**ĐẠI HỌC QUỐC GIA TP.HCM  
TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN LẬP TRÌNH TRỰC QUAN

ĐỒ ÁN PHẦN MỀM LẬP LỊCH

**GV Hướng Dẫn: Huỳnh Tuấn Anh**

Nhóm TH:

**Nguyễn Sang Viết (N.Trưởng)**

**Hoàng Đức Minh**

**Lò Thế Vĩ**

TP. HỒ CHÍ MINH, NĂM 2018

Mục Lục

[Danh mục bảng biểu 4](#_Toc517062388)

[Chương 1. Hiện trạng và yêu cầu từ thực tế 5](#_Toc517062389)

[1.1. Hiện trạng vấn đề 5](#_Toc517062390)

[1.1.1. Đặt Vấn đề 5](#_Toc517062391)

[1.1.2. Phương hướng giải quyết 5](#_Toc517062392)

[1.2. Hiện trạng cơ sở vật chất và con người 5](#_Toc517062393)

[1.2.1. Tin học 5](#_Toc517062394)

[1.2.2. Con người 5](#_Toc517062395)

[1.3. Yêu cầu sơ bộ về phần mềm 5](#_Toc517062396)

[1.3.1. Tin học 5](#_Toc517062397)

[1.3.2. Con người 5](#_Toc517062398)

[Chương 2. Phân tích yêu cầu phần mềm và mô hình hóa 6](#_Toc517062399)

[2.1. Yêu cầu phần mềm 6](#_Toc517062401)

[2.1.1. Yêu cầu chức năng 6](#_Toc517062402)

[2.1.2. Yêu cầu phi chức năng 7](#_Toc517062403)

[2.1.3. Bảng FURPS 7](#_Toc517062404)

[2.2. Mô hình hóa 8](#_Toc517062405)

[2.2.1. Các trường hợp sử dụng thông thường 8](#_Toc517062406)

[2.2.2. Mô hình thực thể - mối quan hệ 8](#_Toc517062407)

[Chương 3. Thiết kế 10](#_Toc517062408)

[3.1. Thiết kế kiến trúc phần mềm 10](#_Toc517062410)

[3.1.1. Ưu điểm 11](#_Toc517062411)

[3.2. Thiết kế dữ liệu 11](#_Toc517062412)

[3.3. Thiết kế giao diện và thành phần xử lí của giao diện 12](#_Toc517062413)

[3.3.1. Tổng thể về giao diện hệ thống 12](#_Toc517062414)

[3.3.2. Giao diện Trang Chủ (Homepage) 13](#_Toc517062415)

[14](#_Toc517062416)

[3.4. Thiết kế xử lý 14](#_Toc517062417)

[3.4.1. Events 14](#_Toc517062418)

[3.4.2. Hide và Time 14](#_Toc517062419)

[Chương 4. Cài Đặt Phần Mềm 15](#_Toc517062420)

[4.1. Tổng quan về công nghệ sử dụng 15](#_Toc517062422)

[4.1.1. Ngôn ngữ lập trình C# 15](#_Toc517062423)

[4.1.2. Json 16](#_Toc517062424)

[4.1.3. Một số framework được sử dụng 17](#_Toc517062425)

[Chương 5. Kiểm Thử 18](#_Toc517062426)

[5.1. Unit Test 18](#_Toc517062428)

[5.2. Integration Test 18](#_Toc517062429)

[5.3. System Test 18](#_Toc517062430)

[5.3.1. Phương pháp kiểm tra 18](#_Toc517062431)

[Chương 6. Tổng Kết 19](#_Toc517062432)

[6.1. Tổng Kết 19](#_Toc517062434)

[6.2. Nhận Xét & Đánh Giá 19](#_Toc517062435)

[6.3. Phân Rã Công Việc 19](#_Toc517062436)

# Danh mục bảng biểu

[Hình 1: Use case Events 11](#_Toc485138611)

[Hình 2: Use case Hide và Time 15](#_Toc485138616)

[Hình 3: Sơ đồ giao diện tổng thể 16](#_Toc485138617)

[Hình 4: Giao diện Evevt 16](#_Toc485138618)

[Hình 5: Giao diện Hide và Time 17](#_Toc485138619)

[Hình 10: Json 17](#_Toc485138620)

[Hình 11: Convert Json](#_Toc485138621) [32](#_Toc485138636)

# Hiện trạng và yêu cầu từ thực tế

## Hiện trạng vấn đề

### Đặt Vấn đề

Bạn có thường cảm thấy choáng ngợp bởi số lượng công việc mà bạn có, hay bạn thường trễ lịch hẹn? Hoặc có bao giờ bạn quên làm điều gì đó quan trọng trong ngày?

### Phương hướng giải quyết

## Hiện trạng cơ sở vật chất và con người

### Tin học

Hiện nay, việc sở hữu 1 chiếc máy tính cá nhân hay điện thoại di động là rất phổ biến. Hầu như bất cứ lúc nào, du khách cũng mang theo bên mình 1 chiếc smartphone và laptop.

### Con người

Nếu bạn là người hay quên và thích “nước đến chân mới nhảy” thì Phần mềm thật sự là ứng dụng hoàn hảo dành riêng cho bạn. Bạn sẽ chẳng cần phải nhớ gì cả, ứng dụng của Nhóm sẽ nhớ và nhắc giùm bạn . Ứng dụng sẽ liên tục nhắc nhở những nhiệm vụ tới hạn cho đến khi bạn hoàn thành chúng.

Nhanh chóng, đơn giản và hiệu quả.

## Yêu cầu sơ bộ về phần mềm

### Tin học

Phần mềm chạy có thể chạy được trên các máy tính chạy hệ điều hành Window, và có .Net Framework 4.5 trở lên.

### Con người

* Ngôn ngữ sử dụng: Tiếng Anh
* Giao diện: đẹp, quen thuộc và dễ sử dụng,
* Độ phức tạp: trung bình, dễ học.

# Phân tích yêu cầu phần mềm và mô hình hóa



## Yêu cầu phần mềm

### Yêu cầu chức năng

* Bảng tổng hợp và định dạng các yêu cầu:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Định danh*** | | ***Độ Ưu tiên*** | ***Mô tả Yêu Cầu*** |
| YC\_1 | Nhắc Nhở | 5 | Liên tục nhắc nhở mỗi 1, 5, 10, 15, 30 hay 60 phút đến khi hết thời gian của Task. |
| YC\_2 | TKB | 5 | Cung cấp Lịch trưc quan cho người dùng. |
| YC\_3 | Lập Kế Hoạch | 4 | Người dùng tạo bản kế hoạch theo ngày. |
| YC\_4 | Hiện thị Thông báo theo ngày | 4 | Hệ thống tự động Thông báo nhắc nhở công việc cho người dùng |
| YC\_6 | Hướng dẫn | 4 | Cung cấp thông tin, hướng dẫn sử dụng. |
| YC\_9 | Description | 2 | Người dùng có thể thêm ghi chú và lưu lại. |
| YC\_10 | HIDE | 1 | Chế độ ẩn/Hiện |

Bảng 1: Bảng tổng hợp yêu cầu chức năng

### Bảng FURPS

|  |  |
| --- | --- |
| ***Tiêu chí chất lượng*** | ***Mô tả*** |
| Functionality | * Phần mềm hướng tới phục vụ cá nhân. |
| Usability | * Cung cấp kèm theo tài liệu và hướng dẫn từng trang và chức năng của hệ thống. * Giao diện được thiết kế bắt mắt, đơn giản, dễ hiểu, và dễ sử dụng. * Chức năng lên nhắc lịch trình thân thiện, dể sử dụng |
| Reliability | * Hệ thống với độ tin cậy trung bình. |
| Performance | * Công việc Ghi chú nhanh , hiệu quả hơn vì các thông tin sẽ được tập trung lại tại một giao diện. * Hiệu suất cao, khả năng chiếm cpu nhỏ. |
| Supportability | * Không có database, dữ liệu được lưu dưới dạng file JSON, tiện dụng * Có khả năng ghi log mỗi khi có lỗi. |

Bảng 3: Các tiêu chí FURPS

## Mô hình hóa

### Các trường hợp sử dụng thông thường

|  |  |  |  |
| --- | --- | --- | --- |
| ***Use Case*** | ***Tên*** | ***Mô tả*** | ***Yêu cầu liên quan*** |
| UC-1 | Events | Ghi chú lại những việc cần làm trong ngày | TKB, Mẫu Ghi chú,Status, Description, Hiện thị Thông báo theo ngày |
| UC-2 | Time | Đặt thời gian nhắc thông báo. | Nhắc Nhở |

Bảng 4: Bảng tổng hợp các use-case thông dụng

### Mô hình thực thể - mối quan hệ

#### Use Case 1 (Events):

Người dùng

Hình 1: Use case Events

#### Use Case 2 (Time)

Người dùng

Hình 2: Use case Time

Hình 4: Mô hình thực thể - mối quan hệ (bản lớn tại phụ lục)

# Thiết kế



## Thiết kế kiến trúc phần mềm

Phần mềm sẽ được thiết kế theo mô hình kiến trúc 3 lớp:

– Presentation Layer (GUI) : Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện ( win form, web form,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liêu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL).

– Business Logic Layer (BLL) : Layer này phân ra 2 thành nhiệm vụ :

* Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
* Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

– Data Access Layer (DAL) : Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,…).

### Ưu điểm

* Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.
* Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.
* Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại  khi có sự thay đổi giữa hai môi trường ( Winform sang Webfrom ) thì chỉ việc thay đổi lại lớp GUI.
* Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.
* Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

## Thiết kế dữ liệu

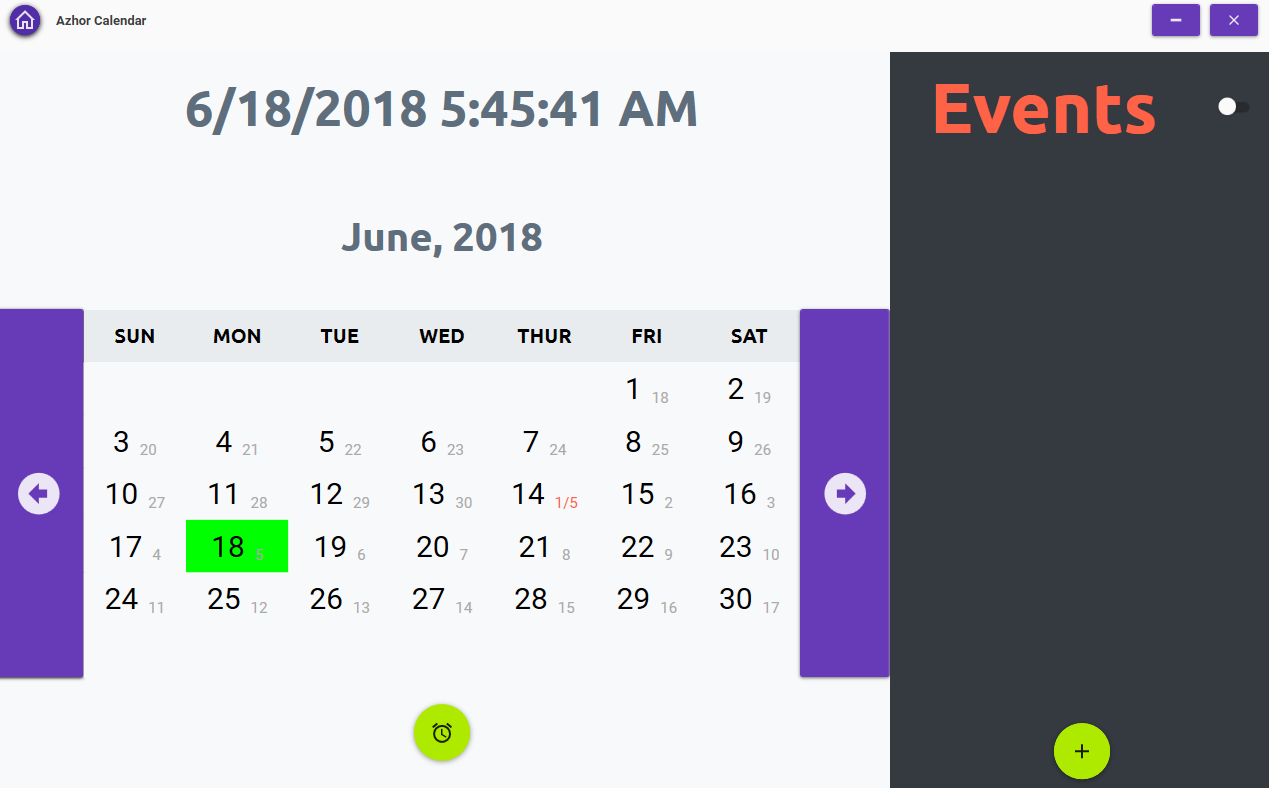
* Dữ liệu gồm Plan và Task được thiết kế như sau:



* Plan được chứa trong 1 dictionary với key là ngày lập Plan
* Mỗi Plan đại diện cho kế hoạch trong 1 ngày.
* Các Task là các công việc cần được thực hiện trong ngày đó và được lữu trữ trong mảng Tasks của Plan.
* Vì dữ liệu có dạng Key – Value nên lữu trữ dưới dạng chuỗi Json là hợp lý nhất.

## Thiết kế giao diện và thành phần xử lí của giao diện

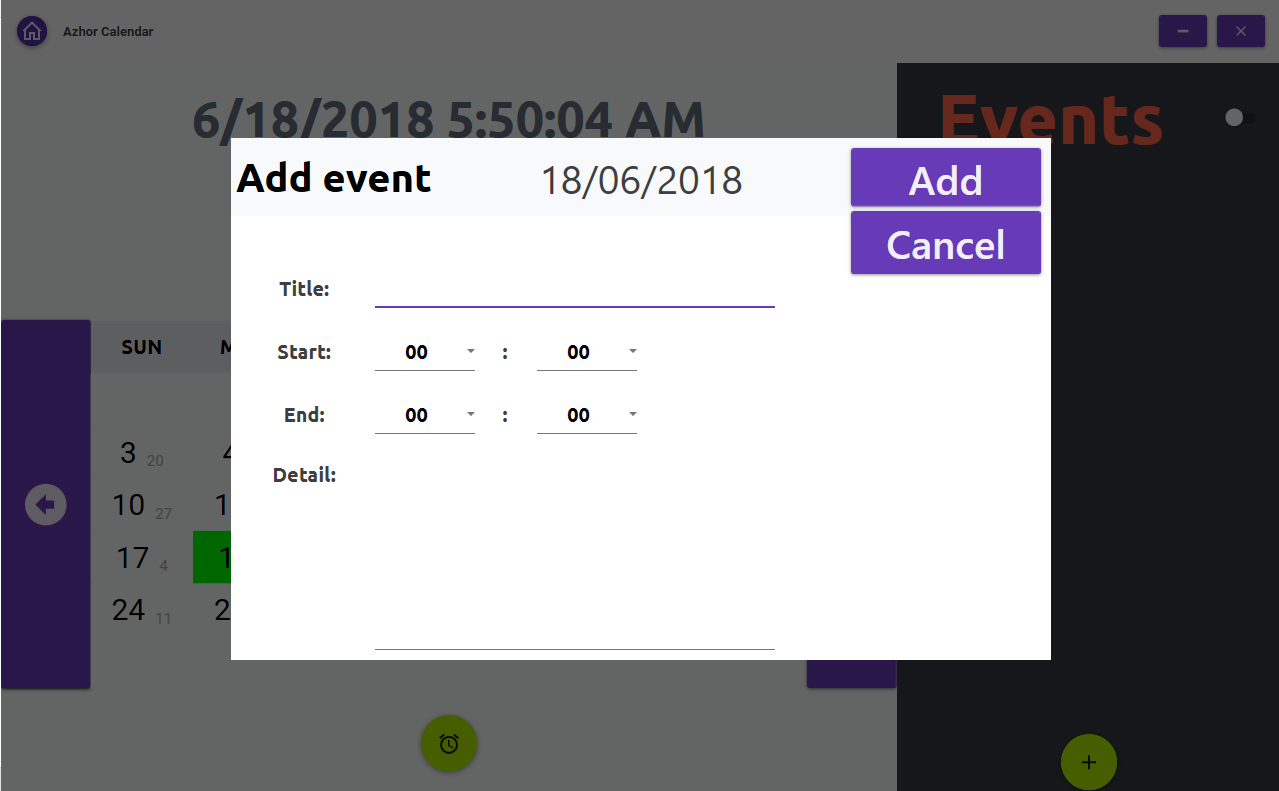
### Tổng thể về giao diện hệ thống



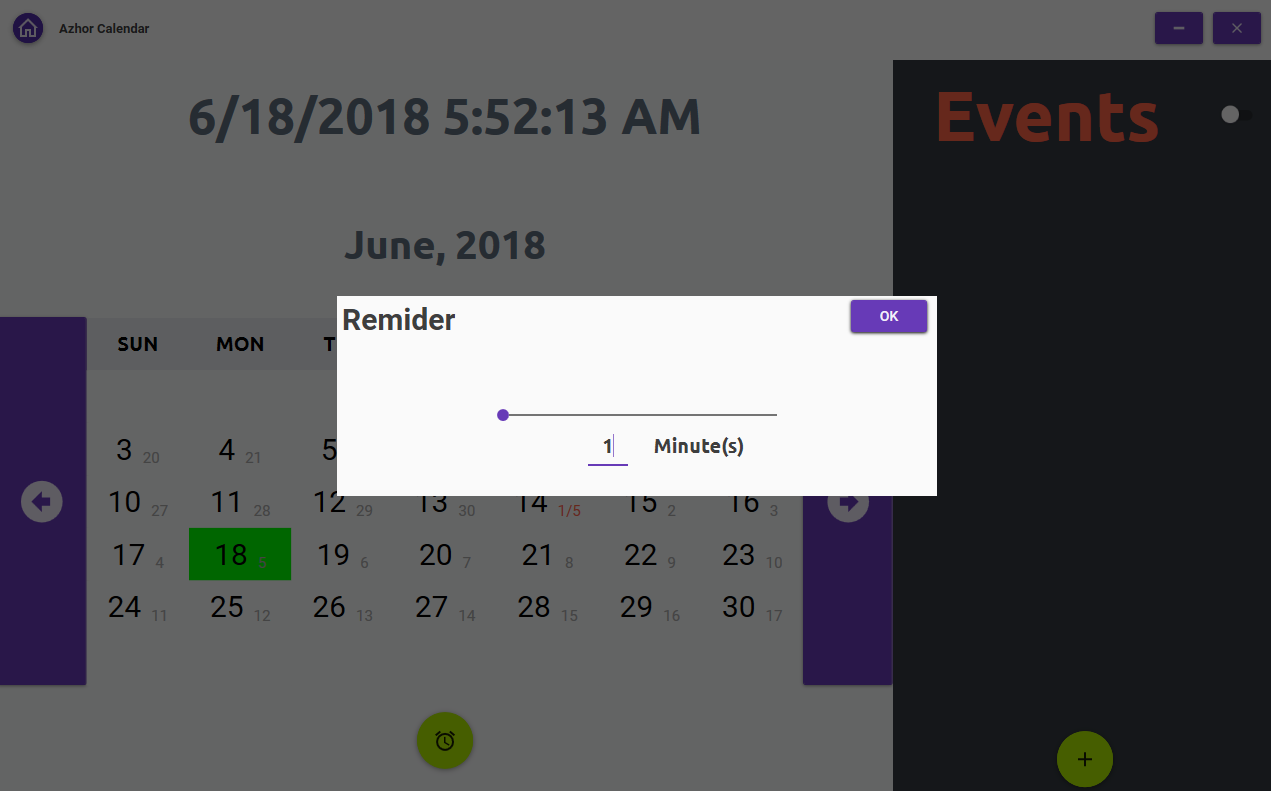
Hình 3: Giao diện chính

Mô tả thành phần giao diện: Giao diện chia làm 2 phần: phần Thời khóa biểu bên phía tay trái và phần sự kiện bên phía tay phải, TKB có nhiệm vụ hiện Lịch, EVENTS có nhiệm vụ hiện công việc cần làm trong ngày.

### Giao diện thêm sự kiện ( Add Event )



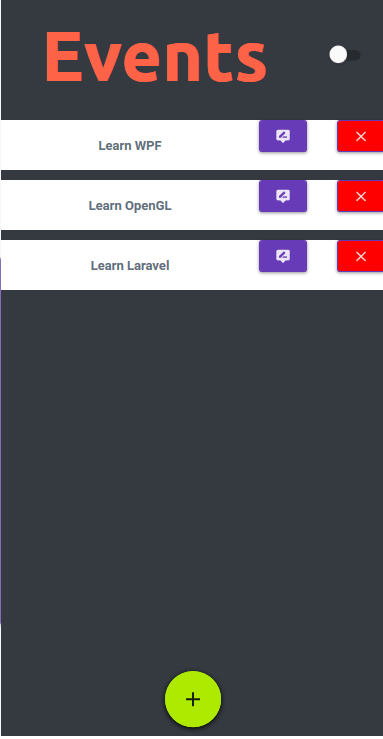
Hình 4: Giao diện thêm sự kiện



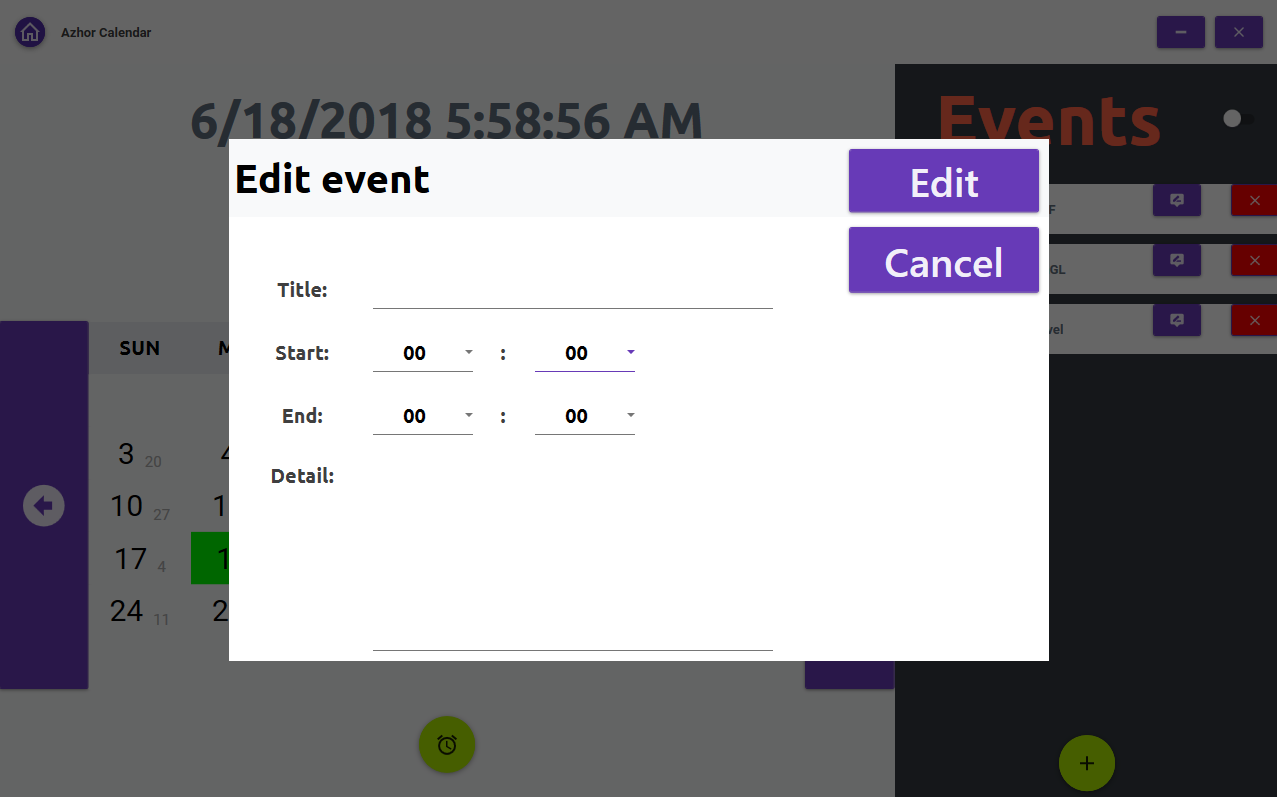
Hình 5: Giao diện thay đổi thời gian thông báo

### 

Hình 6: Giao diện thay đổi thời gian trên lịch



Hình 7: Giao diện sau khi thêm sự kiện.



*Hình 8: Giao diện sửa đổi sự kiện.*

## Thiết kế xử lý

### Events

Events là việc Lưu lại lịch trình, công việc sắp tới ,gồm có:

* Tittle:Tiêu đề
* Thời gian bắt đầu (Start Time)
* Thời gian kết thúc
* Detail: Mô tả chi tiết về sự kiện.

Mục đích của việc thiết kế này là vì đây là giao diện phổ biến thực tế, thông dụng

### Hide và Time

Time : Gồm có Set Reminder: Liên tục nhắc nhở mỗi (tùy chọn từ 1 đến 60) phút cho đến khi hết thời gian của sự kiện đó.

Hide: phần mềm sẽ chạy ẩn ở Tray Icon của Taskbar.

# Cài Đặt Phần Mềm



## Tổng quan về công nghệ sử dụng

### Ngôn ngữ lập trình C#

C# là một ngôn ngữ cấp cao và thuần hướng đối tượng, hỗ trợ rất mạnh trong việc xây dựng các ứng dụng trên môi trường Windows. Đối với những lập trình viên đã từng sử dụng qua C, C++, hoặc Java, việc nắm bắt và sử dụng C# một cách hiệu quả là khá dễ dàng và ít tốn thời gian.

Mã nguồn được viết bằng ngôn ngữ C# sẽ được dịch sang một ngôn ngữ tầm trung (Intermediate Language) phù hợp với các đặc tả của Common Language Infrastructure (CLI), và được lưu trữ trong các tập tin thực thi (thường có đuôi là .exe hay .dll.



Hình 21: .NET Framework platform architecture  
Credit: msdn.microsoft.com

### Json

JSON là một định dạng dùng để trao đổi dữ liệu phổ biến  biến hiện nay.  JSON là chữ viết tắt của Javascript Object Notation, cũng giống như XML nó có cấu trúc và quy tắc để lấy dữ liệu và 2 định dạng này có thể dễ dàng chuyển đổi cho nhau.

 Nhưng điểm khác là JSON là một phần  của Javascript, Cú pháp để thao tác và truy vấn trong với Json được hỗ trợ sẵn trong Javascript.

 Ví dụ : đối tượng Json lưu 1 danh sách cách sinh viên :



Hình 6: Json demo

  Với sự hỗ trợ đăc lực của các ngôn ngữ và framework phía client như Javascript và Jquery với Json, các services phía server khi trả dữ liệu có thể trả về dữ liệu với định dạng Json. Các đối tượng trong các ngôn ngữ lập trình hướng đối tượng như Java, C#, VB.net đều có thể chuyển đổi thành các chuỗi Json để trả về phía Client.

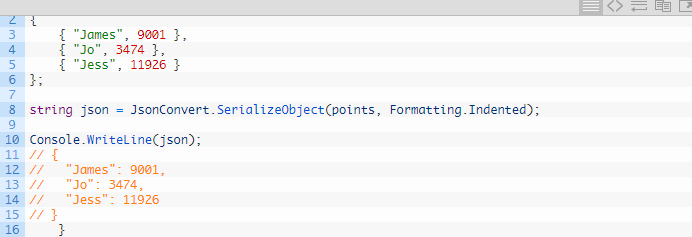
     Trong nền tảng .Net, thư viện Newtonsoft.Json là một thư viện mã nguồn mở tốt nhất để thao tác với JSON hiện nay và được tích hợp sẵn trong các Project phát triển Web của Microsoft. Nếu các bạn phát triển cần trao đổi dữ liệu dạng Json thì Newtonsoft.Json (Json.NET) là một sự lựa chọn không hề tồi.

+ Json.NET là một thư viện mã nguồn mở có thể sử dụng miễn phí cho mục đích thương mại [(xem)](https://github.com/JamesNK/Newtonsoft.Json/blob/master/LICENSE.md)

+ Json.NET hỗ trợ trong các nền tảng Windows, Windows Store, Windows Phone, Mono, and Xamarin.

+ Json.NET hỗ trợ convert  giữa XML và JSON.

 Dưới đây là ví dụ sử dụng : Convert một đối tượng Dictionary trong C# tới một chuỗi JSON

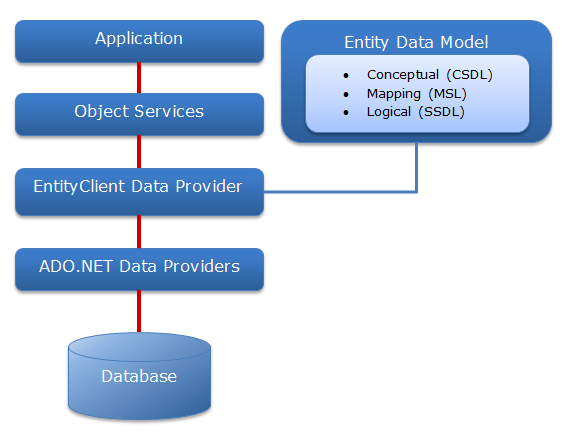


Hình 7: Convert Json

### Một số framework được sử dụng

* **Entity Framework Core**

Nền tảng được sử dụng để làm việc với database thông qua cơ chế ánh xạ Object/Relational Mapping (ORM). Nhờ đó, bạn có thể truy vấn, thao tác với database gián tiếp thông qua các đối tượng lập trình.



# Kiểm Thử



## Unit Test

Để đảm bảo chất lượng của chương trình, từng lớp sẽ được kiểm tra một cách riêng rẽ bằng phương pháp hộp trắng. Người tiến hành sẽ là những người trực tiếp cài đặt các lớp và phương thức ấy.

## Integration Test

Kiểm tra sự liên kết và giao diện giữa các lớp trong phần mềm.

## System Test

### Phương pháp kiểm tra

Hệ thống sẽ được kiểm tra theo phương pháp hộp đen. Tức là dựa vào những đặc tả và yêu cầu của hệ thống đã thu thập được từ đầu để kiểm tra output chung của hệ thống mà không cần quan tâm đến việc thực hiện bên trong của chúng.

# Tổng Kết



## Tổng Kết

Phần mềm bước đầu đã được phát triển đi đúng theo dự tính và kế hoạch. Tuy gặp khá nhiều khó khăn trong vấn đề tiếp cận với công nghệ lập trình mới (WPF) , nhóm vẫn đảm bảo được các tiến độ làm việc do đã đặt ra.

## Nhận Xét & Đánh Giá

Trong thời gian có hạn, việc hoàn thành dự án đã vượt qua được nhiều thử thách nhất định nhưng phần mềm vẫn còn nhiều điểm có thể cải tiến nhằm nâng cao chất lượng phục vụ và trải nghiệm của người dùng:

* Cải thiện database lớn hơn
* Cài đặt và hoàn thiện nhiều tính năng mới hơn
* Hướng đến kết nối Internet, đưa phần mềm vào mạng,kết hợp php, mysql, Json,….

## Phân Rã Công Việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Công việc** | **Duration (days)** | **Assign To** | **% Finished** |
| 1 | Tìm hiểu sơ bộ & đăng ký đồ án |  | Viết | 100% |
| 2 | Tìm hiểu công nghệ WPF, Winform |  | Viết, Vỹ, Minh | 80% |
| **I. VIẾT BÁO CÁO** | | | | |
| 3 | **Chương 1 – Hiện trạng** |  |  |  |
| 4 | 1.1. Hiện trạng vấn đề |  | Vỹ ,Minh | 100% |
| 5 | 1.2. Hiện cơ sở vật chất và con người |  | Vỹ | 100% |
| 6 | 1.3. Yêu cầu về phần mềm |  | Viết | 100% |
| 7 | **Chương 2: Phân tích** |  |  |  |
| 8 | 2.1. Yêu cầu phần mềm |  | Viết | 90% |
| 9 | 2.2. Mô hình hóa |  | Viết,Vỹ | 100% |
| 10 | **Chương 3: Thiết kế** |  |  |  |
| 14 | 3.3. Thiết kế giao diện và thành phần xử lí của giao diện |  | Viết, Vỹ ,Minh | 90% |
| 15 | **Chương 4: Cài đặt** |  | Vỹ, Viết | 100% |
| 19 | **Chương 5: Kiểm thử** |  | Minh, Vỹ | 80% |
| 23 | **Chương 6: Kết luận** |  | Minh | 85% |
| 24 | **Tài liệu tham khảo** |  |  |  |
| **II. LẬP TRÌNH** | | | | |
|  | 2.Hiển Thị Lịch | 20 | Minh | 80% |
|  | 3. Giao Diện,Events, | 15 | Viết | 100% |
|  | Json, Hide, Reminder | 15 | Vỹ | 80% |
| **III. KIỂM THỬ** | | | | |
|  | 1. Unit Test |  |  |  |
|  | 2. Integration Test |  |  |  |
|  | 3. System Test |  |  |  |
|  | 4. Acceptance Test |  |  |  |
| **IV. NỘP BÁO CÁO TIẾN ĐỘ & SẢN PHẨM** | | | **Tiến độ sản phẩm** | |
|  | **Ngày 26 tháng 03 năm 2018** | | Hoàn thiện ý tưởng đồ án. | |
|  | **Ngày 10 tháng 05 năm 2018** | | Hoàn Thiện chức năng chính của phần mềm | |
|  | **Ngày 20 tháng 05 năm 2018** | | Kết nối các code thành 1 sản phẩm hoàn chỉnh, thiết kế giao diện | |
|  | **Ngày 15 tháng 06 năm 2018** | | Hoàn thiện sản phẩm, cài đặt test thử. | |

TÀI LIỆU THAM KHẢO

(IEEE Citation Format)

<http://hoctoantap.com/2016/09/09/thu-vien-xu-ly-du-lieu-dang-json-trong-c-newtonsoft-json.html>

<https://www.youtube.com/watch?v=Vjldip84CXQ&list=PLrW43fNmjaQVYF4zgsD0oL9Iv6u23PI6M>

[C# Yellow Book](https://www.robmiles.com/c-yellow-book/) – Rob Miles

 WPF 4 – Unleashed

Giao trình Lập trình trực quan ĐH CNTT\_DHQG Tp HCM