**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

**TÌM HIỂU VỀ MÔ HÌNH QUY TRÌNH WATERFALL VÀ VẬN DỤNG TRONG DỰ ÁN THIẾT KẾ VÀ PHÁT TRIỂN PHẦN MỀM QUẢN LÝ HOẠT ĐỘNG GIẢNG DẠY CỦA TRUNG TÂM TIẾNG ANH ARABICA**

|  |  |  |
| --- | --- | --- |
| **GVHD**: | ThS. Nguyễn Đức Lưu |  |
| **Nhóm - Lớp:** | 09 - 20222IT6082016 |  |
| **Sinh viên thực hiện:** | 1. Lê Quỳnh Giang 2. Vũ Thị Huế 3. Phan Nhật Huyền 4. Vũ Thị Bích Huyền 5. Cam Thị Liễu 6. Nguyễn Minh Quang 7. Hoàng Văn Trung | 2021601194  2021601676  2021602302  2021601408  2021601210  2021600455  2021602078 |

***Hà Nội – 2023***

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 1](#_Toc137248213)

[DANH MỤC BẢNG 2](#_Toc137248214)

[LỜI CẢM ƠN 3](#_Toc137248215)

[MỞ ĐẦU 4](#_Toc137248216)

[1. Tên đề tài 4](#_Toc137248217)

[2. Lý do chọn đề tài 4](#_Toc137248218)

[3. Mục tiêu của đề tài 6](#_Toc137248219)

[4. Phạm vi thực hiện 6](#_Toc137248220)

[5. Bố cục đề tài 6](#_Toc137248221)

[CHƯƠNG 1. TỔNG QUAN VỀ DỰ ÁN PHẦN MỀM 8](#_Toc137248222)

[1.1. Giới thiệu chung 8](#_Toc137248223)

[1.2. Tổng quan về quy trình phát triển phần mềm 8](#_Toc137248224)

[1.2.1. Quy trình phát triển phần mềm 8](#_Toc137248225)

[1.2.2. Mô hình quy trình 11](#_Toc137248226)

[1.3 Mô hình quy trình thác nước 11](#_Toc137248227)

[1.3.1. Giới thiệu về mô hình 11](#_Toc137248230)

[1.3.2. Các hoạt động của mô hình 12](#_Toc137248231)

[1.3.3. Đánh giá 13](#_Toc137248232)

[1.3.3.1. Ưu điểm 14](#_Toc137248233)

[1.3.3.2. Nhược điểm 14](#_Toc137248234)

[1.3.4 Tính ứng dụng của mô hình thác nước 15](#_Toc137248235)

[1.3.5. Kết luận 15](#_Toc137248236)

[1.4 Cấu trúc tài liệu đặc tả yêu cầu phần mềm 16](#_Toc137248237)

[1.5 Công nghệ áp dụng và phương pháp trong phát triển Arabica 18](#_Toc137248238)

[1.5.1. Giới thiệu công cụ Rational Rose 18](#_Toc137248239)

[1.5.2. Giới thiệu công cụ CASE Studio 18](#_Toc137248240)

[1.5.3. Giới thiệu công cụ Balsamiq Mockups 18](#_Toc137248241)

[1.5.4. Phương pháp phân tích và thiết kế phần mềm 19](#_Toc137248242)

[1.6 Kết luận chương 1 19](#_Toc137248243)

[CHƯƠNG 2. PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU 20](#_Toc137248244)

[2.1 Quy trình kỹ thuật yêu cầu 20](#_Toc137248245)

[2.1.1. Giới thiệu về quy trình kỹ thuật yêu cầu 20](#_Toc137248246)

[2.1.2 Các hoạt động trong quy trình kỹ thuật yêu cầu 20](#_Toc137248247)

[2.1.2.1 Phát hiện các yêu cầu phần mềm 20](#_Toc137248248)

[2.1.2.2 Phân tích các yêu cầu phần mềm 22](#_Toc137248249)

[2.1.2.3 Mô tả các yêu cầu phần mềm 23](#_Toc137248250)

[2.1.2.4 Thẩm định yêu cầu phần mềm 24](#_Toc137248251)

[2.2 Các tác nhân tham gia vào quy trình kỹ thuật yêu cầu 24](#_Toc137248252)

[2.2.1 Giới thiệu về các tác nhân tham gia quy trình kỹ thuật yêu cầu 24](#_Toc137248253)

[2.2.1.1 Nhà tài trợ/Chủ Đầu tư 25](#_Toc137248254)

[2.2.1.2 Người quản lý dự án 25](#_Toc137248255)

[2.2.1.3 Người dùng 25](#_Toc137248256)

[2.2.1.4 Các bên liên quan khác 25](#_Toc137248257)

[2.2.1.5 Người phát triển 25](#_Toc137248258)

[2.2.1.6 Người kiểm thử (Tester) 26](#_Toc137248259)

[2.3 Đặc tả yêu cầu phần mềm cho trung tâm tiếng Anh Arabica 26](#_Toc137248260)

[2.3.1 Giới thiệu 26](#_Toc137248261)

[2.3.1.1 Mục đích của tài liệu yêu cầu 26](#_Toc137248262)

[2.3.1.2 Phạm vi sản phẩm 27](#_Toc137248263)

[2.3.2 Mô tả chung 27](#_Toc137248264)

[2.3.2.1 Các tác nhân hệ thống 27](#_Toc137248265)

[2.3.1.2 Các yêu cầu chức năng 28](#_Toc137248266)

[2.3.3. Quy trình nghiệp vụ 33](#_Toc137248267)

[2.3.3.1 Quy trình sử dụng phần mềm 33](#_Toc137248268)

[2.3.4. Yêu cầu chức năng 34](#_Toc137248269)

[2.3.4.1 Chức năng đăng nhập 34](#_Toc137248270)

[Hình 2.6: Biểu đồ trình tự chức năng đăng nhập 36](#_Toc137248271)

[2.3.4.2. Chức năng đăng ký 36](#_Toc137248272)

[2.3.4.3. Chức năng cập nhật thông tin cá nhân 38](#_Toc137248273)

[2.3.4.4. Chức năng xem thông tin khóa học 40](#_Toc137248274)

[2.3.4.5. Chức năng đăng ký khóa học 42](#_Toc137248275)

[2.3.4.6. Chức năng quản lý bài giảng 44](#_Toc137248276)

[2.3.4.7. Chức năng quản lý giảng viên 50](#_Toc137248277)

[2.3.4.8. Chức năng quản lý học viên 53](#_Toc137248278)

[2.3.5 Các yêu cầu phi chức năng 56](#_Toc137248280)

[2.3.5.1 Yêu cầu thực thi 56](#_Toc137248281)

[2.3.5.2 Yêu cầu an toàn 56](#_Toc137248282)

[2.3.5.3 Yêu cầu bảo mật 56](#_Toc137248283)

[2.3.5.4 Các luật vận hành 59](#_Toc137248284)

[2.3.5.5 Các đặc điểm chất lượng phần mềm 59](#_Toc137248285)

[2.4 Biểu đồ lớp thực thể 60](#_Toc137248286)

[2.5 Kết luận chương 2 60](#_Toc137248287)

[CHƯƠNG 3. THIẾT KẾ PHẦN MỀM 61](#_Toc137248288)

[3.1. Tổng quan về thiết kế phần mềm 61](#_Toc137248289)

[3.1.1. Vai trò và các nhiệm vụ trong hoạt động thiết kế phần phần 61](#_Toc137248290)

[3.1.2. Chiến lược thiết kế phần mềm 62](#_Toc137248295)

[3.1.3*.* Công cụ hỗ trợ thiết kế phần mềm 64](#_Toc137248296)

[3.2. Tài liệu thiết kế phần mềm 65](#_Toc137248297)

[3.2.1. Giới thiệu 65](#_Toc137248298)

[3.2.1.1. Mục đích tài liệu 65](#_Toc137248299)

[3.2.1.2. Phạm vi tài liệu 65](#_Toc137248300)

[3.2.1.3. Tài liệu tham khảo 65](#_Toc137248301)

[3.2.1.4. Bố cục tài liệu thiết kế phần mềm 65](#_Toc137248302)

[3.2.2. Thiết kế kiến trúc phần mềm 66](#_Toc137248303)

[3.2.2.1. Mô hình kiến trúc 66](#_Toc137248304)

[3.2.2.2. Mô tả kiến trúc 71](#_Toc137248305)

[3.2.3. Thiết kế dữ liệu 72](#_Toc137248306)

[3.2.3.1 Thực thể TAIKHOAN 72](#_Toc137248307)

[3.2.3.2 Thực thể KHOAHOC 73](#_Toc137248308)

[3.2.3.3 Thực thể BAIGIANG 74](#_Toc137248309)

[3.2.3.4 Thực thể HOCVIEN 75](#_Toc137248310)

[3.2.3.5 Thực thể GIANGVIEN 76](#_Toc137248311)

[3.2.4. Thiết kế các thành phần phần mềm 77](#_Toc137248312)

[3.2.4.1. Mã giả cho chức năng Đăng nhập 77](#_Toc137248313)

[3.2.4.2. Mã giải cho chức năng Đăng ký 77](#_Toc137248314)

[3.2.4.3. Mã giải cho chức năng Xem khoá học 78](#_Toc137248315)

[3.2.4.4. Mã giải cho chức năng Mua khoá học 79](#_Toc137248316)

[3.2.4.5. Mã giải cho chức năng Bảo trì tài khoản 80](#_Toc137248317)

[3.2.5. Thiết kế giao diện người sử dụng 82](#_Toc137248318)

[3.2.5.1. Chức năng Đăng ký 82](#_Toc137248319)

[3.2.5.2. Chức năng Xem khoá học 83](#_Toc137248320)

[3.2.5.3. Chức năng Mua khoá học 84](#_Toc137248321)

[3.2.5.4. Chức năng Đăng nhập 85](#_Toc137248322)

[3.2.5.5. Chức năng Quản lý tài khoản 86](#_Toc137248323)

[3.2.5.6. Hình dung tất cả các màn hình 88](#_Toc137248324)

[3.3 Kết luận chương 3 89](#_Toc137248325)

[KẾT LUẬN 90](#_Toc137248326)

[1. Kết quả đạt được 90](#_Toc137248327)

[2. Hướng phát triển: 90](#_Toc137248328)

[TÀI LIỆU THAM KHẢO 91](#_Toc137248329)

# **DANH MỤC HÌNH ẢNH**

[Hình 1.1: Các pha trong mô hình thác nước 12](#_Toc137066702)

[Hình 2.1: Sơ đồ use case tổng quát 31](#_Toc137066703)

[Hình 2.2: Sơ đồ phân rã use case phía front end 31](#_Toc137066704)

[Hình 2.3: Sơ đồ phân rã use case “Quản trị hệ thống” 32](#_Toc137066705)

[Hình 2.4: Sơ đồ phân rã use case “Giảng viên” 32](#_Toc137066706)

[Hình 2.5: Biểu đồ hoạt động sử dụng phần mềm 34](#_Toc137066707)

[Hình 2.6: Biểu đồ trình tự chức năng đăng nhập 36](#_Toc137066708)

[Hình 2.7: Biểu đồ trình tự chức năng đăng ký 38](#_Toc137066709)

[Hình 2.8: Biểu đồ trình tự chức năng cập nhật thông tin cá nhân 40](#_Toc137066710)

[Hình 2.9: Biểu đồ trình tự chức năng xem chi tiết sản phẩm 42](#_Toc137066711)

[Hình 2.10: Biểu đồ trình tự chức năng xem cẩm nang 44](#_Toc137066712)

[Hình 2.11: Biểu đồ trình tự thêm, sửa, xóa các bài giảng 49](#_Toc137066713)

[Hình 2.12: Biểu đồ trình tự quản lý giảng viên 52](#_Toc137066714)

[Hình 2.13: Biểu đồ trình tự thêm, xóa thông tin học viên 55](#_Toc137066715)

[Hình 2.14: Biểu đồ lớp thực thể 60](#_Toc137066716)

[Hình 3.1: Giao diện làm việc của Balasmiq Mockups 64](#_Toc137066717)

[Hình 3.2: Minh họa vai trò của kiến trúc phần mềm 66](#_Toc137066718)

[Hình 3.3: Mô hình client – server 70](#_Toc137066719)

[Hình 3.4: Thành phần cơ bản của mô hình MVC 71](#_Toc137066720)

[Hình 3.5: Luồng xử lý trong mô hình MVC 72](#_Toc137066721)

[Hình 3.6: Thực thể TAIKHOAN 73](#_Toc137066722)

[Hình 3.7: Thực thể KHOAHOC 74](#_Toc137066723)

[Hình 3.8: Thực thể BAIGIANG 75](#_Toc137066724)

[Hình 3.9: Thực thể HOCVIEN 76](#_Toc137066725)

[Hình 3.10: Thực thể GIANGVIEN 76](#_Toc137066726)

[Hình 3.11: Biểu đồ lớp Đăng kí 82](#_Toc137066727)

[Hình 3.12: : Biểu đồ tuần tự đăng kí 82](#_Toc137066728)

[Hình 3.13: : Biểu đồ lớp Xem khóa học 83](#_Toc137066729)

[Hình 3.14: Biểu đồ tuần tự đăng kí 83](#_Toc137066730)

[Hình 3.15: Biểu đồ lớp Mua khóa học 84](#_Toc137066731)

[Hình 3.16: Biểu đồ tuần tự Mua khóa học 84](#_Toc137066732)

[Hình 3.17: Biểu đồ lớp Đăng nhập 88](#_Toc137066733)

[Hình 3.18: Biểu đồ tuần tự Đăng nhặp 88](#_Toc137066734)

[Hình 3.19: Biểu đồ lớp Quản lý tài khoản 86](#_Toc137066735)

[Hình 3.20: Biểu đồ tuần tự Quản lý tài khoán 87](#_Toc137066736)

[Hình 3.21: Giao diện phía front-end 88](#_Toc137066737)

[Hình 3.22: Giao diện phía back-end](#_Toc137066738) 89

# **DANH MỤC BẢNG**

[Bảng 2.1: Mô tả thông tin về tác nhân và các chức năng tương ứng 28](#_Toc137247494)

[Bảng 2.2: Mô tả chức năng đăng nhập 34](#_Toc137247495)

[Bảng 2.3: Mô tả chức năng đăng ký 36](#_Toc137247496)

[Bảng 2.4: Mô tả chức năng cập nhật thông tin cá nhân 39](#_Toc137247497)

[Bảng 2.5: Mô tả chức năng xem thông tin khóa học 41](#_Toc137247498)

[Bảng 2.6: Mô tả chức năng đăng ký khóa học 43](#_Toc137247499)

[Bảng 2.7: Mô tả chức năng quản lý bài giảng 45](#_Toc137247500)

[Bảng 2.8: Mô tả chức năng quản lý giảng viên 50](#_Toc137247501)

[Bảng 2.9: Mô tả chức năng quản lý học viên 53](#_Toc137247502)

# **LỜI CẢM ƠN**

Lời đầu tiên, nhóm 9 chúng em xin gửi lời cảm ơn chân thành tới thầy hướng dẫn ThS.Nguyễn Đức Lưu. Trong quá trình học tập và thực hiện đề tài này, chúng em đã nhận được sự quan tâm giúp đỡ, hướng dẫn tận tình, tâm huyết của thầy. Những gì chúng em nhận được không chỉ dừng lại ở kiến thức môn học mà nhiều hơn thế đó là những lời khuyên, chia sẻ thực tế từ thầy. Chính nhờ phương pháp dạy học của thầy mà chúng em có cơ hội khám phá và phát huy khả năng của bản thân. Những buổi thuyết trình chính là cơ hội tuyệt vời giúp chúng em rèn luyện sự tự tin, kỹ năng giao tiếp, làm việc nhóm, … Đây cũng chính là hành trang quan trọng giúp chúng em tự tin bước chân vào môi trường làm việc thực tế.

Để hoàn thành được đề tài này, nhóm chúng em đã cùng nhau nghiên cứu, thảo luận, áp dụng những kiến thức được học trên lớp cùng với các nguồn tài liệu trên Internet và cả những trải nghiệm của bản thân. Chúng em rất mong sẽ nhận được những lời nhận xét, góp ý từ thầy cô và bạn đọc để đề tài này có thể hoàn thiện hơn nữa.

Chúng em xin chân thành cảm ơn!

***Nhóm sinh viên thực hiện***

# **MỞ ĐẦU**

## Tên đề tài

*“Tìm hiểu về mô hình quy trình Waterfall và vận dụng trong dự án thiết kế và phát triển phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica”.*

## 2. Lý do chọn đề tài

Ngày nay, ngành công nghiệp phần mềm trên thế giới đang rất phát triển. Nhiều người đã thống nhất rằng công nghệ phần mềm đã trở thành một mũi nhọn để con người tiến nhanh vào nên kinh tế tri thức. Tuy nhiên, việc tạo ra một sản phẩm phần mềm không phải là một công việc đơn giản và dễ dàng. Để làm một sản phẩm phức tạp và ít hữu hình này đòi hỏi những người tham gia phát triển phải có những phần kiến thức chuyên môn sâu – đủ để tạo ra được một sản phẩm phần mềm chất lượng, đáp ứng yêu cầu người dùng và mang lại hiệu quả kinh tế cao. Vì vậy kiến thức về công nghệ phần mềm là một phần rất cần thiết đối với những người bắt đầu tiếp cận với lĩnh vực công nghệ thông tin nói chung và đặc biệt là đối với sinh viên ngành Công nghệ thông tin và Kỹ thuật phần mềm nói riêng.

Việc làm các ứng dụng, Website không theo một quy định chung, không sử dụng một quy trình phát triển hệ thống thích hợp thì sẽ giảm chất lượng hệ thống và tăng chi phí xây dựng. Vì vậy, việc áp dụng các quy trình phát triển phần mềm vào để xây dựng hệ thống là việc vô cùng quan trọng. Có 1 số quy trình phát triển phổ biến nhất như: Thác nước, xoắn ốc, mô hình chữ V. Đối với các dự án nhỏ thì việc áp dụng quy trình thác nước mang lại hiệu quả lớn và hiệu quả.

Mô hình thác nước là một ví dụ của mô hình Sequential (Tuần tự). Trong mô hình này, hoạt động phát triển phần mềm được chia thành các giai đoạn khác nhau và từng giai đoạn bao gồm hàng loạt các nhiệm vụ và có các mục tiêu khác nhau. Ưu điểm của mô hình này là Đơn giản, dễ hiểu và sử dụng.

Vì các giai đoạn của mô hình thác nước cứng nhắc và chính xác, một pha được thực hiện một lần. Các tiêu chí đầu vào và đầu ra được xác định rõ ràng, do đó nó dễ dàng và có hệ thống để tiến hành chất lượng.

Có thể nói, đời sống càng phát triển, Tiếng Anh càng ngày ăn sâu vào từng khía cạnh trong đời sống của chúng ta. Do đó, việc biết Tiếng Anh là vô cùng cần thiết để chúng ta có thể hiểu cách giao tiếp của thời hiện đại, đọc được nhãn và công dụng của sản phẩm nước ngoài, tận dụng tối đa các công cụ như Google, Facebook, Grab… Nói một cách vui vui: “Học Tiếng Anh sẽ giúp đời thêm sang chảnh”. Chính vì thế, để đáp ứng nhu cầu học tập của tất cả mọi người, đã có rất nhiều trung tâm tiếng anh “mọc” lên trên khắp cả nước.

Cùng với đó, hiện nay, dưới thời đại của công nghệ số và khoa học kỹ thuật, việc phát triển phần mềm là hết sức phổ biến. Và để một trung tâm Tiếng Anh hoạt động hiệu quả nhất thì cần rất nhiều thứ như tài chính, cơ sở vật chất tốt, đội ngũ nhân viên, giảng viên,... đặc biệt, thứ không thể thiếu đó chính là các phần mềm hỗ trợ.

Tuy nhiên, không phải phần mềm nào cũng chính xác và làm hài lòng người sử dụng. Vậy, phần mềm nào phù hợp với trung tâm Tiếng Anh và làm thế nào để phát triển phần mềm tốt nhất? Câu hỏi thú vị này đã thôi thúc nhóm 9 chúng em thực hiện đề tài tìm hiểu mô hình thác nước và ứng dụng viết tài liệu đặc tả yêu cầu phần mềm, cụ thể ở đây là phần mềm quản lý hoạt động giảng dạy của trung tâm Tiếng Anh Arabica.

Tìm hiểu mô hình phát triển phần mềm đi sâu vào mô hình thác nước và ứng dụng viết tài liệu đặc tả yêu cầu phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica.

## 3. Mục tiêu của đề tài

* Tìm hiểu dự án thiết kế và phát triển phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica.
* Áp dụng những kiến thức đã học được từ học phần NMCNPM, tập trung đi sâu tìm hiểu mô hình thác nước và ứng dụng đặc tả phần mềm, sử dụng các công cụ Rational Rose, Case Studio, Balsamiq Mockups vào thiết kế và phát triển phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica.

## 4. Phạm vi thực hiện

* Về mô hình thác nước:
* Lịch sử hình thành: quá trình phát triển, sử dụng mô hình
* Mô tả: Mô tả chung về đặc trưng của mô hình
* Các giai đoạn của mô hình: Các giai đoạn cần thực hiện của mô hình
* Khi nào nên sử dụng mô hình: Khảo sát quy mô, mức độ dự án để căn cứ áp dụng mô hình
* Ưu điểm
* Nhược điểm
* Kết luận : Đánh giá mức áp dụng của mô hình, những cơ chế cải tiến so với một số mô hình khác
* Về tài liệu đặc tả phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica: khảo sát,mô tả hoạt động, liệt kê được các tác nhân và use case, vẽ được biểu đồ use case hệ thống.

## 5. Bố cục đề tài

*(Nội dung chính chia thành 3 chương)*

***Chương 1. Tổng quan về dự án phần mềm:***

Nội dung trình bày về: dự án phần mềm; mô hình quy trình thác nước; nghiên cứu và viết cấu trúc đặc tả yêu cầu phần mềm; công cụ; phương pháp, kỹ thuật áp dụng để phát triển phần mềm.

***Chương 2. Phân tích và đặc tả yêu cầu phần mềm:***

Nội dung trình bày về quá trình phân tích và đặc tả yêu cầu phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica.

***Chương 3. Thiết kế phần mềm:***

Nội dung trình bày về tổng quan về thiết kế phần mềm, thiết kế kiến trúc, thiết kế các thành phần phần mềm, thiết kế giao diện người dùng, thiết kế CSDL*.*

# **CHƯƠNG 1. TỔNG QUAN VỀ DỰ ÁN PHẦN MỀM**

## **Giới thiệu chung**

Trung tâm tiếng anh Arabica là trung tâm phục vụ mọi người về lĩnh vực học thuật, cụ thể là ngôn ngữ tiếng Anh. Với mong muốn đào tạo cho tất cả mọi người các kiến thức về tiếng Anh từ cơ bản đến nâng cao, trung tâm tiếng Anh Arabica đã được ra đời.

Trung tâm tiếng Anh Arabica luôn đề cao yêu cầu của người dùng cũng như của khách hàng đề ra, đồng thời góp phần cho quản trị viên dễ dàng quản lý các khóa học cũng như quản lý thời gian các khóa học, hoạt động giảng dạy của các lớp đang được diễn ra hoặc sắp diễn ra.

Để đáp ứng nhu cầu về học tập, quảng bá về các chương trình các khóa học tiếng Anh và phát triển phần mềm quản lý hoạt động giảng dạy trong thời đại công nghệ 4.0, Arabica đã có ý tưởng xây dựng phần mềm nền tảng Website. Điều này sẽ giúp Arabica mở rộng thị trường và tiết kiệm nhân công. Từ đó, Arabica tăng thêm khả năng cạnh tranh trên thị trường và có chỗ đứng vững trãi trong lĩnh vực học thuật.

## 1.2. Tổng quan về quy trình phát triển phần mềm

### 1.2.1. Quy trình phát triển phần mềm

Quy trình Vòng đời phát triển phần mềm (SDLC-Software Development Life Cycle) là một quá trình bao gồm một loạt các hoạt động được lên kế hoạch để phát triển hoặc thay đổi Sản phẩm Phần mềm. SDLC còn được gọi là Quy trình phát triển phần mềm.

SDLC là một quy trình được ngành công nghiệp phần mềm sử dụng để thiết kế, phát triển và kiểm tra phần mềm chất lượng cao. SDLC nhằm mục đích tạo ra một phần mềm chất lượng cao đáp ứng hoặc vượt quá mong đợi của khách hàng, hoàn thành trong thời gian và chi phí ước tính.

SDLC là một quy trình theo sau cho một dự án phần mềm, trong một tổ chức phần mềm, bao gồm một kế hoạch chi tiết mô tả cách phát triển, bảo trì, thay thế và thay đổi hoặc nâng cao phần mềm cụ thể.

SDLC xác định một phương pháp luận để cải thiện chất lượng của phần mềm và quá trình phát triển tổng thể.

SDLC là một khuôn khổ xác định các tác vụ được thực hiện ở mỗi bước trong quy trình phát triển phần mềm.

**Các hoạt động cơ bản của quy trình phát triển phần mềm:**

*Stage 1: Planning and Requirement Analysis (Lập kế hoạch và phân tích yêu cầu):* Được tiến hành ngay khi có nhu cầu về việc xây dựng phần mềm.

* Mục tiêu: Mô tả lại thế giới thực thông qua các mô hình (mô hình thế giới thực) trước khi thiết kế.
* Kết quả nhận: Danh sách các yêu cầu cùng các thông tin có liên quan.
* Kết quả chuyển giao:
* Mô hình xử lý (hệ thống các công việc trong thế giới thực cùng với quan hệ giữa chúng)
* Mô hình dữ liệu (hệ thống các loại thông tin được sử dụng trong thế giới thực cùng với quan hệ giữa chúng)
* Các mô hình khác (không gian, thời gian, con người…) nếu cần thiết.

*Stage 2: Defining Requirements (Xác định/Định nghĩa yêu cầu)*: Được tiến hành ngay sau khi kết thúc việc phân tích.

* Mục tiêu: Xác định chính xác các yêu cầu đặt ra cho phần mềm sẽ xây dựng.
* Kết quả nhận: Thông tin về hoạt động của thế giới thực.
* Kết quả chuyển giao: Danh sách các yêu cầu (công việc sẽ thực hiện trên máy tính) cùng với các thông tin miêu tả chi tiết về các yêu cầu (cách thức thực hiện trong thế giới thực)

*Stage 3: Designing the Product Architecture (Thiết kế kiến trúc sản phẩm)*: Được tiến hành ngay sau khi kết thúc việc xác định yêu cầu.

* Mục tiêu: Mô tả các thành phần của phần mềm (mô hình của phần mềm) trước khi tiến hành cài đặt.
* Kết quả nhận: Mô hình thế giới thực.
* Kết quả chuyển giao:
* Mô tả thành phần giao diện: các hàm nhập/xuất, cấu trúc dữ liệu nhập/xuất.
* Mô tả thành phần xử lý: các hàm kiểm tra xử lý.
* Mô tả thành phần dữ liệu: các hàm đọc/ ghi, tổ chức lưu trữ trên bộ nhớ phụ.

*Stage 4: Building or Developing the Product (Xây dựng/phát triển sản phẩm)*: Được tiến hành ngay sau khi kết thúc việc thiết kế.

* Mục tiêu: Tạo lập phần mềm theo yêu cầu.
* Kết quả nhận: Mô hình phần mềm.
* Kết quả chuyển giao: Chương trình nguồn của phần mềm với cấu trúc cơ sở dữ liệu tương ứng (nếu cần thiết) và chương trình thực hiện được trên máy tính (chương trình nguồn đã được biên dịch).

*Stage 5: Testing the Product (Kiểm tra sản phẩm)*: Được tiến hành ngay sau khi đã xây dựng phần mềm.

* Mục tiêu: Tăng độ tin cậy của phần mềm.
* Kết quả nhận:
* Danh sách yêu cầu.
* Mô hình phần mềm.
* Phần mềm.
* Kết quả chuyển giao: Phần mềm với độ tin cậy cao (đã tìm và sửa lỗi).

*Stage 6: Deployment in the Market and Maintenance (Triển khai/phát hành trên thị trường và duy trì/bảo trì)*: Công việc của giai đoạn bao gồm việc cài đặt và vận hành phần mềm trong thực tế.

* Mục tiêu: đảm bảo phần mềm vận hành tốt.
* Kết quả nhận: phần mềm đã hoàn thàng.
* Kết quả chuyển giao: các phản ánh của khách hàng trong quá trình sử dụng phần mềm.

### 1.2.2. Mô hình quy trình

Mô hình quy trình- Procedure model: Mô hình quy trình (Procedure model) là một chiến lược phát triển phần mềm, ở đây: bao gồm các cách thức kết hợp, sử dụng tiến trình (process) phần mềm, cách vận dụng các phương pháp và các công cụ trong mỗi giai đoạn phát triển. Mỗi mô hình quy trình tuân theo một loạt các bước duy nhất cho kiểu của nó để đảm bảo sự thành công trong quá trình phát triển phần mềm. Một số mô hình quy trình phổ biến: Waterfall, Iterative, Spiral, Big Bang, V-Model,…

## **Mô hình quy trình thác nước**



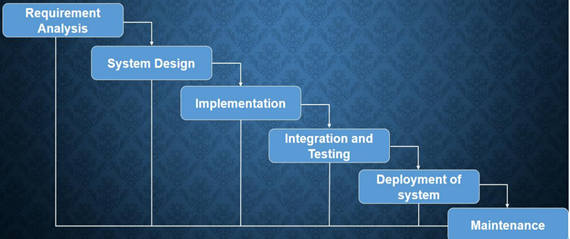
#### Giới thiệu về mô hình

Mô hình thác nước (Waterfall model) là mô hình quy trình đầu tiên được giới thiệu. Nó cũng được gọi là mô hình vòng đời tuần tự tuyến tính. Trong mô hình thác nước, mỗi giai đoạn phải được hoàn thành trước khi giai đoạn tiếp theo có thể bắt đầu và không có sự chồng chéo trong các giai đoạn.

Mô hình thác nước là mô hình đơn giản, dễ áp dụng, quy trình rõ ràng theo từng bước. Dễ quản lý và bảo trì bởi cách tiếp cận tuyến tính và cố định theo từng bước. Các tiêu chí đầu vào và đầu ra được xác định rõ ràng nên dễ dàng trong công tác kiểm tra chất lượng. Hoạt động hiệu quả trong các dự án nhỏ, với các yêu cầu rõ ràng. Có nhiều tài liệu cung cấp cho khách hàng.

#### Các hoạt động của mô hình

Dưới đây minh họa về các hoạt động trong mô hình quy trình thác nước.



Hình .1: Các pha trong mô hình thác nước

Các hoạt động bao gồm:

* ***Thu thập và phân tích yêu cầu (Requirement Analysis):*** Tất cả các yêu cầu có thể có của hệ thống được phát triển đều được ghi lại trong giai đoạn này và được ghi lại trong tài liệu đặc tả yêu cầu để phục vụ cho các giai đoạn sau.
* ***Thiết kế hệ thống (System Design):***Các thông số kỹ thuật yêu cầu từ giai đoạn đầu được nghiên cứu trong giai đoạn này và thiết kế hệ thống được chuẩn bị. Thiết kế hệ thống này giúp xác định các yêu cầu phần cứng và hệ thống cũng như giúp xác định kiến trúc hệ thống tổng thể.
* ***Thực hiện (Implementation):*** Với đầu vào từ thiết kế hệ thống, hệ thống được phát triển đầu tiên trong các chương trình nhỏ gọi là đơn vị, được tích hợp trong giai đoạn tiếp theo. Mỗi đơn vị được phát triển và kiểm tra chức năng của nó (hay còn có thể được gọi là Kiểm thử đơn vị).
* ***Tích hợp và Kiểm thử (Integration and Testing****):* Tất cả các đơn vị được phát triển trong giai đoạn triển khai được tích hợp vào một hệ thống sau khi thử nghiệm của mỗi đơn vị. Sau khi tích hợp, toàn bộ hệ thống được kiểm tra xem có bất kỳ lỗi và hỏng hóc nào không (giai đoạn này còn được gọi là  kiểm thử tích hợp).
* ***Triển khai hệ thống (Deployment of system):*** Sau khi kiểm tra chức năng và phi chức năng được thực hiện; sản phẩm được triển khai trong môi trường  khách hàng hay được tung ra thị trường.
* ***Bảo trì (Maintenance):*** Có một số vấn đề xảy ra trong môi trường khách hàng. Để khắc phục những vấn đề đó, các bản vá lỗi được phát hành. Ngoài  ra để nâng cao sản phẩm một số phiên bản tốt hơn được phát hành. Bảo trì được thực hiện để mang lại những thay đổi này trong môi trường khách hàng.

Trong sơ đồ trên cũng đã thể hiện thứ tự của các giai đoạn trong mô hình này. Ứng dụng thực mô hình thác nước trong một dự án là một quá trình rõ ràng, do áp dụng đặc điểm tuần tự từng bước một của mô hình này để phát triển một ứng dụng hoàn thiện.

#### Đánh giá

Mô hình thác nước minh họa quá trình phát triển phần mềm theo dòng tuần tự tuyến tính. Điều này có nghĩa là bất kỳ giai đoạn nào trong quá trình phát triển chỉ bắt đầu nếu giai đoạn trước đó hoàn thành. Tất cả các giai đoạn này được xếp tầng với nhau, trong đó tiến trình được xem như chảy đều đặn xuống dưới (giống như thác nước) qua các giai đoạn. Giai đoạn tiếp theo chỉ được bắt đầu sau khi đạt được tập hợp mục tiêu đã xác định cho giai đoạn trước và nó được ký kết, vì vậy có tên “Mô hình thác nước”. Trong mô hình này, các giai đoạn không chồng chéo lên nhau.

#### Ưu điểm

Mô hình thác nướccho phép thiết lập một lịch trình với các thời hạn cho từng giai đoạn phát triển và một sản phẩm, và có thể tiến hành từng giai đoạn của mô hình quy trình phát triển. Sự phát triển di chuyển từ ý tưởng, thông qua thiết kế, thực hiện, thử nghiệm, cài đặt, khắc phục sự cố và kết thúc là vận hành và bảo trì. Mỗi giai đoạn phát triển diễn ra theo thứ tự nghiêm ngặt. Một số ưu điểm chính của Mô hình thác nước như sau:

* Đơn giản, dễ hiểu và sử dụng.
* Dễ dàng quản lý do độ cứng của mô hình.
* Mỗi giai đoạn có các phân phối cụ thể và một quy trình xem xét.
* Các giai đoạn được xử lý và hoàn thành cùng một lúc.
* Hoạt động tốt cho các dự án nhỏ hơn, nơi các yêu cầu được hiểu rất rõ.
* Các giai đoạn được xác định rõ ràng.
* Các mốc quan trọng được hiểu rõ.
* Dễ dàng sắp xếp các công việc.
* Quá trình và kết quả được ghi chép đầy đủ

**1.3.3.2. Nhược điểm**

Mô hình thác nướckhông cho phép phản ánh hoặc sửa đổi nhiều. Một khi ứng dụng đang trong giai đoạn thử nghiệm, rất khó để quay lại và thay đổi một cái gì đó không được ghi chép đầy đủ hoặc được nghĩ đến trong giai đoạn khái niệm. Những nhược điểm chính của Mô hình thác nước như sau:

* Không có phần mềm đang hoạt động nào được sản xuất cho đến cuối vòng đời.
* Lượng rủi ro cao và không chắc chắn.
* Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng.
* Mô hình kém cho các dự án dài và đang diễn ra. Không phù hợp với các dự án mà các yêu cầu có nguy cơ thay đổi từ trung bình đến cao.
* Vì vậy, rủi ro và sự không chắc chắn là cao với mô hình quy trình này.
* Rất khó để đo lường sự tiến bộ trong các giai đoạn.
* Không thể đáp ứng các yêu cầu thay đổi.

### Tính ứng dụng của mô hình thác nước

Mỗi phần mềm được phát triển đều khác nhau và đòi hỏi phải tuân theo một cách tiếp cận SDLC phù hợp dựa trên các yếu tố bên trong và bên ngoài.

Một số dự án phần mềm thích hợp với mô hình Waterfall như:

* Các yêu cầu được ghi chép rất đầy đủ, rõ ràng và cố định.
* Mô tả (định nghĩa) về sản phẩm ổn định.
* Công nghệ được hiểu và không mang tính năng động.
* Không có yêu cầu mơ hồ.
* Có sẵn các nguồn lực dồi dào với kiến thức chuyên môn cần thiết để hỗ trợ sản phẩm.
* Dự án ngắn hạn.

Trong mô hình thác nước, điều rất quan trọng là đi theo dấu hiệu của các sản phẩm của từng giai đoạn. Mô hình thác nước vẫn phù hợp cho các dự án nhỏ hơn. Nếu yêu cầu là đơn giản, mô hình thác nước sẽ mang lại kết quả tốt nhất.

### 1.3.5. Kết luận

Tóm lại, ***mô hình thác nước (waterfall)*** là một phương thức quản lý dự án bao gồm các giai đoạn được sắp xếp logic. Khi áp dụng mô hình, tất cả thành viên tham gia dự án sẽ vai trò của mình nằm ở đâu, khi nào mình cần bắt đầu thực hiện vai trò đó.

Tuy nhiên, không phải dự án nào cũng có thể áp dụng waterfall. Mô hình này phù hợp với kiểu dự án có yêu cầu cụ thể, chính xác ngay từ đầu và ít thay đổi. Vì vậy, ngoài kiểu mô hình này, hãy tham khảo thêm những phương pháp quản lý dự án khác để sử dụng linh hoạt với từng khách hàng khác nhau.

* 1. **Cấu trúc tài liệu đặc tả yêu cầu phần mềm**

Mẫu đặc tả yêu cầu phần mềm gồm các phần:

|  |  |
| --- | --- |
| 1. ***Giới thiệu*** | 1.1. Mục đích của tài liệu yêu cầu  1.2. Phạm vi của sản phẩm  1.3. Các định nghĩa, thuật ngữ, từ viết tắt  1.4. Tài liệu tham khảo |
| ***2. Mô tả chung*** | 2.1. Giới thiệu chung về sản phẩm  2.2. Các chức năng của sản phẩm  2.3. Đặc điểm của người sử dùng  2.4. Các điều kiện phụ thuộc |
| ***3. Đặc tả yêu cầu*** | 3.1. Yêu cầu chức năng  3.2. Yêu cầu phi chức năng |

***\* Phần giới thiệu***

Mục đích: Trình bày vai trò, mục đích của tài liệu SRS: Tài liệu mô tả một cách đầy đủ, toàn diện các yêu cầu của phần mềm – đó là các yêu cầu chức năng, phi chức năng, các ràng buộc về mặt thiết kế.

Phạm vi: Mô tả ngắn gọn đặc điểm phần mềm; lĩnh vực ứng dụng phần mềm; phạm vi, đối tượng phục vụ của phần mềm;…

Các định nghĩa, thuật ngữ, từ viết tắt: Giải thích cho các thuật ngữ và từ viết tắt dùng cho tài liệu, các định nghĩa sử dụng trong tài liệu.

Tài liệu tham khảo: Ghi nguồn các tài liệu tham khảo

***\* Phần giới thiệu chung***

Liệt kê và mô tả các tác nhân của hệ thống, liệt kê các use case theo mô hình use case (các use case tương ứng với các chức năng nào).

***\* Phần đặc tả yêu cầu***

Mô tả một các chi tiết từng yêu cầu cụ thể, cho phép các thành viên tham gia dự án căn cứ vào đó để xây dựng phần mềm có chất lượng tốt nhất. Với cách tiếp cận theo mô hình use case, các yêu cầu phần mềm được mô tả theo các use case và trong các đặc tả bổ sung.

***Mô tả use case Ten\_use\_case:***

|  |  |  |  |
| --- | --- | --- | --- |
| Use case: Ten\_use\_case | | | |
| Mục đích | Kết quả cần đạt được của use case | | |
| Mô tả | Mô tả chi tiết use case, vai trò của use case | | |
| Tác nhân | Các tác nhân động đến use case | | |
| Tiền điều kiện | Các điều kiện cần phải thực hiện trước khi thực hiện use case | | |
| Luồng sự kiện chính | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Tên tác nhân / hệ thống | Mô tả hoạt động |
| Luồng sự kiện phụ | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Tên tác nhân /hệ thống | Mô tả hoạt động |
| Hậu điều kiện | Kết quả thu được khi thực hiện đúng và kết thúc use case | | |

* 1. **Công nghệ áp dụng và phương pháp trong phát triển Arabica**
     1. **Giới thiệu công cụ Rational Rose**

Rational Rose là một bộ công cụ được sử dụng cho phát triển các hệ phần mềm hướng đối tượng theo ngôn ngữ mô hình hóa UML. Với chức năng của một bộ công cụ trực quan, Rational Rose cho phép chúng ta tạo, quan sát, sửa đổi, quản lý các biểu đồ. Tập kí hiệu mà Rational Rose cung cấp thống nhất với các kí hiệu trong UML.

Ngoài ra Rational Rose còn cung cấp:

* Use case: được sử dụng để miêu tả những yêu cầu mang tính chức năng của một hệ thống.
* Actor: giữ vai trò người dùng hệ thống, phần cứng hoặc các hệ thống khác bên ngoài.
* Relationship: để kết nối các đối tượng với nhau tạo nên bản vẽ use case. Có các kiểu quan hệ cơ bản: Association, Generalization, Include, Extend.
  + 1. **Giới thiệu công cụ CASE Studio**

Case Studio – Phần mềm vẽ mô hình Entity Relationship Diagrams (ERD) và Data Flow Diagrams (DFD) cũng như tạo script SQL cho nhiều cơ sở dữ liệu một cách tự động. Phần mềm hỗ trợ đầy đủ cho 20 cơ sở dữ liệu như Oracle, DB2, MS SQL, Sybase, MySQL, Firebird, PostgreSQL v.v.

Chức năng chính của CASE Studio 2 : tạo mô hình ERD từ mã script SQL, Đảo mã từ file HTML cụ thể hay file dữ liệu RTF, xuất mô hình Data Flow Diagrams thành file quản lý định dạng XML, Template editor…

* + 1. **Giới thiệu công cụ Balsamiq Mockups**

***Balsamiq Mockups*** là một ứng dụng giúp phác họa ý tưởng thiết kế website thật nhanh chóng với giao diện đơn giản, dễ sử dụng.

Đây là công cụ giúp bạn phác họa các ý tưởng thiết kế website một cách dễ dàng bằng cách tạo ra các Mockup thô đơn giản để mọi người có thể nhanh chóng hình dùng ra ý tưởng thiết kế của bạn bằng những hình ảnh trực quan, sinh động.

Việc sử dụng Balsamiq Mockups với đồ họa đơn giản sẽ giúp cải thiện vấn đề làm việc nhóm. Bằng việc phác thảo nhanh các ý tưởng thông qua các đường vẽ tuy thô nhưng chính điều ấy lại giúp mọi người đoán ra ngay được ý tưởng của bạn mà không phải ngồi mất công suy nghĩ như những bản vẽ phức tạp và nó đặc biệt hữu ích khi sử dụng làm việc với khách hàng của bạn.

Với công cụ này, bạn hoàn toàn có thể xây dựng một website từ đơn giản đến phức tạp, từ trang chung cho đến các trang chi tiết một cách dễ dàng và nhanh chóng.

* + 1. **Phương pháp phân tích và thiết kế phần mềm**

Phương pháp thiết kế hướng đối tượng là sự kết hợp của phương pháp hướng dữ liệu và phương pháp hướng chức năng nó chú trọng đến cả thành phần dữ liệu và chức năng của hệ thống.

Theo phương pháp hướng đối tượng thì một hệ thống phần mềm là một tập hợp các đối tượng có khả năng tương tác với nhau. Các đối tượng chính là các sự vật và hiện tượng vật lý cũng như trừu tượng mà chúng ta có trong thế giới thực.

Mỗi đối tượng có dữ liệu riêng được che dấu với thế giới bên ngoài và các thao tác mà đối tượng có thể thực hiện trên các thành phần dữ liệu của đối tượng. Các đối tượng liên lạc, trao đổi thông tin với nhau bằng cách gửi các thông điệp cho nhau.

## Kết luận chương 1

Chương 1 đã giới thiệu chung về mô hình thác nước và các hoạt động, ưu nhược điểm của mô hình. Nghiên cứu và tìm hiểu cấu trúc tài liệu đặc tả yêu cầu phần mềm theo chuẩn IEEE về khảo sát, phân tích yêu cầu và đặc tả. Giới một số công cụ để sử dụng và phát triển phần mềm. Giới thiệu phương pháp, kỹ thuật áp dụng để phát triển phần mềm.

# **CHƯƠNG 2. PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU**

**PHẦN MỀM**

* 1. **Quy trình kỹ thuật yêu cầu**
     1. **Giới thiệu về quy trình kỹ thuật yêu cầu**

Phổ rộng của các kỹ thuật xác định yêu cầu phần mềm dẫn đến sự hiểu biết tường minh về các yêu cầu được gọi là kỹ thuật yêu cầu hay kỹ thuật yêu cầu phần mềm. Từ quan điểm quy trình phần mềm, kỹ thuật yêu cầu là một hành động kỹ thuật phần mềm chính bắt đầu trong hoạt động giao tiếp và tiếp tục vào hoạt động mô hình hóa. Nó phải được điều chỉnh cho phù hợp với nhu cầu của quá trình, dự án, sản phẩm và những người tham gia dự án.

**Các thành phần cơ bản của kỹ thuật yêu cầu phần mềm**

Kỹ thuật yêu cầu là một cách tiếp cận theo hướng quy trình, theo đúng trình tự các bước được xác định trước để có tài liệu yêu cầu và duy trì các yêu cầu phần mềm trong suốt vòng đời phát triển phần mềm. Kỹ thuật yêu cầu phần mềm được tạo thành từ hai quy trình chính:

* Phát triển yêu cầu
* Quản lý yêu cầu
  + 1. **Các hoạt động trong quy trình kỹ thuật yêu cầu**

Phát triển yêu cầu bao gồm tất cả các hoạt động liên quan đến việc phát hiện, phân tích, xác định và xác thực các yêu cầu:

#### 2.1.2.1 Phát hiện các yêu cầu phần mềm

Phát hiện yêu cầu liên quan đến việc các yêu cầu đến từ đâu và làm thế nào chúng có thể được thu thập. Gợi mở yêu cầu là giai đoạn đầu tiên trong việc xây dựng sự hiểu biết về vấn đề mà phần mềm cần giải quyết.

Đây là một hoạt động của con người và là nơi các bên liên quan được xác định và thiết lập các mối quan hệ giữa nhóm phát triển và khách hàng.

1. **Các nhiệm vụ cần thực hiện:**

* Đánh giá tính khả thi về kỹ thuật và nghiệp vụ của phần mềm định phát triển.
* Tìm kiếm các nhân sự có những hiểu biết sâu sắc nhất, chi tiết nhất về hệ thống giúp chúng ta xác định yêu cầu phần mềm.
* Xác định môi trường kỹ thuật trong đó sẽ triển khai phần mềm.
* Xác định các ràng buộc về lĩnh vực ứng dụng của phần mềm.
* Xác định các phương pháp sử dụng để phát hiện các yêu cầu phần mềm.
* Xác định các yêu cầu còn nhập nhằng để làm mẫu thử
* Thiết kế các kịch bản sử dụng của phần mềm để giúp khách hàng định rõ các yêu cầu chính.

1. **Đầu ra của bước phát hiện yêu cầu phần mềm:**

* Bảng kê (statement) các yêu cầu và chức năng khả thi của phần mềm;
* Bảng kê phạm vi ứng dụng của phần mềm;
* Mô tả môi trường kỹ thuật của phần mềm;
* Bảng kê tập hợp các kịch bản sử dụng của phần mềm;
* Các nguyên mẫu xây dựng, phát triển hay sử dụng trong phần mềm;
* Danh sách nhân sự tham gia vào quá trình phát hiện các yêu cầu phần mềm, kể cả các nhân sự từ phía công ty – khách hàng.

1. **Các phương pháp thực hiện khảo sát phổ biến:**

* Quan sát
* Phỏng vấn trực tiếp
* Thu thập thông tin, tài liệu
* Hỏi ý kiến chuyên gia
* Bảng câu hỏi

1. **Quy trình thực hiện:**

* Tìm hiểu tổng quan về thế giới thực, bao gồm: Quy mô hoạt động; Các hoạt động mà đơn vị có tham gia.
* Tìm hiểu hiện trạng về cơ cấu tổ chức.
* Tìm hiểu hiện trạng nghiệp vụ. Việc tìm hiểu dựa trên các ý sau:
* Thông tin đầu vào
* Quá trình xử lý
* Thông tin kết xuất
* Sau đó tiến hành xếp loại các nghiệp vụ đã thu thập được vào 1 trong 4 loại nghiệp vụ sau:
* Nghiệp vụ lưu trữ
* Nghiệp vụ tra cứu
* Nghiệp vụ tính toán
* Nghiệp vụ tổng hợp, thống kê

**2.1.2.2. Phân tích các yêu cầu phần mềm**

* Phân tích các yêu cầu phần mềm bao gồm các nhiệm vụ:
* Phân loại các yêu cầu phần mềm và sắp xếp chúng theo các nhóm liên quan.
* Khảo sát tỉ mỉ từng yêu cầu phần mềm trong mối quan hệ của nó với các yêu cầu phần mềm khác.
* Kiểm tra từng yêu cầu phần mềm theo các tính chất: Phù hợp, đầy đủ, rõ ràng, không trùng lặp.
* Phân cấp các yêu cầu phần mềm theo dựa trên nhu cầu và đòi hỏi khách hàng/người sử dụng.
* Đánh giá từng yêu cầu phần mềm để xác định chúng có khả năng thực hiện được trong môi trường kỹ thuật hay không, có khả năng kiểm định các yêu cầu phần mềm hay không.
* Đánh giá các rủi ro có thể xảy ra với từng yêu cầu phần mềm.
* Đánh giá tương đối về giá thành và thời gian thực hiện của từng yêu cầu phần mềm trong giá thành sản phẩm phần mềm và thời gian thực hiện phần mềm.
* Giải quyết tất cả các bất đồng về yêu cầu phần mềm với khách hàng / người sử dụng trên cơ sở thảo luận và thương lượng các yêu cầu đề ra.

#### 2.1.2.3. Mô tả các yêu cầu phần mềm

Mô tả các yêu cầu phần mềm hay còn được gọi là đặc tả yêu cầu phần mềm. Khi xây dựng tài liệu mô tả yêu cầu, có thể sử dụng tới các công cụ như: mô hình hóa, mô hình toán học hình thức (a formal mathematical model), tập hợp các kịch bản sử dụng, các nguyên mẫu hoặc bất kỳ một tổ hợp các công cụ nói trên để giúp cho tài liệu đặc tả yêu cầu được tường minh.

Các phương pháp đặc tả:

* Đặc tả phi hình thức: viết bằng ngôn ngữ tự nhiên.
* Đặc tả hình thức: viết bằng tập các ký pháp có các quy định về cú pháp và ngữ nghĩa rất chặt chẽ, thí dụ ký pháp đồ họa dùng các lưu đồ.

Các tiêu chí đánh giá chất lượng của hồ sơ đặc tả:

* Tính rõ ràng, chính xác
* Tính phù hợp
* Tính đầy đủ, hoàn thiện

Các thành phần của hồ sơ đặc tả:

* Đặc tả chức năng mô tả các hoạt động của hệ thống phần mềm sẽ xây dựng:
* Các dịch vụ mà hệ thống phải cung cấp
* Hệ thống sẽ phản ứng với đầu vào cụ thể ra sao
* Hành vi của hệ thống trong các tình huống đặc biệt
* Các công cụ đặc tả tiêu biểu: Biểu đồ Actor – Use case (UML), Biểu đồ luồng dữ liệu (Data Flow Diagrams),…
* Đặc tả phi chức năng và các ràng buộc
* Đặc tả các đặc tính, đặc trưng của phần mềm, các ràng buộc về các dịch vụ hay các chức năng hệ thống cung cấp như thời gian, ràng buộc về các quá trình phát triển, các chuẩn, …
* Các công cụ tiêu biểu: Biểu đồ thực thể liên kết, Đặc tả Logic, Đặc tả đại số.

#### 2.1.2.4. Thẩm định yêu cầu phần mềm

Các nhiệm vụ thực hiện trong quá trình thẩm định yêu cầu phần mềm:

* Rà soát tài liệu yêu cầu để phát hiện các sai sót. Việc ra soát do đồi rà soát thực hiện.
* Tạo bản mẫu: Tạo mẫu thường được sử dụng để xác thực cách giải thích của kỹ sư yêu cầu về các yêu cầu hệ thống, cũng như để gợi ra các yêu cầu mới. Ưu điểm của nguyên mẫu là chúng có thể giúp dễ dàng hơn trong việc giải thích các giả định của kỹ sư yêu cầu và đưa ra phản hồi hữu ích.
* Xác nhận mô hình: Chất lượng của các mô hình được phát triển trong quá trình phân tích cần được xác nhận. Việc xác nhận đảm bảo trong mô hình, các đường truyền thông tin tồn tại giữa các đối tượng có liên quan đến việc trao đổi dữ liệu.
* Thực hiện kiểm thử chấp nhận đối với các yêu cầu. Điều này là có thể gặp khó khăn với các yêu cầu phi chức năng.

## 2.2. Các tác nhân tham gia vào quy trình kỹ thuật yêu cầu

### 2.2.1. Giới thiệu về các tác nhân tham gia quy trình kỹ thuật yêu cầu

Nhà phân tích yêu cầu nghiệp vụ (Business Requirements Analysis) cần các cuộc thảo luận để xác định yêu cầu phần mềm, bao gồm:

1. Nhà tài trợ/chủ đầu tư
2. Người quản lý dự án
3. Người dùng
4. Các bên liên quan khác
5. Người phát triển: yêu cầu chức năng và phi chức năng
6. Người kiểm thử (Tester): yêu cầu chức năng và phi chức năng

#### 2.2.1.1. Nhà tài trợ/Chủ Đầu tư

Là người trực tiếp đầu tư vào dự án và tài trợ các khoản thể quy trình làm phần mềm có thể diễn ra. Nhà tài trợ/chủ đầu tư là tác nhân trực tiếp đưa ra các yêu cầu về nghiệp vụ trong dự án mà quy trình kỹ thuật yêu cầu phải tuân thủ một cách thống nhất.

#### 2.2.1.2. Người quản lý dự án

Là người quản lý về tất cả thông tin của dự án trong đó có cả về mức độ về độ lớn và độ phức tạp của dự án. Là người chịu trách nhiệm chính trong quy trình phát triển phần mềm đối với các vấn đề xảy ra. Nắm rõ được thông tin, bản chất từ đó đưa ra những quyết định những quyết định, hướng đi cho quy trình kỹ thuật.

#### 2.2.1.3. Người dùng

Là người trực tiếp sử dụng phần mềm, từ đó quá trình thực hiện kỹ thuật yêu cầu phải đáp ứng được yêu cầu người dùng từ đó mới có thể phát triển và hoàn thiện hệ thống. Phần mềm làm ra khi đưa đến tay người dùng phải hoàn thiện phục vụ được nhu cầu cầu của người dùng mong muốn.

#### 2.2.1.4. Các bên liên quan khác

Ngoài ra còn có các bên liên quan tới nhiều vấn đề như pháp lý, năm bắt kỳ vọng của khách hàng, chiến lược phát triển, quản lý nhân sự, …

#### 2.2.1.5. Người phát triển

Là người trực tiếp tham gia và thực hiện các quy trình kỹ thuật yêu cầu. Có thể nói đây là tác nhân chính để tạo ra một sản phẩm phần mềm, là bộ phận phát triển những yêu cầu chức năng và phi chức năng của hệ thống sau đó phát triển và hoàn thiện phần mềm.

#### 2.2.1.6. Người kiểm thử (Tester)

Là bộ phận chịu trách nhiệm kiểm thử hệ thống trong đó có các yêu cầu chức năng và phi chức năng của hệ thống xem có gặp các lỗi hay có vấn đề gì trước khi giao nó đến tay khách hàng.

Nhà phân tích yêu cầu nghiệp vụ (Business Requirements Analysis) cần các cuộc thảo luận để xác định yêu cầu phần mềm, bao gồm:

1. Nhà tài trợ/chủ đầu tư
2. Người quản lý dự án
3. Người dùng
4. Các bên liên quan khác
5. Người phát triển: yêu cầu chức năng và phi chức năng
6. Người kiểm thử (Tester): yêu cầu chức năng và phi chức năng

**2.3 Đặc tả yêu cầu phần mềm cho trung tâm tiếng Anh Arabica**

**2.3.1 Giới thiệu**

## 2.3.1.1 Mục đích của tài liệu yêu cầu

Mục đích của tài liệu yêu cầu này là:

* Giúp khách hàng có thể hiểu rõ về các thành phần, chức năng và các yêu cầu phần mềm mà khách hàng đã yêu cầu.
* Là tài liệu giúp cho nhóm người thiết kế phần mềm theo đúng yêu cầu của khách hàng.
* Đây là tài liệu đầu tiên được tạo ra trong tiến trình thiết kế phần mềm.
* Là tài liệu tham khảo cho tất cả mọi người trong nhóm thiết kế phần mềm.

## 2.3.1.2 Phạm vi sản phẩm

Phạm vi của sản phẩm bao gồm:

* Sản phẩm sẽ là phần mềm bán đồ chơi trực tuyến nhằm bán đồ chơi qua mạng bao gồm thanh toán qua mạng.
* Mục tiêu của sản phẩm là đáp ứng yêu cầu của khách hàng đã đặt ra.
* Phần mềm có thể chạy tốt trên nhiều môi trường làm việc khác nhau như các trình duyệt khác nhau như Chrome, Cốc Cốc, FireFox, Brave, Opera, Tor,…
* Phần mềm sử dụng hệ cơ sở dữ liệu MySQL để lưu trữ.Sản phẩm hoàn thành phải có mục liên hệ khách hàng tức liên hệ giữa khách hàng và giáo viên. Hệ thống website cung cấp thông tin giới thiệu về khóa là tên giáo viên, lịch học,... bao gồm 2 module lớn là trang thông tin và trang quản trị website.

**2.3.2 Mô tả chung**

**2.3.2.1 Các tác nhân hệ thống**

* Hệ thống gồm 3 tác nhân chính:
* Học viên
* Giảng viên
* Quản trị viên
* Mô tả tác nhân:
* *Khách hàng*: những khách hàng đã có tài khoản đăng nhập trên trang web. Khách hàng khi mua khóa học trên trang web thì thông tin của khách hàng đã được lưu lại trong cơ sở dữ liệu và khi người dùng muốn mua hàng chỉ cần đăng nhập bằng tài khoản thông tin của khách hàng sẽ hiển thị và khách hàng chỉ cần tiến hành mua.
* *Giảng viên*: là những khách hàng đăng nhập tài khoản trên trang web với vai trò giảng viên. Giảng viên có thể tạo các khóa học, học dung, buổi học,giá,…
* *Quản trị viên*: là người quản lý trang web và được truy cập vào các chức năng quản lý của hệ thống.

**2.3.1.2 Các yêu cầu chức năng**

Bảng 2.1: Mô tả thông tin về tác nhân và các chức năng tương ứng

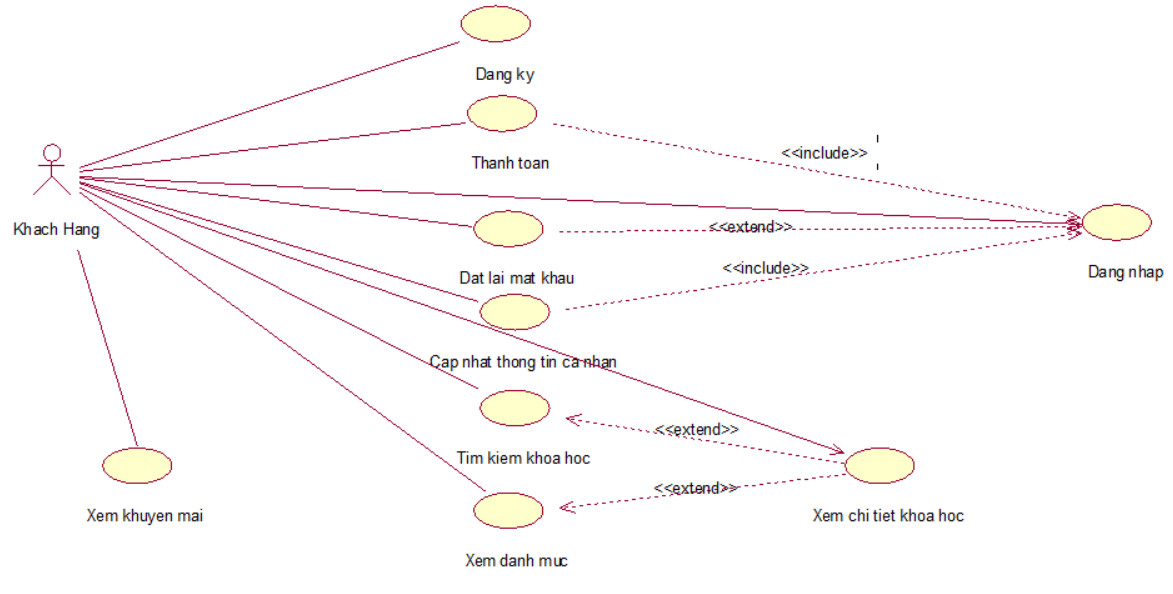
|  |  |  |
| --- | --- | --- |
| ***Tác nhân*** | ***Chức năng*** | ***Mô tả*** |
| Khách hàng | 1. Đăng ký | Người dùng có thể đăng ký tài khoản trên website, các thông tin đăng ký bao gồm: họ tên, ngày sinh, giới tính, địa chỉ, email, số điện thoại, mật khẩu, nhập lại mật khẩu. |
| 2. Đăng nhập | Sau khi đăng ký thành công, khách hàng phải đăng nhập để sử dụng các chức năng nâng cao của website, thông tin đăng nhập bao gồm tên đăng nhập và mật khẩu. |
| 1. Đặt lại mật khẩu | Nếu khách hàng quên mật khẩu, khách hàng có thể yêu cầu đặt lại mật khẩu tài khoản của mình, thông tin yêu cầu bao gồm email hoặc số điện thoại. |
| 4. Cập nhật thông tin cá nhân | Sau khi khách hàng đăng ký tài khoản, khách hàng có thể thay đổi lại các thông tin cá nhân của mình. |
| 5. Tìm kiếm khóa học | Giúp học viên và giảng viên tìm kiếm khóa học theo từ khoá. Người dùng tìm kiếm khóa học, thông tin sẽ hiện ra thông tin khóa học như mã khóa học , tên khóa học,tên giảng viên, lịch học, giá … |
| 6. Xem khóa học trong danh mục | Khách hàng có thể xem các khóa học của cửa hàng theo danh mục được chọn. |
| 7. Xem chi tiết khóa học | Học viên và giảng viên có thể xem thông tin chi tiết của một khóa học bất kỳ trên website, các thông tin bao gồm: mã khóa học , tên khóa học, tên giảng viên, lịch học, giá... |
| 8. Mua khóa học | Học viên có thể bấm vào mua khóa hàng để tiến hành mua khóa học và thanh toán. |
| 9. Thanh toán | Tính tổng số tiền mà học viên cần thanh toàn, nếu có giảm giá thì trừ tiền giảm giá cho học viên sau đó thanh toán thông qua phương thức thanh toán mà học viên lựa chọn. |
| 10. Xem khuyến mãi | Khách hàng có thể xem các khuyến mãi của cửa hàng. |
| Quản trị viên/Giáo viên | 11. Đăng nhập | Quản trị viên có thể đăng nhập vào hệ thống của cửa hàng để có thể sử dụng các chức năng quản trị. |
| 12. Quản lý khóa học | Quản trị viên/Giáo viên có thể quản lý khóa học một cách dễ dàng. Quản trị viên sẽ quản lý số lượng khóa học đi kèm với các thông tin như tên khóa học, mã giá học,tên giảng viên, giá cả,… Giảng viên sẽ quản lý tên khóa học, nội dung khóa học, lịch học, … |
| 13. Quản lý khuyến mãi | Hỗ trợ quản trị viên quản trị viên quản lý khuyến mãi một cách dễ dàng, gồm các hoạt động xem, thêm, sửa, xóa. Quản trị viên sẽ quản lý các thông tin mã khuyến mãi, tên khuyến mãi, ngày khuyến mãi, phần trăm khuyến mãi, mã khóa học, tên khóa học. |
| 14. Quản lý danh mục | Quản lý các danh mục, menu của chương trình. |
| 15. Quản lý tài khoản | Hỗ trợ quản trị viên quản lý các thông tin của tài khoản như xem và xóa các tài khoản. Phần quản lý khách hàng: Khách hàng cần đăng ký tài khoản và đi kèm các thông tin phục vụ cho việc thanh toán (nếu có). |

Dưới đây là sơ đồ use case tổng quan mô tả sự tương tác giữa người dùng mà hệ thống.



Hình .1: Sơ đồ use case tổng quát

Dưới đây là sơ đồ phân rã use case Khách hàng để hình dung rõ ràng các cách mà một khách hàng có thể tương tác với phần mềm.



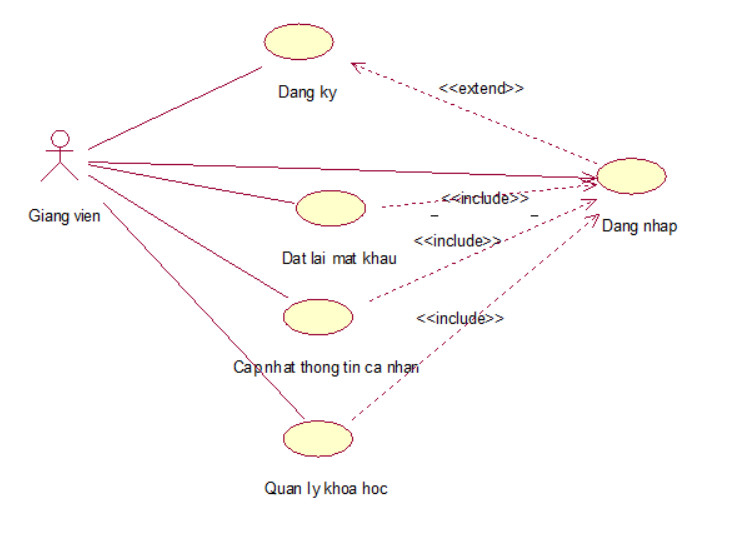
Hình 2.2: Sơ đồ phân rã use case phía front end

Dưới đây là sơ đồ phân rã use case Quản trị hệ thống để hình dung rõ ràng các cách mà một người quản trị có thể tương tác với phần mềm.



Hình 2.3: Sơ đồ phân rã use case “Quản trị hệ thống”

Dưới đây là sơ đồ phân rã use case Giảng viên để hình dung rõ ràng các cách mà một người quản trị có thể tương tác với phần mềm.

**

Hình 2.4: Sơ đồ phân rã use case “Giảng viên”

* + 1. **Quy trình nghiệp vụ**

**2.3.3.1 Quy trình sử dụng phần mềm**

Khách hàng có thể truy cập vào website để xem các khóa học hiện có trên hệ thống. Khách hàng cũng có thể đăng kí để tạo tài khoản cho mình. Sau đó có thể đăng nhập để sử dụng các chức năng khác như tìm kiếm khóa học, mua khóa học,…

Nếu người dùng chưa có tài khoản, người dùng có thể yêu cầu hệ thống cho phép mình đăng ký tài khoản. Lúc này hệ thống sẽ yêu cầu người dùng cung cấpmột số thông tin cần thiết và gửi liên kết qua email mà người dùng vừa cung cấp. Sau đó người dùng có thể vào email để thực hiện xác thực tài khoản.

Nếu người dùng quên mật khẩu, họ có thể yêu cầu hệ thống cho phép mình thiết lập lại mật khẩu. Lúc này hệ thống sẽ gửi liên kết qua email mà người dùng đã đăng ký và họ có thể vào liên kết đó để thực hiện việc thiết lập lại mật khẩu.

Sau khi đăng nhập thành công vào hệ thống, người dùng có thể xem, cập nhật thông tin cá nhân của mình, thay đổi mật khẩu và thực hiện các chức năng khác trong phạm vi mà hệ thống cung cấp.

Quy trình sử dụng phần mềm được minh họa trên Hình 2.5 sau đây:

Diagram

Description automatically generated

Hình 2.5: Biểu đồ hoạt động sử dụng phần mềm

* + 1. **Yêu cầu chức năng**

#### 2.3.4.1 Chức năng đăng nhập

* 1. **Mô tả chức năng**

Bảng .2: Mô tả chức năng đăng nhập

|  |  |  |
| --- | --- | --- |
| Tên Use case | Đăng nhập | |
| Mô tả Use case | Use case này cho phép người dùng, người quản trị đăng nhập để xác nhận quyền truy cập hệ thống. | |
| Luồng sự kiện | Luồng cơ bản | 1. Use case này bắt đầu khi người dùng nhấn vào biểu tượng đăng nhập hệ thống. Hệ thống hiển thị màn hình đăng nhập gồm Email và mật khẩu.  2. Người dùng nhập Email và mật khẩu sau đó kích nút “ Đăng nhập”. Hệ thống kiểm tra Email và mật khẩu đã nhập từ bảng TAIKHOAN và hiển thị thông báo đăng nhập thành công. Use case kết thúc. |
| Luồng rẽ nhánh | - Tại bước 2 trong luồng cơ bản:  + Nếu người dùng nhập sai Email hoặc mật khẩu, hệ thống hiển thị thông báo lỗi. Người dùng có thể chọn quay về luồng cơ bản và nhập lại, hoặc bỏ qua thao tác khi đó use case kết thúc.  + Nếu người dùng nhập thiếu thông tin, hệ thống sẽ thông báo lỗi và use case kết thúc.  - Tại thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| Điều kiện đặc biệt | Nếu nhập sai chỉ được tối đa 5 lần. | |
| Tiền điều kiện | Không | |
| Hậu điều kiện | Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống, nếu không hệ thống không thay đổi. | |
| Các điểm mở rộng | Không | |

1. **Biểu đồ trình tự**

Dưới đây là biểu đồ trình tự mô tả các thao tác của chức năng đăng nhập.

Diagram

Description automatically generated with medium confidence

*Hình 2.6: Biểu đồ trình tự chức năng đăng nhập*

#### 2.3.4.2. Chức năng đăng ký

* + - * 1. **Đặc tả chức năng**

Bảng 2.: Mô tả chức năng đăng ký

|  |  |  |
| --- | --- | --- |
| Tên Use case | Đăng ký | |
| Mô tả Use case | Use case này cho phép người dùng đăng ký tài khoản.ZSS | |
| Luồng sự kiện | Luồng cơ bản | 1. Use case này bắt đầu khi người dùng nhấn vào biểu tượng đăng ký, hệ thống hiển thị màn hình giao diện đăng kí gồm Email, mật khẩu, nhập lại mật khẩu, thông tin cá nhân yêu cầu người dùng nhập.  2. Người dùng nhập Email, mật khẩu, nhập lại mật khẩu, thông tin cá nhân, sau đó kích nút “ Đăng ký”, hệ thống kiểm tra Email và mật khẩu đã nhập và hiển thị thông báo “ Đăng ký thành công”. Use case kết thúc. |
| Luồng rẽ nhánh | - Tại bước 2 trong luồng cơ bản:  + Nếu người dùng nhập thông tin không đúng định dạng, hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại.  Nếu không thể cập nhật thông tin, hệ thống sẽ thông báo lỗi lên màn hình và use case kết thúc.  - Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| Điều kiện đặc biệt | Không | |
| Tiền điều kiện | Không | |
| Hậu điều kiện | Nếu use case thành công, hiển thị menu chính nếu người dùng đăng nhập thành công. | |
| Các điểm mở rộng | Không | |

* 1. **Biểu đồ trình tự**

Dưới đây là biểu đồ trình tự mô tả các thao tác của chức năng đăng ký.

Diagram

Description automatically generated

Graphical user interface, diagram

Description automatically generated

Hình 2.7: Biểu đồ trình tự chức năng đăng ký

#### 2.3.4.3. Chức năng cập nhật thông tin cá nhân

* + - * 1. **Mô tả chức năng**

Bảng 2.: Mô tả chức năng cập nhật thông tin cá nhân

|  |  |  |  |
| --- | --- | --- | --- |
| Tên use case: Cập nhật thông tin cá nhân | | | |
| Mục đích | Use case này cho phép tác nhân cập nhật thông tin cá nhân | | |
| Mô tả | Tác nhân sau khi truy cập vào hệ thống có thể thay đổi một số thông tin cá nhân của mình. | | |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Khách hàng truy cập vào hệ thống | | |
| Luồng sự kiện chính | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Khách hàng | Click vào “Thông tin tài khoản” trên thanh menu |
| 2 | Hệ thống | Lấy thông tin thông tin của khách hàng bao gồm: họ tên, ngày sinh, giới tính, email, số điện thoại, … |
| 3 | Khách hàng | Khách hàng nhập các thông tin cần sửa và kích vào nút “Lưu thông tin” |
| 4 | Hệ thống | Sửa thông tin của khách hàng và hiển thị thông báo cập nhật thành công và thông tin vừa cập nhật ra màn hình |

* + - * 1. **Biểu đồ trình tự**

Dưới đây là biểu đồ trình tự mô tả các thao tác của chức năng cập nhật thông tin cá nhân.

Diagram

Description automatically generated

Graphical user interface, diagram

Description automatically generated

Hình 2.8: Biểu đồ trình tự chức năng cập nhật thông tin cá nhân

#### 2.3.4.4. Chức năng xem thông tin khóa học

**Mô tả chức năng**

Bảng 2.5: Mô tả chức năng xem thông tin khóa học

|  |  |  |
| --- | --- | --- |
| Tên Use case | Xem thông tin khoá học | |
| Mô tả Use case | Use case này cho phép học viên, giáo viên xem thông tin khoá học. | |
| Luồng sự kiện | Luồng cơ bản | 1. Use case nay bắt đầu khi học viên/giảng viên kích vào khoá học hiện trên website. Hệ thống sẽ lấy thông tin về mã khoá học, tên khoá học, Ngày bắt đầu,ngày kết thúc, tên giảng viên, số lượng học viên từ bảng KHOAHOC rồi hiện thị lên màn hình và use case kết thúc |
| Luồng rẽ nhánh | + Trong luồng cơ bản nếu trong bảng KHOA\_HOC chưa có dữ liệu thì hệ thống hiện thị thông báo: “Chưa có khoá học nào được đăng ký” và use case kết thúc.  + Trong luồng cơ bản nếu học viên kích vào xem khoá học mà khoá học đó đã đủ học viên thì sẽ hiển thị 1 thông báo: “Khoá học đã hết chỗ” và use case kết thúc.  - Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| Điều kiện đặc biệt | Không | |
| Tiền điều kiện | Phải Đăng Nhập vào trước | |
| Hậu điều kiện | Không | |
| Các điểm mở rộng | Không | |

**Biểu đồ trình tự**

Dưới đây là biểu đồ trình tự mô tả các thao tác của chức năng xem thông tin khóa học



Hình 2.9: Biểu đồ trình tự chức năng xem chi tiết sản phẩm

#### 2.3.4.5. Chức năng đăng ký khóa học

* 1. **Mô tả chức năng**

Bảng 2.: Mô tả chức năng đăng ký khóa học

|  |  |  |  |
| --- | --- | --- | --- |
| Tên use case: Đăng ký khóa học | | | |
| Mục đích | Use case này cho phép tác nhân đăng ký khóa học | | |
| Mô tả | Tác nhân có thể đăng ký khóa học trên website | | |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Không có | | |
| Luồng sự kiện chính | STT | **Thực hiện bởi** | **Hành động** |
| 1.1 | Khách hàng | Khách hàng kích vào “Đăng ký khóa học”. |
| 1.2 | Hệ thống | Hiển thị ra danh sách các khóa học bao gồm tên khóa học lấy từ bảng **KHOAHOC** |
| 2.1 | Khách hàng | Kích vào tên khóa học |
| 2.2 | Hệ thống | Hiển thị các thông tin của khóa học tương ứng trong bảng **KHOAHOC** ra màn hình gồm tên khóa học, nội dung, ngày mở lớp và các ảnh liên quan đến khóa học. |
| Luồng sự kiện phụ | STT | **Thực hiện bởi** | **Hành động** |
| 2.2a | Hệ thống | Nếu hệ thống không kết nối được với cơ sở dữ liệu. Hệ thống hiển thị thông báo lỗi. |
| Hậu điều kiện | Không có | | |

* 1. **Biểu đồ trình tự**

Dưới đây là biểu đồ trình tự mô tả các thao tác của chức năng đăng kí khóa học.



Hình 2.10: Biểu đồ trình tự chức năng đăng kí khóa học

#### 2.3.4.6. Chức năng quản lý bài giảng

* + - * 1. **Mô tả chức năng**

Bảng 2.: Mô tả chức năng quản lý bài giảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên use case: Quản lý bài giảng | | | |
| Mục đích | Use case này cho phép người quản trị thêm ,sửa, xóa các bài giảng | | |
| Mô tả | Use case này cho phép người quản trị thêm ,sửa, xóa các sản phẩm | | |
| Tác nhân | Người quản trị | | |
| Tiền điều kiện | Người quản trị cần đăng nhập với vai trò người quản trị hệ thống trước khi có thể thực hiện use case. | | |
| Luồng sự kiện chính | STT | **Thực hiện bởi** | **Hành động** |
| 1.1 | Người quản trị | Use case này bắt đầu khi người quản trị kích vào nút “Quản lý bài giảng” trên thanh menu quản trị |
| 1.2 | Hệ thống | Hệ thống lấy thông tin các sản phẩm gồm Mã bài giảng, Tên bài giảng, Tên giảng viên, video bài giảng từ bảng **BAIGIANG** trong CSDL và hiển thị ra màn hình. |
| 2.1 | Người quản trị | Use case cho phép người quản trị Thêm bài giảng. Người quản trị kích vào nút “Thêm bài giảng” trên cửa sổ danh sách bài giảng |
| 2.2 | Hệ thống | Hệ thống hiển thị lên màn hình yêu cầu nhập thông tin chi tiết bài giảng gồm Mã bài giảng, Tên bài giảng, Tên giảng viên, video bài giảng. |
| 2.3 | Người quản trị | Người quản trị nhập thông tin và kích vào nút “Thêm bài giảng” |
| 2.4 | Hệ thống | Hệ thống sẽ sinh một mã bài giảng mới, tạo một bài giảng trong bảng **BAIGIANG** và và hiển thị danh sách các bài giảng đã được cập nhật. |
| STT | **Thực hiện bởi** | **Hành động** |
| 3.1 | Người quản trị | Use case cho phép người quản trị Sửa bài giảng. Người quản trị kích vào nút “Sửa bài giảng” trên một dòng sản phẩm |
| 3.2 | Hệ thống | Hệ thống sẽ lấy thông tin cũ của bài giảng được chọn gồm: Mã bài giảng, Tên bài giảng, Tên giảng viên, video bài giảng từ bảng **BAIGIANG** và hiển thị lên màn hình. |
| 3.3 | Người quản trị | Người quản trị nhập thông tin mới Mã bài giảng, Tên bài giảng, Tên giảng viên, video bài giảng và kích vào nút “Cập nhật”. |
| 3.4 | Hệ thống | Hệ thống sẽ sửa thông tin của bài giảng trong bảng **BAIGIANG** và hiển thị danh sách sản phẩm đã cập nhật. |
| 4.1 | Người quản trị | Use case cho phép người quản trị Xóa bài giảng. Người quản trị kích vào nút “Xóa bài giảng” trên một dòng sản phẩm |
| 4.2 | Hệ thống | Hệ thống sẽ hiển thị một màn hình yêu cầu xác nhận xóa. |
| 4.3 | Người quản trị | Người quản trị kích vào nút “Đồng ý” |
| STT | **Thực hiện bởi** | **Hành động** |
| 4.4 | Hệ thống | Hệ thống sẽ xóa  sản phẩm được chọn khỏi bảng  **BAIGIANG** và hiển thị danh sách các bài giảng đã cập nhật |
| Luồng sự kiện phụ | **STT** | **Thực hiện bởi** | **Hành động** |
| 2.3a | Người quản trị | Nếu người quản trị nhập thông tin bài giảng bị lỗi |
| 2.3b | Hệ thống | Yêu cầu nhập lại |
| 2.3c | Người quản trị | Kích nút “Hủy bỏ”, các thao tác tạo bài giảng sẽ được hủy bỏ |
| 3.3a | Người quản trị | Nếu người quản trị nhập thông tin bài giảng bị lỗi |
| 3.3b | Hệ thống | Yêu cầu nhập lại |
| 3.3c | Người quản trị | Kích nút “Hủy bỏ”, các thao tác cập nhật bài giảng sẽ được hủy bỏ |
| 4.3a | Khách hàng | Click nút “Không đồng ý” |
| 4.3a | Hệ thống | Không thao tác xóa bài giảng đã chọn, hiển thị thông tin các bài giảng |
| Hậu điều kiện | Nếu use case kết thúc thành công thì thông tin về bài giảng sẽ được cập nhật trong CSDL. | | |

* + - * 1. **Biểu đồ trình tự**



Hình 2.11: Biểu đồ trình tự thêm, sửa, xóa các bài giảng

#### 2.3.4.7. Chức năng quản lý giảng viên

**Mô tả chức năng**

Bảng 2.8: Mô tả chức năng quản lý giảng viên

|  |  |  |
| --- | --- | --- |
| Tên Use case | Quản lý giảng viên | |
| Mô tả Use case | Use case này cho phép quản trị viên thực hiện các tác vụ như thêm, sửa, xóa tài khoản giảng viên. | |
| Luồng sự kiện | Luồng cơ bản | 1. Sửa thông tin giảng viên   Quản trị viên chọn, chỉnh sửa thông tin giảng viên, hệ thống lấy, kiểm tra, cập nhật thông tin giảng viên.   1. Xóa thông tin giảng viên   Quản trị viên chọn giảng viên cần xóa và nhấn yêu cầu xóa, hệ thống hiển thị, xóa và thông báo lên màn hình. Sau đó, người dùng xác nhận yêu cầu xóa giảng viên, hệ thống xóa giảng viên và hiển thị lại trang.   1. Thêm giảng viên   Quản trị viên yêu cầu và nhập các thông tin thêm giảng viên, hệ thống hiển thị, kiểm tra dữ liệu vào và thê mới thông tin giảng viên vào CSDL. |
| Luồng rẽ nhánh | Tại thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| Điều kiện đặc biệt | Không | |
| Tiền điều kiện | Quản trị viên cần đăng nhập để thực hiện use case này. | |
| Hậu điều kiện | Hiển thị danh sách tương ứng với thông tin cần tìm kiếm; Cập nhật thành công, thông tin mới sẽ được lưu trữ vào hệ thống;  Xóa thành công giảng viên; Giảng viên khi tạo mới được lưu trữ trong cơ sở dữ liệu của hệ thống kèm theo trạng thái của giảng viên (Locked hay Unlocked) tương ứng với chức năng Mở khóa hay Khóa giảng viên. | |
| Các điểm mở rộng | Không | |

**Biểu đồ trình tự**



Hình 2.12: Biểu đồ trình tự quản lý giảng viên

#### 2.3.4.8. Chức năng quản lý học viên

1. **Mô tả chức năng**

Bảng 2.9: Mô tả chức năng quản lý học viên

|  |  |  |
| --- | --- | --- |
| Tên Use case | Quản lý Học viên | |
| Mô tả Use case | Use case này cho phép người quản trị thêm, xóa thông tin học viên. | |
| Luồng sự kiện | Luồng cơ bản | Use case bắt đầu khi người quản trị nhấn chọn “quản lý học viên” trên menu. Hệ thống lấy thông tin gồm Họ tên, email, số điện thoại, ngày sinh, giới tính, địa chỉ từ bảng HOCVIEN và hiển thị lên màn hình.   1. Xóa học viên khỏi lớp   Quản quản trị kick vào nút “Xóa” để xóa học viên và bấm xác nhận để xóa học viên khỏi lớp, hệ thống xóa học viên khỏi CSDL.   1. Thêm 1 học viên vào lớp   Người quản trị bấm” Thêm học viên”, sau đó nhập tên học viên trên mục tìm kiếm, hệ thống hiển thị danh sách của học viên, người quản trị chọn 1 học viên và bấm “Thêm”. Use case kết thúc. |
| Luồng rẽ nhánh | Nếu người dùng bấm “Hủy”, hệ thống sẽ đóng yêu cầu và hiển thị danh sách học viên, use case kết thúc.  Nếu học viên đã có trong lớp thì hiển thị thông báo học viên đã có trong lớp, use case kết thúc  Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc. |
| Điều kiện đặc biệt | Không | |
| Tiền điều kiện | Người quản trị cần đăng nhập để thực hiện use case này. | |
| Hậu điều kiện | Nếu use case thành công, CSDL sẽ thêm hoặc bớt học viên. | |
| Các điểm mở rộng | Không | |

1. **Biểu đồ trình tự**



Hình 2.13: Biểu đồ trình tự thêm, xóa thông tin học viên

**2.3.5 Các yêu cầu phi chức năng**

**2.3.5.1 Yêu cầu thực thi**

Phần mềm có các yêu cầu thực thi sau:

* Khi đăng ký khách hàng phải điền đầy đủ thông tin đã đưa ra mới cho đăng kí.
* Khi tìm kiếm một sản phẩm nào đó thì thời gian thực hiện khoảng 3 giây.
* Khi khách hàng muốn liên hệ với người phụ trách website thì khách hàng phải nhập đầy đủ thông tin về họ tên và nội dung trong phần liên hệ thì hệ thống mới thực hiện gửi yêu cầu.
* Khách hàng khi muốn mua hàng thì bắt buộc phải đăng nhập.
* Không có yêu cầu theo thời gian thực

#### 2.3.5.2 Yêu cầu an toàn

Phần mềm phải được thiết kế an toàn có thể chống lại sự lây nhiễm virus, thiết kế cẩn thận tránh các lỗi bảo mật đề phòng trang web bị hack làm cho thông tin khách hàng bị mất hay bị thay đổi.

Chú ý trong việc phân quyền cho các thành viên, không được cấp quyền cho thành viên có quyền thay đổi thuộc tính, chức năng của trang web.

#### 2.3.5.3 Yêu cầu bảo mật

Khi hệ thống lưu trữ, truy xuất dữ liệu cá nhân như thông tin khách hàng, thẻ tín dụng, doanh số bán hay thông tin riêng tư, chúng ta cần phải có biện pháp đảm bảo an toàn những dữ liệu này.

1. **Xác định vai trò**

* Toàn bộ ứng dụng không chỉ có một mức độ bảo mật.
* Người dùng cuối chỉ cần quyền truy xuất giới hạn vào hệ thống, người thao tác viên cập nhật và người dùng có quyền truy cập cao hơn ở mọi cấp độ.
* Bảo mật dựa trên vai trò là kĩ thuật dùng để cấp quyền mức độ bảo mật khác nhau tương ứng quyền hạn và độ chuyên nghiệp của mỗi người dùng trong hệ thống.

**Lưu ý:** Nhận biết những lớp chính của những người dùng cần truy cập đến ứng dụng của chúng ta. Gán tên vai trò cho mỗi lớp người dùng. Cuối cùng, gán mức độ tối thiểu có thể truy xuất đến mỗi vai trò. Mỗi lớp người dùng nên có đủ quyền truy xuất đến công việc của họ và không nhiều hơn.

1. **Xác định môi trường bảo mật ứng dụng**

* Độ bảo mật không bị giới hạn người dùng hệ thống.
* Chỉ người dùng đăng nhập vào ứng dụng, ứng dụng phải “login” để kiểm soát tài nguyên chia sẻ như tập tin, dịch vụ hệ thống, cơ sở dữ liệu.
* Mức độ kiểm soát của ứng dụng được gọi là ngữ cảnh bảo mật.
* Chúng ta cần phải làm việc với nhiều người dùng khác như quản trị mạng, cấp quyền truy xuất phù hợp ứng dụng để chia sẻ tài nguyên.

1. **Xác định ảnh hưởng bảo mật**

Nếu công ty có sẵn cơ chế bảo mật thay vào đó hệ thống của chúng ta nên điều chỉnh cho phù hợp với cơ chế đã có. Nếu chúng ta đang thực thi hệ thống bảo mật mới hay một hệ thống khác, cần phải phân tích tác động của hệ thống trên hệ thống hiện tại:

* Hệ thống mới có làm hỏng chức năng của phần mềm hiện tại?
* Hệ thống đòi hỏi phải hỗ trợ thêm một phần người dùng - đăng nhập mở rộng?
* Hệ thống sẽ khóa một vài người dùng trên những tập tin hay những tài nguyên mà họ được truy cập trước đây.

1. **Kế hoạch vận hành**

* Khi tổ chức phát triển và thay đổi, người dùng mới được thêm vào, người củ được cập nhật và bỏ đi.
* Những thao tác này đòi hỏi thay đổi CSDL bảo mật, đó là nơi thông tin người dùng và quyền hạn truy cập của họ được lưu.
* Những thông tin này được lưu trữ hiện thời.

1. **Kế hoạch kiểm soát và ứng dụng**

* Một hệ thống bảo mật tốt không là cơ chế thụ động. Thay vào đó, chứa chức năng trợ giúp kiểm soát hoạt động của hệ thống cho vấn đề bảo mật.
* Vấn đề chung của chức năng này là nhật kí.
* Toàn bộ thao tác của hệ thống có thể được ghi nhận hầu như toàn bộ sự kiện liên quan đến bảo mật hệ thống.
* Có thể ghi nhận mỗi khi đăng nhập, truy xuất đến mọi tài nguyên nhưng điều này hiếm khi hiệu quả. Thường chúng ta sẽ ghi nhận một số tập thông tin này như việc cố gắng đăng nhập lỗi.

1. **Xác định mức độ yêu cầu bảo mật**

* Bảo mật cũng giống như những phần khác trong thiết kế ứng dụng, là sự cân nhắc giữa hiệu quả và chi phí.
* Nếu hệ thống không lưu những dữ liệu có tính nhạy cảm cao.
* Cách tốt nhất để triển khai hệ thống đó là “giữ sự xác thực của người dùng” đòi hỏi lưu trữ.
* Nếu chúng ta lưu trữ thông tin cần cho bảo mật, chi phí cho bảo mật thông tin đặc biệt phải được kiểm chứng.
* Không có hệ thống nào bảo mật 100%. Chúng ta phải xác định mức độ rủi ro bảo mật có thể chấp nhận được.
* Độ rủi ro bảo mật diễn tả tỉ lệ phần trăm tương xứng khả năng mà bảo mật hệ thống không bao giờ đạt đến.
* Điều đó có thể nhưng chi phí phí tốn để xây dựng hệ thống bảo mật 99%.
* Chúng ta hay khách hàng phải xác định mức độ rủi ro có thể chấp nhận được dựa trên dữ liệu nhạy cảm của hệ thống.

#### 2.3.5.4 Các luật vận hành

* Chúng ta có thể giảm bớt chi phí vận hành theo nhiều cách.
* Cách tốt nhất để giảm chi phí vận hành là đảm bảo chương trình được kiểm thử và chạy thử trước khi đưa vào triển khai.
* Chi phí triển khai có thể được giảm bớt bởi phân phối trực tuyến hay những thủ tục tự động cài đặt và quy trình vận hành có thể tự động bằng các quy trình tin học.
* Trong trường hợp phần cứng, phần mềm là thành phần được mua chứ không được phát triển, chúng ta có thể nhận sự chấp thuận vận hành từ nhà xưởng hay người ủy thác của sản phẩm.
* Vận hành sản phẩm trung gian tiết kiệm cho chúng ta chi phí thuê mướn nhân viên.

#### 2.3.5.5 Các đặc điểm chất lượng phần mềm

Phần mềm được làm ra phải đảm bảo các yêu cầu về chất lượng như:

* ***Tính thích ứng*** có thể chạy trên nhiều môi trường làm việc khác nhau, trên nhiều trình duyệt khác nhau.
* ***Tính sẵn có, tính chính xác*** là phần mềm phải hoạt động chính xác không tính toán số liệu sai dẫn đến thanh toán tiền cho khách hàng sai.
* ***Tính linh hoạt; tính thao tác giữa các phần; tính có thể bảo trì*** là trong quá trình hoạt động của phần nếu có xảy ra lỗi thì phải có thể sửa chữa được hoặc nâng cấp lên phiên bản mới.
* ***Tính khả chuyên; tính tin cậy*** là phần mềm phải hoạt động ổn định, chạy đúng yêu cầu không chạy sai, đáp ứng được yêu cầu của người dùng.
* ***Tính có thể tái sử dụng; tính có thể kiểm thử*** là phần mềm hoàn thành có thể được kiểm tra thông qua cài đặt sử dụng thử để kiểm tra tính đúng đắn, tính dễ sử dụng, kiểm tra lỗi trước khi phát hành sản phẩm.
* ***Tính dễ sử dụng*** là phần mềm khi hoàn thành phải dễ sử dụng, giao diện phải thân thiện với người sử dụng, nội dung rõ ràng, dễ hiểu, chức năng đơn giản.

**2.4 Biểu đồ lớp thực thể**

Dưới đây là biểu đồ lớp thực mô tả các lớp của hệ thống.

Diagram

Description automatically generated

Hình 2.14: Biểu đồ lớp thực thể

## 2.5 Kết luận chương 2

Chương 2 đã giới thiệu về quy trình kỹ thuật yêu cầu và các hoạt động, tác nhân trong quy trình kỹ thuật yêu cầu, phân tích và đặc tả yêu cầu phần mềm quản lý giảng dạy của trung tâm Tiếng anh Arabica dựa trên cấu trúc tài liệu đặc tả yêu cầu phần mềm đã chuẩn bị ở chương 1. Phân tích các yêu cầu chức năng và phi chức năng của phần mềm.

**CHƯƠNG 3. THIẾT KẾ PHẦN MỀM**

**3.1. Tổng quan về thiết kế phần mềm**

**3.1.1. Vai trò và các nhiệm vụ trong hoạt động thiết kế phần phần**

* 1. **Khái niệm của thiết kế phần mềm là**
* Là quá trình chuyển hóa các đặc tả yêu cầu phần mềm thành một biểu diễn thiết kế của hệ thống phần mềm cần xây dựng, sao cho người lập trình có thể ánh xạ nó thành một chương trình.
  1. **Một số hoạt động chính:**
* Nghiên cứu để hiểu vấn đề
* Chọn một số giải pháp thiết kế và xác định các đặc điểm thô của nó
* Mô tả trừu tượng cho mỗi giải pháp thiết kế, các sai sót cần phát hiện và chỉnh sửa trước khi lập tài liệu thiết kế chính thức
  1. **Vai trò:**
* Là cách duy nhất để chuyển hóa một cách chính xác các yêu cầu của khách hàng thành mô hình thiết kế hệ thống phần mềm cuối cùng làm cơ sở cho việc triển khai chương trình phần mềm.
* Là công cụ giao tiếp giữa các nhóm cùng tham gia phát triển sản phẩm, quản lý rủi ro, đạt được phần mềm hiệu quả.
* Là tài liệu cung cấp đầy đủ các thông tin cần thiết cho để bảo trì hệ thống.
* Nếu không có thiết kế thì hệ thống không tin cậy dẫn tới nguy cơ thất bại.
* Thiết kế tốt là chìa khóa làm cho phần mềm trở nên hữu hiệu.
  1. **Các khái niệm trong thiết kế:**
* Trừu tượng (abstraction): chia ra mức cao nhất, mức vừa, mức thấp, có các dạng trừu tượng như trừu tượng thủ tục, trừu tượng dữ liệu, trừu tượng điều khiển.
* Phân rã (Decomposition): Chia nhỏ đối tượng
* Làm mịn (refinement): Chiến lược thiết kế từ trên xuống
* Modul: Chia thành các phần riêng có tên và địa chỉ
* Thủ tục phần mềm (software procedure)
* Che dấu thông tin (information hidding)
  1. **Nhiệm vụ:**
* Nhiệm vụ của thiết kế là chuyển đổi những yêu cầu của hệ thống (kết quả của quá trình phân tích) sang dạng biểu diễn của hệ thống phần mềm. Nghĩa là xây dựng các mô tả văn bản (thiết kế chi tiết) nêu rõ mối quan hệ giữa tiền điều kiện và hậu điều kiện cho tất cả các chức năng (quá trình) của hệ thống.

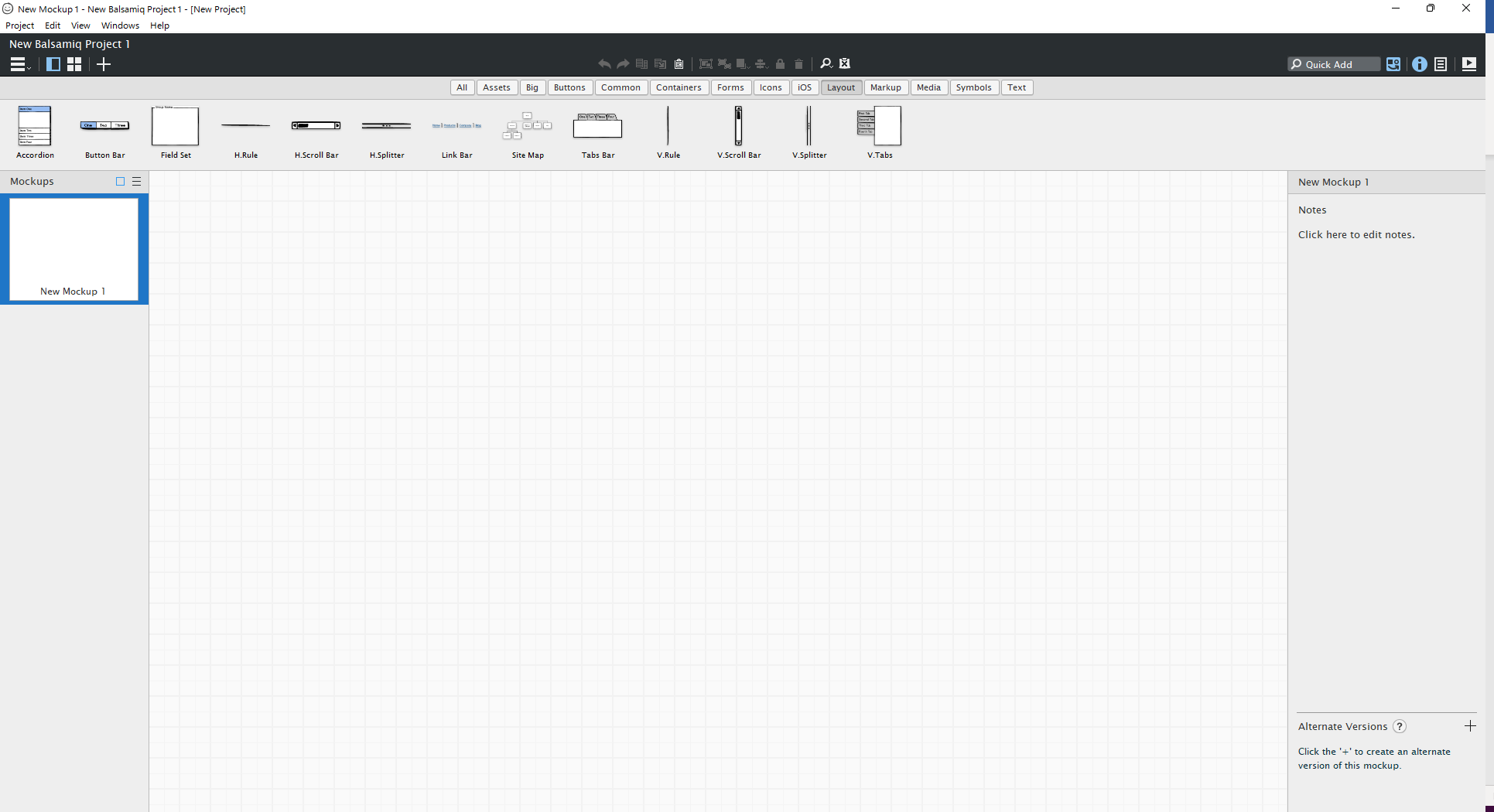
3. 1. 1. **Chiến lược thiết kế phần mềm**

***\* Thiết kế hướng đối tượng (Object Oriented Design - OOD):***

* Hệ thống được nhìn nhận như một bộ các đối tượng (chứ không phải là bộ các chức năng). Hệ thống được phân tán, mỗi đối tượng có những thông tin trạng thái riêng của nó. Đối tượng là một bộ các thuộc tính xác định trạng thái của đối tượng đó và các phép toán của nó. Nó được thừa kế từ một vài lớp đối tượng lớp cao hơn, sao cho dễ định nghĩa nó chỉ cần nêu đủ các khác nhau giữa nó và các lớp cao hơn nó.
* Che dấu thông tin là chiến lược thiết kế dấu càng nhiều thông tin trong các thành phần càng hay. Cái đó ngầm hiểu rằng việc kết hợp điều khiển logic và cấu trúc dữ liệu được thực hiện trong thiết kế càng chậm càng tốt. Liên lạc thông qua các thông tin trạng thái dùng chung (các biến tổng thể) là ít nhất, nhờ vậy khả năng hiểu là được tăng lên. Thiết kế là tương đối dễ thay đổi vì sự thay đổi một thành phần không thể không dự kiến các hiệu ứng phụ trên các thành phần khác.
* Thiết kế hướng đối tượng là dựa trên việc che dấu thông tin, nhìn hệ phần mềm như là một bộ các đối tượng tương tác với nhau chứ không phải là một bộ các chức năng như cách tiếp cận chức năng. Các đối tượng này có một trạng thái được che dấu và các phép toán trên các trạng thái đó. Thiết kế biểu thị các dịch vụ được yêu cầu và được cung cấp bởi các đối tượng có tương tác với nó.
* Thiết kế hướng đối tượng có ba đặc trưng:
* Vùng dữ liệu dùng chung là bị loại bỏ. Các đối tượng liên lạc với nhau bằng cách trao đổi thông báo chứ không phải bằng các biến dùng chung.
* Các đối tượng là các thực thể độc lập mà chúng sẵn sàng được thay đổi vì rằng tất cả các trạng thái và các thông tin biểu diễn là chỉ ảnh hưởng trong phạm vi chính đối tượng đó thôi. Các thay đổi về biểu diễn thông tin có thể được thực hiện không cần sự tham khảo tới các đối tượng hệ thống khác.
* Các đối tượng có thể phân tán và có thể hành động tuần tự hoặc song song. Không cần có quyết định về tính song song ngay từ một giai đoạn sớm của quá trình thiết kế.
* Các ưu điểm:
* Dễ bảo trì vì các đối tượng là độc lập. Các đối tượng có thể hiểu và cải biên như là một thực thể độc lập. Thay đổi trong thực hiện một đối tượng hoặc thêm các dịch vụ sẽ không làm ảnh hưởng tới các đối tượng hệ thống khác.
* Các đối tượng là các thành phần dùng lại được thích hợp (do tính độc lập của chúng). Một thiết kế có thể dùng lại được các đối tượng đã được thiết kế trong các bản thiết kế trước đó.
* Đối với một vài lớp hệ thống, có một phản ánh rõ ràng giữa các thực thể có thực (chẳng hạn như các thành phần phần cứng) với các đối tượng điều khiển nó trong hệ thống. Điều này cải thiện được tính dễ hiểu của thiết kế.
* Nhược điểm:
* Sự nhận minh các đối tượng hệ thống thích hợp là khó khăn. Cách nhìn tự nhiên nhiều hệ thống là cách nhìn chức năng và việc thích nghi với cách nhìn hướng đối tượng đôi khi là khó khăn.
* Phương pháp thiết kế hướng đối tượng vẫn còn là tương đối chưa chín muồi và đang thay đổi mau chóng.

**3.1.3*.* Công cụ hỗ trợ thiết kế phần mềm**

**Balsamiq Mockups** là một ứng dụng giúp phác họa ý tưởng thiết kế website thật nhanh chóng với giao diện đơn giản, dễ sử dụng.



Hình 3.1: Giao diện làm việc của Balasmiq Mockups

***Mockup Balsamiq***hỗ trợ để phát thảo ý tưởng về giao diện khi thiết kế web và file xuất ra dưới dạng định dạng PNG, Bmml,... nên chất lượng hình ảnh cực tốt. Các Mockup sẽ giống như bạn đang vẽ bằng tay nên người khác có thể hiểu ngay đó là khung bố cục chứ không phải bản thiết kế cuối.

**3.2. Tài liệu thiết kế phần mềm**

### 3.2.1. Giới thiệu

#### 3.2.1.1. Mục đích tài liệu

Tài liệu này mô tả chi tiết về thiết kế phần mềm quản lý hoạt động giảng dạy phục vụ hoạt động giảng dạy tiếng anh của trung tâm tiếng Anh Arabica .

#### 3.2.1.2. Phạm vi tài liệu

Tài liệu này là cơ sở giao tiếp của các thành viên trong đội phát triển dự án phần mềm quản lý hoạt động giảng dạy và tài liệu này cũng là căn cứ trong hoạt động kiểm thử, vận hành và bảo trì phần mềm.

#### 3.2.1.3. Tài liệu tham khảo

* Giáo trình Nhập môn Công nghệ phần mềm
* Đề cương bài giảng môn Công nghệ phần mềm của Đại học Công nghiệp Hà Nội
* Tài liệu Nhập môn Công nghệ phần mềm của Đại học Bách khoa Hà Nội

#### 3.2.1.4. Bố cục tài liệu thiết kế phần mềm

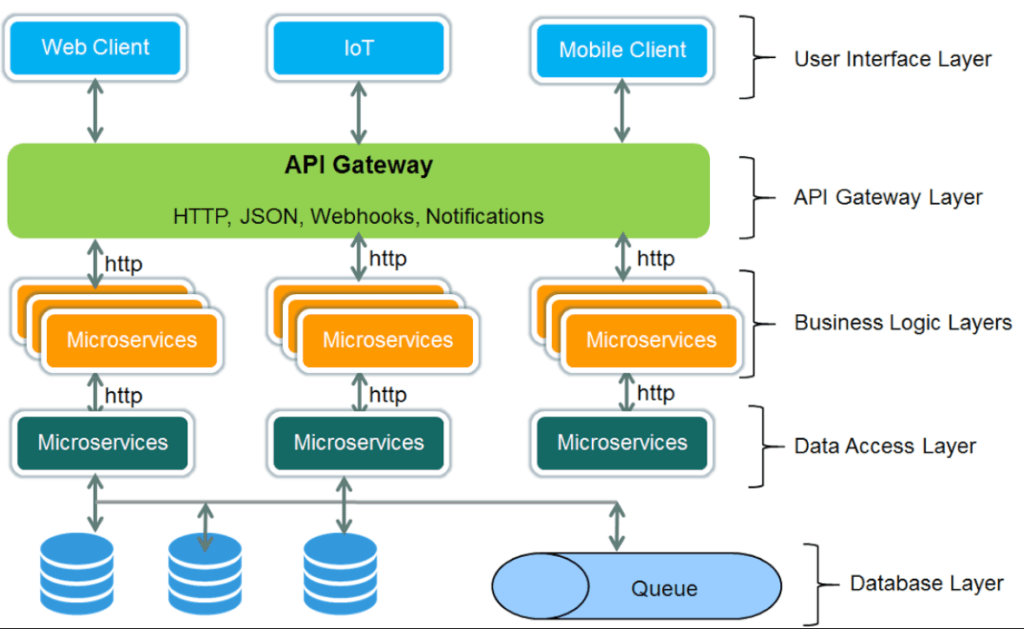
Ngoài phần giới thiệu, cấu trúc Tài liệu mô tả thiết kế phần mềm quản lý hoạt động giảng dạy còn gồm các nội dung chính sau:

* Tổng quan về phần mềm;
* Thiết kế kiến trúc phần mềm;
* Thiết kế dữ liệu;
* Thiết kế các thành phần phần mềm;
* Thiết kế giao diện;

### 3.2.2. Thiết kế kiến trúc phần mềm

#### 3.2.2.1. Mô hình kiến trúc

* Kiến trúc phần mềm là một hệ thống cấu trúc các thành phần cần thiết, đặc điểm, tính chất cũng như mối quan hệ giữa chúng. Thiết kế kiến trúc phần mềm là quá trình mô tả và phát triển hệ thống ấy.
* Hiện nay, các mẫu mô hình kiến trúc phần mềm phổ biến là:
* Kiểu kiến trúc hướng đối tượng
* Kiểu kiến trúc phân tầng
* Thiết kế kiến trúc kho chứa
* Mẫu kiến trúc khách – chủ
* Mẫu kiến trúc ống – lọc
* Mô hình kiến trúc tương tác
* Vai trò của kiến trúc phần mềm:



Hình 3.2: Minh họa vai trò của kiến trúc phần mềm

* Cung cấp thông tin cần thiết, định hướng cho các giai đoạn khác trong quá trình phát triển phần mềm: Dựa trên thiết kế đề ra, các bộ phận có thể tiến hành lập trình, xây dựng phần mềm theo đúng yêu cầu và hạn chế thấp nhất sai sót không mong muốn.
* Là cơ sở để kiểm tra, phân tích tính chính xác, mức độ phù hợp, khả năng đáp ứng yêu cầu được đặt ra của một sản phẩm sau khi hoàn thành: Mỗi phần mềm ra đời đều có những sai số nhất định so với dự tính ban đầu. Lúc này, việc đối chiếu với thiết kế sẽ giúp tìm ra và khắc phục vấn đề nhanh chóng.
* Là nguồn tư liệu để tham khảo cho những dự án phát triển phần mềm sau này: Đôi khi, bạn sẽ gặp một số dự án có cấu trúc tương tự nhau. Lúc này, việc tham khảo thông tin sẵn có sẽ giúp tiết kiệm được nguồn lực, rút ngắn tối đa thời gian.

1. **Kiến trúc hướng đối tượng (Objects based Architecture)**

Gửi thông điệp

Object n

Object 2

Object 3

Object 1

* Đặc tả: Hệ thống phần mềm gồm 1 tập các đối tượng độc lập được ghép nối lỏng lẻo.
* Đối tượng: là nguyên tử cấu thành phần mềm, nó có 1 số tính chất sau :
* Reusable: dễ dàng dùng lại cho ứng dụng khác
* Replaceable: dễ dàng được thay thế bằng đối tượng mới hơn
* Extensible, Heritable: thừa kế và dễ nới rộng
* Encapsulated: đóng gói và ẩn chi tiết hiện thực
* Independent: độ độc lập cao
* Persistent: thường trù để sẵn sàng phục vụ
* Aggregation: bao gộp từ nhiều đối tượng nhỏ
* Các nguyên tắc chính yếu của kiến trúc hướng đối tượng:
* Abstraction: trừu tượng
* Composition: tích hợp
* Inheritance: thừa kế
* Encapsulation: đóng gói
* Polymorphism: đa xạ
* Decoupling: quan hệ nhau (phụ thuộc) rất ít
* Ưu & nhược điểm:

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Understandable : dễ hiểu * Reusable : dễ dùng lại * Testable : dễ kiểm thử * Extensible : dễ nới rộng * Highly Cohesive : kết dính giữa các thành phần trong từng đối tượng cao | * Là mẫu kiến trúc có độ tổng quát cao nên khi hiện thực ta phải tốn nhiều chi phí để vận dụng nó. |

* Tình huống nên dùng: bất kì hệ thống phần mềm phức tạp nào.

1. **Kiến trúc hướng dịch vụ (Service-Oriented Architecture)**

* Đặc tả: Cho phép tạo phần mềm bằng cách sử dụng các dịch vụ sẵn có.

Service 1

App Module

Service n

Service 2

Internet

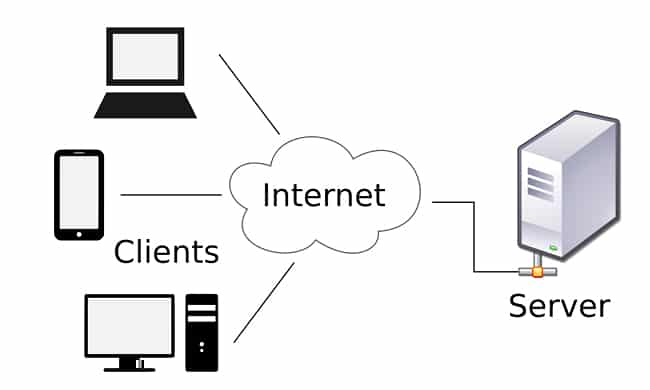
* Service: phần tử cung cấp 1 số chức năng đa dụng nào đó và thường đã có sẵn. Các nguyên tắc chính yếu của kiến trúc hướng dịch vụ là:
* Services are autonomous: tự trị
* Services are distributable: dễ dàng phân phối
* Services are loosely coupled: nối ghép rất lỏng
* Services share schema and contract, not class: dùng chung mô hình và hợp đồng chứ không phải là class cụ thể.
* Compatibility is based on policy: độ tương thích phụ thuộc vào chính sách cụ thể.
* Ưu & nhược điểm:

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| * Domain alignment * Abstraction: độ trừu tượng cao * Discoverability: dễ dàng khám phá * Interoperability: dễ dàng làm việc chung * Rationalization: hợp lý hóa | * Độ hiệu quả phụ thuộc vào cơ sở hạ tầng mạn và máy chạy service. |

* Tình huống nên dùng: bất kỳ hệ thống phần mềm phức tạp nào mà muốn chạy trên nền Internet.

1. **Kiến trúc Client – Server (client-server Architecture)**

* Mô hình Client server là một mô hình nổi tiếng trên mạng máy tính, phổ biến và được áp dụng rộng rãi ở trên các trang Web hiện nay.
* Ý tưởng của mô hình này như sau: Client (máy khách) sẽ gửi yêu cầu (request) đến Server (máy chủ, cung ứng các dịch vụ cho các máy khách). Server sẽ xử lý dữ liệu và gửi kết quả về cho Client.



Hình 3.3: Mô hình client – server

* Máy chủ: Đây thường là một mánh tính đặc biệt có tốc độ xử lý lớn và được dùng để phục vụ việc chia sẻ tài nguyên như file ảnh, file HTML, dữ liệu, máy in… cho nhiều máy khách.
* Máy khách: Là cá máy tính nhỏ như desktop hay laptop. Trong mô hình này các máy khách sẽ gửi yêu cầu tới máy chủ để máy chủ thực hiện một nhiệm vụ nào đó như lấy dữ liệu từ database, in ấn, gửi email…
* Việc giao tiếp giữa Client với Server phải dựa trên các giao thức chuẩn. Các giao thức chuẩn được sử dụng phổ biến nhất hiện nay là: giao thức TCP/IP, giao thức SNA của IBM, OSI, ISDN, X.25 hay giao thức LAN-to-LAN NetBIOS.
* Mô hình web client-server là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Một mô hình ngược lại là mô hình master-slaver, trong đó máy chủ (đóng vai trò ông chủ) sẽ gửi dữ liệu đến máy con (đóng vai trò nô lệ) bất kể máy con có cần hay không.
* Client/Server là mô hình tổng quát nhất, trên thực tế thì một server có thể được nối tới nhiều server khác nhằm làm việc hiệu quả và nhanh hơn. Khi nhận được 1 yêu cầu từ client, server này có thể gửi tiếp yêu cầu vừa nhận được cho server khác ví dụ như database server vì bản thân nó không thể xử lý yêu cầu này được. Máy server có thể thi hành các nhiệm vụ đơn giản hoặc phức tạp.

#### 3.2.2.2. Mô tả kiến trúc

Mô hình đã lựa chọn: MVC

Sử dụng mô hình MVC viết tắt của cụm từ “Model-View-Controller”.

Diagram

Description automatically generated

Hình 3.4: Thành phần cơ bản của mô hình MVC

* Model: có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller
* View: phần giao diện (theme) dành cho người sử dụng
* Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View

**Luồng xử lý trong của mô hình MVC**

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

Diagram

Description automatically generated

Hình 3.5: Luồng xử lý trong mô hình MVC

Nhóm chúng em chọn mô hình MVC để thiết kế kiến trúc vì: thứ nhất, đây là mô hình được dùng để phát triển giao diện người dùng và được chia thành 3 phần trong source code và mỗi phần đảm bảo nhiệm vụ riêng biệt. Thứ hai, kết cấu mô hình MVC khá đơn giản nên mang sự dễ hiểu cho người dùng.

### 3.2.3. Thiết kế dữ liệu

#### 3.2.3.1 Thực thể TAIKHOAN

* Thuộc tính: maTaiKhoan, tenTaiKhoan, matKhau, sdt, diaChi
* Phương thức: setMaTaiKhoan(), getMaTaiKhoan(), setTenTaiKhoan(), getTenTaiKhoan(), getMatKhau(), setMatKhau(), getSdt(), setSdt(), getDiaChi(), setDiaChi(), luu tai khoan moi(), check tai khoan(), getTaiKhoan(), createTaiKhoan(), getTaiKhoanByMaTaiKhoan(), setTaiKhoanByMaTaiKhoan(), deleteTaiKhoanByMaTaiKhoan().

Text

Description automatically generated

Hình 3.6: Thực thể TAIKHOAN

#### 3.2.3.2 Thực thể KHOAHOC

* Thuộc tính: tenKhoaHoc, maKhoaHoc, tenGiangVien, maGiangVien, gia, noiDung
* Phương thức: gettenKhoaHoc(), settenKhoaHoc(), getmaKhoaHoc(), setmaKhoaHoc(), gettenGiangVien(), settenGiangVien(), getmaGiangVien(), setmaGiangVien(), setgia(), getgia(), getnoiDung(), setnoiDung(), getKHOAHOC(), setKHOAHOC(), deleteKHOAHOCBymaKhoaHoc(), createKHOAHOC()

A picture containing text

Description automatically generated

Hình 3.7: Thực thể KHOAHOC

#### 3.2.3.3 Thực thể BAIGIANG

* Thuộc tính: maBaiGiang, tenBaiGiang, videoBaiGiang, gioiThieu, luotHoc, tenGiangVien,
* Phương thức: getmaBaiGiang(), setmaBaiGiang(), gettenBaiGiang(), settenBaiGiang(), getvideoBaiGiang(), setvideoBaiGiang(), getluotHoc(), setluotHoc(), gettenGiangVien(), getBAIGIANG(), setBAIGIANG(), deleteBAIGIANGBymaBaiGiang()



Hình 3.8: Thực thể BAIGIANG

#### 3.2.3.4 Thực thể HOCVIEN

* Thuộc tính: maHocVien, tenHocVien, sdt, phuongThucThanhToan
* Phương thức: getmaHocVien(), setmaHocVien(), gettenHocVien(), settenHocVien(), getsdt(), setsdt(), getphuongThucThanhToan(), setphuongThucThanhToan(), getHOCVIEN(), setHOCVIEN(), deleteHOCVIENBymaHocVien()



Hình 3.9: Thực thể HOCVIEN

#### 3.2.3.5 Thực thể GIANGVIEN

- Thuộc tính: maGiangVien, tenGiangVien, sdtGV

- Phương thức: getmaGiangVien(), setmaGiangVien(), gettenGiangVien(), settenGiangVien(), getsdtGV(), setsdtGV()



Hình 3.10: Thực thể GIANGVIEN

### 3.2.4. Thiết kế các thành phần phần mềm

#### 3.2.4.1. Mã giả cho chức năng Đăng nhập

Tên mã giả: Đăng nhập

Input: Tài khoản, mật khẩu của người dùng (học viên,giảng viên và người quản trị)

Output: Đăng nhập thành công vào hệ thống

BEGIN

Display nút “Đăng Nhập”

<Sau khi người dùng nhấn nút “Đăng nhập”>

DISPLAY form đăng nhập bao gồm: “Tài khoản”, “Mật khẩu”

INPUT TAIKHOAN, MATKHAU

IF (TAIKHOAN AND MATKHAU)

DISPLAY “Đăng nhập thành công”

ELSE

DISPLAY “Bạn nhập sai tài khoản hoặc mật khẩu”

END IF

END

#### 3.2.4.2. Mã giải cho chức năng Đăng ký

Tên mã giả: Đăng ký

Input: “Tên tài khoản”, “Mật khẩu”, “Xác nhận mật khẩu”, “Số điện thoại”, “Địa chỉ”,”Ngày sinh” của người dùng

Output: Đăng ký thành công

BEGIN

Display nút “Đăng ký tài khoản”

<Sau khi người dùng nhấn nút “Đăng ký tài khoản”>

Display form Đăng ký bao gồm: “Tên tài khoản”, “Mật khẩu”, “Xác nhận mật khẩu”, “Số điện thoại”, “Địa chỉ”,”Ngày sinh”

IF Người dùng nhập đúng các điều kiện bao gồm: “Tài khoản”, “Xác nhận mật khẩu”

Print “Đăng ký thành công”

ELSE

Print “Bạn làm không đúng điều kiện”

END IF

END

#### 3.2.4.3. Mã giải cho chức năng Xem khoá học

Tên mã giả: Xem khoá học

Input: Không

Output: Chi tiết khoá học muốn xem

BEGIN

Display nút “Khoá học”

<Sau khi người dùng nhấn nút “khoá học”>

Display các sách trong cơ sở dữ liệu bao gồm: “Tên khoá học”, “Ảnh minh họa”

<Khách hàng click vào “Ảnh minh họa” hoặc “Tên khoá học”>

Display chi tiết về sách mà khách hàng vừa click bao gồm: “mã khoá học”, “tên khoá học”, “tên giảng viên”, “lịch học”, “giá”

END

#### 3.2.4.4. Mã giải cho chức năng Mua khoá học

Tên mã giả: Mua khoá học

Input: “Tên khoá học”, “Tên giảng viên”, “Mã khoá học”, thông tin của khách hàng

Output: Đặt hàng thành công

BEGIN

DISPLAY nút “Mua Khoá học”

<Khách hàng nhấn nút “Mua khoá học”>

DISPLAY form “Mua khoá học” bao gồm: “Mã khoá học”, “Tên khoá học”, “Tên giảng viên”, “lịch học”, “Giá”,

INPUT MAKHOAHOC,TENKHOAHOC,TENGIANGVIEN

<Khách hàng điền đầy đủ thông tin và nhấn “mua”>

IF (MAKHOAHOC AND TENKHOAHOC AND TENGIANGVIEN)

DISPLAY form Xác nhận đặt hàng

IF Người dùng nhấn “Xác nhận”

ADD dữ liệu vào CSDL

ELSE

DISPLAY “Bạn đa hủy mua”

END IF

ELSE

DISPLAY “Bạn điền thiếu thông tin”

END IF

END

#### 3.2.4.5. Mã giải cho chức năng Bảo trì tài khoản

Tên mã giả: Quản lý tài khoản

Input: “Mã tài khoản”, “Tên tài khoản”, “Mật khẩu”, “Số điện thoại”, “Địa chỉ”,”Email”,”Ngày sinh”

Output: Thay đổi dữ liệu thành công

BEGIN

DISPLAY nút “Tài khoản”

<Quản trị viên click nút “Tài Khoản”>

DISPLAY các tài khoản gồm: “Mã tài khoản”, “Tên tài khoản”, “Mật khẩu”, “Số điện thoại”, “Địa chỉ”, ”Email”,”Ngày sinh”

IF Quản trị viên nhấn “Thêm mới”

DISPLAY bảng Thêm mới gồm: “Mã tài khoản”, “Tên tài khoản”, “Mật khẩu”, “Số điện thoại”, “Địa chỉ”, ”Email”,”Ngày sinh”

INPUT MATAIKHOAN, TENTAIKHOAN, MATKHAU, SODIENTHOAI, DIACHI,EMAIL,NGAYSINH

<Quản trị viên nhập đầy đủ thông tin và nhấn nút “Tạo”>

ADD thông tin mới vào CSDL và Display các tài khoản sau khi thêm mới

ELSE IF Quản trị viên nhấn “Sửa” trên từng dòng tài khoản

DISPLAY bảng Sửa thông tin gồm: “Mã tài khoản”, “Tên tài khoản”, “Mật khẩu”, “Số điện thoại”, “Địa chỉ”, ”Email”,”Ngày sinh”

INPUT MATAIKHOAN, TENTAIKHOAN, MATKHAU, SODIENTHOAI, DIACHI, EMAIL,NGAYSINH

<Quản trị viên sửa các thông tin và nhấn nút “Sửa”>

UPDATE thông tin trong CSDL và Display bảng tài khoản sau khi sửa

ELSE IF Quản trị viên nhấn “Xóa” trên từng dòng tài khoản

DISPLAY bảng “Xác nhận xóa”

<Quản trị viên nhấn nút “Đồng ý”>

Delete dữ liệu trong CSDL và Display bảng tài khoản sau khi xóa

END

**3.2.5. Thiết kế giao diện người sử dụng**

#### 3.2.5.1. Chức năng Đăng ký



Hình 3.11: Biểu đồ lớp Đăng kí



Hình 3.12: Biểu đồ tuần tự đăng kí

#### 3.2.5.2. Chức năng Xem khoá học



Hình 3.13: Biểu đồ lớp Xem khóa học



Hình 3.14: Biểu đồ tuần tự đăng kí

#### 3.2.5.3. Chức năng Mua khoá học



Hình 3.15: Biểu đồ lớp Mua khóa học



Hình 3.16: Biểu đồ tuần tự Mua khóa học

#### 3.2.5.4. Chức năng Đăng nhập



Hình 3.17: Biểu đồ lớp Đăng nhập



Hình 3.18: Biểu đồ tuần tự Đăng nhặp

#### 3.2.5.5. Chức năng Quản lý tài khoản

Biểu đồ lớp màn hình:



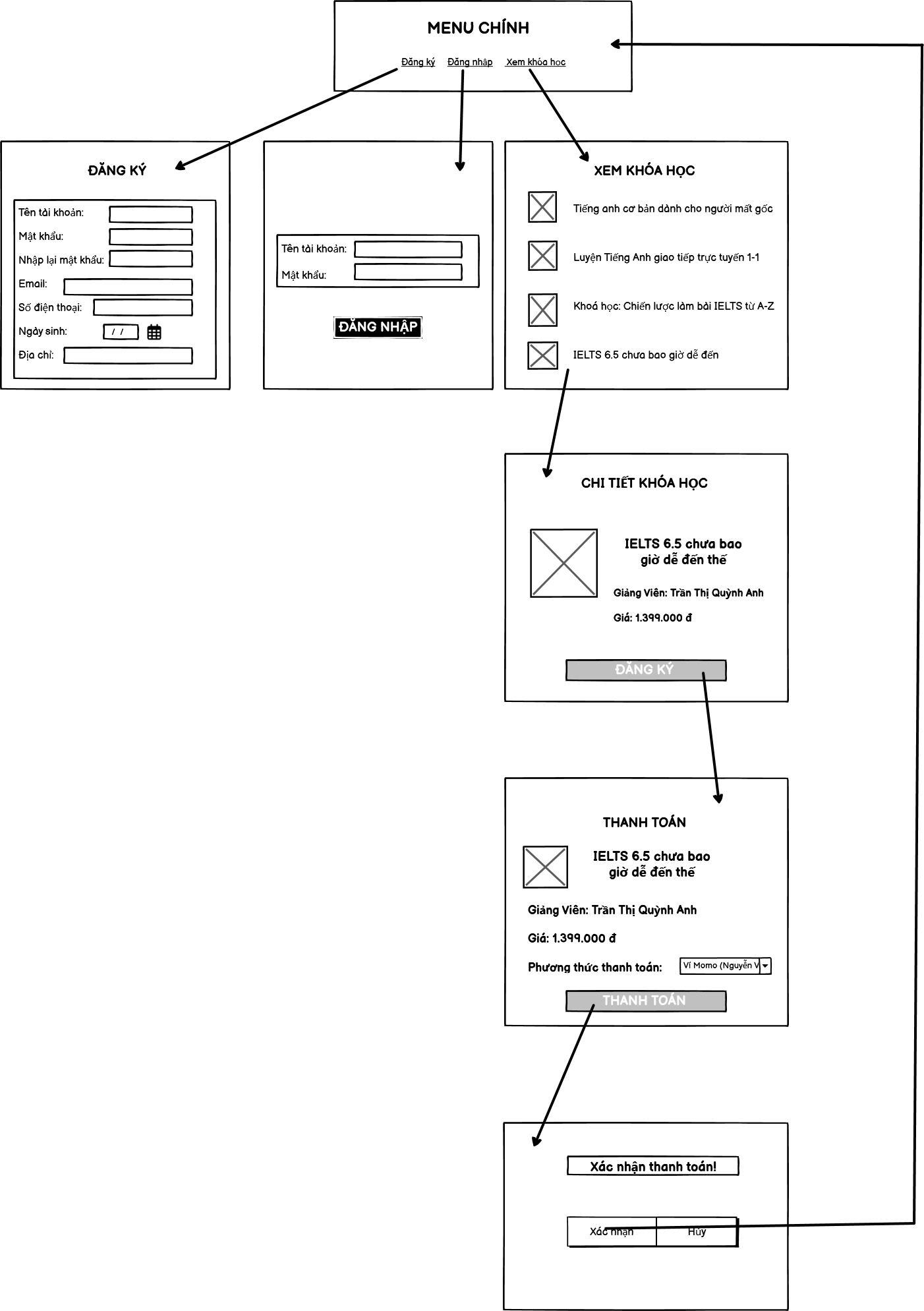
Hình 3.19: Biểu đồ lớp Quản lý tài khoản



Hình 3.20: Biểu đồ tuần tự Quản lý tài khoán

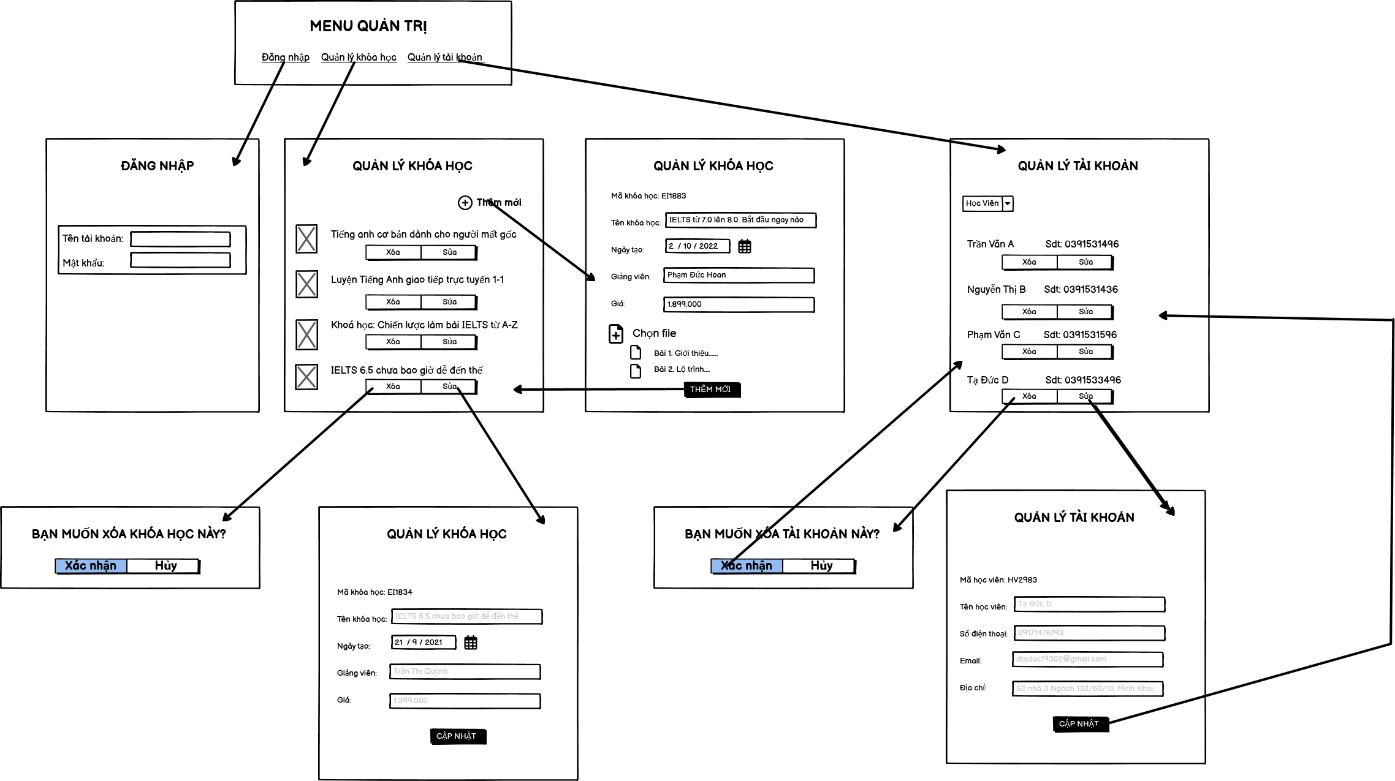
#### 3.2.5.6. Hình dung tất cả các màn hình

Hình dung màn hình front-end:



Hình 3.21: Giao diện phía front-end

Hình dung màn hình back-end:



Hình 3.22: Giao diện phía back-end

## 3.3 Kết luận chương 3

Chương 3 đã giới thiệu về quy trình thiết kế phần mềm quản lý giảng dạy của trung tâm Tiếng anh Arabica dựa trên cấu trúc tài liệu đặc tả yêu cầu phần mềm đã chuẩn bị ở chương 1 và chương 2, hoàn thiện sản phẩm phần mềm dựa trên các yếu tố cần thiết và nhất quán.

# KẾT LUẬN

## 1. Kết quả đạt được

Sau khi thực hiện để tài: ***“Tìm hiểu về mô hình quy trình Waterfall và vận dụng trong dự án thiết kế và phát triển phần mềm quản lý hoạt động giảng dạy của trung tâm tiếng anh Arabica”*** đã đạt được một số kết quả sau:

* Nắm vững quy trình phát triển phần mềm và mô hình quy trình thác nước.
* Hoàn thành đặc tả và phân tích chi tiết website giảng dạy cho trung tâm Tiếng Anh Arabica.
* Trang web được thiết kế tương đối rõ ràng và chi tiết các ca sử dụng và các chức năng được giới thiệu, mô tả chi tiết và dễ hình dung.
* Thiết kế cơ bản bố cục giao diện phần mềm.

## 2. Hướng phát triển:

* Tiếp tục tìm hiểu để hoàn thiện, mở rộng thêm các chức năng mới phù hợp với nhu cầu giảng dạy tại trung tâm tiếng anh.
* Phát triển sản phẩm dựa trên bản đặc tả và đưa vào sử dụng trong đời sống.

# **TÀI LIỆU THAM KHẢO**

[1]. Bài giảng điện tử môn " Nhập môn công nghệ phần mềm'' của thầy Nguyễn Đức Lưu.

[2]. Slide bài giảng môn " Nhập môn công nghệ phần mềm''.

[3]. Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, *Nhập môn công nghệ phần mềm*, NXB Đại học Công nghiệp Hà Nội.

[4]. Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, *Giáo trình phân tích thiết kế hệ thống*, NXB Giáo dục VN.

[5] Trang tài liệu UML <https://www.tutorialspoint.com/uml/index.htm>

[6].<https://viblo.asia/p/mo-hinh-xoan-oc-la-gi-khi-nao-nen-su-dung-uu-va-nhuoc-diem-cua-mo-hinh-vyDZOWVxZwj>

[7].<https://viblo.asia/p/tim-hieu-ve-mo-hinh-thac-nuoc-trong-phat-trien-phan-mem-stlc-Ljy5Ve7ylra>

[8].<https://tailieu.vn/docview/tailieu/2018/20180824/trieuwinner/bai_giang_cong_nghe_phan_mem_phan_tich_va_dac_ta_yeu_cau_5626.pdf?rand=241324>