Orderella Mobile App Developer Candidate Test

Assignment

The task is to design and implement a (simplified) version of our "Recent Orders" screen.

You should also design and write unit test code that submits appropriate test data to your classes and verifies the results.

Submit your source project as a zip file, back to <engineering +developertest@orderella.co.uk>.
 Assume that it will be compiled and run in the appropriate platform tool (Xcode - iOS - or Android Studio - Android).

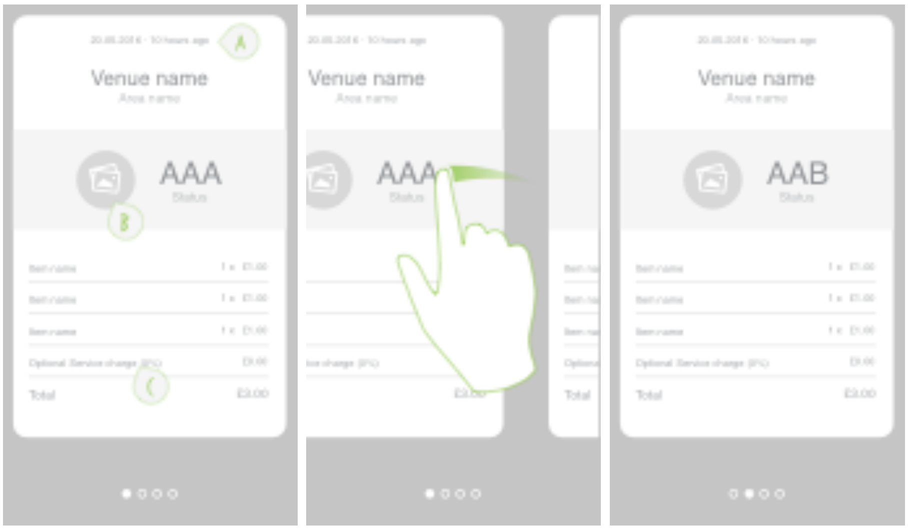
Recent Orders

Each order is a single page of a sideways-scrollable view (e.g. UIPageViewController on iOS, or PagerAdapter on Android), the number of orders as well as the current page is indicated via an appropriate method (e.g. UIPageControl on iOS, PagerTitleStrip on Android) at the bottom.

Some corresponding data, resembling a server response, has been provided in a JSON format file called *recentOrders.json* which you can use to test your implementation.

Use the given data to create the screens dynamically. Status images are in the *images* folder.

(This folder also contains higher resolution versions of the layout, below, for legibility purposes.
Note: we're not looking for a pixel-perfect copy. Just a good, comprehensible solution.)



‘Recent Orders’ data structure

The JSON response is an array of *Order* objects, ordered by most recently updated first.

Object	**Property**	**Type**	**Values**	**Notes**
Order	code	String		3-letter code (not necessarily unique)
	createdOn	DateTime (ISO-8601)		creation time of order
	deliveryType	String	“pickup”, “delivery”	Order is for pickup, or delivery
	status	String	“new”, “preparing”, “ready”, “delivered”	Order state
	tip	Float	0, 8.0, 12.5, 15.0	Percentage tip added by customer to Order
	total	Float		Total value of order, including tip
	deliveryTableNumber	String		Table number, for deliveryType “delivery”
	orderedItems	List of OrderedItem objects		(see below)
	location	Location object		(see below)
	locationArea	LocationArea object		Optional (see below)
OrderedItem	item	Item object		(see below)
	quantity	Integer		Total quantity of this item
Item	id	Integer		Internal database ID of Item object
	name	String		Name of item
	price	Float		Price charged for this item (for quantity:1)

Location	id	Integer		Internal database ID of Location object
	name	String		Name of Location
LocationArea	id	Integer		Internal database ID of LocationArea object
	name	String		Name of LocationArea

Additional information:

A: indicates the age of the order - i.e. the elapsed time from *createdOn* until the current time. It is displayed, alongside the data, in a “friendly form”, i.e.: “a few seconds ago”, “1 min ago”, “n mins ago”, “1 hour ago”, “N hours ago”, “1 week ago”, “N weeks ago” etc.

B: Each Order has one of four states (the status field): new, preparing, ready and delivered. A label corresponding to the state is displayed below the order code, which varies according to the delivery type.

For deliveryType=“delivery”:

new	- “New”
preparing	- “Received”
delivered	- “Delivered”

For deliveryType=“pickup”:

new	- “New”
preparing	- “Received”
ready	- “Ready”
delivered	- “Collected”

and the icon shown matches the status field name (i.e. ‘new’ is shown with status_new.png)

(Optional credit: In our current app, the icon spins clockwise during new and preparing, and stops when ready or delivered.)

C: If the “tip” is zero (0), then no Tip/Optional Service charge has been added: omit the service charge from the display. The “tip” is a percentage of the total item price to be added to get the final total.

For example: if the “tip” = 10, and the total of the items - Sum (quantity * price) - is 5, then Tip amount is 5 * 0.10 = 0.50.