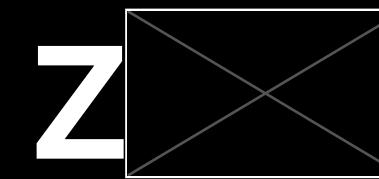


# Anatomy of a What's Open Client

 Knox @zmknox

high level  
overview

# Anatomy of What's Open for iPhone

# What is What's Open?

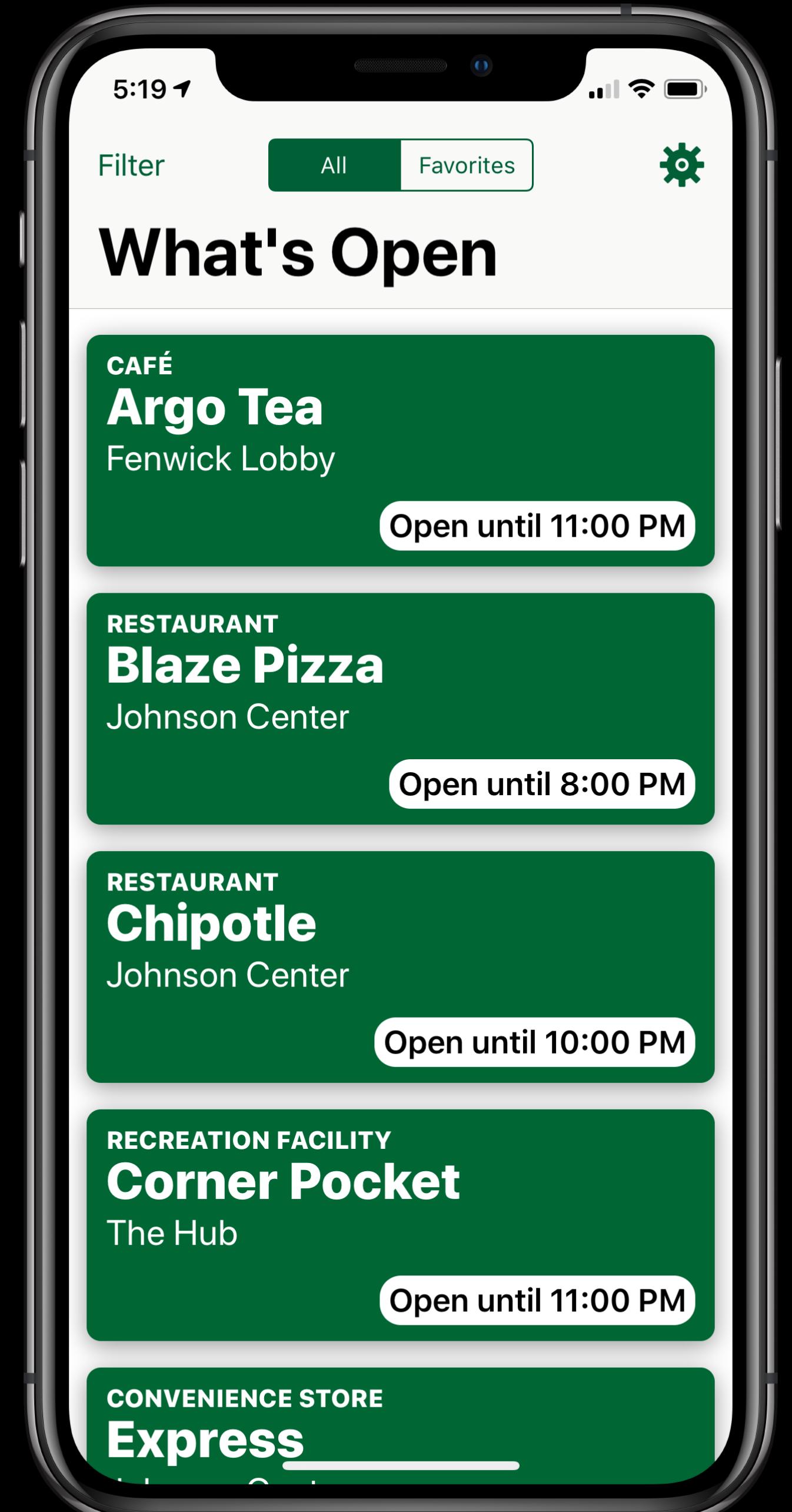
```
"modified": "2018-01-21T13:14:55.557334-05:00",
"start_day": 6,
"end_day": 6,
"start_time": "11:00:00",
"end_time": "23:00:00"
},
],
"modified": "2018-08-21T09:46:16.992915-04:00",
"name": "Argo Tea [Spring/Fall 2018]",
"valid_start": "2018-01-21T13:11:51-05:00",
"valid_end": "2018-12-16T13:12:13-05:00",
"twenty_four_hours": false,
"schedule_for_removal": true,
"promote_to_main": false
},
"special_schedules": [
{
"id": 243,
"open_times": [
{
"schedule": 243,
"modified": "2018-07-02T12:35:20.071358-04:00",
"start_day": 5,
"end_day": 5,
"start_time": "00:00:00",
"end_time": "00:00:00"
},
{
"schedule": 243,
"modified": "2018-07-02T12:35:20.072228-04:00",
"start_day": 5,
```

**It's just an API.**

**[api.srct.gmu.edu/whatsopen/v2/](https://api.srct.gmu.edu/whatsopen/v2/)**

**JSON isn't a user experience**

**Let's build a user experience**



# Components Needed

# Components Needed

Model

View

Controller

# Components Needed

HTTP

MVC

You

# Components Needed

Model

View

Controller

# Model



```
class WOPFacility: Object, MapContext, Mappable {
    @objc dynamic var slug = ""
    @objc dynamic var facilityName = ""
    @objc dynamic var facilityLocation: WOPLocations? = WOPLocations()
    @objc dynamic var category: WOPCategories? = WOPCategories()
    var facilityTags: List<WOPFacilityTag>? = List<WOPFacilityTag>()
    @objc dynamic var mainSchedule: WOPSchedule? = WOPSchedule()
    var specialSchedules: List<WOPSchedule>? = List<WOPSchedule>()
    var labels: List<WOPFacilityTag>? = List<WOPFacilityTag>()
    @objc dynamic var tapingoURL = ""
    @objc dynamic var note = ""
    @objc dynamic var logoURL = ""
```

```
{  
  "slug": "dunkin-donuts",  
  "facility_name": "Dunkin' Donuts",  
  "logo": "https://wopen-logos.s3.amazonaws.com/dining/  
dunkin-donuts.png",  
  "facility_location": {  
    "id": 7,  
    "created": "2017-08-26T20:52:59.576217-04:00",  
    "modified": "2018-01-21T13:51:26.751417-05:00",  
    "building": "Tidewater Hall",  
    "friendly_building": "",  
    "address": "4349B Chesapeake River Wy",  
    "campus_region": "fairfax",  
    "on_campus": true,  
    "coordinate_location": {  
      "type": "Point",  
      "coordinates": [  
        -77.30605422374416,  
        38.83191871206628  
      ]  
    }  
  },  
  "facility_category": {  
    "id": 4,  
    "created": "2017-08-26T20:49:23.468882-04:00",  
    "modified": "2017-08-26T21:04:56.781846-04:00",  
    "name": "Café"  
  },  
  "phone_number": "703-691-4090",  
  "facility_product_tags": [  
    "donuts",  
    "coffee",  
    "breakfast"  
  ],  
  "facility_labels": [  
    "Donuts"  
  ],  
  "facility_classifier": [  
    "s", "h", "o", "p", "m", "a", "s", "o", "n"  
  ],  
  "tapingo_url": "",  
  "note": "",  
  "main_schedule": {  
    "id": 324,  
    "name": "Main Schedule",  
    "start_time": "2017-08-26T12:00:00-04:00",  
    "end_time": "2017-08-26T23:59:59-04:00",  
    "days": ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"],  
    "times": [{"start": "07:00", "end": "08:00"}, {"start": "08:00", "end": "09:00"}, {"start": "09:00", "end": "10:00"}, {"start": "10:00", "end": "11:00"}, {"start": "11:00", "end": "12:00"}, {"start": "12:00", "end": "13:00"}, {"start": "13:00", "end": "14:00"}, {"start": "14:00", "end": "15:00"}, {"start": "15:00", "end": "16:00"}, {"start": "16:00", "end": "17:00"}, {"start": "17:00", "end": "18:00"}, {"start": "18:00", "end": "19:00"}, {"start": "19:00", "end": "20:00"}, {"start": "20:00", "end": "21:00"}, {"start": "21:00", "end": "22:00"}, {"start": "22:00", "end": "23:00"}, {"start": "23:00", "end": "00:00"}, {"start": "00:00", "end": "01:00"}, {"start": "01:00", "end": "02:00"}, {"start": "02:00", "end": "03:00"}, {"start": "03:00", "end": "04:00"}, {"start": "04:00", "end": "05:00"}, {"start": "05:00", "end": "06:00"}, {"start": "06:00", "end": "07:00"}],  
    "order": 1  
  }  
}
```

```
class WOPFacility: Object, MapContext, Mappable {  
  @objc dynamic var slug = ""  
  @objc dynamic var facilityName = ""  
  @objc dynamic var facilityLocation: WOPLocations? = WOPLocations()  
  @objc dynamic var category: WOPCategories? = WOPCategories()  
  var facilityTags: List<WOPFacilityTag>? = List<WOPFacilityTag>()  
  @objc dynamic var mainSchedule: WOPSchedule? = WOPSchedule()  
  var specialSchedules: List<WOPSchedule>? = List<WOPSchedule>()  
  var labels: List<WOPFacilityTag>? = List<WOPFacilityTag>()  
  @objc dynamic var tapingoURL = ""  
  @objc dynamic var note = ""  
  @objc dynamic var logoURL = ""  
  
  required convenience init?(map: Map) {  
    self.init()  
  }  
  func mapping(map: Map){  
    slug <- map["slug"]  
    facilityName <- map["facility_name"]  
    facilityLocation <- map["facility_location"]  
    category <- map["facility_category"]  
    facilityTags <- (map["facility_product_tags"], TagTransform())  
    mainSchedule <- map["main_schedule"]  
    specialSchedules <- (map["special_schedules"],  
    ListTransform<WOPSchedule>())  
    labels <- (map["facility_labels"], TagTransform())  
    tapingoURL <- map["tapingo_url"]  
    note <- map["note"]  
    logoURL <- map["logo"]  
  }  
}
```

```
{  
  "slug": "dunkin-donuts",  
  "facility_name": "Dunkin' Donuts",  
  "logo": "https://wopen-logos.s3.amazonaws.com/dining/  
dunkin-donuts/  
  "facility_location": {  
    "id": 7,  
    "created": "2017-08-26T20:52:59.576217-04:00",  
    "modified": "2018-01-21T13:51:26.751417-05:00",  
    "building": "Tidewater Hall",  
    "friendly_building": "",  
    "address": "4349B Chesapeake River Wy",  
    "campus_region": "fairfax",  
    "on_campus": true,  
    "coordinate_location": {  
      "type": "Point",  
      "coordinates": [  
        -77.30605422374416,  
        38.83191871206628  
      ]  
    }  
  },  
  "facility_category": {  
    "id": 4,  
    "created": "2017-08-26T20:49:23.468882-04:00",  
    "modified": "2017-08-26T21:04:56.781846-04:00",  
    "name": "Café"  
  },  
  "phone_number": "703-691-4090",  
  "facility_product_tags": [  
    "donuts",  
    "coffee",  
    "breakfast"  
  ],  
  "facility_labels": [  
    "Donuts"  
  ],  
  "facility_classifier": [  
    "s", "h", "o", "p", "m", "a", "s", "o", "n"  
  ],  
  "tapingo_url": "",  
  "note": "",  
  "main_schedule": {  
    "id": 324,  
    "start": "2017-08-26T12:00:00-04:00",  
    "end": "2017-08-26T13:00:00-04:00",  
    "days": "Monday",  
    "status": "Open",  
    "order": 1  
  },  
  "closed": false  
}
```

```
class WOPLocations: Object, Mappable {  
  @objc dynamic var id = 0  
  @objc dynamic var created = ""  
  @objc dynamic var lastmodified = ""  
  @objc dynamic var building = ""  
  @objc dynamic var address = ""  
  @objc dynamic var campus = ""  
  @objc dynamic var onCampus = false  
  @objc dynamic var abbreviation = ""  
  @objc dynamic var coordinates: WOPCoordinates? = WOPCoordinates()  
  
  required convenience init?(map: Map){  
    self.init()  
  }  
  
  func mapping(map: Map){  
    id <- map["id"]  
    created <- map["created"]  
    lastmodified <- map["modified"]  
    building <- map["building"]  
    address <- map["address"]  
    campus <- map["campus_region"]  
    onCampus <- map["on_campus"]  
    abbreviation <- map["friendly_building"]  
    coordinates <- map["coordinate_location"]  
  }  
}
```

# More Examples

```
class WOPSchedule: Object, Mappable {  
    @objc dynamic var id = 0  
    var openTimes = List<WOPOpenTimes>()  
    @objc dynamic var lastModified = ""  
    @objc dynamic var name = ""  
    @objc dynamic var validStart = ""  
    @objc dynamic var validEnd = ""  
    @objc dynamic var twentyFourHours = false
```

```
class WOPOpenTimes: Object, Mappable {  
    @objc dynamic var schedule = 0  
    @objc dynamic var lastModified = ""  
    @objc dynamic var startDay = 0  
    @objc dynamic var endDay = 0  
    @objc dynamic var startTime = ""  
    @objc dynamic var endTime = ""
```

**Further Reading on how we map JSON to Swift**  
<https://github.com/Hearst-DD/ObjectMapper>

# Storing the Model?



realm

[realm.io/products/realm-database](https://realm.io/products/realm-database)

```
let realm = try! Realm()

var f = WOPFacility()
f.facilityName = "Dunkin' Donuts"
// Set Facility Params . . .

try! realm.write {
    realm.add(model)
}
```

```
let realm = try! Realm()

let results = realm.objects(WOPFacilitiesModel.self)

if results.count > 0 {
    let model = results[0]
    let facilities = model.facilities

    facilitiesArray = facilities.filter({ (facility: WOPFacility) -> Bool in
        return WOPUtilities.isFavoriteFacility(facility)
    })
}
```

# Model

# Components Needed

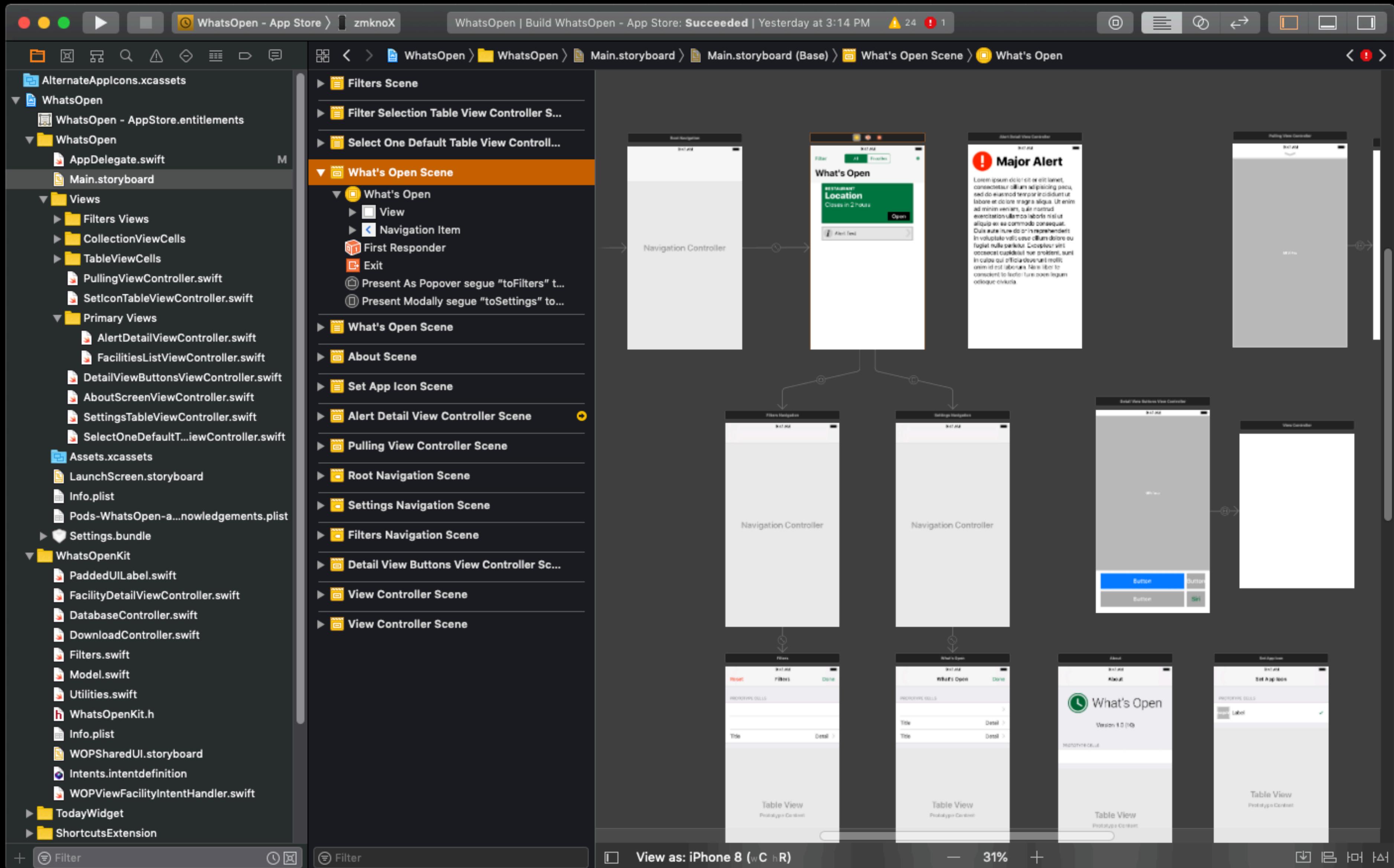
Model

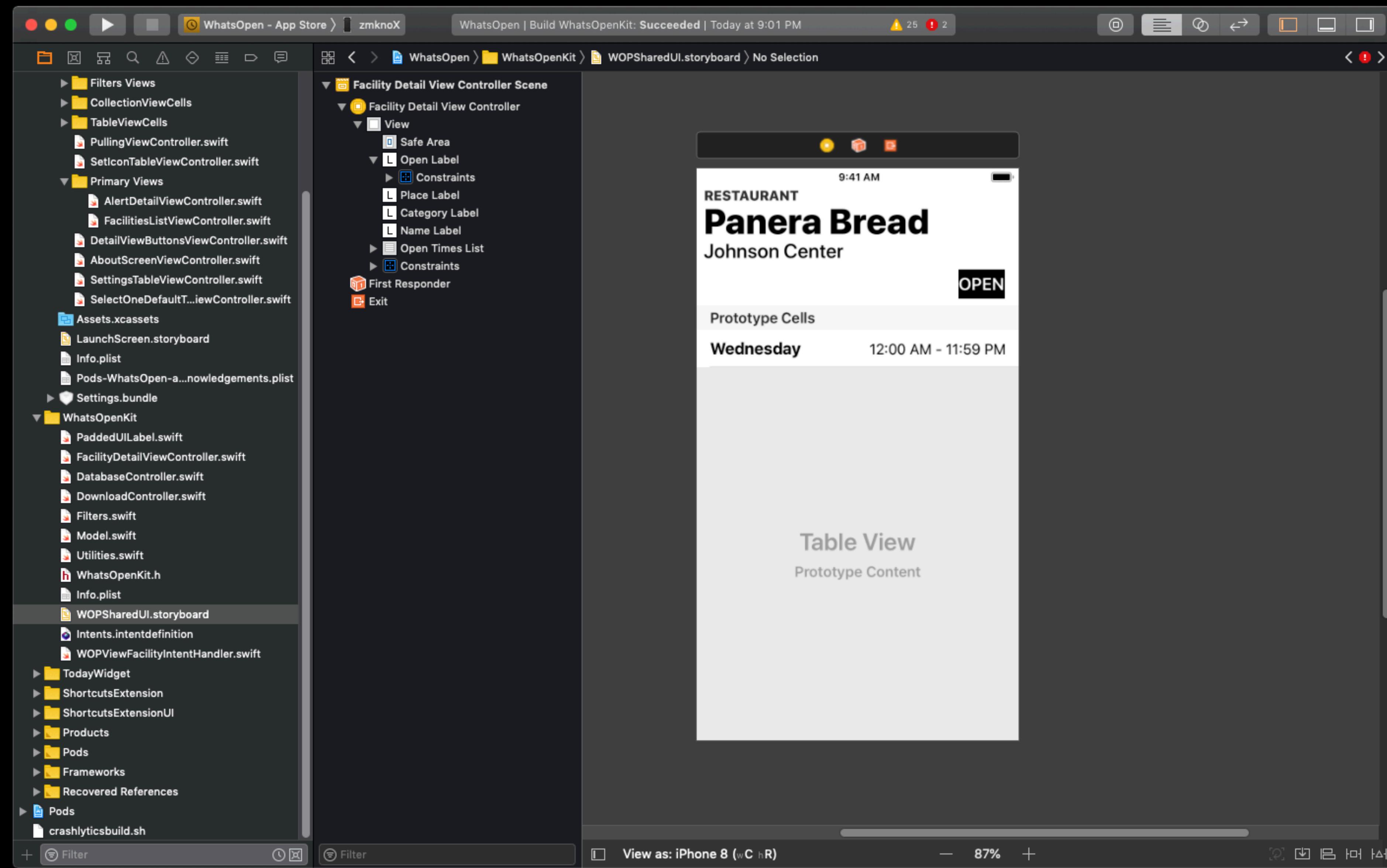
View

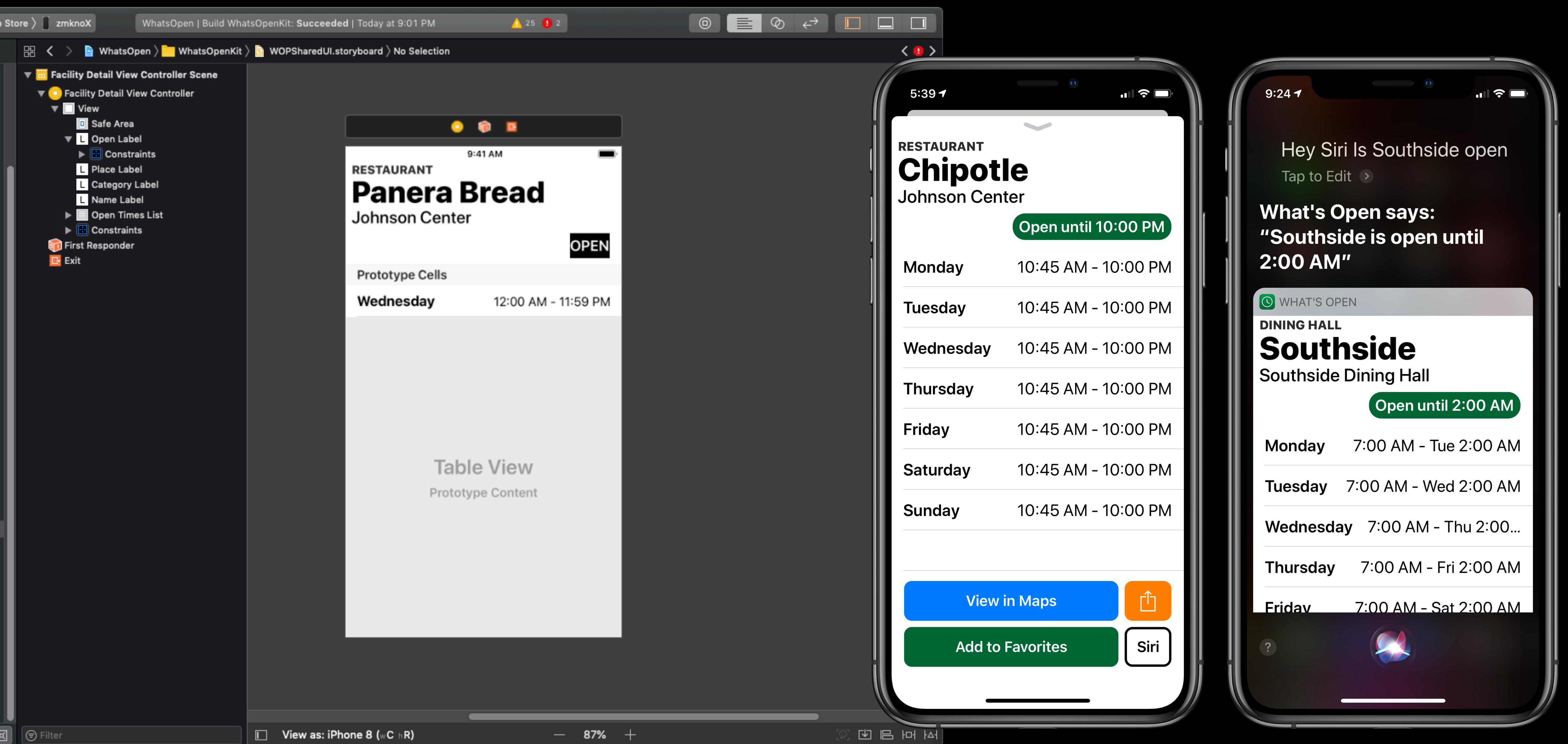
Controller

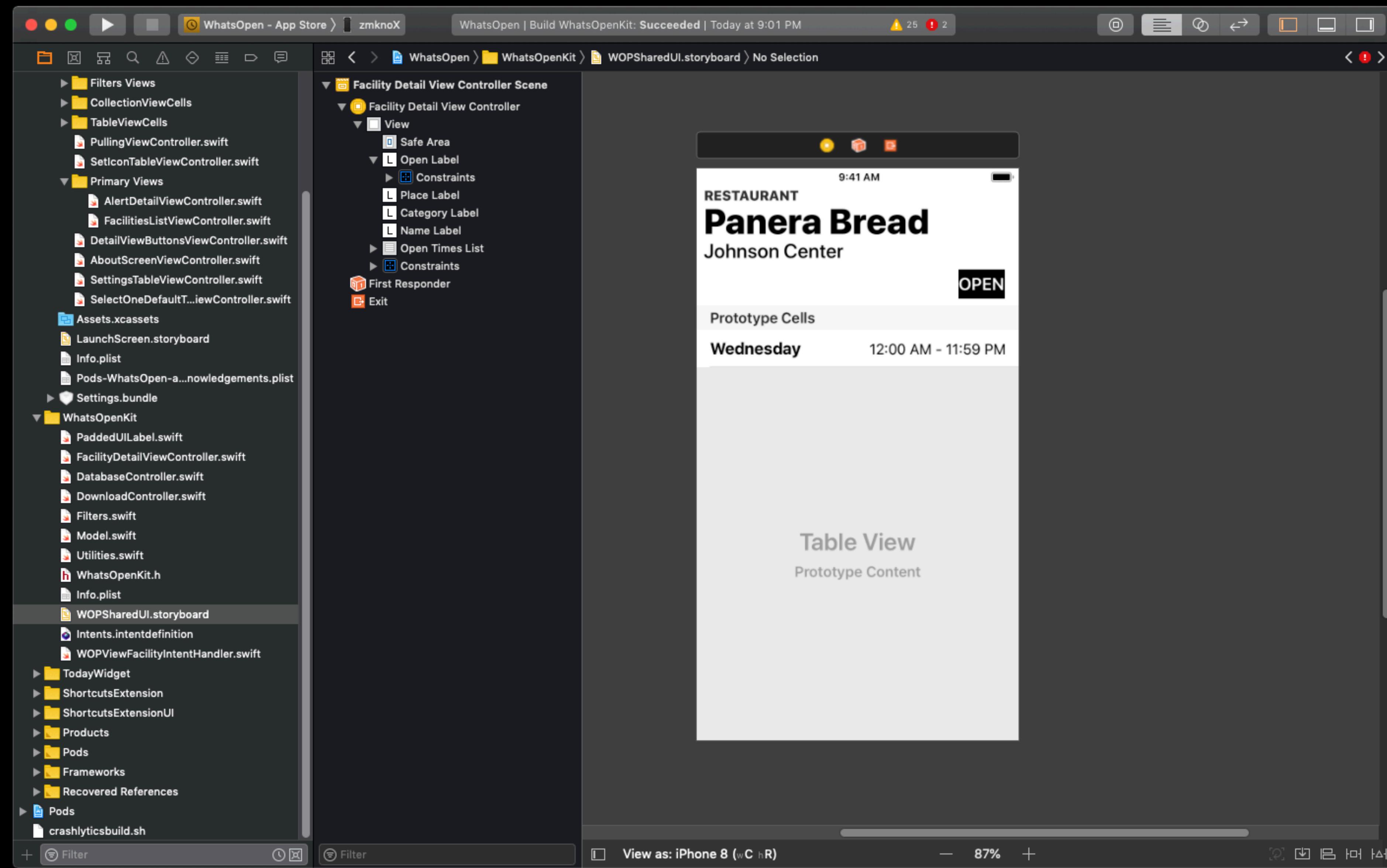
# View

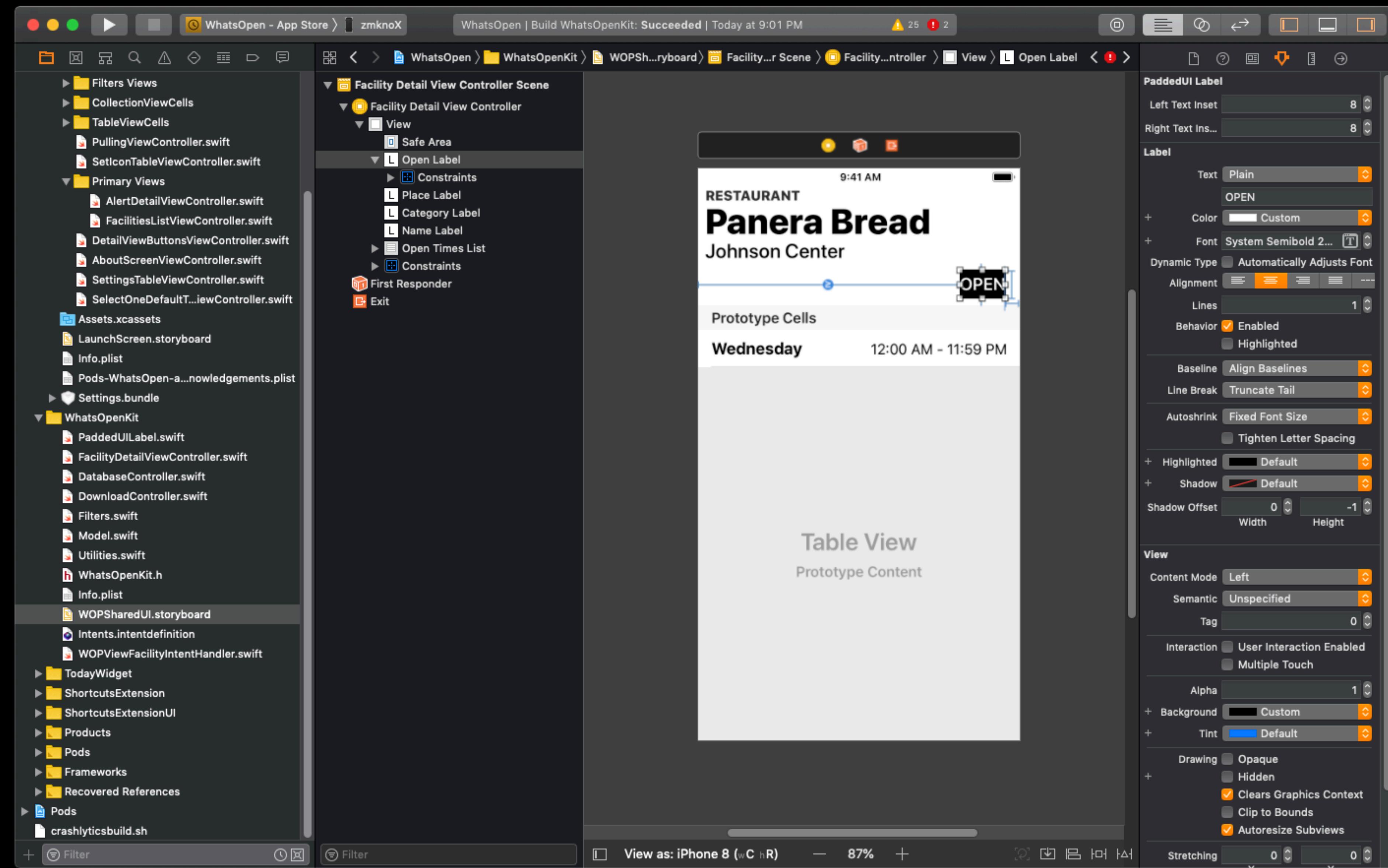
# Storyboards

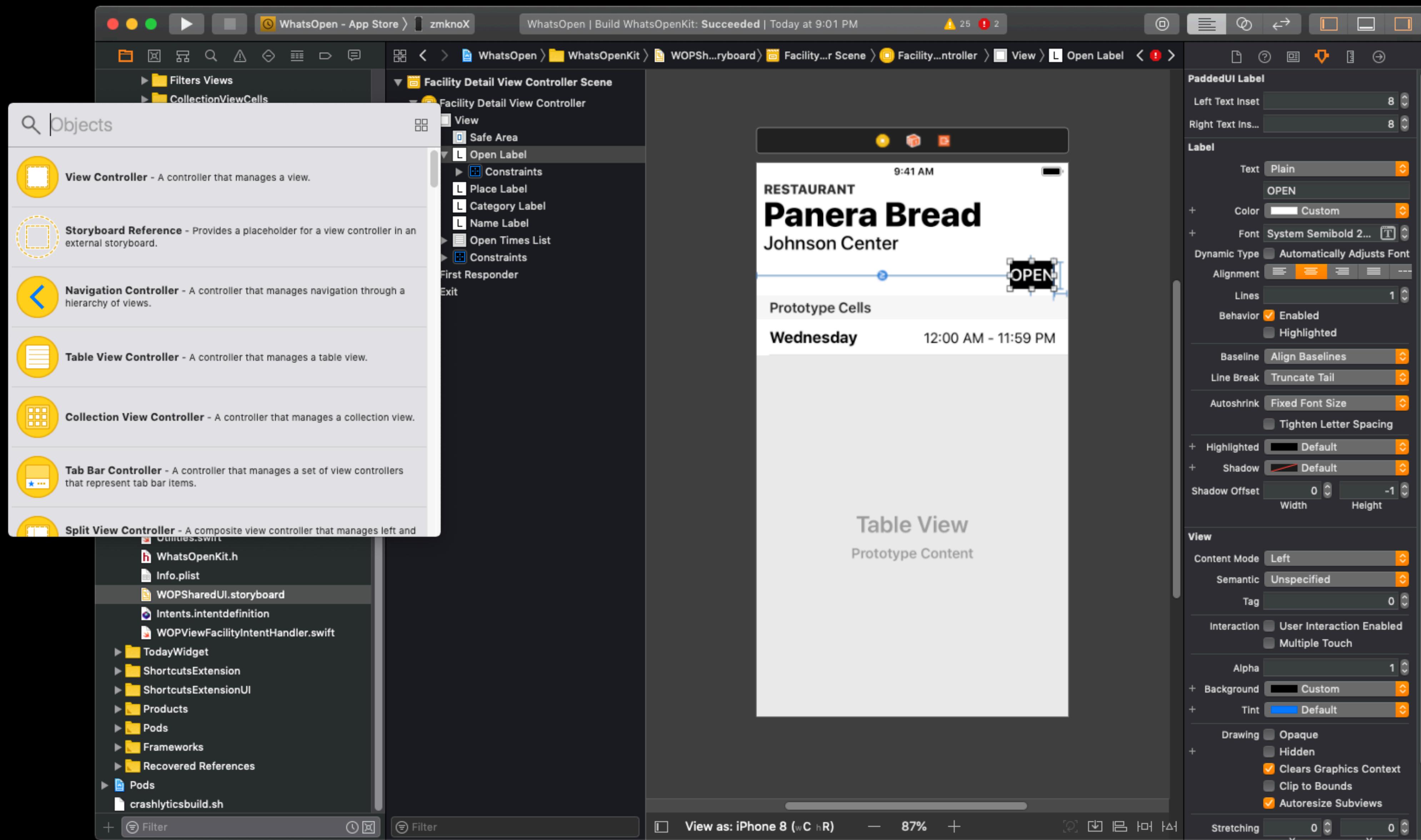


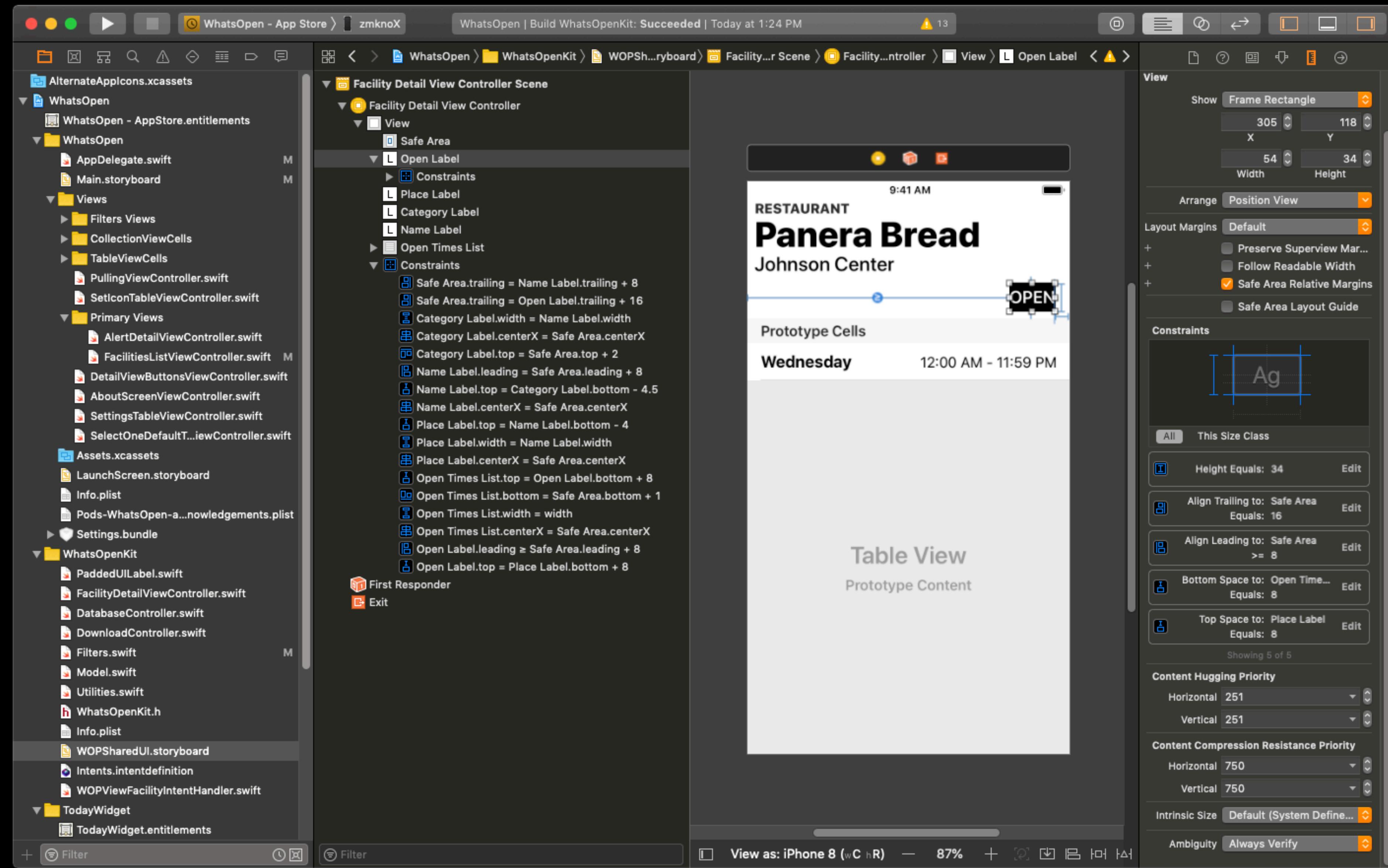


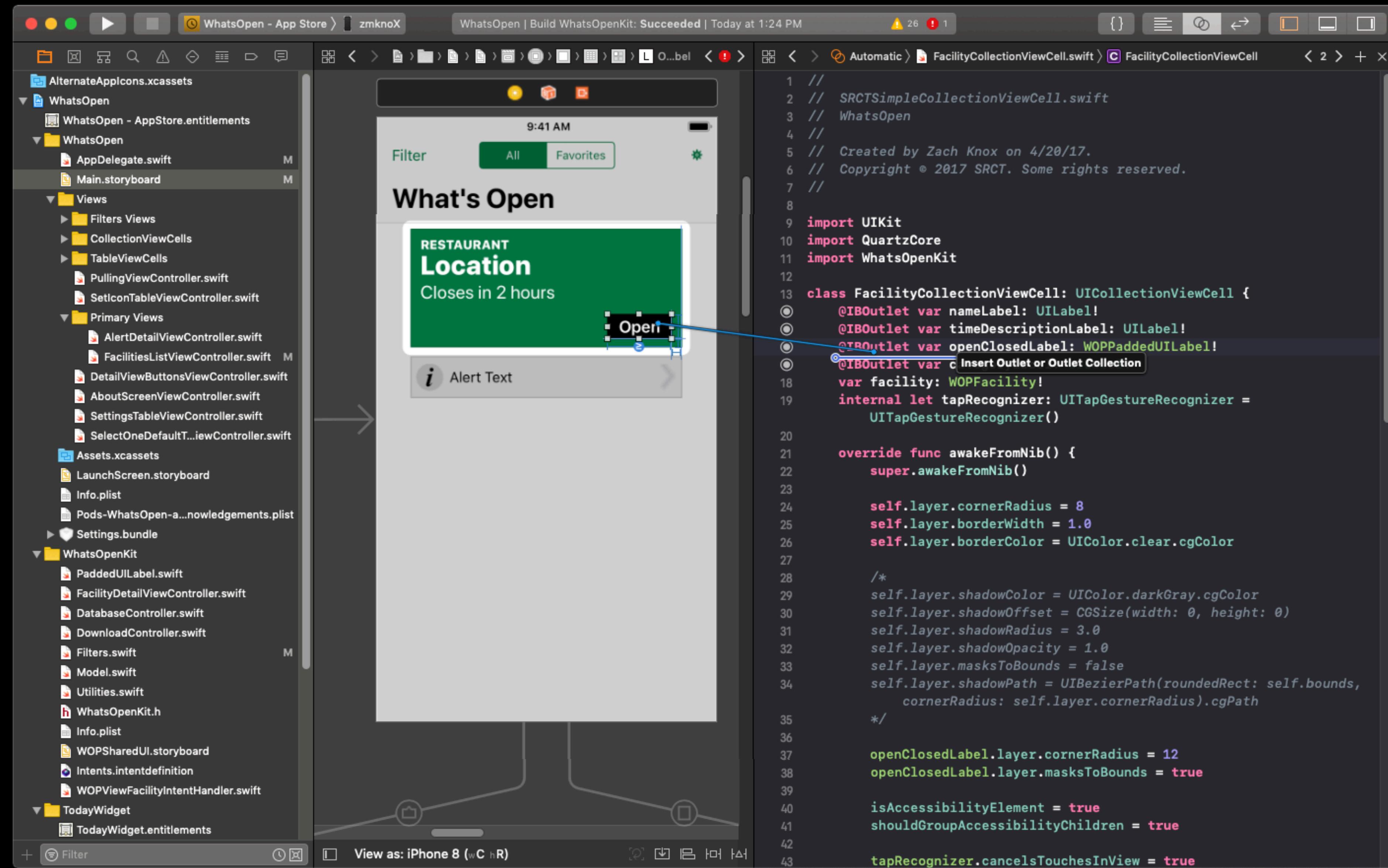












**Note: You can do all of this in code**

# View

# Components Needed

Model

View

Controller

# Controller

# Utilities

# View Controllers

# Utilities

## View Controllers

**“Business Logic”**

# Time

**Is Facility A open today?**

**Is Facility A open now?**

**When does Facility A close?**

**When will Facility A be open next?**

```
static func isOpen(facility: WOPFacility) -> Bool {  
    var open = false  
    let current = getCurrentSchedule(facility)  
  
    if current!.twentyFourHours {  
        return true  
    }  
  
    if !(current!.openTimes.isEmpty) {  
        if today(facility: facility) != nil {  
            open = time(facility)  
        }  
    } else {  
        open = false  
    }  
  
    return open  
}
```

```
static func today(facility: WOPFacility, special: Bool = false) -> WOPOpenTimes? {
    let scheduleOpenTimes = getCurrentSchedule(facility)?.openTimes

    let currentDay = getCurrentDayOfWeek()
    for openTime in scheduleOpenTimes! {
        if(currentDay! >= openTime.startDay && currentDay! <= openTime.endDay) {
            return openTime
        }
    }
    return nil
}
```

```
static func time(_ facility: WOPFacility) -> Bool {
    let nowTime          = getCurrentTime()
    guard let startEnd = getStartEndDates(facility) else { return false }
    let startTime        = startEnd.startTime
    var endTime          = startEnd.endTime

    if endTime < startTime {
        endTime = Date.endOfDay()
        if nowTime > Date.startOfDay() && nowTime < startEnd.endTime {
            endTime = startEnd.endTime
        }
    }

    // Check if the current time is
    // In between the start & end time of the facility
    return (nowTime >= startTime) && nowTime <= (endTime)
}
```

**Time is hard. Please fix our bugs.**

**Time is hard. Please fix our bugs.**

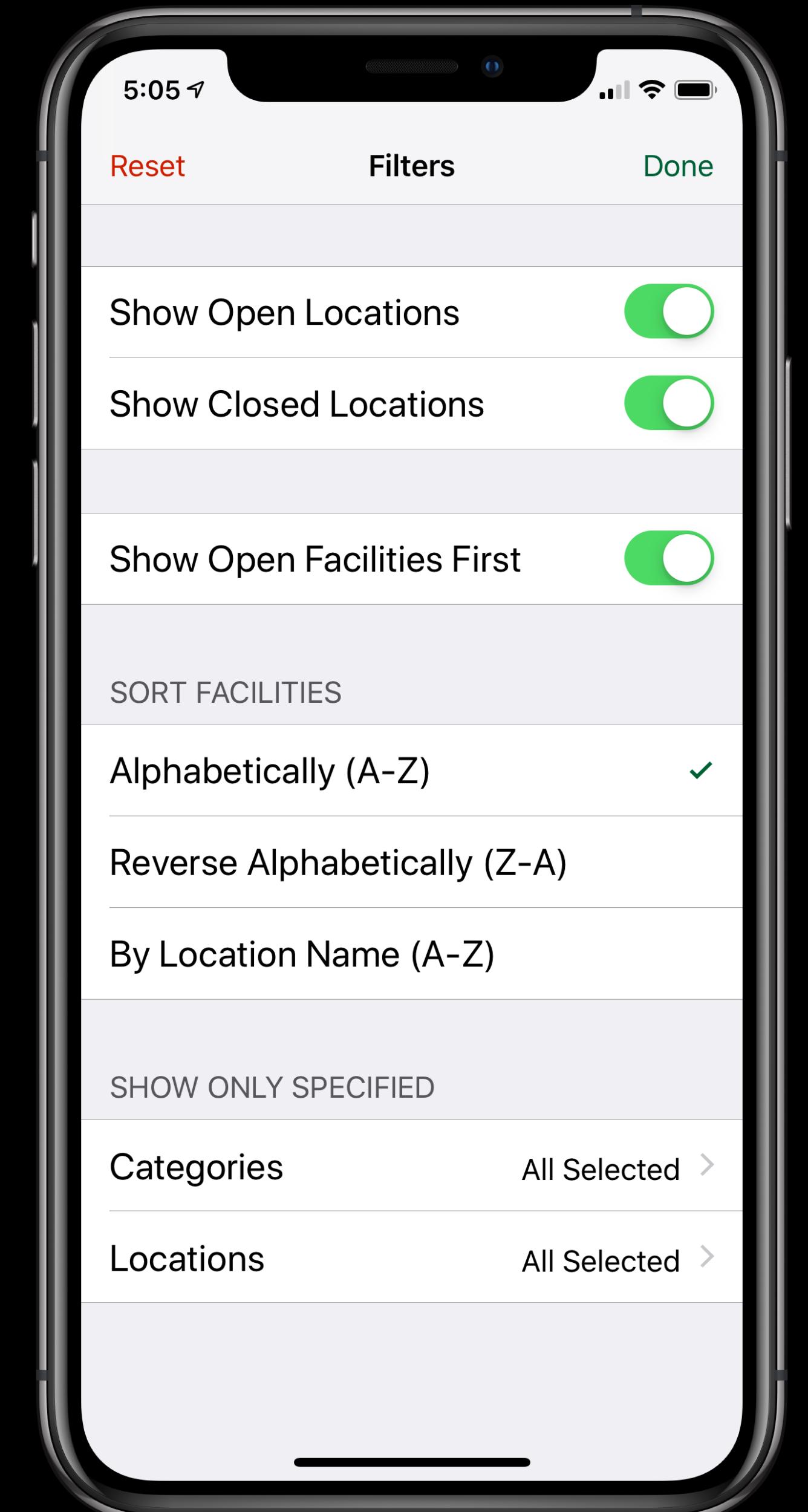
[go.gmu.edu/pleasefixourtimebugs](http://go.gmu.edu/pleasefixourtimebugs)

# Filters

Show me only facilities in Fairfax

Show me only Dining Halls

**Sort facilities by location name**



```
class WOPFilters {
    var showOpen = true
    var showClosed = true
    var sortBy = WOPSortMethod.alphabetical
    var openFirst = true

    var showAlerts = ["Informational":true, "Minor Alerts":true, "Major Alerts":true]

    var onlyFromLocations = [String: Bool]()
    var onlyFromCategories = [String: Bool]()
    var onlyFromCampuses = [String: Bool]()

    // . . .

}
```

```
class WOPFilters {
    var showOpen = true
    var showClosed = true
    var sortBy = WOPSortMethod.alphabetical
    var openFirst = true

    var showAlerts = ["Informational":true, "Minor Alerts":true, "Major Alerts":true]

    var onlyFromLocations = [String: Bool]()
    var onlyFromCategories = [String: Bool]()
    var onlyFromCampuses = [String: Bool]()

    // . . .

}

func filterByLocation(_ facilities: [WOPFacility], filters: WOPFilters) -> [WOPFacility] {
    return facilities.filter {
        filters.onlyFromLocations[$0.facilityLocation?.building!.lowercased()] ?? false
    }
}
```

\*Not actually implemented like this yet

# Utilities

## View Controllers

# Utilities

# View Controllers

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure with files like `AlternateAppIcons.xcassets`, `WhatsOpen - AppStore.entitlements`, `AppDelegate.swift`, `Main.storyboard`, `Views` folder containing `Filters Views`, `CollectionViewCells`, `TableViewCellCells`, `PullingViewController.swift`, `SetIconTableViewController.swift`, `Primary Views` folder containing `AlertDetailViewController.swift` and `FacilitiesListViewController.swift` (selected), `DetailViewControllerButtonsViewController.swift`, `AboutScreenViewController.swift`, `SettingsTableViewController.swift`, `SelectOneDefaultTableViewCellController.swift`, `Assets.xcassets`, `LaunchScreen.storyboard`, `Info.plist`, `Pods-WhatsOpen-a...nowledgements.plist`, `Settings.bundle`, and `WhatsOpenKit` folder containing `PaddedUILabel.swift`, `FacilityDetailViewController.swift`, `DatabaseController.swift`, `DownloadController.swift`, `Filters.swift` (marked as modified), `Model.swift`, `Utilities.swift`, `WhatsOpenKit.h`, `Info.plist`, `WOPSharedUI.storyboard`, `Intents.intentdefinition`, and `WOPViewFacilityIntentHandler.swift`.
- Editor:** Displays the source code for `FacilitiesListViewController.swift`. The code defines a class `FacilitiesListViewController` that implements `UIViewController`, `UICollectionViewDelegate`, `UICollectionViewDataSource`, `UIViewControllerPreviewingDelegate`, and `UICollectionViewDelegateFlowLayout`. It manages facilities and alerts lists, applies filters, and handles search controllers.
- Build Bar:** Shows the build status as "Succeeded" with the message "Build WhatsOpenKit: Succeeded | Today at 1:24 PM".
- Top Bar:** Shows the project name "WhatsOpen - App Store" and the file path "WhatsOpen > WhatsOpen > Views > Primary Views > FacilitiesListViewController.swift".

The screenshot shows the Xcode interface with the project navigation bar at the top. The left sidebar displays the project structure:

- AlternateAppIcons.xcassets
- WhatsOpen

  - WhatsOpen - AppStore.entitlements
  - WhatsOpen

    - AppDelegate.swift
    - Main.storyboard
    - Views
      - Filters Views
      - CollectionViewCells
      - TableViewCells
      - PullingViewController.swift
      - SetIconTableViewController.swift
    - Primary Views
      - AlertDetailViewController.swift
      - FacilitiesListViewController.swift

  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
  - Pods-WhatsOpen-a...nowledgements.plist
  - Settings.bundle
  - WhatsOpenKit

    - PaddedUILabel.swift
    - FacilityDetailViewController.swift
    - DatabaseController.swift
    - DownloadController.swift
    - Filters.swift
    - Model.swift
    - Utilities.swift
    - WhatsOpenKit.h
    - Info.plist
    - WOPSharedUI.storyboard
    - Intents.intentdefinition
    - WOPViewFacilityIntentHandler.swift

  - TodayWidget

    - TodayWidget.entitlements

go.gmu.edu/WOViewController

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure with files like `AlternateAppIcons.xcassets`, `WhatsOpen - AppStore.entitlements`, `AppDelegate.swift`, `Main.storyboard`, `Views` folder containing `Filters Views`, `CollectionViewCells`, `TableViewCellCells`, `PullingViewController.swift`, `SetIconTableViewController.swift`, `Primary Views` folder containing `AlertDetailViewController.swift`, `FacilitiesListViewController.swift`, `DetailViewButtonsViewController.swift`, `AboutScreenViewController.swift`, `SettingsTableViewController.swift`, `SelectOneDefaultT...iewController.swift`, `Assets.xcassets`, `LaunchScreen.storyboard`, `Info.plist`, `Pods-WhatsOpen-a...nowledgements.plist`, `Settings.bundle`, `WhatsOpenKit` folder containing `PaddedUILabel.swift`, `FacilityDetailViewController.swift`, `DatabaseController.swift`, `DownloadController.swift`, `Filters.swift`, `Model.swift`, `Utilities.swift`, `WhatsOpenKit.h`, `Info.plist`, `WOPSharedUI.storyboard`, `Intents.intentdefinition`, and `WOPViewFacilityIntentHandler.swift`, and `TodayWidget` folder containing `TodayWidget.entitlements`.
- Editor:** Displays the code for `FacilitiesListViewController.swift`. The code handles tap gestures on collection view cells to present detail views for facilities or alerts.
- Build Bar:** Shows "WhatsOpen | Build WhatsOpenKit: Succeeded | Today at 1:24 PM".
- Top Bar:** Shows the current file path: `WhatsOpen > WhatsOpen > Views > Primary Views > FacilitiesListViewController.swift` and the build status: "12".

```
138     LeftButton.title = "Filter (On)"
139 }
140
141 override func viewWillAppear(_ animated: Bool) {
142     checkFilterState()
143     reloadWithFilters()
144 }
145
146 @objc func tapRecognizer(_ sender: UITapGestureRecognizer) {
147
148     let tapLocation = sender.location(in: LocationsList)
149     let indexPath = LocationsListindexPathForItem(at: tapLocation)
150
151     if(indexPath != nil) {
152         if(indexPath?.section == 1 || currentAlerts.count == 0) {
153             let storyboard = UIStoryboard(name: "WOPSharedUI", bundle: Bundle(for: WOPFacilityDetailViewController.self))
154             let destination = storyboard.instantiateViewController(withIdentifier: "detailView") as?
155                 WOPFacilityDetailViewController
156             let tapped = self.LocationsList.cellForItem(at: indexPath!) as! FacilityCollectionViewCell
157             destination?.facility = tapped.facility
158             self.presentDetailView(destination!, tapped: tapped)
159         } else {
160             let destination = self.storyboard?.instantiateViewController(withIdentifier: "alertDetail") as?
161                 AlertDetailViewController
162             let tapped = self.LocationsList.cellForItem(at: indexPath!) as! AlertCollectionViewCell
163             destination?.alert = tapped.alert
164             self.presentDetailView(destination!, tapped: tapped)
165         }
166     }
167
168     var goodToGo = false
169     @objc func toDetailFromSearch(_ notification: Notification) {
170         func toDetailCompletion() {
171             let storyboard = UIStoryboard(name: "WOPSharedUI", bundle: Bundle(for: WOPFacilityDetailViewController.self))
172             let dest = storyboard.instantiateViewController(withIdentifier: "detailView") as! WOPFacilityDetailViewController
173             let userActivity = notification.object as? NSUserActivity
174             if(userActivity == nil) {
175                 return // don't do anything
176             }
177         }
178         let facility = realm.objects(WOPFacilitiesModel.self)[0].facilities.filter(NSPredicate(format: "facilityName = '" +
179             (userActivity?.title)! + "'")).first
180         if(facility == nil) {
```

WhatsOpen - App Store > zmknoX WhatsOpen | Build WhatsOpenKit: Succeeded | Today at 1:24 PM 12

WhatsOpen > WhatsOpen > Views > Primary Views > FacilitiesListViewController.swift > filterFacilitiesForSearchText(\_:) < ▲ >

```
416
417    /**
418     Filters facilities based on the text inputted into the search controller.
419
420     - parameters:
421         - searchText: text used to filter the facilities.
422     - returns:
423         List of filtered facilities. Facilities whose names, buildings, or categories match the search text are included.
424     */
425    func filterFacilitiesForSearchText(_ searchText: String) -> List<WOPFacility> {
426        var filtered: List<WOPFacility>
427
428        if searchText == "" {
429            filtered = shownFacilities
430            LocationsList.reloadData()
431            return shownFacilities
432        }
433        filtered = filteredFacilities.filter({(facility: WOPFacility) -> Bool in
434            let hasName = facility.facilityName.lowercased().contains(searchText.lowercased())
435            let hasBuilding = facility.facilityLocation?.building.lowercased().contains(searchText.lowercased()) ?? false
436            let hasAbbreviation = facility.facilityLocation?.abbreviation.lowercased().contains(searchText.lowercased()) ??
437                false
438            let hasCategory = facility.category?.categoryName.lowercased().contains(searchText.lowercased()) ?? false
439            var hasTag = false
440            for tag in facility.facilityTags! {
441                if hasTag {
442                    break
443                }
444                if tag.tag.lowercased().contains(searchText.lowercased()) {
445                    hasTag = true
446                }
447            }
448            var hasLabel = false
449            for label in facility.labels! {
450                if hasLabel {
451                    break
452                }
453                if label.tag.lowercased().contains(searchText.lowercased()) {
454                    hasLabel = true
455                }
456            }
457            return hasName || hasBuilding || hasCategory || hasTag || hasAbbreviation
458        })
459        LocationsList.reloadData()
```

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** On the left, it lists the project structure. Key files include `AlternateAppIcons.xcassets`, `WhatsOpen - AppStore.entitlements`, `AppDelegate.swift`, `Main.storyboard`, `Views` folder containing `Filters Views`, `CollectionViewCells`, `TableViewCellCells`, `PullingViewController.swift`, `SetIconTableViewController.swift`, `Primary Views` folder containing `AlertDetailViewController.swift`, `FacilitiesListViewController.swift` (selected), `DetailViewButtonsViewController.swift`, `AboutScreenViewController.swift`, `SettingsTableViewController.swift`, `SelectOneDefaultT...iewController.swift`, `Assets.xcassets`, `LaunchScreen.storyboard`, `Info.plist`, `Pods-WhatsOpen-a...nowledgements.plist`, `Settings.bundle`, `WhatsOpenKit` folder containing `PaddedUILabel.swift`, `FacilityDetailViewController.swift`, `DatabaseController.swift`, `DownloadController.swift`, `Filters.swift`, `Model.swift`, `Utilities.swift`, `WhatsOpenKit.h`, `Info.plist`, `WOPSharedUI.storyboard`, `Intents.intentdefinition`, and `WOPViewFacilityIntentHandler.swift`, and `TodayWidget` folder containing `TodayWidget.entitlements`.
- Editor:** The main editor area displays the `FacilitiesListViewController.swift` file. The code is as follows:

```
529
530     /*
531      * Attempts to update facilitiesArray from the network
532      * and place that new information into Realm
533     */
534    func update(_ sender: Any, completion: (() -> ())? = nil) {
535        WOPDownloadController.performDownload { (facilities) in
536            if(facilities == nil) {
537                DispatchQueue.main.async {
538                    let results = self.realm.objects(WOPFacilitiesModel.self)
539                    if results.count > 0 {
540                        let model = results[0]
541                        let facilitiesFromDB = model.facilities
542                        let lastUpdated = model.lastUpdated
543
544                        self.facilitiesArray = facilitiesFromDB
545                        self.updateFiltersLists()
546                        self.reloadWithFilters()
547                        self.refreshControl.attributedTitle = NSAttributedString(string: "Last Updated: " +
548                            self.shortDateFormat(lastUpdated))
549                        self.refreshControl.endRefreshing()
550                        self.goodToGo = true
551                        if(completion != nil) {
552                            completion!()
553                        }
554                    } else {
555                        self.facilitiesArray = List<WOPFacility>()
556                    }
557                }
558            }
559        } else {
560            self.facilitiesArray = facilities!
561
562            DispatchQueue.main.async {
563                let date = Date()
564                self.refreshControl.attributedTitle = NSAttributedString(string: "Last Updated: " +
565                    self.shortDateFormat(date))
566
567                let results = self.realm.objects(WOPFacilitiesModel.self)
568                if results.count == 0 {
569                    let model = WOPFacilitiesModel()
570                    for f in facilities! {
571                        model.facilities.append(f)
572                    }
573                }
574            }
575        }
576    }
577}
```



The screenshot shows the Xcode interface with the following details:

- Project Navigator:** On the left, it lists the project structure. Key files include `AlternateAppIcons.xcassets`, `WhatsOpen - AppStore.entitlements`, `AppDelegate.swift`, `Main.storyboard`, `Views` folder containing `Filters Views`, `CollectionViewCells`, `TableViewCellCells`, `PullingViewController.swift`, `SetIconTableViewController.swift`, `Primary Views` folder containing `AlertDetailViewController.swift` and `FacilitiesListViewController.swift` (selected), `DetailViewButtonsViewController.swift`, `AboutScreenViewController.swift`, `SettingsTableViewController.swift`, `SelectOneDefaultT...iewController.swift`, `Assets.xcassets`, `LaunchScreen.storyboard`, `Info.plist`, `Pods-WhatsOpen-a...nowledgements.plist`, `Settings.bundle`, `WhatsOpenKit` folder containing `PaddedUILabel.swift`, `FacilityDetailViewController.swift`, `DatabaseController.swift`, `DownloadController.swift`, `Filters.swift`, `Model.swift`, `Utilities.swift`, `WhatsOpenKit.h`, `Info.plist`, `WOPSharedUI.storyboard`, `Intents.intentdefinition`, and `WOPViewFacilityIntentHandler.swift`, `TodayWidget` folder containing `TodayWidget.entitlements`.
- Editor:** The main editor area displays the `FacilitiesListViewController.swift` file. The code implements a collection view layout for different screen widths, calculating item width based on safe area insets and window width.
- Build Bar:** At the top, it shows "WhatsOpen - App Store" and "zmknoX". The status bar indicates "WhatsOpen | Build WhatsOpenKit: Succeeded | Today at 1:24 PM".
- Top Bar:** Standard Xcode navigation icons for file, edit, search, and zoom.

```
774 // MARK - Collection View Cell Layout
775 func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize {
776     if(indexPath.section == 1 || currentAlerts.count == 0) {
777         let height = LocationsListLayout.itemSize.height
778         let width: CGFloat
779
780         let edgeInsets = self.view.safeAreaInsets
781         let windowHeight = self.view.frame.size.width - edgeInsets.left - edgeInsets.right
782
783         if(windowWidth < 640) {
784             width = windowHeight - 20
785         }
786         else if(windowWidth >= 640 && windowHeight < 1024) {
787             width = (windowWidth / 2) - 15
788         }
789         else if(windowWidth >= 1024) {
790             width = (windowWidth / 3) - 15
791         }
792         else {
793             width = windowHeight - 20
794         }
795
796         return CGSize(width: width, height: height)
797     }
798     else {
799         let edgeInsets = self.view.safeAreaInsets
800         return CGSize(width: self.view.frame.size.width - edgeInsets.left - edgeInsets.right, height: 43)
801     }
802 }
803
804 func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, minimumLineSpacingForSectionAt section: Int) -> CGFloat {
805     if(section == 1 || currentAlerts.count == 0) {
806         return 15
807     }
808     else {
809         return 0
810     }
811 }
812
813 func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, minimumInteritemSpacingForSectionAt section: Int) -> CGFloat {
814     if(section == 1 || currentAlerts.count == 0) {
815         return 10
816     }
817 }
```

# Utilities

# View Controllers

**Try to put more in Utilities,  
less in View Controllers**

I should take my own advice...

# Controller

# Components Needed

Model

View

Controller



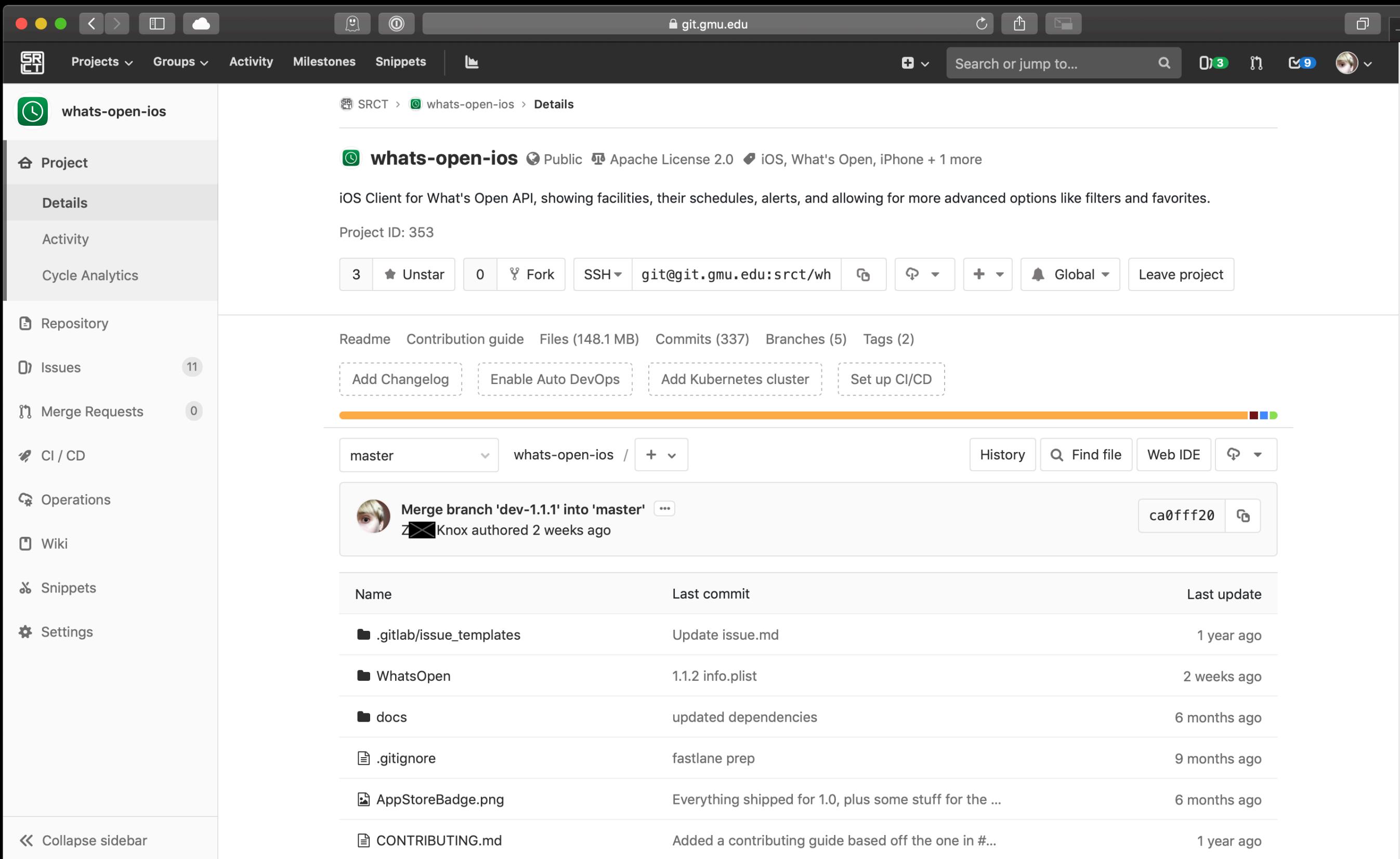




# What's Open for iPhone

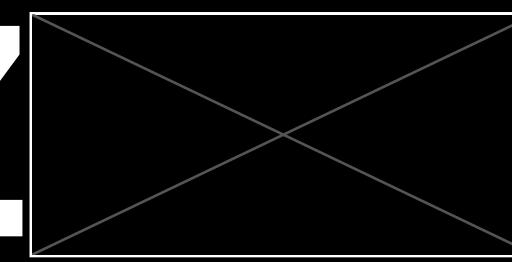


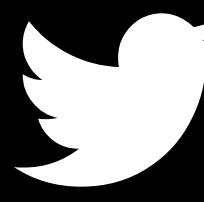
# You can contribute!



[git.gmu.edu/srct/whats-open-ios](https://git.gmu.edu/srct/whats-open-ios)

# Thank You!

Z  Knox

#  @zmknox