|  |  |
| --- | --- |
|  |  |
| DƯƠNG THANH NGÂN | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP** |
| **KỸ THUẬT PHẦN MỀM** |
| **TÊN ĐỀ TÀI:** **PHÁT TRIỂN ỨNG DỤNG WEB HỖ TRỢ TUYỂN DỤNG SỬ DỤNG OUTSYSTEMS** |
|  |
|  |
| **CBHD :T.S Hà Mạnh Đào** |
| KỸ THUẬT PHẦN MỀM | **Sinh viên : Dương Thanh Ngân** |
| **Mã số sinh viên : 2019605480**  **Lớp : Kỹ thuật phần mềm 03** |
|  |
|  |
|  |
| Hà Nội – Năm 2023 |
|  |

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc với những thầy cô của trường Đại Học Công Nghiệp Hà Nội, đặc biệt là thầy cô khoa Công Nghệ Thông Tin đã tạo điều kiện để em có cơ hội và thời gian để làm đồ án.

Em xin chân thành cảm ơn thầy Hà Mạnh Đào đã hướng dẫn, chỉ bảo tận tình để em hoàn thành được đồ án của mình. Trong quá trình làm đồ án em còn mắc phải nhiều sai sót rất mong các thầy cô bỏ qua. Đồng thời với kinh nghiệm và kiến thức còn hạn chế em rất mong nhận được sự đóng góp từ thầy cô để sản phẩm đồ án của em được hoàn thiện hơn.

***Em xin chân thành cảm ơn!***

**MỤC LỤC**

[DANH MỤC CÁC THUẬT NGỮ, CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT 1](#_Toc146964562)

[DANH MỤC CÁC BẢNG 2](#_Toc146964563)

[DANH MỤC HÌNH ẢNH 3](#_Toc146964564)

[MỞ ĐẦU 5](#_Toc146964565)

[i. Lý do chọn đề tài 5](#_Toc146964566)

[ii. Mục tiêu nghiên cứu 5](#_Toc146964567)

[iii. Đối tượng và phạm vi nghiên cứu 5](#_Toc146964568)

[iv. Ý nghĩa khoa học và thực tiễn của đề tài 6](#_Toc146964569)

[v. Cấu trúc đồ án 6](#_Toc146964570)

[CHƯƠNG 1: TỔNG QUAN VỀ PHÁT TRIỂN ỨNG DỤNG VỚI NO-CODE/LOW-CODE 8](#_Toc146964571)

[1.1. Phát triển ứng dụng với No-code/ low-code 8](#_Toc146964572)

[1.2. Một số nền tảng, công cụ, môi trường No-code/Lowcode 20](#_Toc146964573)

[CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG NO-CODE/LOW-CODE OUTSYSTEM 23](#_Toc146964574)

[2.1 Đặc điểm của OutSystem 23](#_Toc146964575)

[2.2. Kiến trúc của OutSystem 25](#_Toc146964576)

[2.3. Các thành phần của OutSystem 26](#_Toc146964577)

[2.4. Phát triển ứng dụng No-code/Low-code với OutSystem 33](#_Toc146964578)

[CHƯƠNG 3: PHÁT TRIỂN ỨNG DỤNG VỚI OUTSYSTEM 45](#_Toc146964579)

[**3.1 Phân tích thiết kế hệ thống** 45](#_Toc146964580)

[**3.2 Giao diện ứng viên** 70](#_Toc146964581)

[**3.3 Giao diện HR, Trainer** 73](#_Toc146964582)

[KẾT LUẬN 78](#_Toc146964583)

[TÀI LIỆU THAM KHẢO 79](#_Toc146964584)

# DANH MỤC CÁC THUẬT NGỮ, CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Kí hiệu** | **Mô tả** |
| API | Application Programming Interface |
| CSDL | Cơ sở dữ liệu |
| HTTP | Hypertext Transfer Protocol |
| JS | JavaScript |
| SQL | Structured Query Language |
| UC | Usecase |
| OOTB | Out-of-the-box |
| WYSIWYG | What you see is what you get |
| HR | Human resources |
| OS | Outsystems |

# DANH MỤC CÁC BẢNG

[Bảng 1.1 Công cụ phát triển No-code 22](#_Toc146964343)

[Bảng 1.2 Nền tảng phát triển Low-code 22](#_Toc146964344)

[Bảng 2.1 Kiến trúc của Outsystems 25](#_Toc146964345)

[Bảng 2.2 Vòng đời của môi trường Outsystems 26](#_Toc146964346)

[Bảng 2.3 Các khối lệnh Font-end 29](#_Toc146964347)

[Bảng 2.4 Các khối lệnh Back-end 32](#_Toc146964348)

[Bảng 3.1 Đặc tả usecase đăng nhập 51](#_Toc146964349)

[Bảng 3.2 Đặc tả usecase đăng ký 52](#_Toc146964350)

[Bảng 3.3 Đặc tả use case nhận bài thi 53](#_Toc146964351)

[Bảng 3.4 Đặc tả usecase nộp bài thi 54](#_Toc146964352)

[Bảng 3.5 Đặc tả usecase xem thông tin cá nhân 55](#_Toc146964353)

[Bảng 3.6 Đặc tả usecase tìm kiếm ứng viên 57](#_Toc146964354)

[Bảng 3.7 Đặc tả usecase bảo trì ứng viên 59](#_Toc146964355)

[Bảng 3.8 Đặc tả usecase bảo trì bài test 61](#_Toc146964356)

[Bảng 3.9 Đặc tả usecase bảo trì lớp học 62](#_Toc146964357)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Mô hình trực quan No-code 9](#_Toc146964376)

[Hình 1.2 Chức năng ngoài hộp 9](#_Toc146964377)

[Hình 1.3 Giao diện kéo thả 10](#_Toc146964378)

[Hình 1.4 Khả năng tương tác nền tảng 11](#_Toc146964379)

[Hình 1.5 Quy mô làm việc 11](#_Toc146964380)

[Hình 1.6 Vòng đời ứng dụng 12](#_Toc146964381)

[Hình 1.7 Khả năng tái sử dụng với các khối dựng sẵn khác nhau 13](#_Toc146964382)

[Hình 1.8 Sự phát triển của Low-Code 15](#_Toc146964383)

[Hình 1.9 Lợi ích khi sử dụng Low-Code 16](#_Toc146964384)

[Hình 1.10 Tính năng phát triển Low-Code 17](#_Toc146964385)

[Hình 1.11 Quy trình phát triển ứng dụng No-code/Low-code 18](#_Toc146964386)

[Hình 2.1 Các thành phần Logic Outsystems 30](#_Toc146964387)

[Hình 2.2 Server Action 30](#_Toc146964388)

[Hình 2.3 Screen action 31](#_Toc146964389)

[Hình 2.4 Ảnh minh họa Platform Server 32](#_Toc146964390)

[Hình 2.5 Môi trường làm việc Service Studio 34](#_Toc146964391)

[Hình 2.6 Tính năng của Outsystems 37](#_Toc146964392)

[Hình 2.7 Trang đăng ký tài khoản 38](#_Toc146964393)

[Hình 2.8 Active tài khoản OS 38](#_Toc146964394)

[Hình 2.9 Tải Service Studio 39](#_Toc146964395)

[Hình 2.10 Cài đặt Service Studio 39](#_Toc146964396)

[Hình 2.11 Đăng nhập tài khoản OS 40](#_Toc146964397)

[Hình 2.12 Màn hình menu app 40](#_Toc146964398)

[Hình 2.13 Màn hình click tạo mới 41](#_Toc146964399)

[Hình 2.14 From scratch 41](#_Toc146964400)

[Hình 2.15 Chọn kiểu ứng dụng 42](#_Toc146964401)

[Hình 2.16 Tạo mới app 42](#_Toc146964402)

[Hình 2.17 Tạo module 43](#_Toc146964403)

[Hình 2.18 Tạo Screen 43](#_Toc146964404)

[Hình 2.19 Publish code 44](#_Toc146964405)

[Hình 3.1 Biểu đồ usecase tổng quát 46](#_Toc146964406)

[Hình 3.2 Usecase ứng viên 47](#_Toc146964407)

[Hình 3.3 Usecase HR 48](#_Toc146964408)

[Hình 3.4 Usecase Trainer 49](#_Toc146964409)

[Hình 3.5 Biểu đồ thực thể liên kết 50](#_Toc146964410)

[Hình 3.6 Biểu đồ tuần tự đăng ký 63](#_Toc146964411)

[Hình 3.7 Biểu đồ tuần tự đăng nhập 64](#_Toc146964412)

[Hình 3.8 Biểu đồ tuần tự nhận bài thi 65](#_Toc146964413)

[Hình 3.9 Biểu đồ tuần tự nộp bài thi 65](#_Toc146964414)

[Hình 3.10 Biểu đồ tuần tự xem thông tin cá nhân 66](#_Toc146964415)

[Hình 3.11 Biểu đồ tuần tự tìm kiếm ứng viên 66](#_Toc146964416)

[Hình 3.12 Biểu đồ tuần tự bảo trì ứng viên 67](#_Toc146964417)

[Hình 3.13 Biểu đồ tuần tự bảo tri bài test 68](#_Toc146964418)

[Hình 3.14 Biểu đồ tuần tự bảo trì lớp học 70](#_Toc146964419)

[Hình 3.15 Trang đăng ký 70](#_Toc146964420)

[Hình 3.16 Trang đăng ký 71](#_Toc146964421)

[Hình 3.17 Trang cảm ơn 71](#_Toc146964422)

[Hình 3.18 Trang đăng nhập 72](#_Toc146964423)

[Hình 3.19 Trang chủ 73](#_Toc146964424)

[Hình 3.20 Trang chủ 74](#_Toc146964425)

[Hình 3.21 Trang quản lý ứng viên 74](#_Toc146964426)

[Hình 3.22 Trang quản lý bài test 76](#_Toc146964427)

[Hình 3.23 Trang quản lý lớp học 77](#_Toc146964428)

# MỞ ĐẦU

## Lý do chọn đề tài

Ngày nay, công nghệ thông tin đang không ngừng phát triển một cách mạnh mẽ và ứng dụng rộng rãi trên nhiều lĩnh vực. Các sản phẩm công nghệ thông tin ngày càng đi sâu vào đời sống con người và trở thành một phần không thể thiếu của cuộc sống. Với xu thế tối ưu hóa các quy trình làm việc, giúp rút ngắn và công nghệ hóa các vấn đề sao lưu cũng như quản lý, đặc biệt là nhu cầu tuyển dụng ngày càng tăng cả về số lượng lẫn chất lượng. Nhận thấy được nhu cầu này hàng loạt sản phẩm hỗ trợ đã ra đời nhằm đáp ứng nhu cầu tuyển dụng online. Các sản phẩm hỗ trợ tuyển dụng cung cấp cho người quản lý sự tiện dụng trong việc tuyển dụng cũng như quản lý ứng viên.

Để bắt nhịp với xu hướng của thời đại công nghệ công ty OCG có nhu cầu xây dựng một website để giúp phòng nhân sự có thể quản lý ứng viên một cách thuận tiện mà không cần phụ thuộc quá nhiều vào việc trao đổi với từng cá nhân cũng như giấy tờ sao lưu.

## Mục tiêu nghiên cứu

Đối với một công ty, việc tuyển dụng để có thể đào tạo ra các nhân viên ưu tú là yếu tố vô cùng quan trọng. Chất lượng của nguồn nhân lực đầu vào sẽ tỉ lệ thuận với sự phát triển của công ty. Đặc biệt trong thời đại 4.0 ngày nay, vấn đề quản lý nhân sự là một yếu tố rất quan trọng trong công cuộc thúc đẩy sự phát triển của công ty để có thể cạnh tranh với các công ty khác. Chính vì thế em xin thực hiện đề tài **“Phát triển ứng dụng web hỗ trợ tuyển dụng sử dụng Outsystems”** để giải quyết được vấn đề nêu trên.

## Đối tượng và phạm vi nghiên cứu

Đối tượng và phạm vi nghiên cứu của đồ án:

- Nghiên cứu, tìm hiểu và nắm bắt được về No-code/Low-code. Từ đó đưa ra được công cụ thích hợp để phát triển sản phẩm.

- Website: Khảo sát thiết kế, chức năng của các website tuyển dụng. Từ đó đưa ra được các phương pháp giải quyết bài toán bằng Outsystems.

- Ngôn ngữ Outsystems: Tìm hiểu đặc điểm, cấu trúc, tính năng và tính ứng dụngcủa Outsystems.

- Hệ quản trị cơ sở dữ liệu MySQL: Nghiên cứu đặc điểm, tính năng, cách thức lưu trữ thông tin của hệ quản trị cơ sở dữ liệu. Tìm hiểu các ứng dụng thực tế của MySQL.

## Ý nghĩa khoa học và thực tiễn của đề tài

Đề xuất đồ án về ứng dụng web hỗ trợ tuyển dụng trong công ty nhằm phục vụ cho nhu cầu tuyển dụng, quản lý cũng như nâng cao trải nghiệm cho ứng viên về quy trình làm việc của công ty so với việc ứng tuyển thủ công như trước đây. Đồng thời giúp công ty tối ưu hóa công việc quản lý và tuyển dụng hơn so với trước đây.

Trong thực tế ứng dụng web hỗ trợ tuyển dụng không chỉ là công cụ giúp quản lý nhân viên một cách tối ưu mà còn giúp cho quy trình làm việc của công ty được rút ngắn, tối ưu và minh bạch hơn trước rất nhiều. Ngoài ra còn mang đến các lợi ích khác như: tăng khả năng quản lý nhân sự của công ty, rút ngắn được các quy tringh thủ công.

## Cấu trúc đồ án

Nội dung chính của đồ án gồm 3 chương:

* Chương 1: Tổng quan về phát triển ứng dụng với No-code/Low-code.
* Nội dung: Tổng quan và các công cụ, môi trường phát triển ứng dụng bằng nền tảng No-code/Low-code.
* Chương 2: Môi trường phát triển ứng dụng No-code/Low-code Outsystems
* Nội dung:
* Đặc điểm.
* Kiến trúc.
* Các thành phần.
* Quy trình phát triển ứng dụng bằng Outsystems.
* Chương 3: Phát triển ứng dụng với Outsystems
* Nội dung: Phân tích thiết kế hệ thống và hoàn thiện sản phẩm.
* Kết luận
* Tài liệu tham khảo

Ngoài ra còn có phần mở đầu và kết luận.

# TỔNG QUAN VỀ PHÁT TRIỂN ỨNG DỤNG VỚI NO-CODE/LOW-CODE

## Phát triển ứng dụng với No-code/ low-code

Trong kỷ nguyên số hiện tại, các công ty, các chính phủ và bản thân các lập trình viên đều chạy đua với nhau để vượt trội hơn về mặt công nghệ, một trong các công nghệ nổi bật lên là No-code/Low Code.

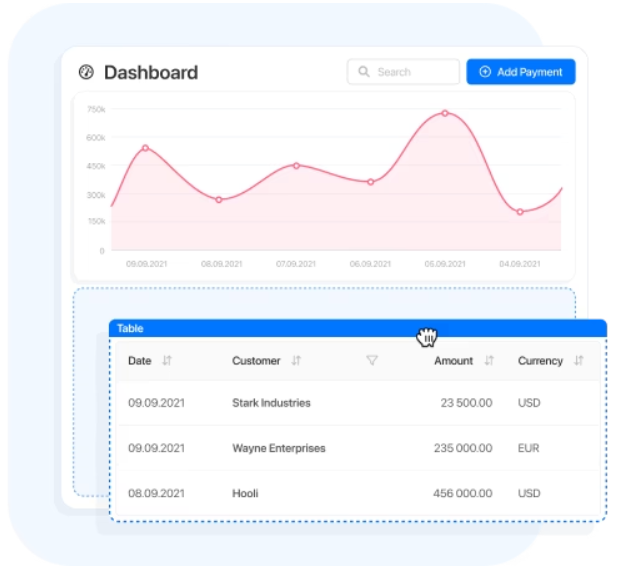
### Phát triển ứng dụng với No-code

No-code là một phương pháp luận phát triển phần mềm mới cung cấp một cách đơn giản hóa để tạo ứng dụng mà không cần bất kỳ kỹ năng viết mã nào. Các công cụ không mã ngày càng trở nên phổ biến và chúng ta có thể hiểu tại sao - chúng thường có giao diện thân thiện với người dùng, có nghĩa là chúng cho phép bạn nhanh chóng hình dung quy trình triển khai để bạn có thể mô tả chi tiết đầy đủ về logic kinh doanh tổng thể.

No-Code sử dụng trình chỉnh sửa kéo và thả; đây là một thành phần chọn giao diện hoặc mô-đun mong muốn bằng một cú nhấp chuột và ngắt chúng lại với nhau trên một không gian rộng trong khi kết quả phản ánh trong các cửa sổ được đặt tương ứng trên màn hình. No-Code tóm tắt các phần và nguyên tắc mà từ đó các chương trình phức tạp hơn được tạo ra thông qua môi trường dành cho nhà phát triển, giúp mọi người học cách viết mã đơn giản hơn, ngay cả khi họ chưa từng làm điều đó trước đây.

#### Tính năng của No-code

* Mô hình hóa trực quan:



Hình 1.1 Mô hình trực quan No-code

Tạo ứng dụng bằng mô hình trực quan nhanh hơn nhiều so với mã hóa chúng từ đầu. Các công cụ mô hình hóa trực quan cho phép biểu diễn thông tin phức tạp dưới dạng đơn giản hóa để mọi người dễ đọc. Những điều này làm cho nền tảng không mã phù hợp cho cả người dùng doanh nghiệp và nhà phát triển

* Chức năng ngoài hộp:

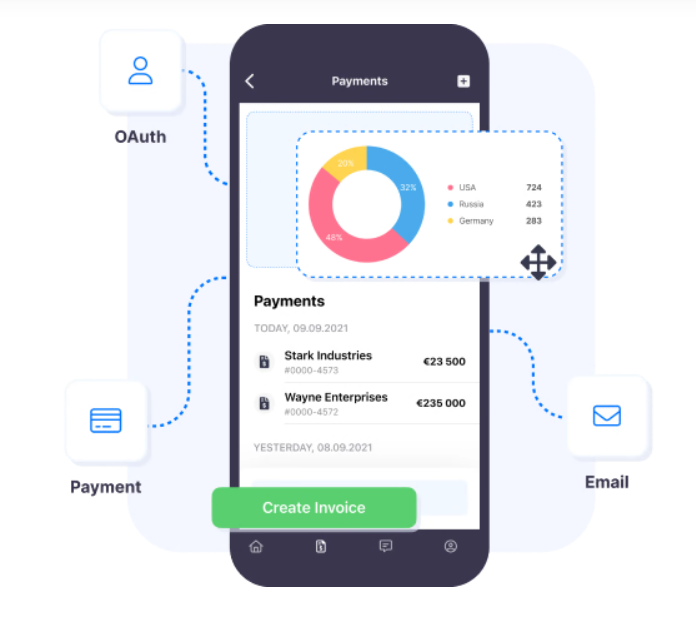
A screenshot of a computer

Description automatically generated

Hình 1.2 Chức năng ngoài hộp

Không cần phải phát minh lại bánh xe bằng cách sử dụng hệ thống không mã, hệ thống này cung cấp tất cả các chức năng bạn có thể cần cho ứng dụng của mình ngay khi bạn bắt đầu làm việc với nó. Hầu hết các nền tảng không mã đều cung cấp các mô-đun có sẵn có thể được sử dụng để giúp bạn tiết kiệm thời gian và công sức. Ví dụ: quản lý thanh toán hoặc các mô-đun đối mặt với khách hàng, như aquản lý bán hàng hoặc quản lý hỗ trợ khách hàng.

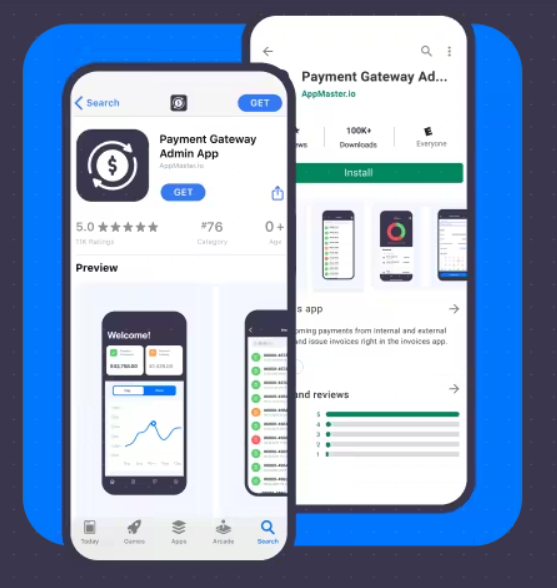
* Giao diện kéo và thả:



Hình 1.3 Giao diện kéo thả

Mọi công cụ không mã đều cung cấp một giao diện dễ sử dụng. Đây là một trong những tính năng hữu ích và thiết thực nhất giúp việc tạo ứng dụng trở nên dễ dàng. Tính năng kéo và thả là một trong những phần quan trọng và hữu ích nhất của quá trình phát triển. Họ không chỉ có thể giúp người dân dễ dàng tạo ứng dụng mà còn cho các nhà phát triển chuyên nghiệp.

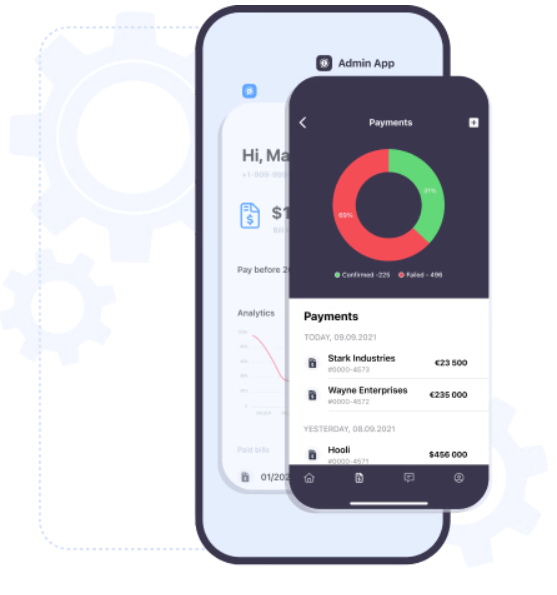
* Khả năng tương tác đa nền tảng:



Hình 1.4 Khả năng tương tác nền tảng

Các công cụ không mã cho phép bạn dễ dàng tạo các ứng dụng di động phong phú và hấp dẫn cũng như xuất bản chúng dễ dàng trên Apple Store hoặc Google Play. Xây dựng thương hiệu cho các ứng dụng của bạn bằng biểu tượng thương hiệu, màu sắc của công ty và các yếu tố khác của nhận dạng hình ảnh của bạn. Bỏ qua nhiều tháng viết mã bằng cách sử dụng các công cụ phát triển kéo và thả.

* Quy mô khi làm việc



Hình 1.5 Quy mô làm việc

Nền tảng No-code giúp dễ dàng hơn trong việc xây dựng các ứng dụng mới có thể đối phó với sự gia tăng số lượng khách hàng, tăng trưởng kinh doanh và tất cả những thay đổi khác trong chiến lược hoặc sự phát triển của bạn. Thêm năng lực và quy mô với các yêu cầu ngày càng tăng của doanh nghiệp của bạn.

* Quản lý vòng đời ứng dụng:

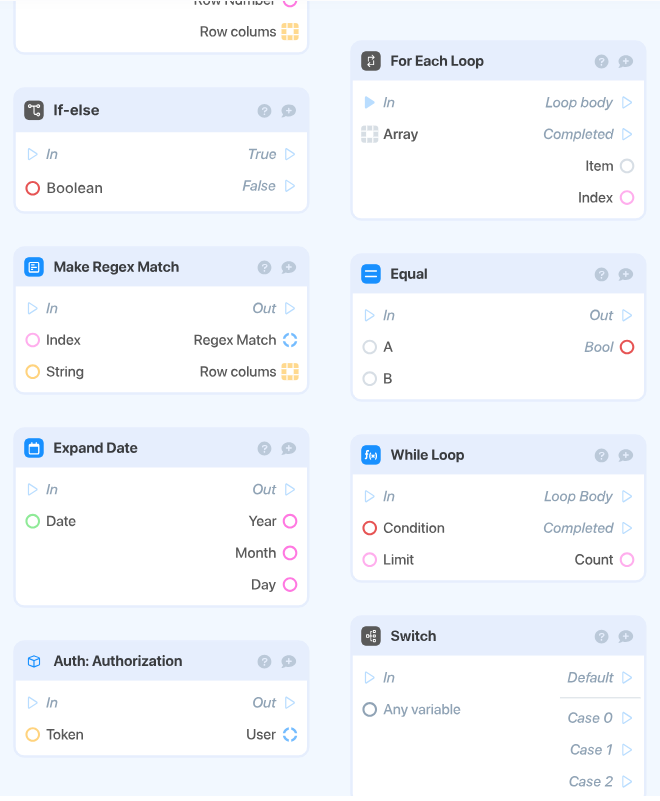
A computer screen with a blue arrow

Description automatically generated

Hình 1.6 Vòng đời ứng dụng

No-code sắp xếp hợp lý tất cả các giai đoạn trong vòng đời phát triển phần mềm, chẳng hạn như gỡ lỗi, kiểm tra và triển khai. Người dùng cũng có thể quay lại các phiên bản trước của ứng dụng nếu họ cảm thấy nó sẽ giải quyết được các vấn đề trong ứng dụng hiện tại của họ.

* Khả năng tái sử dụng với các khối dựng sẵn khác nhau:



Hình 1.7 Khả năng tái sử dụng với các khối dựng sẵn khác nhau

Kết nối ứng dụng web và thiết bị di động cũng như quy trình công việc của bạn với các nền tảng bên thứ ba để có luồng dữ liệu không giới hạn. Tích hợp với CRM, cổng thanh toán, cơ sở dữ liệu bên ngoài, API, v.v. trong một vài bước.

#### Ưu điểm của No-code

* Giảm chi phí: Các giải pháp không mã giảm chi phí phát triển lên đến 80%. No-code thu hẹp khoảng cách CNTT mà không phải chịu thêm chi phí tài nguyên.
* Tăng tốc độ: Thời gian để phát triển một ứng dụng mất hàng tháng. Các nhà phát triển và doanh nghiệp đang tìm cách cung cấp ứng dụng nhanh hơn. Lần này không có mã giảm một nửa hoặc thậm chí ít hơn.
* Ở phía trước: Không mã là tương lai. Các doanh nghiệp triển khai nó ngay hôm nay, hãy đi trước đón đầu. No-code là một công cụ hoàn hảo để cải tiến liên tục.
* Tạo ứng dụng tốt hơn: Việc phát triển không có mã dẫn đến tỷ lệ thành công tốt hơn cho các ứng dụng. Tạo ứng dụng nhanh hơn nhiều và nhận phản hồi tức thì.
* Tăng tốc hoạt động: Nền tảng không mã là điểm tiếp xúc hoàn hảo cho nhóm của bạn. Xây dựng, bắt đầu và xử lý quy trình làm việc mà không mất thời gian cho những công việc đơn điệu, thường ngày.
* Tăng sự nhanh nhẹn: Phát triển hiệu quả và nhanh chóng thông qua các công cụ mô hình hóa trực quan mang đến cho các doanh nghiệp cơ hội thích ứng nhanh chóng với sự thay đổi của thị trường và nhu cầu của khách hàng.

### Phát triển ứng dụng với Low-code

Low Code là một phương pháp phát triển phần mềm yêu cầu ít hoặc không cần viết code để xây dựng các ứng dụng và quy trình. Thay vì lập trình viên phải sử dụng ngôn ngữ lập trình phức tạp. Theo mô hình Low-Code, lập trình viên có thể sử dụng các giao diện trực quan với các logic ở mức bình thường để phát triển ứng dụng.

Các công cụ low code thường cung cấp các thành phần đầu vào, giao diện trực quan và các trình điều khiển dữ liệu để người dùng có thể kết hợp và tùy chỉnh để tạo ra các ứng dụng tùy chỉnh. Nhiều nền tảng low code cũng cung cấp các khối xây dựng sẵn để người dùng có thể kéo và thả vào một mô hình phát triển bản đồ để tạo ra các ứng dụng nhanh chóng.

Low code cũng cho phép các nhà phát triển và các doanh nghiệp tiết kiệm chi phí và thời gian cho việc phát triển ứng dụng. Điều này là do người dùng không cần phải bỏ ra nhiều thời gian và tiền bạc để thuê các nhà phát triển chuyên nghiệp mà có thể sử dụng các công cụ low code để tạo ra các ứng dụng theo yêu cầu một cách nhanh chóng.

#### Low code và hướng phát triển trong tương lai

Công bằng mà nói thì Low Code là cho người sử dụng nền tảng (platform). Còn bản thân lập trình viên vẫn phải là người tạo ra các component, các model có thể trên platform. Nên nếu nói Low-Code là không code hoặc ít code thì thực chất đó là dành cho người sử dụng. Không phải các lập trình viên tạo nên các platform Low-Code.

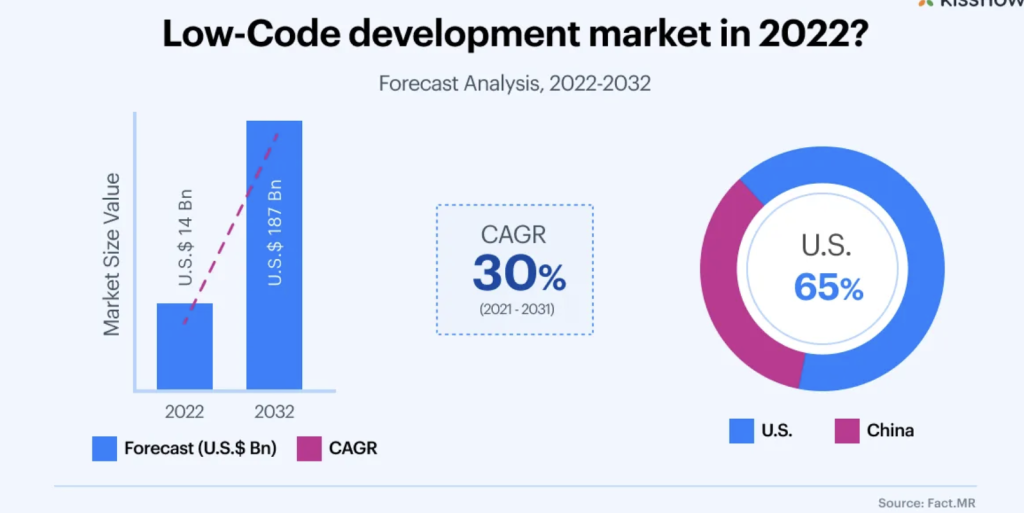
Sự phát triển của Low-Code các năm gần đây được xem như một hiện tượng, khi mà số hoá và chuyển đổi số đi sâu vào từng ngõ ngách trong cuộc sống:

- Theo như báo cáo mới nhất của Gartner, Inc’s thì thị trường toàn cầu cho các công ty phát triển Low Code sẽ đạt khoảng 26.9 tỷ đô vào năm 2023 (tăng 19.6% so với năm 2022).

- Trong tương lai thị trường sẽ đạt khoảng 65 tỷ Biden vào năm 2027 và 187 tỷ Biden vào năm 2030.

- Thực tế mà nói thị trường Low-Code đã tăng 30% từ năm 2020 cho tới năm 2023 và tạo ra 5,8 tỷ Biden năm 2022.

- Thị trường rộng mở , tương lai sáng ngời và chói lọi.



Hình 1.8 Sự phát triển của Low-Code

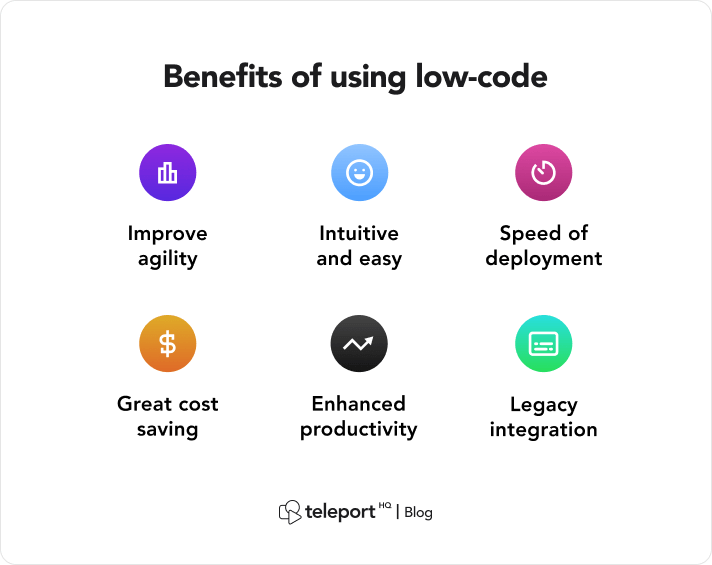
#### Lợi ích đem lại cho businees onwer

Rõ ràng là Low Code sẽ đem lại kết quả nhanh chóng và nhìn thấy được cho các chủ doanh nghiệp. Quá trình phát triển cũng trở nên cực kì nhanh chóng.

* Tăng độ nhanh nhạy

Với Low-Code, các component, model, thậm chí tới các flow phức tạp đều đã được dựng lên từ trước. Việc này giúp việc triển khai trở nên cực kì nhanh chóng. Người dùng không cần phải có kiến thức sâu rộng về lập trình để tạo ra sản phẩm.

Việc này cực kì hữu ích khi mô hình kinh doanh thay đổi nhanh chóng (vô cùng phổ biến ở thời điểm này). Các doanh nghiệp có lợi thế khi họ có thể nhanh chóng thay đổi theo.



Hình 1.9 Lợi ích khi sử dụng Low-Code

* Giảm giá thành

Tất nhiên là việc sử dụng Low Code platform sẽ tạo ưu thế cạnh tranh về giá thành. Mọi người cũng biết là chi phí phát triển phần mềm từ trước tới nay chưa bao giờ là rẻ.

Với các platform đã phát triển Low-Code thì ngược lại. Công sức chỉ bỏ ra một lần ban đầu và tái sử dụng nhiều lần sau đó. Việc này giúp doanh nghiệp giảm được chi phí ở mức tối đa.

* Xuất hiện nhanh chóng

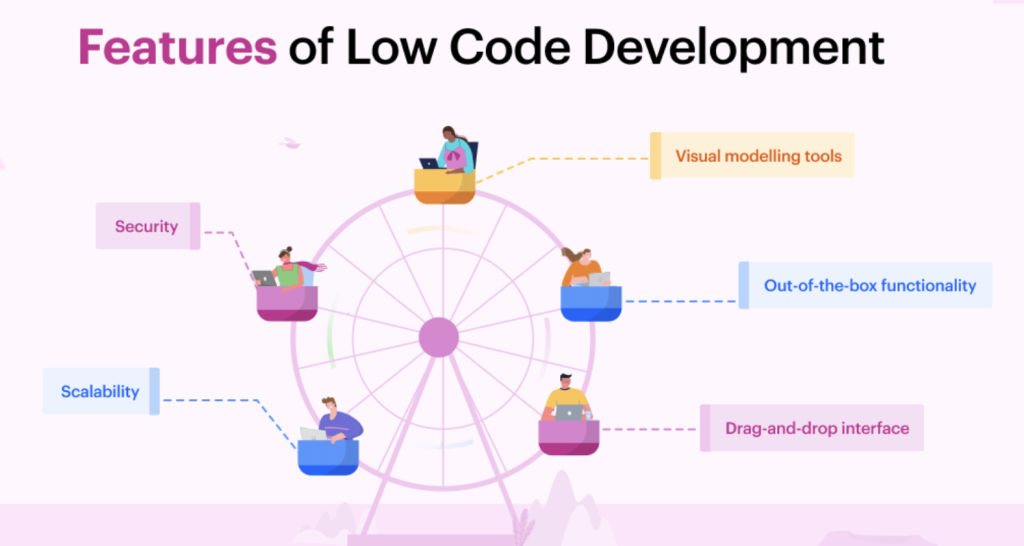
Ở cái thời đại mà ý tưởng chỉ vừa với ra đầu môi thì đã có ngay ứng dụng trên store thì Low-Code giúp doanh nghiệp triển khai ý tưởng nhanh chóng. Thời gian phát triển giảm xuống đồng nghĩa với việc business xuất hiện trên thị trường nhanh hơn. Tạo ưu thế cạnh tranh so với đối thủ.

#### Tính năng của Low code

Sau khi đã hiểu rõ về lợi thế, ưu điểm của Low-Code thì giờ là lúc lướt qua một số tính năng nổi bật của nó. Đầu tiên không thể không nhắc tới Visual Modeling Tools.

* Visual Modeling Tool

Visual Modeling Tools ở đây được hiểu là trực quan hoá các mô hình. Các mô hình phổ biến sẽ được xây dựng sẵn. Các mô hình này sẽ được xây dựng cực kì trực quan, giúp người sử dụng dễ nắm bắt.



Hình 1.10 Tính năng phát triển Low-Code

Ngoài ra, người sử dụng Visual Modeling Tools cũng không cần hiểu biết nhiều về công nghệ. Dễ dàng hơn cho các doanh nghiệp.

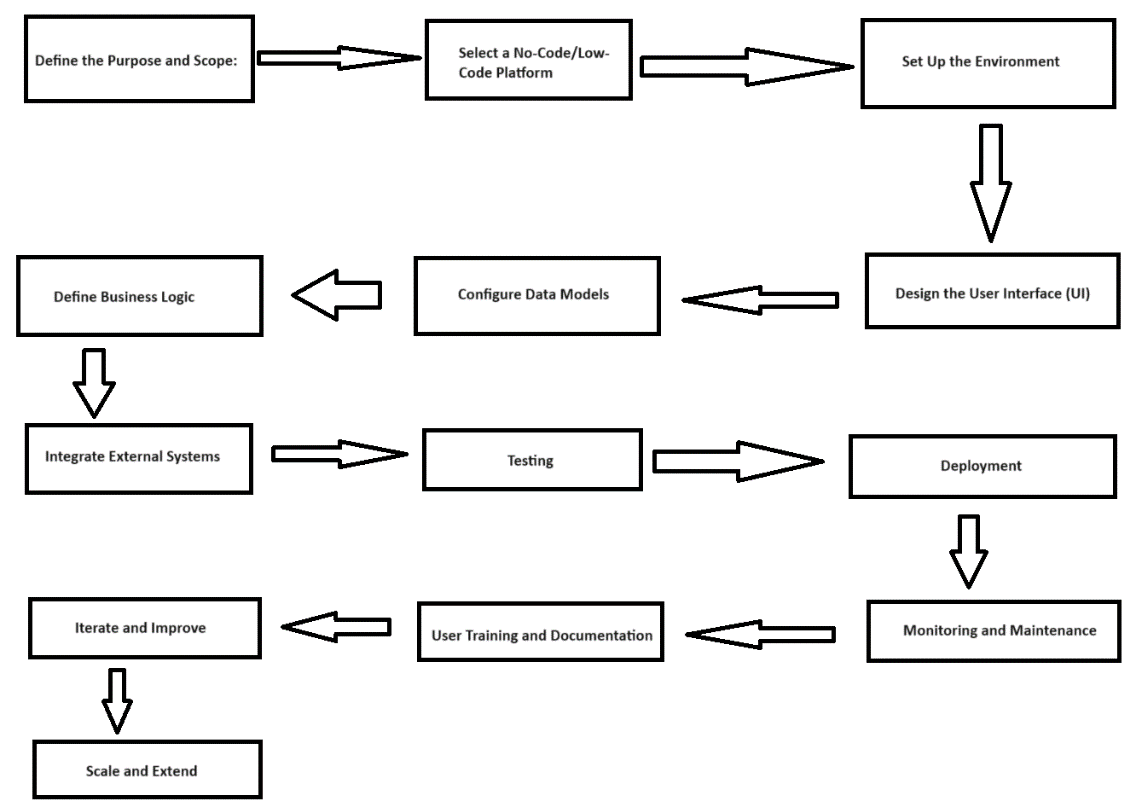
* Out-of-the-box Function

OOTB, viết tắt của (out-of-the-box) là tính năng được phát triển bởi các công ty phát triển hàng đầu cho Low-Code. Bản thân nó giúp người sử dụng có thể bỏ qua các khởi tạo ban đầu khi bắt đầu sử dụng Low Code.

* Drag and Drop interface

Kế tiếp là feature về [Drag và Drop](https://topdev.vn/blog/selenium-webdriver-xu-ly-thao-tac-drag-drop/), suy luận một chút thì hiểu ngay tính năng này quan trọng thế nào với Low-Code. Các model, component có sẵn tất nhiên chỉ cần kéo thả vào là hoạt động.

### Quy trình phát triển ứng dụng No-code/Low-code



Hình 1.11 Quy trình phát triển ứng dụng No-code/Low-code

Bước 1: Xác định mục đích và phạm vi:

* Hiểu rõ mục đích của ứng dụng và vấn đề mà nó dự định giải quyết.
* Xác định các tính năng và chức năng cần thiết.

Bước 2: Chọn nền tảng No-code/Low-code:

* Chọn nền tảng mã nguồn thấp hoặc môi trường phát triển phù hợp với yêu cầu dự án của bạn. Các ví dụ bao gồm Microsoft Power Apps, Mendix, OutSystems và các ứng dụng khác.

Bước 3: Thiết lập môi trường:

* Cài đặt và định cấu hình nền tảng mã thấp đã chọn trên môi trường phát triển.

Bước 4: Thiết kế giao diện người dùng (UI):

* Sử dụng giao diện trực quan được cung cấp bởi nền tảng mã thấp để thiết kế giao diện người dùng.
* Các phần tử kéo và thả như nút, biểu mẫu và tiện ích con để tạo bố cục.

Bước 5: Định cấu hình mô hình dữ liệu:

* Xác định các mô hình và cấu trúc dữ liệu cần thiết cho ứng dụng.

Bước 6: Thiết lập cơ sở dữ liệu hoặc kết nối với các nguồn dữ liệu hiện có.

* Xác định logic nghiệp vụ:
* Sử dụng trình soạn thảo trực quan hoặc giao diện tập lệnh để xác định logic nghiệp vụ của ứng dụng.
* Điều này có thể liên quan đến việc tạo quy trình làm việc, xác định quy tắc và triển khai tự động hóa.

Bước 7: Tích hợp hệ thống bên ngoài:

* Kết nối ứng dụng với các hệ thống hoặc dịch vụ bên ngoài bằng API hoặc trình kết nối.
* Điều này có thể bao gồm việc tích hợp với cơ sở dữ liệu, dịch vụ của bên thứ ba hoặc thiết bị IoT.

Bước 8: Testing:

* Kiểm tra kỹ lưỡng ứng dụng để đảm bảo rằng nó đáp ứng các yêu cầu và chức năng đã xác định như mong đợi.
* Kiểm tra khả năng sử dụng, chức năng, bảo mật và hiệu suất.

Bước 9: Deployment:

* Sau khi ứng dụng vượt qua quá trình kiểm tra, nó có thể được triển khai lên môi trường lưu trữ.
* Một số nền tảng mã thấp cung cấp các tùy chọn triển khai dựa trên đám mây.

Bước 10: Giám sát và bảo trì:

* Thiết lập các công cụ giám sát để theo dõi hiệu suất và cách sử dụng của ứng dụng.
* Giải quyết mọi vấn đề hoặc lỗi có thể phát sinh và thực hiện cập nhật hoặc cải tiến nếu cần.

Bước 11: Hướng dẫn người dùng và tài liệu hướng dẫn:

* Cung cấp đào tạo và tài liệu cho người dùng cuối về cách sử dụng ứng dụng một cách hiệu quả.

Bước 12: Lặp lại và cải thiện:

* Thu thập phản hồi từ người dùng và các bên liên quan để xác định các lĩnh vực cần cải thiện.
* Thực hiện các bản cập nhật lặp đi lặp lại để nâng cao chức năng và khả năng sử dụng của ứng dụng.

Bước 13: Quy mô và mở rộng:

* Khi ứng dụng thu hút được sự chú ý và các yêu cầu ngày càng phát triển, hãy cân nhắc mở rộng quy mô ứng dụng hoặc mở rộng khả năng của ứng dụng.

## Một số nền tảng, công cụ, môi trường No-code/Lowcode

### No-code:

* Nền tảng No-code: Ngày nay, ngày càng có nhiều nền tảng no-code khác nhau về tính năng, ứng dụng và mức độ phức tạp của các tình huống tự động hóa. Một số loại nền tảng no-code cấp doanh nghiệp được sử dụng rộng rãi nhất bao gồm:
* Nền tảng cho các ứng dụng B2E toàn diện: Cho phép bạn tạo các ứng dụng nội bộ đa năng để hỗ trợ các hoạt động hàng ngày của doanh nghiệp. Sử dụng các nền tảng này, bạn có thể xây dựng các ứng dụng cho các nhiệm vụ tổ chức thông thường, chẳng hạn như quản lý các yêu cầu (nghỉ phép, đi lại, chi phí, v.v.), chia sẻ và sắp xếp thông tin hoặc tự động hóa quy trình phê duyệt.
* Nền tảng cho GPA và các ứng dụng dựa trên quy trình: Cho phép bạn xây dựng tự động hóa quy trình kỹ thuật số phức tạp, quy trình dựa trên BPMN, quy trình quản lý trường hợp, v.v., liên quan đến logic kinh doanh phức tạp. Dù bạn làm việc trong ngành nào –Dịch vụ Tài chính, Sản xuất, Công nghệ cao, Viễn thông hay Khu vực công – các công cụ no-code có thể giúp bạn triển khai quy trình công việc kỹ thuật số cho tổ chức của mình để tăng năng suất và đảm bảo hoạt động tối ưu.
* Nền tảng cho các ứng dụng hướng tới khách hàng và CRM: Chẳng hạn như Creatio – cho phép bạn tạo các ứng dụng để tự động hóa marketing, bán hàng và chăm sóc khách hàng. Không giống như các nền tảng CRM truyền thống với các tính năng và ứng dụng OOTB, các nền tảng CRM no-code cho phép người dùng hoàn toàn tự do tùy chỉnh các ứng dụng CRM theo nhu cầu và trường hợp sử dụng riêng của họ.
* Một số công cụ phát triển ứng dụng No-code:

|  |  |
| --- | --- |
| Công cụ | Chức năng |
| Webflow Raises $120M Series C at $4B Valuation Led by YC Continuity | Xây dựng một trang web và tích hợp các dịch vụ khác cần thiết để làm cho trang web của bạn có đầy đủ chức năng |
| Draftbit: Build mobile apps visually. Source-code included. | Y Combinator | Tạo ứng dụng di động iOS hoặc Android có sẵn trong Apple Store hoặc Play Store hoặc dưới dạng PWA |
| Figma là gì? Ứng dụng, ưu điểm, cách sử dụng Figma | CareerBuilder.vn | Thiết kế giao diện và tích hợp chặt chẽ với các công cụ NoCode khác để tạo ra các công cụ thực thi liền mạch |
| Bubble Review | PCMag | Sử dụng nền tảng NoCode để tạo thành phẩm hoặc ứng dụng mà bạn bán cho khách hàng |

Bảng 1.1 Công cụ phát triển No-code

### Low-code

* Nền tảng Low-code:

Các nền tảng Low-code được thiết kế để làm cho quá trình phát triển phần mềm nhanh nhẹn hơn, thân thiện với người dùng và có thể mở rộng. Dưới đây là một số nền tảng phổ biến hiện nay:

|  |  |
| --- | --- |
| Nền tảng | Chức năng |
| OutSystems Review | PCMag | Cung cấp một IDE có thể cài đặt để phát triển các ứng dụng. |
| Microsoft PowerApps | Tangentia | Cho phép người dùng tạo các ứng dụng kinh doanh mà không cần viết mã. Nền tảng này sử dụng giao diện kéo và thả để xây dựng các ứng dụng từ một tập hợp các thành phần được tạo sẵn. |
| AppSheet Reviews 2023: Details, Pricing, & Features | G2 | Cho phép mọi người tạo và xuất bản ứng dụng mà không cần bất kỳ mã hóa nào.Google AppSheet là duy nhất vì nó là một nền tảng Low-code cho phép người dùng tạo và chia sẻ các ứng dụng dành cho thiết bị di động mà không cần phải viết bất kỳ mã nào. |
| A black and grey logo  Description automatically generated | Các ứng dụng được xây dựng bằng cách tạo các mô hình xác định cấu trúc và hành vi của ứng dụng. |

Bảng 1.2 Nền tảng phát triển Low-code

# MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG NO-CODE/LOW-CODE OUTSYSTEM

## 2.1 Đặc điểm của OutSystem

OutSystems là một nền tảng low code được thiết kế để giúp người dùng dễ dàng tạo ra các ứng dụng doanh nghiệp một cách nhanh chóng và hiệu quả. OutSystems cho phép người dùng kéo và thả các thành phần trực quan để tạo ra các ứng dụng, bao gồm cả các ứng dụng di động, web và các ứng dụng đa nền tảng.

Một số tính năng chính của OutSystems bao gồm:

* Mô hình hóa dữ liệu: OutSystems cho phép người dùng mô hình hóa dữ liệu và liên kết dữ liệu giữa các thành phần khác nhau để tạo ra các ứng dụng đáp ứng nhu cầu của doanh nghiệp.
* Công cụ thiết kế giao diện: OutSystems cung cấp các công cụ thiết kế giao diện trực quan để giúp người dùng thiết kế các trang web và các ứng dụng di động một cách dễ dàng.
* Mô hình hóa quy trình: OutSystems cho phép người dùng mô hình hóa quy trình và thiết kế các luồng công việc tự động để tối ưu hóa quy trình làm việc của doanh nghiệp.
* Các khối xây dựng sẵn: OutSystems cung cấp các khối xây dựng sẵn để người dùng có thể tạo ra các ứng dụng nhanh chóng và tiết kiệm thời gian.
* Hỗ trợ đa nền tảng: OutSystems hỗ trợ tạo ra các ứng dụng đa nền tảng, bao gồm cả các ứng dụng di động cho các nền tảng iOS và Android.
* OutSystems cũng cung cấp các tính năng bảo mật và quản lý, bao gồm tính năng tự động kiểm tra lỗi và tối ưu hóa mã, giúp giảm thiểu rủi ro và đảm bảo tính ổn định của ứng dụng. Nền tảng này được sử dụng bởi nhiều doanh nghiệp lớn trên toàn thế giới để tạo ra các ứng dụng doanh nghiệp đáp ứng nhu cầu của họ một cách nhanh chóng và hiệu quả.
* Mặc dù OutSystems là một nền tảng low-code mạnh mẽ và linh hoạt, tuy nhiên nó vẫn có một số nhược điểm như sau:
* Giới hạn về tùy chỉnh: Mặc dù OutSystems cung cấp nhiều khối xây dựng và mô-đun, tuy nhiên nó có giới hạn trong việc tùy chỉnh và mở rộng các tính năng khác nhau. Điều này có nghĩa là các lập trình viên sẽ phải tuân theo những giới hạn này khi phát triển ứng dụng, điều này có thể làm giảm tính linh hoạt và khả năng đáp ứng nhu cầu của doanh nghiệp.
* Giới hạn đối với quản lý mã nguồn: OutSystems sử dụng mã nguồn độc quyền, điều này có nghĩa là các lập trình viên không thể truy cập mã nguồn của ứng dụng để chỉnh sửa hoặc tùy chỉnh các tính năng. Điều này có thể gây khó khăn cho các nhà phát triển khi cần thay đổi hoặc sửa lỗi trong các ứng dụng.
* Khó khăn trong việc tích hợp với các hệ thống khác: Mặc dù OutSystems cho phép tích hợp với các hệ thống bên ngoài, tuy nhiên quá trình tích hợp này có thể khó khăn và tốn nhiều thời gian. Điều này có thể làm giảm tính linh hoạt và độ mở rộng của các ứng dụng phát triển bằng OutSystems.

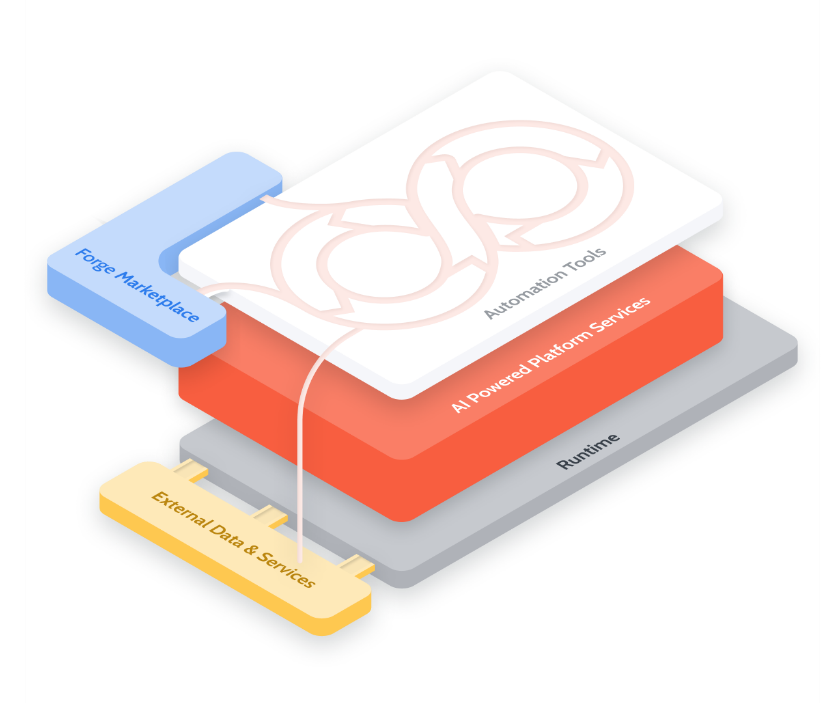
Một số ứng dụng phổ biến được phát triển bằng OutSystems bao gồm:

* Ứng dụng quản lý khách hàng: các doanh nghiệp có thể sử dụng OutSystems để phát triển các ứng dụng quản lý khách hàng để quản lý thông tin khách hàng, quản lý tương tác với khách hàng và tối ưu hóa các chiến dịch tiếp thị.
* Ứng dụng quản lý kho: OutSystems cho phép phát triển các ứng dụng quản lý kho để quản lý lưu trữ và vận chuyển hàng hóa trong doanh nghiệp.
* Ứng dụng quản lý dự án: OutSystems cũng cho phép phát triển các ứng dụng quản lý dự án để quản lý các dự án trong doanh nghiệp, bao gồm quản lý lịch trình, tài nguyên và ngân sách.
* Ứng dụng quản lý nhân sự: OutSystems cung cấp các khối xây dựng để phát triển các ứng dụng quản lý nhân sự để quản lý thông tin nhân viên, lịch sử làm việc, chấm công và bảo hiểm.
* Ứng dụng kinh doanh đa kênh: OutSystems cho phép phát triển các ứng dụng kinh doanh đa kênh để tạo ra trải nghiệm đồng nhất cho khách hàng trên nhiều nền tảng, bao gồm cả web và di động.

Với OutSystems, các ứng dụng có thể được phát triển nhanh chóng, giảm thiểu thời gian và chi phí phát triển và đảm bảo tính ổn định và bảo mật của ứng dụng.

## 2.2. Kiến trúc của OutSystem

Kiến trúc OutSystems là một hệ sinh thái phân lớp cho phép các nhà phát triển xây dựng ứng dụng nhanh chóng, xây dựng đúng cách và xây dựng chúng cho tương lai.



Bảng 2.1 Kiến trúc của Outsystems

Lớp trên cùng của kiến trúc bao gồm các công cụ, kho lưu trữ, trình xây dựng, quy trình và thành phần được tích hợp AI để loại bỏ các khía cạnh thách thức của việc tích hợp ứng dụng và dữ liệu, quản lý SDLC cũng như thiết kế và phát triển ứng dụng. Lớp này được củng cố bởi lớp dịch vụ hỗ trợ AI để tự động hóa các quy trình xem xét kiến trúc và quản lý thay đổi phức tạp, loại bỏ các tác vụ lặp đi lặp lại và phỏng đoán khỏi quá trình phát triển ứng dụng, cung cấp hơn 250 điểm bảo mật, tạo mã và ghi nhật ký.

Lớp thời gian chạy được tách riêng khỏi quá trình phát triển, cho phép nó cung cấp tùy chọn triển khai trên đám mây hoặc tại chỗ.

Nhờ kiểu kiến trúc này, Outsystem mang đến cho nhà phát triển rất nhiều lợi ích để từ đó có thể xây dựng ứng dụng:

* Nhanh chóng: cho phép các nhà phát triển xây dựng các giải pháp phần mềm hoàn chỉnh và hấp dẫn một cách nhanh chóng và hiệu quả, với môi trường phát triển trực quan và các công cụ tự động hóa và tăng cường các giai đoạn quan trọng của vòng đời phát triển phần mềm (SDLC).
* Đúng cách: Kiến trúc OutSystems bao gồm các công cụ tích hợp và cơ sở hạ tầng cần thiết để đảm bảo rằng các ứng dụng hiện đại, cấp doanh nghiệp được an toàn, linh hoạt, sẵn sàng cho đám mây và được xây dựng theo quy mô
* Cho tương lai: Kiến trúc OutSystems được thiết kế để giúp các nhà phát triển và bộ phận CNTT quản lý sự thay đổi. Các dịch vụ nền tảng, AI và các công cụ trực quan cho phép giới thiệu liên tục các tính năng và khả năng để nhà phát triển có thể xây dựng các ứng dụng phát triển nhanh chóng khi các thay đổi trong kinh doanh và công nghệ được giới thiệu

## 2.3. Các thành phần của OutSystem

### Môi trường:

OutSystems là một nền tảng ứng dụng hiện đại được thiết kế để tăng tốc đáng kể sự phát triển của các ứng dụng quan trọng nhất của công ty đồng thời mang lại mức độ linh hoạt và hiệu quả chưa từng có. Các nhà phát triển sử dụng một môi trường phát triển tích hợp , duy nhất bao trùm toàn bộ vòng đời phát triển : phát triển, đảm bảo chất lượng, triển khai, giám sát và quản lý.

A screenshot of a software application

Description automatically generated

Bảng 2.2 Vòng đời của môi trường Outsystems

Toàn bộ ứng dụng được xây dựng trên một môi trường phát triển tích hợp duy nhất bằng cách sử dụng phương pháp mã thấp. Điều này bao gồm front-end, back-end, cơ sở dữ liệu và tích hợp với các hệ thống hoặc dịch vụ hiện có.

* Xây dựng giao diện người dùng của ứng dụng web hoặc thiết bị di động của bạn bằng cách sử dụng các mẫu và khối dựng sẵn hoạt động trên nhiều thiết bị. Bạn có thể mở rộng giao diện người dùng bằng HTML, JavaScript và CSS tiêu chuẩn và truy cập mọi khả năng của thiết bị (máy ảnh, lịch, gps, v.v.) bằng cách tạo các plugin đơn giản.
* Tất cả các khía cạnh của back-end đều được phát triển trực quan, bao gồm API, dịch vụ web, quy trình công việc và quy tắc kinh doanh. Bạn có thể mở rộng khả năng của nền tảng bằng mã tùy chỉnh.
* Lập mô hình cơ sở dữ liệu ứng dụng của bạn một cách trực quan và kết nối với mọi nguồn dữ liệu hiện có.
* Sử dụng trình kết nối nguồn mở hoặc trình hướng dẫn tích hợp để kết nối với các gói phần mềm, ứng dụng hiện tại hoặc cơ sở dữ liệu hiện có. Bạn có thể sử dụng hoặc hiển thị các dịch vụ web một cách trực quan hoặc xây dựng trình kết nối của riêng mình bằng mã tùy chỉnh.

### Các khối lệnh, thành phần của Font-end và Back-end:

#### Font-end

Widget trong OutSystems là các thành phần trực quan giúp ta thiết kế và sắp xếp giao diện người dùng ứng dụng của mình. Chúng đóng vai trò là khối xây dựng để tăng tốc phát triển giao diện người dùng.

Chúng ta có thể kéo các tiện ích vào UI Screen, sau đó đặt các thuộc tính liên quan của chúng. Các widget xuất hiện trong hộp công cụ Service Studio, ở phía bên trái màn hình chính. Các vật dụng có sẵn sẽ khác nhau tùy theo loại ứng dụng.

Ta có thể sử dụng các tiện ích để tăng tốc các tác vụ phát triển sau, trong số nhiều tác vụ khác:

* Hiển thị một hoặc nhiều bản ghi hoặc dòng thông tin bằng các tiện ích hiển thị, chẳng hạn như Table widgets
* Nhận thông tin đầu vào từ người dùng bằng các tiện ích như Text and Input widgets.
* Sử dụng các Button hoặc Link widgets để điều hướng giữa các Màn hình hoặc để kích hoạt các hành động khác
* Hiển thị một trong hai khu vực trên Màn hình dựa trên điều kiện Boolean với widget If
* Cung cấp các tùy chọn đầu vào có hoặc không hoặc một lựa chọn trong số nhiều tùy chọn với các Checkbox hoặc Radio Button widgets
* Sử dụng các Container widgets để nhóm các tiện ích khác và áp dụng kiểu cho chúng
* Tạo các thành phần UI có thể tái sử dụng để sử dụng trong một ứng dụng hoặc trên các ứng dụng với Block or Web Block widgets.

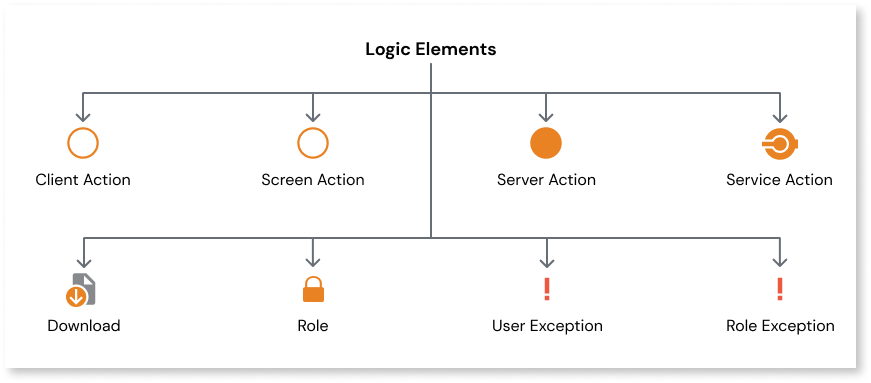
Dưới đây là một vài khối lệnh hay được sử dụng nhất trong Outsystems:

|  |  |  |
| --- | --- | --- |
| STT | Khối lệnh | Chức năng |
| 1 |  | Nơi có thể kéo và thả các tiện ích khác, bao gồm cả các container khác, để sắp xếp bố cục màn hình. |
| 2 |  | người dùng có thể bật hoặc tắt một tùy chọn. Giá trị phải được lưu trữ trong biến Boolean. |
| 3 |  | Người dùng có thể nhấp hoặc nhấn để kích hoạt một hành động, gửi dữ liệu hoặc điều hướng đến một màn hình khác.  Nếu nút thuộc về một biểu mẫu có một số trường nhập, nút sẽ gửi thông tin nếu bạn đặt thuộc tính Is Form Default thành Yes trong Button Widget. |
| 4 |  | Hiển thị dữ liệu trong database lên màn hình theo dạng bảng. |
| 5 |  | Hiển thị những giá trị có thể thay đổi được trên màn hình. |
| 6 |  | Hiển thị siêu liên kết mà người dùng có thể nhấp hoặc nhấn để kích hoạt hành động, gửi dữ liệu hoặc điều hướng đến màn hình khác. |

Bảng 2.3 Các khối lệnh Font-end

#### Back-end

Logic trong OutSystems có thể được xác định thông qua Action. Action là một thành phần logic cho phép ta xác định các luồng logic chạy trên máy chủ hoặc phía máy khách của ứng dụng. Ta có thể sử dụng các phần tử logic trong ứng dụng di động và web phản ứng của mình để xác định các thuật toán tùy chỉnh. Một số thành phần logic OutSystems bao gồm Screen action chứa logic dành riêng cho một màn hình, Client actions chạy logic trên thiết bị và Server action chạy logic trên máy chủ.



Hình 2.1 Các thành phần Logic Outsystems

Client actions: Sẽ chạy các logic phía người dùng.

Server action:

A screenshot of a computer

Description automatically generated

Hình 2.2 Server Action

Trong OutSystems, có thể tạo Server Actions để đóng gói logic thực hiện các quy tắc công việc của ứng dụng của mình và sử dụng chúng trong các hành động khác, chẳng hạn như Preparation actions, Screen Actions hoặc Server Actions khác :

Screen Actions:

A screenshot of a computer

Description automatically generated

Hình 2.3 Screen action

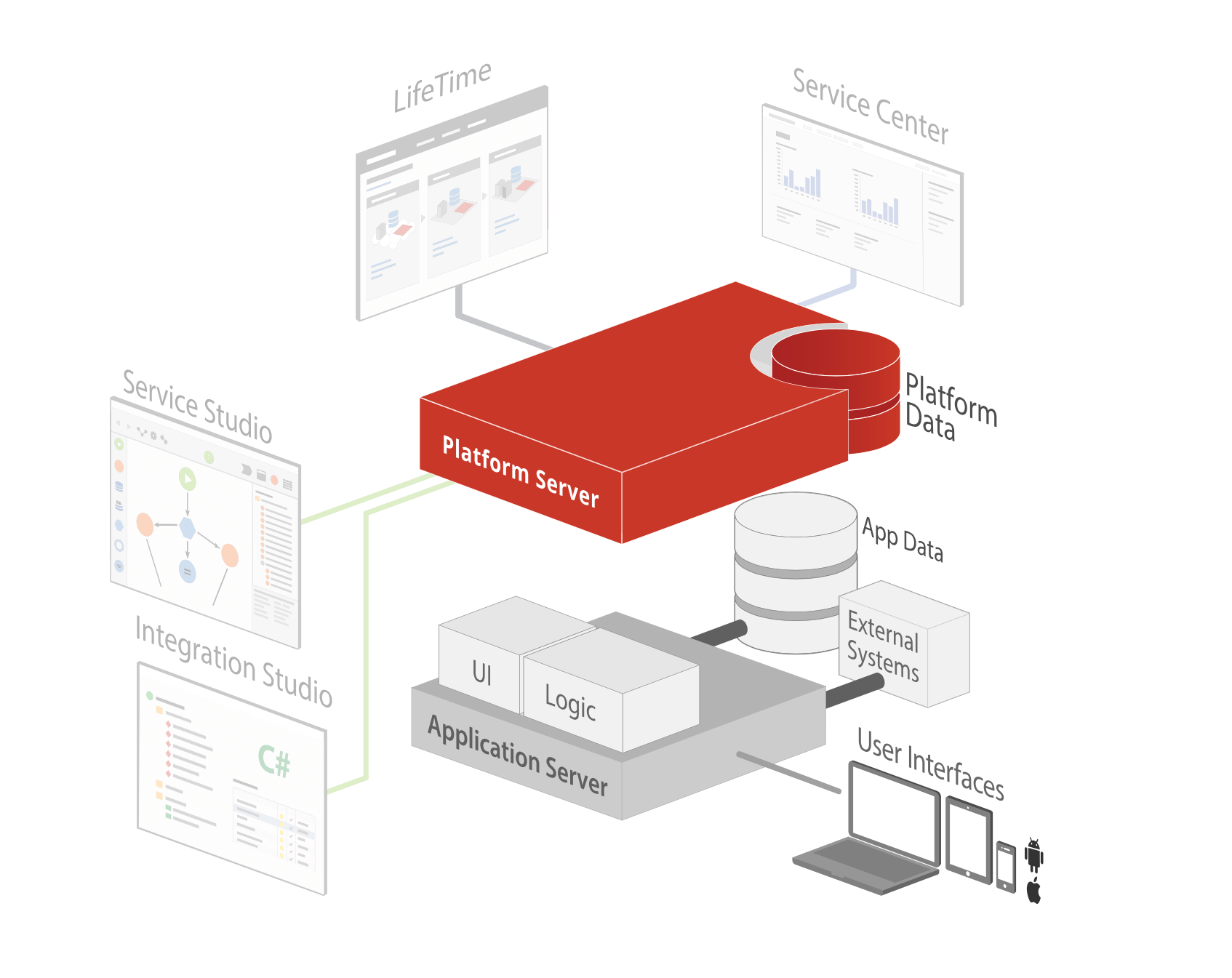
Screen Actions cho phép chạy logic khi người dùng tương tác với màn hình, chẳng hạn như nhấp vào nút hoặc liên kết.

Outsystems hỗ trợ rất nhiều Action giúp người phát triển đẩy nhanh quy tiến độ làm việc. Sau đây là một vài Action được sử dụng nhiều nhất:

|  |  |  |
| --- | --- | --- |
| STT | Action | Chức năng |
| 1 |  | * Cho phép bạn chạy, kiểm tra và xem lại các truy vấn SQL trong ứng dụng của mình. SQL là một công cụ đa năng dành cho các nhà phát triển biết ngôn ngữ SQL * SQL chỉ truy cập dữ liệu thông qua các tham số đầu vào và logic khác chỉ có thể truy cập những gì truy vấn SQL trả về thông qua Đầu ra của nó. |
| 2 |  | Dùng để lấy được dữ liệu từ các bảng trong data. Từ đó, phục vụ các thao tác xử lý phía sau. |
| 3 |  | Chạy các Action phía server hỗ trợ.  VD: ListFilter, ListSort,ListIndex,…. |
| 4 |  | Dùng để gán các giá trị cho nhau |
| 5 |  | Dùng để bắt đầu Action xử lý nghiệp vụ. |
| 6 |  | Dùng để kết thúc Action xử lý nghiệp vụ. |

Bảng 2.4 Các khối lệnh Back-end

### Trình dịch:

****

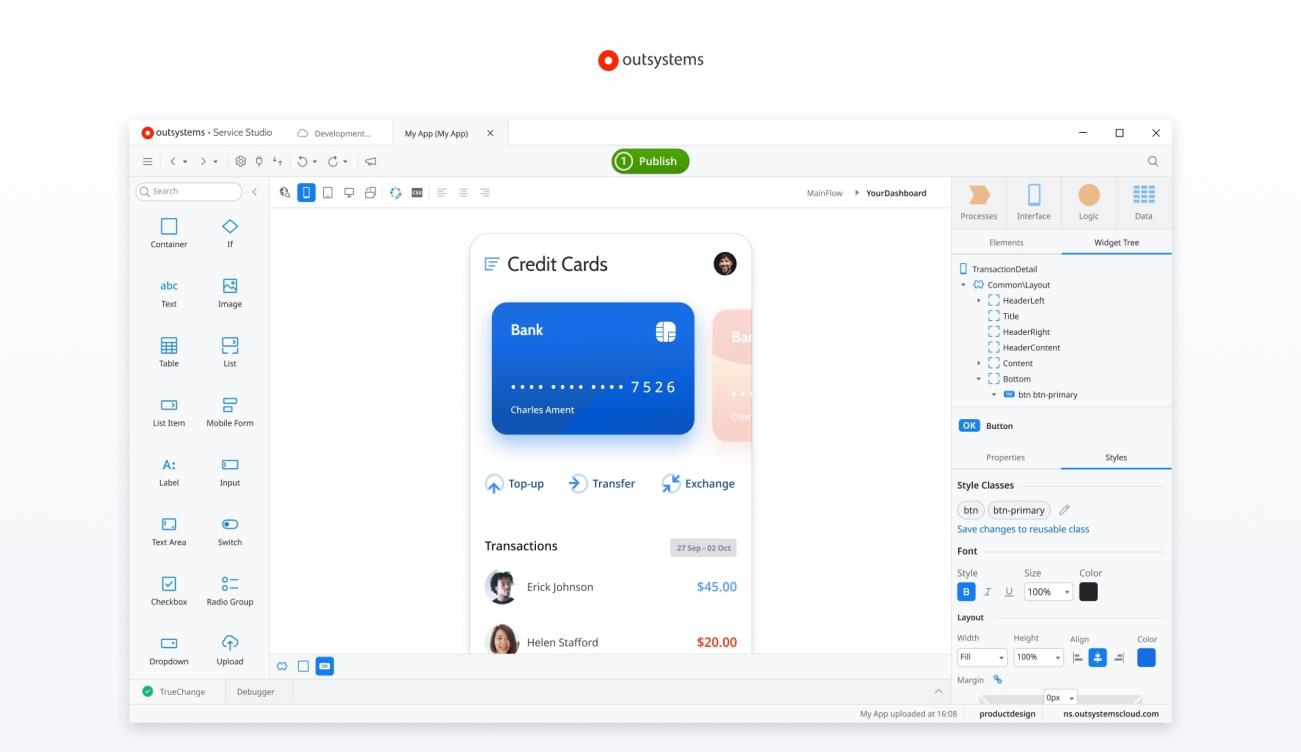
Hình 2.4 Ảnh minh họa Platform Server

* Platform Server: Đây là bộ máy chủ dùng để biên dịch, triển khai, quản lý, thi hành và điều khiển cơ sở hạ tầng bên trong ứng dụng.
* Service Studio:
  + Service Studio là môi trường phát triển ứng dụng ảo, cho phép nhà phát triển tạo ra các ứng dụng Mobile và Web.
  + Sử dụng Service Studio để kết nối với Platform Server. Sau khi chúng được kết nối với nhau, người lập trình có thể tạo và cài đặt ứng dụng trên Platform Server. Mỗi phiên bản của ứng dụng sẽ được lưu trữ trên Platform Data database. Sau đó, Platform Server sẽ biên dịch và tạo ra các dòng code được tối ưu cho ứng dụng và triển khai nó trên Application Server. Application Server sử dụng các databases truyền thống và hệ thống bên ngoài để khởi chạy ứng dụng mà người dùng tạo ra.
* Integration Studio: Cũng là môi trường phát triển như Service Studio, sử dụng để tạo ra các bản mở rộng cho chính nền tảng của nó. Nó cung cấp số lượng lớn các bộ xử lí để tích hợp với các tài nguyên bên ngoài như code C# hay databases. Integration Studio sẽ sử dụng các tài nguyên này và tạo ra các đại diện của nó bên trong OutSystems. Các đại diện tài nguyên này sẽ được gửi lên server và được sử dụng trên Service Studio như một tài nguyên OutSystems thông thường ngay sau khi nó được tạo và xuất hiện trong bản mở rộng.

## 2.4. Phát triển ứng dụng No-code/Low-code với OutSystem

### Công cụ phát triển ứng dụng Service Studio

Studio là một công cụ phát triển ứng dụng visual trong nền tảng OutSystems. Nó cho phép người dùng thiết kế, phát triển và triển khai các ứng dụng di động và web một cách nhanh chóng và dễ dàng. Service Studio cung cấp một giao diện người dùng thân thiện và trực quan, giúp người dùng dễ dàng tạo các thành phần của ứng dụng như giao diện người dùng, cơ sở dữ liệu, logic xử lý và tính năng kết nối với các hệ thống bên ngoài.



Hình 2.5 Môi trường làm việc Service Studio

Service Studio cũng có khả năng đồng bộ hóa các thành phần của ứng dụng giữa môi trường phát triển và môi trường triển khai, giúp đảm bảo tính nhất quán và độ tin cậy của ứng dụng. Nó cũng hỗ trợ tính năng kiểm tra lỗi tự động và quản lý phiên bản ứng dụng, giúp người dùng dễ dàng quản lý và duy trì ứng dụng của mình.

Với Service Studio, người dùng có thể tạo các ứng dụng phức tạp một cách nhanh chóng và dễ dàng mà không cần có nhiều kinh nghiệm về lập trình. Công cụ này giúp tiết kiệm thời gian và chi phí phát triển ứng dụng, giúp người dùng tập trung vào các nhu cầu kinh doanh của mình và nhanh chóng đưa ra các giải pháp cho khách hàng của họ.

#### Cách thức hoạt động

Service Studio là công cụ phát triển ứng dụng visual trong nền tảng OutSystems. Nó hoạt động bằng cách cho phép người dùng thiết kế và phát triển các thành phần của ứng dụng thông qua giao diện người dùng trực quan.

Để bắt đầu tạo một ứng dụng trong Service Studio, người dùng cần tạo một dự án mới và chọn loại ứng dụng (web hoặc di động). Sau đó, người dùng sẽ được chuyển đến một môi trường phát triển tích hợp, nơi họ có thể thiết kế giao diện người dùng, cài đặt cơ sở dữ liệu và xây dựng logic xử lý.

Service Studio hỗ trợ nhiều thành phần khác nhau của ứng dụng, bao gồm các mô-đun, các biểu mẫu và các quy trình. Người dùng có thể kéo và thả các thành phần này vào khung làm việc để tạo các trang web hoặc màn hình di động của ứng dụng.

Ngoài ra, Service Studio cũng hỗ trợ tính năng kết nối với các hệ thống bên ngoài, giúp người dùng dễ dàng tích hợp ứng dụng của mình với các dịch vụ bên thứ ba.

Sau khi tạo xong ứng dụng, người dùng có thể đóng gói và triển khai ứng dụng lên các máy chủ hoặc nền tảng đám mây để sử dụng. Service Studio cũng hỗ trợ việc kiểm tra lỗi tự động và quản lý phiên bản ứng dụng, giúp người dùng dễ dàng quản lý và duy trì ứng dụng của mình.

#### Ưu điểm

Dưới đây là một số ưu điểm của việc sử dụng Service Studio trong việc phát triển ứng dụng trong nền tảng OutSystems:

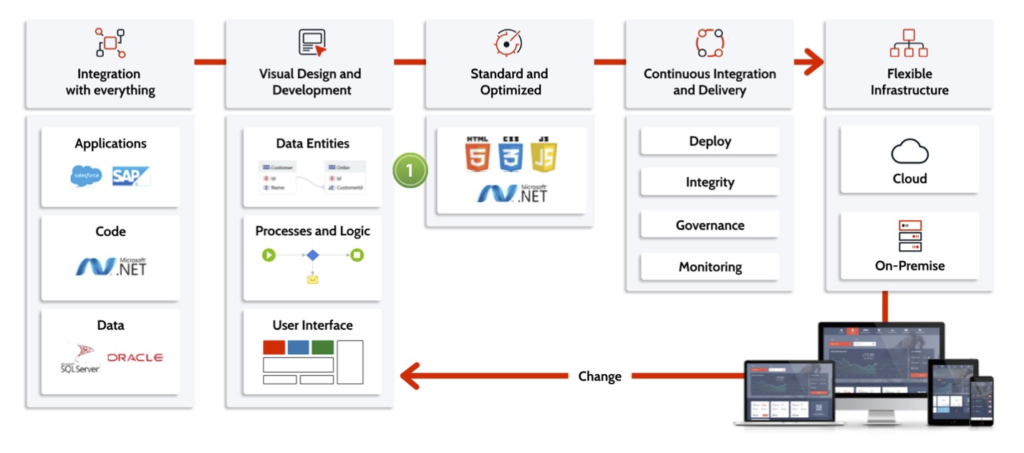
* Phát triển nhanh chóng và tiết kiệm thời gian: Service Studio là một công cụ phát triển visual, cho phép người dùng thiết kế và phát triển ứng dụng nhanh chóng mà không cần có nhiều kinh nghiệm về lập trình. Nó cung cấp các thành phần ứng dụng sẵn có, cho phép người dùng tập trung vào các yêu cầu kinh doanh của họ mà không mất quá nhiều thời gian trong quá trình phát triển.
* Dễ sử dụng và trực quan: Service Studio có giao diện người dùng trực quan và thân thiện, giúp người dùng dễ dàng tạo các thành phần của ứng dụng. Người dùng có thể kéo và thả các thành phần để tạo các trang web hoặc màn hình di động của ứng dụng một cách dễ dàng.
* Tích hợp dễ dàng với các hệ thống bên ngoài: Service Studio cung cấp tính năng kết nối với các hệ thống bên ngoài một cách dễ dàng, giúp người dùng tích hợp các ứng dụng của mình với các dịch vụ bên thứ ba một cách nhanh chóng và tiện lợi.
* Quản lý phiên bản và kiểm tra lỗi tự động: Service Studio hỗ trợ tính năng quản lý phiên bản ứng dụng, giúp người dùng dễ dàng quản lý và duy trì ứng dụng của mình. Nó cũng cung cấp tính năng kiểm tra lỗi tự động, giúp giảm thiểu số lượng lỗi và tăng tính tin cậy của ứng dụng.
* Hỗ trợ cho cả ứng dụng di động và web: Service Studio hỗ trợ phát triển cả ứng dụng di động và web, giúp người dùng tạo ra các ứng dụng đa nền tảng một cách dễ dàng.

Tóm lại, Service Studio là một công cụ phát triển ứng dụng visual mạnh mẽ và tiện lợi trong nền tảng OutSystems, giúp người dùng tạo ra các ứng dụng chất lượng cao một cách nhanh chóng và dễ dàng.

#### Tính năng của Service Studio

Dưới đây là một số tính năng chính của Service Studio:

* Kéo và thả: Người dùng có thể kéo và thả các thành phần và logic để xây dựng giao diện người dùng, các luồng dữ liệu và các quy tắc kinh doanh trong ứng dụng.
* WYSIWYG(What You See Is What You Get): Service Studio cung cấp một trình chỉnh sửa WYSIWYG, cho phép người dùng thiết kế giao diện người dùng một cách trực quan và thấy kết quả ngay lập tức
* Giao diện trực quan: Service Studio cung cấp một giao diện dễ sử dụng và trực quan, giúp người dùng tạo, chỉnh sửa và kiểm tra các thành phần của ứng dụng một cách dễ dàng.
* Kiểm tra và gỡ lỗi: Service Studio hỗ trợ kiểm tra và gỡ lỗi ứng dụng. Người dùng có thể xem trước và kiểm tra các tính năng của ứng dụng trong quá trình phát triển để đảm bảo rằng nó hoạt động đúng như mong đợi.
* Tích hợp dữ liệu: Service Studio cho phép người dùng kết nối với các nguồn dữ liệu bên ngoài như cơ sở dữ liệu, dịch vụ web và API để lấy và ghi dữ liệu.
* Quản lý phiên bản: Service Studio hỗ trợ quản lý phiên bản của ứng dụng. Người dùng có thể lưu trữ và quản lý các phiên bản khác nhau của ứng dụng và quay lại các phiên bản trước đó nếu cần thiết.
* Entities và Data: Service Studio cho phép người dùng định nghĩa các thực thể (entities) để lưu trữ và quản lý dữ liệu. Người dùng có thể tạo các trường (attributes) trong entities để đại diện cho các thuộc tính của đối tượng.
* Screens và Web Blocks: Service Studio cho phép người dùng tạo các màn hình (screens) để hiển thị giao diện người dùng. Mỗi màn hình có thể chứa các thành phần như nút, ô nhập liệu và bảng. Người dùng cũng có thể tạo các web blocks, là các thành phần giao diện có thể được tái sử dụng trên nhiều màn hình.
* Actions và Logic: Người dùng có thể tạo các actions để thực hiện các tác vụ cụ thể trong ứng dụng, chẳng hạn như lưu trữ dữ liệu, xử lý logic kinh doanh, hoặc gọi các dịch vụ bên ngoài. Actions có thể được gọi từcác sự kiện trên giao diện người dùng hoặc từ logic khác trong ứng dụng.
* Integration: Service Studio hỗ trợ tích hợp với các dịch vụ bên ngoài, bao gồm cơ sở dữ liệu, dịch vụ web và API. Người dùng có thể tạo kết nối với các nguồn dữ liệu này và sử dụng chúng trong logic và giao diện của ứng dụng.
* Deployment: Khi ứng dụng đã được xây dựng và kiểm tra hoàn chỉnh trong Service Studio, người dùng có thể triển khai ứng dụng lên môi trường sản xuất thông qua quy trình triển khai được cung cấp bởi OutSystems.



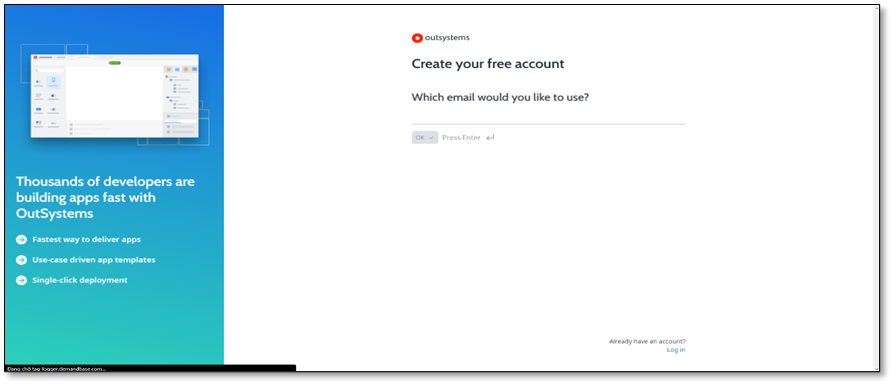
Hình 2.6 Tính năng của Outsystems

Service Studio giúp đơn giản hóa việc phát triển ứng dụng và giảm thiểu việc viết mã lập trình truyền thống. Nó cung cấp một giao diện trực quan và mạnh mẽ cho người dùng để xây dựng, kiểm tra và triển khai các ứng dụng OutSystems một cách dễ dàng và nhanh chóng.

### Quy trình cài đặt và phát triển ứng dụng bằng Service Studio

#### Quy trình cài đặt:

* Bước 1: Tạo tài khoản outsystems và active tài khoản để sử dụng:
  + Truy cập trang đăng ký tài khoản của outsystems theo đường link <https://www.outsystems.com/Platform/Signup> và nhập các thông tin cần thiết để tiến hành tạo tài khoản.



Hình 2.7 Trang đăng ký tài khoản

* + Một email active được gửi đến mail bạn vừa đăng ký . Chọn **Activate My Account** để active tài khoản bạn vừa đăng ký và sẽ được chuyển hướng đến trang login nếu bạn chưa login tài khoản này trên trang web hoặc đến trang My Plaltform nếu bạn đã login tài khoản

A screenshot of a computer

Description automatically generated

Hình 2.8 Active tài khoản OS

* Cài đặt tool Outsystems:
  + Tại trang My Plaltform => scroll xuống cuối trang tại mục tool => click chọn download của Service Studio.

A screenshot of a computer

Description automatically generated

Hình 2.9 Tải Service Studio

* + Vào thư mục vừa tải xuống chọn file cài đặt của outsystems như hình bên dưới và cho phép các quyền mặc định được yêu cầu trong quá trình cài đặt

A screenshot of a computer

Description automatically generated

Hình 2.10 Cài đặt Service Studio

* + Mở tool Outsystem vừa cài đặt lên, nhập username và password đã đăng ký ở trên để đăng nhập vào tool.

A screenshot of a computer

Description automatically generated

Hình 2.11 Đăng nhập tài khoản OS

* Sau khi đăng nhập thành công sẽ hiển thị màn hình các app (như hình bên dưới)

A screenshot of a computer

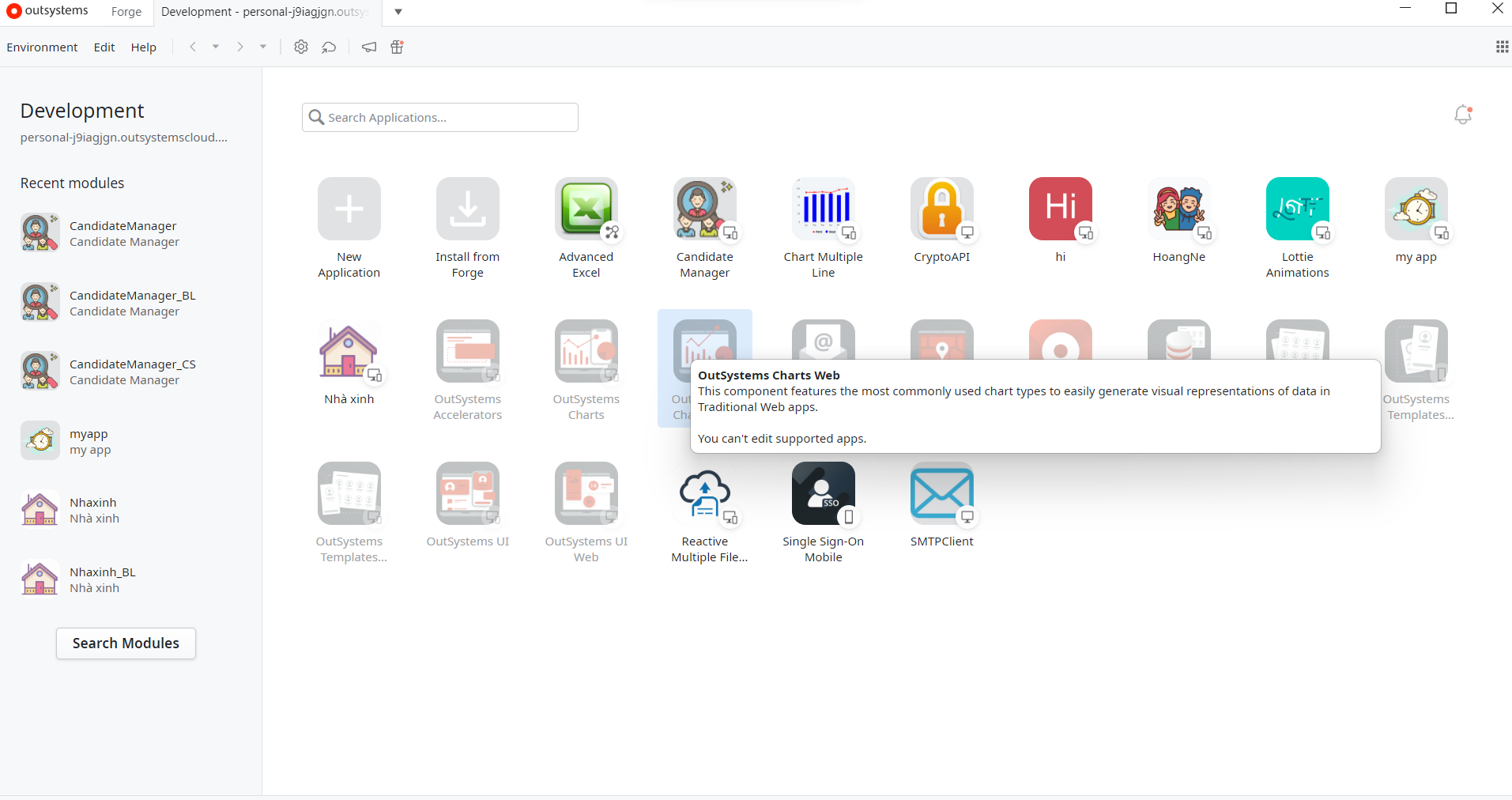
Description automatically generated

Hình 2.12 Màn hình menu app

Trên đây là quy trình cài đặt Service Studio.

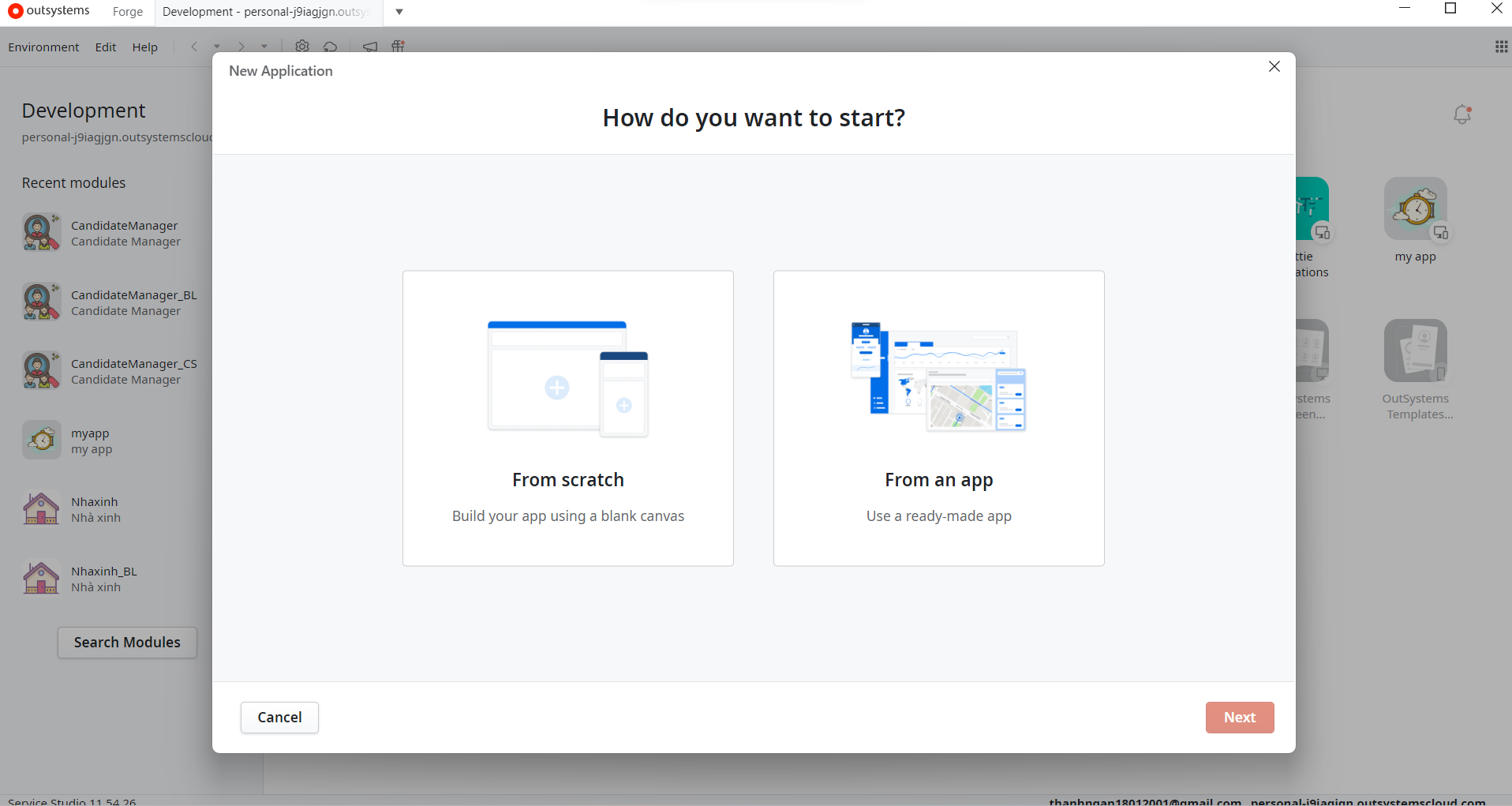
#### Quy trình tạo sản phẩm:

* Bước 1: Click New Application.



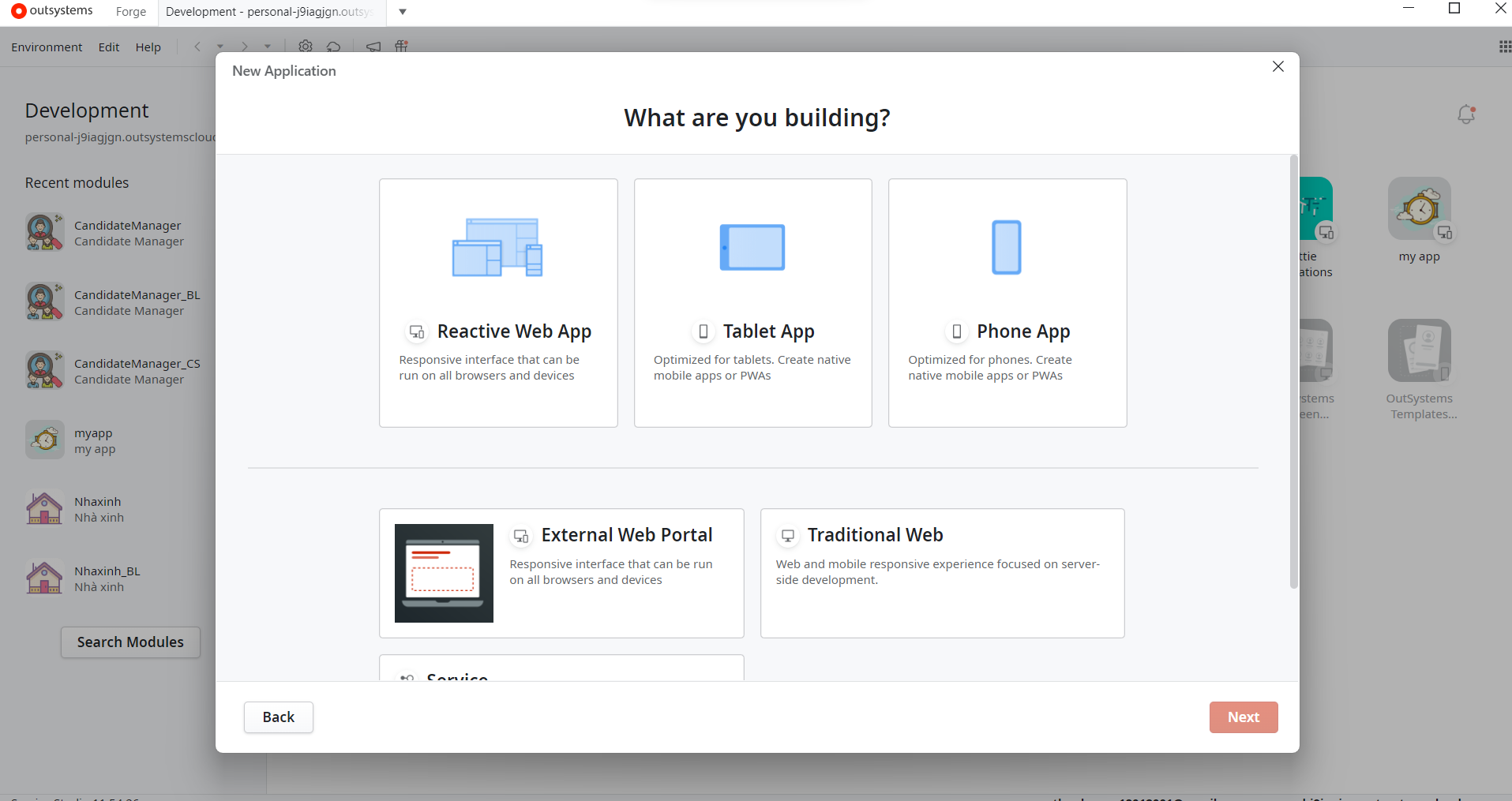
Hình 2.13 Màn hình click tạo mới

* Bước 2: Chọn From scratch để chọn form cho sản phẩm.



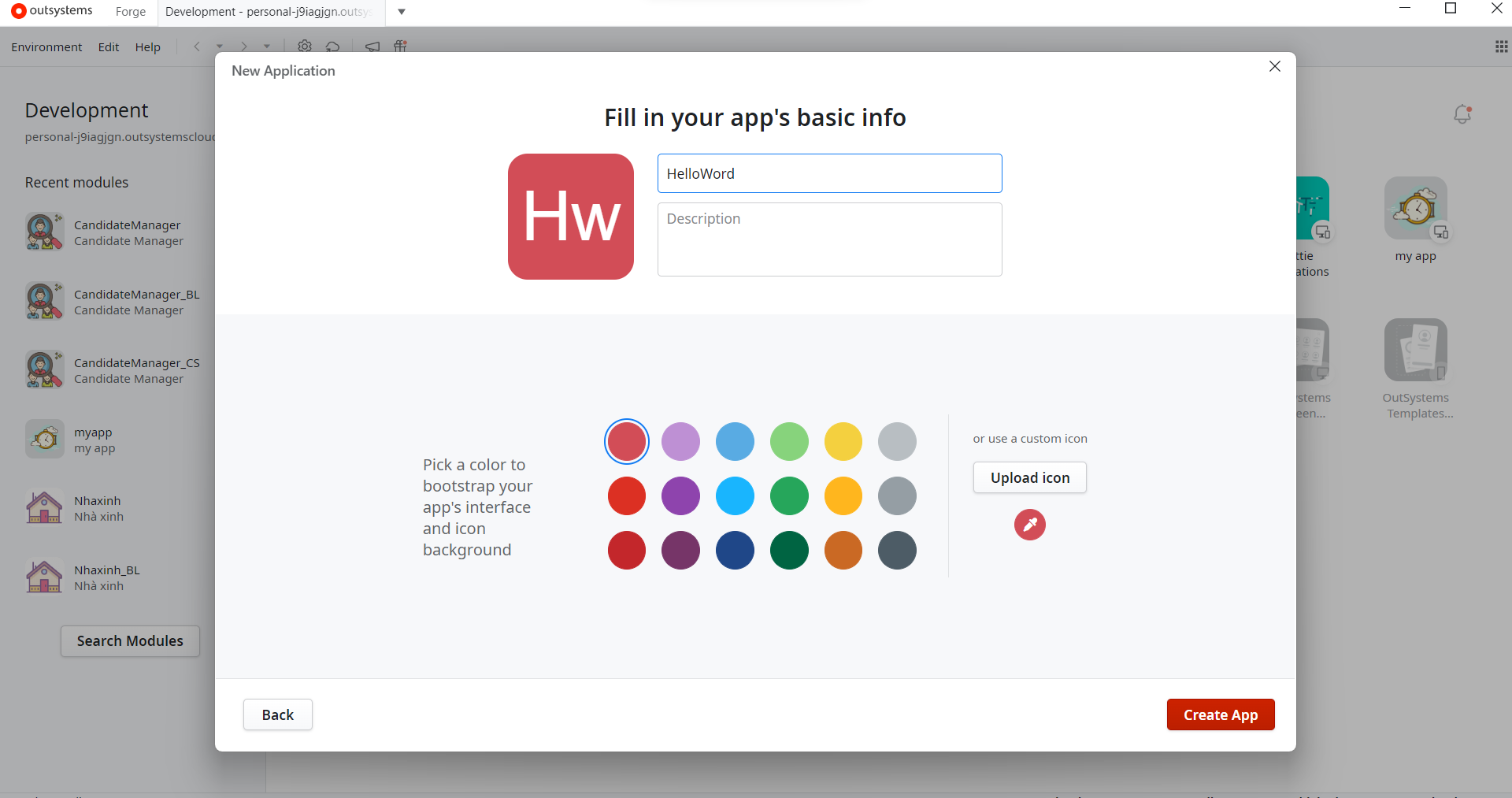
Hình 2.14 From scratch

* Bước 3: c dụng muốn xây dựng, gồm có:
* Traditional web
* Reactive web app
* Table app
* Phone app



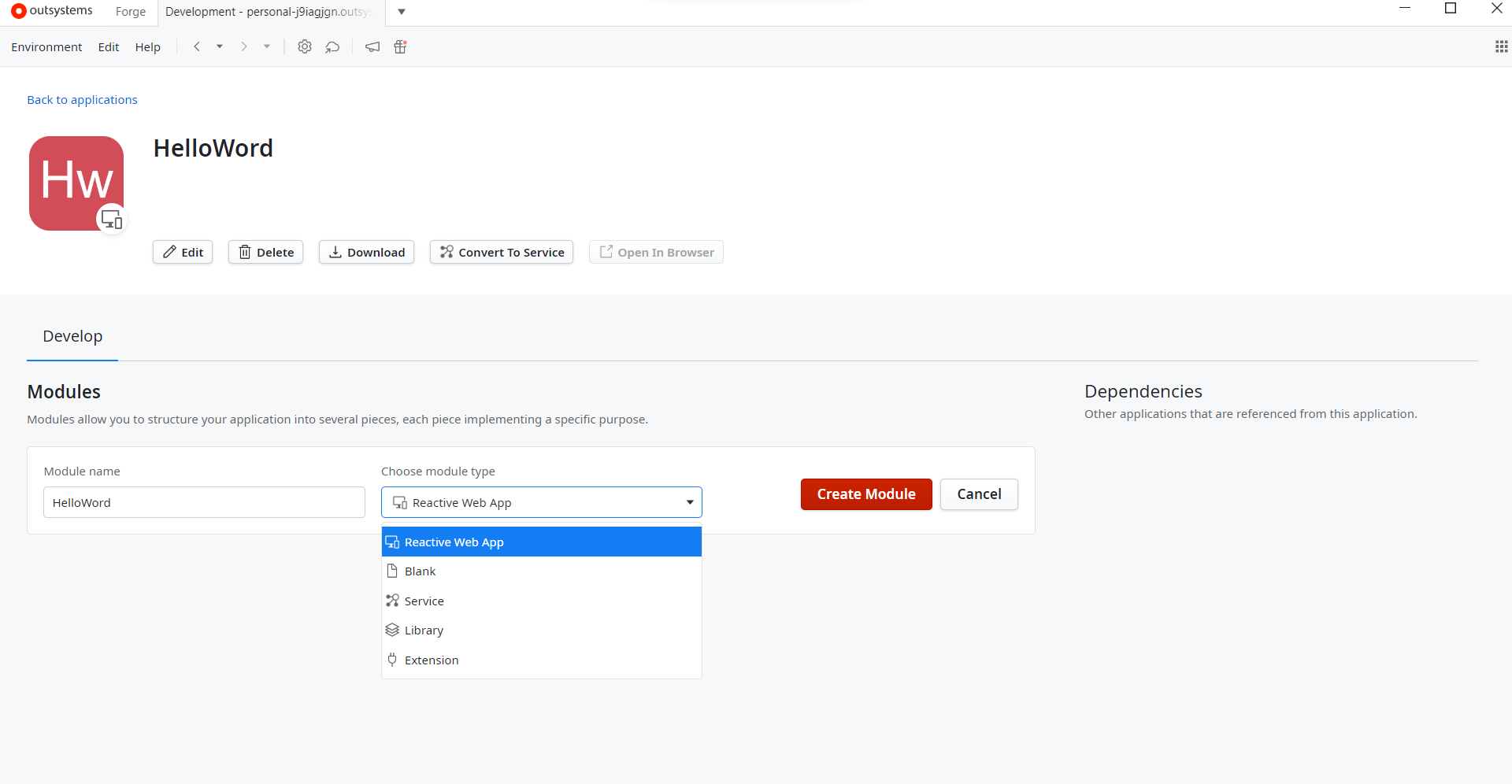
Hình 2.15 Chọn kiểu ứng dụng

* Bước 4: Đặt tên cho ứng dụng và click “Create App”



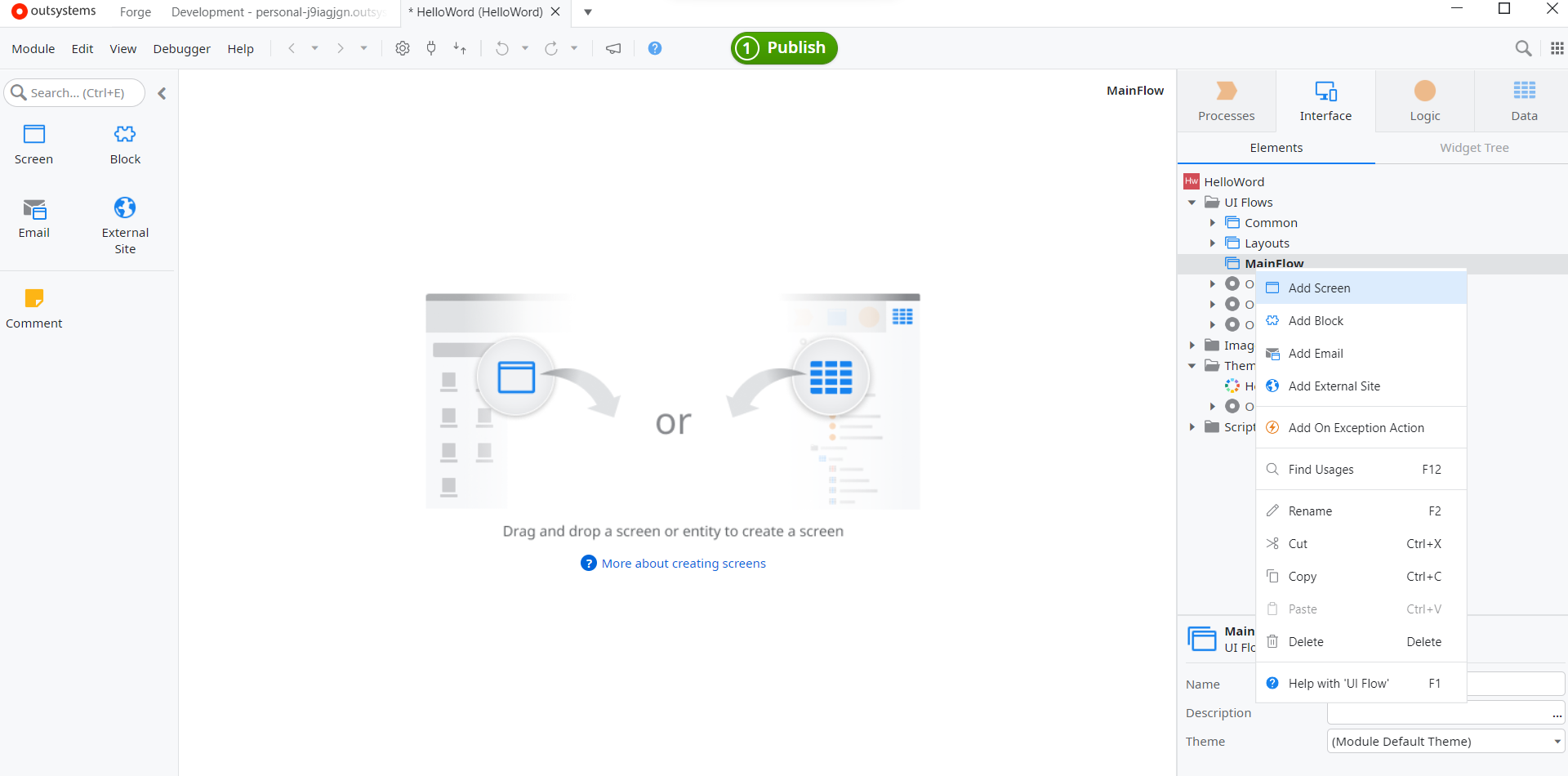
Hình 2.16 Tạo mới app

* Bước 5: Tạo Module cho sản phẩm, bao gồm các dạng module:
* Reactive web app: bao gồm UI, logic, data
* Blank: module rỗng
* Service: bao gồm logic và data
* Library: bao gồm data
* Extention: kết nối và mở Integration Studio để có thể tích hợp code thuần bằng C#.



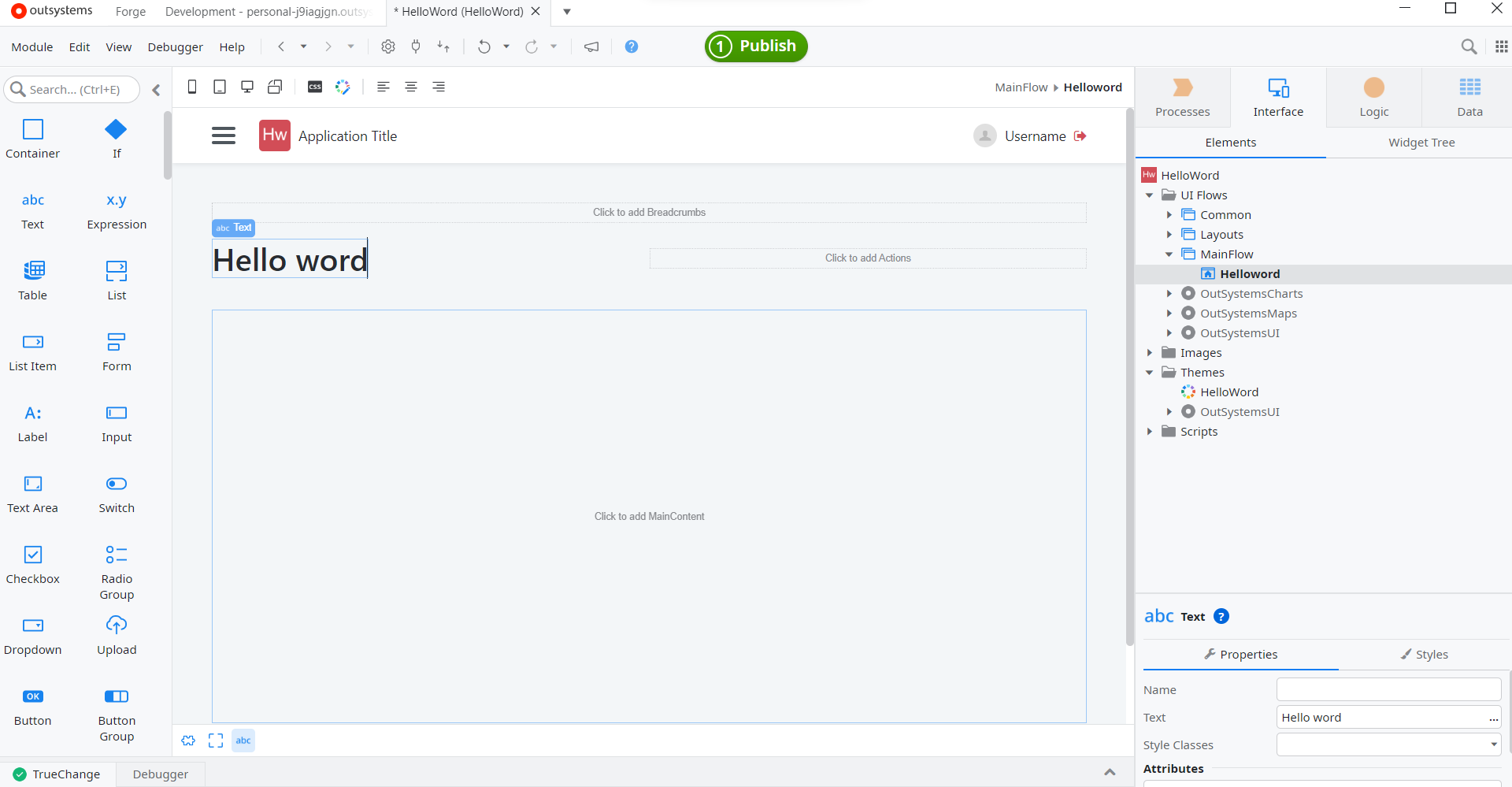
Hình 2.17 Tạo module

* Bước 6: Tạo mới 1 Screen trong module vừa tạo.



Hình 2.18 Tạo Screen

* Bước 7: Sau khi tạo mới xong, thực hiện thiết kế và xử lý sản phẩm rồi click “Publish” để có thể lưu và chạy sản phẩm.



Hình 2.19 Publish code

# PHÁT TRIỂN ỨNG DỤNG VỚI OUTSYSTEM

**3.1 Phân tích thiết kế hệ thống**

### Mô tả bài toán

“Phát triển ứng dụng web hỗ trợ tuyển dụng sử dụng Outsystems” là ứng dụng cho phép người quản lý có thể quản lý ứng viên một cách tối ưu nhất.

Với ứng dụng web tuyển dụng, người ứng tuyển (Candidate) có thể gửi CV kèm các thông tin cá nhân cho bên quản lý nhân sự của công ty ứng tuyển. Ngoài ra, nếu CV được chấp nhận, người ứng tuyển còn có thể đăng nhập và làm bài test đầu vào công ty.

Với người quản lý (HR), sau khi truy cập vào ứng dụng, có thể thêm, sửa, xóa các thông tin liên quan đến bài test và các lớp học. Ngoài ra, người quản lý còn có thể quản lý ứng viên bằng việc xét duyệt các ứng viên ứng tuyển vào công ty thông qua hệ thống.

Với người hướng dẫn (Trainer), sau khi truy cập vào hệ thống, có thể thêm, sửa, xóa các thông tin liên quan đến bài test và tham gia vào quá trình xét duyệt ứng viên cùng người quản lý trong một số trạng thái( liên quan đến bài test)

Để giải quyết bài toán này, nền tảng sẽ được sử dụng là No-code/Low-code với công nghệ Outsystems.

### Phân tích biểu đồ usecase

#### Xác định các tác nhân

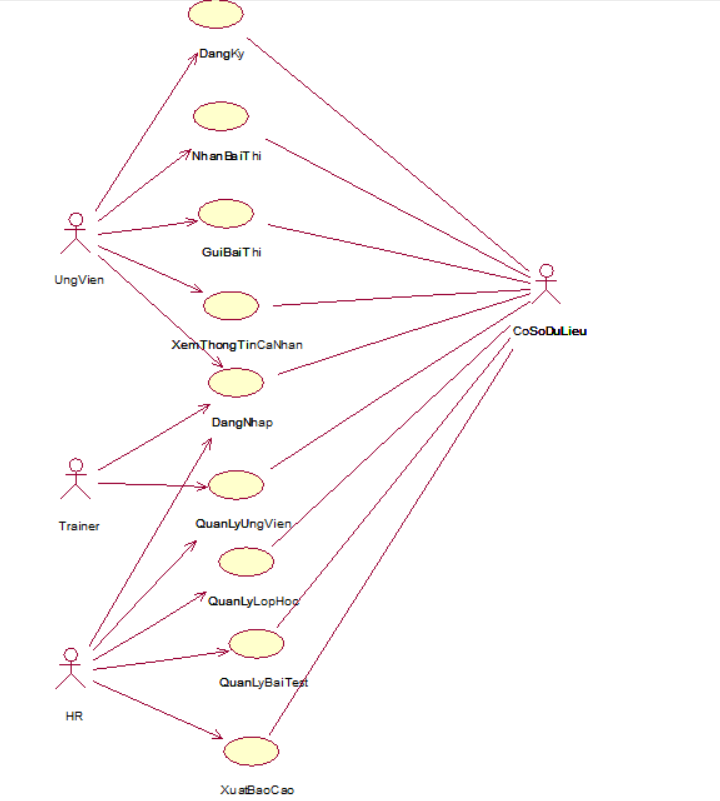
Các tác nhân của hệ thống:

- Ứng viên: Đăng ký, đăng nhập, nhận bài thi, gửi bài thi, xem thông tin cá nhân,…

- HR: Quản lý ứng viên, xuất báo cáo, quản lý bài test, quản lý lớp học,…

- Trainer: Quản lý ứng viên, quản lý bài test,…

#### Biểu đồ usecase tổng quát



Hình 3.1 Biểu đồ usecase tổng quát

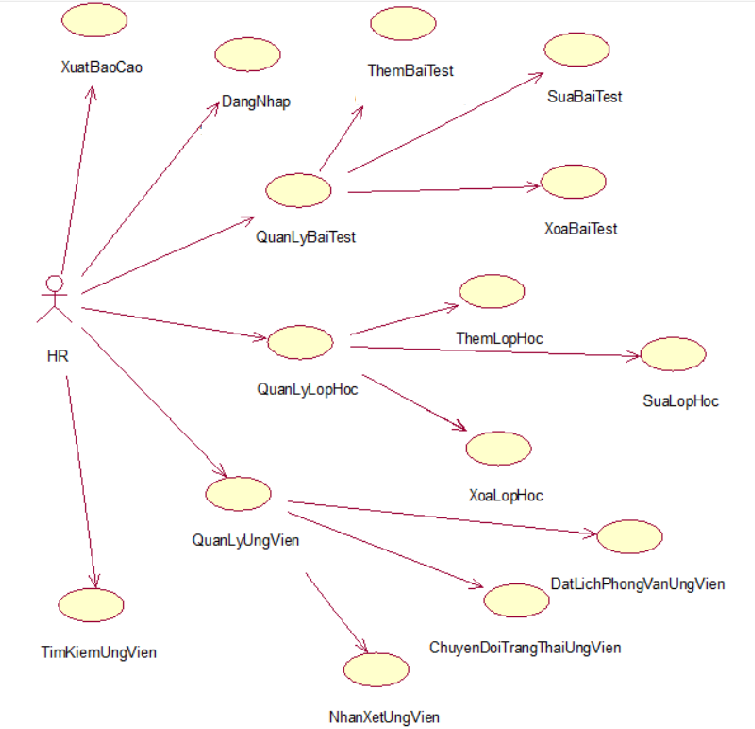
#### Biểu đồ usecase ứng viên

A diagram of a person with text

Description automatically generated

Hình 3.2 Usecase ứng viên

#### Biểu đồ usecase HR



Hình 3.3 Usecase HR

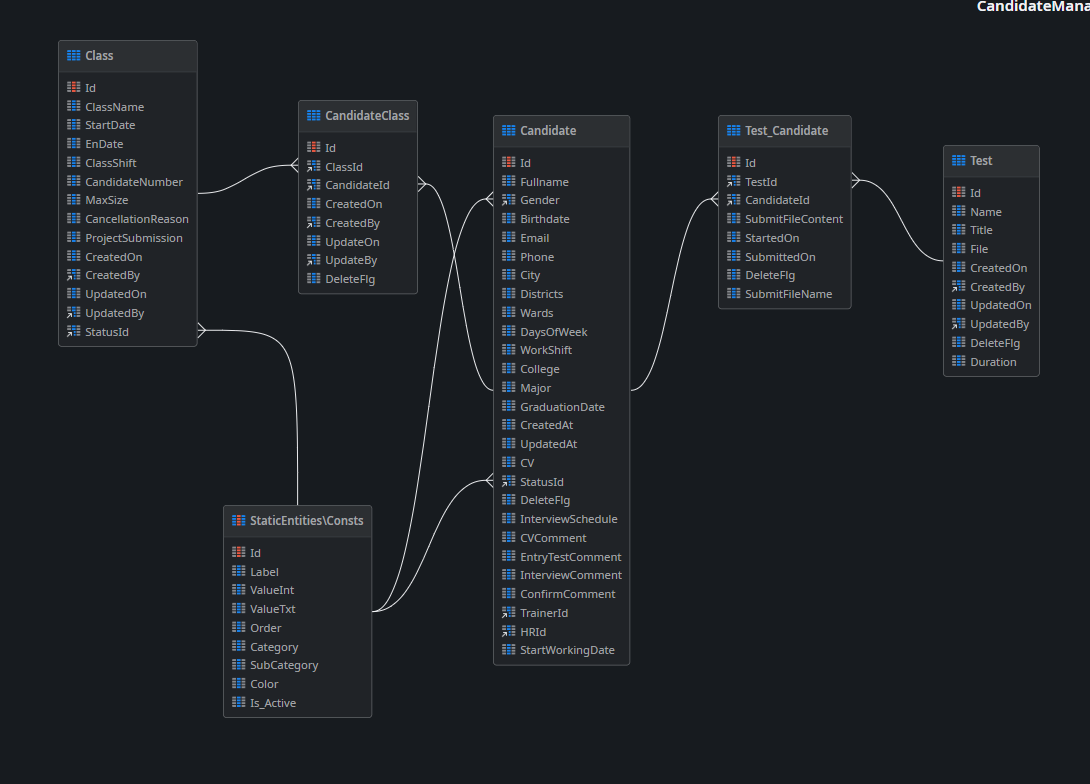
1. Biểu đồ usecase trainer

A diagram of a person with a person's figure

Description automatically generated

Hình 3.4 Usecase Trainer

### Biểu đồ thực thể liên kết



Hình 3.5 Biểu đồ thực thể liên kết

### Đặc tả use case

#### 3.1.3.1 Đặc tả usecase Đăng Nhập

|  |  |
| --- | --- |
| **ID** | UC01 |
| **Tên UseCase** | Đăng nhập |
| **Mô tả** | Use case cho phép ứng viên đăng nhập vào hệ thống. |
| **Actor** | Ứng viên |
| **Tiền điều kiện** | Ứng viên nhận được email có chứa link màn hình đăng nhập của công ty. |
| **Hậu điều kiện** | Đăng nhập thành công. |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi ứng viên muốn đăng nhập vào hệ thống. |
| 1. Ứng viên click vào link đăng nhập trong email. |
| 1. Ứng viên nhập thông tin bắt buộc rồi chọn Send OTP. |
| 1. Hệ thống kiểm tra thông tin đăng nhập của ứng viên, chuyển hướng ứng viên tới màn hình nhập OTP. |
|  | 1. Ứng viên nhập OTP rồi click vào button Login. |
|  | 1. Hệ thống kiểm tra thông tin ứng viên nhập và chuyển hướng ứng viên tới màn hình Trang chủ của ứng viên. |
| **Luồng rẽ nhánh** | 1. Tại bước 3 của luồng cơ bản, nếu ứng viên không nhập đầy đủ thông tin bắt buộc. Hiển thị lỗi không cho gửi mã OTP, yêu cầu ứng viên nhập đầy đủ thông tin cần thiết. 2. Nếu email không tồn tại trong cơ sở dữ liệu, hệ thống hiển thị thông báo lỗi. |
| 1. Tại bước 5 của luồng cơ bản, nếu ứng viên không nhập đầy đủ thông tin bắt buộc. Hiển thị lỗi không cho đăng nhập, yêu cầu ứng viên nhập đầy đủ thông tin cần thiết. 2. Nếu OTP không đúng, hệ thống sẽ hiển thị thông báo lỗi. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.1 Đặc tả usecase đăng nhập

#### 3.1.3.2 Đặc tả usecase Đăng Ký

|  |  |
| --- | --- |
| **ID** | UC02 |
| **Tên UseCase** | Đăng ký |
| **Mô tả** | Use case cho phép ứng viên đăng ký ứng tuyển trên hệ thống của công ty. |
| **Actor** | Ứng viên |
| **Tiền điều kiện** | Ứng viên có được link đăng ký ứng tuyển từ nhà tuyển dụng. |
| **Hậu điều kiện** | Đăng ký ứng tuyển thành công. |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi ứng viên muốn đăng ký ứng tuyển vào công ty. |
| 1. Hệ thống hiện thị trang đăng ký tài khoản. |
| 1. Ứng viên nhập đầy đủ thông tin bắt buộc rồi chọn Apply for Intership Program. |
| 1. Hệ thống lưu thông tin ứng viên vào bảng Candidate và trả về thống báo. |
| **Luồng rẽ nhánh** | 1. Tại bước 3 của luồng cơ bản nếu không nhập đầy đủ thông tin hệ thống hiển thị thông báo lỗi: “Required field!”. 2. Tại bước 3 của luồng cơ bản, nếu nhập email đã có trong cơ sở dữ liệu. Hệ thống thông báo email đã tồn tại. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.2 Đặc tả usecase đăng ký

#### 3.1.3.3 Đặc tả usecase Nhận bài thi

|  |  |
| --- | --- |
| **ID** | UC03 |
| **Tên UseCase** | Nhận bài thi |
| **Mô tả** | Use case cho phép ứng viên nhận bài thi từ công ty |
| **Actor** | Ứng viên |
| **Tiền điều kiện** | Ứng viên đang ở trang chủ |
| **Hậu điều kiện** | Tải thành công bài thi của công ty dành cho ứng viên |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi ứng viên muốn nhận được bài thi của công ty. |
| 1. Ứng viên nhấn chuột chọn Get entry test. |
| 1. Hệ thống lấy bài thi và download bài thi về máy của ứng viên và hiển thị lên màn hình thông tin về hạn nộp bài thi của ứng viên đó. |
| **Luồng rẽ nhánh** | 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.3 Đặc tả use case nhận bài thi

#### 3.1.3.4 Đặc tả usecase Nộp bài thi

|  |  |
| --- | --- |
| **ID** | UC04 |
| **Tên UseCase** | Nộp bài thi |
| **Mô tả** | Use case cho phép ứng viên nộp bài làm của mình |
| **Actor** | Ứng viên |
| **Tiền điều kiện** | Ứng viên đang ở trang chủ |
| **Hậu điều kiện** | Nộp thành công bài làm của ứng viên |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi ứng viên muốn nộp bài thi. |
| 1. Ứng viên click chuột vào “Send test” trên màn hình để gửi bài thi. |
| 1. Hệ thống sẽ kiểm tra hạn làm bài của ứng viên sau đó hiển thị popup để nộp bài thi. |
|  | 1. Ứng viên upload file bài làm lên và click vào button “Send test file” để gửi bài làm. |
|  | 1. Sau khi ứng viên nộp bài thành công, sẽ hiển thị thông báo và chuyển hướng ứng viên về màn hình trang chủ |
| **Luồng rẽ nhánh** | 1. Tại bước 3 của luồng cơ bản, nếu thời gian hiện tại lớn hơn hạn nộp bài của ứng viên, ứng viên sẽ không thể click để hiển thị popup nộp bài. |
| 1. Tại bước 4 của luồng cơ bản, nếu ứng viên không nhập đầy đủ thông tin bắt buộc. Hiển thị lỗi không cho nộp bài, yêu cầu ứng viên nhập đầy đủ thông tin cần thiết. 2. Nếu file bài làm không đúng định dạng, hệ thống sẽ hiển thị thông báo lỗi. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.4 Đặc tả usecase nộp bài thi

#### 3.1.3.5 Đặc tả usecase Xem thông tin cá nhân

|  |  |
| --- | --- |
| **ID** | UC05 |
| **Tên UseCase** | Xem thông tin chi tiết cá nhân. |
| **Mô tả** | Use case cho phép ứng viên xem thông tin chi tiết của cá nhân. |
| **Actor** | Ứng viên |
| **Tiền điều kiện** | Ứng viên đang ở màn hình trang chủ. |
| **Hậu điều kiện** | Hiển thị thông tin chi tiết của ứng viên lên màn hình. |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi ứng viên muốn xem chi tiết thông tin của mình. |
| 1. Ứng viên click chuột vào Information trên màn hình. |
| 1. Hệ thống lấy thông tin chi tiết của ứng viên gồm tên ứng viên, giới tính, ngày sinh, email, số điện thoại, địa chỉ, tên trường, tên ngành học và ngày tốt nghiệp từ bảng Candidate để hiển thị lên màn hình. |
| **Luồng rẽ nhánh** | 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.5 Đặc tả usecase xem thông tin cá nhân

#### 3.1.3.6 Đặc tả usecase Tìm kiếm ứng viên

|  |  |
| --- | --- |
| **ID** | UC05 |
| **Tên UseCase** | Tìm kiếm ứng viên |
| **Mô tả** | Use case cho phép Hr, Trainer tìm kiếm ứng viên theo từ khóa, trạng thái và ngày phỏng vấn. |
| **Actor** | Hr, Trainer |
| **Tiền điều kiện** | HR, Trainer đang ở trong màn hình trang chủ |
| **Hậu điều kiện** | Hiển thị danh sách ứng viên phù hợp. |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi HR, Trainer muốn tìm kiếm ứng viên theo từ khóa.    1. HR, Trainer gõ từ khóa tìm kiếm vào ô input tìm kiếm.    2. Hệ thống hiện thị danh sách ứng viên có tên gần đúng với từ khóa tìm kiếm |
| 1. Ca sử dụng bắt đầu khi HR, Trainer muốn tìm kiếm ứng viên theo trạng thái.    1. HR, Trainer chọn status muốn tìm kiếm từ dropdown tìm kiếm trạng thái.    2. Hệ thống hiện thị danh sách ứng viên có trạng thái giống với trạng thái đã chọn bên trên. |
| 1. Ca sử dụng bắt đầu khi HR, Trainer muốn tìm kiếm ứng viên theo thời gian phỏng vấn.    1. HR, Trainer chọn khoảng thời gian phỏng vấn ở ô nput tìm kiếm.    2. Hệ thống hiện thị danh sách ứng viên có thời gian phỏng vấn nằm trong khoảng thời gian tìm kiếm bên trên. |
| **Luồng rẽ nhánh** | 1. Nếu không có kết quả nào được trả về, hệ thống hiển thị trang danh sách trống. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.6 Đặc tả usecase tìm kiếm ứng viên

#### 3.1.3.10 Đặc tả usecase Bảo trì ứng viên

|  |  |
| --- | --- |
| **ID** | UC10 |
| **Tên UseCase** | Bảo trì ứng viên |
| **Mô tả** | Usecase cho phép HR, Trainer có thể chuyển đổi trạng thái của ứng viên, đặt lịch phỏng vấn cho ứng viên, nhận xét ứng viên khi cần. |
| **Actor** | HR, Trainer |
| **Tiền điều kiện** | Đăng nhập với quyền HR, Trainer |
| **Hậu điều kiện** | Không có |
| **Luồng cơ bản** | * + - 1. Ca sử dụng bắt đầu khi HR, Trainer muốn chuyển đổi trạng thái của ứng viên.   1. HR, Trainer click vào tên ứng viên.   2. Hệ thống sẽ hiển thị thông các thông tin cơ bản của ứng viên theo từng trạng thái.   3. HR, Trainer click vào button hiển thị trên màn hình tùy theo status hiện có của ứng viên.   4. Hệ thống sẽ hiển thị popup để HR, Trainer có thể chuyển đổi trạng thái cho ứng viên và click button Submit.   5. Hệ thống sẽ lưu thông tin thay đổi của ứng viên vào bảng Candidate. Usecase kết thúc |
| 1. Ca sử dụng bắt đầu khi HR muốn đặt lịch phỏng vấn cho ứng viên.    1. HR click vào tên ứng viên.    2. Hệ thống sẽ hiển thị thông các thông tin cơ bản của ứng viên theo từng trạng thái    3. HR click vào button “Set interview schedule” hiển thị trên màn hình và.    4. Hệ thống sẽ hiển thị popup để HR có thể đặt lịch phỏng vấn cho ứng viên và click button Submit.    5. Hệ thống cập nhật các thông tin HR vừa đặt và lưu lại vào bảng Candidate. |
| 1. Ca sử dụng bắt đầu khi HR, Trainer muốn nhận xét    1. HR, Trainer click vào tên ứng viên.    2. Hệ thống sẽ hiển thị thông các thông tin cơ bản của ứng viên theo từng trạng thái.    3. HR, Trainer click vào button hiển thị trên màn hình tùy theo status hiện có của ứng viên.    4. Hệ thống sẽ hiển thị popup để HR, Trainer có thể viết nhận xét cho ứng viên và click button Submit.    5. Hệ thống sẽ lưu thông tin thay đổi của ứng viên vào bảng Candidate. Usecase kết thúc |
| **Luồng rẽ nhánh** | 1. Tại các bước của luồng cơ bản nếu không nhập đầy đủ thông tin hệ thống hiển thị thông báo lỗi: This field is requied!. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.7 Đặc tả usecase bảo trì ứng viên

#### 3.1.3.11 Đặc tả usecase Bảo trì bài Test

|  |  |
| --- | --- |
| **ID** | UC11 |
| **Tên UseCase** | Bảo trì bài test |
| **Mô tả** | Usecase cho phép HR, Trainer có thể thêm, sửa, xóa bài test khi cần thiết. |
| **Actor** | HR, Trainer |
| **Tiền điều kiện** | Đăng nhập với quyền HR, Trainer |
| **Hậu điều kiện** | Không có |
| **Luồng cơ bản** | 11. Ca sử dụng bắt đầu khi HR, Trainer muốn thêm bài test mới.  1.1 HR, Trainer chọn button thêm mới bài test.  1.2 Hệ thống sẽ hiển thị popup chứa form Thêm mới bài test.  1.3 HR, Trainer nhập đầy đủ thông tin cần thiết vào form và click button Submit  1.4 Hệ thống sẽ lưu thông tin của bài test vào bảng Test. Usecase kết thúc |
| 2. Ca sử dụng bắt đầu khi HR, Trainer muốn sửa bài test.  2.1 HR, Trainer click vào tên bài test trong danh sách  2.2 Hệ thống hiển thị popup chứa form sửa thông tin bài test và tất cả các thông tin của bài test vừa chọn  2.3 HR, Trainer sửa các thông tin cần thiết và click button Submit  2.4 Hệ thống cập nhật các thông tin HR, Trainer vừa sửa và lưu lại vào bảng Test. |
| 1. Ca sử dụng bắt đầu khi HR, Trainer muốn xóa bài test.   3.1 HR, Trainer click vào biểu tượng thùng rác ở phía bên phải màn hình  3.2 Hệ thống hiển thị thông báo xác nhận có muốn xóa hay không?  3.3 HR, Trainer click vào button xác nhận xóa  3.4 Hệ thống xóa bài test ra khỏi bảng Test và cập nhật lại danh sách bài test. |
| **Luồng rẽ nhánh** | 1. Tại bước 1,2 của luồng cơ bản nếu không nhập đầy đủ thông tin hệ thống hiển thị thông báo lỗi: This field is requied!. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.8 Đặc tả usecase bảo trì bài test

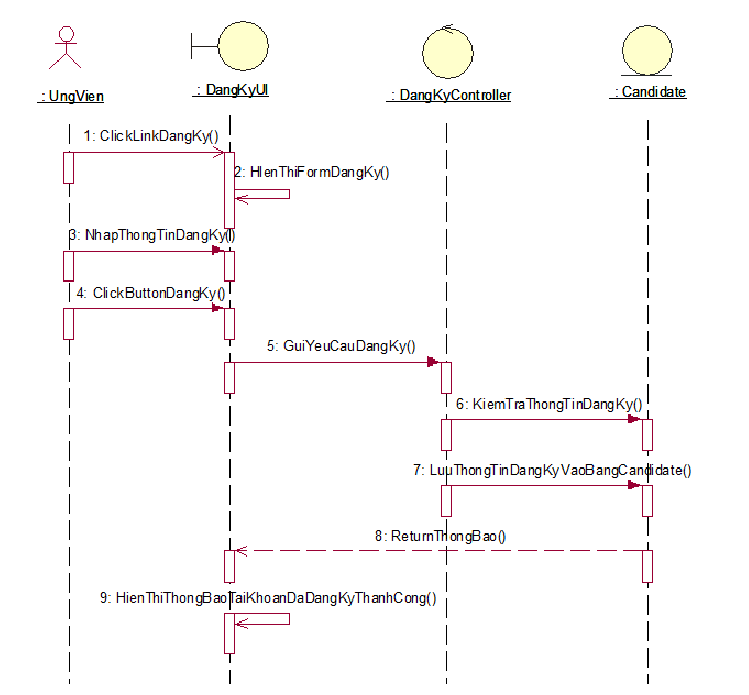
#### 3.1.3.12 Đặc tả usecase Bảo trì lớp học

|  |  |
| --- | --- |
| **ID** | UC12 |
| **Tên UseCase** | Bảo trì lớp học |
| **Mô tả** | Usecase cho phép HR có thể thêm, sửa, xóa lớp học khi cần thiết. |
| **Actor** | HR |
| **Tiền điều kiện** | Đăng nhập với quyền HR |
| **Hậu điều kiện** | Không có |
| **Luồng cơ bản** | 1. Ca sử dụng bắt đầu khi HR muốn thêm lớp học mới.    1. HR chọn button thêm mới lớp học.    2. Hệ thống sẽ hiển thị popup chứa form Thêm mới lớp học.    3. HR nhập đầy đủ thông tin cần thiết vào form và click button Submit    4. Hệ thống sẽ lưu thông tin của lớp học vào bảng Class. Usecase kết thúc |
| 1. Ca sử dụng bắt đầu khi HR muốn sửa lớp học.    1. HR click vào tên lớp học trong danh sách    2. Hệ thống hiển thị popup chứa form sửa thông tin lớp học và tất cả các thông tin của lớp học vừa chọn    3. HR sửa các thông tin cần thiết và click button Submit    4. Hệ thống cập nhật các thông tin HR vừa sửa và lưu lại vào bảng Class. |
|  | 1. Ca sử dụng bắt đầu khi HR muốn xóa lớp học.    1. HR click vào biểu tượng thùng rác ở phía bên phải màn hình    2. Hệ thống hiển thị thông báo xác nhận có muốn xóa hay không?    3. HR click vào button xác nhận xóa    4. Hệ thống xóa lớp học ra khỏi bảng Class và cập nhật lại danh sách lớp học. |
| **Luồng rẽ nhánh** | 1. Tại bước 1,2 của luồng cơ bản nếu không nhập đầy đủ thông tin hệ thống hiển thị thông báo lỗi: This field is requied!. |
| 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case, nếu không kết nối được với CSDL thì hệ thống sẽ thông báo lỗi và use case kết thúc. |
| **Yêu cầu đặc biệt** | Không có |

Bảng 3.9 Đặc tả usecase bảo trì lớp học

### Biểu đồ tuần tự

*3.1.4.1 Biểu đồ tuần tự Đăng Ký*



Hình 3.6 Biểu đồ tuần tự đăng ký

*3.1.4.2 Biểu đồ tuần tự Đăng nhập*

A diagram of a company

Description automatically generated A diagram of a computer

Description automatically generated with medium confidence

Hình 3.7 Biểu đồ tuần tự đăng nhập

*3.1.4.3 Biểu đồ tuần tự Nhận bài thi*

A diagram of a diagram

Description automatically generated

Hình 3.8 Biểu đồ tuần tự nhận bài thi

*3.1.4.4 Biểu đồ tuần tự Nộp bài thi*

A diagram of a project

Description automatically generated

Hình 3.9 Biểu đồ tuần tự nộp bài thi

*3.1.4.6 Biểu đồ tuần tự Xem thông tin cá nhân*

A diagram of a diagram

Description automatically generated

Hình 3.10 Biểu đồ tuần tự xem thông tin cá nhân

*3.1.4.7 Biểu đồ tuần tự Tìm kiếm ứng viên*

A diagram of a computer

Description automatically generated

Hình 3.11 Biểu đồ tuần tự tìm kiếm ứng viên

*3.1.4.8 Biểu đồ tuần tự Bảo trì ứng viên*

A diagram of a computer

Description automatically generated

Hình 3.12 Biểu đồ tuần tự bảo trì ứng viên

*3.1.4.9 Biểu đồ tuần tự Bảo trì bài test*

A diagram of a software project

Description automatically generated with medium confidence

A diagram of a workflow

Description automatically generated

A diagram of a project

Description automatically generated

Hình 3.13 Biểu đồ tuần tự bảo tri bài test

*3.1.4.10 Biểu đồ tuần tự Bảo trì lớp học*

A diagram of a project

Description automatically generated A diagram of a company

Description automatically generated A diagram of a company

Description automatically generated

Hình 3.14 Biểu đồ tuần tự bảo trì lớp học

**3.2 Giao diện ứng viên**

**3.2.1 Trang đăng ký**

Trang đăng ký bao gồm form chứa các thông tin mà ứng viên cần điền để có thể gửi CV cùng các thông tin cá nhân cho công ty tuyển dụng.

A cartoon character holding a clipboard with check marks

Description automatically generated

Hình 3.15 Trang đăng ký

A screenshot of a computer

Description automatically generated

Hình 3.16 Trang đăng ký

**3.2.2 Trang cảm ơn**

Trang cảm ơn hiển thị sau khi ứng viên đăng ký tài khoản thành công để gửi lời cảm ơn của công ty tới ứng viên đã quan tâm và ứng tuyển vào công ty. Ngoài ra còn có thông tin liên quan tới cách thức để có thể liên hệ và trao đổi với công ty.

A screenshot of a computer

Description automatically generated

Hình 3.17 Trang cảm ơn

**3.2.3 Trang đăng nhập**

Trang đăng nhập của ứng viên sẽ bao gồm 2 bước như hình bên dưới chứa các thông tin mà ứng viên cần nhập để có thể đăng nhập và hệ thống của công ty.

A screenshot of a computer screen

Description automatically generated

A screenshot of a login page

Description automatically generated

Hình 3.18 Trang đăng nhập

**3.2.4 Trang chủ**

Trang chủ là trang hoạt động chính của ứng viên, giúp ứng viên có thể nộp bài test của mình và xem các thông tin cá nhân của bản thân.

A screenshot of a computer

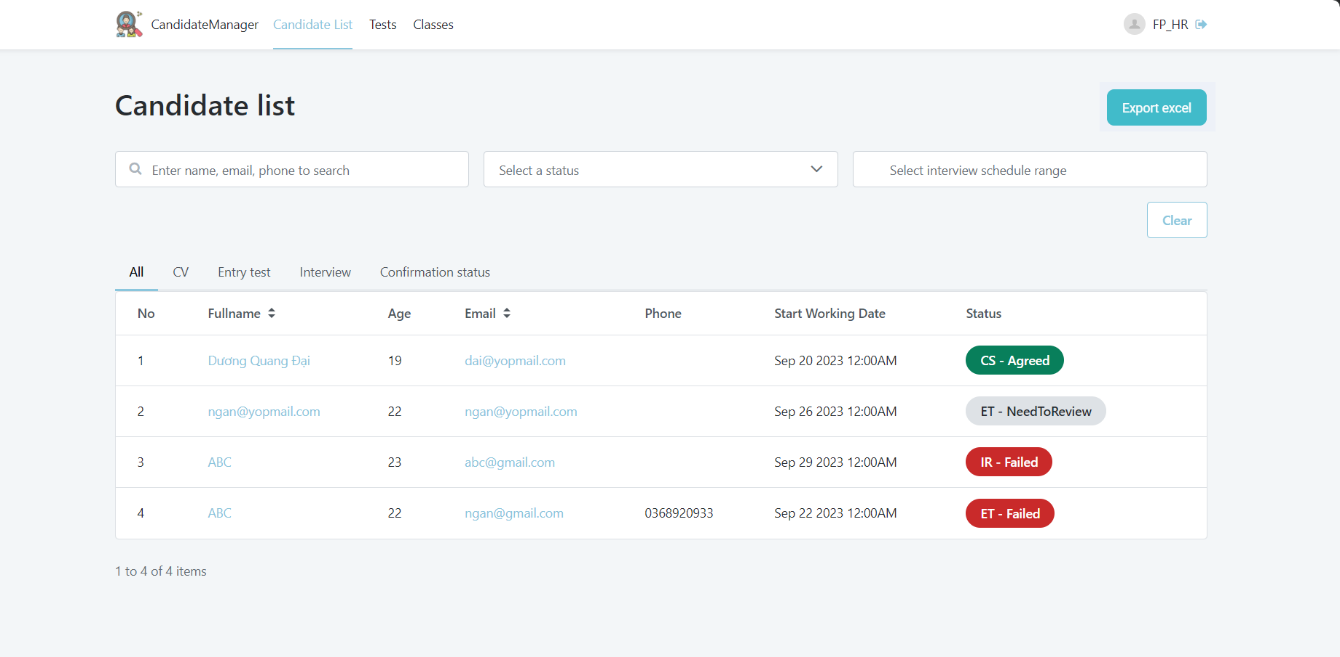
Description automatically generated

Hình 3.19 Trang chủ

**3.3 Giao diện HR, Trainer**

**3.3.1 Trang chủ**

Trang tổng quan cho phép HR, Trainer có thể xem được thông tin tổng quan về các ứng viên của công ty. HR, Trainer có thể thực hiện tìm kiếm ứng viên theo tên, email, trạng thái và ngày phỏng vấn. Ngoài ra, HR còn có thể xuất bản báo cáo bằng file excel chứa các thông tin của ứng viên.



Hình 3.20 Trang chủ

**3.3.2 Trang quản lý ứng viên**

Trang quản lý ứng viên cho phép HR, Trainer được nhìn thấy toàn bộ thông tin về ứng viên theo từng trạng thái và thực hiện chuyển đổi trạng thái cho ứng viên khi đến một vài trạng thái cần sự phản hồi từ HR hoặc Trainer.

A screenshot of a computer

Description automatically generated

Hình 3.21 Trang quản lý ứng viên

Khi click chọn một ứng viên thì sẽ hiển thị ra thông tin chi tiết ứng viên, gồm có các thông tin liên quan đến cá nhân, CV, thời gian làm việc, bài test,…

**3.3.3 Trang quản lý bài test**

Trang danh sách bài test liệt kê tất cả các bài test hiện có, khi click vào “Add Test” sẽ hiển thị popup thêm và các thông tin cần thiết để tạo một bài test mới. Ngoài ra HR, Trainer có thể tìm kiếm bài test theo tên.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Khi HR, Trainer click vào tên bài test, thông tin chi tiết của bài test sẽ hiển thị bên dưới danh sách bài test. Ngoài ra, HR hoặc Trainer còn có thể sửa lại thông tin của bài test khi click vào nút “Edit”.

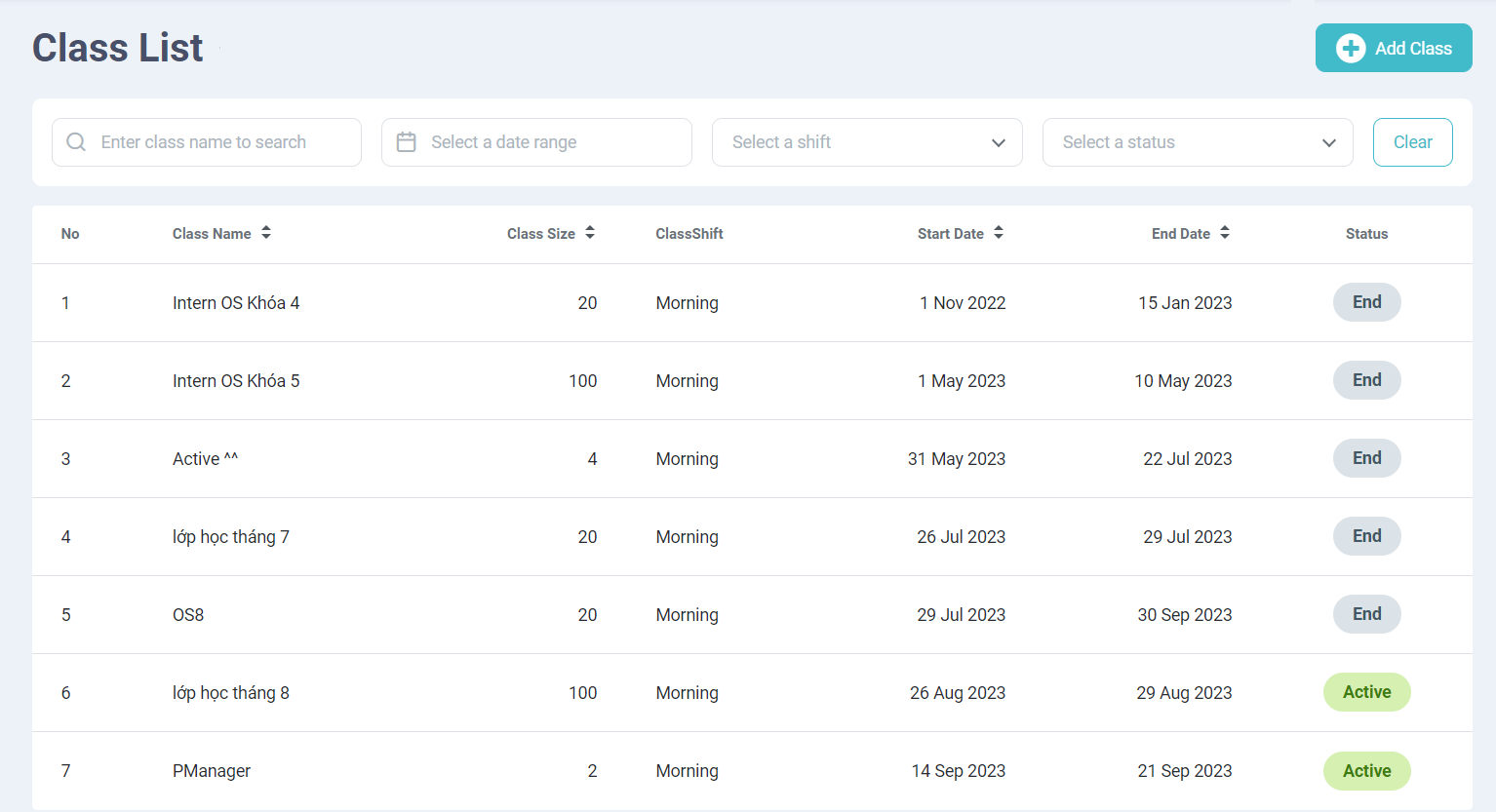
A screenshot of a computer

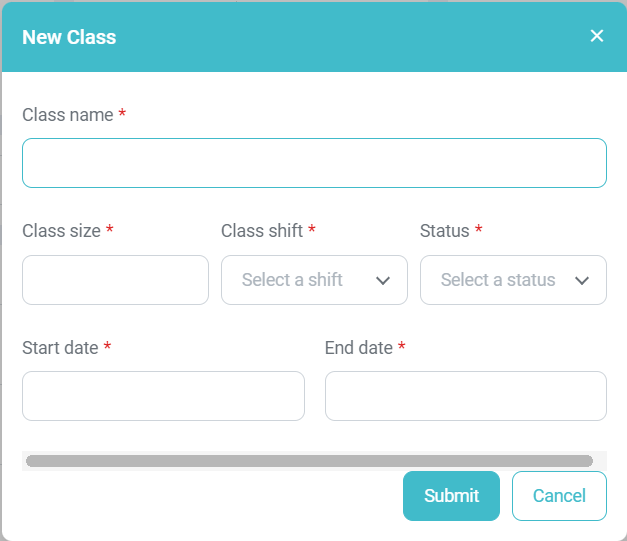
Description automatically generated

Hình 3.22 Trang quản lý bài test

**3.3.4 Trang quản lý lớp học**

Trang quản lý lớp học hiển thị tất cả các lớp học hiện có và cho phép HR có thể thực hiện thêm, sửa, xóa lớp học.





Hình 3.23 Trang quản lý lớp học

# 

# KẾT LUẬN

Sau một thời gian nghiên cứu và xây dựng đề tài “Phát triển ứng dụng Web hỗ trợ tuyển dụng sử dụng Outsystems”, em đã phân tích, thiết kế và hoàn thành về cơ bản theo đúng những yêu cầu về nội dung và thời gian đã định.

Trong quá trình nghiên cứu và thực hiện đề tài vẫn gặp nhiều hạn chế về kinh nghiệm và kiến thức nên đề tài của em không thể tránh khỏi những thiếu xót. Em rất mong nhận được ý kiến đóng góp từ quý thầy cô và các bạn để đề tài được hoàn thiện hơn.

**Kết quả đạt được**

* Khảo sát về No-code/Low-code.
* Nghiên cứu chọn lựa môi trường, công cụ phát triển ứng dụng No-code/Low- code Outsystems.
* Nghiên cứu làm chủ Outsystem.
* Nắm bắt được quy trình phát triển ứng dụng no-code/low-code trong môi trường Outsystems.
* Thiết kế, xây dựng chương trình No-code/Low-code trong môi trường Outsystem.
* Tiến hành các kịch bản thử nghiệm và đánh giá.

**Hướng phát triển**

- Hoàn thiện giao diện và các chức năng cho sản phẩm (chưa hoàn thành) bằng OS.

- Cải thiện tốc độ truy xuất dữ liệu khi có nhiều người cùng truy cập.

- Thêm chức năng quản lý học viên trong từng lớp học, phát triển ứng dụng thành ứng dụng quản lý nguồn nhân lực đầu vào của công ty từ khâu ứng tuyển đến khi hoàn thành khóa thực tập.

- Em rất mong tiếp tục nhận được sự giúp đỡ và tạo điều kiện của quý thầy cô và nhà trường để em có cơ hội phát triển, hoàn thiện ứng dụng tốt hơn trong thời gian tới.

# TÀI LIỆU THAM KHẢO

[1] Nguyễn Thị Thanh Huyền, Phạm Kim Phượng, Ngô Bích Thúy (2011), *Giáo trình Phân tích thiết kế hệ thống,* NXB Giáo dục Việt Nam.

[2] Nguyễn Thị Thanh Huyền (2011), *Giáo trình Cơ sở dữ liệu,* NXB Giáo dục Việt Nam.

[3] Website khóa học Outsystems: [*https://success.outsystems.com/*](https://success.outsystems.com/documentation/11/getting_started/create_your_first_reactive_web_app/)

[4] No-code: *https://appmaster.io/vi/no-code*