CSE 335 CLASS PROJECT PHASE III

HARDIK GUPTA

IOS TRAVEL DIARY

TRAVEL DIARY

- Motivation behind this project: I like to travel and organize pictures I took while traveling.
- UI/ Functionalities added according to Class Project requirements
- Traditional Photos app has a very bad UI based on Map. No list available.
- The user can also save some comments about the picture.
- Targeted Users: Travelers who want to remember the specific point of interest they took image at.


UITableViewDelegate
tableViewController
- var obj : PlacesModel = PlacesModel()
- @IBOutlet weak var tableViewObj : UITableView!
+ func tableView(_ tableView : UITableView, numberOfRowsInSection section : Int) : int
+ func tableView(_ tableView : UITableView, cellForRowAt indexPath : IndexPath) : UITableViewCell
+ func tableView(_ tableView : UITableView, canEditRowAt indexPath : IndexPath) : boolean
+ func tableView(tableView : UITableView, editingStyleForRowAtIndexPath indexPath : IndexPath) : UITableViewCell.EditingStyle
+ func tableView(_ tableView : UITableView, commit editingStyle : UITableViewCell.EditingStyle, forRowAt indexPath : IndexPath) : void
+ override func viewDidLoad() : void
+ override func prepare(for segue : UIStoryboardSegue, sender : Any?) : void
PlacesModel
- var fetchResults : int = [Places]()
- let managedObjectContext : int = (UIApplica...
+ init() : void
+ func addPlaceInfo(pName : String, det : String, picture1 : Data) : void
+ func numberofrecords() : Int
+ func getPlaceObject(nameOfPlace : String) : Places
+ func doesPlaceInfoExist(nameOfPlace : String) : boolean
+ func delete(x : Int) : void
+ func updatePlace(nameS : String, infoS : String, picture1 : Data) : void
Places
- @NSManaged public var namePlace : String?
- @NSManaged public var photo : Data?
- @NSManaged public var info : String?
+ @nonobjc public class func fetchRequest() : NSFetchRequest<Places>
UINavigationControllerDelegate
UITableViewDataSource
UITableViewCell
savedTableViewCell
- @IBOutlet weak var placeLabel : UILabel!
ViewController
- var yDistance : CGFloat
- @IBOutlet weak var mainImageLabel : UIImageView!
- @IBOutlet weak var cityColumn : UITextField!
- @IBOutlet weak var countryColumn : UITextField!
- var coordinate : CLLocationCoordinate2D?
- var manager : CLLocationManager!
- var inputCity : String?
- var inputCountry : String?
- @IBOutlet weak var locationImage : UIImageView!
- @IBOutlet weak var saveImage : UIImageView!
+ @objc func keyboardWillShow(_ sender : Notification) : void
+ touchesBegan(_ touches : Set<UITouch>, with event : UIEvent?) : void
+ func textFieldShouldReturn(_ textField : UITextField) : boolean
+ @objc func keyboardWillHide(_ sender : Notification) : void
+ @IBAction func locationNeed(_ sender : Any) : void
+ @IBAction func proceedPress(_ sender : Any) : void
+ class func isLocationServiceEnabled() : boolean
+ func locationManager(_ manager : CLLocationManager, didUpdateLocations locations : [CLLocation]) : void
+ func locationManager(_ manager : CLLocationManager, didFailWithError error : Error) : void
+ override func didReceiveMemoryWarning() : void
+ @IBAction func unwindToVC(sender : UIStoryboardSegue) : void
+ override func prepare(for segue : UIStoryboardSegue, sender : Any?) : void
detailViewController
- var yDistance : CGFloat
- @IBOutlet weak var placeLabel : UILabel!
- @IBOutlet weak var cityLabel : UILabel!
- @IBOutlet weak var countryLabel : UILabel!
- @IBOutlet weak var mapLabel : MKMapView!
- @IBOutlet weak var imageLabel : UIImageView!
- @IBOutlet weak var descriptionLabel : UILabel!
- @IBOutlet weak var segmentLabel : UISegmentedControl!
- var placeString : String?
- var coordinate : CLLocationCoordinate2D?
- let picker : int = UIImagePickerController()
- var obj : PlacesModel = PlacesModel()
+ @objc func keyboardWillShow(_ sender : Notification) : void
+ override func touchesBegan(_ touches : Set<UITouch>, with event : UIEvent?) : void
+ func textFieldShouldReturn(_ textField : UITextField) : boolean
+ @objc func keyboardWillHide(_ sender : Notification) : void
+ @IBAction func changeDesButton(_ sender : Any) : void
+ @IBAction func changeImageButton(_ sender : Any) : void
+ @IBAction func saveButtonPress(_ sender : Any) : void
+ override func viewDidLoad() : void
+ func imagePickerControllerDidCancel(_ picker : UIImagePickerController) : void
+ func imagePickerController(_ picker : UIImagePickerController, didFinishPickingMediaWithInfo info : [UIImagePickerController.InfoKey : Any]) : void
UIImagePickerControllerDelegate
UINavigationControllerDelegate
UIImagePickerControllerDelegate
CLLocationManagerDelegate
listViewController
- var obj : PlacesModel = PlacesModel()
- @IBOutlet weak var textOut : UITextView!
- @IBOutlet weak var mapLabel : MKMapView!
- var city : String?
- var country : String?
- var resultCity : [String] = []
- var counter : int
- var ans : String
- var coordinate : CLLocationCoordinate2D?
+ @IBAction func continuePress(_ sender : Any) : void
+ @IBOutlet weak var inputPlace() : UITextField!
+ override func viewDidLoad() : void
+ func apiCall() : void
+ override func prepare(for segue : UIStoryboardSegue, sender : Any?) : void
+ @IBAction func unwindToLVC(sender : UIStoryboardSegue) : void
UITextFieldDelegate
addViewController
- var obj : PlacesModel = PlacesModel()
- var namePlace : String
- var city : String
- var country : String
- var yDistance : CGFloat
- let picker : int = UIImagePickerController()
- var image : UIImage?
- var det : String
- var coordinate : CLLocationCoordinate2D?
- var pN : String?
- @IBOutlet weak var mapOfPlace : MKMapView!
- @IBOutlet weak var infoInput : UITextField!
- @IBOutlet weak var imageInput : UIImageView!
- @IBOutlet weak var selectedSegment : UISegmentedControl!
+ @IBAction func addPicButton(_ sender : Any) : void
+ func imagePickerControllerDidCancel(_ picker : UIImagePickerController) : void
+ func imagePickerController(_ picker : UIImagePickerController, didFinishPickingMediaWithInfo info : [UIImagePickerController.InfoKey : Any]) : void
+ @IBAction func saveButton(_ sender : Any) : void
+ override func viewDidLoad() : void

Hardik Gupta

Controller

1. Retrieves JSON data (API Calls) and saves them as a dictionary
2. Retrieves List of Places Saved
3. Calls TableView, imagePickerController, locationManager mapView delegate methods
4. Updates Table Views, Image Views, Text Views
5. Computes User Location

Notifies User Actions:

Whenever a button, or cell in a table, is clicked
View TableView Cells for Name, Picture
When the map is clicked
When a photo is picked or clicked

Notifies:

Changes in JSON Data
Change in image for a place
New PlaceEntity Object created (new place added)

Update:

Creation of a new PlaceEntity Object
Send place name for Modification of PlaceEntity object
List of saved places

Update:

Text View displaying saved places
Table View listing Places To Visit
Map View displaying place location
Updates Place information -image added by user or description

View

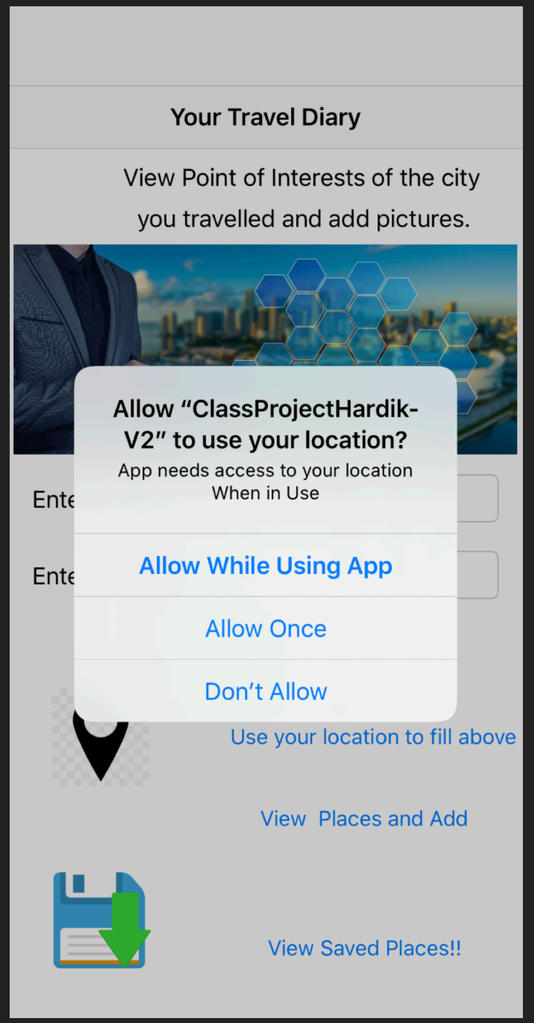
1. Displays home screen, text boxes to get user input, search button, use my location button, travel diary button.
2. Displays Places to Visit in a City which use can select and go to next screen where user can type description and add an image of that city.
3. Displays list of saved places from core data and after user selects it, shows location on map, picture and description.

Model

1. Stores Place Information in Core Data
2. Functions to add, retrieve, and edit PlaceEntity objects from Core Data
3. Returns a list of places been visited.
4. Returns Places to Visit in TextView

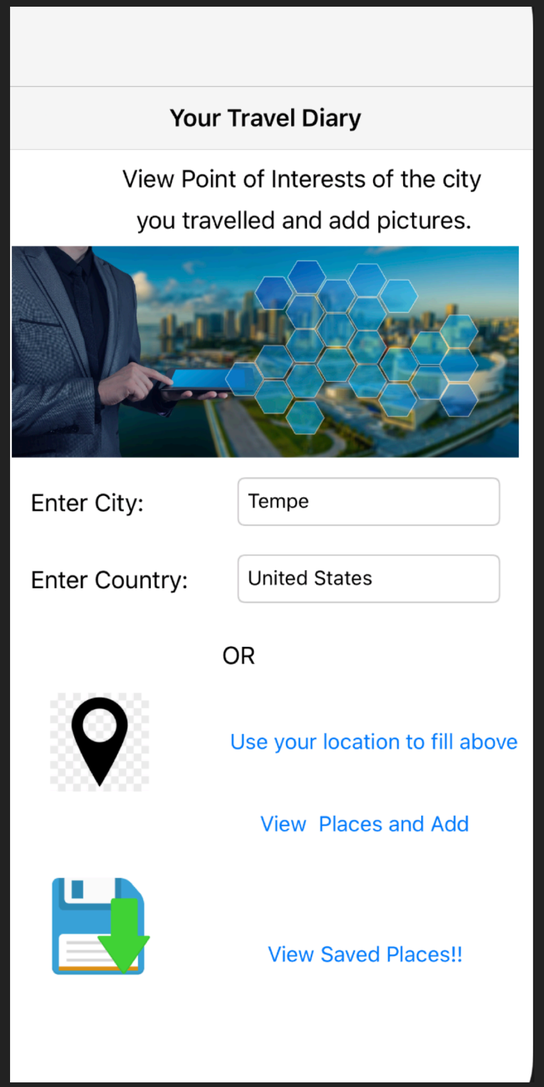
USE OF LOCATION SERVICES

- The app asks for permission from the user to access the user’s current location.
- The user can deny, allow or choose to allow location access only when the app is in use.



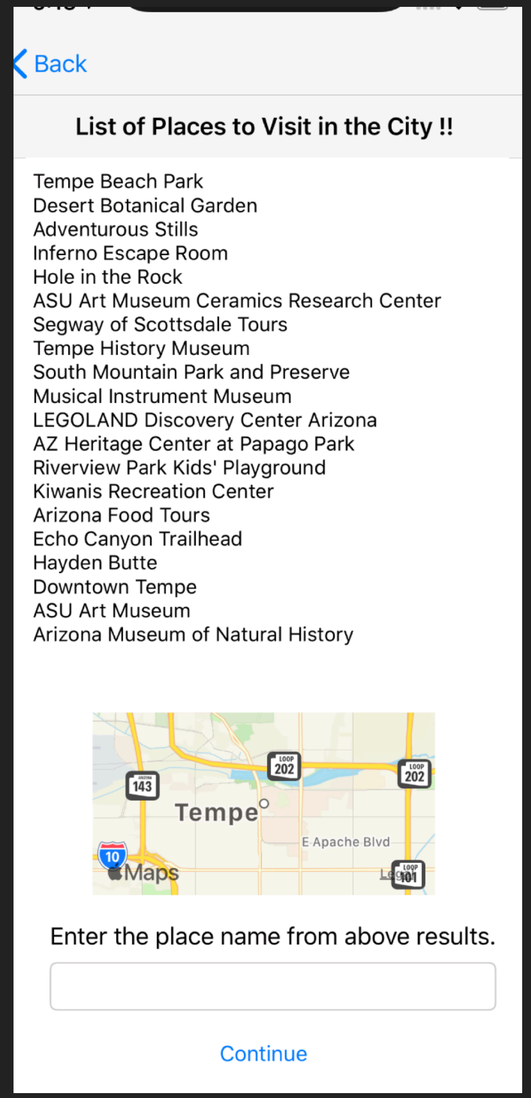
USE OF LOCATION SERVICES

- When “Use your location to fill above” is pressed, user’s current locations coordinates are found out using MapKit and using geocoder the City and Country are found out.



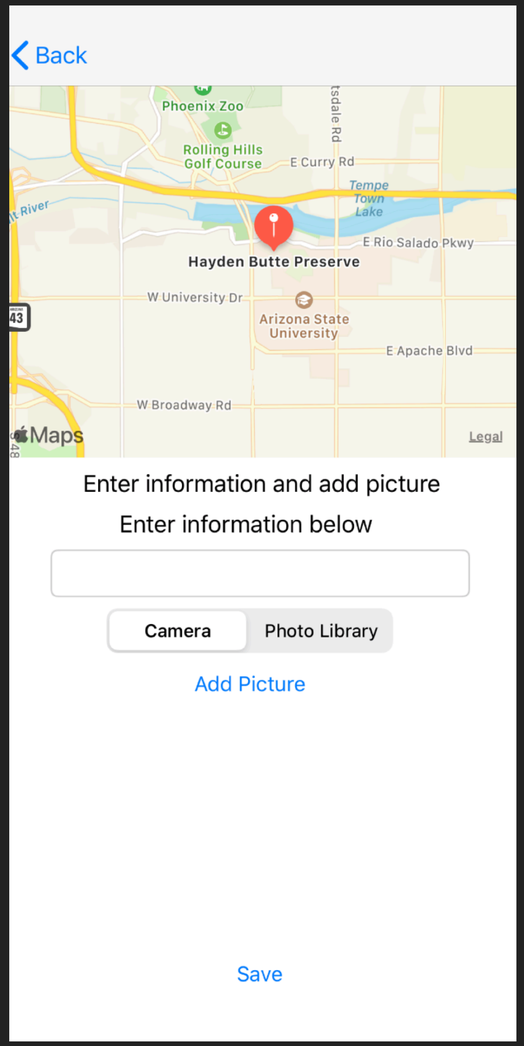
LISTING POINT OF INTERESTS OF THE LOCATION USER MENTIONED

- Gets the City name and Country name using segue from ViewController and API calls from google API “Places API” to list famous tourists spot of the locality.



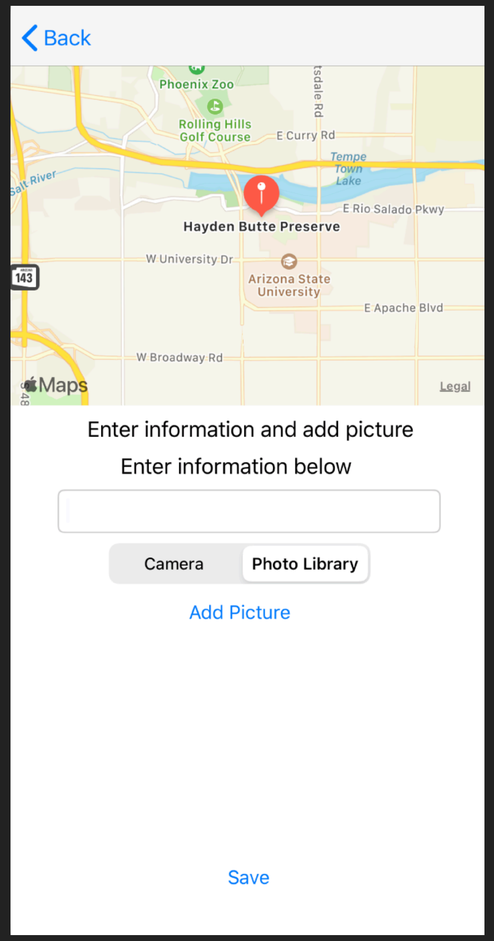
ANNOTATIONS ON MAP OF ENTERED PLACE FROM THE LIST

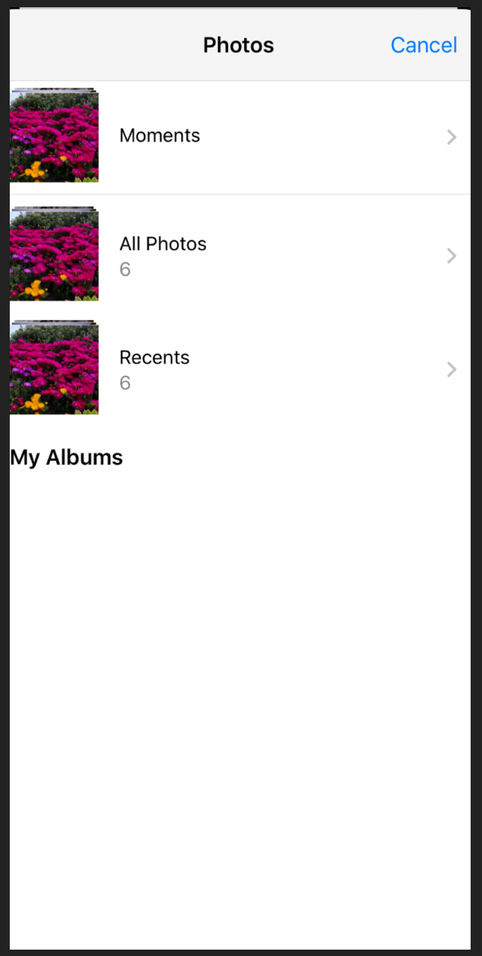
- Drops a pin where the selected place is on the Map View.



ADD A PHOTO FROM CAMERA OR PHOTO LIBRARY

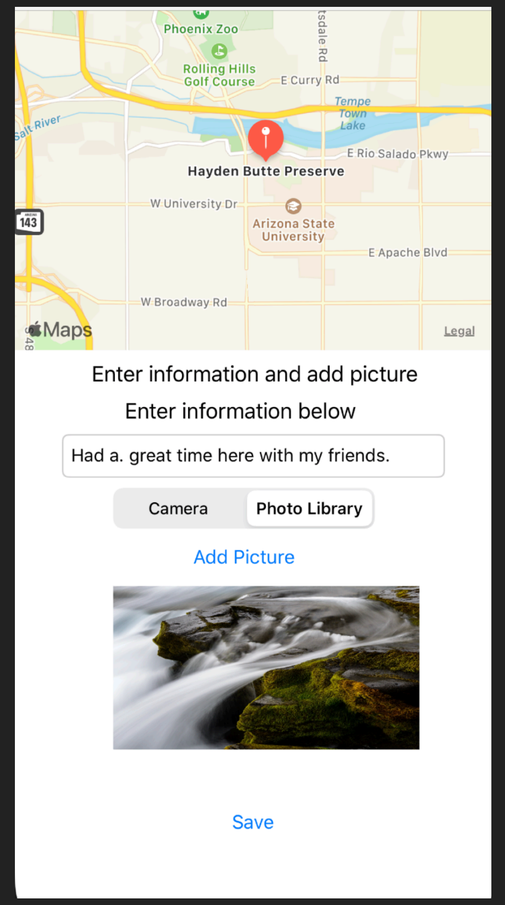
- Uses image picker view.
- After clicking Add Picture, based on selected segment in the Segmented Controller, proceeds to add picture.
- If Photo Library was selected and Add Picture was pressed it would take the user to the photo library to choose image.





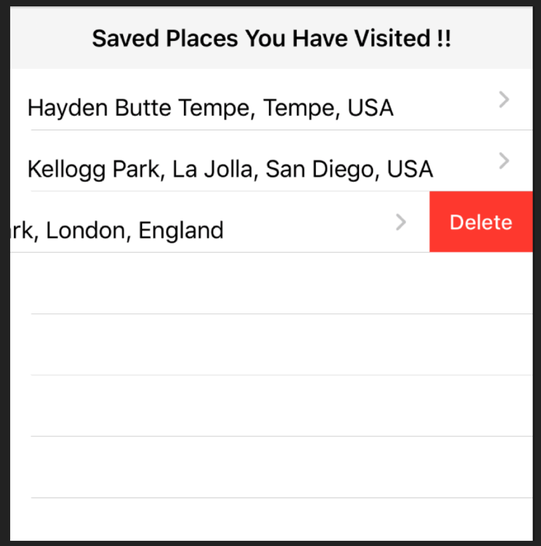
DATA IS STORED IN CORE DATA

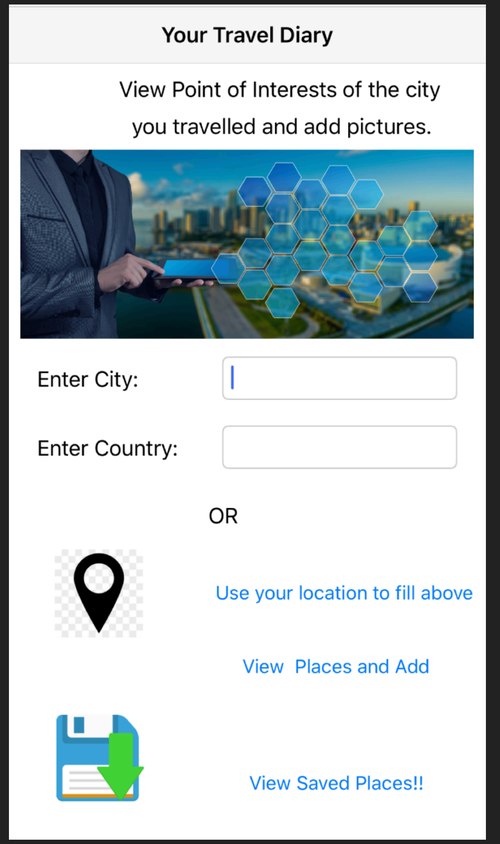
- After save is pressed, the data is stored in the form of Entity with having three attributes i.e Name of Place, Photo and Comment about the photo.

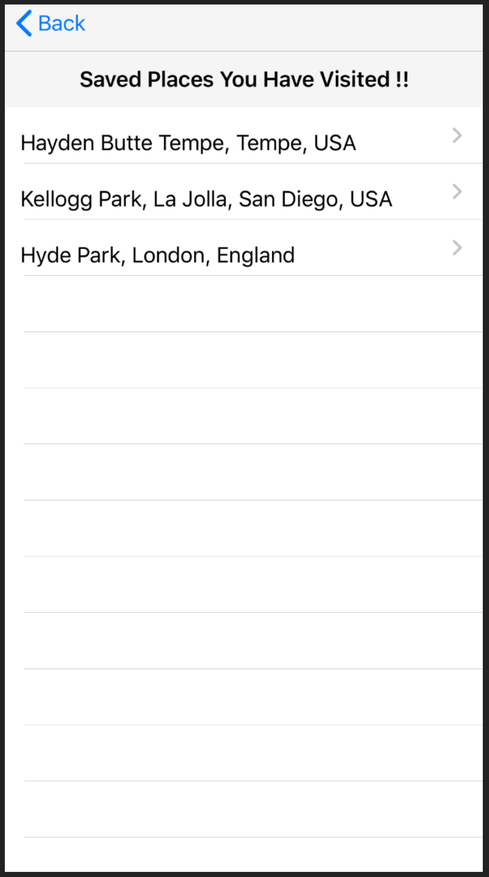


VIEW SAVED PLACES IN CORE DATA IN THE FORM OF TABLE VIEW

- After pressing View Saved Places, user can view the saved places in the form of table view and click on each cell to view their respective picture and Information.
- The user can also swipe left on the cell to delete an entry.

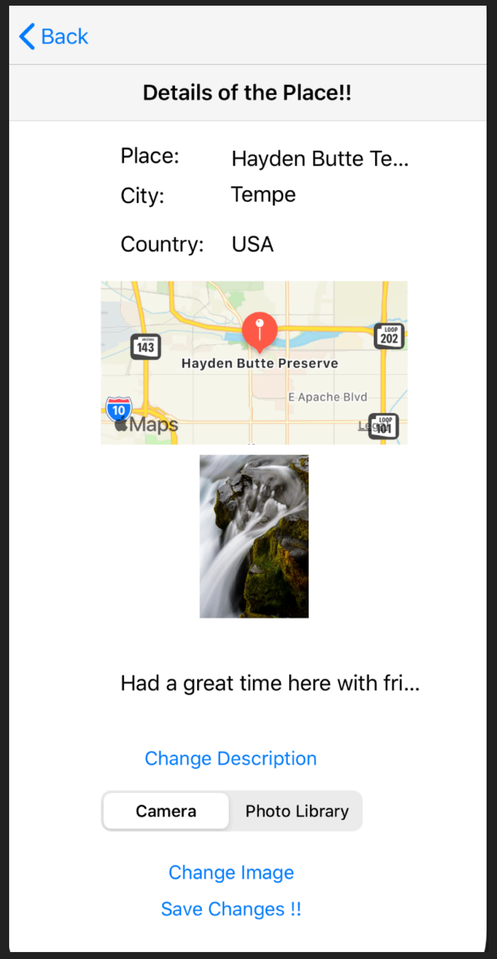


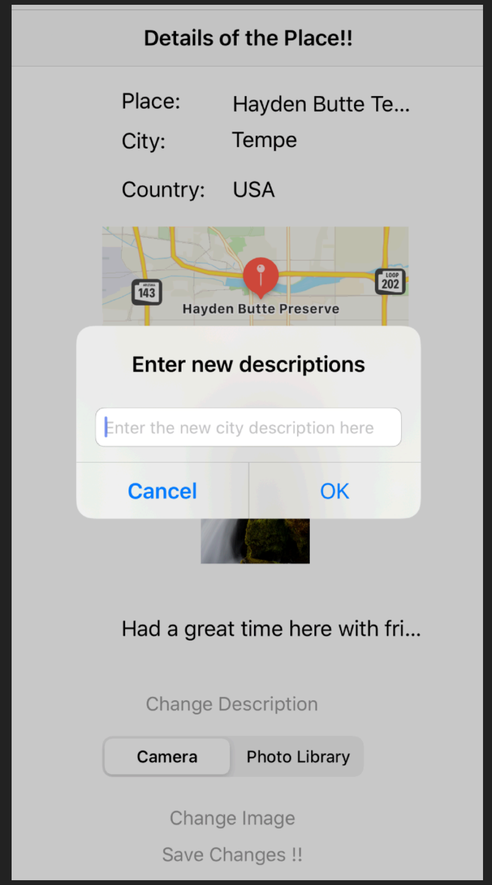




VIEW, EDIT, DELETE SAVED DATA

- The user can view what user saved.
- The user can change description or change image the same way user added image.
- User can save changes in core data.





LIMITATIONS OF THE PROJECT

- API Call results limited to a certain number and a certain city not locality unless the user modifies city name to include locality and city in city column.
  For ex - Enter City : La Jolla, San Diego where La Jolla is a locality and San Diego is a city.
- Speed of saving the core data: It takes a while saving a new city in core data. It takes approximately 5 to 6 seconds.
- Annotations on Map may pin somewhere else based on different locations found on Apple Maps with same name.

LESSONS LEARNED FROM THIS PROJECT

- Start project on time otherwise you would not be able to give good output and have a few sleepless nights completing the project.
- Building an app takes a lot of effort and apps we make in the class are no where near as good as apps like FaceBook, Whatsapp, etc.
- Technical lessons:
  a. How to make an API call using API key and Google API.
  b. Always connect the tableView object to showDelegate and the second option otherwise you will waste a lot of time figuring it out.

SUGGESTIONS FOR THE CLASS/PROJECT

- The pre req should be CSE310. Although I am a senior but it might been difficult for Sophomores.
- Class should include more technical stuff as taught in Stanford (iTunes University Videos) like how to change keyboard options to click done instead of return, etc.