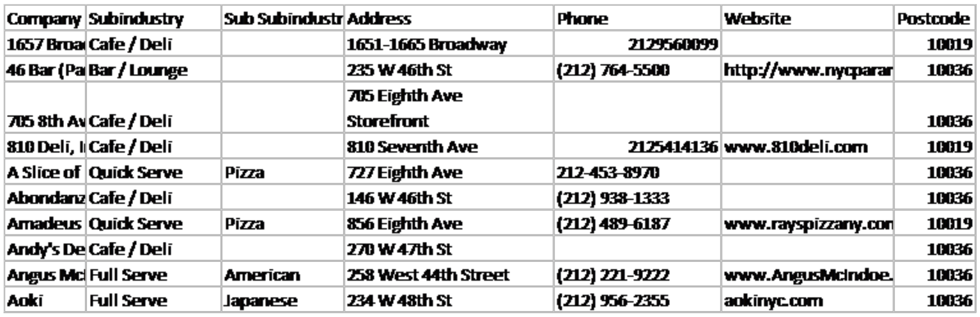
HW2 NY Street Food

This homework exercises your abilities in implementing multiple Fragments and using Navigation Graph to achieve Fragment navigation. The task asks that you properly create Fragments in a navigation graph, and manage and pass parameters between Fragments. The application to be developed is a New York Street Food app. The requirements are:

1. The application uses the CSV data provided on Moodle, which is a spreadsheet of 331 street food companies near Time Square, NYC. Each entry contains information of one restaurant, including company name, subindustry, sub subindustry, address, phone, website, and postcode. Note some fields might be empty for a restaurant.



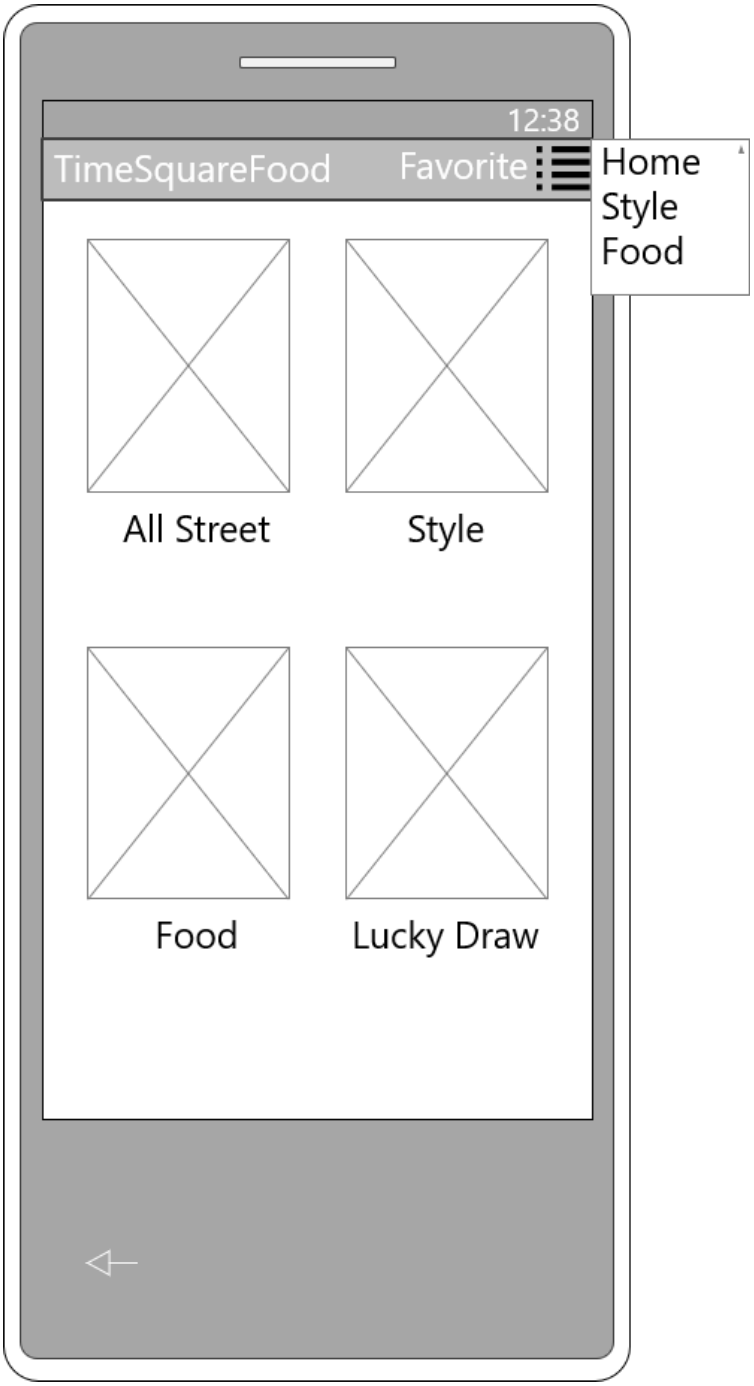
Company	Subindustry	Sub Subindustr	Address	Phone	Website	Postcode
1657 Broa	Cafe / Deli		1651-1665 Broadway	2129560099		10019
46 Bar (Pa	Bar / Lounge		235 W 46th St	(212) 764-5500	http://www.nycparar	10036
705 8th Av	Cafe / Deli		705 Eighth Ave Storefront			10036
810 Deli, I	Cafe / Deli		810 Seventh Ave	2125414136	www.810deli.com	10019
A Slice of	Quick Serve	Pizza	727 Eighth Ave	212-453-8970		10036
Abondanz	Cafe / Deli		146 W 46th St	(212) 938-1333		10036
Amadeus	Quick Serve	Pizza	856 Eighth Ave	(212) 489-6187	www.rayspizzany.con	10019
Andy's De	Cafe / Deli		270 W 47th St			10036
Angus Mc	Full Serve	American	258 West 44th Street	(212) 221-9222	www.AngusMcIndoe.	10036
Aoki	Full Serve	Japanese	234 W 48th St	(212) 956-2355	aokinyc.com	10036

2. The application must contain at least 6 screens. For this homework, you must use Fragments to implement all screens. The 6 screens are the **home screen, list screen, style category screen, food category screen, saved restaurant screen, and detail screen. You must properly use the navigation graph to implement the navigation between different screens.** Use the provided pictures to decorate the views and the application.

3. **Detail screen** shows the actual restaurant details. In this screen, the user must be able to access all available information of a restaurant provided in the spreadsheet. The user must be able to make a phone call from your app by clicking the phone number. The user must be able to view the website from this screen. You must properly use GUI components to handle missing values in some fields. This screen must show a picture banner to indicate its style category ("subindustry" field) and a 5-scale rating bar to let users rate this restaurant.

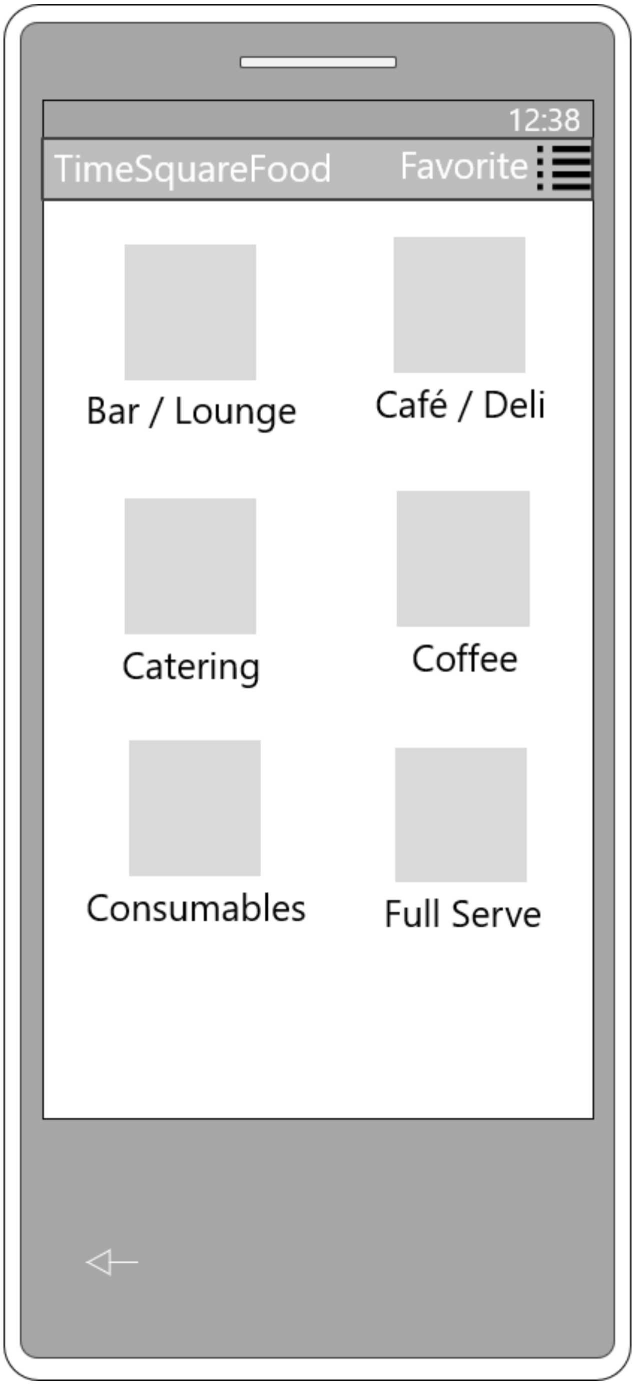
4. **Home screen** launches when the app is first opened. On this screen, there should be four image buttons which provide 4 different ways to browse the restaurants: browse all restaurant, browse by restaurant styles

("subindustry"), browse by food category ("sub subindustry"), and browse a random restaurant. "All restaurant" button navigates to the **list screen**, which shows all restaurant restaurants in the CSV file. "Style" button navigates to **style screen**. "Food" button navigates to **food screen**. "Lucky draw" button shows a **detailed screen** of a random restaurant from the CSV file. Every time the user clicks a lucky draw button, a different restaurant should show up.

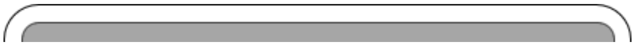


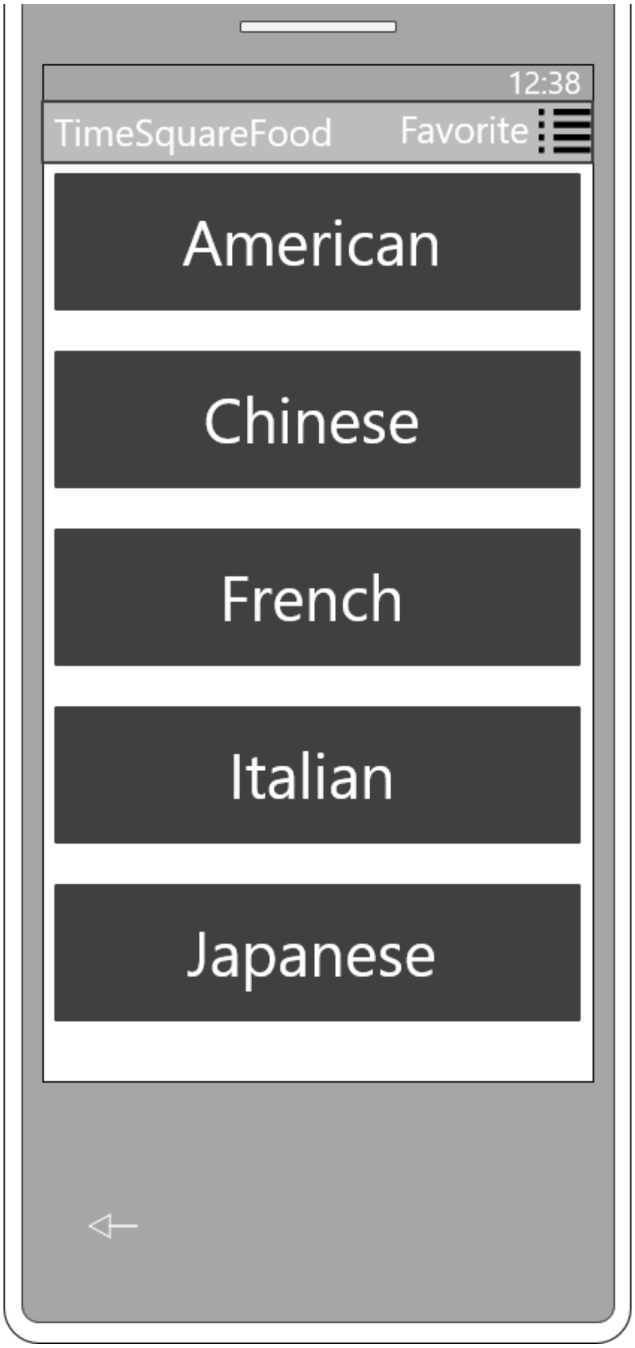
5. **Style screen** must contain a GridView to show all 7 categories of restaurant styles as in the subindustry column (bar/lounge, Café/deli,

restaurant styles as in the subindustry column (bar/lounge, Cafe/deli, Catering, Coffee, Consumables, Full Serve, and Quick Serve). Each item must contain a name and a category picture (you can use the provided pictures). Clicking an item navigates to the **list screen** and shows a list of restaurants of that subindustry.

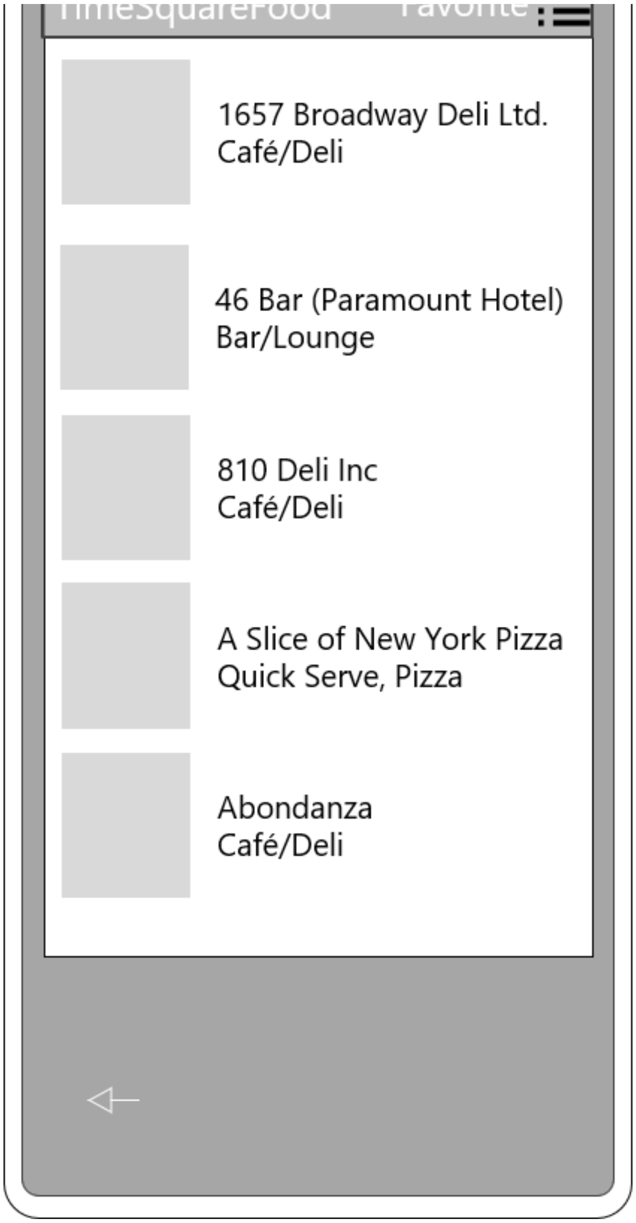


6. **Food screen** must contain a recycler view which shows 10 major food categories (American, Chinese, Japanese, Frech, Italian, etc.) Each item must contain a background image and a text indicating the category. Clicking an item navigates to the **list screen** and shows a list of restaurants of that sub subindustry.





7. **List screen** must contain a recycler view which shows a list of restaurants. The list is filtered based on how the user comes to this screen (all restaurants/style/food). Each item must contain a picture indicating its style (subindustry), the restaurant name, and its style name (subindustry). Clicking an item launches the **detail screen** and shows the information of the selected restaurant.



1657 Broadway Deli Ltd.
Café/Deli

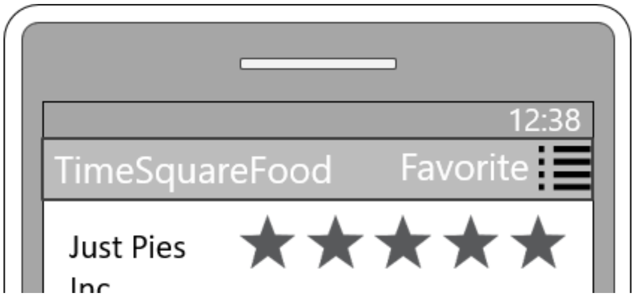
46 Bar (Paramount Hotel)
Bar/Lounge

810 Deli Inc
Café/Deli

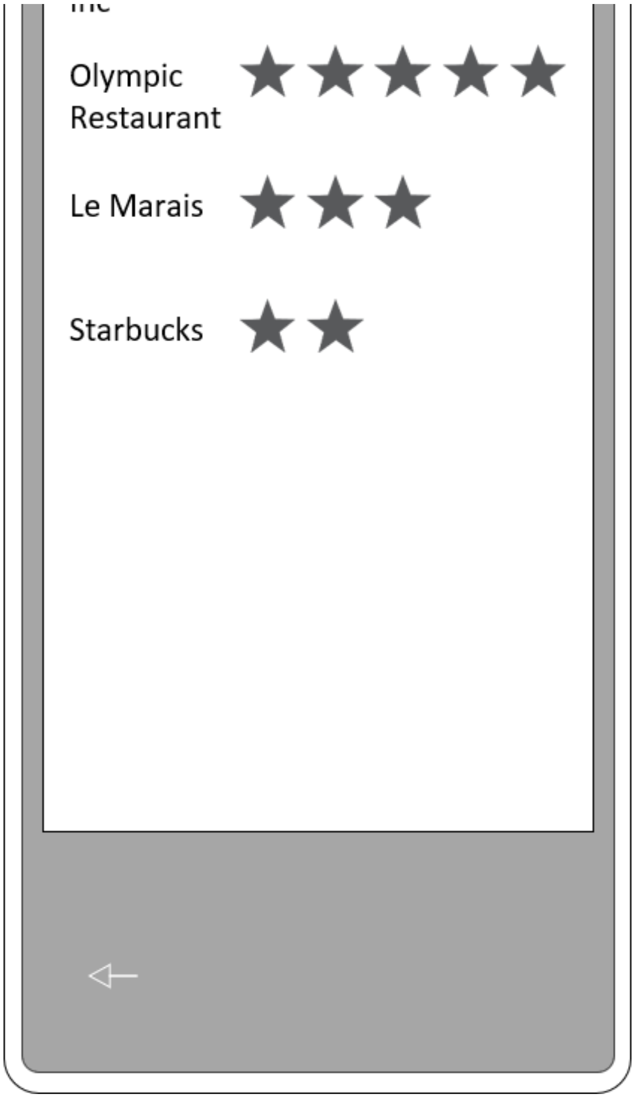
A Slice of New York Pizza
Quick Serve, Pizza

Abondanza
Café/Deli

8. **Saved Restaurant Screen** can be accessed from the "favorite" button on the Action Bar. This screen contains a RecyclerView showing a list of restaurants which are rated by the user in the detailed screen. Each item shows the restaurant name and a rating bar. The restaurants are ordered by their ratings in descending order. Clicking on the restaurant navigates to the detailed screen.



Just Pies Inc ★★★★★



9. The action bar must contain a menu button (hamburger button) to facilitate navigation. Clicking the menu button shows a menu item to go to the home screen, a menu item to style screen, and a menu item to food screen. The action bar should also show the name of the current screen and a back button to go to the previous screen.
10. The navigation graph and GUI of the app must be properly implemented. Use the resources provided on Moodle to decorate your app. Provide instructions or hints by using Toast or Dialogs for improper interaction. The navigation of different Fragment should be direct and intuitive.

11. All required screens (Fragments) must be accessible from one to another. Make sure the user is not stuck at one fragment and won't be able to navigate to other screens.

12. The data, including but not limited to, the filtered restaurants, ratings, and navigation information must be consistent throughout the app usage.

You can assume the user does not press the home button, power button, app-overview button or rotate the screen while using the app. The data can be inconsistent only during those cases. But the data must be consistent while the user switches to different screens of the app.

Your homework will be graded based on the following criteria:

- Required GUI and user input is properly designed and implemented (30%)
- Data is consistent throughout the navigation (30%)
- User navigation is properly implemented (30%)
- Bug-free (10%)

DO NOT RE-SUBMIT YOUR ASSIGNMENT AFTER THE DEADLINE, IT

WILL BE CONSIDERED AS LATE SUBMISSION.