GitHub Username: askazoobee (https://github.com/askazoobee)

Ripely

Description

Write a brief summary of what your app does. What problem does your app solve?

It's hard to eat healthy and at the same time delicious in a global connected world. Each season brings a different bounty of fresh produce.

Finding out what's the freshest produce on stand for each individual U.S. state is even more important to get the best ripe-and-ready-to-eat fruits and vegetables.

Enjoying knowing that you are supporting local farmers while saving the big bucks are just great side effects of eating well.

Intended User

Who is your intended user?

This app is for anyone and everyone that wants a healthier and more sustainable lifestyle. This app serves as a "smart reference book" for produce shopping.

Features

List the main features of your app:

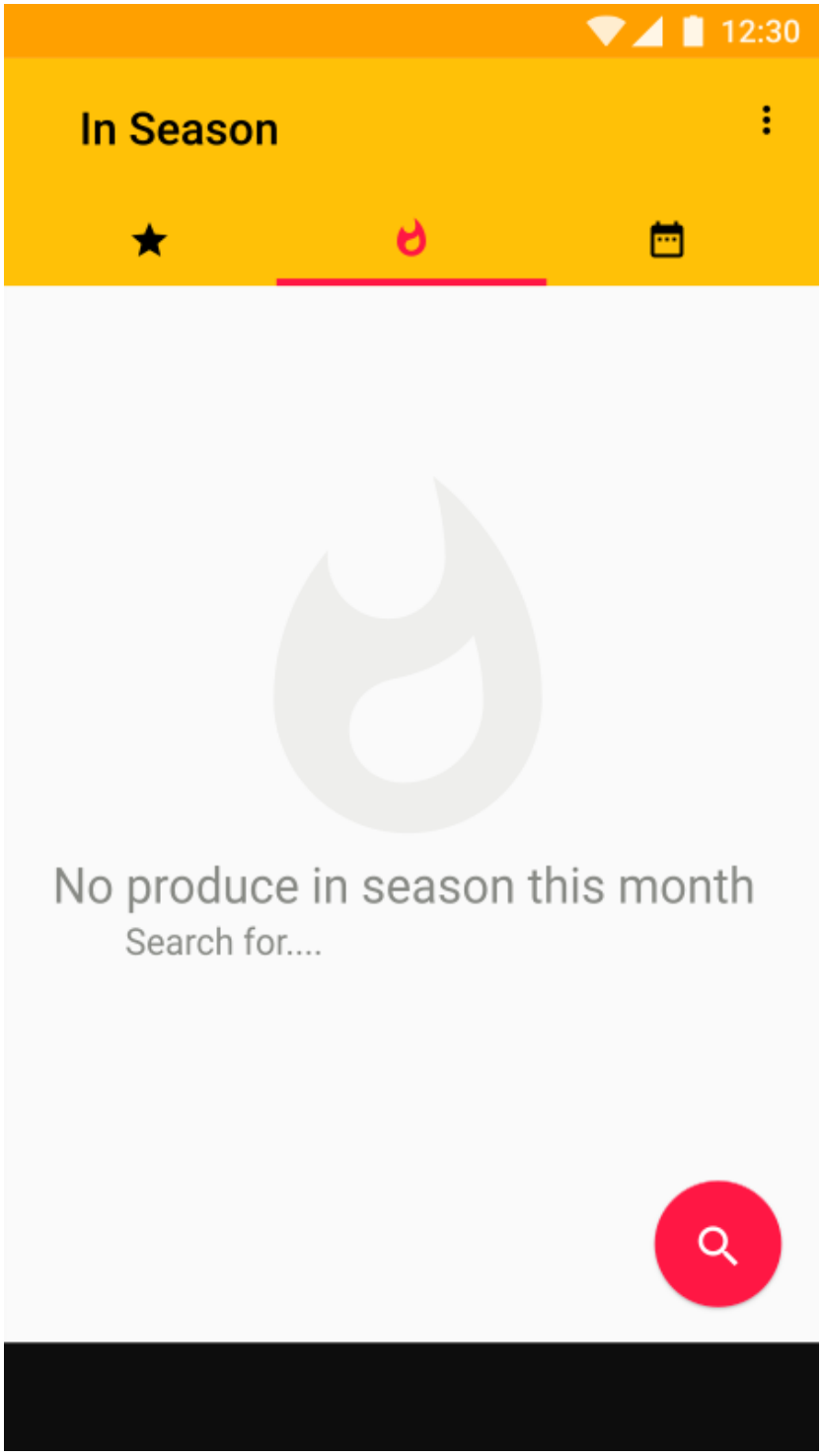
- **Favorite fruits and vegetables that you specially love and want to be reminded to buy when in season.**
- **Location based on U.S state to show relevant fresh produce.**
- **Search for what fruits are in what season based on location and months of the year.**
- **Get relevant links to read and learn more about the produce in your favorite browser.**
- **Get relevant notifications of favorited produce when they are in peak season.**

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

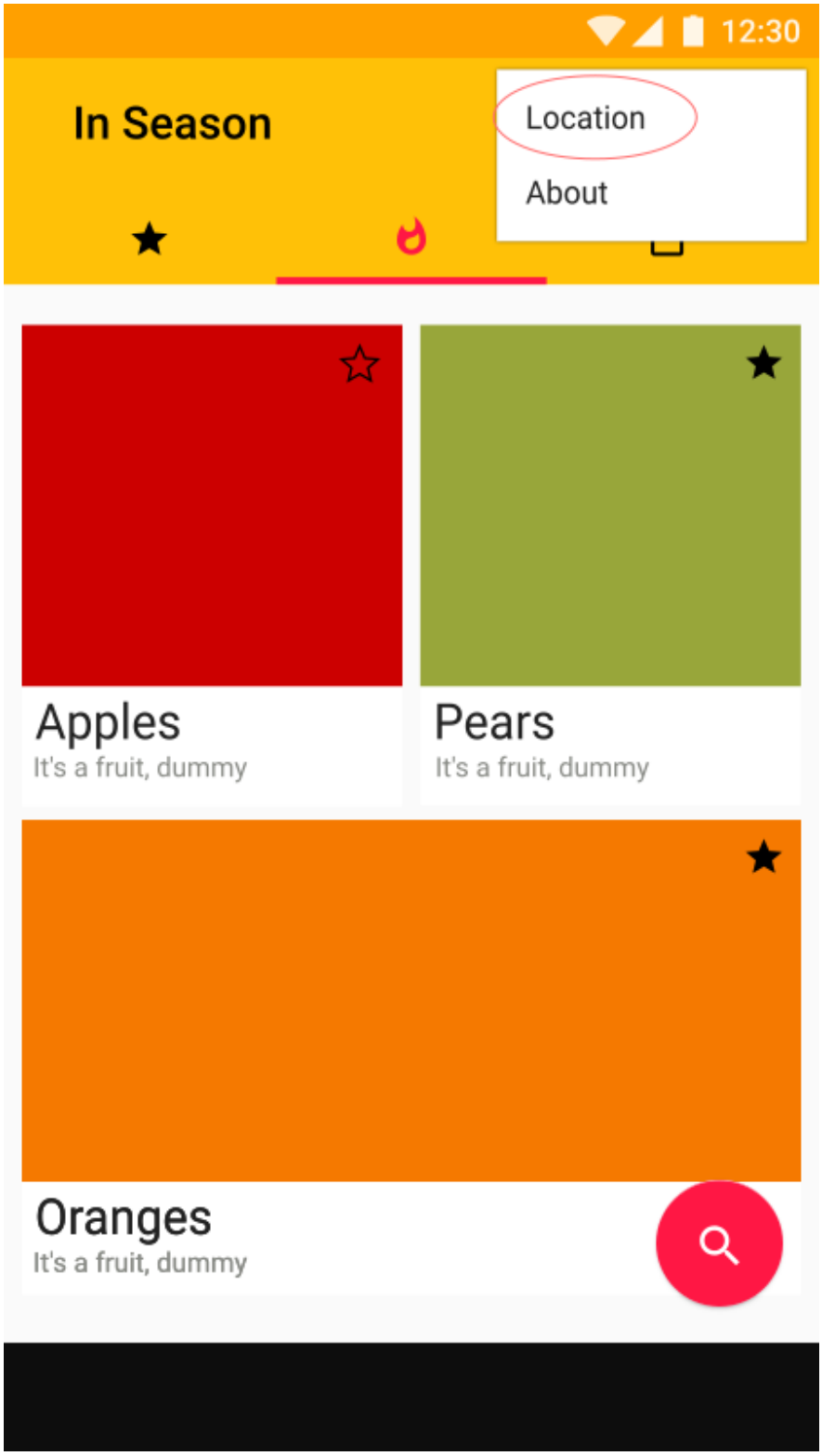
RIPELY ICON

Screen 1



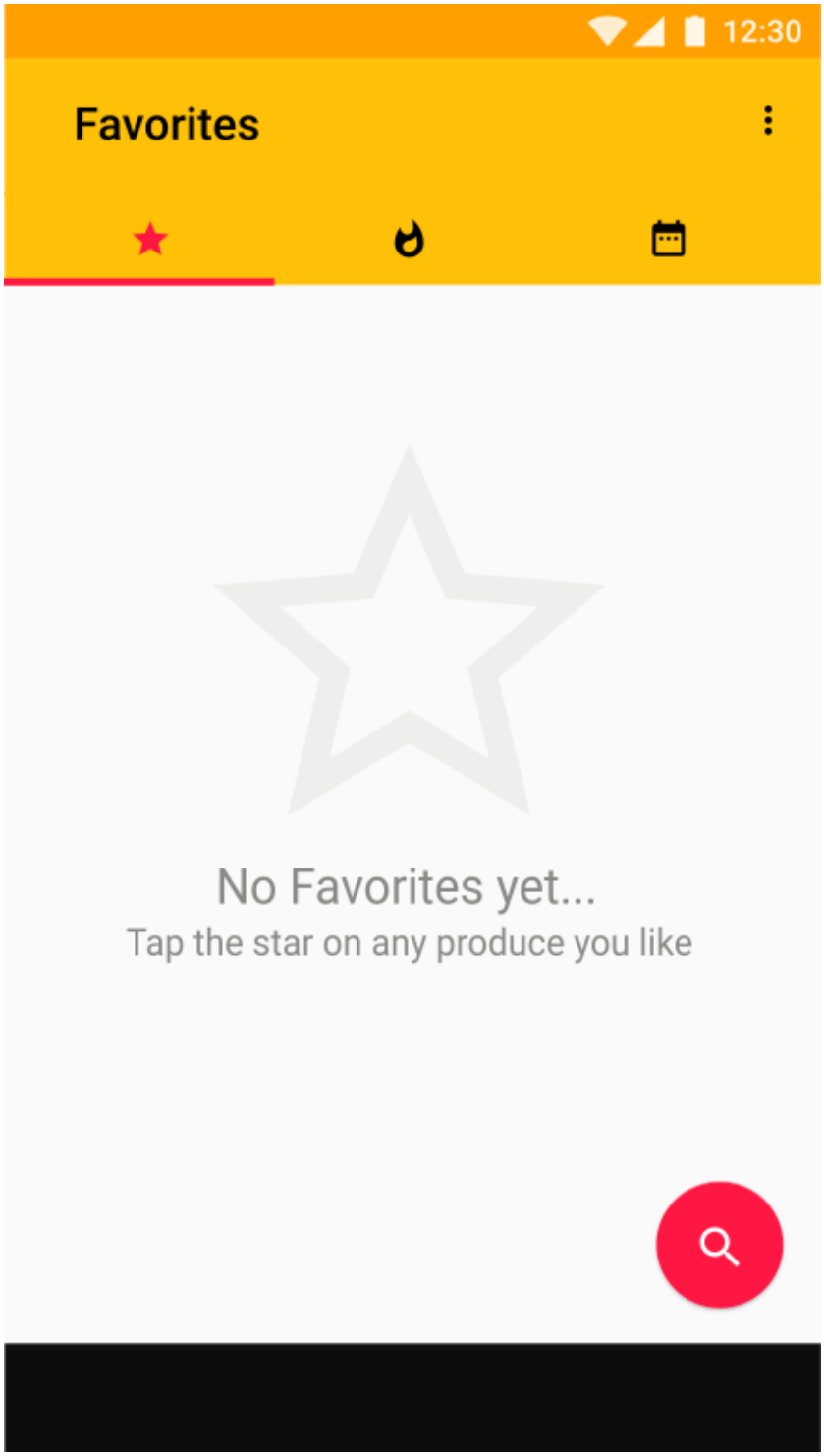
Main Screen in Empty State. Not all months have produce in season in every state. Before this screen there will be a startup dialog to ask user for location in first app open.

Screen 2



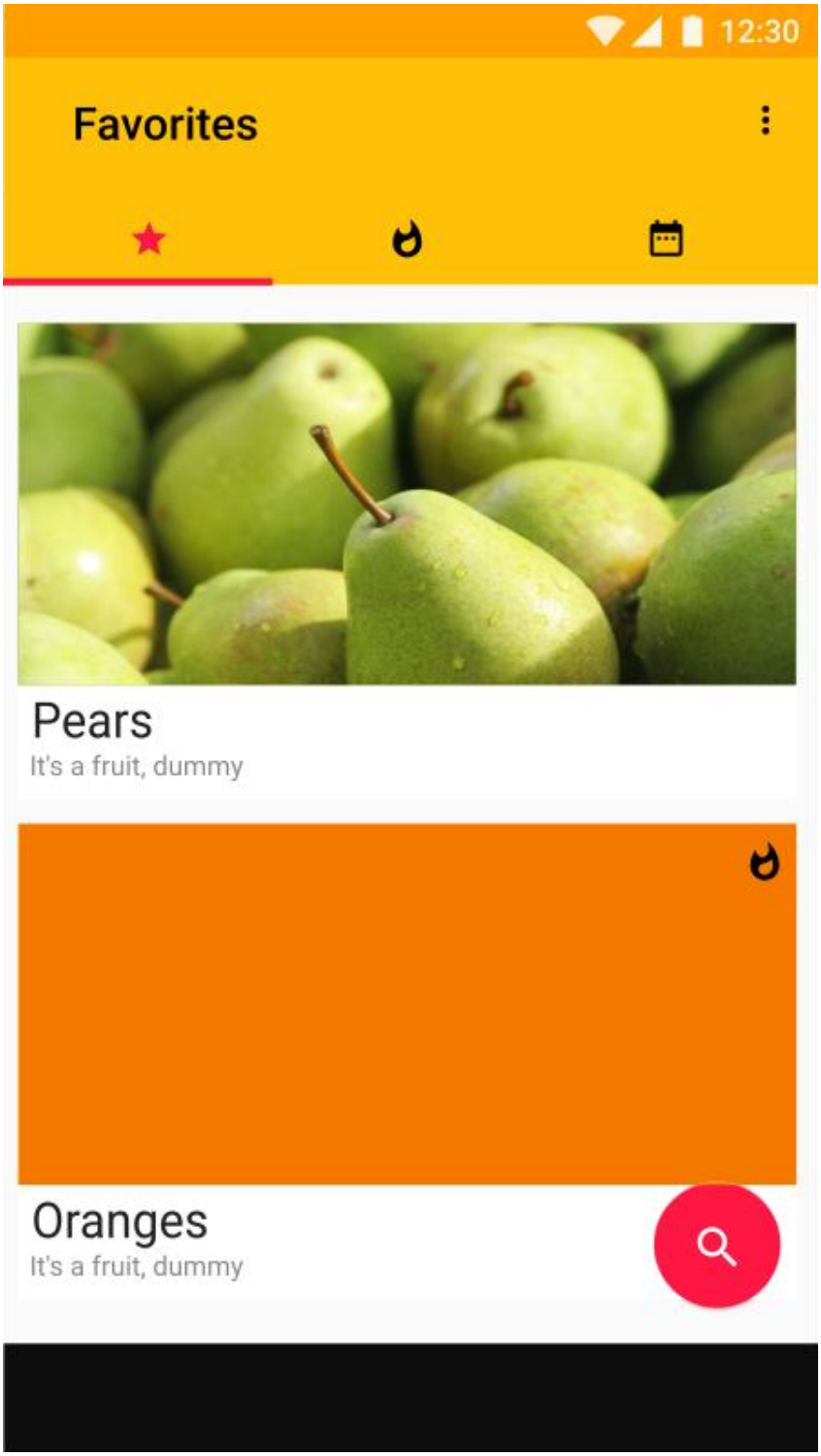
Same tab but with items populating empty space. This screen also shows three button menu.

Screen 3



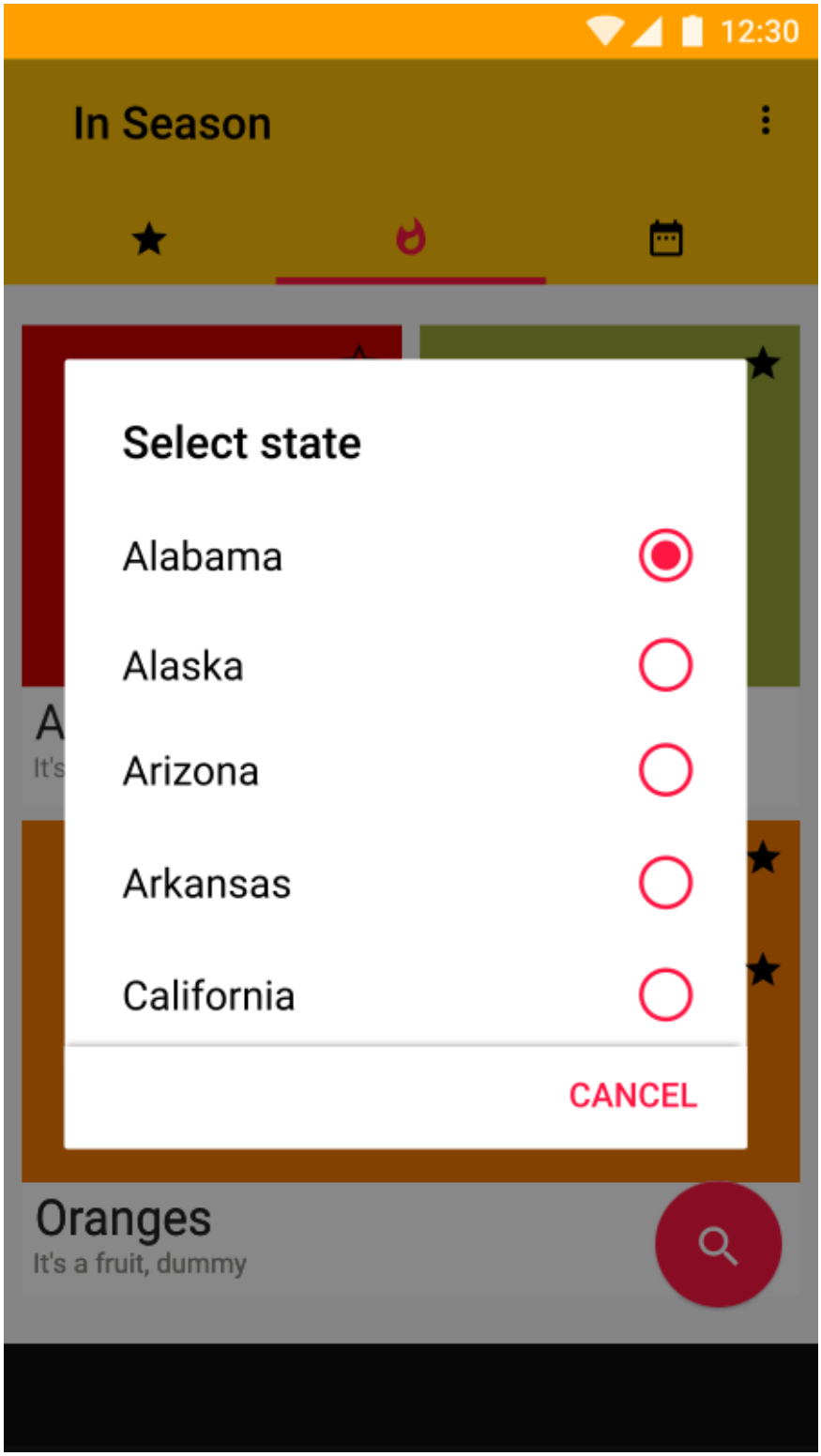
Empty state for favorites screen. User can save their preferred produce here and get notified of when is best to buy.

Screen 4



Populated favorites with indication if favorited item is in season. Cardview will have different layouts depending on # of items (1, 2, 3, 4 or more items at odd #'s, 4 or more items at even #'s).

Screen 5

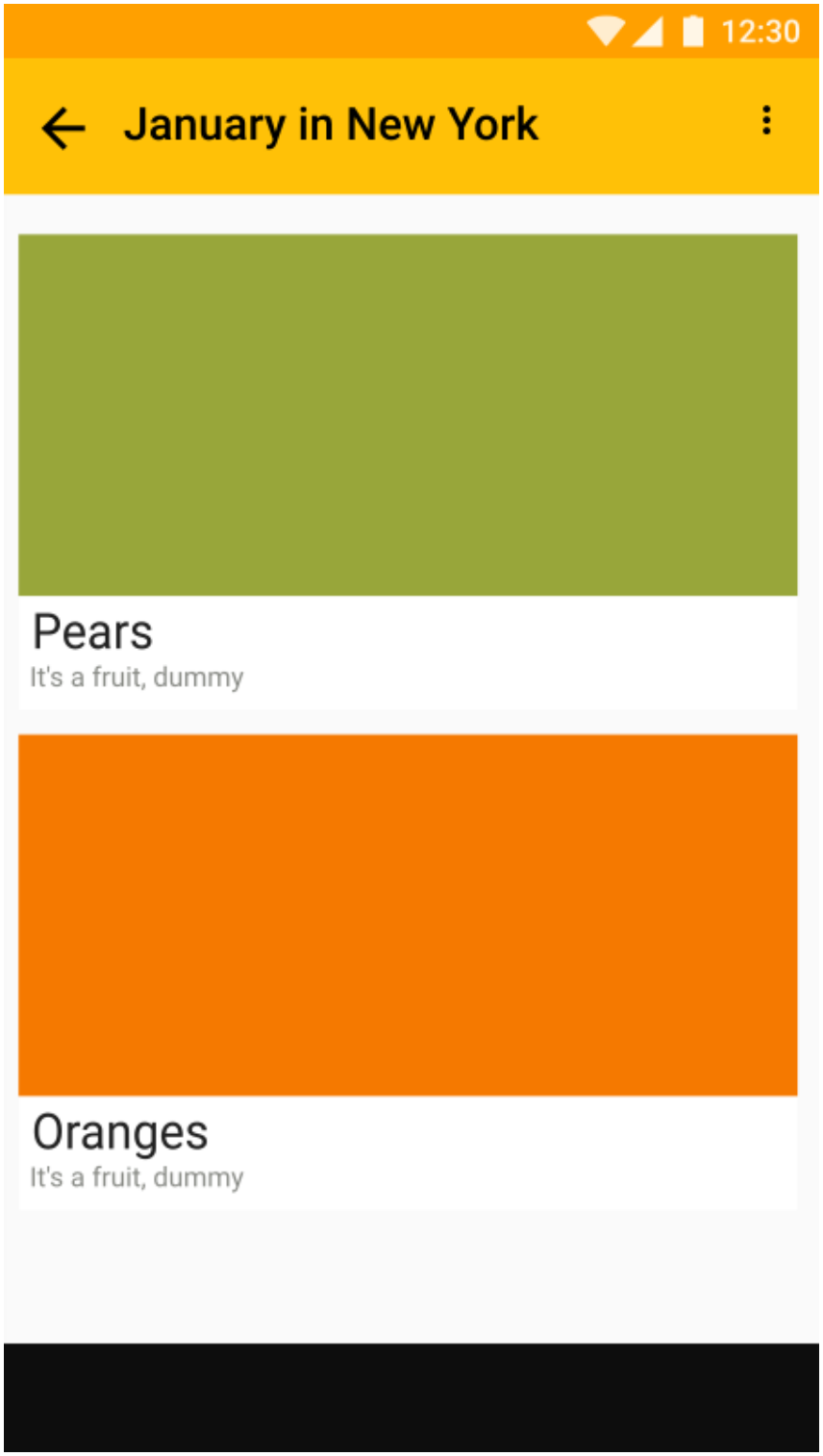


Setting dialog for location set up and change. Start-up dialog will be similar to this.

Screen 6

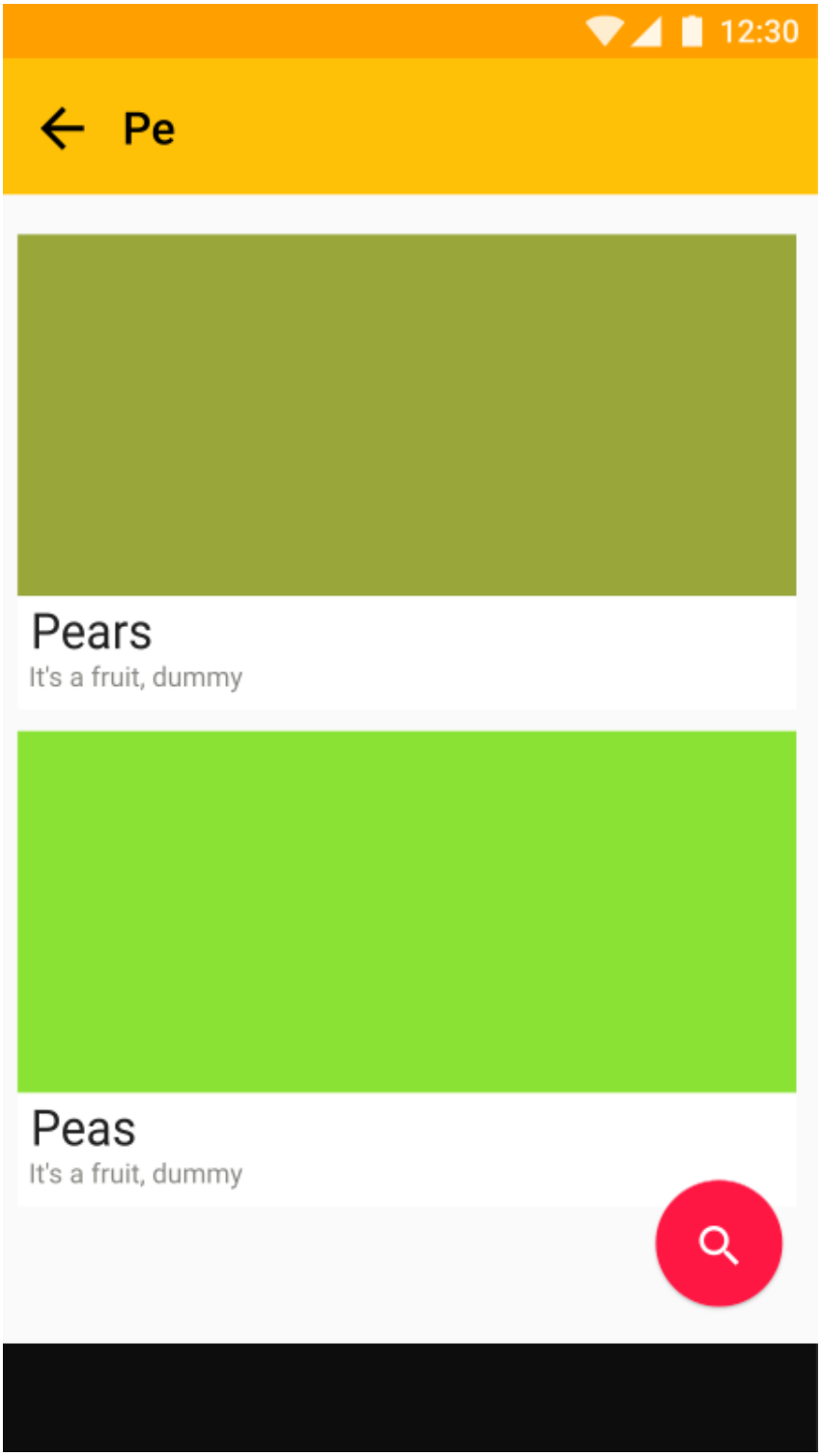
GridView for each calendar month to manually search which fruits are “hot” for the specified state.

Screen 7



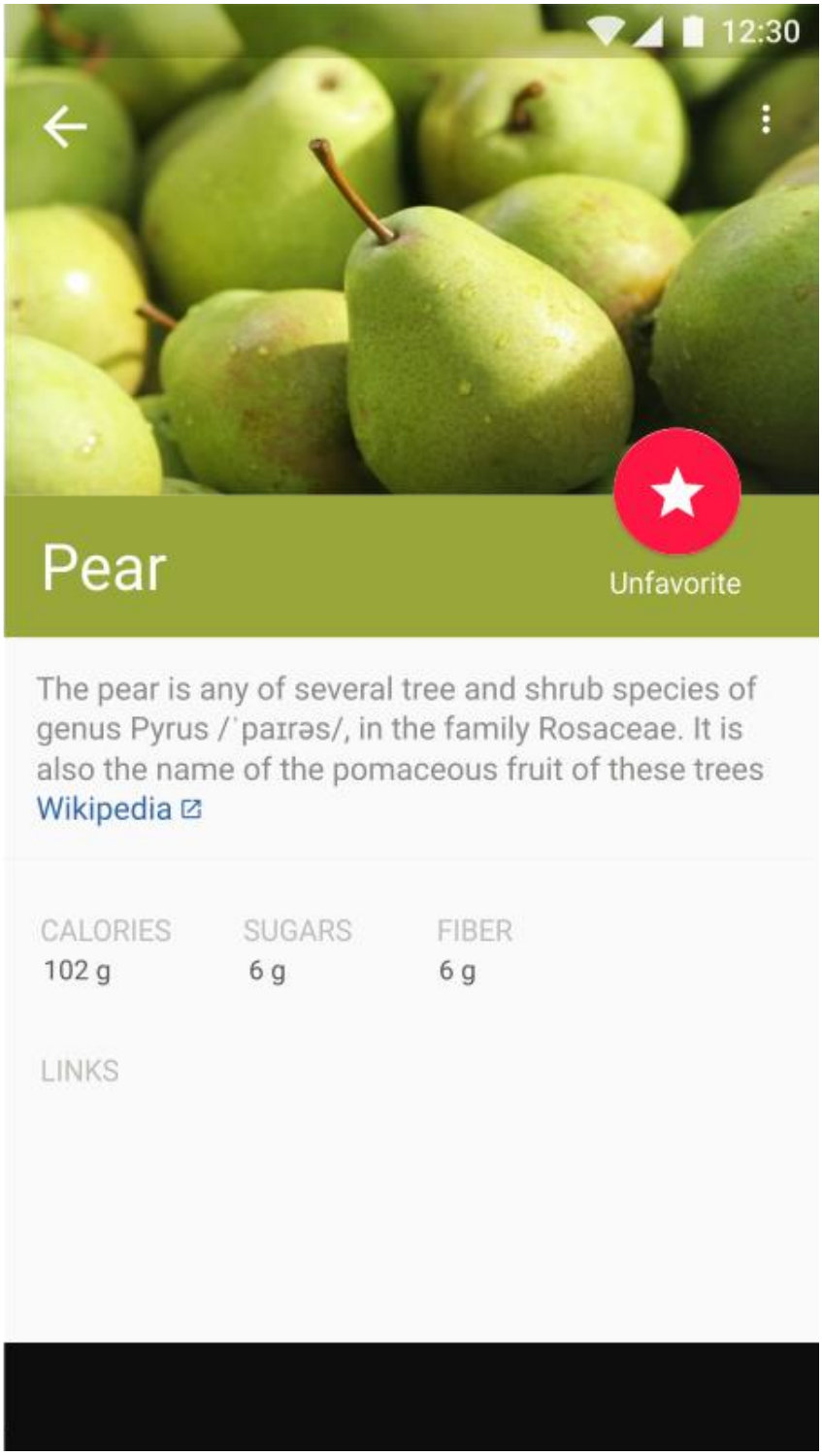
Result screen for manual search.

Screen 8



Results screen for FAB search. Search screen not shown here but will use Google's search view for user input. More here: https://android-arsenal.com/details/1/2589

Screen 9



Detail page for each produce item. Contains desc., photo, nutrition facts and links relevant to the item.

Key Considerations

How will your app handle data persistence?

Describe how your app with handle data.
The app will have a content provider that will feed the data from a MySQL or similar Database. The only user input is a star/favorite feature which will set a notification "reminder to buy at store" for the certain item for each week in the month(s) the item is considered fresh to buy.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?
There is only the Home Screen and Detail screen and there will be a back arrow to go to the previous activity or cancel buttons to close dialogs. From the Home they will be able to query specific items even though they might not be featured (due to time of year) in home screen to favorite it and be reminded when it is relevant.

Describe any libraries you'll be using and share your reasoning for including them.

For example, Picasso or Glide to handle the loading and caching of images.

Glide for image loading and caching. ButterKnife for easy view binding.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Build UI PART 1 (TabView/ViewPager/CollapsingToolbarLayout)

Build the main UI layout composed of three tabs, a menu and a FAB button.

- **MainActivity with three fragments (Favorites, In Season, Months), menu and search FAB.**
- **Menu with two items (Location and About).**

Task 2: Build UI PART 2 (RecyclerView/GridView and list/card items)

Build fragments for each tab in main layout.

- **Favorites tab consists of cardviews with “in season” labels.**
- **In Season tab consists of cardviews with “favorite” labels.**
- **Month tab consists of GridView of each month.**
- **Add empty states for favs and seasons tabs.**

Task 3: Build UI PART 3 (MaterialDesignDialogSearchView/RecyclerView and list/card items)

Additional ResultsActivity screens for the queried results and manual by month results.

- **RecyclerView and card items for manual and query search.**
- **DialogSearchView for FAB button query user input.**

Task 4: Build UI PART 4 (About and Location dialogs)

Describe the next task. List the subtasks. For example:

- **Dialog window for STATE selection and another for ABOUT info.**
- **STATE window will have spinner to choose state location.**

Task 5: Build UI PART 5 (CollapsingToolbarLayout/LinearLayout)

DetailActivity contains product info(photo, desc, links) and FAB button to favorite item.

- **AppBar with parallax scrolling photo.**
- **FAB button to favorite.**
- **Item-relevant links to open in user preferred browser.**
- **Swatch Item title bar to draw color from image palette.**
- **TextViews with item info.**

Task 6: Data Handling (MySQL DB/Content Provider/Custom Object)

Create database schema that will house all data for content provider to access.

- **DB includes Location by State, Produce, Produce in Season (flag?), Favorites (flag?)**

- **Content Provider delivers to the UI.**
- **Each produce item will be a custom object which includes all the data that will be displayed in cardviews and detail page.**

Task 7: First Start Up Set-Up (Dailog/SharedPref)

Dialog will be displayed at first app start up and ask for location, then flag will be set to prevent pop ups in following app starts.

- **Dialog window much like the Location dialog described in "Task 4" asking for user selection.**

Task 8: Notifications for Favorites (Push Notifications)

When user favorites item, a notification will be set to remind the user in a bi-weekly basis to buy the product(s) in season.

- **Simple notification containing a list of items that were favorited and pushed every two weeks.**

Add as many tasks as you need to complete your app.
