```swift
//
//  AppDelegate.swift
//  DatabaseDine
//
//  Created by Jacob on 12/3/17.
//  Copyright © 2017 Jacob. All rights reserved.
//

import UIKit
import CoreData
import Firebase

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?   //Initialize application


    internal func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.

        FirebaseApp.configure() //Creates the connection to Firebase
for commenting


        return true
    }

    func applicationWillResignActive(_ application: UIApplication) {
        // Sent when the application is about to move from active to
inactive state. This can occur for certain types of temporary
interruptions (such as an incoming phone call or SMS message) or when
the user quits the application and it begins the transition to the
background state.
        // Use this method to pause ongoing tasks, disable timers, and
invalidate graphics rendering callbacks. Games should use this method
to pause the game.
    }

    func applicationDidEnterBackground(_ application: UIApplication) {
        // Use this method to release shared resources, save user
data, invalidate timers, and store enough application state
information to restore your application to its current state in case
it is terminated later.
        // If your application supports background execution, this
method is called instead of applicationWillTerminate: when the user
quits.
    }
```

```
    func applicationWillEnterForeground(_ application: UIApplication)
{
        // Called as part of the transition from the background to the
active state; here you can undo many of the changes made on entering
the background.
    }

    func applicationDidBecomeActive(_ application: UIApplication) {
        // Restart any tasks that were paused (or not yet started)
while the application was inactive. If the application was previously
in the background, optionally refresh the user interface.
    }

    func applicationWillTerminate(_ application: UIApplication) {
        // Called when the application is about to terminate. Save
data if appropriate. See also applicationDidEnterBackground:.
    }


}

//
//  SlayterClass.swift
//  Dineson
//
//  Created by Federico Read on 11/10/17.
//  Copyright © 2017 testing. All rights reserved.
//

import UIKit
import WebKit
class SlayterClass: UIViewController {

    @IBOutlet weak var slayterWebView: WKWebView!

    @IBOutlet weak var Slayter_status: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
        let url = URL(string:"http://denison.cafebonappetit.com/cafe/
slayter-market/") //webview link
        let request = URLRequest(url:url!)

        slayterWebView.load(request)
        //segues from Slayter to curtis pages
        let RightSwipe = UISwipeGestureRecognizer(target: self,
action: #selector(SlayswipeAction(swipe:)))
```

```
        RightSwipe.direction = UISwipeGestureRecognizerDirection.left
        self.view.addGestureRecognizer(RightSwipe)

        self.dateCheck()
    }

    func dateCheck()
    {
        let today = Date().dayOfWeek()

        if today == 1 || today == 7
        {
            //print("Sunday")
            self.weekend()
        }

        else
        {
            self.weekDay()
        }
    }

    func getTime() -> (hour:Int, minute:Int, second:Int) {
        let currentDateTime = Date()
        let calendar = NSCalendar.current
        let component =
calendar.dateComponents([.hour,.minute,.second], from:
currentDateTime)
        let hour = component.hour
        let minute = component.minute
        let second = component.second
        return (hour!,minute!,second!)
    }


    func closed(endTime:String)
    {
        Slayter_status.text = "CLOSED until"+endTime
        Slayter_status.backgroundColor = UIColor.red
        Slayter_status.textColor = UIColor.white

    }

    func opened(endTime:String)
    {
        Slayter_status.text = "Open! Declining NOT accepted until
"+endTime
        Slayter_status.backgroundColor = UIColor.green
        Slayter_status.textColor = UIColor.white
    }
```

```
    func openedDeclining(endTime:String)
    {
        Slayter_status.text = "Open and accepting Declining until
"+endTime
        Slayter_status.backgroundColor = UIColor.green
        Slayter_status.textColor = UIColor.white
    }

    func weekDay()
    {
        let time = getTime().hour

        switch time
        {
        case 0: openedDeclining(endTime: " 11")

        case 7...10: openedDeclining(endTime: " 10:45")
        case 11...12: opened(endTime: " 1")
        case 13...23: openedDeclining(endTime: " 1")
        default:closed(endTime: " 7")
        }
    }

    func weekend()
    {
        let time = getTime().hour

        switch time
        {
        case 12...23: openedDeclining(endTime: " 2:30")

        default:closed(endTime: " 12")
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a
little preparation before navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?)
{
        // Get the new view controller using
```

```
segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */

}

extension UIViewController
{
    @objc func SlayswipeAction(swipe:UISwipeGestureRecognizer)
    {
        switch swipe.direction.rawValue {
        case 2:
            performSegue(withIdentifier: "Slay2Curt", sender: self) //
segues name and location
        default:
            break
        }
    }
}
//
//  HuffmanClass.swift
//  Dineson
//
//  Created by Federico Read on 11/10/17.
//  Copyright © 2017 testing. All rights reserved.
//

import UIKit
import WebKit


class HuffmanClass: UIViewController {

    @IBOutlet weak var huffmanWebView: WKWebView!

    @IBOutlet weak var HuffStatus: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.

        let url = URL(string:"http://denison.cafebonappetit.com/cafe/
huffman-cafe/") //webview link
        let request = URLRequest(url:url!)

        huffmanWebView.load(request)
        //segues from Huffman to slayter pages
        let RightSwipe = UISwipeGestureRecognizer(target: self,
```

```
action: #selector(HuffswipeAction(swipe:)))
        RightSwipe.direction = UISwipeGestureRecognizerDirection.left
        self.view.addGestureRecognizer(RightSwipe)
        self.dateCheck()
    }

    func dateCheck()
    {
        let today = Date().dayOfWeek()

        if today == 1 || today ==  7
        {
            //print("Sunday")
            self.weekend()
        }
        else
        {
            //print("Saturday")
            self.weekDay()
        }
    }

    func getTime() -> (hour:Int, minute:Int, second:Int) {
        let currentDateTime = Date()
        let calendar = NSCalendar.current
        let component =
calendar.dateComponents([.hour,.minute,.second], from:
currentDateTime)
        let hour = component.hour
        let minute = component.minute
        let second = component.second
        return (hour!,minute!,second!)
    }


    func closed(endTime:String)
    {
        HuffStatus.text = "Will be CLOSED until"+endTime
        HuffStatus.backgroundColor = UIColor.red
        HuffStatus.textColor = UIColor.white

    }

    func opened(endTime:String)
    {
        HuffStatus.text = "Open until  "+endTime
        HuffStatus.backgroundColor = UIColor.green
        HuffStatus.textColor = UIColor.white
    }
```

```
    func openContinental(endTime:String)
    {
        HuffStatus.text = "Hot food starts at  "+endTime
        HuffStatus.backgroundColor = UIColor.green
        HuffStatus.textColor = UIColor.white
    }

    func openSnack(endTime:String)
    {
        HuffStatus.text = "Snack Break ends at  "+endTime
        HuffStatus.backgroundColor = UIColor.green
        HuffStatus.textColor = UIColor.white
    }

    //template for hours func uptoNight()
    // {
    //
    //     let time = getTime().hour

    //     switch time
    //     {
    //     case 09...20: opened(endTime: "21") //set time for 09:00 to
20:59
    //     default:closed(endTime: "4:30")
    //     }

    // }
    func weekDay()
    {

        let time = getTime().hour

        switch time
        {
        case 1...6: closed(endTime: " 7:45")
        case 7:
            let minute = getTime().minute
            if minute < 45{
                closed(endTime: " 7:45")
            }
            else{
                opened(endTime:" 10:45")
            }
        case 8...10:
            let minute = getTime().minute
            if time == 10 && minute < 45 {
                closed(endTime: " 11")
            }
            else{
                opened(endTime:" 10:45")
```

```
        }
    case 11...12:
        let minute = getTime().minute
        if time == 11 && minute < 30{
            closed(endTime:" 11:30")
        }
        else{
            opened(endTime: " 1")
        }
    case 13...16:
        let minute = getTime().minute
        if time == 15 && minute >= 45{
            closed(endTime: " 5")
        }
        else{
            openSnack(endTime:" 4:45")
        }
    case 17...18:
        opened(endTime:" 8")

    default:closed(endTime: " tomorrow")
    }

}


func weekend()
{

    let time = getTime().hour

    switch time
    {
    case 1...8: closed(endTime: " 9")
    case 9: openContinental(endTime: " 10")

    case 10...12:
        opened(endTime:" 2:30")

    case 15: closed(endTime: " 4:30")

    case 16...18:
        let minute = getTime().minute
        if time == 16 && minute < 30{
            closed(endTime: " 4:30")
        }
        else{
            opened(endTime:" 7")
        }
```

```
            break

        default:closed(endTime: " tomorrow")
        }

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}

extension UIViewController
{
    @objc func HuffswipeAction(swipe:UISwipeGestureRecognizer)
    {
        switch swipe.direction.rawValue {
        case 2:
            performSegue(withIdentifier: "Huff2Slay", sender: self) //
segue name and location
        default:
            break
        }
    }
}
    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a
little preparation before navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?)
{
        // Get the new view controller using
segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */



//
//  CurtisClass.swift
//  Dineson
//
//  Created by Federico Read on 11/10/17.
//  Copyright © 2017 testing. All rights reserved.
//

import UIKit
```

```
import WebKit

extension Date {    //used for quick view dining times
    func dayOfWeek() -> Int? {
        let calender: Calendar = Calendar.current
        let component: DateComponents = (calender as
NSCalendar).components(.weekday, from: self)
        return component.weekday
    }
}

class CurtisClass: UIViewController {

    @IBOutlet weak var curtisWebView: WKWebView!

    @IBOutlet weak var Curtis_status: UILabel!



    override func viewDidLoad() {
        super.viewDidLoad()

        self.dateCheck()

        // Do any additional setup after loading the view.
        let url = URL(string:"http://denison.cafebonappetit.com/cafe/
curtis-cafe/") //webview link
        let request = URLRequest(url:url!)

        curtisWebView.load(request)

        let RightSwipe = UISwipeGestureRecognizer(target: self,
action: #selector(CurtswipeAction(swipe:)))
        RightSwipe.direction =
UISwipeGestureRecognizerDirection.left   //segues from Curtis to
huffman pages
        self.view.addGestureRecognizer(RightSwipe)


    }

    func dateCheck()
    {
        let today = Date().dayOfWeek()

        if today == 1 || today ==  7
        {
            //print("Sunday")
            self.weekend()
        }
```

```
        else
        {
            //print("Saturday")
            self.weekDay()
        }
    }

    func getTime() -> (hour:Int, minute:Int, second:Int) {
        let currentDateTime = Date()
        let calendar = NSCalendar.current
        let component =
calendar.dateComponents([.hour,.minute,.second], from:
currentDateTime)
        let hour = component.hour
        let minute = component.minute
        let second = component.second
        return (hour!,minute!,second!)
    }


    func closed(endTime:String)
    {
        Curtis_status.text = "Will be CLOSED until"+endTime
        Curtis_status.backgroundColor = UIColor.red
        Curtis_status.textColor = UIColor.white

    }

    func opened(endTime:String)
    {
        Curtis_status.text = "Open until "+endTime
        Curtis_status.backgroundColor = UIColor.green
        Curtis_status.textColor = UIColor.white
    }

    func openContinental(endTime:String)
    {
        Curtis_status.text = "Hot food starts at  "+endTime
        Curtis_status.backgroundColor = UIColor.green
        Curtis_status.textColor = UIColor.white
    }

    func openSnack(endTime:String)
    {
        Curtis_status.text = "Snack Break ends at  "+endTime
        Curtis_status.backgroundColor = UIColor.green
        Curtis_status.textColor = UIColor.white
    }

    //template for hours func uptoNight()
```

```
    // {
    //
    //     let time = getTime().hour

    //     switch time
    //     {
    //     case 09...20: opened(endTime: "21") //set time for 09:00 to
20:59
    //     default:closed(endTime: "4:30")
    //     }

    // }
    func weekDay()
    {

        let time = getTime().hour

        switch time
        {
        case 1...6: closed(endTime: " at 7:45")
        case 7:
            let minute = getTime().minute
            if minute < 45{
                closed(endTime: " at 7:45")
            }
            else{
                opened(endTime:" 10:45")
            }
        case 8...10:
            let minute = getTime().minute
            if time == 10 && minute < 45 {
                closed(endTime: " at 11")
            }
            else{
                opened(endTime:" 10:45")
            }
        case 11...13: opened(endTime: " 2:45")

        case 14:
            let minute = getTime().minute
            if minute >= 45{
                openSnack(endTime: " 4:15")
            }
            else{
                opened(endTime:" 2:45")
            }
        case 15:
            openSnack(endTime: " 4:15")

        case 16:
```

```
        let minute = getTime().minute
        if minute < 15{
            openSnack(endTime: " 4:15")
        }
        else if minute < 30{
            closed(endTime: " at 4:30")
        }
        else{
            opened(endTime:" 7")
        }
    case 17...18:
        opened(endTime:" 7")

    default:closed(endTime: " tomorrow")
    }
}

func weekend()
{
    let time = getTime().hour

    switch time
    {
    case 1...8: closed(endTime: " at 9")
    case 9: openContinental(endTime: " 10")

    case 10...14:
        let minute = getTime().minute
        if time == 14 && minute >= 30{
            closed(endTime: " at 4:30")
        }
        else{
            opened(endTime:" 2:30")
        }

    case 15: closed(endTime: " at 4:30")

    case 16...18:
        let minute = getTime().minute
        if time == 16 && minute < 30{
            closed(endTime: " at 4:30")
        }
        else{
            opened(endTime:" 7")
        }

        break

    default:closed(endTime: " tomorrow")
    }
```

```
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a
little preparation before navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?)
{
        // Get the new view controller using
segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */

}

extension UIViewController
{
    @objc func CurtswipeAction(swipe:UISwipeGestureRecognizer)
    {
        switch swipe.direction.rawValue {
        case 2:
            performSegue(withIdentifier: "Curt2Huff", sender: self) //
segue name and location
        default:
            break
        }
    }
}
//
//  ViewController.swift
//  Dineson
//
//  Created by Federico Read on 10/31/17.
//  Copyright © 2017 testing. All rights reserved.
//

import UIKit


class ViewController: UIViewController {
```

```
   // @IBOutlet var docPopView: UIView!
    @IBOutlet var docuPopView: UIView!

   // @IBOutlet weak var visualEffectVie: UIVisualEffectView!

    @IBOutlet weak var visualEffectView: UIVisualEffectView!

    var effect:UIVisualEffect!


    override func viewDidLoad() {   //Main interface
        super.viewDidLoad()
        effect = visualEffectView.effect    //used for popup
        visualEffectView.effect = nil

        docuPopView.layer.cornerRadius = 5
    }

    func animateIn() {  //popup appear
        self.view.addSubview(docuPopView)
        docuPopView.center = self.view.center

        docuPopView.transform = CGAffineTransform.init(scaleX: 1.3, y:
1.3)
        docuPopView.alpha = 0

        UIView.animate(withDuration: 0.4){
            self.visualEffectView.effect = self.effect
            self.docuPopView.alpha = 1
            self.docuPopView.transform = CGAffineTransform.identity

        }
    }

    func animateOut(){  //popup disapear
        UIView.animate(withDuration: 0.3, animations: {
            self.docuPopView.transform =
CGAffineTransform.init(scaleX: 1.3, y: 1.3)
            self.docuPopView.alpha = 0

            self.visualEffectView.effect = nil
        }) { (success:Bool) in
            self.docuPopView.removeFromSuperview()
        }
    }

    @IBAction func popUpLoad(_ sender: AnyObject) { //on help button
press
        animateIn()
    }
```

```
    //@IBAction func popUpLoad(_ sender: AnyObject) {
    //    animateIn()
    //}
    @IBAction func popUpClose(_ sender: AnyObject) {    //on selecting
"back"
        animateOut()
    }

    //@IBAction func popUpClose(_ sender: AnyObject) {
   //     animateOut()
   // }



    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}

//
//  SweetsTableViewController.swift
//  DatabaseDine
//
//  Created by Jacob on 12/3/17.
//  Copyright © 2017 Jacob. All rights reserved.
//

import UIKit
import FirebaseDatabase
import FirebaseAuth

class SweetsTableViewController: UITableViewController {

    var dbRef:DatabaseReference!  //To be used for sending comments
    var sweets = [Sweet]()    //Holds string of characters for
comments

    override func viewDidLoad() {
        super.viewDidLoad()
        dbRef = Database.database().reference().child("Curtis")    //
initial value (top) is dining hall name

        startObservingDB()                                          //
Read current data to mobile device

    }
```

```
    func startObservingDB () {//Read in data from Firebase and display
on table

        dbRef.observe(.value, with: { (snapshot:DataSnapshot) in
            var newSweets = [Sweet]()

            for sweet in snapshot.children {
                let sweetObject = Sweet(snapshot: sweet as!
DataSnapshot)
                newSweets.append(sweetObject)
            }

            self.sweets = newSweets
            self.tableView.reloadData()

        })
    }

    @IBAction func addSweet(_ sender: Any) {//on "+" button press
        //First user prompt (comment)

        let sweetAlert = UIAlertController(title:"New Comment",
message: "Enter your Comment", preferredStyle: .alert)
        sweetAlert.addTextField { (textField:UITextField) in
            textField.placeholder = "Your comment"
        }
        sweetAlert.addAction(UIAlertAction(title: "Send",
style: .default, handler: { (action:UIAlertAction) in
            let sweetContent = sweetAlert.textFields?.first?.text
            //Second user prompt (Name)
            let sweetAlert2 = UIAlertController(title:"Name", message:
"Enter your Name", preferredStyle: .alert)
            sweetAlert2.addTextField { (textField:UITextField) in
                textField.placeholder = "Your name"
            }
            sweetAlert2.addAction(UIAlertAction(title: "Send",
style: .default, handler: { (action:UIAlertAction) in
                //update reference in database
                if let sweetName = sweetAlert2.textFields?.first?.text
{
                    let sweet = Sweet(content: sweetContent!,
addedByUser: sweetName)

                    let sweetRef =
self.dbRef.child(sweetName.lowercased())

                    sweetRef.setValue(sweet.toAnyObject())
                }
            }))
```

```
            self.present(sweetAlert2, animated: true, completion: nil)
        }))
        self.present(sweetAlert, animated: true, completion: nil)

    }



    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    override func tableView(_ tableView: UITableView,
numberOfRowsInSection section: Int) -> Int {
        return sweets.count
    }

    //Establish table view, vertically inputing comments (older
towards the bottom)
    override func tableView(_ tableView: UITableView, cellForRowAt
indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier:
"Cell", for: indexPath)

        let sweet = sweets[indexPath.row]

        cell.textLabel?.text = sweet.content
        cell.detailTextLabel?.text = sweet.addedByUser

        return cell
    }




}
//
//  Sweet.swift
//  DatabaseDine
//
//  Created by Jacob on 12/3/17.
//  Copyright © 2017 Jacob. All rights reserved.
//

import Foundation
import FirebaseDatabase
```

```
struct Sweet {

    let key:String! //Title name (slayter, Name)
    let content:String! //Child of key
    let addedByUser:String! //Child of key and key name
    let itemRef:DatabaseReference?  //location in database

    init (content:String, addedByUser:String, key:String = "") {
        self.key = key
        self.content = content
        self.addedByUser = addedByUser
        self.itemRef = nil
    }

    init (snapshot:DataSnapshot) {  //initialize location
        key = snapshot.key
        itemRef = snapshot.ref

        let snapshotValue = snapshot.value as? NSDictionary
        if let sweetContent = snapshotValue!["content"] as? String {
            content = sweetContent
        }else {
            content = ""
        }

        let snapshotValue2 = snapshot.value as? NSDictionary
        if let sweetUser = snapshotValue2!["addedByUser"] as? String {
            addedByUser = sweetUser
        }else {
            addedByUser = ""
        }
    }

    func toAnyObject() -> Any { //create comment with timestamp
        return ["timestamp": ServerValue.timestamp(), "content":
content, "addedByUser": addedByUser]
    }

}
//
//  HuffmanTableViewController.swift
//  DatabaseDine
//
//  Created by Jacob on 12/4/17.
//  Copyright © 2017 Jacob. All rights reserved.
//

import UIKit
import FirebaseDatabase
import FirebaseAuth
```

```
class HuffmanTableViewController: UITableViewController {

    var dbRef:DatabaseReference!  //To be used for sending comments
    var sweets = [Sweet]()     //Holds string of characters for
comments

    override func viewDidLoad() {
        super.viewDidLoad()
        dbRef = Database.database().reference().child("Huffman")    //
initial value (top) is dining hall name
        startObservingDB()                                          //
Read current data to mobile device
    }


    func startObservingDB () {//Read in data from Firebase and display
on table
        dbRef.observe(.value, with: { (snapshot:DataSnapshot) in
            var newSweets = [Sweet]()

            for sweet in snapshot.children {
                let sweetObject = Sweet(snapshot: sweet as!
DataSnapshot)
                newSweets.append(sweetObject)
            }

            self.sweets = newSweets
            self.tableView.reloadData()

        })
    }

    @IBAction func addSweet(_ sender: Any) {    //on "+" button press
        //First user prompt (comment)
        let sweetAlert = UIAlertController(title:"New Comment",
message: "Enter your Comment", preferredStyle: .alert)
        sweetAlert.addTextField { (textField:UITextField) in
            textField.placeholder = "Your comment"
        }
        sweetAlert.addAction(UIAlertAction(title: "Send",
style: .default, handler: { (action:UIAlertAction) in
            let sweetContent = sweetAlert.textFields?.first?.text
            //Second user prompt (Name)
            let sweetAlert2 = UIAlertController(title:"Name", message:
"Enter your Name", preferredStyle: .alert)
            sweetAlert2.addTextField { (textField:UITextField) in
                textField.placeholder = "Your name"
            }
            sweetAlert2.addAction(UIAlertAction(title: "Send",
```

```
style: .default, handler: { (action:UIAlertAction) in
                //update reference in database
                if let sweetName = sweetAlert2.textFields?.first?.text
{
                    let sweet = Sweet(content: sweetContent!,
addedByUser: sweetName)

                    let sweetRef =
self.dbRef.child(sweetName.lowercased())

                    sweetRef.setValue(sweet.toAnyObject())
                }
            }))
            self.present(sweetAlert2, animated: true, completion: nil)
        }))
        self.present(sweetAlert, animated: true, completion: nil)

    }



    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    override func tableView(_ tableView: UITableView,
numberOfRowsInSection section: Int) -> Int {
        return sweets.count
    }

        //Establish table view, vertically inputing comments (older
towards the bottom)
    override func tableView(_ tableView: UITableView, cellForRowAt
indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier:
"CellH", for: indexPath)

        let sweet = sweets[indexPath.row]

        cell.textLabel?.text = sweet.content
        cell.detailTextLabel?.text = sweet.addedByUser

        return cell
    }
```

```
}
//
//  SlayterTableViewController.swift
//  DatabaseDine
//
//  Created by Jacob on 12/4/17.
//  Copyright © 2017 Jacob. All rights reserved.
//

import UIKit
import FirebaseDatabase
import FirebaseAuth

class SlayterTableViewController: UITableViewController {

    var dbRef:DatabaseReference!    //To be used for sending comments
    var sweets = [Sweet]()          //Holds string of characters for
comments

    override func viewDidLoad() {
        super.viewDidLoad()
        dbRef = Database.database().reference().child("Slayter")    //
initial value (top) is dining hall name
        startObservingDB()                                          //
Read current data to mobile device
    }


    func startObservingDB () {          //Read in data from Firebase
and display on table
        dbRef.observe(.value, with: { (snapshot:DataSnapshot) in
            var newSweets = [Sweet]()

            for sweet in snapshot.children {
                let sweetObject = Sweet(snapshot: sweet as!
DataSnapshot)
                newSweets.append(sweetObject)
            }

            self.sweets = newSweets
            self.tableView.reloadData()

        })
    }

    @IBAction func addSweet(_ sender: Any) {    //on "+" button press
        //First user prompt (comment)
        let sweetAlert = UIAlertController(title:"New Comment",
message: "Enter your Comment", preferredStyle: .alert)
```

```
        sweetAlert.addTextField { (textField:UITextField) in
            textField.placeholder = "Your comment"
        }
        sweetAlert.addAction(UIAlertAction(title: "Send",
style: .default, handler: { (action:UIAlertAction) in
            let sweetContent = sweetAlert.textFields?.first?.text
            //Second user prompt (Name)
            let sweetAlert2 = UIAlertController(title:"Name", message:
"Enter your Name", preferredStyle: .alert)
            sweetAlert2.addTextField { (textField:UITextField) in
                textField.placeholder = "Your name"
            }
            sweetAlert2.addAction(UIAlertAction(title: "Send",
style: .default, handler: { (action:UIAlertAction) in
                //update reference in database
                if let sweetName = sweetAlert2.textFields?.first?.text
{
                    let sweet = Sweet(content: sweetContent!,
addedByUser: sweetName)

                    let sweetRef =
self.dbRef.child(sweetName.lowercased())

                    sweetRef.setValue(sweet.toAnyObject())
                }
            }))
            self.present(sweetAlert2, animated: true, completion: nil)
        }))
        self.present(sweetAlert, animated: true, completion: nil)

    }



    // MARK: - Table view data source

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    override func tableView(_ tableView: UITableView,
numberOfRowsInSection section: Int) -> Int {
        return sweets.count
    }

    //Establish table view, vertically inputing comments (older
towards the bottom)
    override func tableView(_ tableView: UITableView, cellForRowAt
indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier:
```

```
"CellS", for: indexPath)

        let sweet = sweets[indexPath.row]

        cell.textLabel?.text = sweet.content
        cell.detailTextLabel?.text = sweet.addedByUser

        return cell
    }




}
```