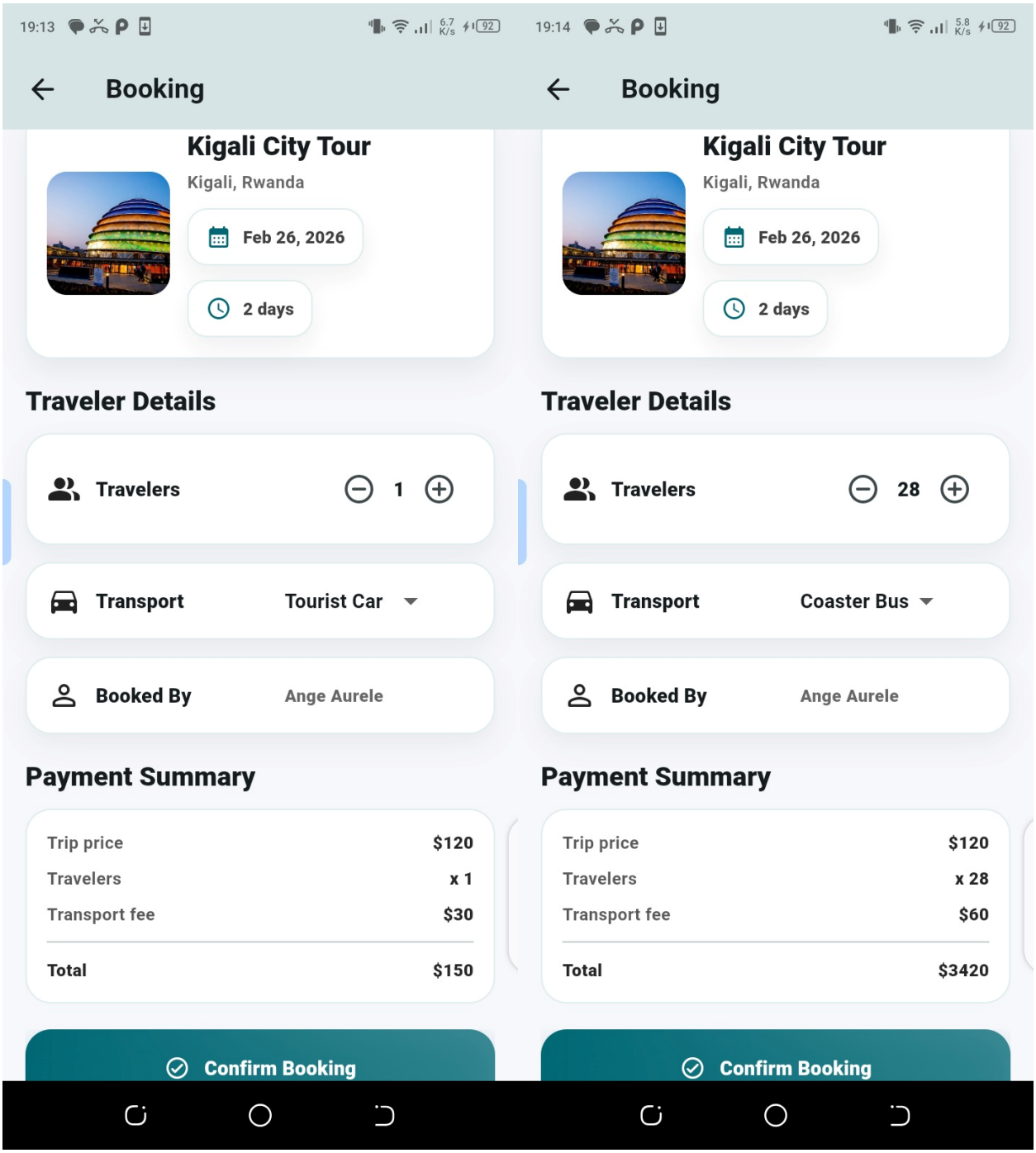
Layout Choices

- **CustomScrollView + SliverAppBar for collapsible header:** A SliverAppBar inside a CustomScrollView creates a smooth collapsing header effect, improving visual interaction and space management.
- **Gradient overlay for text readability:** A gradient is layered over the header image to enhance text contrast and ensure readability against bright backgrounds.
- **Positioned sticky booking button:** The “Book Now” button is positioned at the bottom of the screen to remain easily accessible and encourage user action.
- **Reusable InfoChip and RatingStars widgets:** Custom reusable widgets maintain consistent styling and simplify the UI structure across different screens.
- **Responsive text wrapping:** Text elements are designed to wrap and adapt to different screen sizes, preventing overflow and improving readability.

Widgets Used

- CustomScrollView
- SliverAppBar
- FlexibleSpaceBar
- Stack
- Positioned
- Column
- Wrap
- InfoChip (custom)
- RatingStars (custom)
- PrimaryButton (custom)

2.4 Booking Screen



19:14

Booking

Kigali City Tour

Kigali, Rwanda

Feb 26, 2026

2 days

Tra

Success

Your booking for Kigali City Tour has been confirmed.

Transport: Coaster Bus
Travelers: 28
Total: $3420

Done

Payment Summary

Trip price	$120
Travelers	x 28
Transport fee	$60
Total	**$3420**

Confirm Booking

Description

The Booking Screen allows users to customize their trip details before confirmation. Users can select the number of travelers, choose a preferred transport type (Tourist Car, Coaster Bus, or Helicopter), and view the dynamically calculated total price. Upon confirmation, the system displays a success dialog to simulate a complete booking process.

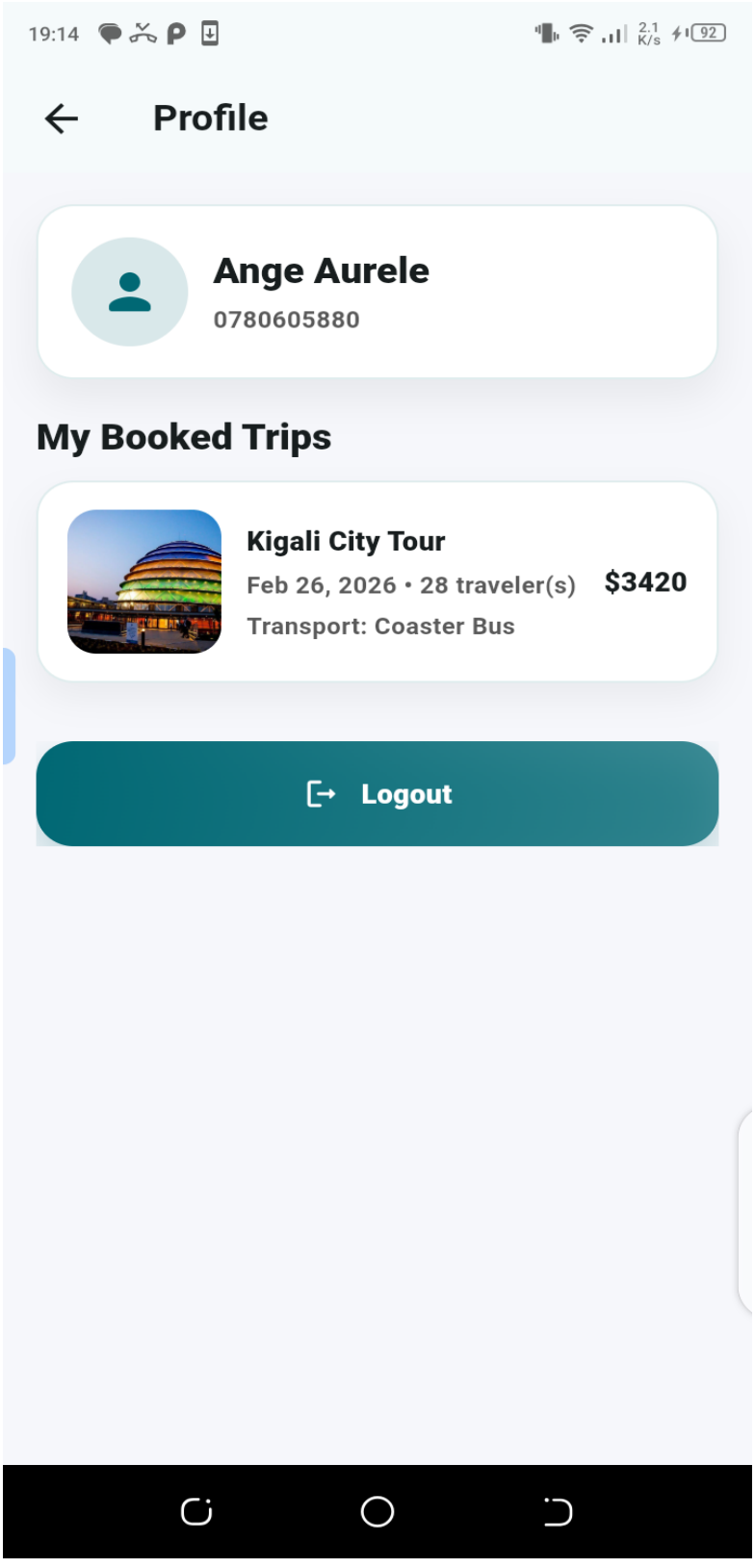
Layout Choices

- **SingleChildScrollView to avoid overflow:** The layout is wrapped in a SingleChildScrollView to ensure content remains scrollable and prevents overflow on smaller devices.
- **Dynamic total price calculation:** The total booking cost is calculated in real time based on selected travelers and transport options, improving interactivity.
- **Hard-coded date auto-generated using DateTime.now():** The booking date is generated dynamically using the current system date to simulate real booking behavior.
- **Reusable _FieldCard UI component:** A custom _FieldCard container ensures consistent styling and modular design for form sections.
- **Booking stored using in-memory state (BookingStore):** Bookings are managed using a ValueNotifier-based in-memory store to update the UI instantly without using a database.

Widgets Used

- SingleChildScrollView
- Container
- Row
- Column
- DropdownButton
- IconButton
- AlertDialog
- SnackBar
- Divider
- ValueListenableBuilder

2.5 Profile Screen



Description

The Profile Screen displays the logged-in user's information along with a list of all booked trips. It also includes a logout button for ending the session. When the user logs out, the application clears the user session, removes all stored bookings and favorites, and resets the navigation history to ensure a complete and secure logout process.

Layout Choices

- **ListView for scrollable content:** The Profile screen uses ListView to ensure all user information and bookings remain accessible through smooth vertical scrolling.
- **Card-style containers for bookings:** Each booking is displayed inside a styled container with rounded corners and shadows to visually separate items and maintain consistency.
- **ValueListenableBuilder for real-time updates:** ValueListenableBuilder allows the booking list and user data to update automatically whenever changes occur in the in-memory state.
- **pushAndRemoveUntil for full logout reset:** The pushAndRemoveUntil method clears the navigation stack during logout, ensuring the user cannot return to previous screens after signing out.

3. Hard-Coded Data Implementation

All destination data is stored in: lib/data/travel_data.dart

Includes:

- id
- name
- location
- category
- days
- price
- rating
- description

No API or database is used, as required.

4. State Management Approach

This project uses simple in-memory state management:

- UserSession
- BookingStore
- FavoritesStore

This allows UI updates without external packages.

5. List of Widgets Used (Requirement: At least 18)

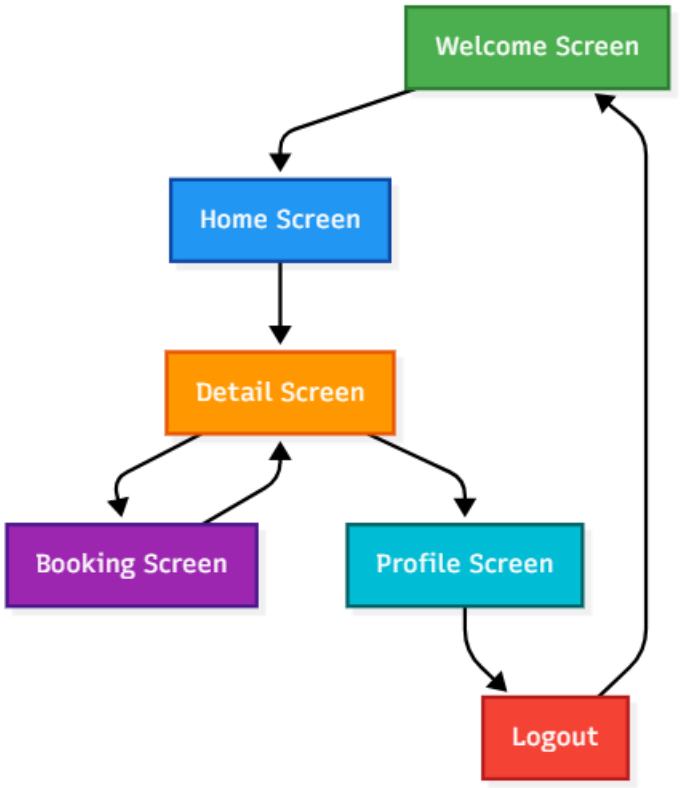
The following Flutter widgets were used:

1. Scaffold
2. AppBar
3. SafeArea
4. ListView
5. GridView.builder
6. CustomScrollView
7. SliverAppBar
8. FlexibleSpaceBar
9. Stack
10. Positioned
11. Container
12. Column
13. Row
14. Expanded
15. Wrap
16. Text
17. Icon

18. IconButton
19. InkWell
20. DropdownButton
21. ValueListenableBuilder
22. AlertDialog
23. SnackBar
24. ClipRRect
25. LayoutBuilder

(Custom widgets were also created.)

6. Navigation Flow Diagram



7. UI Design Considerations

- Consistent color theme
- Rounded corners throughout
- Soft shadows for elevation
- Gradient overlays for readability
- Responsive grid layout
- Clean spacing using SizedBox
- No overflow errors
- Local Rwanda-based theme
- Reusable components for maintainability

8. Conclusion

The Travel App UI successfully demonstrates the implementation of multi-screen navigation, complex layout structuring, responsive design techniques, reusable custom widgets, functional UI state management, and a clean, user-centered experience. The application shows effective use of Flutter's layout system through organized widget composition, dynamic content rendering, and smooth screen transitions. All lab requirements were fully satisfied, including the use of hard-coded data, implementation of more than the minimum required widgets, proper navigation flow between screens, structured UI organization, and the consistent application of gradients and shadow effects for enhanced visual presentation. Overall, the project highlights strong understanding of Flutter UI development principles and responsive mobile interface design.