# **CHƯƠNG 1. TỔNG QUAN**

## **Giới thiệu đề tài**

### **Tên đề tài:** Công ty cổ phần Dược Phẩm

### **Trưởng nhóm:**

Trần Kiến An

### **Danh sách thành viên**

* Trần Kiến An – 15DH110210
* Đặng Phạm Phú Duy – 15DH110
* Lâm Minh Hiếu – 15DH110
* Nguyễn Khải – 15DH110

### **Tổng mức đầu tư**

* Dự án có tổng mức đầu tư lên đến 299000 triệu vnđ.

### **Thời gian thực hiện:**

* Ngày bắt đầu: 01/02/2018
* Ngày kết thúc: 20/05/2018

### **Mục đích đề tài:**

Giúp công ty quản lý nhân viên, quản lý việc xuất nhập hàng, quản lí quá trình giao hàng của công ty, quản lí đơn đặt hàng.

### **Mục tiêu đề tài cần thực hiện**

* Yêu cầu chức năng: phải có các chức năng

+ Cho khách đặt hàng

+ Quản lí việc giao hàng

+ Thu tiền

+ Ghi lại thông tin khách hàng

+ Báo cáo hoạt động giao hàng

* Yêu cầu phi chức năng:

+ Chương trình chạy tốt trên HĐH windows

+ Load dữ liệu nhanh

+ Có phân quyền quản lí

+ Giao diện thân thiện, dễ dung

* Các module yêu cầu cho phần mềm:

+ Module quản lí

+ Module báo cáo

+ Module tính tiền

+ Module giao hàng

### **Công cụ và môi trường phát triển phần mềm**

* Phần cứng;

+ Laptop

+ PC

* Phần mềm

+ Microsoft Office

+ Visio

+ Microsoft Visual Studio

+ SQL server

* Môi trường test:

+ Windows OS

## **Tổ chức dự án phần mềm**

### **Nhân sự:**

#### Tổ chức nhóm:

* Nhóm em tổ chức theo tổ chức ma trận

#### Vai trò của từng thành viên:

* Kiến An thiên về hướng bản thân (LEADER)
* Phú Duy thiên về hướng công việc
* Minh Hiếu thiên về hướng tương tác
* Khải thiên về hướng tương tác và hướng công việc

### **Mô hình quản lý**



### **Ma trận trách nhiệm:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên người thực hiện** | **Quản lí project** | **Phân tích** | **Thiết kế** | **Lập trình** | **Test** |
| Kiến An | P | A |  | R | P |
| Phú Duy | C | P | P | P | P |
| Minh Hiếu | C | C | I |  | C |
| Khải | C | C | R |  | I |

Chú thích: Các kiểu trách nhiệm khác nhau trên công việc

A (Approving): Xét duyệt

P (Performing): Thực hiện

R (Reviewing): Thẩm định

C (Contributing): Tham gia đóng góp

I (Informing): Báo cho biết

# **CHƯƠNG 2. LẬP KẾ HOẠCH THỰC DỰ ÁN PHẦN MỀM**

## **Giới thiệu nhân sự:**

**Trần Kiến An**

Lập trình viên

**Ngày sinh:** 17/08/1997

**Giới tính:** Nam

**Điện thoại:** 0995326751

**Email:** k.antran1781997@gmail.com

**Địa chỉ:** 59/6A Lý Chiêu Hoàng, P10, Q.6, Hồ Chí Minh

**Website:** <https://www.facebook.com/kienan.tran>

**Bằng cấp:** Đại học

**MỤC TIÊU NGHỀ NGHIỆP**

Áp dụng những kinh nghiệm về kỹ năng lập trình và sự hiểu biết về thiết kế giao diện của một lập trình viên, mang đến nhiều giá trị cho khách hàng. Trao dồi vốn nghề nghiệp cho bản thân, nâng cao kinh nghiệm trong công việc, hoàn thiện bản thân.

**KĨ NĂNG**

**Ngôn ngữ lập trình** Hiểu biết về C#, Java, Java Script, MS SQL Server, Unity, …

**Tin học văn phòng** Sử dụng được các phần mềm Word, Power Point, Excel, …

**Tiếng Anh** Khả năng giao tiếp Tiếng Anh ổn

**Kĩ năng mềm** Có khả năng lãnh đạo và làm việc nhóm hiệu quả

Tiếng Anh giao tiếp và ngôn ngữ khá

**THÁI ĐỘ**



Làm việc nghiêm túc và có tâm huyết với công việc

**SỞ THÍCH**



Nghe nhạc Bóng đá

Xem phim Ăn uống

**Đặng Phạm Phú Duy**

Lập trình viên

**Ngày sinh:** 09/08/1997

**Giới tính:** Nam

**Điện thoại:** 0916576568

**Email:** duydang0908@gmail.com

**Địa chỉ:** 188 Nghĩa Phát, phường 7, Tân Bình, Hồ Chí Minh

**Website:** <https://www.facebook.com/duy.dang.524934>

**Bằng cấp:** Đại học

**MỤC TIÊU NGHỀ NGHIỆP**

Áp dụng những kinh nghiệm về kỹ năng lập trình của một lập trình viên, mang đến nhiều giá trị cho khách hàng. Từ đó giúp Công ty tăng số lượng khách hàng và mở rộng tập khách hàng. Trao dồi vốn nghề nghiệp cho bản thân, nâng cao kinh nghiệm trong công việc, hoàn thiện bản thân.

**KĨ NĂNG**

**Ngôn ngữ lập trình** Hiểu biết về C#, Java, MS SQL server, ...

**Tin học văn phòng** Sử dụng thành thạo các công cụ Word, Excel, Power Point

**Tiếng Anh** Khả năng giao tiếp Tiếng Anh khá trôi chảy

**Kĩ năng mềm** Biết lắng nghe

Bình tĩnh trong mọi tình huống

Khả năng kết nối các thành viên

**THÁI ĐỘ**



Làm việc tốt và có tâm huyết với công việc

Luôn có những ý tưởng hay

**SỞ THÍCH**



Đọc sách Bóng đá

**Lâm Minh Hiếu**

Lập trình viên

**Ngày sinh :** 27/09/1997

**Giới tính :** Nam

**Điện thoại :** 0918 45 46 15

**Email :** lamminhhieu2709@gmail.com

**Địa chỉ :** 266A Tô Hiến Thành, F.15, Q.10, TP.HCM

**Website :** https://www.facebook.com/lamminhhieu.339

**Bằng cấp:** Đại học

**MỤC TIÊU NGHỀ NGHIỆP**

Với sự đam mê lập trình, nguyện vọng của tôi là được làm việc ở vị trí lập trình viên trong môi trường năng động, chuyên nghiệp, cởi mở và chú trọng phát triển con người. Từ đó hoàn thiện kiến thức, kỹ năng bản thân, đóng góp nhiều hơn cho công ty.

**KĨ NĂNG**

**Ngôn ngữ lập trình** Hiểu biết về C#, Java, Java Script, MS SQL Server, Unity, …

**Tin học văn phòng** Sử dụng được các phần mềm Word, Power Point, Excel, …

**Tiếng Anh** Tiếng Anh giỏi và khả năng giao tiếp tốt

**Kĩ năng mềm** Có khả năng thuyết trình lưu loát trôi chảy

Tiếng Anh giao tiếp và ngữ pháp tốt

**THÁI ĐỘ**



Kỹ năng tuy chưa tốt lắm nhưng rất cố gắng để hoàn thành công việc được giao

**SỞ THÍCH**



Đọc sách Bóng đá

Xem phim Ăn uống

**Nguyễn Khải**

Lập trình viên

**Ngày sinh :** 16/08/1997

**Giới tính :** Nam

**Điện thoại :** 01263182043

**Email :** nguyenkhai.huflit@gmail.com

**Địa chỉ :** 631/22 Lê Đức Thọ , Gò Vấp

**Website :** https://www.facebook.com /nkohaki

**Bằng cấp:** Đại học

**MỤC TIÊU NGHỀ NGHIỆP**

Ngắn hạn: Học hỏi thêm kinh nghiệm từ những người đi trước

Dài hạn: Áp dụng kinh nghiệm về lập trình để giúp công ty phát triển các phần mềm

**KĨ NĂNG**

**Ngôn ngữ lập trình** C#, Java, HTML, JavaScript, CSS

**Tin học văn phòng** Sử dụng được các phần mềm Word, Power Point, Excel, …

**Tiếng Anh** Tiếng Anh khá và khả năng giao tiếp tốt

**Kĩ năng mềm** Làm việc nhóm tốt

Khả năng thuyết trình lưu loát

Biết lắng nghe và học tập từ những lời phê bình

**THÁI ĐỘ**



Làm việc tốt và luôn hoàn thành tốt công việc được giao

Có những ý tưởng rất táo bạo

**SỞ THÍCH**



Đọc sách Nghe nhạc

Đi du lịch Chơi game

## **Lịch biểu công việc**

### **Thời gian tổng thể**

+ Tổng thời gian dự kiến: 13 tuần

* Quản lý project: 1 tuần
* Phân tích: 1 tuần
* Thiết kế: 2 tuần
* Lập trình và tích hợp hệ thống: 6 tuần
* Kiểm thử và sửa lỗi: 2 tuần

Vẽ sơ đồ pert

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 tuần  1 tuần  1 tuần  3 tuần  6 tuần   |  | | --- | |  | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Tìm đường gang: Giai đoạn 1 -> Giai đoạn 3 -> Giai đoạn 4 -> Giai đoạn 5

### **Thời gian chi tiết**

Ngày khởi động dự án: 01/2/2018

Giai đoạn 1. Quản Lý Project

* 1. Người thực hiện: Kiến An, Phú Duy, Nguyễn Khải, Minh Hiếu
  2. Bảng chi tiết:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Số ngày thực hiện** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Lên ké hoạch | 1 | 01/2 | 02/2 |
| Họp nhóm | 2 | 02/2 | 04/2 |
| Phân chia công việc | 1 | 04/2 | 05/2 |

Giai đoạn 2. Phân tích

1. Người thực hiện: Nguyễn Khải, Kiến An, Phú Duy
2. Bảng chi tiết:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Số ngày thực hiện** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Xác định yêu cầu | 1 | 27/2 | 28/2 |
| Phân tích Actor | 2 | 03/3 | 05/3 |
| Phân tích Usecase | 2 | 05/3 | 07/3 |

Giai đoạn 3. Thiết kế

1. Người thực hiện: Nguyễn Khải, Phú Duy, Minh Hiếu
2. Bảng chi tiết:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Số ngày thực hiện** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Thiết kế CSDL | 8 | 08/3 | 16/3 |
| Thiết kế giao diện | 6 | 17/ | 23/03 |

Giai đoạn 4. Thiết kế

1. Người thực hiện: Phú Duy, Kiến An, Nguyễn Khải
2. Bảng chi tiết:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Số ngày thực hiện** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Kết nối SQL | 2 | 24/3 | 26/3 |
| Code từng chức năng | 30 | 26/3 | 26/4 |

Giai đoạn 5. Thiết kế

1. Người thực hiện: Phú Duy, Minh Hiếu, An Kiến
2. Bảng chi tiết:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Số ngày thực hiện** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Chạy thử Project | 1 | 03/5 | 04/5 |
| Tìm lỗi sai | 5 | 04/5 | 09/5 |
| Sửa hoàn chỉnh | 7 | 09/5 | 16/5 |

* **Vẽ sơ đồ gantt**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhiệm vụ** | **Số ngày thực hiện** | **Ngày bắt đầu** | **Ngày kết thúc** |
| Lên ké hoạch | 1 | 1/2/2018 | 2/2/2018 |
| Họp nhóm | 2 | 2/2/2018 | 4/2/2018 |
| Phân chia công việc | 1 | 4/2/2018 | 5/2/2018 |
| Xác định yêu cầu | 1 | 27/2/2018 | 28/2/2018 |
| Phân tích Actor | 2 | 3/3/2018 | 5/3/2018 |
| Phân tích Usecase | 2 | 5/3/2018 | 7/3/2018 |
| Thiết kế CSDL | 8 | 8/3/2018 | 16/3/2018 |
| Thiết kế giao diện | 6 | 17/3/2018 | 23/3/2018 |
| Kết nối SQL | 2 | 24/3/2018 | 26/3/2018 |
| Code từng chức năng | 30 | 26/3/2018 | 26/4/2018 |
| Chạy thử Project | 1 | 3/5/2018 | 4/5/2018 |
| Tìm lỗi sai | 5 | 4/5/2018 | 9/5/2018 |
| Sửa hoàn chỉnh | 7 | 9/5/2018 | 16/5/2018 |

### **Cấu trúc bảng công việc (WBS)**



### **Tài chính**

#### Chi phí tổng quan

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Công việc | Chi phí | Ghi Chú |
| 1 | Khởi động dự án | 1.000.000 |  |
| 2 | Quản lí dự án | 5.000.000 |  |
| 3 | Phân tích | 10.000.000 |  |
| 4 | Thiết kế | 20.000.000 |  |
| 5 | Lập trình | 25.000.000 |  |
| 6 | Test | 15.000.000 |  |

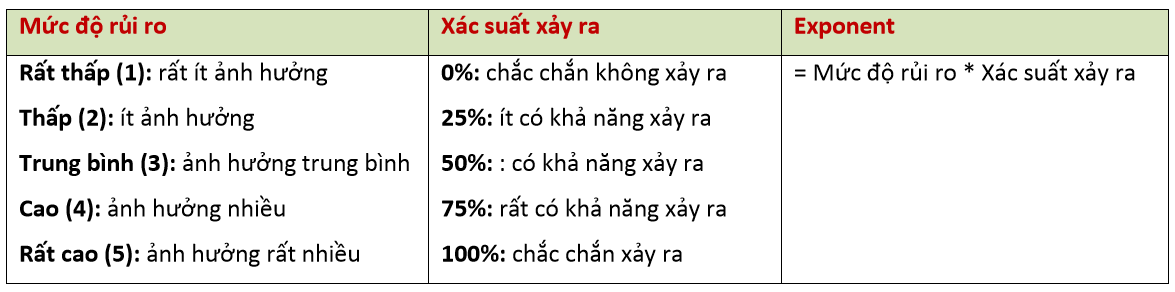
#### Chi phí chi tiết

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Chi tiết | Đơn giá | Thời gian (Giờ) | Thành tiền |
| 1 | Lên kế hoạch | 500.000 | 24 | = |
| Họp nhóm | 355.000 |  |  |
| Phân chia công việc | 145.000 |  |  |
| 2 | Xác định yêu cầu | 2.450.000 |  |  |
| Phân tích Actor | 2.000.000 |  |  |
| Phân tích Usecase | 5.550.000 |  |  |
| 3 | Thiết kế CSDL | 8.500.000 |  |  |
| Thiết kế giao diện | 11.500.000 |  |  |
| 4 | Kết nối CSDL | 6.500.000 |  |  |
| Lập trình từng chức năng | 18.500.000 |  |  |
| 5 | Chạy thử project | 4.000.000 |  |  |
| Tìm lỗi sai | 5.500.000 |  |  |
| Sửa lỗi sai và hoàn toán | 5.500.000 |  |  |

#### Lập bảng phân tích rủi ro

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung rủi ro** | **Ưu tiên** | **Thiệt hại chi phí** | **Thiệt hại lợi ích** | **Kế hoạch giải quyết** |
| 1 | Lỗi code | 2 | Tốn thời gian | 80% | Tìm ra lỗi và sửa lại |
| 2 | Cúp điện | 3 | 50$/ngày | 99% | Chạy máy phát điện |
| 3 | Mất laptop | 2 | >700$/Laptop | 80% | Mua laptop mới |
| 4 | Trễ giờ làm | 2 | 1$/phút | 99% | Đặt ra quy định trễ sẽ bị phạt |
| 5 | Việc gia đình | 2 | 100$/ngày | 60% | Tìm người thay thế |
| 6 | Bệnh tật | 3 | 100$/ngày | 50% | Tìm người thay thế |
| 7 | Đồng nghiệp chơi xấu | 2 | Tùy theo mức độ | 100% | Báo quản lí |
| 8 | Nhân viên bỏ việc | 5 | 300$/ngày | 80% | Tuyển nhân viên mới |
| 9 | Nhân viên thái độ lười biếng | 4 | Tùy theo dự án | 80% | Đuổi việc |
| 10 | Đi sai mục đích công việc | 4 | Tốn thời gian | 90% | Định hướng lại mục đích công việc |
| 11 | Khách hàng thay đổi yêu cầu liên tục | 5 | Tùy yêu cầu | 90% | Định hướng lại mục đích công việc |
| 12 | Mâu thuẫn ý kiến thành viên trong nhóm | 5 | Tốn thời gian | 80% | Họp lại thống nhất ý kiến |
| 13 | Hỏng hệ thống | 4 | 500$/ngày | 99% | Bảo trì hê thống |
| 14 | Thiết bị hư hỏng | 4 | Tùy theo thiết bị | 100% | Mua mới hoặc sửa |
| 15 | Thiếu nhân lực | 5 | Tùy theo dự án | 90% | Thông báo tuyển thêm và đào tạo nhân viên mới vào nghề |
| 16 | Vấn đề bản quyền | 3 | Tùy theo dự án | 100% | Nhờ bên pháp lý giải quyết |

Giải thích



# **CHƯƠNG 3. PHÂN TÍCH USECASE**

(dựa vào đặc tả tình huống có thể mở rộng thêm nếu đặc tả ít)

## **Xác định các actor**

* Bộ phân phân phối hàng hóa
* Khách hàng
* Nhân viên giám sát giao hàng
* Nhân viên giao hàng

## **Xác định các UseCase**

* **Bộ phận phân phối hàng hóa của công ty**

+ Quản lý việc điều phối giao hàng

+ Thu tiền

+ Báo cáo hoạt động giao hàng của công ty

* **Khách hàng**

+ Đặt hàng:

* Trực tiếp đến công ty yêu cầu đặt các mặt hàng
* Gọi điện thoại đặt hàng
* Đặt hàng qua mạng

+ Ghi lại đầy đủ thông tin của khách hàng

+ Thanh toán đơn đặt hàng

+ Kí nhận đơn hàng

+ Nhận 1 tờ hóa đơn đỏ

* **Nhân viên giám sát giao hàng**

**+** Điều phối toàn bộ việc quản lý giao hàng của các nhân viên

**+** Quản lý việc điều phối bộ phận giao hàng

* **Nhân viên giao hàng**

+ Nhận đơn đặt hàng

+ Nhận thanh toán của khách hàng

+ Báo cáo cho GSGH tình hình giao hàng

+ Đưa khách hàng ký nhận

+ Hóa đơn đỏ:

* Gửi cho khách hàng
* Đem về lưu

+ Nộp tiền

## **Biểu đồ UseCase**

















## **Đặc tả Usecase**

* **Quản lý việc điều phối giao hàng**

**Actor:**

+ Khách hàng

+ Nhân viên giám sát giao hàng

+ Nhân viên giao hàng

**Description:** Cho phép nhân viên giám sát điều phối nhân viên giao hàng đi giao hàng

**Goal:** Giao hàng thành công

**Pre-condition:** Phải có dơn hàng

**Trigger:** Nhân viên điều phối điều phối nhân viên giao hàng

**Post-condition:**

+ Đơn hàng trong trạng thái đang chuyển

+ Số hàng trong kho bị giảm đi

**Normal flow:**

**+** Khách hàng đặt hàng

**+** Nhân viên điều phối điều phối nhân viên chuyển hàng

+ Nhân viên chuyển hang chuyển hang đến khách hàng

**Alternative flow:**

+ Khi khách hàng hủy đơn hàng

+ Nhân viên giao hàng đem hàng về kho

+ Exception flow: Lỗi xảy ra khi giao hàng -> điều phối cho chuyển hàng lại

* **Thu tiền**

**Name:** Thu TIền

**Actor:**

+ Khách hàng

+Nhân viên giao hàng

**Description:** Thu tiền khi khách nhận ~~hàng~~

**Goal:** Thu tiền thành công

**Pre-condition:** Phải có đơn hàng

**Trigger:** Nhân viên đưa hàng đến tay khach hàng

**Post-condition:** Khách hàng nhận được hàng, đơn hàng được chuyển

**Normal flow:**

+ Nhân viên đưa hàng đến khách hàng

+ Khách hàng nhận hàng

+ Khách hàng trả tiền đủ

**Alternative flow**:

Chuyển hàng sai

* Nhân viên quay về và đưa lại đúng đơn

**Exception flow:**

Không gặp được khách hàng

* Nv đưa hàng về kho, hủy đơn
* **Báo cáo hoạt động giao hàng của công ty**

**Name:** Báo cáo hoạt động giao hàng của công ty

**Actors:**

+ Bộ phận phân phối hàng hóa

+ Nhân viên điều phối giao hàng

**Description:** Báo cáo hoạt động giao hàng của công ty

**Goal:** Báo cáo hoàn tát

**Pre-condition:** Có đơn hàng

**Trigger:** Nhân viên nhấn nút gửi báo cáo

**Post-condition:** Báo cáo được gửi

**Normal flow :**

+ Nhân viên điều phối tổng hợp đơn

+ Nv điều phối gửi bản tổng hợp đến bộ phận phân phối

**Alternative flow:** Báo cáo sai doanh thu > Tổng hợp lại

**Exception flow:** Không gửi đượcb áo cáo > gửi lại

* **Đặt hàng**

**Name:** Đặt hàng

**Actors:** Khách hàng

**Description:** Cho phép khách hàng đặt hàng

**Goal:** Đặt hàng thành công

**Pre-conditon:** Website đặt hàng phải hiện thông tin đầy đủ

**Trigger:** Khách hàng click vào nút Đặt hàng

**Post-condition:** Đặt hàng thành công

**Normal flow:**

+ Khách hàng đăng nhập

+ Tìm sản phẩm

+ Xem thông tin sản phẩm

+ Xem thông tin hóa đơn

+ Confirm hóa đơn

**Alternative flow:** Không còn sản phẩm muốn mua > Tìm những sản phẩm tương tự mua

**Exception flow:** Thanh toán trục trặc > Đặt lại

* **Ghi lại đầy đủ thông tin của khách hàng**

**Name:** Ghi lại đầy đủ thông tin của khách hàng

**Actors:** Bộ phận phân phối giao hàng

**Description:** Khai báo thông tin của khách hàng.

**Goal:** Lưu trữ thông tin khách hàng.

**Pre-condition:** Khách hàng phải mua hàng.

**Trigger:** Khách hàng nhấn vào nút đặt hàng

**Post-condition:** Thông tin khách hàng được lưu lại

**Normal flow:**

+ Khách hàng đăng kí tài khoản

+ Khách hàng điền thông tin cá nhân

+ Khách hàng nhấn tạo tài khoản

+ Thông tin khách hàng được lưu lại

**Alternative flow:**

+ Khách hàng không đăng nhập để đặt hàng

+ Yêu cầu đăng nhập.

**Exception flow:** Tài khoản bị mất -> tạo lại tài khoản

* **Thanh toán đơn đặt hàng**

**Actors:** Khách hàng, nhân viên giao hàng.

**Description:** Khách hàng trả tiền để thanh toán đơn hàng.

**Goal:** Đơn hàng được thanh toán.

**Pre-condition:** Phải có đơn hàng.

**Trigger:** Khách hàng nhấn vào nút đặt hàng

**Post-condition:**

+ Đơn hàng được thanh toán

+ Khách hàng nhận được hàng.

**Normal flow:**

+ Khách hàng đặt hàng

+ Đơn hàng được xử lí

+ Hàng được chuyển đến khách hàng

+ Khách hàng nhận hàng và trả tiền

+ Đơn hàng được thanh toán

**Alternative flow:** Đơn hàng bị hủy

**Exception flow:** Khách hàng không nhận hàng được đặt -> Hủy đơn hàng

* **Kí nhận đơn hàng**

**Name:** Kí nhận đơn hàng

**Actors:** Khách hàng, nhân viên giao hàng

**Description:** Khách hàng kí xác nhận đơn hàng

**Goal:** Đơn hàng được kí nhận

**Pre-condition:** Có đơn hàng

**Trigger:** Nhân viên giao hàng đến khách hàng

**Post-condition:**

+ Khách hàng kí nhận

+ Hàng được chuyển thành công

**Normal low:**

+ Nhân viên giao hàng đến cho khách hàng

+ Khách hàng kiểm tra hàng

+ Khách hàng kí nhận đơn hàng

**Alternative flow:** Nhân viên giao hàng đến trễ -> Khách hàng đợi nv đến giao

**Exception flow:** Sai đơn hàng > Khách hàng không kí nhận

* **Nhận 1 tờ hóa đơn đỏ**

**Name:** Nhận 1 tờ hóa đơn đỏ

**Actor:**

**+** Khách hàng

+ Nhân viên giao hàng

**Description:** Khách hàng nhận 1 tờ hóa đơn đỏ

**Goal:** Nhận hóa đơn đỏ thành công

**Pre-condition:** Có đơn hàng

**Trigger:** Nhân viên giao hàng đến cho khách hàng

**Post-condition:** Hóa đơn được giao

**Normal flow:**

+ Nhân viên giao hàng đến cho khách hàng

+ Khách hàng kiểm tra hàng

+ Khách hàng nhận hóa đơn đỏ

**Alternative flow:** Khách hàng không có ở nhà. > Nhân viên giao hàng đưa hóa đơn cho hàng xóm nhận giùm

**Exception flow:** Khách hàng hủy đơn > Nhân viên mang hàng về kho.

* **Nhận đơn đặt hàng**

**Name:** Nhận đơn đặt hàng

**Actors:** Nhân viên giao hàng

**Description:** Nhân viên giao hàng nhận đơn đặt hàng

**Goal:** Đi giao hàng

**Pre-condition:** Có khách mua hàng

**Trigger:** Khách hàng nhấn vào nút đặt hàng

**Post-condition:** số hàng có thể đặt trong kho giảm

**Normal flow:** Khách hàng đặt hàng Nhân viên nhận đơn đặt hàng

**Alternative flow:** Khách hàng hủy đơn đặt hàng -> Nhân viên hủy giao hàng

* **Đem 1 tờ hóa đơn đỏ về lưu**

**Actors:** Nhân viên giao hàng, khách hàng

**Description:** Nhân viên giao hàng xuất hóa đơn đỏ cho khách hàng khi khách thanh toán và lưu lại đem về công ty

**Goal:** Lưu lại hóa đơn đỏ

**Pre-condition:** Khách hàng yêu cầu hóa đơn đỏ khi thanh toán

**Trigger:** Khách hàng nhấn nút đặt hàng

**Post-condition:** Nhân viên giao hàng đem hóa đơn đỏ về công ty

**Normal flow:**

+ Khách hàng đặt hàng

+ Nhân viên giao hàng giao hàng

+ Khách yêu cầu xuất hóa đơn đỏ

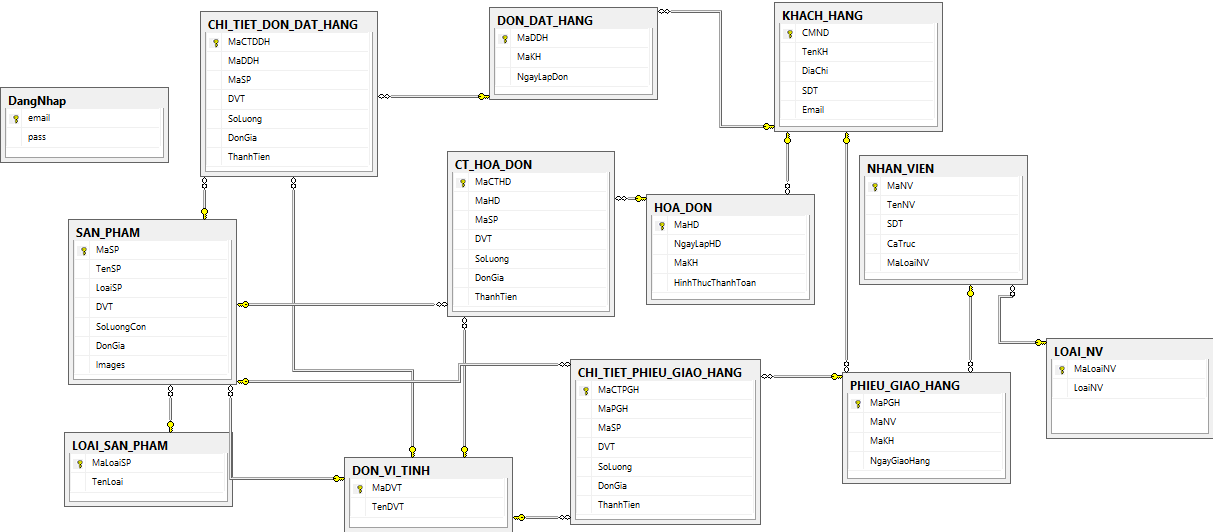
+ Nhân viên giao hàng xuất hóa đơn đỏ

+ Nhân viên giao hàng đem 1 bản hóa đơn đỏ về công ty lưu lại

**Alternative flow:** Nhân viên làm mất hóa đơn đỏ -> Trừ lương

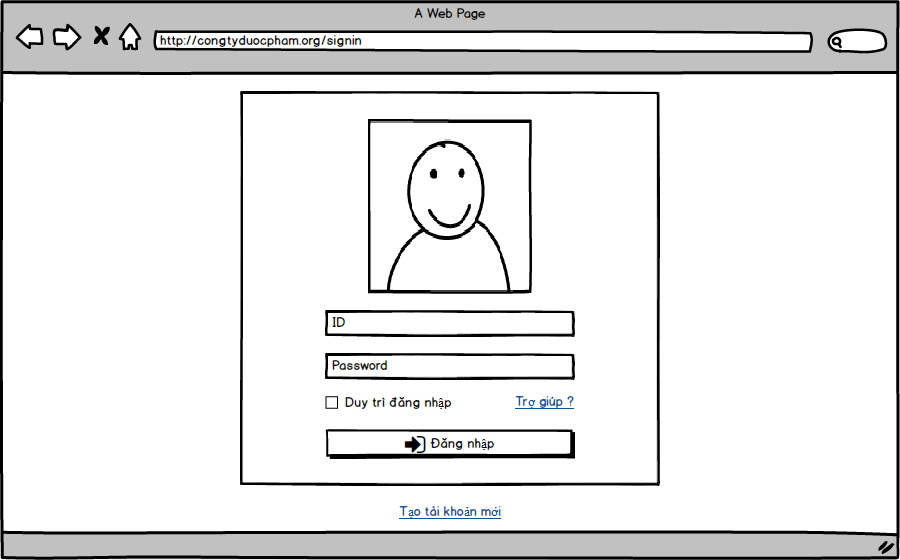
# **CHƯƠNG 4. ƯỚC LƯỢNG GIÁ PHẦN MỀM**

## **Cơ sở dữ liệu**

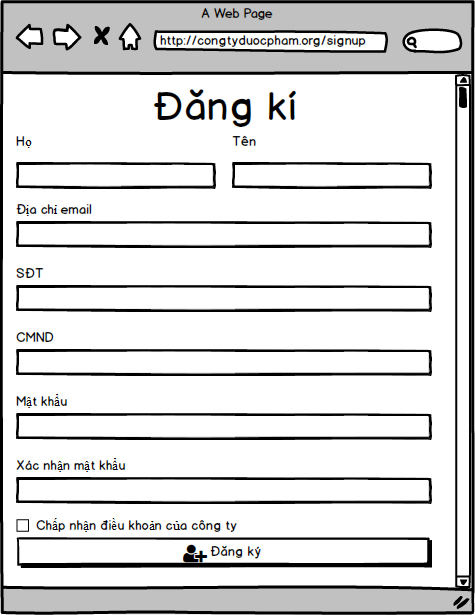


## **Giao diện**

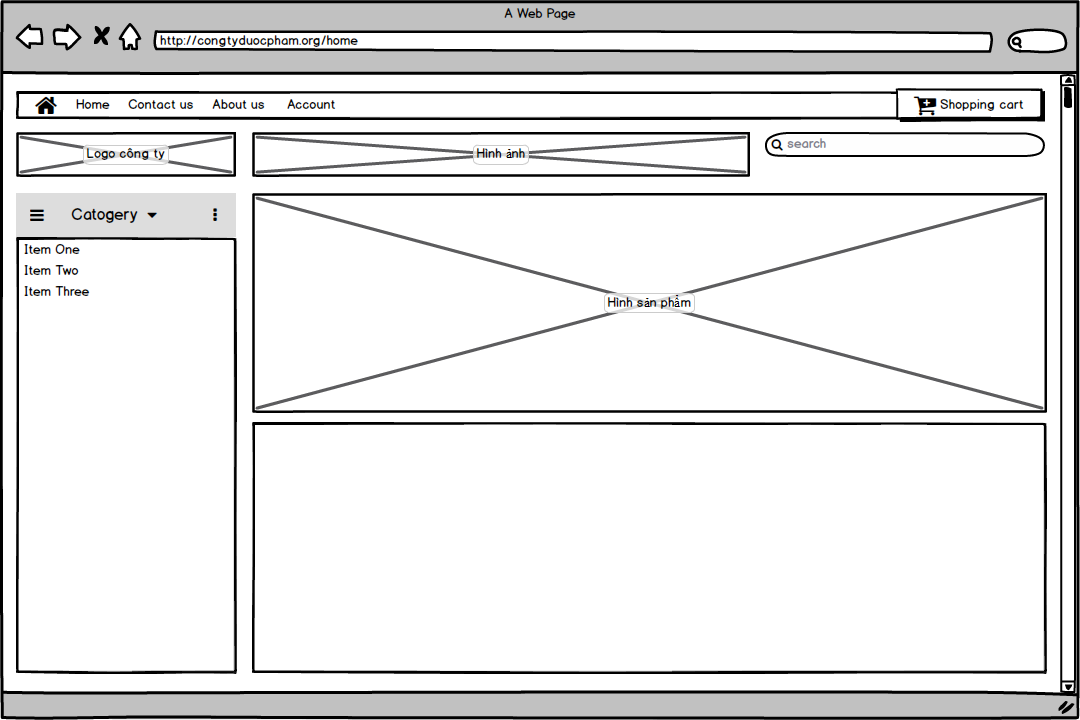
### **Đăng nhập**



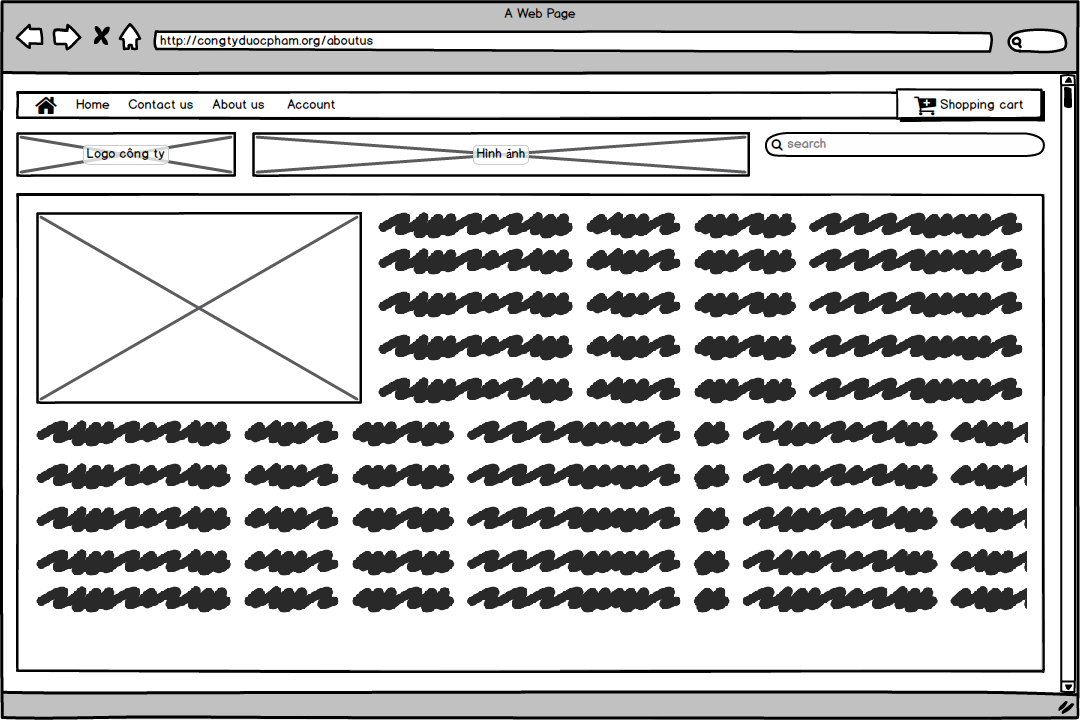
### **Đăng kí**



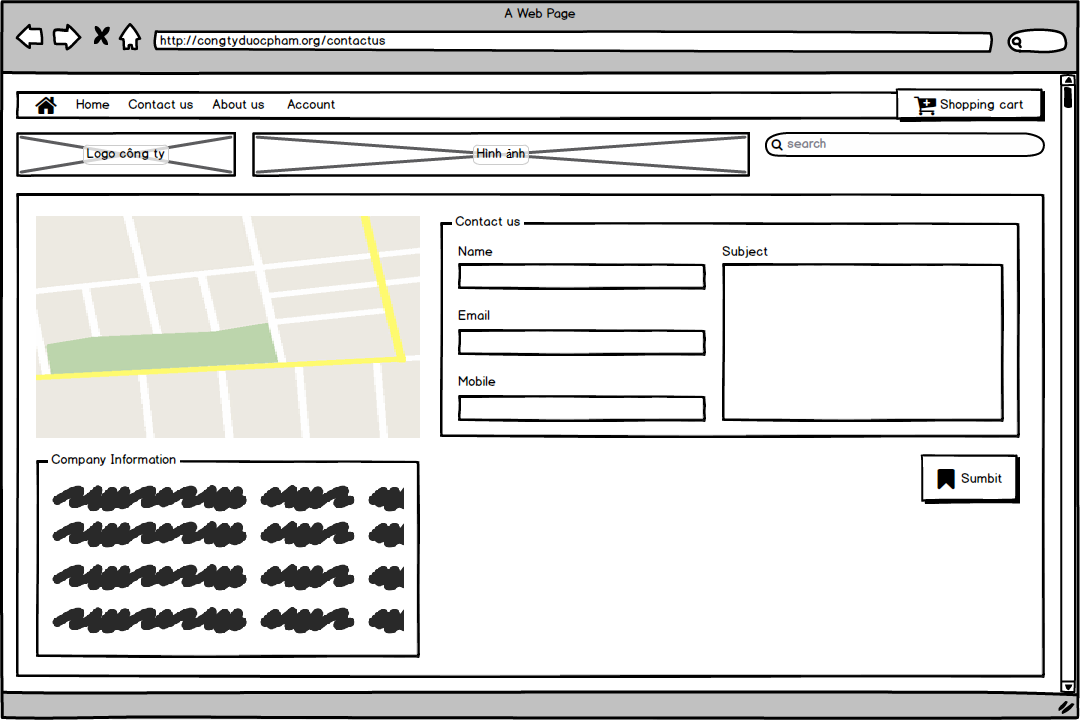
### **Home**



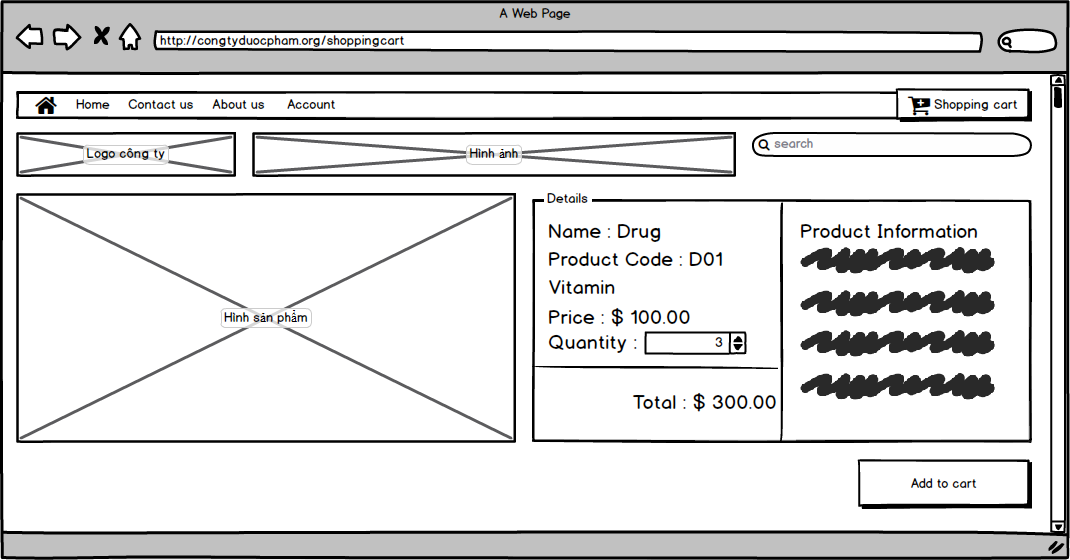
### **About us**



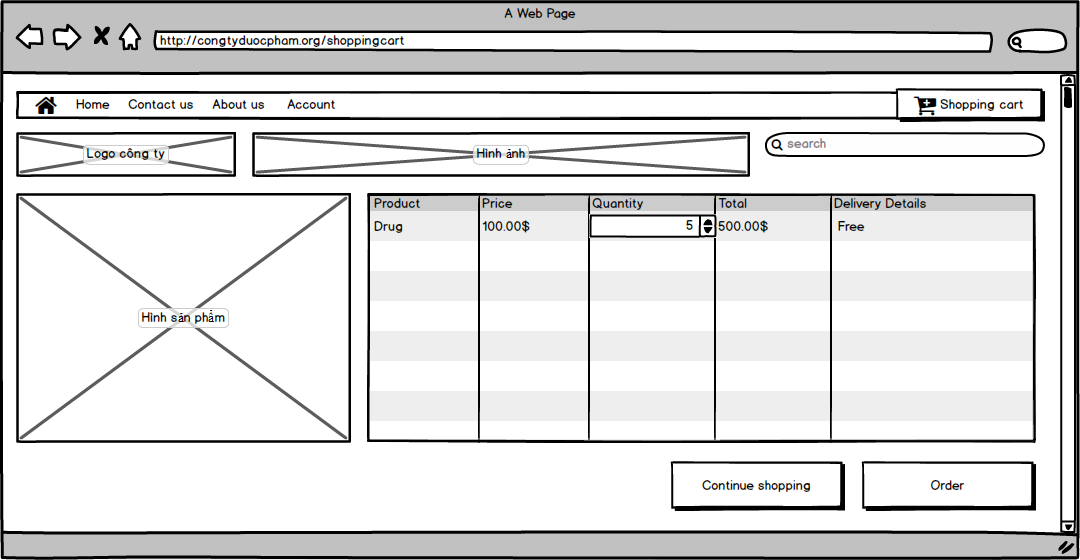
### **Contact us**



### **Product details**



### **Shopping Cart**



## **Mô hình tính toán**

**Function Point:** 182

**AVC:** 55(C#) + 13(SQL) = 68

**LOC =** AVC \* FP = 68 \* 167 = 12376

### **Mô hình Application Composition**

**Công thức:** **E = (NAP x (1 - %reuse/100)) / PROD** = 2.45 (người/tháng)

NAP = 14

reuse = 30%

PROD = 4

**Thời gian triển khai:**

**D = x =** 2.5 x = 3.5(tháng)

**Số người:** **P = E / D =** 2.45 / 3.5 = 1 (người)

**Mức độ hiệu quả:** **KLOC/E** = 12 / 2.45 = 505%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **For Screens** | | | | **For Reports** | | | |
|  | **Number and source of data tables** | | |  | **Number and source of data tables** | | |
| **Number of views contained** | Total < 4 (<2 server,  < 3 client) | Total < 8 (2-3 server,  3-5 client) | Total 8+ (>3 server, >5 client) | **Number of sections Contained** | Total < 4 (<2 server,  < 3 client) | Total < 8  (2-3 server, 3-5 client) | Total 8+ (>3 server, >5 client) |
| < 3 |  |  |  | 0 or 1 | **Simple** |  |  |
| 3 – 7 |  | **Medium** |  | 2 or 3 |  |  |  |
| 6+ |  |  |  | 4 + |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Object type** | **Simple** | **Medium** | **Difficult** |
| Screen |  | **2** |  |
| Report | **2** |  |  |
| 3GL component |  |  | **10** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Developer’s experience and capability** | **Very low** | **Low** | **Nominal** | **High** | **Very high** |
| **Case maturity and capability** | **Very low** | **Low** | **Nominal** | **High** | **Very high** |
| **PROD ( NOP/month)** | **4** |  |  |  |  |

### **Mô hình Early Design**

**Công sức: E = a x Sb x M =** 36656.26 (người/tháng)

a = 2.94

S = 12

M = 6 \* 6 \* 4 \* 5 = 720

b = 1.01 + 0.01 \* (3.24+2.43+3.38+1.98+2.73) = 1.1476

**Thời gian triển khai:**

**D = x =** 2.5 x = 135,61 (Tháng)

**Số người: P = E / D =** 36656.26 / 135.61 = 270 (người)

**Mức độ hiệu quả:** **KLOC / E =** 12 / 36656.26 = 0.03%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cost driver** | **Very low** | **Low** | **Normal** | **High** | **Very High** | **Extra High** |
| Độ tin cậy và phức tạp của sản phẩm |  |  |  |  | 5 |  |
| Yêu cầu sử dụng lại |  | 2 |  |  |  |  |
| Độ khó nền tảng |  | 2 |  |  |  |  |
| Trải nghiệm sự phức tap của cá nhân |  |  |  | 4 |  |  |
| Năng lực cá nhân |  |  | 3 |  |  |  |
| Yêu cầu lịch trình |  | 2 |  |  |  |  |
| Các phương tiện hỗ trợ nhóm |  |  | 3 |  |  |  |

### **Mô hình Reuse**

**Công sức: E = (ASLOC \* AT/100) / ATPROD =** 928.2

ASLOCT = FP \* AVC =182 \* 68 = 12376

AT = 30%

ATPROD = 4

**Thời gian triển khai:**

**D = x**  = 2.5 x = 34 (tháng)

**Số người:** **P = E / D** = 928.2 / 34 = 27 (người)

**Mức độ hiệu quả:** **KLOC/E** = 12 / 928.2 = 0.013 %

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Developer’s experience and capability** | **Very low** | **Low** | **Nominal** | **High** | **Very high** |
| **Case maturity and capability** | **Very low** | **Low** | **Nominal** | **High** | **Very high** |
| **PROD ( NOP/month)** | **4** |  |  |  |  |

### **Mô hình Post – architecture**

**Công sức:** **E = a x x M =** 40 (người/tháng)

M = 0.7609

A = 2.94

S(KLOC) = 12376

B = 1.01 + 0.01 \* (3.24+2.43+3.38+1.98+2.73) = 1.1476

**Thời gian triển khai:**

**D = x**  = 2.5 x = 10 (tháng)

**Số người:** **P = E / D** = 40 / 10 = 4 (người)

**Mức độ hiệu quả:** **KLOC/E** = 12 / 40 = 30%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cost driver** | **Very low** | **Low** | **Normal** | **High** | **Very High** | **Extra High** |
| **Đặc trưng về sản phẩm** | | | | | | |
| Yêu cầu vệ độ tin cậy của SP |  |  |  | 1.15 |  |  |
| Kích thước dữ liệu |  |  |  | 1.09 |  |  |
| Độ phức tạp của SP |  |  | 1.00 |  |  |  |
| Yêu cầu về tính tái sử dụng |  |  | 1.00 |  |  |  |
| Tài liệu cần thiết |  |  |  | 1.06 |  |  |
| **Đặc trưng về phần cứng** | | | | | | |
| Ràng buộc về thời gian |  |  |  |  | 1.31 |  |
| Ràng buộc bộ nhớ |  |  |  |  | 1.21 |  |
| Biến động nền tảng |  | 0.87 |  |  |  |  |
| **Đặc trưng về con người** | | | | | | |
| Khả năng phân tích |  |  | 1.00 |  |  |  |
| Khả năng lập trình |  |  |  | 0.87 |  |  |
| Kinh nghiệm ứng dụng |  |  |  | 0.89 |  |  |
| Kinh nghiệm nền tảng |  |  |  | 0.92 |  |  |
| EXP về Ngôn ngữ và công cụ |  |  |  | 0.88 |  |  |
| Sự kết nói cá nhân |  |  |  | 0.92 |  |  |
| **Đặc trưng về Project** | | | | | | |
| Sử dụng các công cụ phần mềm |  |  |  |  | 0.72 |  |
| Phát triển đa trang web |  |  | 1.00 |  |  |  |
| Yêu cầu kế hoạch phát triển |  |  |  | 1.00 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **W(i)** | **Very low** | **Low** | **Normal** | **High** | **Very High** | **Extra High** |
| Tính tiên phong |  | 3.24 |  |  |  |  |
| Tính linh hoạt phát triển |  |  |  | 2.43 |  |  |
| Kiến trúc giải pháp rủi ro |  | 3.38 |  |  |  |  |
| Sự gắn kết đồng đội |  |  |  | 1.98 |  |  |
| Qúa trình phát triển |  |  | 2.73 |  |  |  |

# **SQL Injection**

# **I. Khái niệm**

* **SQL Injection** là một kỹ thuật tấn công vào câu lệnh cơ sở dữ liệu thông qua chèn các đoạn mã độc vào SQL để đánh lạc hướng xử lý với mục đích phá hủy dữ liệu hoặc ăn cắp.
* **SQL injection** có thể cho phép những kẻ tấn công thực hiện các thao tác như một người quản trị web, trên cơ sở dữ liệu của ứng dụng.
* **SQL injection** thường được biết đến như là một vật trung gian tấn công trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu có cấu trúc như [SQL Server](http://toidicode.com/tu-hoc-mssql-20), [MySQL](http://toidicode.com/tu-hoc-mysql-database-19), [Oracle](http://toidicode.com/tu-hoc-oracle-database-21),... .

-Có 2 loại tấn công:

+Tấn công về mệnh đề luôn đúng.

+Tấn công phá hoại trực tiếp dữ liệu.

# **II. Cách thức hoạt động**

* Cơ chế tấn công **SQL injection – SQLI** là cách thức tận dụng hoặc khai thác triệt để những khuyết điểm, thiếu sót về mặt công nghệ được sử dụng để xây dựng **website**, và thông thường hacker sẽ kết hợp với những lỗ hổng trong quy trình bảo mật cơ sở dữ liệu.
* **SQL Injection** được thực hiện bằng thông qua các trường nhập dữ liệu của ứng dụng để thêm vào các đoạn mã SQL nguy hại.

# **1. Tấn công về mệnh đề luôn đúng**

# **Nội dung:**

Những lỗi lập trình này thường xảy ra ở các chức năng tìm kiếm hoặc đăng nhập tài khoản. Ví dụ một câu lệnh SQL như sau:

* **Trường hợp bình thường:** Nếu ta thay **username = admin** thì dữ liệu sẽ trả về username có dữ liệu là **admin**.

**SELECT \* FROM users WHERE username = ‘$name’;**

* **Trường hợp bị tấn công:** Nếu như thay đổi **username = admin** thành **username = ‘ ‘ OR ‘1’ = ‘1’** thì câu truy vấn của chúng ta thành mệnh đề luôn đúng bởi vì 1=1 luôn đúng và kết hợp với mệnh đề OR.

**SELECT \* FROM users WHERE username = ‘ ’ OR ‘1’ = ’1’ ;**

# **Đánh giá:**

* **Mục đích:** Dùng để đánh cắp tài khoản cá nhân của người khác.
* **Mức độ thiệt hại:** Nhỏ, có thể khôi phục lại được.

# **2. Tấn công phá hoại trực tiếp**

# **Nội dung:**

Cũng là câu lệnh đúng trên nhưng ta không nhập **OR ‘1’ = ‘1’** nữa mà thêm như sau:

**SELECT \* FROM users WHERE username = ‘$name ’ ; DROP TABLE users;**

* Trường hợp này gồm 2 câu truy vấn, câu đầu không có gì cả, câu 2 sẽ xóa tất cả các dữ liệu của user.

# **Đánh giá:**

* **Mục đích:** Nhằm xóa dữ liệu tài khoản của người khác.
* **Mức độ thiệt hại:** Lớn, dữ liệu không thể khôi phục.

# **III. Cách ngăn chặn**

* **Loại bỏ các kí tự meta** như '"/\ và các kí tự extend như NULL, CR, LF, ... trong các string nhận được từ: input do người dùng nhập vào các tham số từ URL các giá trị từ cookie
* Đối với các giá trị numeric, hãy **chuyển nó sang integer** trước khi query SQL, hoặc dùng ISNUMERIC để chắc chắn nó là một số integer.
* **Thay đổi "Startup and run SQL Server"** dùng mức low privilege user trong tab SQL Server Security.
* **Xóa các stored procedure** trong database master mà không dùng như:
* xp\_cmdshell
* xp\_startmail
* xp\_sendmail
* sp\_makewebtask
* **Lọc dữ liệu từ người dùng:** Cách phòng chống này tương tự như [XSS](https://toidicodedao.com/2016/10/18/lo-hong-bao-mat-xss/). Ta sử dụng filter để lọc các kí tự đặc biệt (; ” ‘) hoặc các từ khoá (SELECT, UNION) do người dùng nhập vào.
* **Sử dụng thư viện/function** được cung cấp bởi framework. Viết lại từ đầu vừa tốn thời gian vừa dễ sơ sót.
* **Không cộng chuỗi để tạo SQL:** Sử dụng parameter thay vì cộng chuỗi. Nếu dữ liệu truyền vào không hợp pháp, SQL Engine sẽ tự động báo lỗi, ta không cần dùng code để check.
* **Sử dụng các Parameters** khi làm việc với object SqlCommand (hoặc OleDbCommand) chứ không sử dụng các câu lệnh SQL trực tiếp. Khi đó .NET sẽ tự động validate kiểu dữ liệu, nội dung dữ liệu trước khi thực hiện câu lệnh SQL.
* **Phân quyền rõ ràng trong DB:** Nếu chỉ truy cập dữ liệu từ một số bảng, hãy tạo một account trong DB, gán quyền truy cập cho account đó chứ **đừng dùng account root hay sa**. Lúc này, dù hacker có inject được sql cũng không thể đọc dữ liệu từ các bảng chính, sửa hay xoá dữ liệu.
* **Backup dữ liệu thường xuyên**: Dữ liệu phải thường xuyên được backup để nếu có bị hacker xoá thì ta vẫn có thể khôi phục được.

**Ngăn chặn trong ASP.NET :** có cách ngăn chặn đơn giản

* Là sử dụng các Parameters khi làm việc với object SqlCommand (hoặc OleDbCommand) chứ không sử dụng các câu lệnh SQL trực tiếp. Khi đó .NET sẽ tự động validate kiểu dữ liệu, nội dung dữ liệu trước khi thực hiện câu lệnh SQL.
* Ngoài ra, cũng cần kiểm soát tốt các thông báo lỗi. Và mặc định trong ASP.NET là thông báo lỗi sẽ không được thông báo chi tiết khi không chạy trên localhost.

**Tài liệu tham khảo**

How I hacked PacketStorm (Rain Forest Puppy) <http://www.wiretrip.net/rfp/p/doc.asp?id=42&iface=6>

Great article on gathering information from ODBC error messages

[http://www.blackhat.com/presentations/win-usa 01/Litchfield/BHWin01Litchfield.doc](http://www.blackhat.com/presentations/win-usa%2001/Litchfield/BHWin01Litchfield.doc)  
A good summary of SQL Injection on various SQL Server on

<http://www.owasp.org/asac/input_validation/sql.shtml>  
Senseport's article on reading SQL Injection

<http://www.sensepost.com/misc/SQLinsertion.htm>  
Khác:  
<http://www.digitaloffense.net/warga.../IOWargames.ppt>  
<http://www.wiretrip.net/rfp/p/doc.asp?id=7&iface=6>  
<http://www.wiretrip.net/rfp/p/doc.asp?id=60&iface=6>

**Fun fact:**

* Tại thời điểm này thì nó cũng đã giảm hẳn, nhưng vẫn còn.
* Do gần đây các khái niệm framework đã và đang được sử dụng rất nhiều (Các framework đều đã được test cẩn thận để phòng tránh các lỗi, trong đó có SQL Injection).

# **CHƯƠNG 5. QUẢN LÝ CẤU HÌNH**

# **Trello**

# **Trello là gì ?**

* Công cụ để phối hợp công việc hiệu quả giúp cho mọi người trong team
* Chỉ cần nhìn qua là biết được có những đầu việc nào, ai đang làm gì, và làm đến giai đoạn nào rồi
* Trello là một công cụ Freemium, tức là bạn có thể xài miễn phí cho đến khi cần những tích năng nâng cao và đặc biệt chỉ có bản trả tiền mới có thì bạn mới phải trả

# **Những thành phần cơ bản của Trello**

* **Card :** là một thẻ thông tin có
* Title : tiêu đề
* Description : dùng để lưu trữ công việc.
* Card : có thể dùng để ghi
* Feature (tính năng sản phẩm)
* Task (công việc)
* Question (câu hỏi),
* Note (ghi chú)
* Mỗi card sẽ có checklist để bạn chia đầu việc nhỏ hơn.
* Bạn có thể comment, attach hình ảnh trên card để thuận tiện cho việc trao đổi giữa các thành viên.
* **List :** là một danh sách tổng hợp các card thuộc chung 1 tình trạng hoặc 1 tính chất khác nhau.
* **Board :** là một bảng thường tương đương với 1 project hoặc 1 mảng công việc. Trong một board có thể có nhiều List để chia tình trạng của những đầu việc của project đó cho tiện theo dõi.
* **Organization :** là tổng hợp những board và member của 1 công ty

# **Trello có gì hay**

**Simple**

* Tính đơn giản của Trello còn nằm ở chỗ những thao tác trong đó cũng được đơn giản hóa cực kỳ:
* Khi 1 task được làm xong, chỉ cần kéo card từ cột Doing sang Done. Không phải ngồi gõ, đổi tình trạng như những công cụ quản lý công việc khác.
* Để nhập một card (task) mới
* Click vào trong List To-Do để bắt đầu ghi chú title của card cần làm. Như vậy khi ghi chú công việc sẽ rất nhanh chóng và hiệu quả.

**Sophisticated**

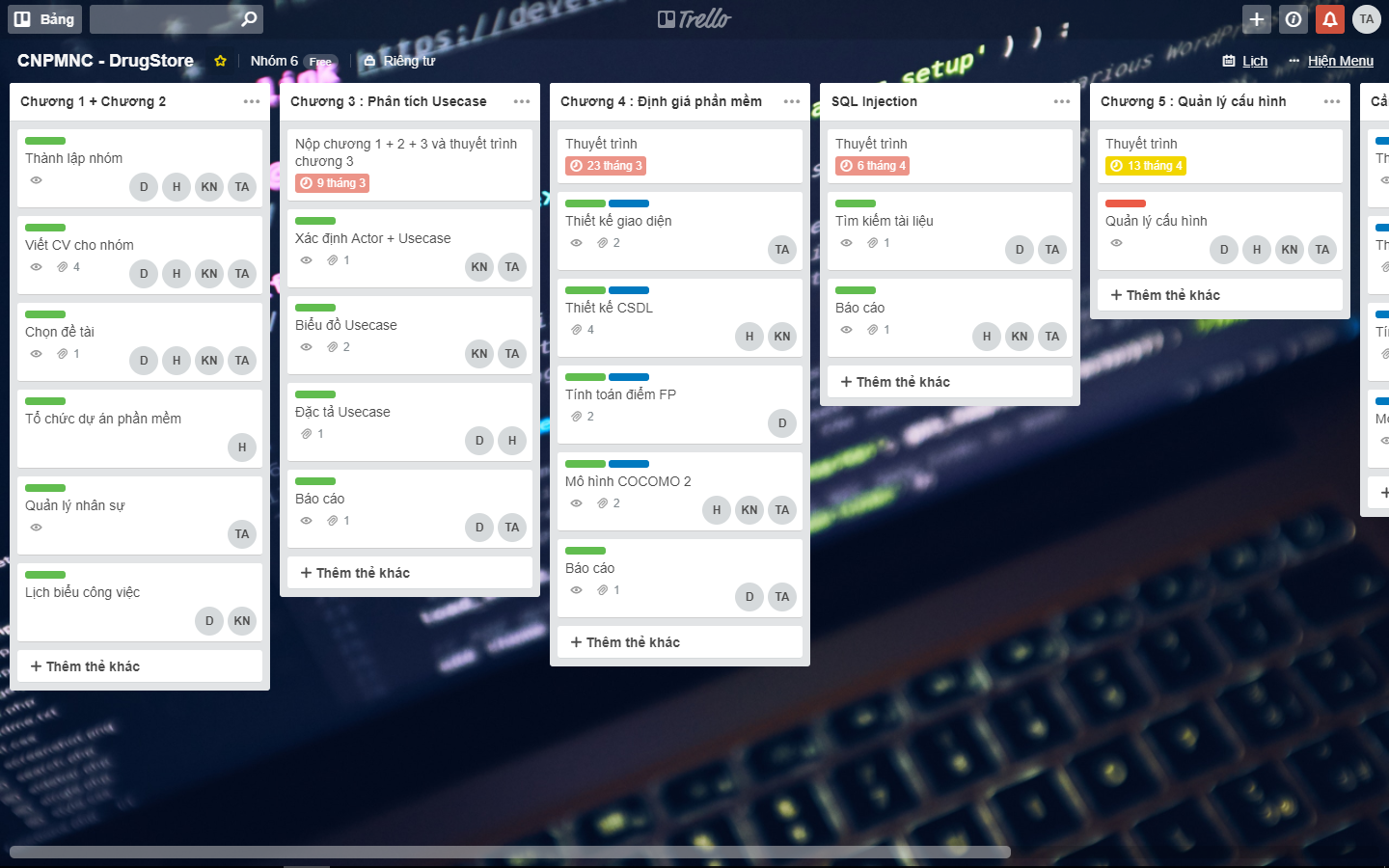
* Mặc dù sử dụng rất dễ dàng, nhưng Trello cũng tạo điều kiện quản lý công việc hiệu quả thông qua hàng loạt những tính năng và tiện ích – rất tinh vi nhưng không làm bạn rối.

**Flexible**

* Linh hoạt ở chỗ là có thể hỗ trợ quản lý theo cách bạn muốn
* Không bắt ép phải theo một quy chuẩn nào cả.
* Có thể dùng Trello cho nhiều thứ khác nhau chứ không phải chỉ là quản lý dự án thông thường.

**Free**

* Phiên bản Free vẫn rất đầy đủ và thoải mái để sử dụng cho tất cả những nhu cầu thông thường.



**So sánh giữa Trello và Git Hub**

|  |  |  |
| --- | --- | --- |
|  | **GitHub** | **Trello** |
| Pros | Pull requests là điều cần thiết để phát triển và xây dựng thành công, xem những gì user đã làm và thảo luận về project. | * Tốt cho tất cả các loại quy trình công việc, bao gồm, nhưng không giới hạn đến các sản phẩm có liên quan đến nhiều thứ * Tổ chức rất linh hoạt với ba thành phần: board , list. label * Thứ tự của những thành phần trong danh sách là tính năng tuyệt vời. |
| Cons | * Trung tâm của quy trình làm việc của việc phát triển không mang lại nhiều giá trị cho phần còn lại của công việc, bởi vì git không hề quan tâm đến điều đó. * Không có kết cấu linh hoạt, những phần quan trọng thì rất rườm rà khi code. | * Không có pull request và hội nhập yếu ("chỉ cần post 1 link!") * Đoạn code nhìn không được đẹp (không có cú pháp làm nổi bật). |
| Neutral | Các vấn đề cá nhân là một nơi khá tốt để thảo luận, nhưng không phải là một tính năng tuyệt vời như PRs. | Card có tốt cho thảo luận khi tranh luận vấn đề?  Có lẽ, nhưng tôi không chắc. Nhận xét chuyển theo hướng ngược lại, mọi thứ đều hẹp hơn vì đó là một phương thức thay vì một trang, và bạn chỉ có thể đính kèm hình ảnh vào thẻ mình, chứ không phải là inline trong comment |