ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Semestrálna práca z predmetu Vývoj aplikácií pre mobilné zariadenia

Michal Červenec
5ZYI22
2021/2022

Moja aplikácia

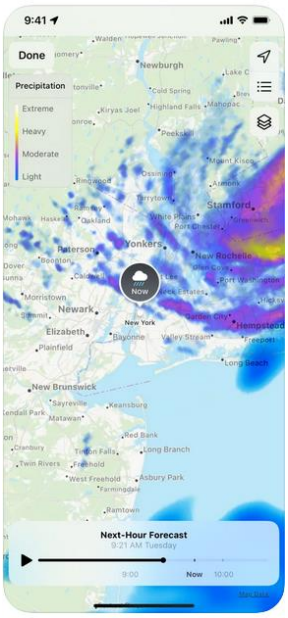
Na základe polohy používateľa dokáže so servera získať dáta s predpoveďou počasia pre danú lokalitu. Aplikácia zobrazuje miesto na ktorom sa používateľ nachádza s aktuálnou teplotou a stavom počasia. Tiež predpoveď na niekoľko najbližších dni. Zobrazuje sa aj mapa na ktorej momentálne je možné len prehliadať si ju. Každé miesto sa dá uložiť medzi obľúbené a používateľ by sa k nemu následne mal vedieť dostať cez záložku obľúbené, bez potreby hľadania na mape. Aplikácia má zabudovanú funkciu na hľadanie, ktorá podľa výrazu zadaného používateľom vyhľadá lokality s daným názvom.
Aplikácia by sa po spustení otvorí na ploche s predpoveďou pre danú lokalitu používateľa. Medzi jednotlivými plochami sa dá prechádzať pomocou ikon v spodnom paneli. Po kliknutí na ikonu sa otvorí mapa/aktuálna poloha/záložka s obľúbenými lokalitami/záložka s citátmi o počasí. Na ploche aktuálnej lokality (alebo každej ďalšej vyhľadanej) je v hornej lište ikona lupy s funkciou vyhľadávania. Po kliknutí sa objaví vstupné textové pole a po zadaní vstupu sa vypíšu lokality s daným názvom (alebo žiadne ak neexistujú), na ktoré sa dá kliknúť pre zobrazenie počasia v danej lokalite. Každá lokalita má v hornej lište aj ikonku na pridanie medzi obľúbené.
Na záložke s citátmi je pomocou tlačidla možné vygenerovať iný citát. Tiež tlačidlo pre vytvorenie nového citátu.

Porovnanie s podobnými aplikáciami v App Store:

1. Weather

https://apps.apple.com/us/app/weather/id1069513131

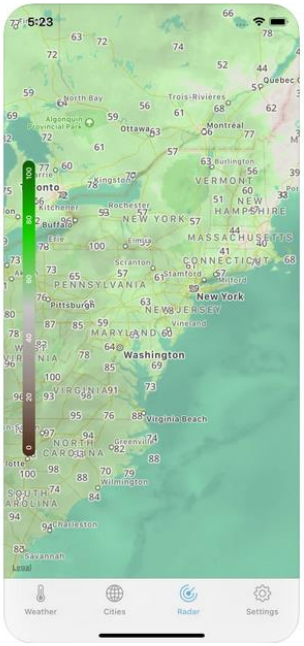
Originálna aplikácia priamo od Apple



Michal Červenec
5ZYI22
2021/2022

2. Weather ·

https://apps.apple.com/sk/app/weather/id1137406490?l=sk&platform=iphone

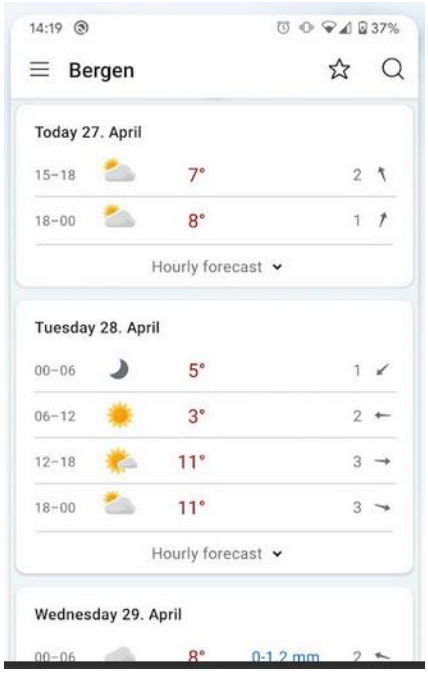


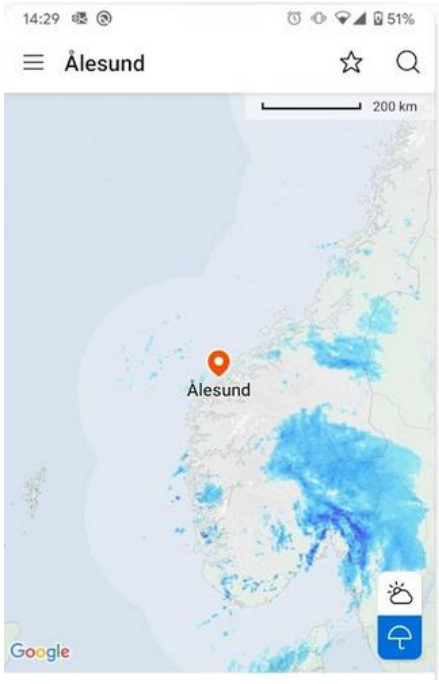
3. Yr

https://play.google.com/store/apps/details?id=no.nrk.yr

https://apps.apple.com/jo/app/yr-no/id490989206

Rovnaká aplikácia počasia na App Store aj Play Store.

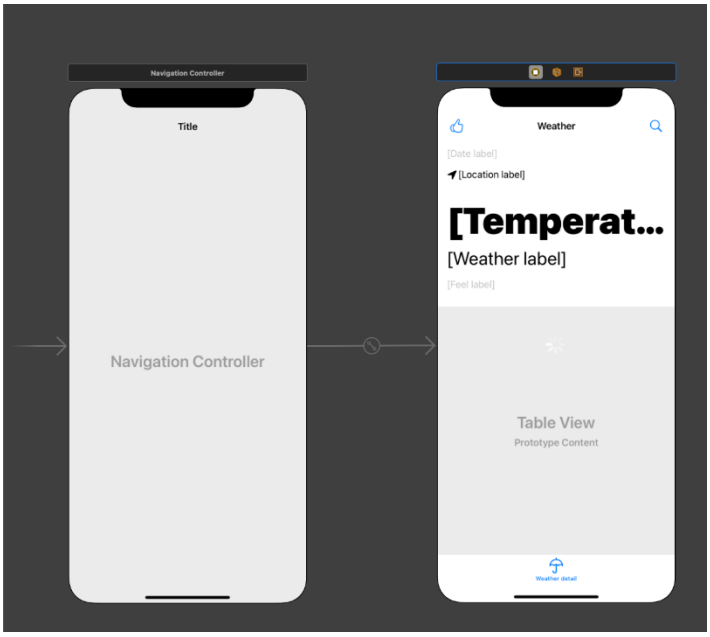


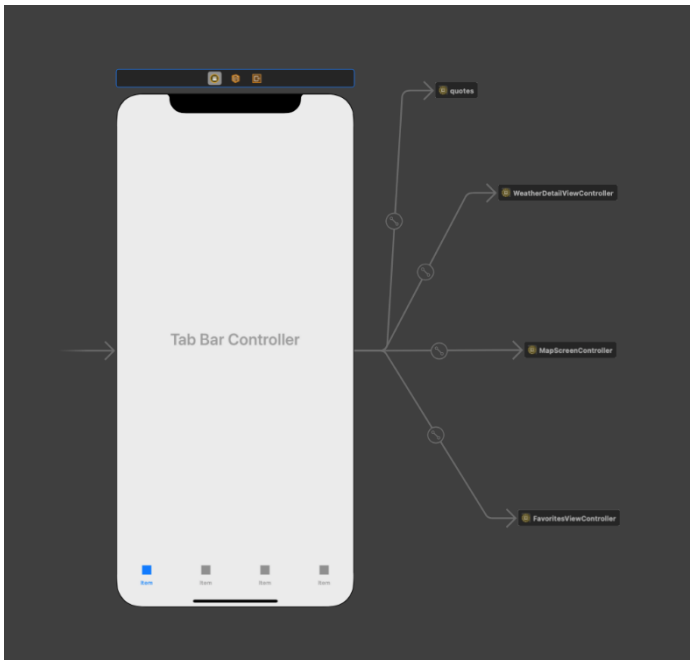


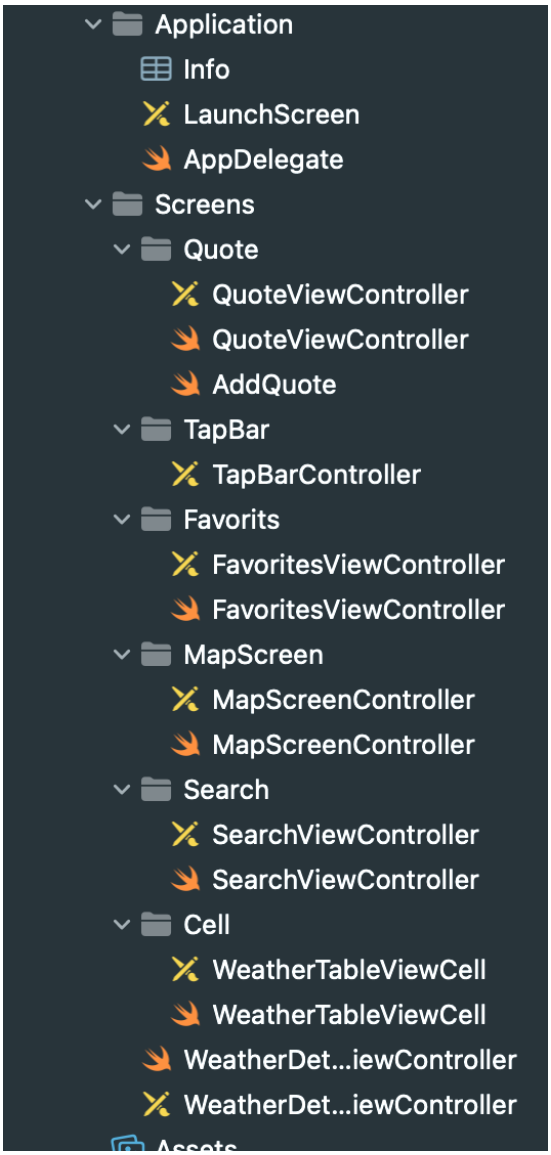
Michal Červenec
5ZYI22
2021/2022

Storyboards

Aplikácia bola programovaná v jazyku swift vyvíjanom pre tvorbu aplikácii pre platformu iOS. Na tvorbu aplikácie sme využívali storyboardy, starší spôsob stále sa využíva a dajú sa na ňom naučiť princípy. Využívajú sa na návrh a tvorbu view controllerov, na tvorbu prechodov medzi nimi a celkovú navigáciu v aplikácii. V rámci storyboardu sa využíva AutoLayout na ukladanie komponentov v rámci controllera. Pridávame potrebne constrainy, ktoré ukotvujú objekt vzhľadom k okrajom alebo iným objektom. Je potrebne ukotviť objekty správne pre zamedzenie straty usporiadania na iných veľkostiach obrazovky. Využité komponenty sú napríklad: UITableView, UILabel, UIImage, UIButton, ... Na navigáciu v rámci aplikácie využívam dva spôsoby: TabBar a NavigationController.







Michal Červenec
5ZYI22
2021/2022

Table View

Pre zobrazovanie počasia pre jednotlivé dni je potrebne, aby sa každý deň zobrazoval vo vlastnej časti. Na vyhnutia sa tvorbe samostatných komponentov pre každý deň je vhodne využiť TableView s možnosťou pridania recyklovateľnej bunky. TableView ma hlavičku v ktorej zobrazujeme stav v danom dni na danej lokalite (teplota, stav počasia, dátum, miesto, pocitová teplota). Jednotlivé bunky potom obsahujú predpoveď na nadchádzajúce dni. V hlavičke sú vložené komponenty Label s placeholder textom, do ktorých sa potom načítavajú dáta z API. Zo storyboardu je vytvorené prepojenie (outlet) do kódu kde je daný komponent definovaný ako premenná. Bunku ktorú ma tabuľka stačí vytvoriť raz a pomocou preťažených funkcii z UITableViewDataSource do nej posielam dáta. Bunka je vytvorená ako Xib a pracuje sa s ňou podobne ako so storyboardom v ktorom ju je možné navrhovať a upravovať. Ma placeholdery pre popisky a ikonu stavu počasia. Cela TableView sa nainicialuzuje volaním funkcie setupTableView().


```
@IBOutlet weak var tableView: UITableView!

@IBOutlet weak var activityIndicator: UIActivityIndicatorView!

@IBOutlet weak var dateLabel: UILabel!
@IBOutlet weak var locationLabel: UILabel!
@IBOutlet weak var temperatureLabel: UILabel!
@IBOutlet weak var weatherStatusLabel: UILabel!
@IBOutlet weak var feelsLikeLabel: UILabel!

@IBOutlet weak var emptyView: UIView!
@IBOutlet weak var errorMessageLabel: UILabel!
```

```
extension WeatherDetailViewController: UITableViewDataSource {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return days.count    }


    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        guard let weatherCell = tableView.dequeueReusableCell(withIdentifier: WeatherTableViewCell.classString, for: indexPath)
            as? WeatherTableViewCell
        else {
            return UITableViewCell()
        }

        weatherCell.setupCell(with: days[indexPath.row])

        return weatherCell
    }
}
```

Location

Aplikácia potrebuje pre svoje fungovanie prístup k polohe zariadenia. Toto zabezpečuje trieda LocationManager, ktorá využíva framework CoreLocation. LocationManager je vytvorený ako singleton, raz sa vytvorí a je možné ho používať počas behu aplikácie. Pomocou delegáta zisťujeme zmenu polohy a dekódujeme údaje o polohe, zo získaných koordinátov určí mesto a krajinu. Pri zisťovaní polohy je potrebné aby bolo ošetrený prípad ak nepríde nijaký vstup (napríklad ak je vypnuté určovanie polohy zariadenia). LocationManager v sebe tiež rieši problém s autorizáciu používateľa na prístup aplikácie k polohe zariadenia, aplikácia bez nej nemôže pracovať správne. Do plistu aplikácie je nutné uviesť prečo aplikácia vyžaduje dané povolenie (potrebné pre schválenie v prípade pridávania do AppStore).

```
    func locationManagerDidChangeAuthorization(_ manager: CLLocationManager) {
        switch manager.authorizationStatus {
        case .denied:
            authorizationCompletion?(false)
        case .authorizedWhenInUse, .authorizedAlways:
            authorizationCompletion?(true)
            print("authorized")
        case .notDetermined:
            print("not yet")
        case .restricted:
            print("restricted")
        @unknown default:
            fatalError()
        }
    }
}
```

```
extension LocationManager: CLLocationManagerDelegate {

    func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
        guard let location = locations.last else {
            return
        }

        geocoder.reverseGeocodeLocation(location) { [weak self] placemarks, error in
            guard let self = self else {return}

            guard let placemark = placemarks?.first, let city = placemark.locality, error == nil
                else {
                if let completion = self.completion {
                    completion(nil, error)
                }
                return
            }
            let currentLocation = CurrentLocation(city: city, coordinates: location.coordinate)
            self.completion?(currentLocation, nil)
        }
    }
```

Michal Červenec
5ZYI22
2021/2022

Libraries

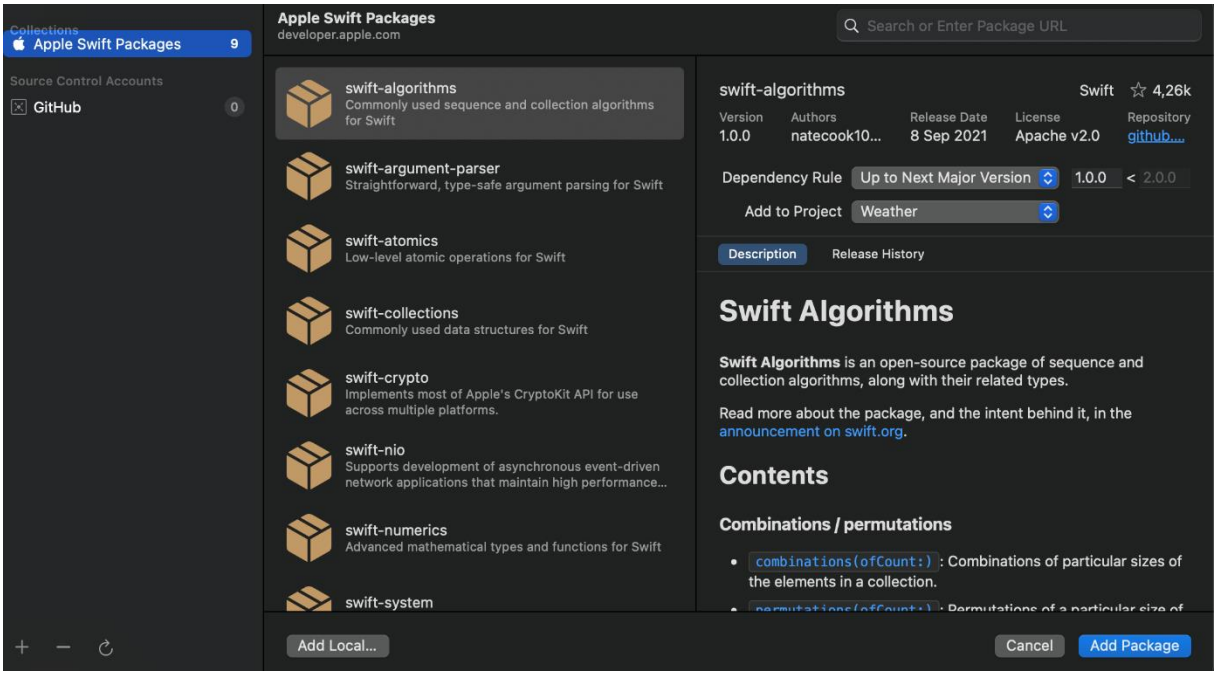
Pre zjednodušenie a zefektívnenie aplikácie je vhodné pre prácu s API využiť knižnice. Konkrétne ide o knižnicu Alamofire, ktorá uľahčí komunikáciu so vzdialeným serverom a získavanie dát. Na pridanie knižníc do projektu sa dá využiť Cocoapods, ktorý pridaním podfile do projektu stiahne a integruje zadané knižnice. Cocoapods je potrebne nainštalovať do zariadenia na ktorom prebieha vývoj a je možné ho používať v každom projekte, je postavený na Ruby (defaultne v macOS). Druhou možnosťou je pridať knižnicu pomocou relatívne novej funkcie Xcode - Swift Package Collections (keďže je novšia ešte neobsahuje niektoré knižnice).

Podfile

```
# Uncomment the next line to define a global platform for your project
# platform :ios, '14.0'

target 'Weather' do
  # Comment the next line if you don't want to use dynamic frameworks
  use_frameworks!

  # Pods for Weather
        pod 'Alamofire'
end
```



Michal Červenec
5ZYI22
2021/2022

API

Aby aplikácia zobrazila údaje o počasí potrebuje ich získať zo servera. Údaje o počasí získavame z open source projektu OpenWeather. Je potrebne sa zaregistrovať aby používateľ dostal jedinečný token nutný pre rquesty na server. Pomocou štruktúry RequestManager sa vytvorí URL adresa, ktorá slúži ako http request na server. Parametre pre URL sa dajú zistiť na stránke projektu OpenWeather. Pre potreby našej aplikácie sú nutné parametre – zem. šírka a dĺžka, token, jednotky a vylúčia sa údaje, ktoré netreba – hodinová a minútová predpoveď a upozornenia. Dáta zo serveru prídu vo formáte JSON a je potrebne ich parsovat aby sa dali správne využiť. Dáta pre dnešný deň sa rozdelia v štruktúre CurrentWeather a naformátujú do správneho tvaru. Podobne aj predpoveď pre ďalšie dni sa rozdelí v štruktúre DailyWeather a naformátuje sa, podľa popisku sa nastavuje ikona, formátuje sa výpis teploty. Jednotlivé štruktúry implementujú protokoly Decodable pre parsovanie dát a Encodable na zakódovanie údajov pre request. Takto získané dáta sa posúvajú TableView a nahrádzajú placeholdery v storyboarde (v hlavičke aj v bunkách sú nastavene príslušné dáta).

```swift
struct RequestManager {

    static let shared = RequestManager()

    func getWeatherData(for coordinates: CLLocationCoordinate2D, completion: @escaping (Result<WeatherResponse, AFError>) ->
        Void) {

        let request = WeatherRequest(
            latitude: "\(coordinates.latitude)",
            longitude: "\(coordinates.longitude)",
            appId: "7a55c30a1fb98bed4baedf14e2c9476e",
            exclude: "hourly,minutely,alerts",
            units: "metric"
        )

        let decoder = JSONDecoder()
        decoder.dateDecodingStrategy = .secondsSince1970

        AF.request("https://api.openweathermap.org/data/2.5/onecall", method: .get, parameters: request)
            .responseDecodable(of: WeatherResponse.self, decoder: decoder) { completion($0.result)
            }
    }
}
```

```swift
// MARK: - CurrentWeather
struct CurrentWeather: Decodable {

    let date: Date
    let temperature: Double
    let feelsLike: Double
    let weather: [Weather]

    var temperatureWithCelsius: String { "\(Int(temperature))°C" }
    var feelsLikeWithCelsius: String { "Feels like \(Int(feelsLike))°C" }

    enum CodingKeys: String, CodingKey {

        case date = "dt"
        case feelsLike = "feels_like"
        case temperature = "temp"
        case weather
    }
}
```

```swift
// MARK: - DailyWeather
struct DailyWeather: Decodable {

    let date: Date
    let temperature: Temperature
    let weather: [Weather]
    let precipitation: Double

    var formattedPercipitation: String { "\(Int(precipitation * 100))%"}
    enum CodingKeys: String, CodingKey {

        case date = "dt"
        case temperature = "temp"
        case precipitation = "pop"
        case weather
    }
}
```

Michal Červenec
5ZYI22
2021/2022

Search

Pre možnosť vyhľadania miesta je možné použiť searchbar. Po kliknutí na ikonku lupy v hornom pravom rohu sa zavolá viewController s vyhľadávaním. Do kolónky je možné zadávať vstup z klávesnice. Už počas písania sa vola aktualizovanie údajov pre vyhľadané subreťazce. Pomocou ďalšieho TableView sú tieto údaje zobrazované pod lištou s vyhľadávaním. Používateľ môže kliknúť na danú lokalitu a mali by sa mu zobraziť údaje o počasí pre dané miesto. Tiež si môže uložiť dané miesto medzi obľúbené.

Map

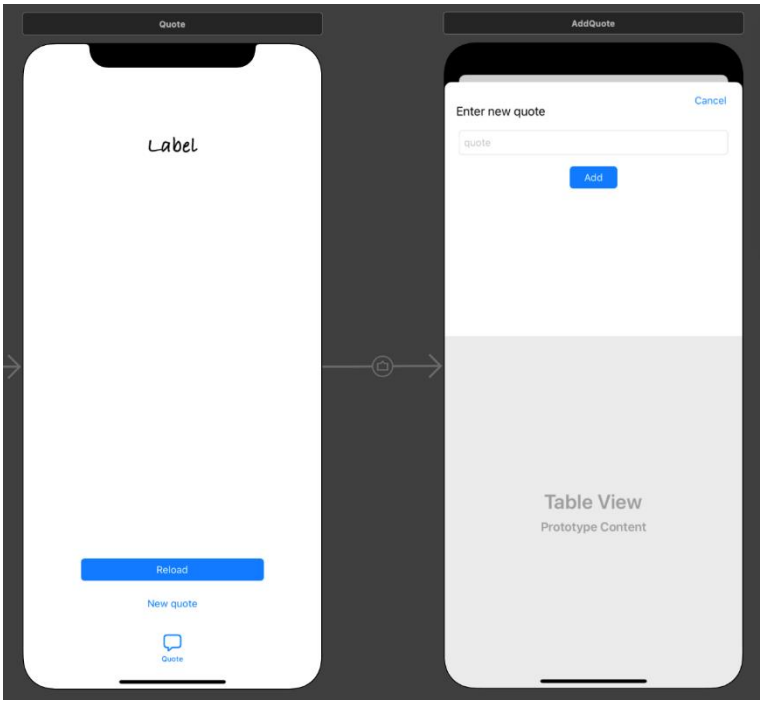
Na záložke map je možné pohybovať sa v zobrazenej mape ako v bežnej mobilnej mape, planom bolo aby bolo možné vybrať miesto a po kliknutí sa malo zobraziť počasie pre danú lokalitu podobne ako pri vyhľadaní. Momentálne zobrazuje len aktuálnu polohu zariadenia.

Favorites

Pod záložkou Favorites je storyboard v ktorom sa zobrazuje TableView s bunkami naplnenými miestami, ktoré používateľ označil ako obľúbené. Po kliknutí na konkrétne uložené miesto by sa mal zobraziť detail o počasí na danom mieste.

Quote

Poslednou funkciou aplikácie je záložka Quotes na ktorej sa zobrazujú uložené citáty o počasí. Citáty sú uložené v UserDefault ako pole stringov a pri spustení aplikácie sa náhodne jedna vyberie a zobrazí v storyboarde. Kliknutím na tlačidlo reload je možné nechať vybrať náhodne iný citát. Po kliknutí na New quote sa otvorí view s možnosťou zadať novy citát a uložiť ho medzi existujúce.



Michal Červenec
5ZYI22
2021/2022