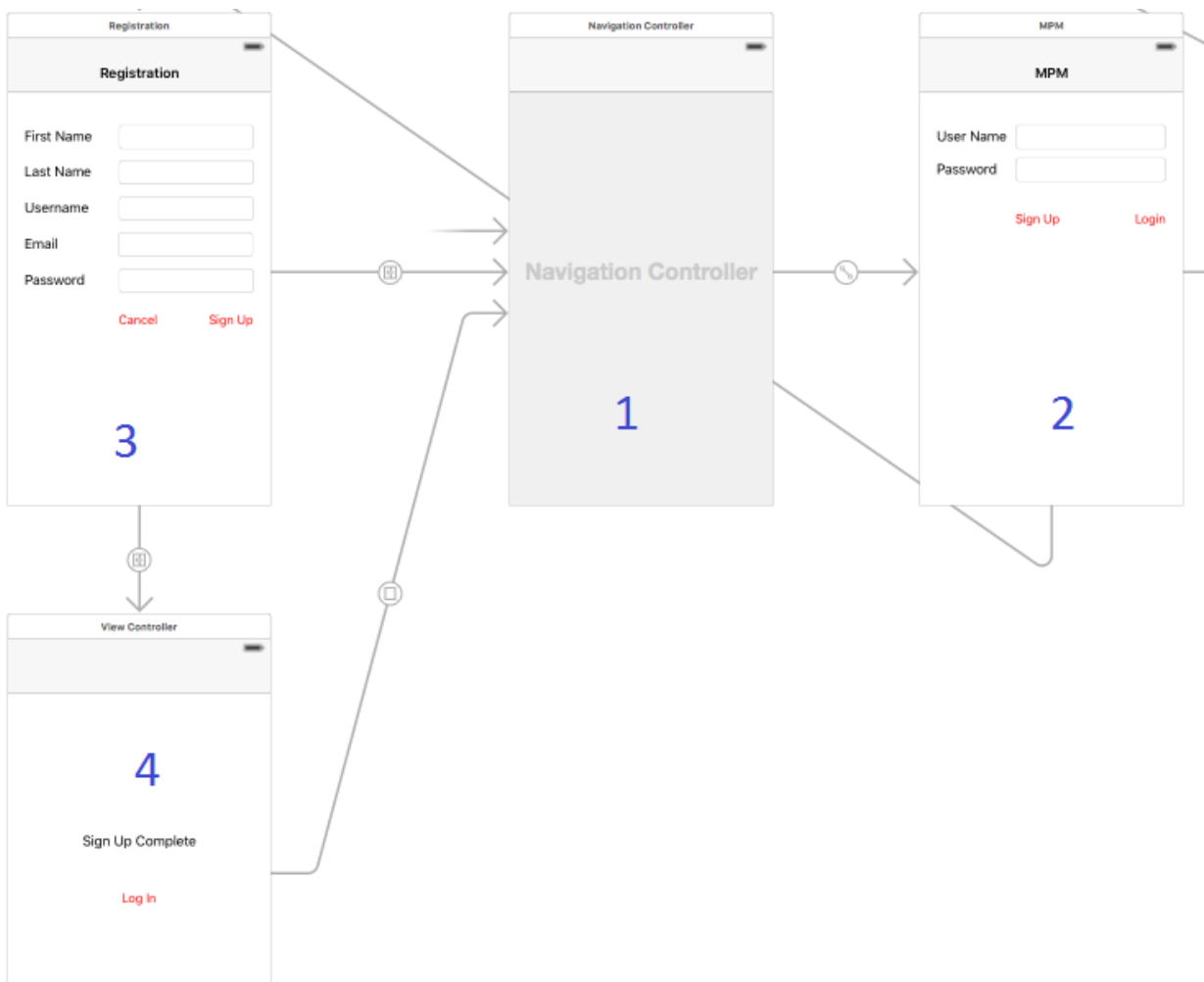


Design HW2: Components & Interfaces for MyPerfectMeal

- 1) Navigation Controller: This is the initial launch screen. This controller simply tells the app to embed all following view controllers into a navigation mode so the user can go back and forth between all view controllers. As soon as the app launches, the navigation controller (1) will send the user to the LoginViewController (2)

REGISTRATION PROCESS



- 2) This View controller allows access to the application or allows the user to create a user account with MPM.

if login selected

 check if user login good

 if so go to tab bar view controller

else

 return detailed error

- 3) Assuming the user decides to create an account, they will end up on the SignUpViewController
(3) All text fields are required.

If submit button pressed

 Check parameters

 If all parameters are accepted

 Create a user

 Submit all textfields to database

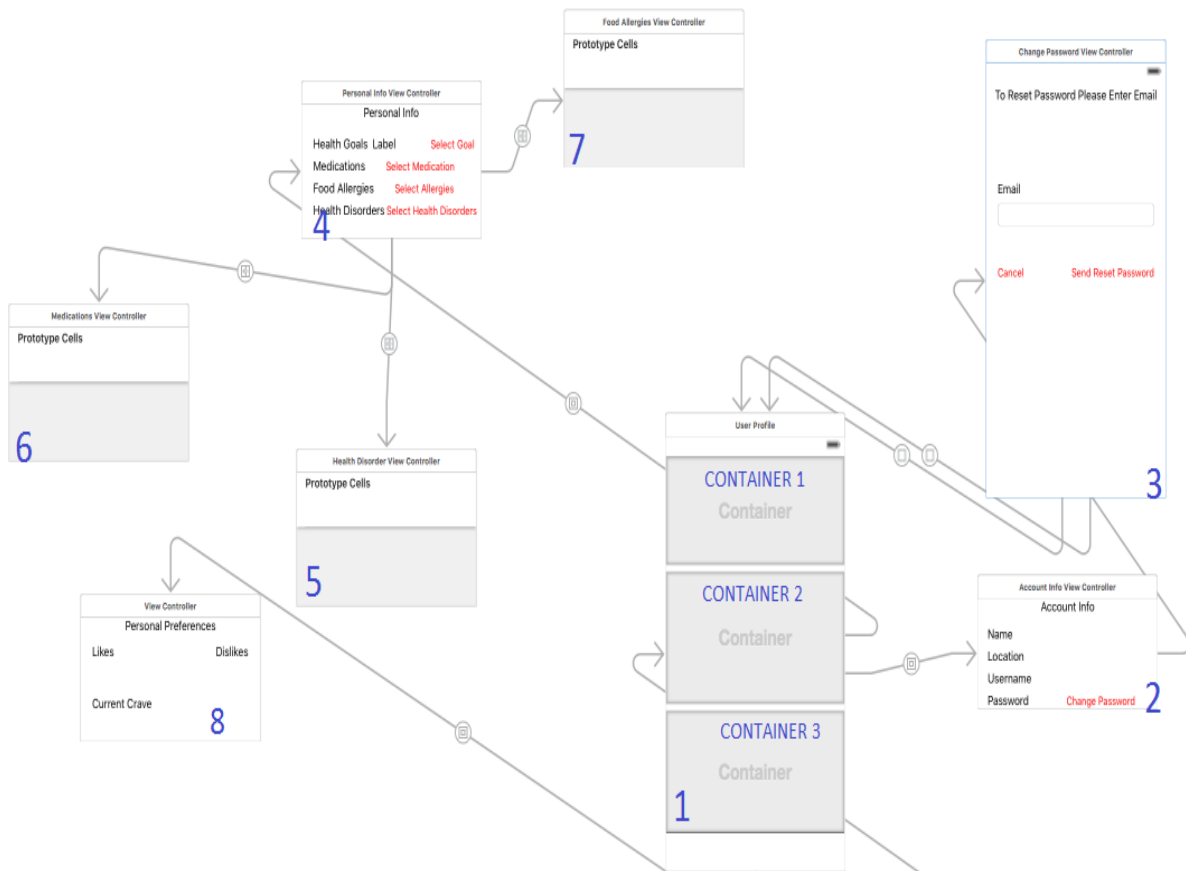
 Secretly grab and submit user location using gps

Else

 Display detailed error why

- 4) If all the fields have been properly filled out lets show the user their signup is complete and link them back to view controller (2) so they can login.

USER PROFILE TAB



- 1) Once the user is logged in, the first tab that shows up in the tab navigation controller is the User Profile Table.

The USER PROFILE TAB is divided into 3 Containers for organizational purposes

- a. Container 1: User account Management view controller represented by view controller (2)
- b. Container 2: Personal info view controller represented by view controller (4)
- c. Container 3: Personal Preferences view controller represented by view controller (7)

- 2) The first container shown in (2) pulls the user's name, location, and username

```
nameLBL.text = [NSString stringWithFormat:@"%s %s", [[PFUser currentUser]
valueForKey:@"First_Name"], [[PFUser currentUser] valueForKey:@"Last_Name"
]];
```

```
locationLBL.text = [NSString stringWithFormat:@"%s", [[PFUser currentUser]
valueForKey:@"currentLocation"]];
```

```
userNameLBL.text = [NSString stringWithFormat:@"%s", [[PFUser currentUser]
valueForKey:@"username"]];
```

- 3) Assuming the user clicks on “Change Password” the user will be taken to view controller (3), where they can input their email and have a reset password link emailed to them.

If current user email is the same as text in textfield

Send the request:

```
[PFUser requestPasswordResetForEmailInBackground:emailResetTF.text];
```

Else

Show alert, reset attempt has failed check parameters

- 4) The second container is represented under (4). This allows the user to select some of their personal information. Assuming the user clicks on “Select goal”, a View Picker will popup with a few options. The option selected will replace “Label” on the same line.

```
numberOfComponentsInPickerView: 1 column
```

```
numberOfRowsInComponent: as many as the number of rows from the database
```

```
return [goalsArray objectAtIndex:row]: name to display in the pick view
```

```
[selectedGoalLBL setText: [goalsArray objectAtIndex:row]]; display selection  
on “Label”
```

HIDE OR SHOW THE PICKVIEWER IF USE CLICKS BUTTON

```
-(void) HideORShowPV{  
    if([selectGoalsPV isHidden]){ //initially this is true, turned off in the  
        storyboard  
  
        [selectGoalsPV setHidden:NO];  
  
    }  
    else{  
        [selectGoalsPV setHidden:YES];  
    }  
}
```

- 5) Assuming the user clicks on “Select Health Disorders” the user will be taken to view controller (5) where they can “select” health disorders, the selections are displayed using “checkmarks”

Pull maximum 1000 entries of healthdisorders from the database, add them to a mutable array "HealthDisorders" and display them in the cell.

```
PFQuery *query = [PFQuery queryWithClassName:@"healthDisorders"];
[query setLimit: 1000];
[query findObjectsInBackgroundWithBlock:^(NSArray *objects, NSError
*error) {
    if (!error) {
        for (PFObject *object in objects) {

            [HealthDisorders addObject:[object objectForKey:@"HDName"]];
            //NSLog(@"Disorder name in parse is: %@", HealthDisorders);
            [self loadObjects];
        }
    }

}];
```

- 6) Assuming the user clicks on "Select Medications" the user will be taken to view controller (6) where they can "select" medications, the selections are displayed using "checkmarks"

Query all medications, save each object from the database to a mutable array: "Medications" display each entry into the cell

```
PFQuery *query = [PFQuery queryWithClassName:@"Medications"];
[query setLimit: 1000];
[query findObjectsInBackgroundWithBlock:^(NSArray *objects, NSError
*error) {
    if (!error) {
        for (PFObject *object in objects) {

            [Medications addObject:[object objectForKey:@"medName"]];
            //NSLog(@"Meds in parse are: %@", Medications);
            [self loadObjects];
        }
    }

}];
```

- 7) Assuming the user clicks on “Select Allergies” the user will be taken to view controller (7) where they can “select” medications, the selections are displayed using “checkmarks”

Query food allergies from database, display max 1000 results, store results in mutable array “foodAllergies”, display these results into the cell

```
PFQuery *query = [PFQuery queryWithClassName:@"foodAllergies"];
[query setLimit: 1000];
[query findObjectsInBackgroundWithBlock:^(NSArray *objects, NSError
*error) {
    if (!error) {
        for (PFObject *object in objects) {

            [foodAllergies addObject:[object
objectForKey:@"foodAllergen"]];
            //NSLog(@"Allergen name in parse is: %@", foodAllergies);
            [self loadObjects];
        }

    }

}];
```

- 8) Container (3) will take the user to view controller (3) where their likes and dislikes are shown, this is currently being developed.

RESTAURANT TAB



- 1) The Tab Bar Controller (1) allows the addition of multiple tabs. Recall the first tab was the USER PROFILE, now we will discuss the Second tab: Restaurants
- 2) We use another navigation controller (2) to embed the subsequent view controllers in a navigation capable mode
- 3) Once the user clicks the tab “Restaurants” at the bottom, they will automatically be taken to view controller (3). This controller will pull all restaurants from the database. Sectionalize and alphabetically order the restaurant based on their Name. If the user click on the top left item “Refresh”, the table will reload with potentially more restaurant if another user has added a restaurant

Using a NSMutableDictionary store the the StartWith letter of each restaurant to a section. In total we will have 26 sections because the alphabet is 26 letters long.

So if a restaurant starts with A assign it section 0, if restaurant starts with I assign it section 1. As we add more and more restaurants reorder the dictionary so that it's in alphabetical order.

sections to startwith

0	A
1	I
2	W

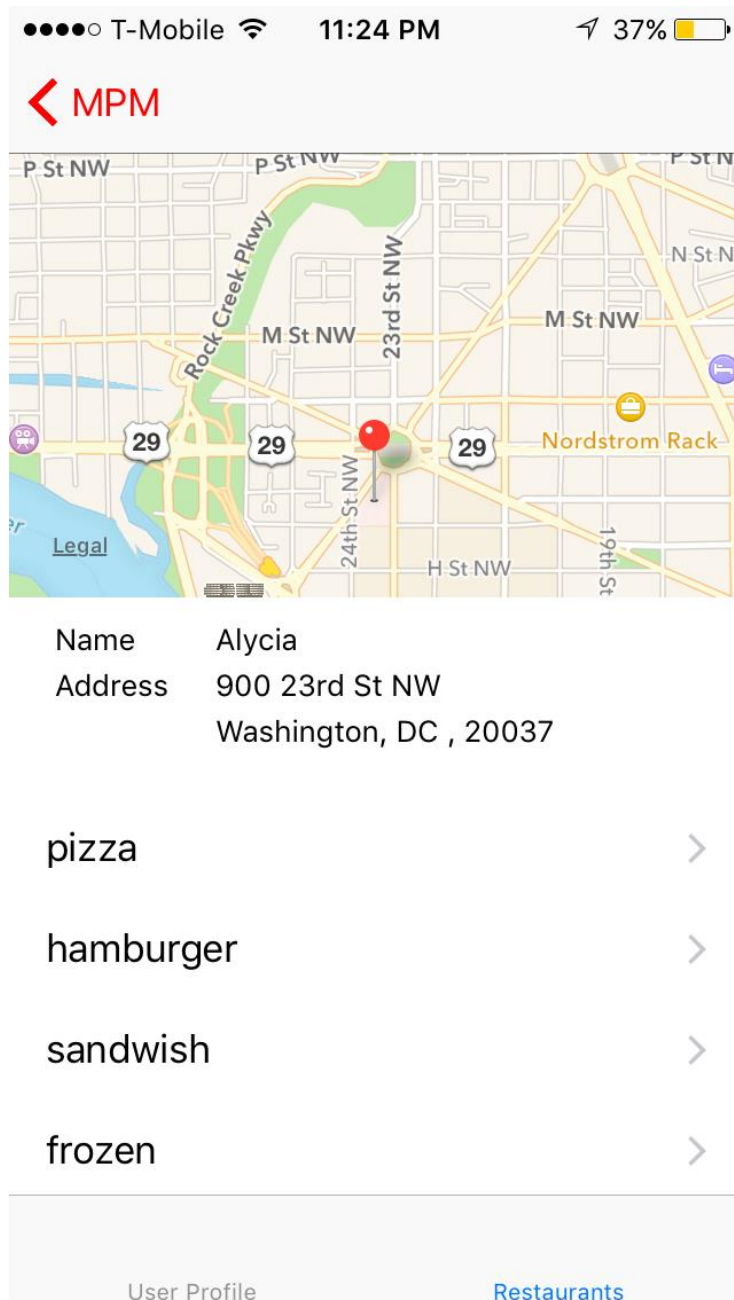
Using a second NSMutableDictionary map the startwith to how many objects (in our case restaurants) for each section.

Startswith

A	2 objects
i	1 object
w	1 object



- 4) Assuming the user clicked on the “+” on the top right corner of view controller (3) they will be taken to view controller (4). This controller will allow any user to add restaurants
- 5) If the user selects a restaurant from view controller (3) they will be taken to view controller (5). This view controller contains a container (6) that is displayed at the top portion of the screen.
- 6) The container (6) will display the location of the restaurant on a map and will display the information (name and address) at the bottom of the container.



- 7) If the user decides to add a meal for that specific restaurant by clicking “+” at the top right corner of view controller (5), they will be taken to view controller (7). On this view controller (7) the user can simply enter a meal and add it to the list of meals.

- 8) Assuming the user selects meal from view controller (5), they will be taken to view controller (8). This will display the ingredients included in the meal previously selected. The user can select multiple ingredients to be selected using a checkmark and added into their preferences under container 3 under their user profile.
- 9) Assuming the user clicks on the “+” on view controller (8) they will be taken to view controller (9) where they can add an ingredient for that meal.
- 10) The restaurants tab also includes a search bar, which is although show in the storyboard, it is not implemented.

RESULTS TAB

Although the results tab is not included in the tab bar controller, it will be implemented in the second semester where it will display the results of the algorithm.