



My FavCoffee App

Team More Cowbell:

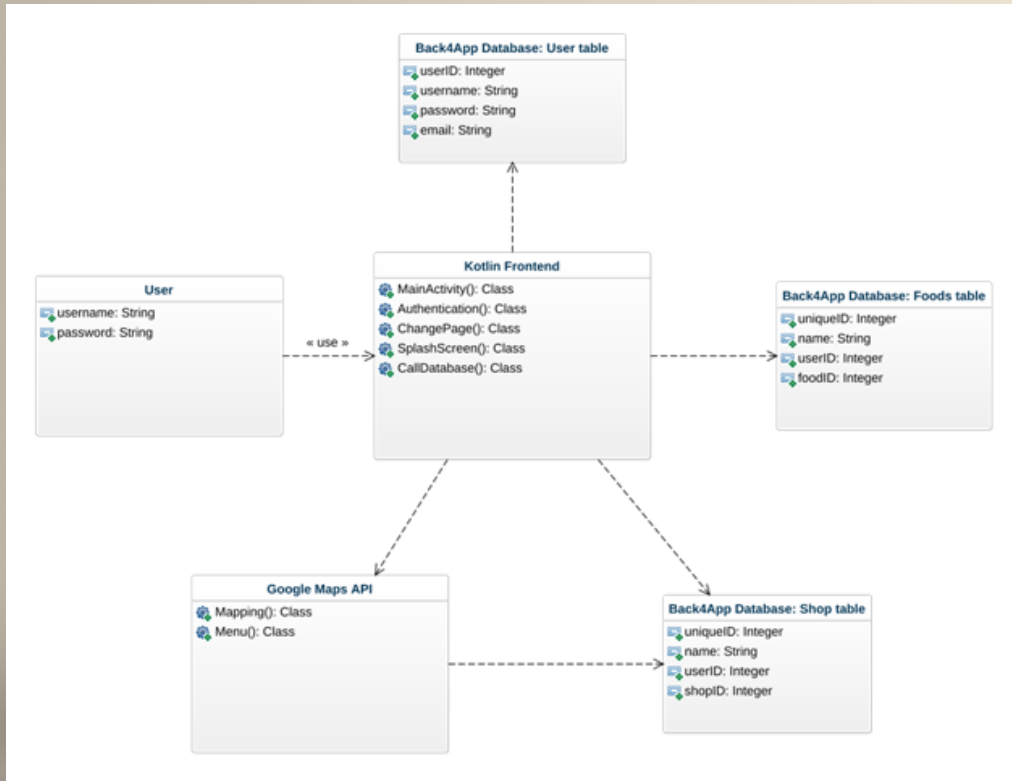
Kyu, John, Mario, Luiza, Kyle

OUR BIG IDEA

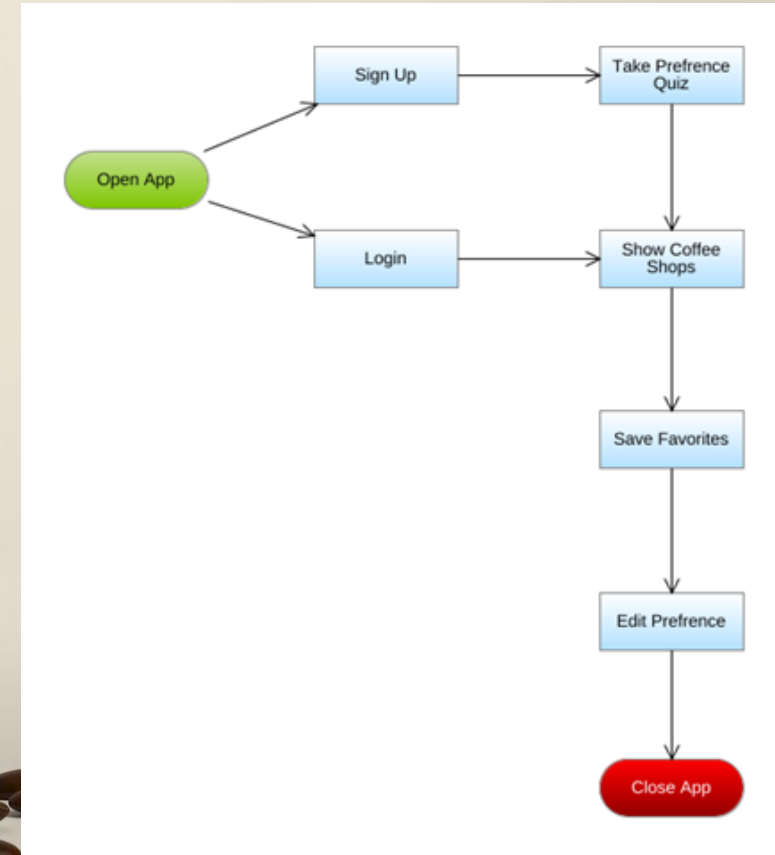
My FavCoffee's mission is to make it possible to find a new coffee shop place that the user has higher chances of liking because it matches their personal preferences and it's close to the user's location! Also, the user has the option to save their favorite ones, and have their own favorite list of coffee shops, independently of their location.



Design Documents

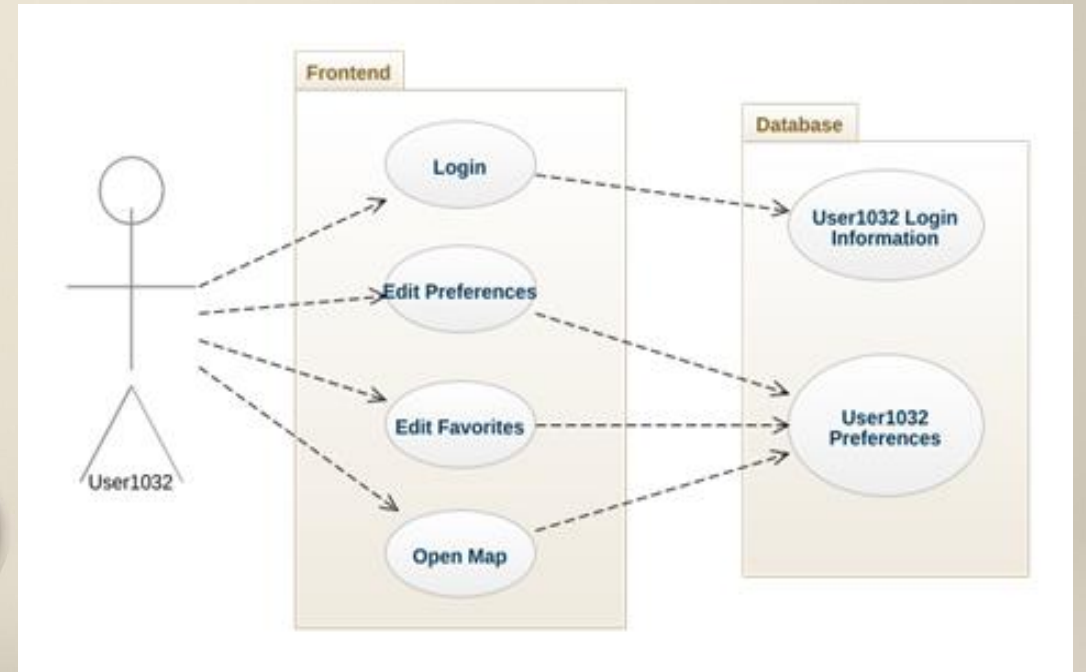
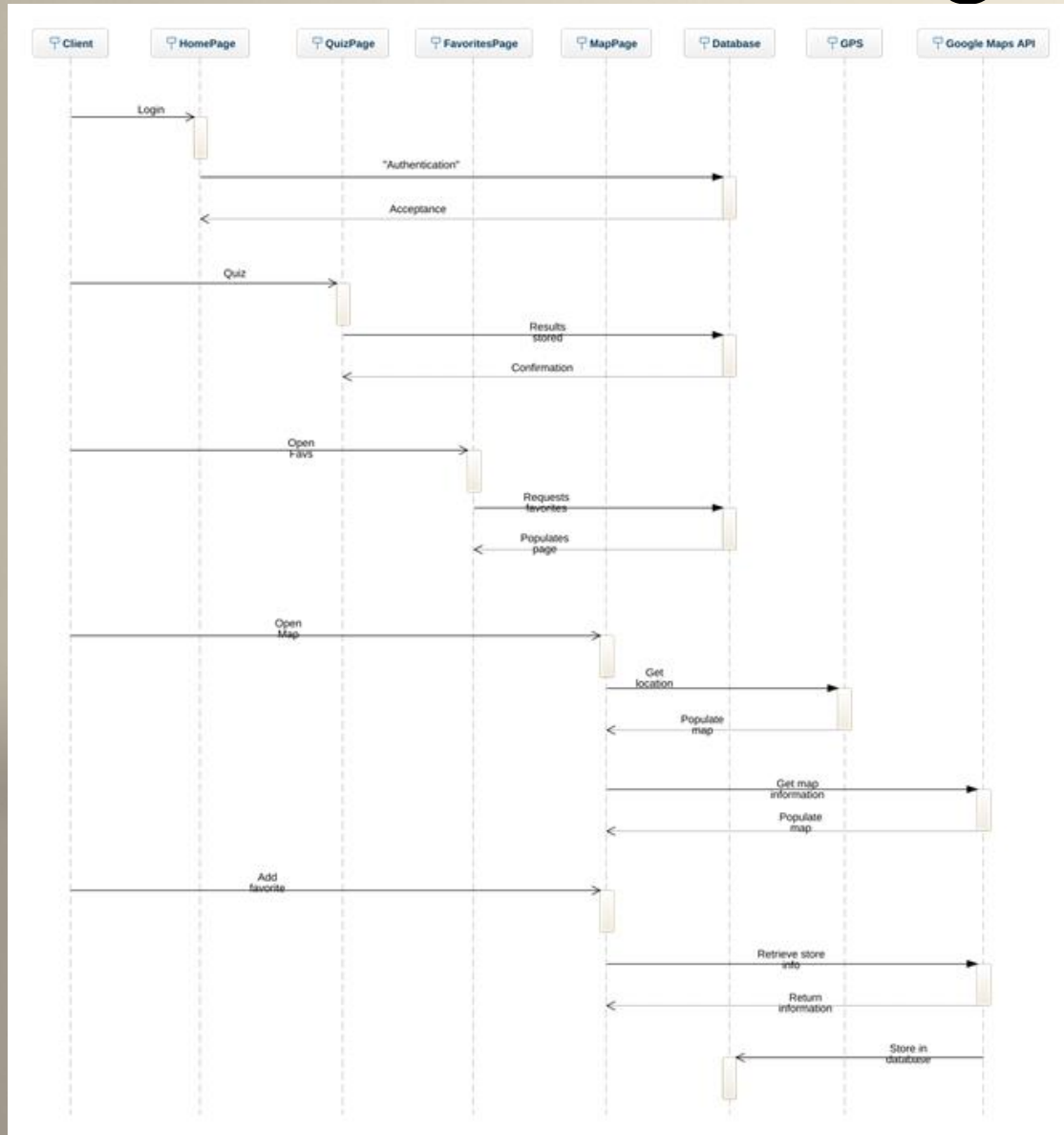


Class Diagram

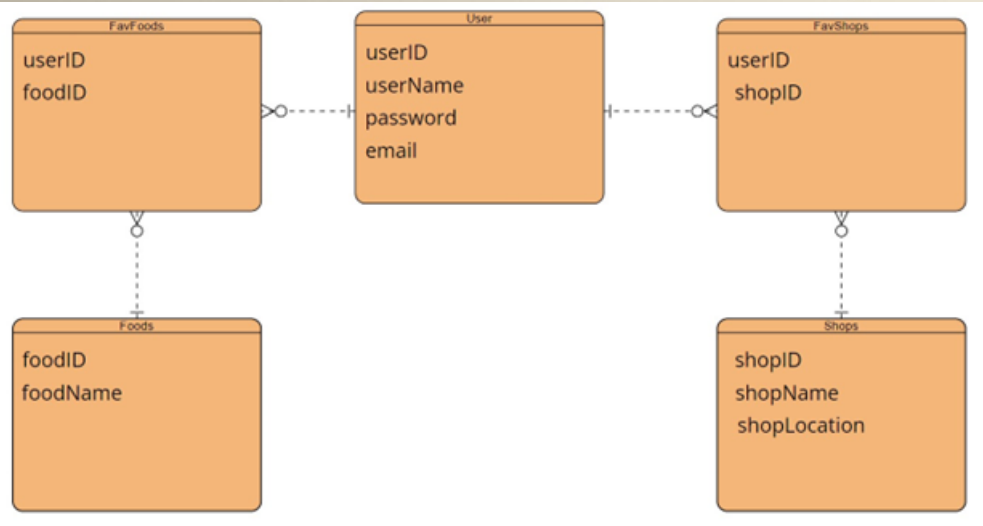


Activity Diagram

Design Documents



Database



- Parse database stored on a Back4App server
 - Extremely simple implementation requiring little to no upkeep
- Basic structure involves storing information on three objects
 - User, Shop, Food
- User-specific preferences are saved as tuples in the relevant tables

15 objects

MyFavCoffee_Shop

	shopId	shopName	latitude	longitude
<input type="checkbox"/>	1	Capitol Coffee Works	47.6156928534968	-122.32064786604806
<input type="checkbox"/>	2	Cascade Coffee Works	47.62298866949087	-122.33523547659989
<input type="checkbox"/>	3	Pegasus Coffee Bar	47.609796628653754	-122.33627420053199
<input type="checkbox"/>	4	Analog Coffee	47.6223443703369	-122.32574978042835
<input type="checkbox"/>	5	Elm Coffee Roasters	47.600937978742735	-122.33134633216545
<input type="checkbox"/>	6	Better Buzz Coffee	32.75203981660922	-117.15767389578889
<input type="checkbox"/>	7	Dark Horse Coffee Roast...	32.769073918813696	-117.1219683295316
<input type="checkbox"/>	8	Seven Seas Roasting Com...	32.73355846223757	-117.1288347845811
<input type="checkbox"/>	9	Communal Coffee	32.763765406884495	-117.14231814239241
<input type="checkbox"/>	10	The Forum Coffee House	32.831998504844876	-117.18406459139895
<input type="checkbox"/>	11	5 Stones Coffee Co.	47.58696894748818	-122.14311717468456
<input type="checkbox"/>	12	Cypress Coffee Company	47.58349527874228	-122.14003040816425
<input type="checkbox"/>	13	Mini's Coffee & Donuts	47.578856597606	-122.169755821256
<input type="checkbox"/>	14	Copper Kettle Coffee Bar	47.610708765945525	-122.17898864841133
<input type="checkbox"/>	15	Woods Coffee	47.617545478996725	-122.18722706839905

- Here you can see the favShop table pairing the userId with the shopId, indicating this user has favorited Capitol Coffee Works

1 object

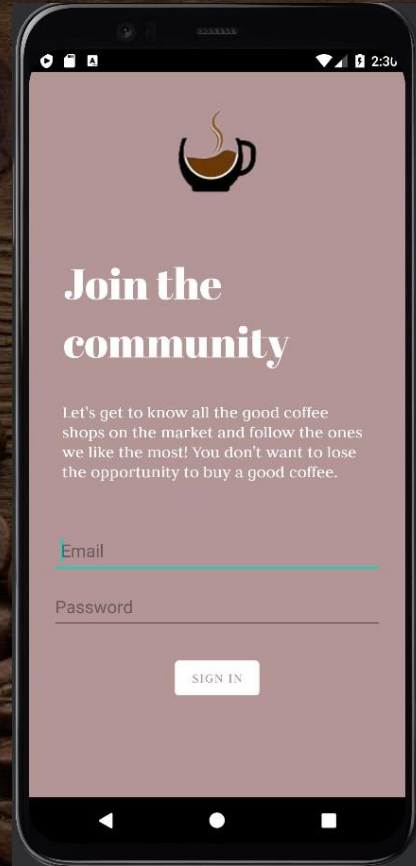
MyFavCoffee_favShop

	userId	shopId
<input type="checkbox"/>	1	1

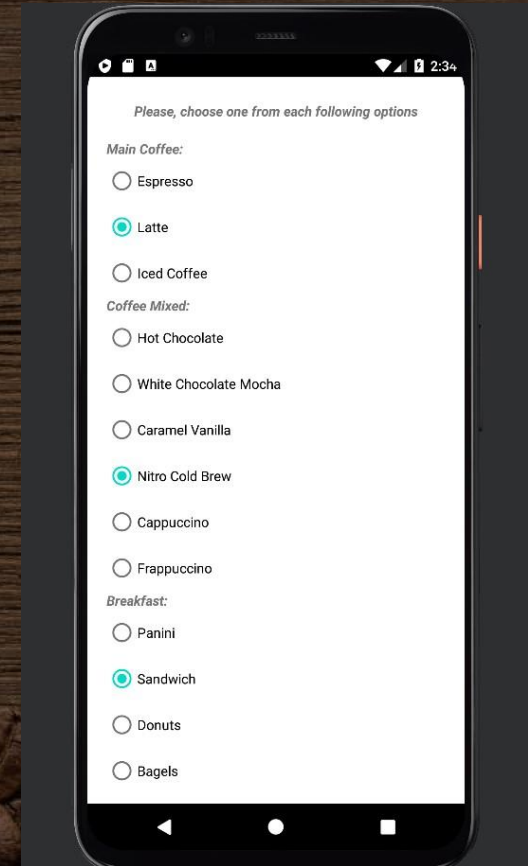
User Interface Design



Splash screen



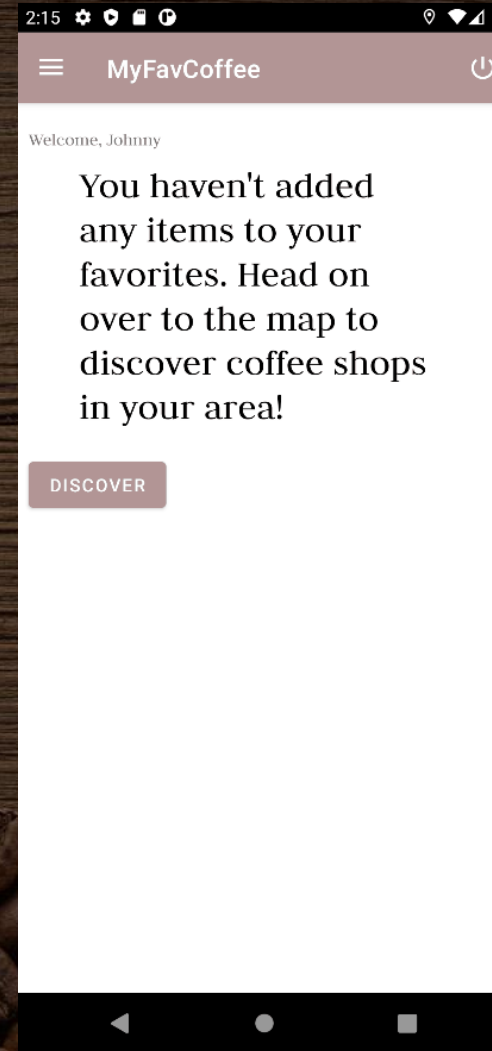
Login page



User Quiz page

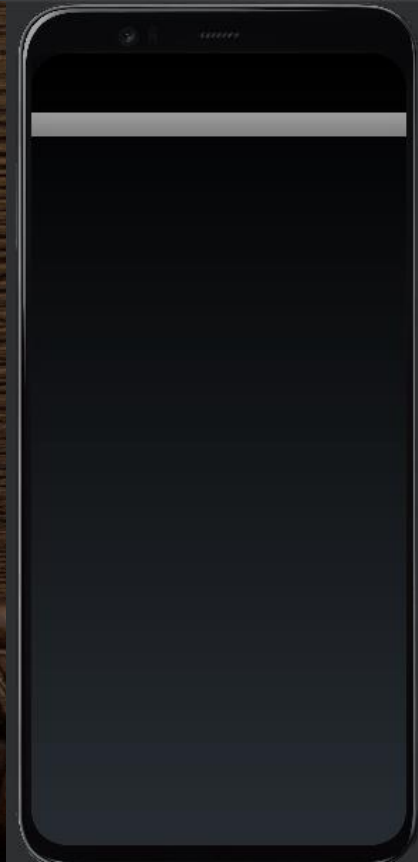
Integrating the GPS sensor and the Places API

```
override fun onMapReady(googleMap: GoogleMap) {  
    map = googleMap  
  
    var latitude = 47.6062  
    var longitude = -122.3321  
    var seattleLatLng = LatLng(latitude, longitude)  
    var zoomLevel = 12f  
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(seattleLatLng, zoomLevel))  
    map.addMarker(MarkerOptions().position(seattleLatLng).title(title: "Emerald City")?.tag = -1  
    map.setOnMarkerClickListener(this)  
  
    // set markers on the map  
    for (item: CoffeeShop in dbObj.coffeeShops) {  
        latitude = item.latitude  
        longitude = item.longitude  
        var currentMarker = LatLng(latitude, longitude)  
        map.addMarker(MarkerOptions().position(currentMarker).title(item.ShopName)?.tag =  
            item.ShopId  
        map.setOnMarkerClickListener(this)  
    }  
  
    setMapLongClick(map)  
    setPoiClick(map)  
    setMapStyle(map)  
    enableMyLocation()  
}
```



Testing

Tests	Duration	Pixel_4_API_30
Test Results	1 s	1/2
ActivityLoginTest	1 s	1/2
LoginTitleRender_ShouldBePresent	1 s	✓
useAppContext	159 ms	✗



```
@Test
fun d_PreferenceQuizSubmission_ShouldDirectUserToShopActivity() {
    //
    val shouldNotBeSelectedArray = arrayOf<Int>(R.id.Bagels, R.id.donuts,
    val shouldBeSelectedArray = arrayOf<Int>(R.id.sandwich, R.id.Espresso,
    //Iterate through clicks
    for (i in shouldBeSelectedArray.indices) {
        onView(withId(shouldBeSelectedArray[i])).perform(click())
    }
    // iterate to ensure buttons are selected
    for (i in shouldBeSelectedArray.indices) {
        onView(withId(shouldBeSelectedArray[i]))
            .check(matches(isChecked()))
    }
    // iterate to ensure buttons are NOT selected
    for (i in shouldNotBeSelectedArray.indices) {
        onView(withId(shouldNotBeSelectedArray[i]))
            .check(matches(isNotChecked()))
    }
    //Click Form Submit
    onView(withId(R.id.btnSubmit))
        .perform(scrollTo())
        .perform(click())
    //Check for activity_main element.
    // **** Helper functions wait on element to load.
    //onView(isRoot()).perform(waitForView(R.id.test, 5000))
    Thread.sleep(millis: 5000)
    onView(withId(R.id.btn_map_activity))
        .check(matches(isDisplayed()))
}
```



Challenges



- **Fragments vs. Activities and navigating between the two**
 - **Solution? Make everything an Activity**
 - **PreferenceQuiz in particular had to be re-made**
- **Accessing data from the database on multiple pages**
 - **Solution? John created a dbHelper object to easily read current state of database, cutting down on unnecessary db interaction**
- **Complexity of a real Login method for the application**
 - **Solution? Stub the login process**

Thank You!

Questions?

