Flutter Navigator API Scenarios Storyboards v2

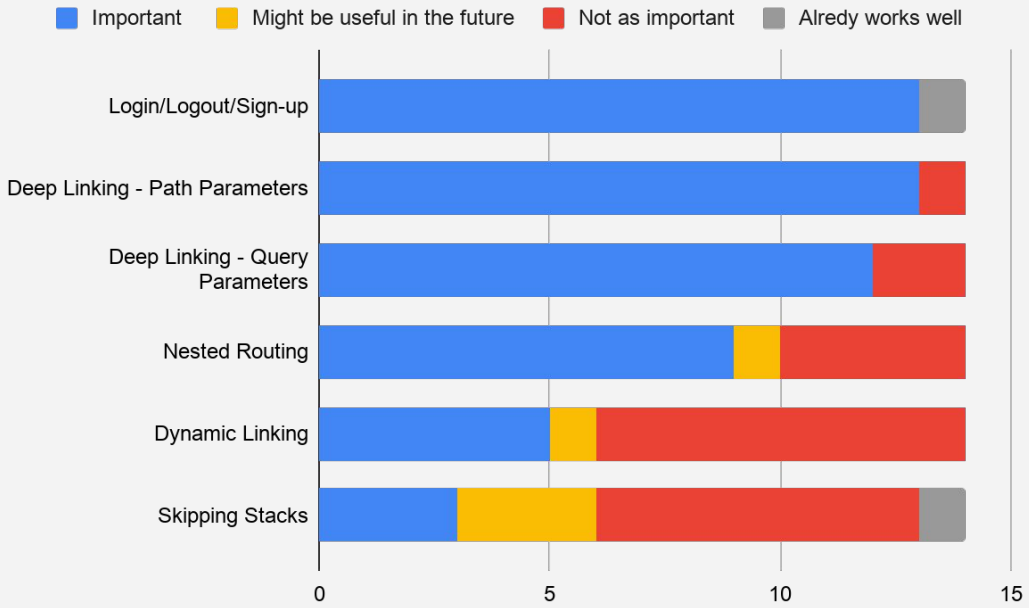
Flutter UXR Team
Last edited: 3/15/2021

TL;DR

- Through usability research, the Flutter team looks into a high-level API that will make Navigator 2.0 easier to use. (Learn more about the research here: project overview)
- In our past user interviews and surveys, we have collected six navigation and routing scenarios Flutter developers targeting the Web browser considered important and yet difficult to implement. We created six storyboards to illustrate these scenarios: Storyboards v1.
- After gathering feedback on these initial storyboards from both app developers and router package authors, we updated the storyboards based on this feedback. Below are links to the storyboards of these 6 scenarios:
    - 1. Deep Linking - Path Parameters
    - 2. Deep Linking - Query Parameters
    - 3. Login/Logout/Sign-up Routing - Routing with Validation
    - 4. Nested Routing
    - 5. Skipping Stacks
    - 6. Dynamic Linking

User Feedback on Storyboards v1

We interviewed users (n=14) who reported finding deep linking, login/logout/sign-up routing, and nested routing relatively more important than other scenarios. They, however, saw some value in skipping stacks and dynamic linking as they imagine their app to scale in the future.



What's New in Storyboard v2

- We modified the scenarios within the context of one app: **a book app**. We chose a book app because 1) we came across many router packages that are building examples on top of John Ryan's book app, and 2) it helped us imagine a more advanced version of the app that would require all six scenarios we have identified.

These are the changes that are made to the storyboards:

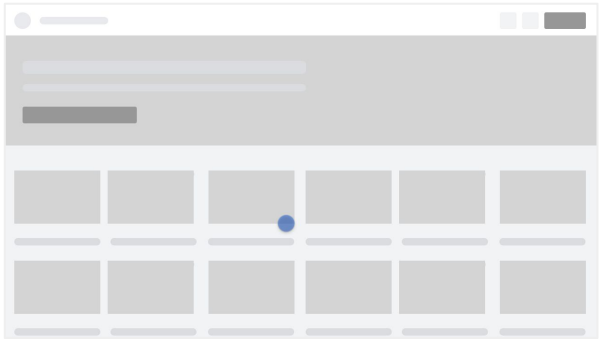
- **Deep Linking - Path Parameters** - no changes from v1
- **Deep Linking - Query Parameters** - no changes from v1
- **Login/Logout/Sign-up Routing - Routing with Validation** - has a new scenario where login is the initial and separate stack of the app
- **Nested Routing** - no major changes from v1 except that the example is a book app
- **Skipping Stacks** - has 1) improved visualization of the stack change and 2) two new "Manipulation of History Stacks" scenarios.
- **Dynamic Linking** - no major changes from v1 except that the example is a book app

1. Deep Linking - Path Parameters

Deep Linking - Path Parameters

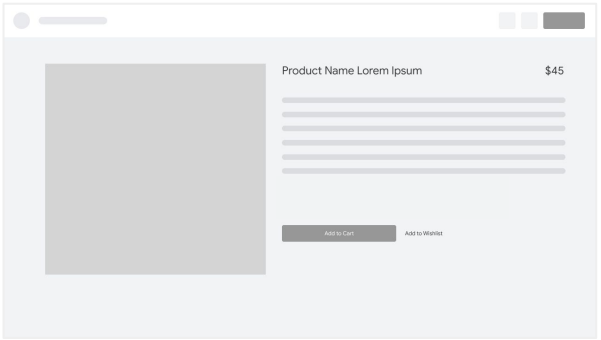
Example Scenario

Home



Kai taps on a book he likes

Book Details



Kai copies the URL

E.g., url.com/book/:id

Kai emails himself the link to the book for future reference

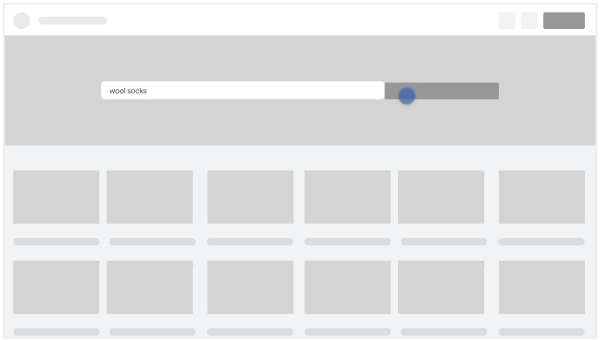
Kai clicks on the URL in his email

2. Deep Linking - Query Parameters

Deep Linking - Query Parameters

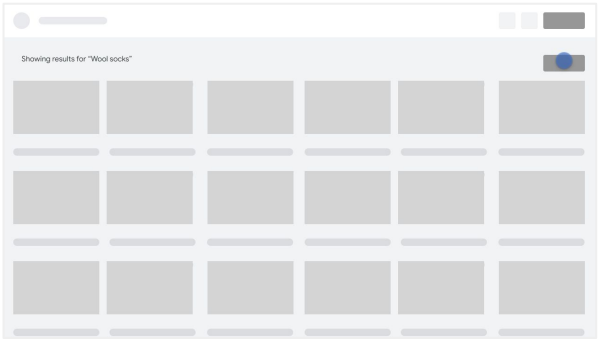
Example Scenario: E-Commerce

Home



Sharon searches for books with a keyword ‘fantasy’ to look for books on fantasy genre

Search Results



Sharon sorts the results by newest and copies the URL

E.g., url.com/book/q?=fantasy&sort=?newest

Sharon emails herself the link to the results for future reference

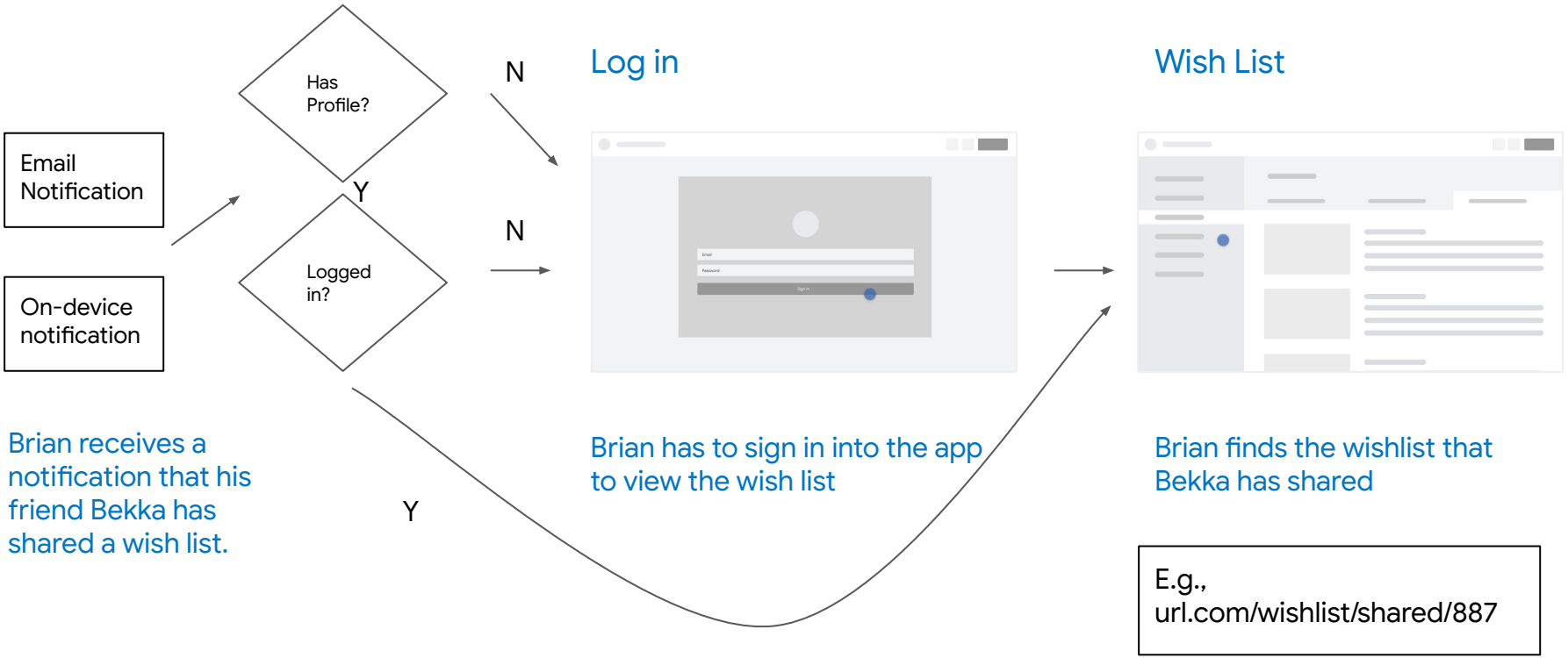
Sharon clicks on the URL in her email

3. Login/Logout/Sign-up Routing

Routing with validation

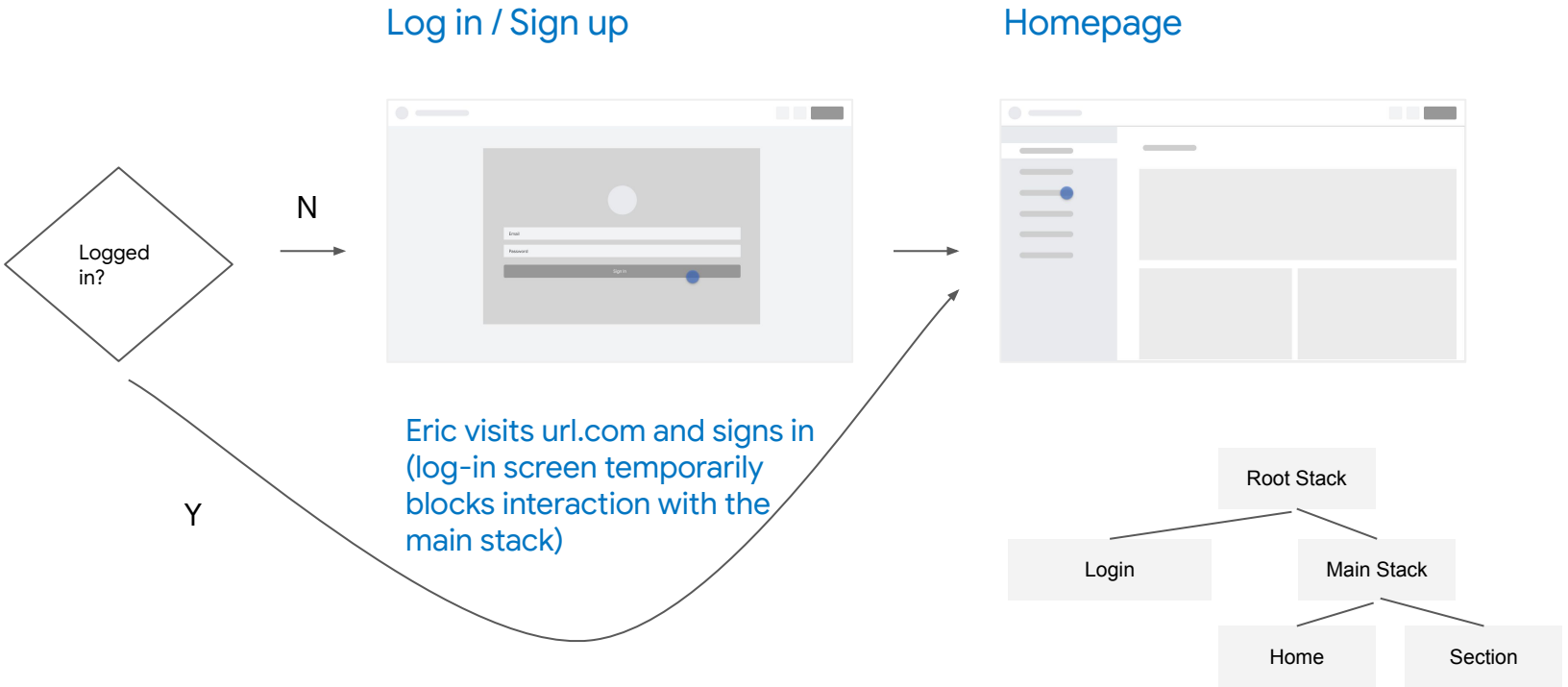
Login/Logout/Sign-up Routing - Deep link

Example Scenario



Login/Logout/Sign-up Routing - Home requires logging in

Example Scenario

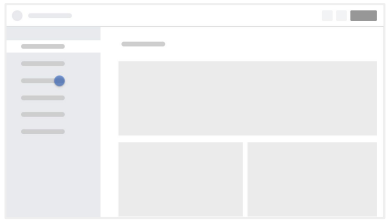


4. Nested Routing

Nested Routing (with Tabs)

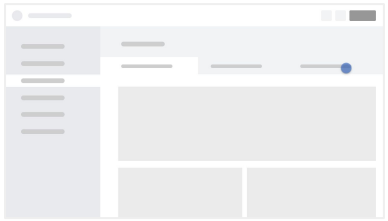
Example Scenario

Home



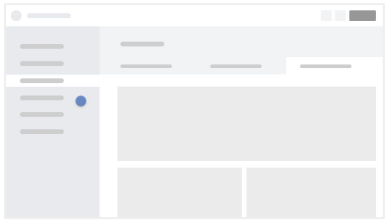
Declan clicks on the “Audiobook” section on the side navigation

Audiobook > All



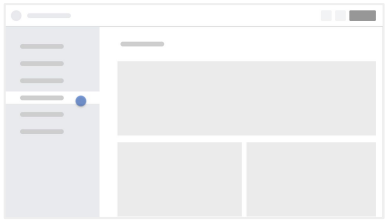
Declan clicks on “Staff Picks” tab

Audiobook > Staff Picks



Declan clicks on the “Fiction” section on the left navigation bar

Fiction



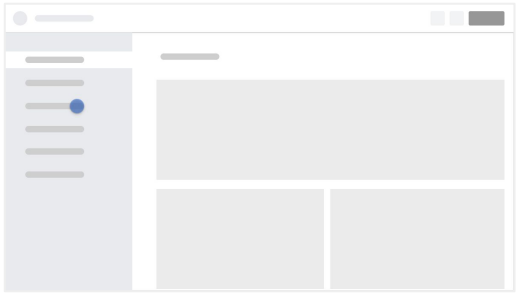
Presses browser back button and Declan is route to the first tab which he selected prior to the third tab

Presses browser back button to return to the “Staff Picks” tab on “Audiobook” Tab state is preserved.

Nested Routing with Modal Dialog

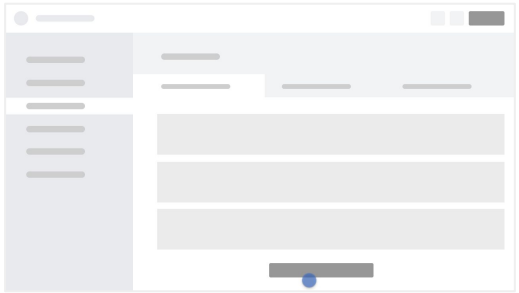
Example Scenario

Home



Declan clicks on the “Wishlist” section

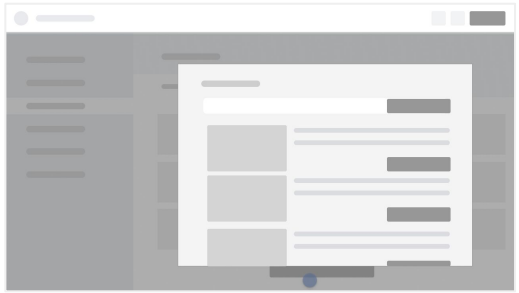
Wish Lists > My Wish Lists



Declan clicks on the “Add a new wishlist” button

E.g., url.com/wishlist/user123/

Wish Lists > My Wish Lists > Create New

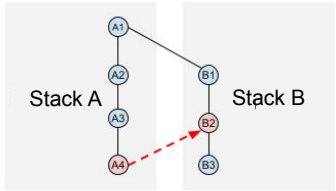


Declan taps on the browser back button

E.g., url.com/wishlist/user123/createnew

5. Skipping Stacks

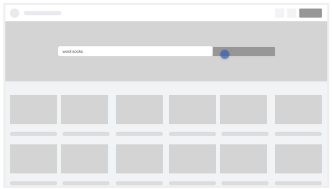
Skipping Stacks



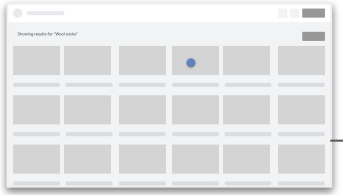
Example Scenario

‘Home’ Stack

Homepage



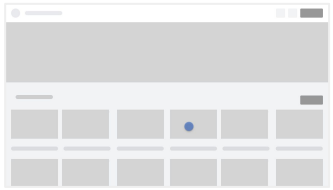
Theo searches for books using a keyword: “historical fiction”



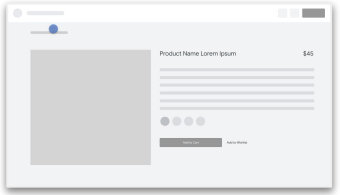
Theo discovers a book that he likes and clicks on it

Historical Fiction’ Stack

Historical Fiction Category Page



Book Detail



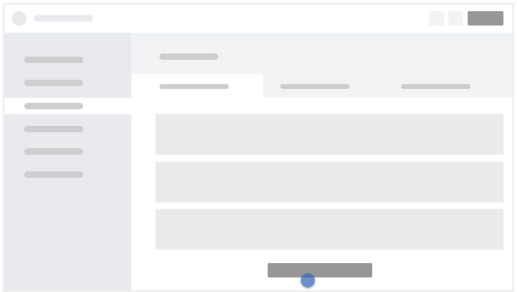
Theo clicks on the in-app back button

6. Dynamic Linking

Dynamic Linking

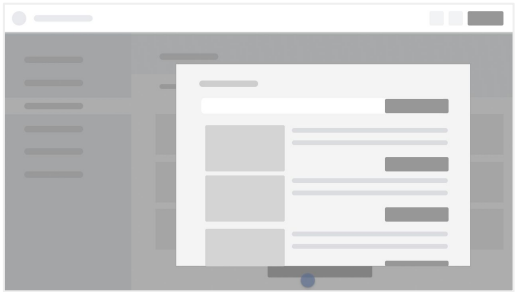
Example Scenario

Wish Lists > My Wish Lists



Declan clicks on the “Add a new wishlist” button

Wish Lists > My Wish Lists > Create New



Declan adds books to the wishlist and saves it

Wish list URL is created
E.g., url.com/wishlist/user123/223

Appendix: editing the navigator’s history stack

This section illustrates a scenario related to the Skipping Stacks scenario that the community found relevant.

Manipulation of the History Stack - Remove Duplicate Pages

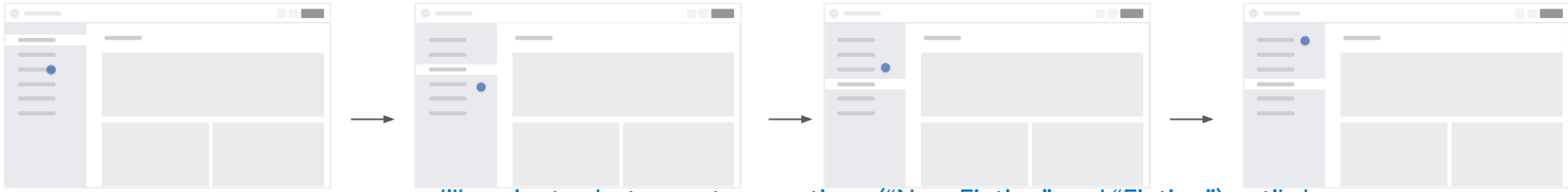
Example Scenario

Home Page

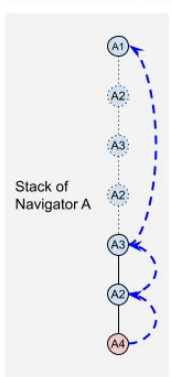
Non-Fiction

Fiction

Non-Fiction



Jill navigates between two sections (“Non-Fiction” and “Fiction”) until she decides to use back button to navigate to home



Jill taps on the in app back button (This time “Non-Fiction” section is skipped.)

Jill taps on the in app back button

What's Next

- Our immediate next step is to conduct a more detailed evaluation (#7) of usability benefits and disadvantages based on code snippets (#9) provided by package authors based on the soon-to-be-finalized navigator scenarios in #4. We have several authors who have expressed interest so far to participate in the next round and contribute code snippets for the usability evaluation. We will share a more detailed plan here: #9