iTeam

**Pupil**

Software Design Document

# Beau Miller, Jordan Alnaser, Roger Campbell, Brian Lee

# 

# 05/05/2017

# 

# TABLE OF CONTENTS

1. INTRODUCTION  
 1.1 Purpose  
 1.2 Scope  
 1.3 Overview  
 1.4 Reference Material  
 1.5 Definitions and Acronyms

2. SYSTEM OVERVIEW

3. SYSTEM ARCHITECTURE  
 3.1 Architectural Design  
 3.2 System Boundaries  
 3.3 Architecture Interaction

4. DATA DESIGN  
 4.1 Data Description  
 4.2 Low Level Data Design: data.swift

5. COMPONENT DESIGN: OBJECTS OVERVIEW  
 5.1 View Controller

5.2 Course Object

5.3 Assignment Object

5.4 Timer Object

6. HUMAN INTERFACE DESIGN  
 6.1 Overview of User Interface  
 6.2 Screen Images  
 6.3 Screen Objects and Actions

7. Production Details

7.1 Expected Total Lines of Code

7.2 IDE, Compilers, Configuration Management, Test Tools

7.3 Target System

7.4 Open Source Software and Licensing

8. APPENDICES

**1. INTRODUCTION**

**1.1 Purpose**  
 This software design document describes the architecture and system design of the Pupil App for those it may concern. In congruence with the guidelines laid down by IEEE.  
**1.2 Scope**  
 The objective is to create an easy way for student to keep track of their assignments in an effort to be more productive. This will be achieved by creating an iOS app targeting iPhones 5,5c,5s,SE,6+,6s,6s+,7: 80% of iPhones, which are able to run iOS 10.  
**1.3 Overview**  
 This document will cover the interface and object design down to a low level of all elements in the application.  
**1.4 Reference Material**  
 This document was made with the influence of IEEE’s software design suggestions and the requirements of professor Pushkar Ogale.  
**1.5 Definitions and Acronyms**  
 **iOS**: Apple's Operating System that is used on their Mobile Devices  
 **UI**: User Interface   
 **Swift**: Apple's Proprietary Programming Languages

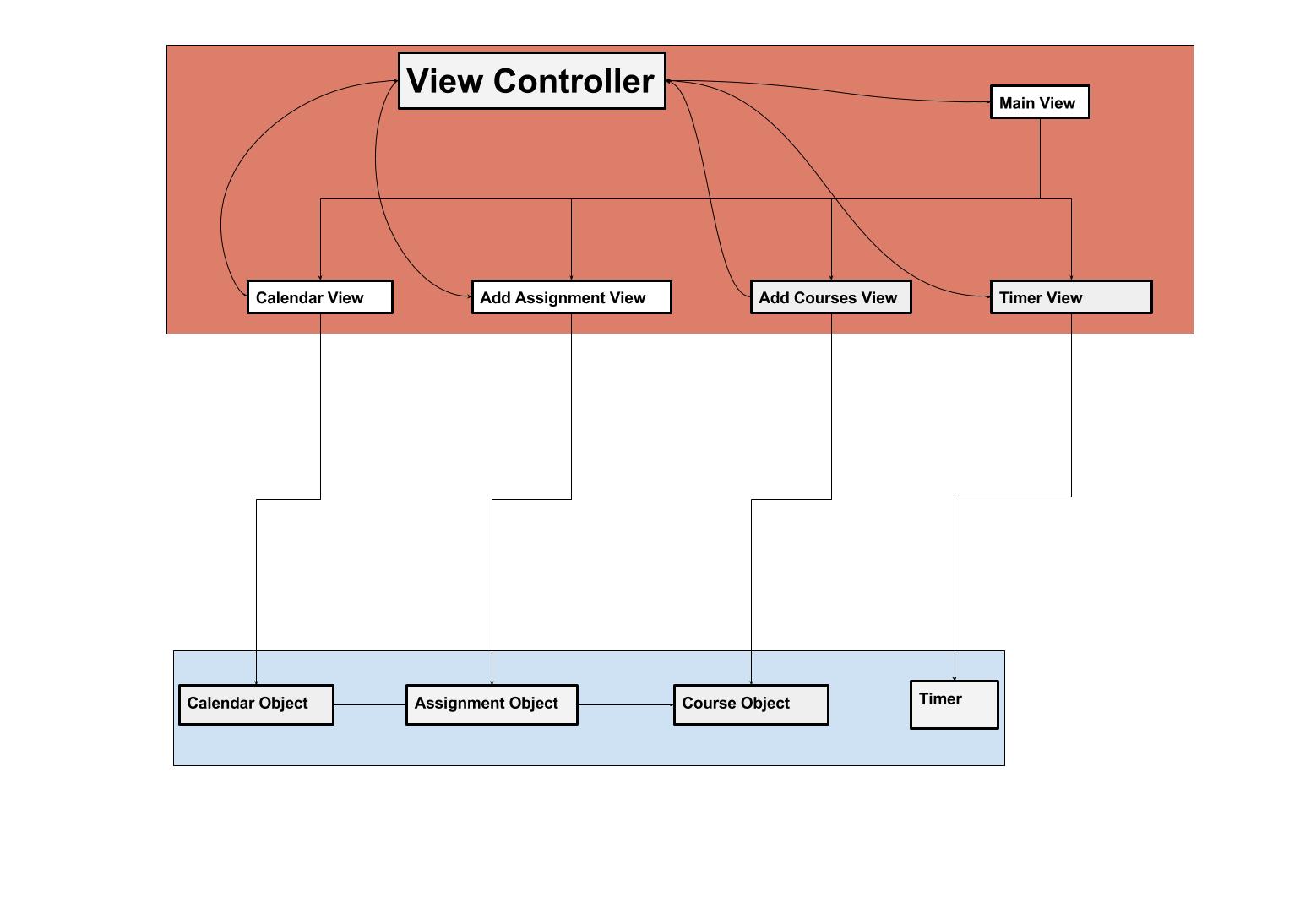
**JTAppleCalendar**: Open source calendar library licenced for this project under the MIT licence. Created by patch the code.

**2. SYSTEM OVERVIEW**

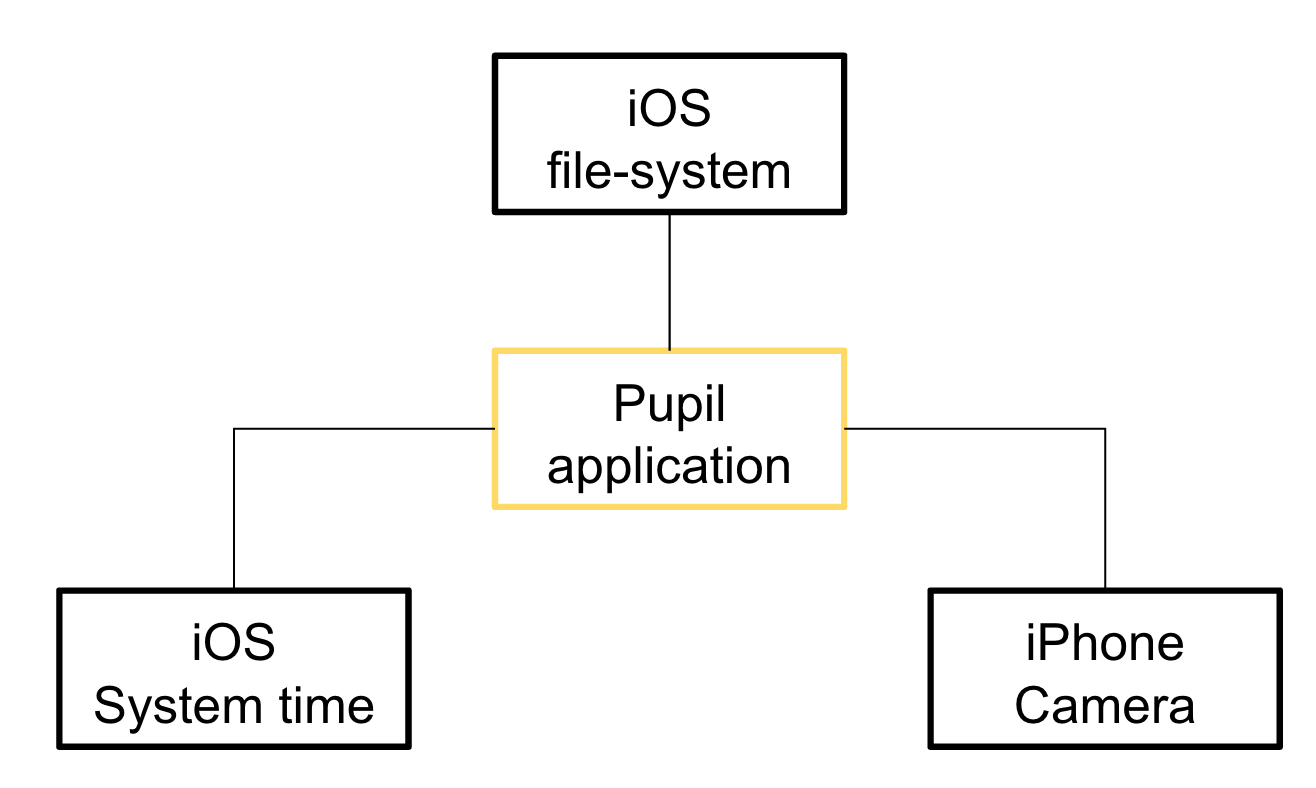
Pupil is a stand alone student assistant. The system will be for iPhones running iOS 10 and will incorporate a calendar system that will store assignments and dates, a grade calculator, a syllabi storing system, and a study timer.

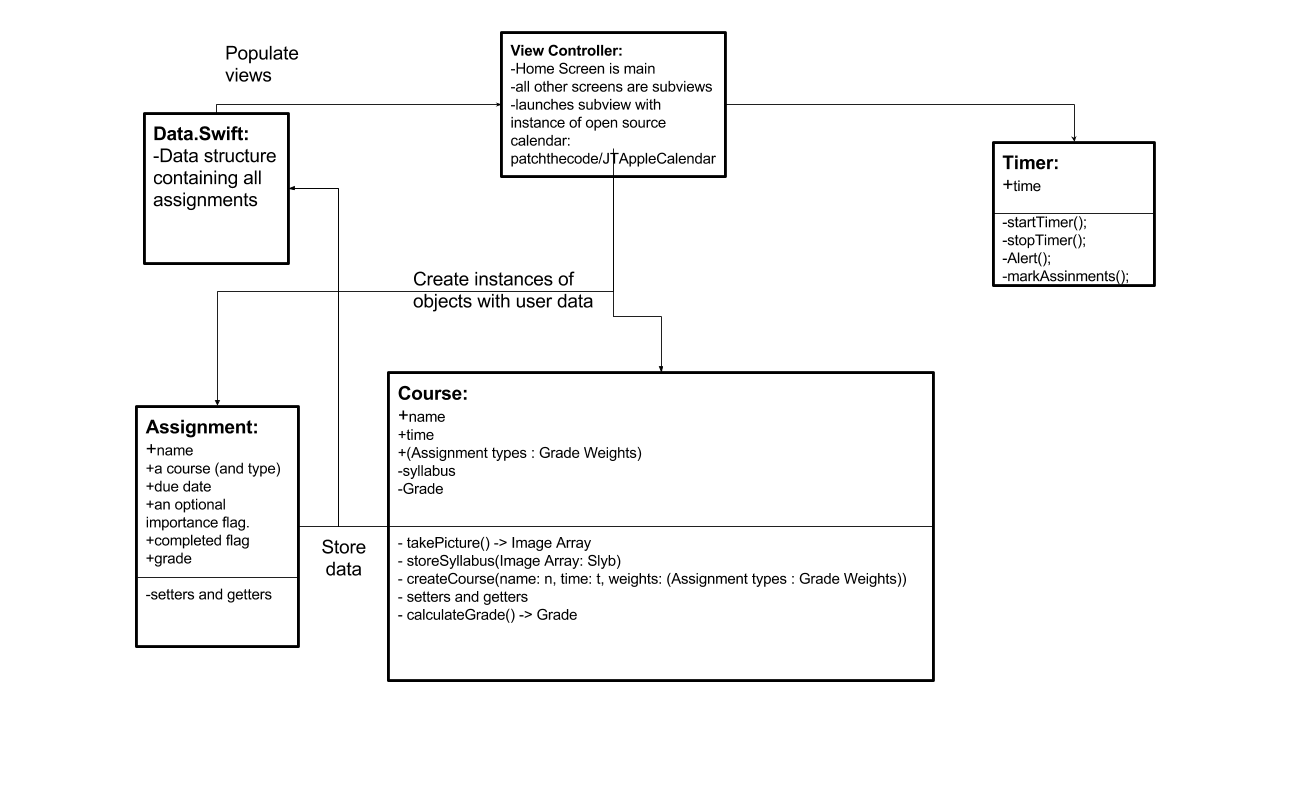
Contextually, it will need to interact with the iPhone system time, system file storage, and system interface.

**3. SYSTEM ARCHITECTURE**

**3.1 Architectural Design**  


**3.2 System Boundaries:**  
Within pupil there are many systems that depend upon each other to function properly such as the calendar system which relies upon the assignment and reminder systems  
There is a grading and GPA system which rely on the data storage system of the user’s IOS device  
There is the IOS system and hardware systems of the iphone which rely on Pupil to safely function



**3.3 Architecture Interaction**  
  
  
**4. DATA DESIGN**

**4.1 Data Description**

The information input by users such as courses and assignments is stored in the memory of the users device via the data.swift file.  
**4.2 Low Level Data Design: data.swift**  
Courses: stored as a dictionary of Courses [String: Course]  
Syllabus: stored as an array of images  
Assignments: stored as a dictionary [String: Assignment] with the assignment key being (assignmentName + courseName), corresponding to each assignment in the dictionary.

**5. COMPONENT DESIGN: OBJECTS OVERVIEW**

**5.1 View Controller**

View Controller

IBOutlets from the GUI:

@IBOutlet dateLabel: UILabel!

@IBOutlet mainTableView: UITableView!

@IBOutlet plusButton: UIImageView!

Attributes:

let addHomeWorkButton:UIButton()

let addCourseButton:UIButton()

let cellReuseIdentifier:String

Methods:

func viewDidLoad()  
 This function is implicitly called when the view is loaded, it allows for further

objects and subviews to start getting initialized inside the view after it is loaded.  
  
func plusButtonPressed(tapGestureRecognizer: UITapGestureRecognizer)  
 This is triggered when the user taps the plus button, it then performs the opening

animation displays the options under the plus button options buttons.  
  
func dissmissPlusButtonOptions(\_ addHomeWorkButton:UIButton)  
 This is triggered with the plus button is opened and tapped to close, it performs

necessary cleanup to dismiss the options and close the plus button.

func plusButtonOptions(motherView: UIImageView, \_ addHomeWorkButton:UIButton))

This is triggered by the plusButtonPressed Method, and is used to load the subview options under the plus button, as well as set their methods to be triggered when they're pressed.

func addCourseView(\_ button:UIButton), addAssignmentView(\_ button:UIButton)  
 This is triggered when the user pressed the option to add course under the plus

button. It loads a subview of the main view, and it plays its animation.   
  
func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath)  
 This is implicitly called by the Table view Object, when it needs to fill a given cell

at a certain index with data to be displayed in the table.

**5.2 Course Object**

Attributes:

private var name = String()

private var syllabus = [UIImage]()

private var grade:Double

Methods:

func addSyllabus(\_ syllabusArray: [UIImage])

This function is used to trigger the camera app and allow the user to take images of his syllabus that will be stored in the private field syllabus under the calling course

func addGrade(\_ grade:Double)

This function is used to add the grade that is passed in and setting the private field in the calling course object to it.

func getName() -> String

This is a getter function that returns the name of the calling course object.

**5.3 Assignment Object**

Attributes:

private var name:String

private var course:Course

private var dueDate = Date()

private var importance = String()

private var completedFlag:Bool

private var grade:Double

private var tag:Int

Methods:

func editAssignment(\_ name:String, \_ dueDate:Date, \_ importance:String)

This function accepts a name and a due with an importance string, and it looks for that assignment in the dictionary of assignments and edits its content.

func getID() -> String

This is a getter function that returns the id of this assignment which is a combination of the assignment name and the course id

func getName() ->String

This is a getter function that returns the name of the assignment

func getCourseName() -> String

This is a getter function that returns the name of the course that this assignment is associated with.

func getDueDate() -> Date

This is a getter functions that returns the due date that this assignment is assigned to.

**5.4 Timer Object**

attributes:

private var intialTime:Int

private var timerUI:UITimer()

private var timerAlert:UIAlert()

Methods:

func setInitialTime(\_ time:Int)

This function starts the timer with the passed in time parameter, and it runs it down.

func displayTime()

This function monitors the status of the initial time, and it displays the result of it to a graphical UI element.

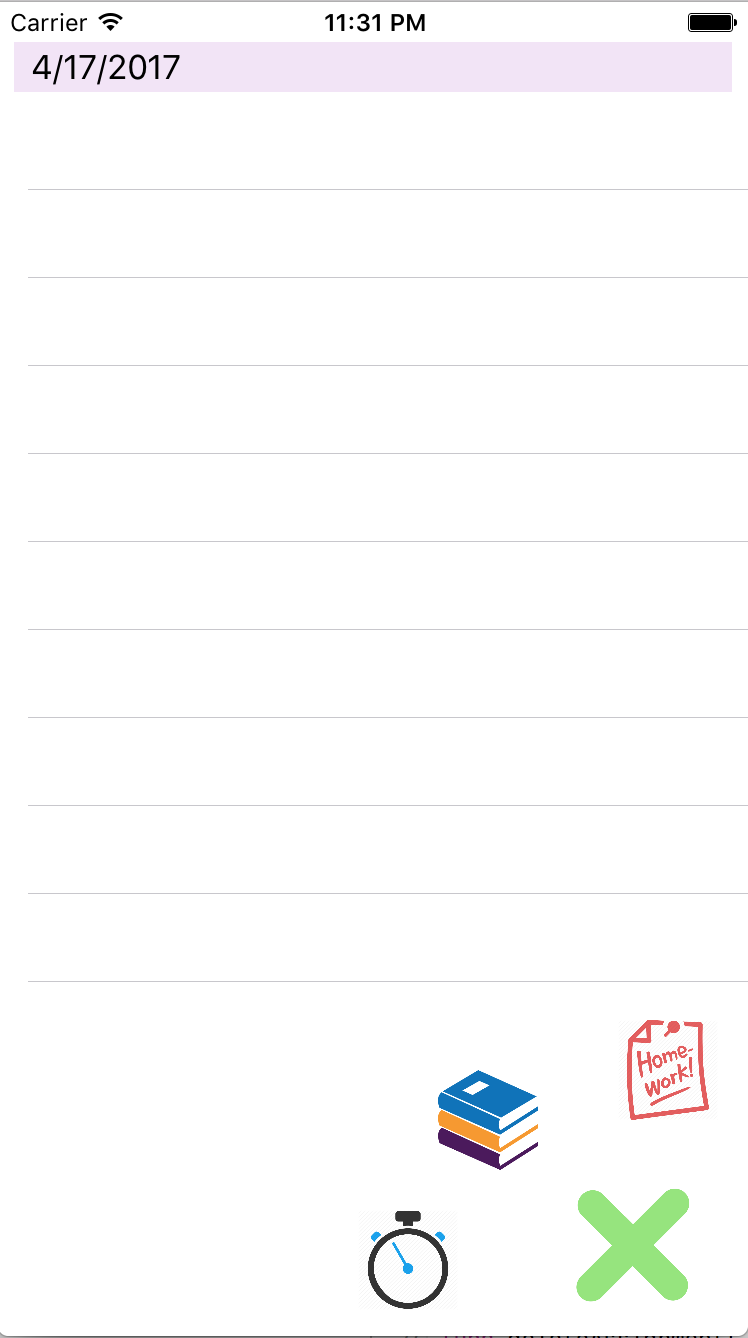
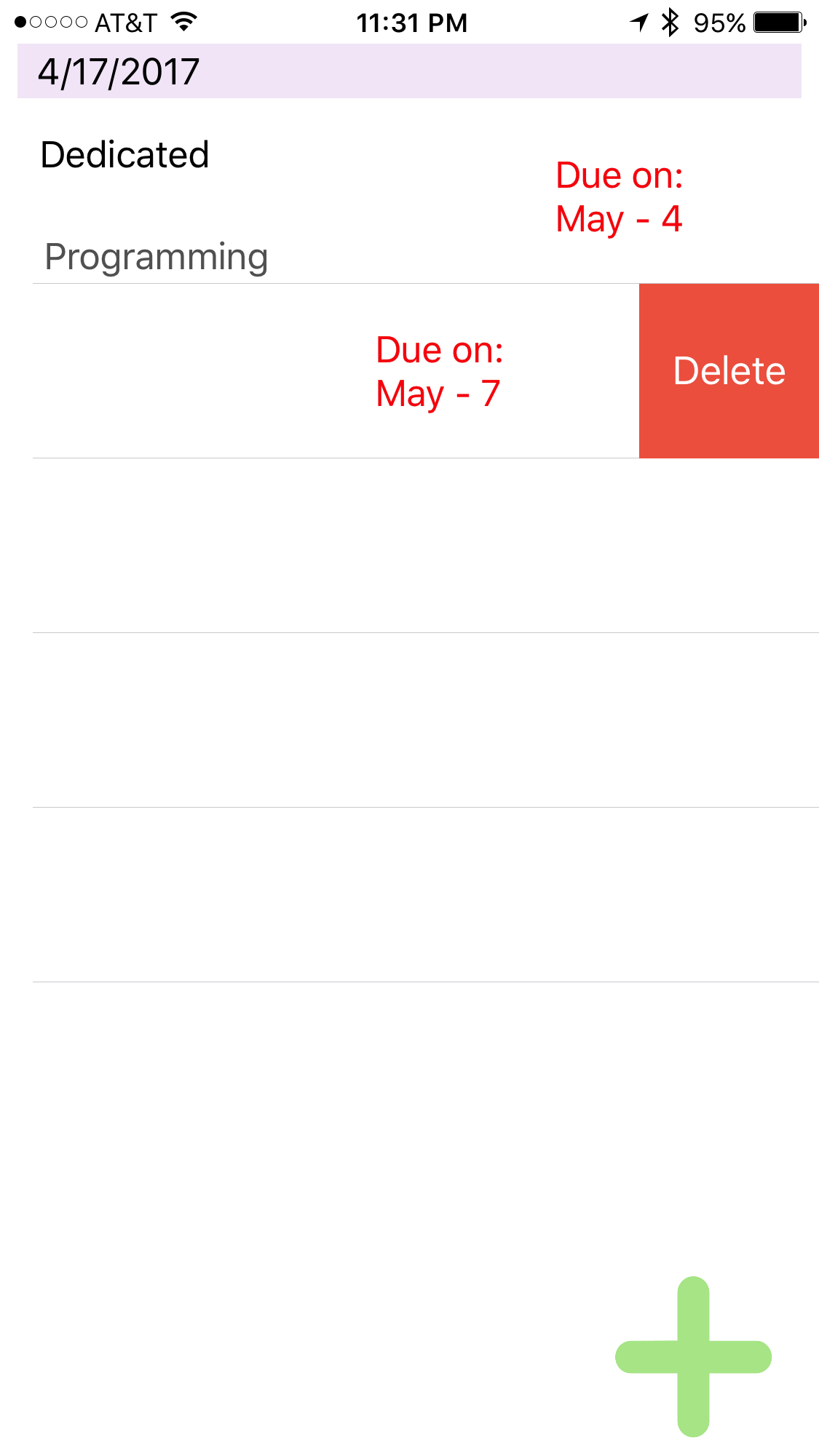
func timerStopped()

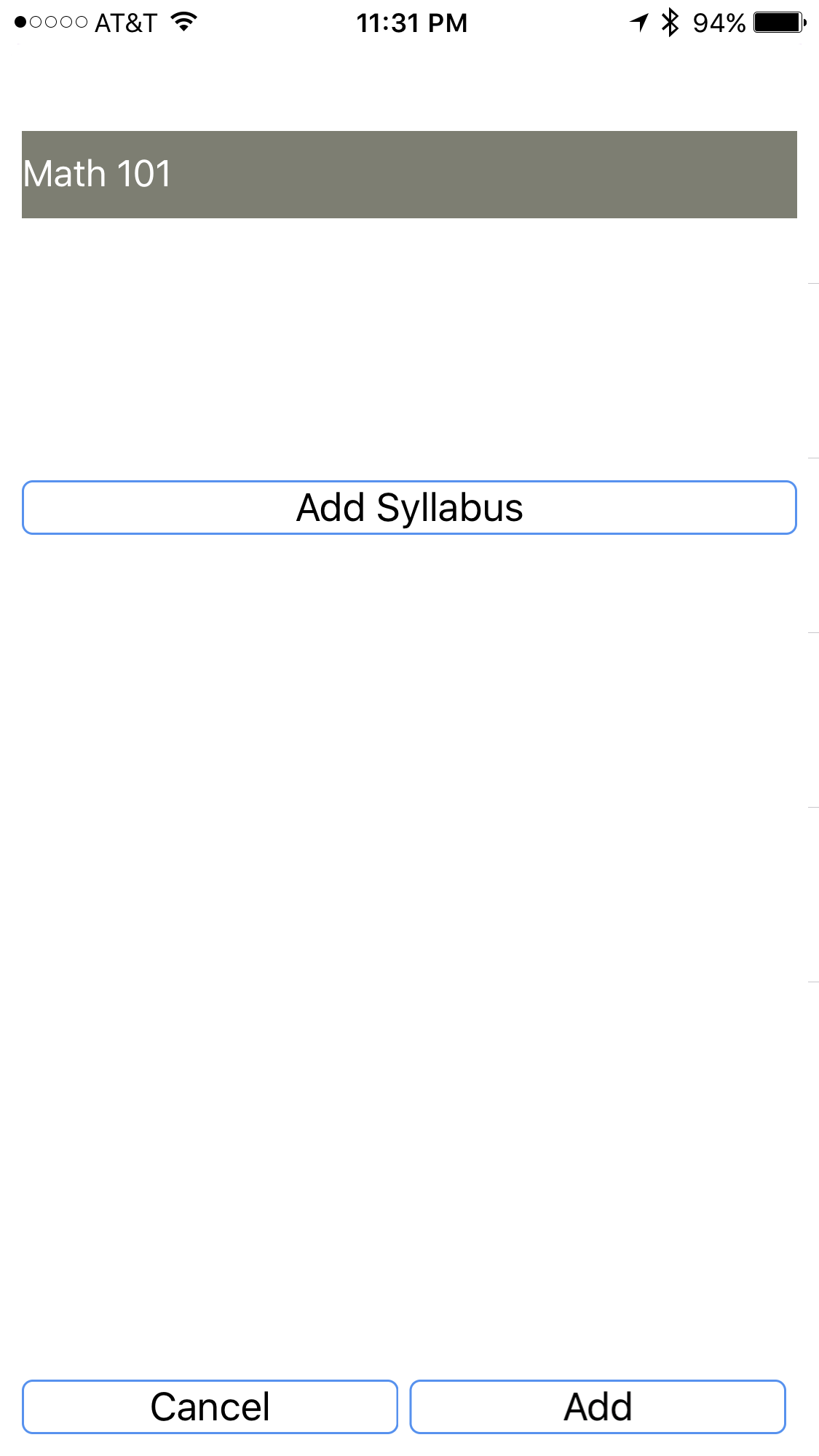
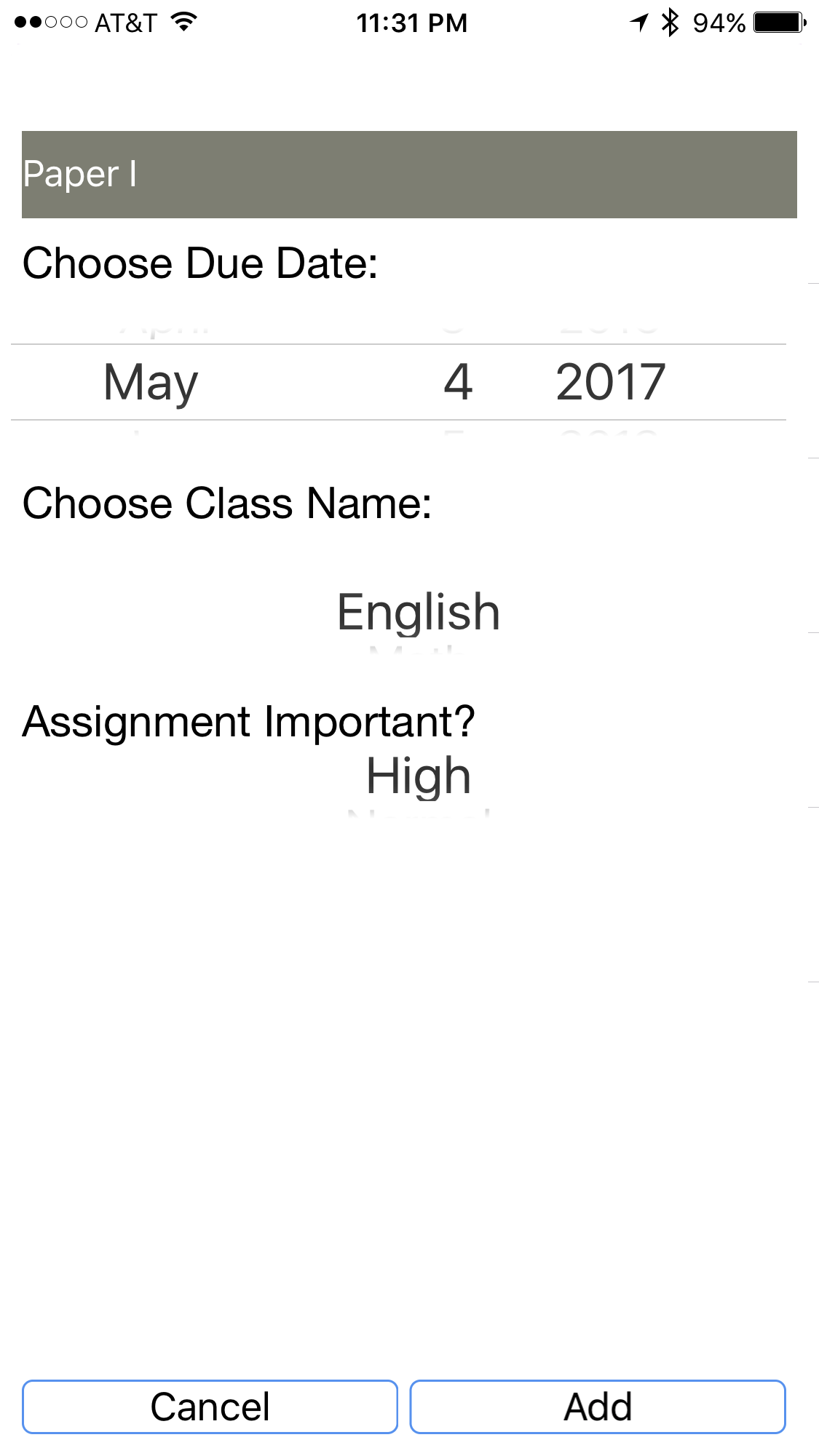
This is a function that is triggered when the timer object has finished running, function displays an alert to the user and tells him that the time has run out.

**6. HUMAN INTERFACE DESIGN**

**6.1 Overview of User Interface**

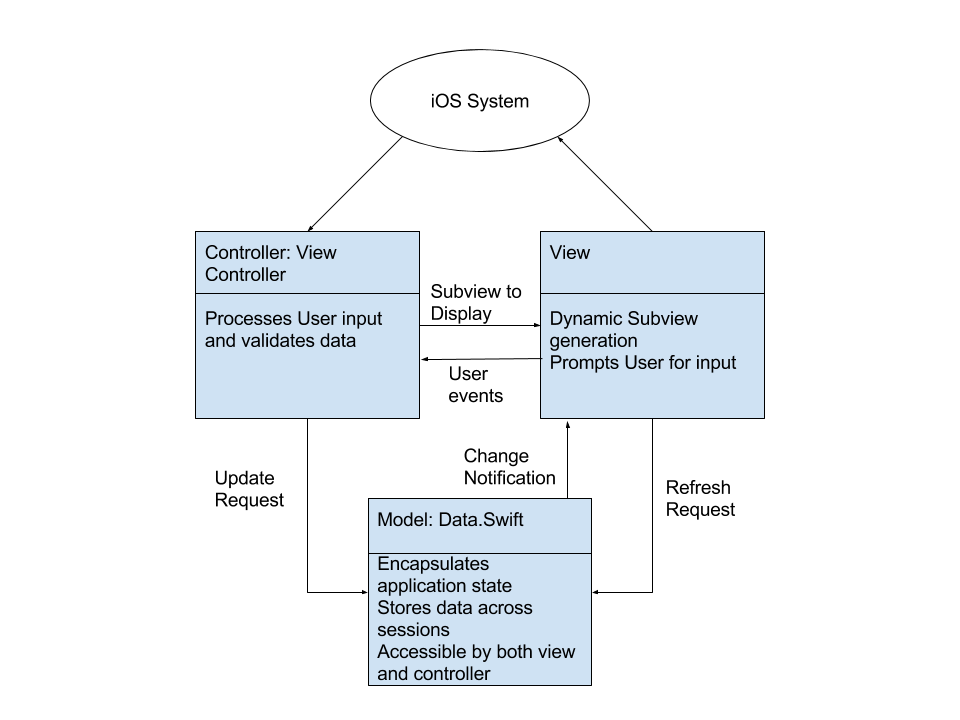
The system allows the user to track the assignments for all of their classes and provides a timer for studying. The user is able to navigate from the main screen to add an assignment or course which has a due date. The user is also able to use the timer function

**6.2 Screen Images**  




**6.3 Screen Objects and Actions**

Model-View-Controller diagram:



**7. PRODUCTION DETAILS**

**7.1 Expected Total Lines of Code:**

240+280+160+30+50+40+80+100+50+100+100 ~=~ 1500 total, not including GUI

Because of Xcode’s graphical GUI creator, a lot of code is created in the background.

**7.2 IDE, Compilers, Configuration Management, Test Tools**

**IDE:** Xcode

**Compilers:** Swift compiler for Windows, REPL.it

**Configuration Management System(Version Control):** GitHub, Xcode built-in git

**Test Tools:** manual testing, xcode iOS simulator

**7.3 Target System**

**Target Execution Systems:** iPhones 5,5c,5s,SE,6+,6s,6s+,7: 80% of iPhones

**Minimum System Requirements:** Compatibility with IOS 10, 2MB of free memory

**Libraries:** Swift’s standard library, JTAppleCalendar

**Support Software:** JTAppleCalendar

**Database:** stored on the memory of the user's device

**Version Control:** Github

**7.4 Open Source Software and Licensing**

**Swift:** Open source programming language which Pupil was created in.

**Licensing:** We plan to put our app on the app store so we have to comply with Apple’s guidelines and have our app approved before it goes live.

**patchthecode/JTAppleCalendar:** Open source iOS calendar library

**Licensing:** patchthecode/JTAppleCalendar is licensed under the MIT License:

Copyright (c) 2016 JayT <patchthecode@gmail.com>  
  
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:  
  
The above copyright notice and this permission notice shall be included in  
all copies or substantial portions of the Software.  
  
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
THE SOFTWARE.

**8. APPENDICES**

Pupil project github: <https://github.com/jordanalnaser/Pupil>

JTAppleCalendar github: <https://github.com/patchthecode/JTAppleCalendar>