iOS 프로그래밍 Term 프로젝트 보고서

*// 텀 프로젝트는 1인 1프로젝트입니다.*

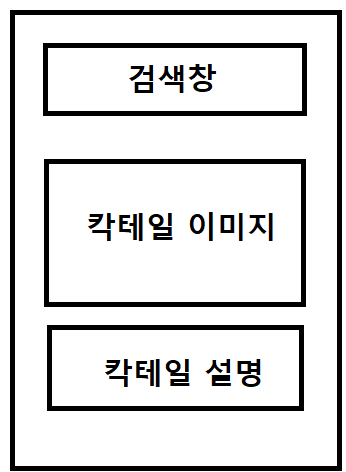
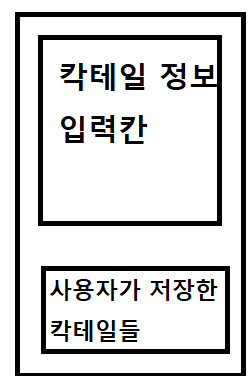
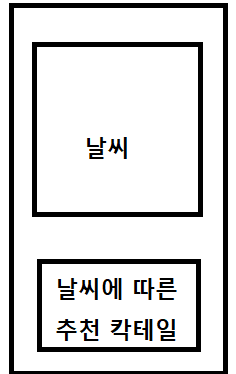
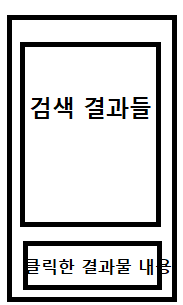
*// 모든 내용은 개조식으로 작성하지 말고 서술식으로 작성하세요(존대말 사용금지)*

*// 1-3까지 제안서, 1-6까지 보고서*

1. *학번 및 이름: 1971102 정용욱*
2. *제목: 칵테일 어플리케이션*
3. *앱 개요 및 구조(엄마, 아빠, 비컴공전공 친구 등이 이해할 수 있는 수준으로 기술)*

*개요: 궁금한 칵테일을 검색해서 사용자들이 칵테일의 정보를 찾을 수 있게 하고 혹은 사용자가 자기만의 칵테일을 만들 수 있는 기능과 위치정보를 이용한 날씨 정보로 날씨에 맞는 칵테일 추천*

*구조:*

*-------------------- 여기까지가 제안서입니다. (보고서 작성시 수정될수도 있습니다 --------*

1. *앱 기능(기술적 용어를 포함하나 일반 컴공전공 학생들을 이해할 수 있는 수준으로 기술)*

*// 각 메뉴 또는 화면 위주로 사용자들이 이 기능을 사용하여 얻게 되는 효과를 서술하시오.*

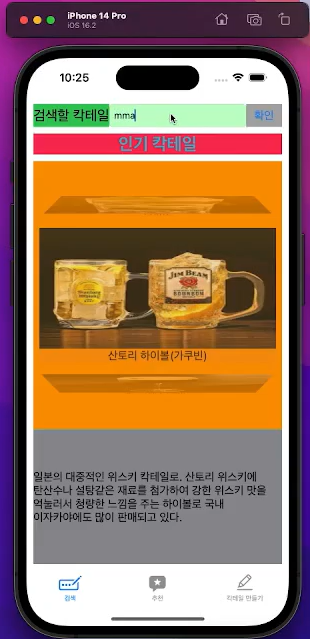
*// 예) 듣기 화면: 이 화면에서 버턴을 클릭하면 입력한 음악의 타이틀이 스트리밍으로 연주 된다.*

1. *검색 및 메인 화면* 
   1. *검색창*
      1. *사용자의 입력어가 들어가는 단어를 입력하고 검색버튼을 누르면 입력어가 포함된 칵테일 이름과 만드는 법이 있는 창으로 이동된다.*
   2. *메인화면*
      1. *인기 칵테일들을 PickerView 형식으로 인기 칵테일들에 대한 사진과 설명을 볼 수있다.*
   3. *탭바*
      1. *밑에 3개의 TabBar 아이템을 이용해서 현재의 메인화면 , 날씨에 따른 칵테일 추천, 사용자의 칵테일 레시피 등록 View로 이동할 수 있게 했다.*
   4. *기타*
      1. *메인화면 처음 입장 시 위치정보 허용 팝업이 뜨는데 이는 위치를 이용해 날씨 정보를 통한 칵테일 추천을 위해 팝업이 발생*
2. *검색 결과 화면*
   1. *검색결과들*
      1. *칵테일 API에 사용자의 검색어를 첨부하여 검색어가 포함된 모든 칵테일들이 TableView 형태로 보여준다.*
      2. *클릭 시 하단의 Label에 클릭한 칵테일의 레시피가 뜬다*
3. *날씨에 따른 칵테일 추천 화면*
   1. *날씨 이미지*
      1. *사용자의 위도 경도 정보를 이용해서 현재 날씨(강우,흐림,맑음,눈) 등의 정보를 축약해서 이미지로 보여준다*
   2. *추천 칵테일 이미지*
      1. *사용자가 있는 곳의 온도를 Weather API를 이용해서 찾아 해당 온도에 맞는 칵테일의 이미지를 하나 보여준다.*
   3. *추천 칵테일들과 해당 칵테일의 설명*
      1. *사용자가 있는 날씨에 맞는 여러 칵테일들의 추천을 Label을 통해 글로 보여준다.*
4. *칵테일 레시피 저장 화면*
   1. *칵테일 레시피 입력창*
      1. *칵테일 이름,제조비용,레시피를 입력할 수 있는 TextField가 있고 저장버튼을 누르면 파이어베이스에 저장되고 하단의 TableView에 해당 데이터도 reload된다.*
   2. *사용자가 저장한 레시피들*
      1. *사용자가 저장했던 레시피가 TableView를 통해 보여준다.*
5. 구현 내용(작성자와 담당 교수만 이해하면 될 정도의 심도 깊게 서술)

// 각 기능별로 구현 내용(사용된 API 또는 알고리즘)을 기술적 용어를 사용하여 서술한다. 각 서술에서 필요하다면 아래와 같이 코드(또는 pseudo code)를 직접 삽입 할 수 있다.

|  |
| --- |
| extension StillImageViewController: UINavigationControllerDelegate, UIImagePickerControllerDelegate{    func imagePickerController(\_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {  // 이미지를 가져온다  if let image = info[UIImagePickerController.InfoKey.originalImage] as? UIImage{  // 가져온 이미지를 화면에 보이도록 imageView에 적용한다  imageView.image = image    // 가져온 이미지를 ImageClassifier에 분류를 요청한다.  let handler = VNImageRequestHandler(ciImage: CIImage(image: image)!)  try! handler.perform([request])  }  // ImagePickerController을 제거한다  picker.dismiss(animated: true, completion: nil)  }  } |

1. *검색 및 메인 화면*

**

*시작하기 전에 날씨에 따른 칵테일 추천을 위해 사용자의 위도 경도를 알아내기 위해*

*위치정보 허용 메시지를 요구, 허용해야 정상적으로 앱이 실행되며*

*화면 구성은 해당 메인 화면에서는 사용자로부터 검색을 입력 받을 수 있는 TextField가 존재하며*

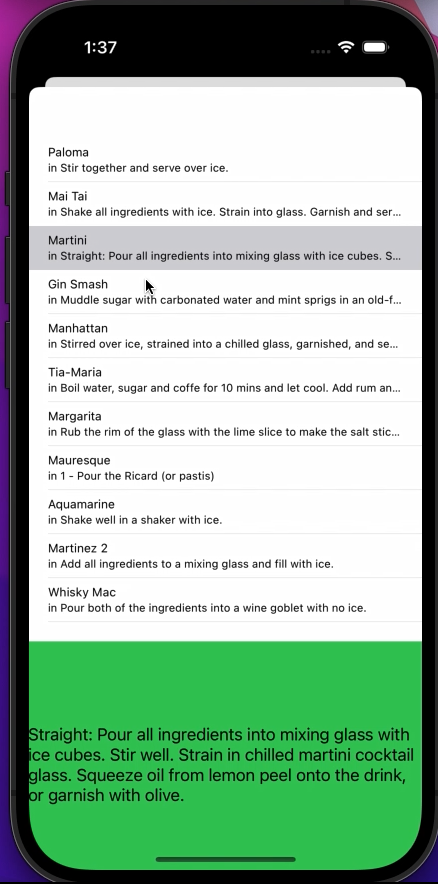
*검색 버튼을 누르면 textField의 내용이 검색결과 화면으로 segue로 전달한다.*

*또한 칵테일 정보가 담긴 json파일의 내용에서 Cocktail클래스에 정보를 옮기고 배열에*

*Cocktail클래스를 담아 해당 배열을 PickerView의 dataSource를 지정하여 PickerView를 통해 칵테일의 이미지를 보여주고 밑에 Label에 해당 칵테일의 설명을 나타낸다.*

*그리고 TabBar를 사용하여 검색(메인페이지),날씨에 따른 추천,칵테일 만들기 페이지로 이동할 수 있다.*

1. *검색 결과 화면*



*검색 및 메인 화면에서 segue를 통해 받은 검색어를 받아서 API URL에 첨부 후 요청*

*응답 받은 내용을 내가 만든 Cocktail 구조체 배열로 로컬에 저장*

*코드에서는 self.cocktails = result.drinks 이 부분에서 로컬 배열인 cocktails에 응답 받은 result의 drink[{}]의 정보를 집어넣는다*

*그리고 cocktails 배열을 TableView dataSource로 지정*

*해당 API의 주소는 밑에 있는 api를 사용했다.*

*사용한 API의 URL 예시*

*(*[www.thecocktaildb.com/api/json/v1/1/search.php?s=](http://www.thecocktaildb.com/api/json/v1/1/search.php?s=)*)(검색키워드)*

*TableView의 cell을 클릭하면 해당하는 칵테일의 설명이 밑에 Label에 나타나게 된다.*

*밑에 코드와 같이 API의 응답을 Json 데이터로 받아 클래스 배열에 집어 넣는*

*일부코드)*

*let decoder = JSONDecoder()*

*let result = try decoder.decode(CocktailResponse.self, from: data)*

*print(result)*

*self.cocktails = result.drinks*

*print("\*\*\*\*\*\*\*\*\*\*")*

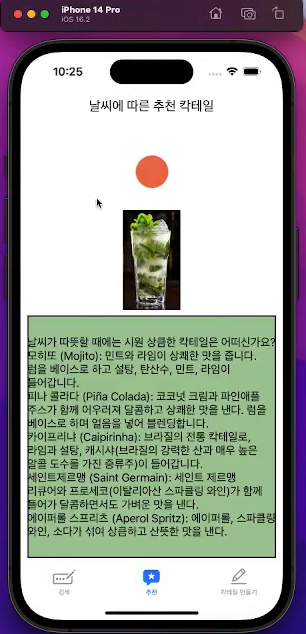
*print(self.cocktails.first)*

*DispatchQueue.main.async {*

*self.tableView.reloadData()*

*}*

1. *날씨에 따른 칵테일 추천 화면*

**

*처음에 디바이스의 위치정보 열람을 허용했기 때문에 CLLocationManager를 사용해 현재 디바이스의 위도 경도를 알아냄*

*밑에는 디바이스의 위도 경도를 찾는 코드이고,*

*CLLocationManagerDelegate 프로토콜을 준수하는 LocationManager 클래스를 만들어*

*locationManager 함수에서 위도 경도를 찾는다*

*찾은 위도 경도 정보를 Weather API를 이용해서 현재 날씨에 대한 정보를 받고, 응답 받은 내용 중 기온 정보를 활용하여 해당 기온에 맞는 칵테일을 추천한다.*

*Ex) 30도 이상엔 시원하고 갈증을 해결할 수 있는 칵테일, 20~30도의 따뜻한 날씨엔 시원하고 달달한 칵테일 15도 이하의 선선하거나 추운 날씨에는 따뜻한 칵테일이나 도수가 높은 칵테일을 추천하는 식으로 날씨에 따른 칵테일 추천을 함*

*위도 경도를 알아내기 위한 LocationManager 일부 코드)*

*class LocationManager: NSObject, CLLocationManagerDelegate {*

*private let locationManager = CLLocationManager()*

*var onLocationUpdate: ((CLLocationCoordinate2D) -> Void)?*

*override init() {*

*super.init()*

*locationManager.delegate = self*

*locationManager.desiredAccuracy = kCLLocationAccuracyBest*

*locationManager.requestWhenInUseAuthorization()*

*}*

*func startUpdatingLocation() {*

*if CLLocationManager.locationServicesEnabled() {*

*locationManager.startUpdatingLocation()*

*}*

*}*

*func locationManager(\_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {*

*if let location = locations.last {*

*onLocationUpdate?(location.coordinate)*

*locationManager.stopUpdatingLocation()*

*}*

*}*

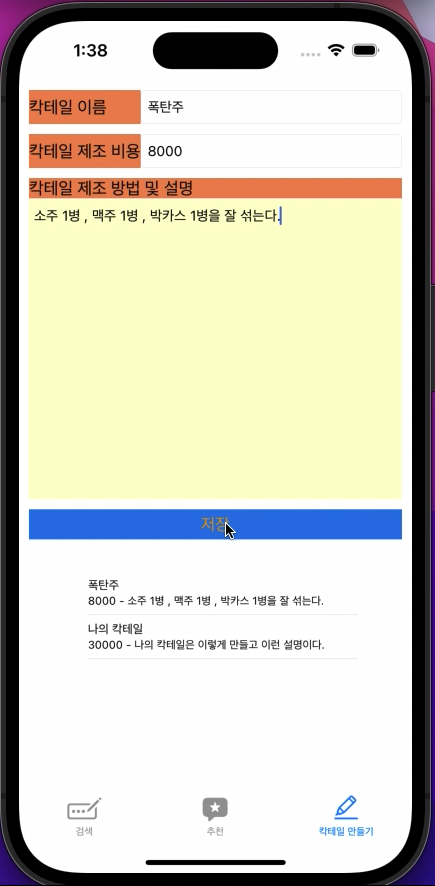
*func locationManager(\_ manager: CLLocationManager, didFailWithError error: Error) {*

*print("Failed to get location: \(error.localizedDescription)")*

*}*

*}*

1. *칵테일 레시피 저장 화면*



*사용자가 칵테일 이름,칵테일 제조 비용,칵테일 설명 등을 기술하고 저장 버튼을 누르면 해당 정보는 Firebase Database에 저장 그리고 칵테일 레시피에 대한 클래스 배열에도*

*저장되고 tableView를 reload*

*그래서 사용자가 저장버튼을 누르면 바로바로 TableView에 저장한 레시피가 뜸*

*밑에 코드와 같이 저장버튼이 클릭하는 경우를 IBAction으로 받아서 saveCocktail()에서 사용자가 입력한 정보를 saveCocktailToFirestore()로 넘겨 해당 부분에서 파이어베이스에 데이터를 저장한다,*

*그리고 fetchCocktailsFromFirestore() 함수는 파이어베이스에 있는 칵테일 레시피 정보를 불러와 로컬 배열에 넣고 reload하는 동작을 수행*

*code) 파이어베이스의 데이터를 가져오는 함수 fetchCocktailsFromFirestore 와*

*저장 버튼 Acion에 따라 해당 데이터를 파이어베이스 저장하고 TableView를 reload 코드*

*@IBAction func saveCocktail(\_ sender: UIButton) {*

*guard let name = cocktailName.text, !name.isEmpty,*

*let price = cocktailPrice.text, !price.isEmpty,*

*let description = cocktailDescription.text, !description.isEmpty else {*

*let alert = UIAlertController(title: "Error", message: "All fields must be filled out", preferredStyle: .alert)*

*alert.addAction(UIAlertAction(title: "OK", style: .default, handler: nil))*

*present(alert, animated: true, completion: nil)*

*return*

*}*

*let cocktail = CocktailUser(cocktailName: name, cocktailPrice: price, cocktailDescription: description)*

*saveCocktailToFirestore(cocktail)*

*}*

*func saveCocktailToFirestore(\_ cocktail: CocktailUser) {*

*let db = Firestore.firestore()*

*do {*

*let data = try JSONEncoder().encode(cocktail)*

*let json = try JSONSerialization.jsonObject(with: data, options: []) as! [String: Any]*

*db.collection("cocktails").addDocument(data: json) { [weak self] error in*

*if let error = error {*

*print("Error adding document: \(error)")*

*} else {*

*print("Document added successfully")*

*self?.fetchCocktailsFromFirestore()*

*}*

*}*

*} catch {*

*print("Error encoding cocktail: \(error)")*

*}*

*}*

*func fetchCocktailsFromFirestore() {*

*let db = Firestore.firestore()*

*db.collection("cocktails").getDocuments { [weak self] (querySnapshot, error) in*

*if let error = error {*

*print("Error getting documents: \(error)")*

*} else {*

*self?.cocktails = querySnapshot?.documents.compactMap { document -> CocktailUser? in*

*do {*

*let data = try JSONSerialization.data(withJSONObject: document.data(), options: [])*

*let cocktail = try JSONDecoder().decode(CocktailUser.self, from: data)*

*return cocktail*

*} catch {*

*print("Error decoding cocktail: \(error)")*

*return nil*

*}*

*} ?? []*

*self?.cocktailTableView.reloadData()*

*}*

*}*

*}*

1. 결론**(반드시 작성)**

// 개발에서 느꼈던 좋았던 점, 아쉬운 점 등을 자유롭게 서술하시오.

개발을 하면서 그동안 배웠던 TabBar,TableView,Firebase,Segue 등 다양한 기술들을 직접 내가 프로그래밍하면서 iOS 프로그래밍에 대한 이해도가 더 높아졌다고 생각하고 내가 직접 만드는 창작의 기쁨을 느껴서 좋았다. 아쉬운 점으로는 맥북이 없는 상태에서 vmware를 사용하는 것이 매우 느려 개발에 어려움을 겪었고, 그래서 강의실의 컴퓨터를 사용하면서 꾸준히 개발을 해왔는데 아무래도 해당 강의실이 수업 중이거나 내가 시간을 낼 수 없는 경우가 있어서 개발에 들인 시간이 부족했던 것이 아쉬웠다.

1. 제출(elcass에 기한내 업로드)

* 제안서: project-제안서-학번-이름.[docx | hwp] // 5월20일
* 보고서: project-보고서-학번-이름.[docx | hwp] // 6월 17일 08시
* 시연 동영상: project-동영상-학번-이름.mp4 // 6월 17일 08시
* 소스코드: project-소스코드-학번-이름.zip // 6월 17일 08시