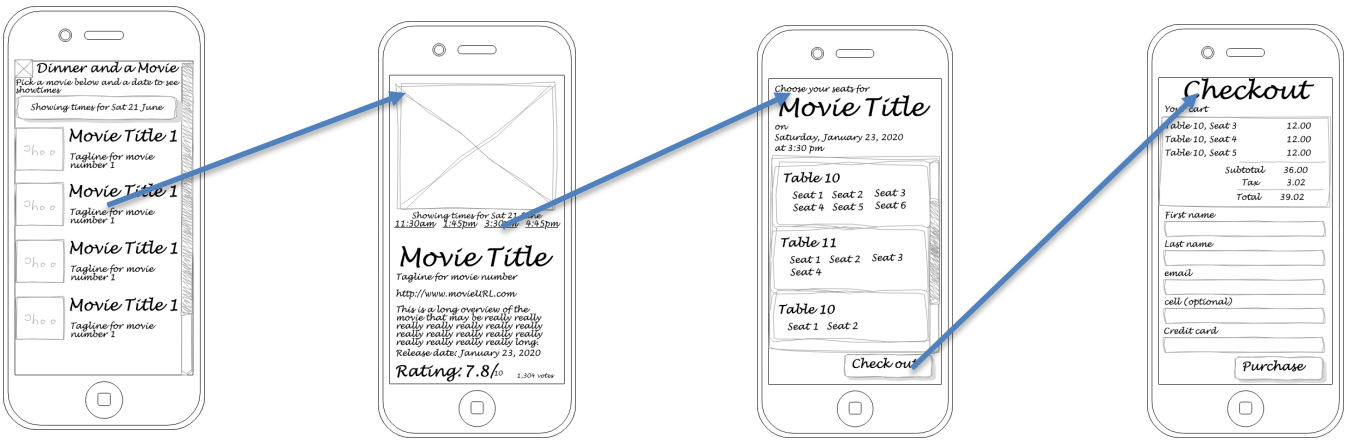
Routing and stack navigation

Someday we may change how users get around our app but for now we're not seeing the need for tab navigation or drawer navigation. But it is crying out for stack navigation. You know how up until now you've been changing main.dart, swapping out widgets in the MaterialApp widget? It felt pretty hokey, didn't it? Surely there's a better way to have the user navigate from one scene to the other! Well, there is and after this lab, your user will be able to start with the Landing widget, then ...

- Tapping a movie will navigate them to the FilmDetails scene.
- Tapping a showing time will navigate them to the PickSeats scene.
- Tapping the checkout button will navigate them to the Checkout scene.
- Tapping on the purchase button will navigate them to the Ticket scene.



Let's do this!

Creating the scene widgets

As of now, we have a Landing and Checkout widgets. But we don't have widgets for the FilmDetails scene, the PickSeats scene, or the Ticket scene. Let's create them real quick.

1. In lib create three widgets, film_details.dart as a StatelessWidget, and pick_seats.dart, and ticket.dart as Stateful widgets.
2. Each's build() method should just have a Text() with the name of the widget for now.

Setting up stack navigation

3. For each scene widget we wish to show, we need to create a route. Here's a table of routes. Assign a route name of your choosing to each of your scene widgets. We've filled one out for you.

Scene widget	Route name
Landing	
FilmDetails	/film
PickSeats	
Checkout	

Ticket	

4. Edit main.dart. Find the MaterialApp widget. Remove the home property.

5. Add a routes property. What is the data type of the value for the routes property?
   _____

6. Add route entries for all the routes you defined above. We've done the first one for you and here's how it would look:

```
routes: {
  "/film": (ctx) => FilmDetails(),
},
```

7. If you chose "/" for your Landing route name, you are ready to go. But if not, you'll need to tell the MaterialApp which is your startup route. You'll do so with the initialRoute property. Either add one of those and set it to your Landing route or change your Landing route to "/". Or both. Both is fine, too.

8. Run and test. You should see your Landing page.

Cool, you can see a scene. Let's navigate from one scene to another.

Navigating from FilmBrief to FilmDetails

9. Edit film_brief.dart. Remember how in the Gestures Lab, we had a GestureDetector respond to a tap?

10. In its onTap event handler do something like this:

```
onTap: () {
  Navigator.pushNamed(context, "/film");
},
```

11. Run and test. When you tap the FilmBrief, you should be routed to your FilmDetails scene.

Navigating from ShowingTimes to PickSeats

Let's do another one. Edit ShowingTimes. Remember that it fetches a list of showings. In the build method we are displaying those showings so the user can choose one. And the _makeShowingWidget() method creates each showing:

```
Widget _makeShowingWidget(dynamic dateString) {
  DateTime showingTime = DateTime.parse(dateString);
  String timeString = DateFormat.jm().format(showingTime.toLocal());
  return TextButton(
      //TODO: Navigate to PickSeats
      onPressed: () => print("Make this the current showing"),
      child: Text(timeString));
}
```

12. Do you see that "TODO"?. Go ahead and add the navigation to send the user to the PickSeats scene.

13. Run and test. Make sure your new navigation takes you from FilmBrief to PickSeats.

Simple enough? You take the next ones without as much direction.

14. In PickSeats, add a floatingActionButton (FAB). Make its onTap event navigate the user from PickSeats to Checkout.

15. Now write the navigation from Checkout to Ticket in Checkout's FAB.

16. Run and test.

Navigating back

You can navigate forward, but how about back? Remember that we can do this in a couple of ways. The one that makes sense for our purposes is to use the build-in back button navigation that is in the AppBar. Let's add one and try it out.

17. Edit checkout.dart. If it doesn't already have a Scaffold with an Appbar with a title, go ahead and add all of those.

18. Run your app again. Tap the checkout FAB on PickSeats to get to Checkout. Glance at your AppBar. See the "<" or " ←" button? Go ahead and tap it. Did you navigate back to PickSeats?

Congratulations! You're finished.