

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC
CÔNG NGHỆ PHẦN MỀM

XÂY DỰNG HỆ THỐNG QUẢN LÝ NHÀ HÀNG

Giáo viên hướng dẫn:
TS. Nguyễn Bảo Ân

<i>Sinh viên thực hiện:</i>	MSSV
Phạm Phước Vinh	110121144
Cao Khải Minh	110121145
Trần Quốc Ninh	110121079
Lớp : DA21TTC – DA21TTA	

Trà Vinh, tháng 6 năm 2024

LỜI CẢM ƠN

Nhóm em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các Thầy Cô của Trường Đại học Trà Vinh, đặc biệt là Thầy Nguyễn Bảo Ân đã tận tình truyền đạt kiến thức trong môn công nghệ phần mềm. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu bài báo cáo mà nó còn là hành trang quý báu để nhóm em áp dụng vào thực tế một cách vững chắc và tự tin.

Do còn hạn chế về kiến thức cũng như những kinh nghiệm thực tế cho nên không tránh khỏi được những sai sót trong quá trình tìm hiểu và trình bày rất mong nhận được sự đóng góp ý kiến của cô để bài báo cáo được hoàn chỉnh hơn.

Sau cùng, nhóm em xin kính chúc Thầy thật nhiều sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN	i
MỤC LỤC	iv
DANH MỤC HÌNH ẢNH	vi
CHƯƠNG 1: GIỚI THIỆU	1
1.1 Lý do chọn đề tài	1
1.2 Yêu cầu của ứng dụng	1
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	2
2.1 Agile.....	2
2.2 Lịch sử hình thành và phát triển của Agile.....	2
2.3 Đặc trưng của Agile.....	3
2.4 Tìm hiểu 12 nguyên tắc trong Agile.....	4
2.5 Ưu nhược điểm của Agile.....	4
2.5.1 Ưu điểm của Agile.....	5
2.5.2 Nhược điểm của Agile.....	5
2.6 Các phương pháp Agile phổ biến	6
2.7 Quy trình triển khai Agile.....	6
2.8 Scrum.....	8
2.9 Các khái niệm cơ bản Scrum.....	8
2.10 Các công cụ (artifacts) Scrum	10
2.11 Mô hình 3 lớp	11
2.12 Winform application.....	12
2.12 Ngôn ngữ lập trình C#	13
2.13 SQL Server	13
2.14 Docker.....	15
CHƯƠNG 3 XÁC ĐỊNH NHU CẦU	17
3.1 Đặc tả nhu cầu	17
CHƯƠNG 4 LẬP KẾ HOẠCH SCRUM	19
4.1 Backlogs	19
4.1.1 Product backlog.....	19
4.1.2 Sprint backlog.....	19

4.1.3	Technical backlog.....	20
4.1.4	Support backlog.....	20
4.1	Lập bảng kế hoạch Scrum	20
CHƯƠNG 5 HIỆN THỰC HÓA KẾ HOẠCH.....		22
5.1	Thiết kế giao diện	22
5.1.1	Giao diện đăng nhập.....	22
5.1.2	Giao diện gọi món ăn	22
5.1.3	Giao diện thông tin cá nhân.....	23
5.1.4	Giao diện doanh thu.....	24
5.1.5	Giao diện thức ăn.....	24
5.1.6	Giao diện danh mục.....	25
5.1.7	Giao diện bàn ăn.....	25
5.1.8	Giao diện tài khoản.....	26
5.1.9	Giao diện Report.....	26
5.2	Triển khai SQL Server với Docker.....	26
5.3	Mã nguồn	30
5.4	Biểu đồ Sprint Burndown chart	30
CHƯƠNG 6 KẾT LUẬN		33
3.1	Kết quả đạt được.....	33
3.2	Hạn chế	33
3.3	Hướng phát triển.....	33
TÀI LIỆU THAM KHẢO.....		34

DANH MỤC HÌNH ẢNH

Hình 1 Các phương pháp Agile phổ biến	6
Hình 2 Quy trình Agile	6
Hình 3 Scrum Agile	8
Hình 4 Mô hình 3 lớp	11
Hình 5 Cấu trúc thư mục	12
Hình 6 SQL Server	14
Hình 7 Docker	15
Hình 8 Giao diện đăng nhập	22
Hình 9 Giao diện gọi món ăn	22
Hình 10 Giao diện thông tin cá nhân	23
Hình 11 Giao diện doanh thu	24
Hình 12 Giao diện thức ăn	24
Hình 13 Giao diện danh mục	25
Hình 14 Giao diện bàn ăn	25
Hình 15 Giao diện tài khoản	26
Hình 16 Giao diện Report	26
Hình 17 Docker Desktop	27
Hình 18 Command Prompt (CMD)	27
Hình 19 Chạy lệnh trên Command Prompt (CMD)	27
Hình 20 Docker Images sau khi chạy lệnh	28
Hình 21 Chạy lệnh trên Command Prompt (CMD)	28
Hình 22 Docker Container	29
Hình 23 Đăng nhập SQL Server	29
Hình 24 Đăng nhập SQL thành công	29
Hình 25 Mã nguồn ứng dụng	30
Hình 26 Biểu đồ Burndown chart Sprint 1	30
Hình 27 Biểu đồ Burndown chart Sprint 2	31
Hình 28 Biểu đồ Burndown chart Sprint 3	31
Hình 29 Biểu đồ Burndown chart Sprint 4	32
Hình 30 Biểu đồ Burndown chart Sprint 5	32

CHƯƠNG 1: GIỚI THIỆU

1.1 Lý do chọn đề tài

Trong những năm gần đây, việc kinh doanh lĩnh vực nhà hàng đang ngày càng phát triển và có sự cạnh tranh, đòi hỏi hiệu quả cao hơn trong vấn đề quản lý. Để tối ưu hóa các quy trình, giảm thiểu sai sót, đáp ứng nhanh chóng nhu cầu thị trường đòi hỏi một hệ thống quản lý nhà hàng tự động và thông minh.

Lợi ích mà hệ thống quản lý nhà hàng này mang lại không chỉ đối với doanh nghiệp mà còn có ý nghĩa to lớn đối với khách hàng và nhân viên của doanh nghiệp.

Nó cung cấp giải pháp tối ưu từ việc lên món, quản lý bàn ăn, các danh mục, món ăn đến thống kê doanh thu. Giúp nhiều tác vụ được tự động hóa làm giảm khối lượng công việc cho nhân viên từ đó tăng hiệu suất làm việc lên đáng kể.

Ở phía đối diện, khách hàng được trải nghiệm dịch vụ tự động, món ăn được nhân viên lựa chọn thông qua ứng dụng và được mang đến đúng bàn ăn của khách hàng một cách nhanh chóng. Ngoài ra khách hàng không cần phải chờ đợi nhân viên thanh toán tại quầy quá lâu, mọi thứ đã được tính toán tự động từ trước khi khách hàng ra về, đảm bảo tính chính xác và tiết kiệm thời gian cho khách hàng.

Sử dụng hệ quản trị cơ sở dữ liệu SQL server để lưu trữ dữ liệu, nghiệp vụ ứng dụng được viết bằng ngôn ngữ lập trình C# với thư viện Windows Form, kiến trúc mô hình 3 lớp.

1.2 Yêu cầu của ứng dụng

Yêu cầu chức năng:

Đối với người sử dụng: Có thể thêm món ăn vào bàn, sửa và xóa khỏi bàn, thanh toán, sửa thông tin tài khoản.

Đối với Admin: Có toàn quyền của người dùng và thêm một số chức năng nâng cao như thêm sửa xóa tất cả tài khoản, thêm sửa xóa danh mục và món ăn, thêm sửa xóa bàn ăn.

Yêu cầu phi chức năng:

Ứng dụng có hiệu suất cao, dễ dàng bảo trì, bảo mật ở mức ổn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Agile

Agile (Agile Software Development) là một phương thức phát triển phần mềm linh hoạt, được thực hiện bằng cách sử dụng các bước lặp ngắn từ 1 đến 4 tuần. Mục tiêu của Agile là giúp rút ngắn thời gian phát triển sản phẩm, đưa sản phẩm đến với tay khách hàng càng sớm càng tốt.

Về bản chất, Agile giống như một phương pháp luận, một triết lý dựa trên nguyên tắc phân đoạn vòng lặp (iterative) và tăng trưởng (incremental) nên sở hữu tính linh hoạt cao. Tính chất này đi ngược lại với các phương pháp quản lý dự án truyền thống – vốn dĩ triển khai các giai đoạn một cách tuyến tính và vô cùng bị động trước các thay đổi bất ngờ.

Ngày nay, triết lý Agile đã vượt xa khỏi khu vực truyền thống của mình là phát triển phần mềm để đóng góp sự thay đổi trong cách thức làm việc, quản lý, sản xuất ở các ngành khác như sản xuất, dịch vụ, sales, marketing, giáo dục,... và trở thành một phương pháp quản lý dự án phổ biến nhất hiện nay với nhiều đại diện được gọi là các phương pháp “họ Agile”.

2.2 Lịch sử hình thành và phát triển của Agile

Trước khi có sự xuất hiện của Agile, mô hình Waterfall được xem như tiêu chuẩn vàng để phát triển phần mềm. Thông thường, quy trình sẽ bắt đầu bằng việc một nhà phân tích kinh doanh viết tài liệu yêu cầu nghiệp vụ, mô tả những yêu cầu mà họ cần từ phần mềm. Những tài liệu này thường rất dài và chi tiết, chứa đựng tất cả mọi thông tin từ chiến lược tổng thể đến các thông số kỹ thuật. Sau khi tất cả các tài liệu hoàn tất, các nhà phát triển mới bắt đầu code phần mềm, hoàn thiện nó và test. Tất cả quy trình này có thể mất đến một vài năm để kết thúc và để phần mềm được đưa vào sử dụng.

Việc phát triển các phần mềm bắt đầu thay đổi với sự phát triển của mạng Internet. Đây cũng là lúc mà hiệu quả của phương pháp Waterfall bị đặt một dấu chấm hỏi. Nó làm tiêu tốn quá nhiều thời gian và công sức của nhóm thực hiện, trong khi lại thiếu đi tính khả thi khi nó chỉ hoàn toàn được xây dựng từ các nghiên cứu mà thiếu sự phản hồi thực tế từ người dùng.

Vào năm 2001, một nhóm gồm 17 nhà phát triển phần mềm kinh nghiệm, bao gồm Kent Beck, Martin Fowler, Ron Jeffries, Ken Schwaber và Jeff Sutherland, đã cùng nhau nghiên cứu và xuất bản Tuyên ngôn Agile với 4 giá trị quan trọng và 12

nguyên tắc hướng dẫn cách tiếp cận lặp đi lặp lại và tập trung vào con người để phát triển phần mềm. Tuyên ngôn Agile được xem như là niềm tin của họ vào cách vận hành hoàn hảo mà quá trình phát triển phần mềm hiện đại nên có.

2.3 Đặc trưng của Agile

Tính lặp (Iterative): Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại, thường có khung thời gian ngắn (từ 1-4 tuần). Trong mỗi phân đoạn đó, nhóm phát triển dự án sẽ thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử để cho ra các phần nhỏ của sản phẩm.

Tính tăng trưởng và tiến hóa (Incremental & Evolutionary): Cuối các phân đoạn, nhóm cho ra các phần nhỏ của sản phẩm cuối cùng, thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng. Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn.

Tính thích nghi (adaptive): Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu,...) đều có thể được đáp ứng theo cách thích hợp.

Nhóm tự tổ chức và liên chức năng: Các cấu trúc nhóm này tự phân công công việc mà không dựa trên các mô tả cứng nhắc về chức danh hay một sự phân cấp rõ ràng. Nhóm tự tổ chức đã đủ các kỹ năng cần thiết để có thể được trao quyền tự ra quyết định, tự quản lý và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.

Quản lý tiến trình thực nghiệm (Empirical Process Control): Các nhóm Agile ra các quyết định dựa trên các dữ liệu thực tiễn (data-driven) thay vì tính toán lý thuyết hay các tiên giả định. Agile rút ngắn vòng đời phản hồi để dễ dàng thích nghi và gia tăng tính linh hoạt nhờ đó có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

Giao tiếp trực diện (face-to-face communication): Agile không phản đối việc tài liệu hóa, nhưng đánh giá cao hơn việc giao tiếp trực diện thay vì thông qua giấy tờ. Agile khuyến khích nhóm phát triển trực tiếp nói chuyện để hiểu rõ hơn về những gì khách hàng thực sự cần. Trong giao tiếp giữa nội bộ nhóm, Agile khuyến khích trực

tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.

Phát triển dựa trên giá trị (value-based development): Một trong các nguyên tắc cơ bản của Agile là “sản phẩm chạy tốt chính là thước đo của tiến độ”. Nhóm Agile thường cộng tác trực tiếp và thường xuyên với khách hàng để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án.

2.4 Tìm hiểu 12 nguyên tắc trong Agile

1. Ưu tiên cao nhất của Agile là làm hài lòng khách hàng thông qua việc cung cấp sớm và liên tục các sản phẩm có giá trị.

2. Agile luôn chào đón các yêu cầu thay đổi, kể cả khi nó xuất hiện ở giai đoạn cuối của quá trình triển khai, khi mọi thứ đã gần như hoàn thiện. Quy trình Agile luôn khai thác sự thay đổi để tạo nên lợi thế cạnh tranh cho khách hàng.

3. Giao phần mềm chạy được cho khách hàng một cách thường xuyên, từ vài tuần đến vài tháng.

4. Các nhà kinh doanh và nhà phát triển phần mềm phải làm việc cùng nhau hàng ngày trong suốt dự án.

5. Tập trung xây dựng dự án xung quanh những cá nhân có động lực, cung cấp cho họ môi trường và sự hỗ trợ cần thiết, cùng với niềm tin vào việc hoàn thành tốt công việc.

6. Phương pháp hiệu quả nhất để truyền đạt thông tin là trao đổi trực tiếp.

7. Phần mềm hoạt động tốt chính là thước đo của tiến độ triển khai Agile.

8. Quy trình Agile thúc đẩy sự phát triển bền vững.

9. Sự chú ý liên tục đến yếu tố kỹ thuật và thiết kế sẽ cải tiến sự linh hoạt.

10. Nghệ thuật tối đa hóa lượng công việc chưa xong: Sự đơn giản là cần thiết.

11. Những thiết kế và yêu cầu tốt nhất chỉ xuất hiện tại những nhóm làm việc có khả năng tự tổ chức tốt.

12. Nhóm thực thi cần suy nghĩ về việc làm việc hiệu quả một cách thường xuyên để có sự điều chỉnh hành vi phù hợp.

2.5 Ưu nhược điểm của Agile

2.5.1 Ưu điểm của Agile

Mô hình Agile ngày càng trở nên phổ biến, được nhiều doanh nghiệp áp dụng để phát triển sản phẩm của mình bởi nó sở hữu những ưu điểm vượt trội so với những phương pháp truyền thống.

- Dễ dàng thực hiện thay đổi ở bất kỳ giai đoạn nào của dự án, thích nghi nhanh và hiệu quả với sự thay đổi, bao gồm cả yêu cầu sửa lại sản phẩm, sự biến động từ thị trường,...

- Phát triển và bàn giao sản phẩm nhanh hơn, bởi việc chia nhỏ dự án cho phép đội ngũ có thể tiến hành kiểm tra theo từng phần, phát hiện sự cố và sửa chữa vấn đề nhanh hơn.

- Sản phẩm đạt chất lượng tốt hơn, do luôn nhận được phản hồi ngay lập tức từ phía khách hàng và được tối ưu lại ngay sau đó.

- Lãng phí ít tài nguyên hơn vì nhóm luôn thực hiện các công việc đã được cập nhật, giảm tải thời gian chờ, giảm tải thời gian sửa lỗi sản phẩm, giảm tải số lượng lớn giấy tờ tài liệu,...

- Người tham gia dự án không cần phải nắm mọi thông tin ngay từ đầu, phù hợp với những dự án chưa xác định rõ ràng được mục tiêu cuối cùng.

2.5.2 Nhược điểm của Agile

Bên cạnh những ưu điểm vượt trội, cũng không khó để nhận ra một số nhược điểm còn tồn tại khi triển khai mô hình Agile trong thực tế:

- Khó lên kế hoạch dự án, đặc biệt là khó xác định rõ ràng thời gian bàn giao sản phẩm cuối cùng, không nắm rõ được chi phí thực sự của dự án là bao nhiêu,... vì dự án được chia nhỏ thành các vòng lặp khác nhau.

- Bắt buộc phải hướng dẫn và đào tạo chi tiết, thì thành viên dự án mới có thể hiểu rõ được mô hình Agile và thực hiện theo nó một cách rõ ràng, đặc biệt là trong thời gian đầu.

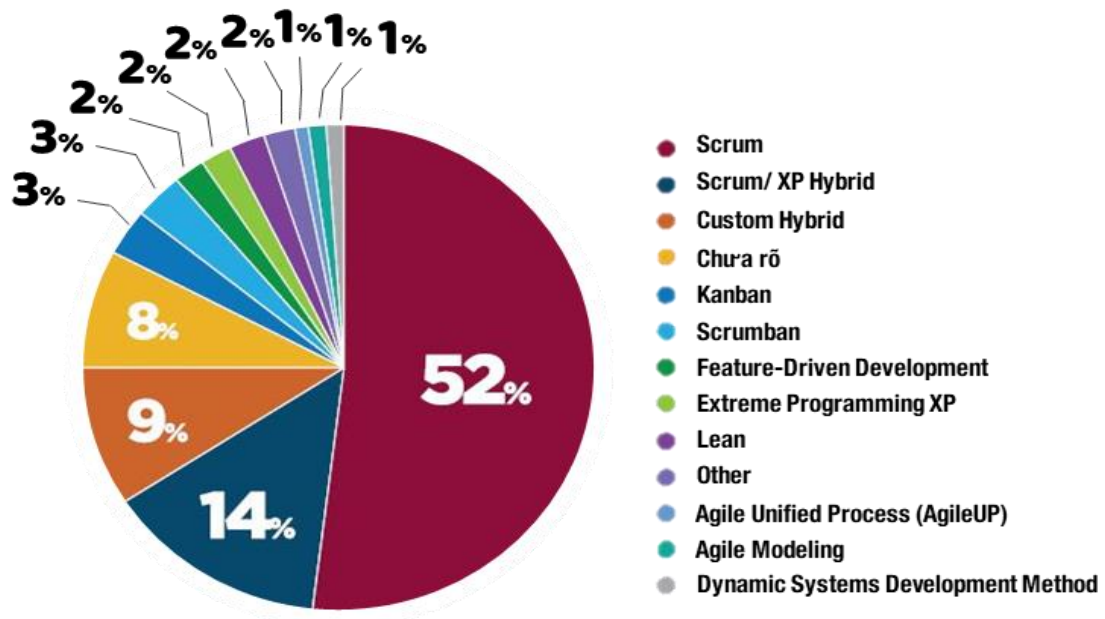
- Ít tài liệu hướng dẫn về dự án và không xác định rõ được kỳ vọng và thành phẩm ngay từ đầu, do người thực thi mô hình Agile thường tin rằng mọi thứ sẽ thay đổi rất nhiều, không quá cần thiết phải ghi chép và lưu trữ tài liệu.

- Đòi hỏi sự cam kết về thời gian và công sức từ các bên hơn, vì tất cả mọi người cần phải liên tục tương tác với nhau trong suốt quá trình thực thi dự án.

- Chi phí thực hiện dự án theo mô hình Agile thường cao hơn so với các phương pháp phát triển khác.

2.6 Các phương pháp Agile phổ biến

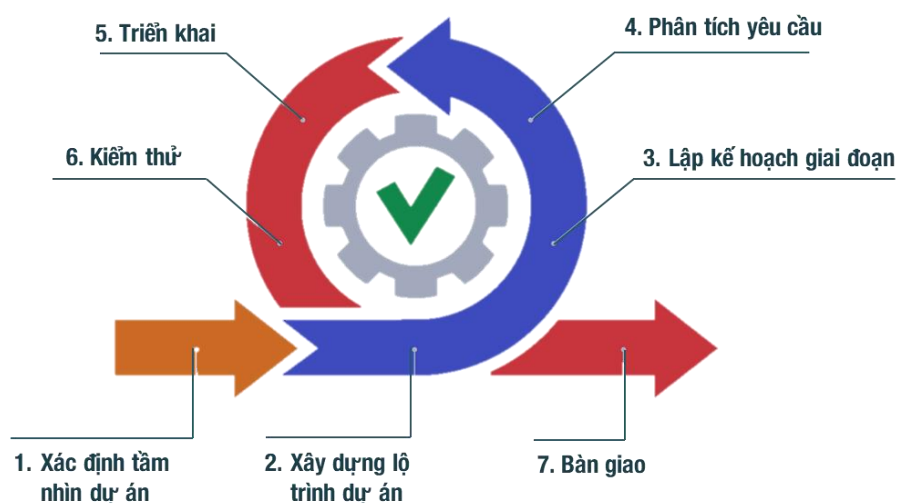
Bảng thống kê dưới đây liệt kê **13 phương pháp họ Agile**, cho thấy phần lớn các công ty hiện nay đã sử dụng Scrum như một cách tiếp cận cơ bản. Bên cạnh đó, nhiều công ty đã kết hợp các phương pháp lại với nhau.



Hình 1 Các phương pháp Agile phổ biến

2.7 Quy trình triển khai Agile

Quy trình Agile



Hình 2 Quy trình Agile

Giai đoạn 1: Lên ý tưởng và xây dựng lộ trình dự án

Mọi dự án phát triển sản phẩm đều bắt đầu từ giai đoạn xác định tầm nhìn, lên ý tưởng và xây dựng lộ trình, với sự tham gia của tất cả các bên liên quan: kỹ sư phần mềm, nhà đầu tư và bất kỳ ai khác có kiến thức về sản phẩm.

Ví dụ, với mô hình Scrum, nhóm thực thi sẽ triển khai các nhiệm vụ ở giai đoạn 1 như sau:

- Xác định và ghi lại tất cả các yêu cầu về kinh doanh và người dùng
- Gắn các yêu cầu này vào trong tính năng của sản phẩm
- Thiết lập mục tiêu cho sản phẩm mới
- Chia nhỏ các đầu mục công việc và các nhiệm vụ ưu tiên
- Phân công nhiệm vụ cho các thành viên trong nhóm
- Ước tính thời gian và chi phí cho dự án

Giai đoạn này đặc biệt quan trọng vì nó quyết định phạm vi ban đầu của dự án. Vì quy trình triển khai bao gồm các vòng lặp liên tục, nên các nhóm thực thi cần giữ các yêu cầu ở mức tối thiểu, và bổ sung dần các tính năng trong các giai đoạn phát triển về sau.

Giai đoạn 2: Lập kế hoạch giai đoạn (Phát triển dự án)

Sau khi giai đoạn lên ý tưởng hoàn thành, nhóm thực thi sẽ nhận được một tài liệu xác định toàn bộ phạm vi dự án. Đây là lúc các kỹ sư phần mềm bắt đầu sắp xếp các công việc của mình vào các vòng lặp (Sprint), phân tích yêu cầu của khách hàng để hoàn thiện Product Backlog – danh sách sắp xếp theo trật tự về những điều cần thiết để cải tiến sản phẩm, và xây dựng phiên bản đầu tiên của sản phẩm.

Giai đoạn phát triển dự án sẽ khác nhau tùy thuộc vào loại sản phẩm mà nhóm triển khai. Đây cũng là giai đoạn dài nhất trong mô hình Agile, nơi mà phần lớn các công việc sẽ được thực hiện. Mục tiêu cốt lõi của nhóm thực thi trong giai đoạn này chính là xây dựng một số phần chức năng của sản phẩm vào cuối mỗi Sprint.

Giai đoạn 3: Triển khai dự án

Trong quá trình phát triển theo mô hình Agile, giai đoạn triển khai phụ thuộc rất nhiều vào những gì nhóm đã tạo ra trong các giai đoạn trước đó. Ở giai đoạn này, nhóm cần hiểu rõ về nơi sản phẩm sẽ được triển khai và cách thức triển khai.

Giai đoạn 4: Kiểm thử

Các hoạt động kiểm thử (Testing) được diễn ra trong suốt quá trình triển khai Agile. Trước khi phát hành sản phẩm ra thị trường, phải luôn đảm bảo rằng nó đã trải qua hết các bài kiểm tra và các thử nghiệm đảm bảo chất lượng. Các bộ phận chuyên trách sẽ đưa ra quyết định liệu sản phẩm có đầy đủ chức năng và hoạt động bình thường hay không.

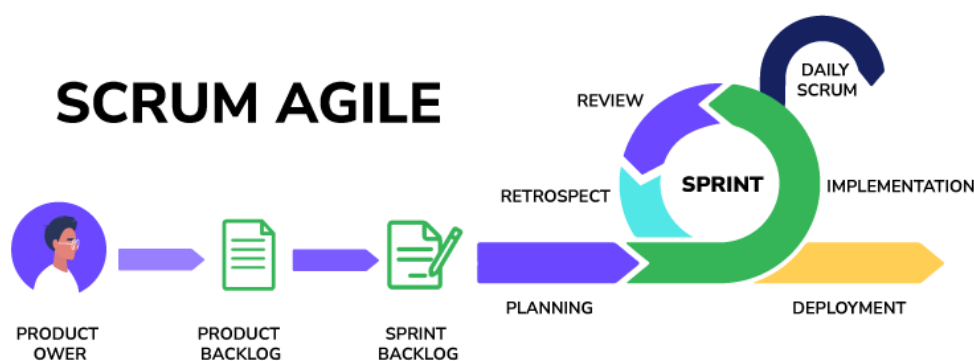
Giai đoạn 5: Bàn giao và bảo trì sản phẩm

Trong giai đoạn cuối cùng của mô hình Agile, sau khi đã bàn giao sản phẩm cho khách hàng vào cuối Sprint, nhóm thực thi dự án sẽ vẫn tập trung vào việc bảo trì liên tục để giải quyết mọi vấn đề, và để duy trì chức năng cũng như hiệu suất làm việc ở mức cao.

Nhiều vấn đề sẽ xảy ra khi người dùng bắt đầu tương tác với sản phẩm, nên nhóm có thể phải xử lý một số lỗi tính năng hoặc vấn đề về hiệu suất. Nhưng điều này cũng mang đến cho nhóm cơ hội thu thập phản hồi và áp dụng các cải tiến mới trong các phân đoạn Agile tiếp theo.

2.8 Scrum

Scrum là một “bộ khung làm việc” cơ bản để tiếp cận những công việc phức tạp. Dựa trên bộ khung này, nhóm làm việc có thể áp dụng những quy trình, kỹ thuật khác nhau cho công việc của mình... Nó là một thành viên của họ Agile.



Hình 3 Scrum Agile

2.9 Các khái niệm cơ bản Scrum

1. Scrum Team

Scrum team chia làm 3 vai trò bao gồm những thành phần sau:

Product Owner: Nhiệm vụ của Product Owner là đảm bảo việc quản lý những công việc còn tồn đọng (**Product backlog**) của việc phát triển sản phẩm phần mềm.

Product Owner phải liên tục cập nhật thông tin cho các thành viên trong team để họ hiểu về yêu cầu hay các tính năng cần có của sản phẩm ngay cả khi họ không trực tiếp phát triển tính năng đó.

Development Team: là những lập trình viên sẽ tham gia vào việc phát triển từng tính năng cụ thể. Các lập trình viên này có thể sẽ có kỹ năng khác nhau và một số sẽ giỏi về những kỹ năng nhất định. Tuy nhiên khi sử dụng Scrum thì tất cả các thành viên của Development Team yêu cầu phải có khả năng làm việc thay thế vị trí của nhau và không ai chỉ chịu trách nhiệm phát triển một (hoặc một số) tính năng nhất định.

Scrum Master: sẽ chịu trách nhiệm cho việc lên kế hoạch để phân công công việc, sắp xếp thứ tự ưu tiên giải quyết những công việc tồn đọng nào có trong Backlog trước, tổ chức các buổi họp với Product Owner để theo dõi tình hình và nắm thông tin cần thiết.

2. Sprint

Sprint là một phân đoạn lặp đi lặp lại trong quy trình phát triển phần mềm, có khung thời gian thường là 1 tháng (từ 1 - 4 tuần) mà theo đó sản phẩm sẽ được release phiên bản mới. Khi một Sprint kết thúc thì Scrum Master cần phải chuyển trạng thái của nó sang Done.

Khi bắt đầu một Sprint thì Scrum Master cần đưa ra mục tiêu của Sprint đó và mục tiêu này không được phép thay đổi cho tới khi Sprint hoàn thành. Tuy nhiên Product Owner vẫn có quyền huỷ một Sprint trước thời hạn kết thúc của nó.

Mặc dù để làm điều này thì Product Owner cần sự đồng thuận của Development Team cũng như Scrum Master. Sau khi một Sprint kết thúc thì các bên sẽ dựa trên kết quả của Sprint đó để lên kế hoạch cho Sprint tiếp theo.

3. Sprint Planning

Đây là bước đầu tiên cần phải thực hiện trước khi một Sprint bắt đầu. Development team họp với Product Owner để lên kế hoạch cho một sprint. Những công việc nào cần phải được hoàn thành trong Sprint này và làm sao để có thể hoàn thành những công việc này.

Sau khi thống nhất được số lượng công việc, thời gian hoàn thành thì chúng ta có thể bắt đầu Sprint. Trong khi thực hiện một Sprint chúng ta sẽ phải có những buổi họp được gọi là Daily Sprint hay Daily Meeting.

4. Daily Sprint

Các buổi họp Daily Sprint thường kéo dài khoảng 15 phút, trong buổi họp này tất cả các thành viên sẽ lần lượt báo cáo lại:

- Những gì họ đã làm được ngày hôm qua
- Những gì họ cần làm ngày hôm nay
- Những khó khăn mà họ gặp phải

Mỗi buổi họp này sẽ giúp việc dự kiến được kế hoạch đưa ra trong Sprint đang làm sẽ tiến triển ra sao và liệu có cần phải cập nhật lại bản kế hoạch đã đưa ra hay không. Tất nhiên cần nhớ rằng việc thay đổi kế hoạch này không bao gồm thay đổi mục tiêu đã đưa ra của Sprint.

5. Sprint Review

Là công việc được thực hiện bởi nhóm phát triển và product owner ở cuối mỗi Sprint nhằm đánh giá lại kết quả thực hiện được. Từ lúc Sprint mới hoàn thành và qua đó đưa ra những chỉnh sửa, thay đổi cần thiết ở Sprint sau.

6. Sprint Restrospective

Dưới sự trợ giúp của Scrum master, team phát triển sẽ tổng kết những kiến nghị và đánh giá từ bước Sprint Review ở trên để đưa ra những cải tiến nhằm nâng cao hiệu quả làm việc cũng như sản phẩm.

2.10 Các công cụ (artifacts) Scrum

Scrum sử dụng các công cụ rất đơn giản nhưng hiệu quả để trợ giúp công việc.

Product backlog

Đây là danh sách ưu tiên các tính năng (feature) hoặc đầu ra khác của dự án. Có thể hiểu như là danh sách yêu cầu (requirement) của dự án.

Product Owner chịu trách nhiệm sắp xếp độ ưu tiên cho từng hạng mục (Product Backlog Item) trong Product Backlog dựa trên các giá trị do Product Owner định nghĩa (thường là giá trị thương mại – business value).

Sprint backlog

Đây là bản kế hoạch cho một Sprint; là kết quả của buổi họp lập kế hoạch (Sprint Planning).

Với sự kết hợp của Product Owner, nhóm sẽ phân tích các yêu cầu theo độ ưu tiên từ cao xuống thấp để hiện thực hóa các hạng mục trong Product Backlog dưới dạng danh sách công việc (TODO list).

Burndown Chart

Đây là biểu đồ hiển thị xu hướng của dự án dựa trên lượng thời gian cần thiết còn lại để hoàn tất công việc.

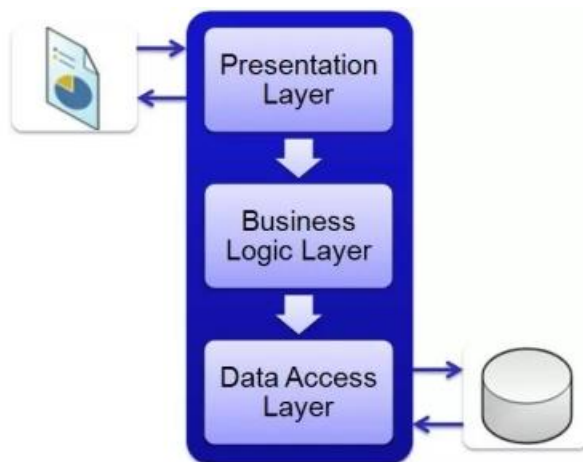
Burndown Chart có thể được dùng để theo dõi tiến độ của Sprint (được gọi là Sprint Burndown Chart) hoặc của cả dự án (Project Burndown Chart).

Biểu đồ burndown không phải là một thành tố tiêu chuẩn của Scrum theo định nghĩa mới, nhưng vẫn được sử dụng rộng rãi do tính hữu ích của nó.

2.11 Mô hình 3 lớp

Mô hình 3 lớp hay còn được gọi là mô hình Three Layer(3-Layer), mô hình này ra đời nhằm phân chia các thành phần trong hệ thống, các thành phần cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo và chạy lộn xộn.

Mô hình này phát huy hiệu quả nhất khi bạn xây dựng một hệ thống lớn, việc quản lý code và xử lý dữ liệu lỗi dễ dàng hơn.

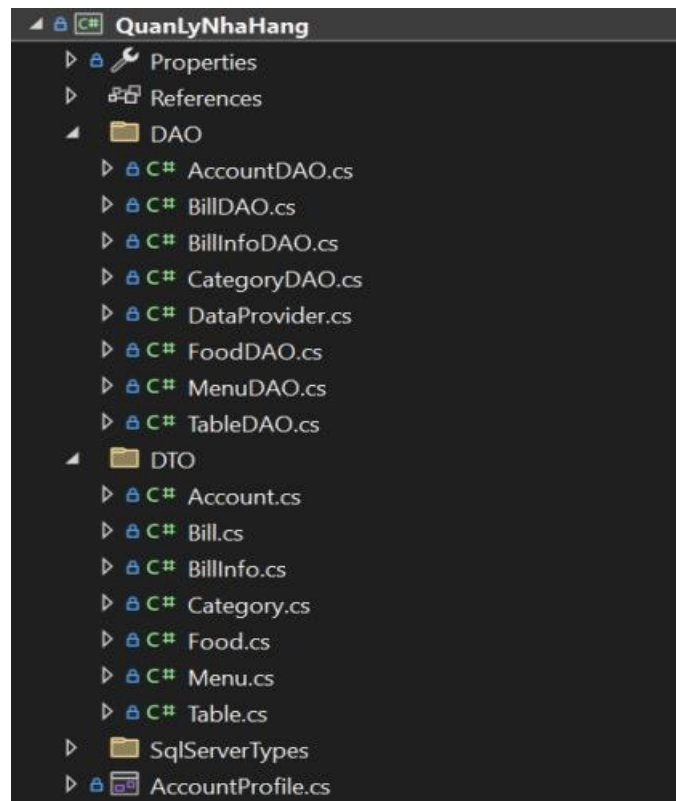


Hình 4 Mô hình 3 lớp

Ưu điểm:

- Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và maintain project tốt hơn.
- Dễ dàng phân loại các hành động tại Business.
- Dễ dàng phân loại các hàm truy xuất tại Database, phân loại hàm theo table,...
- Ứng dụng được cho các project lớn ở bên ngoài.

Cấu trúc thư mục:



Hình 5 Cấu trúc thư mục

Các tệp giao diện

Thư mục DTO: chứa những dữ liệu được xây dựng dưới dạng lớp đối tượng.

Thư mục DAO: Chứa các class thực hiện các thao tác với cơ sở dữ liệu.

2.12 Winform application

Winform là thuật ngữ mô tả một ứng dụng được viết dùng .NET Framework và có giao diện người dùng Windows Forms.

Mỗi màn hình windows cung cấp một giao diện giúp người dùng giao tiếp với ứng dụng. Giao diện này được gọi là giao diện đồ họa (GUI) của ứng dụng.

Là các ứng dụng windows chạy trên máy tính – mã lệnh thực thi ngay trên máy tính: Microsoft, Word, Excel, Access, Calculator, yahoo, Mail... là các ứng dụng Windows Forms.

Ưu điểm:

- Giao diện kéo thả dễ sử dụng.
- Tốc độ xử lý nhanh.
- Thao tác trên nhiều giao diện.

Nhược điểm:

- Winform chỉ phù hợp các ứng dụng trên desktop: ứng dụng quản lý thông tin, ứng dụng tương tác trực tiếp với người dùng.
- Đồ họa không quá cao nên giao diện sẽ thiếu tính trực quan, khó thao tác, không thân thiện với người dùng.

2.12 Ngôn ngữ lập trình C#

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.

Đặc trưng của ngôn ngữ lập trình C#:

- C# là ngôn ngữ lập trình thuần hướng đối tượng.
- C# là ngôn ngữ đơn giản.
- C# là ngôn ngữ lập trình hiện đại.
- C# là ngôn ngữ lập trình ít từ khóa.

2.13 SQL Server

SQL Server hay Microsoft SQL Server là phần mềm ứng dụng cho hệ thống quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) được phát triển bởi Microsoft vào năm 1988. Nó được sử dụng để tạo, duy trì, quản lý và triển khai hệ thống RDBMS.



Hình 6 SQL Server

Cấu trúc của SQL Server: SQL Server bao gồm 2 thành phần chính: Database Engine và SQLOS.

Database Engine: Thành phần cốt lõi của SQL Server là Database Engine. Nó bao gồm Relation Engine giúp xử lý các truy vấn và Storage Engine giúp quản lý các tệp cơ sở dữ liệu, các trang, chỉ mục, ...

SQLOS: Dưới Database Engine là hệ điều hành SQL Server hay SQLOS (SQL Server Operating System). SQLOS giúp quản lý bộ nhớ và I/O, lên lịch nhiệm vụ và khóa dữ liệu để tránh các xung đột xảy ra khi update.

Microsoft cung cấp công cụ, dịch vụ quản lý dữ liệu (data management) và BI.

Với quản lý dữ liệu, SQL Server cung cấp các dịch vụ như: SQL Server Integration Services (SSIS), SQL Server Data Quality Services và SQL Server Master Data Services. Để phát triển cơ sở dữ liệu, ta có thể sử dụng công cụ SQL Server Data. Để quản lý, triển khai và giám sát cơ sở dữ liệu ta có SQL Server Management Studio (SSMS).

Với việc phân tích dữ liệu, SQL Server cung cấp SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS) và The Machine Learning Services (R Services).

Ưu điểm:

- Giao diện SQL Server rất dễ dàng để các Backend Developer tập trung nhiều vào việc chăm sóc dữ liệu hơn là việc nó hoạt động như thế nào.
- Tích hợp với giao diện người dùng: SQL Server được tích hợp với ứng dụng giao diện người dùng, thường là các ứng dụng web để cung cấp cơ chế thay đổi dữ liệu động.
- Vì là sản phẩm của Microsoft nên việc tích hợp framework .Net sẽ dễ dàng hơn do chúng có cùng một tổ chức.
- So với các phương tiện lưu trữ dữ liệu khác như Excel, văn bản,... cơ sở dữ liệu

luôn được ưa chuộng hơn chủ yếu do khả năng lưu trữ lớn, tính bảo mật cao và sức mạnh xử lý dữ liệu.

2.14 Docker

Docker là nền tảng cung cấp cho các công cụ, service để các developers, admins/systems có thể phát triển, thực thi, chạy các ứng dụng với containers. Hay nói một cách khác nó là một nền tảng để cung cấp cách để building, deploy và run các ứng dụng một cách dễ dàng trên nền tảng ảo hóa - "Build once, run anywhere". Hay nói một cách dễ hiểu như sau: Khi chúng ta muốn chạy app thì chúng ta phải thiết lập môi trường chạy cho nó. Thay vì chúng ta sẽ đi cài môi trường chạy cho nó thì chúng ta sẽ chạy docker.



Hình 7 Docker

Docker Images: Là một template chỉ cho phép đọc, ví dụ một image có thể chứa hệ điều hành Ubuntu và web app. Images được dùng để tạo Docker container. Docker cho phép chúng ta build và cập nhật các image có sẵn một cách cơ bản nhất, hoặc bạn có thể download Docker images của người khác.

Docker Containers: Docker container có nét giống với các directory. Một Docker container giữ mọi thứ chúng ta cần để chạy một app. Mỗi container được tạo từ Docker image. Docker container có thể có các trạng thái run, started, stopped, moved và deleted.

Dockerfile: như một script dùng để build các image trong container. Dockerfile bao gồm các câu lệnh liên tiếp nhau được thực hiện tự động trên một image gốc để tạo ra một image mới. Dockerfile giúp đơn giản hóa tiến trình từ lúc bắt đầu đến khi kết thúc.

Ưu điểm của Docker:

Tiện lợi: Khi bạn được phân bổ vào dự án mới, bạn đọc file **README**, thấy dự án hiện tại cần cài rất nhiều thứ, nào là ruby, rồi rails, redis, mysql, nginx, ... mỗi thứ lại phải kèm theo version bao nhiêu, Bạn lên google search cách cài đặt, config

những thứ này sẽ mất một khoảng thời gian khá lâu. Chưa hết, có thể chưa cài xong đã conflic tùm lum, cái nọ xung đột cái kia chẳng hạn, lại còn ảnh hưởng tới những chương trình cũ đã cài đặt trong máy nữa chứ, thôi cài lại luôn cả hệ điều hành cho máy. Thế là mất thời gian, mệt mỏi với những thao tác phụ mà chưa tập trung được vào việc chính. Nhưng với docker mọi thứ đơn giản hơn nhiều, chỉ vài dòng lệnh thôi, bạn sẽ có thể nhanh chóng tạo được môi trường ảo hóa chứa đầy đủ những cài đặt cần thiết cho project rồi.

Dễ dàng sử dụng: Docker rất dễ cho mọi người sử dụng từ developers, systems admins, architects...v...v.. nó tận dụng lợi thế của container để build, test nhanh chóng. Có thể đóng gói ứng dụng trên laptop của họ và chạy trên public cloud, private cloud..v.v... “Build once, run anywhere”.

Tốc độ: Docker container rất nhẹ và nhanh, bạn có thể tạo và chạy docker container trong vài giây so sánh với VMs thì mỗi lần chạy VMs cần rất nhiều thời gian khởi động.

Khả năng di động: môi trường develop được dựng lên bằng docker có thể chuyển từ người này sang người khác mà không làm thay đổi cấu hình ở trong.

Chia sẻ: DockerHub là một “app store for docker images”. Trên DockerHub có hàng ngàn public images được tạo bởi cộng đồng. Dễ dàng tìm thấy những image mà bạn cần và chỉ cần pull về và sử dụng với một số sửa đổi nhỏ.

Môi trường chạy và khả năng mở rộng: Bạn có thể chia nhỏ những chức năng của ứng dụng thành các container riêng lẻ. Ví dụ Database chạy trên một container và Redis cache có thể chạy trên một container khác trong khi ứng dụng Node.js lại chạy trên một cái khác nữa. Với Docker, rất dễ để liên kết các container với nhau để tạo thành một ứng dụng, làm cho nó dễ dàng scale, update các thành phần độc lập với nhau.

Các trường hợp cần sử dụng Docker.

- Khi bạn cần triển khai kiến trúc Microservices.
- Khi ứng dụng bạn cần scale một cách linh hoạt.
- Khi bạn cần build 1 lần và chạy ở nhiều máy khác nhau mà không cần quan tâm đến config.
- Khi bạn cần một cách tiếp cận mới về building, shipping, running ứng dụng một cách nhanh chóng dễ dàng

CHƯƠNG 3 XÁC ĐỊNH NHU CẦU

3.1 Đặc tả nhu cầu

Ứng dụng quản lý nhà hàng	
1. Mục đích	Phát triển ứng dụng quản lý nhà hàng cho phép người dùng thực hiện việc lên món, quản lý bàn ăn, các danh mục, món ăn đến thống kê doanh thu. Giúp nhiều tác vụ được tự động hóa làm giảm khối lượng công việc cho nhân viên từ đó tăng hiệu suất làm việc lên đáng kể
2. Chức năng chính	<p>Đăng nhập: Người dùng có thể đăng nhập vào hệ thống bằng tài khoản và mật khẩu được cung cấp từ trước.</p> <p>Đăng ký: Người dùng có thể đăng ký tài khoản mới bằng cách điền vào mẫu đăng ký với thông tin cần thiết.</p> <p>Xem thông tin tài khoản: Người dùng có thể xem thông tin về tài khoản của mình, bao gồm tên đăng nhập, tên hiển thị và các thông tin khác.</p> <p>Xem lịch sử thanh toán: Người dùng có thể xem lịch sử thanh toán đã thực hiện trước đó trên tài khoản của mình.</p> <p>Thêm, sửa, xóa, xem thức ăn, danh mục: Người dùng có thể thêm thức ăn, danh mục...</p>
3. Yêu cầu kỹ thuật	<ul style="list-style-type: none">- Ứng dụng sử dụng ngôn ngữ lập trình C#- Sử dụng mô hình 3 lớp để phân lớp ứng dụng thành ba phần chính: giao diện người dùng (UI), lớp logic ứng dụng (business logic), và lớp truy cập dữ liệu (data access layer).- Sử dụng SQL để lưu trữ dữ liệu- Sử dụng Jira để quản lý các Issue và Sprint.
4. Yêu cầu hiệu suất	<ul style="list-style-type: none">- Ứng dụng phải có thời gian phản hồi nhanh chóng và ổn định khi người dùng thao tác trên giao diện.

	<ul style="list-style-type: none"> - Các chức năng phải được triển khai một cách mượt mà và không gây ra sự cố cho người dùng.
5. Yêu cầu bảo mật	<ul style="list-style-type: none"> - Mọi thông tin cá nhân của người dùng phải được bảo vệ và không được tiết lộ cho bất kỳ bên thứ ba nào. - Tích hợp các biện pháp bảo mật như xác thực người dùng và mã hóa dữ liệu để đảm bảo tính an toàn của hệ thống.
6. Yêu cầu về kiểm thử	<ul style="list-style-type: none"> - Cần tiến hành kiểm thử tích hợp để đảm bảo tính tương thích giữa các phần của ứng dụng. - Phải thực hiện kiểm thử chức năng để đảm bảo rằng tất cả các chức năng hoạt động đúng như mong đợi. - Cần thực hiện kiểm thử bảo mật để phát hiện và khắc phục các lỗ hổng bảo mật có thể tồn tại.

CHƯƠNG 4 LẬP KẾ HOẠCH SCRUM

4.1 Backlogs

4.1.1 Product backlog

ID	Nội dung
PBI1	Thiết kế định nghĩa một số thành phần quan trọng của ứng dụng: Sql, giao diện, cách kết nối csdl, ...
PBI2	Xây dựng tính năng đăng nhập: Tạo form đăng nhập cho người dùng có thể đăng nhập vào hệ thống
PBI3	Xây dựng các tính năng phục vụ cho nghiệp vụ kinh doanh: Tạo giao diện gọi món thanh toán, in hóa đơn, ...
PBI4	Chức năng thay đổi thông tin cá nhân: Tạo form giúp người dùng có thể thay đổi thông tin cá nhân.
PBI5	Xây dựng các tính năng phục vụ cho nghiệp vụ quản lý: Tạo giao diện giúp người dùng có quyền có thể quản lý được doanh thu, nhân sự, món ăn của nhà hàng,

4.1.2 Sprint backlog

ID	Nội dung
SP1	Sprint 1: Thiết kế định nghĩa một số thành phần quan trọng của ứng dụng: Sql, giao diện, cách kết nối csdl, ...
SP2	Sprint 2: Xây dựng tính năng đăng nhập: Phát triển form đăng nhập cho người dùng có thể đăng nhập vào hệ thống
SP3	Sprint 3: Xây dựng các tính năng phục vụ cho nghiệp vụ kinh doanh: Phát triển giao diện gọi món thanh toán, in hóa đơn, ...
SP4	Sprint 4: Thêm chức năng thay đổi thông tin cá nhân: Phát triển form giúp người dùng có thể thay đổi thông tin cá nhân.
SP5	Sprint 5: Xây dựng các tính năng phục vụ cho nghiệp vụ quản lý: Phát triển giao diện giúp người dùng có quyền có thể quản lý được doanh thu, nhân sự, món ăn của nhà hàng,

4.1.3 Technical backlog

ID	Nội dung
TE1	Tạo kiến trúc dữ liệu cho cơ sở dữ liệu: Thiết kế kiến trúc dữ liệu cho cơ sở dữ liệu để lưu trữ hóa đơn, doanh thu, danh sách món, ...
TE2	Kiểm thử tích hợp: Thực hiện kiểm thử tích hợp đảm bảo tính tương thích và ổn định của hệ thống.

4.1.4 Support backlog

ID	Nội dung
SU1	Sửa lỗi và bảo trì: Theo dõi xử lý các lỗi phát sinh trong quá trình vận hành cung cấp bản vá và bảo trì hệ thống.
SU2	Hỗ trợ người dùng: Cung cấp hỗ trợ và giải đáp thắc mắc cho người dùng liên quan đến tính năng và chức năng của hệ thống.

4.1 Lập bảng kế hoạch Scrum

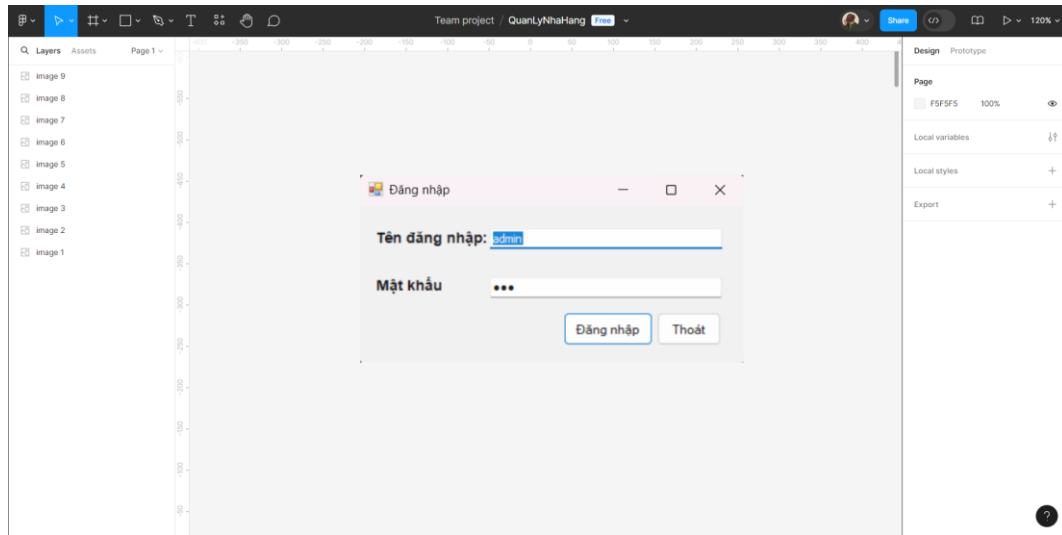
ID	Issue	Person	Sprint	Est Story Points	Start	End
IS1	Thiết kế CSDL viết các trigger và proc	Ninh	Sprint 1	1	1/5/2024	2/5/2024
IS2	Thiết kế giao diện	Minh	Sprint 1	1	2/5/2024	3/5/2024
IS3	Tạo lớp DataProvider	Vinh	Sprint 1	1	3/5/2024	4/5/2024
IS4	Thiết kế patem Singleton	Minh	Sprint 1	1	4/5/2024	5/5/2024
IS5	Thiết kế giao diện đăng nhập	Ninh	Sprint 2	1	5/5/2024	6/5/2024
IS6	Xử lý logic đăng nhập	Vinh	Sprint 2	1	6/5/2024	7/5/2024
IS7	Hiển thị các bàn ăn	Minh	Sprint 3	1	7/5/2024	8/5/2024
IS8	Hiển thị hóa đơn theo bàn	Ninh	Sprint 3	1	8/5/2024	9/5/2024
IS9	Hiển thị tổng tiền	Vinh	Sprint 3	1	9/5/2024	10/5/2024
IS10	Thêm bớt món cho hóa đơn	Ninh	Sprint 3	1	10/5/2024	11/5/2024
IS11	Thanh toán hóa đơn	Minh	Sprint 3	1	11/5/2024	12/5/2024
IS12	Thiết kế giao diện	Vinh	Sprint 4	1	12/5/2024	13/5/2024

	thông tin tài khoản					
IS13	Xử lý cập nhật thay đổi thông tin, xử lý đăng nhập	Ninh	Sprint 4	1	13/5/2024	14/5/2024
IS14	Thiết kế giao diện Admin	Minh	Sprint 5	1	14/5/2024	15/5/2024
IS15	Hiển thị danh thu thời gian	Vinh	Sprint 5	1	16/5/2024	16/5/2024
IS16	Hiển thị thức ăn, thêm, xóa, sửa, xem thức ăn	Minh	Sprint 5	2	16/5/2024	18/5/2024
IS17	Xử lý tìm kiếm gần đúng theo tên thức ăn	Minh	Sprint 5	1	18/5/2024	19/5/2024
IS18	Hiển thị danh mục, thêm, xóa, sửa, xem danh mục	Vinh	Sprint 5	2	19/5/2024	21/5/2024
IS19	Hiển thị các bàn ăn	Ninh	Sprint 5	1	21/5/2024	22/5/2024
IS20	Hiển thị danh sách tài khoản	Minh	Sprint 5	1	22/5/2024	23/5/2024
IS21	Xử lý thêm, xóa, sửa, xem tài khoản	Vinh	Sprint 5	2	23/5/2024	25/5/2024

CHƯƠNG 5 HIỆN THỰC HÓA KẾ HOẠCH

5.1 Thiết kế giao diện

5.1.1 Giao diện đăng nhập

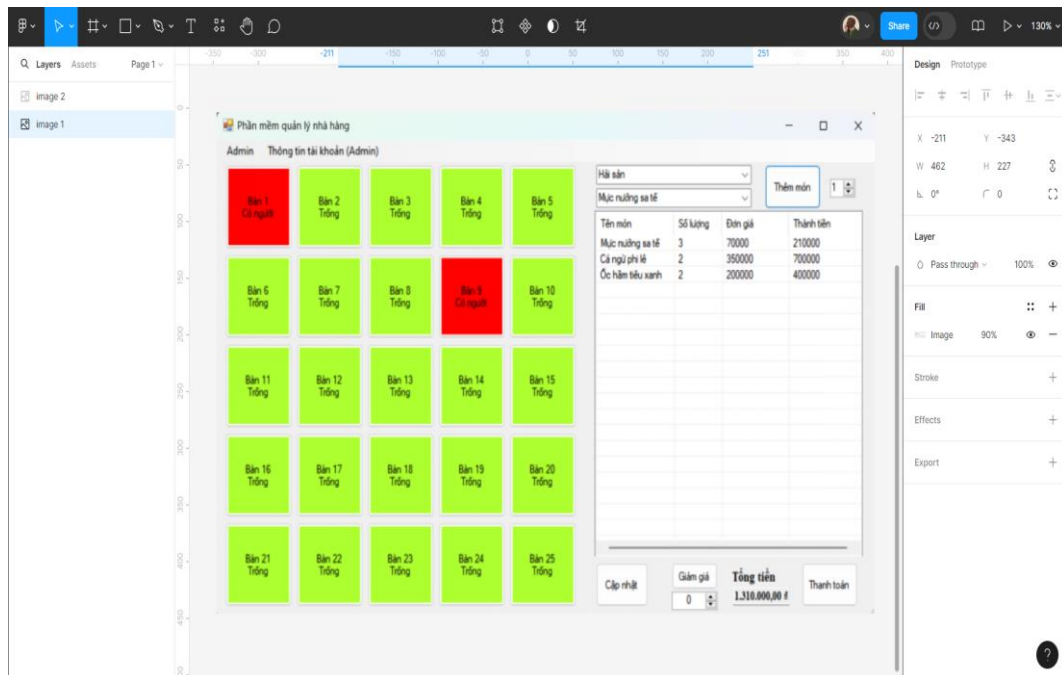


Hình 8 Giao diện đăng nhập

Mô tả Form đăng nhập:

Admin và nhân viên có thể tiến hành đăng nhập vào form với tài khoản và mật khẩu được cung cấp. Khi đăng nhập vào đúng tài khoản và mật khẩu được cung cấp thì được chuyển đến form tiếp theo.

5.1.2 Giao diện gọi món ăn



Hình 9 Giao diện gọi món ăn

Mô tả Form gọi món:

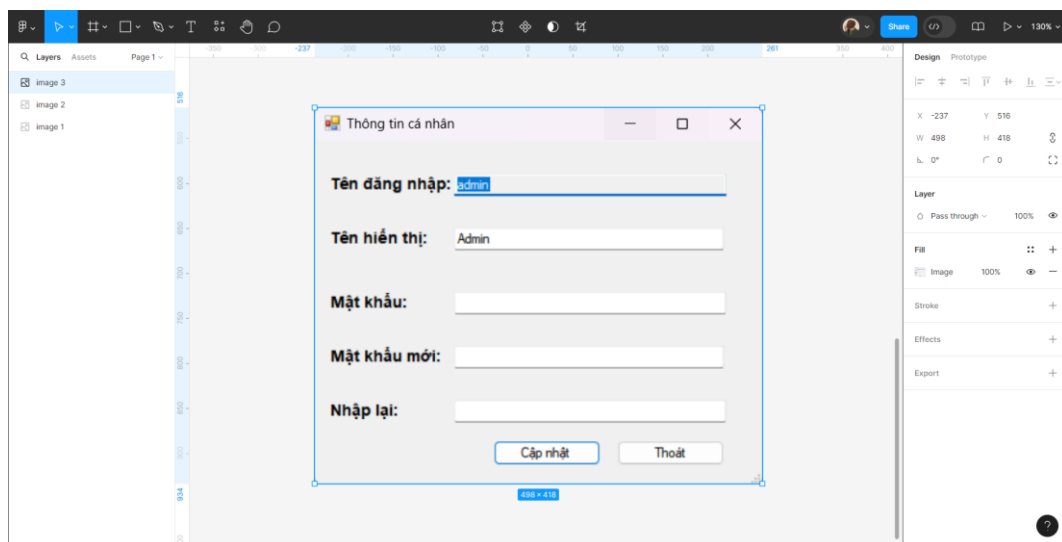
Sau khi đăng nhập thành công, hệ thống sẽ chuyển nhân viên đến giao diện gọi món. Tại đây nhân viên được cung cấp một danh sách toàn bộ danh mục món ăn tại nhà hàng kèm theo danh sách các món ăn thuộc từng danh mục.

Đầu tiên nhân viên chọn danh sách danh mục món ăn tương ứng để chọn món ăn mà khách hàng yêu cầu, sau đó tùy chỉnh số lượng, chọn bàn và ấn “Thêm món” để thêm món đây vào danh sách chờ chuẩn bị mang lên cho khách hàng. Nhân viên có thể lặp lại thao tác nhiều lần để thêm nhiều món vào cùng một bàn.

Sau khi thêm món thành công, tên món được thêm cùng với tổng tiền của từng món được hiển thị ngay bên dưới đồng thời trạng thái của bàn cũng được thay đổi từ “Trống” sang “Có người” (chuyển từ màu xanh sang màu đỏ). Những bàn đã được thêm món, hệ thống sẽ tự động tính toán cho ra tổng tiền của bàn đó và hiển thị cho nhân viên khi chọn vào bàn tương ứng. Trong trường hợp khách hàng gọi thêm món, hệ thống sẽ tự động cộng vào và cập nhật tổng tiền ngay lập tức.

Khi khách hàng ra về nhân viên chỉ cần báo với khách số tiền hiển thị theo bàn tương ứng (có thể tùy chọn giảm giá) và ấn “Thanh toán” khi khách hàng đã thanh toán, trạng thái của bàn sẽ tự trở về trạng thái ban đầu.

5.1.3 Giao diện thông tin cá nhân

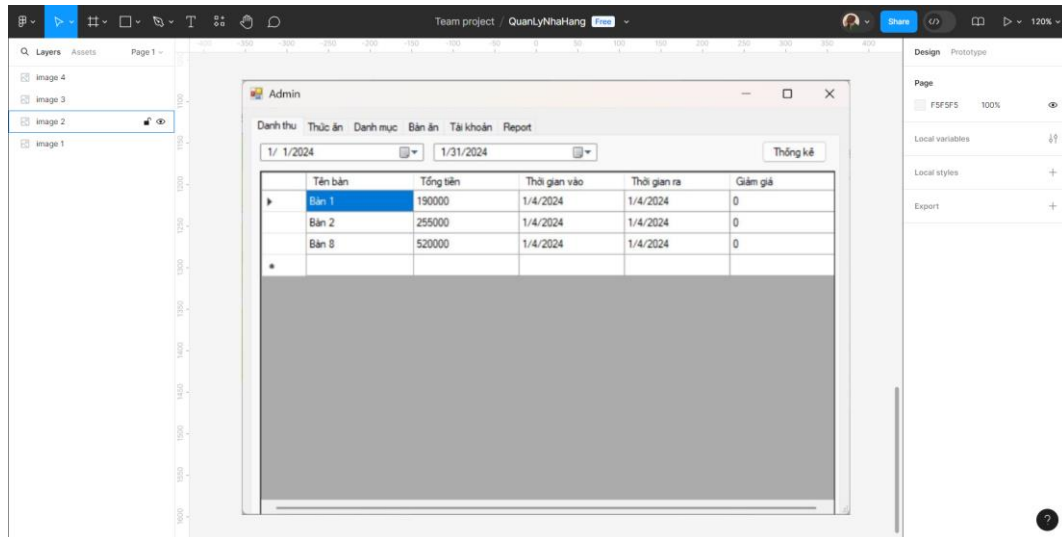


Hình 10 Giao diện thông tin cá nhân

Mô tả Form thông tin cá nhân:

Form này có chức năng đổi tên hiển thị và mật khẩu. Chỉ cần nhập đầy đủ và chính xác các thông tin vào các ô tương ứng sau đó ấn “Cập nhật”, thông tin cá nhân đã được thay đổi

5.1.4 Giao diện doanh thu



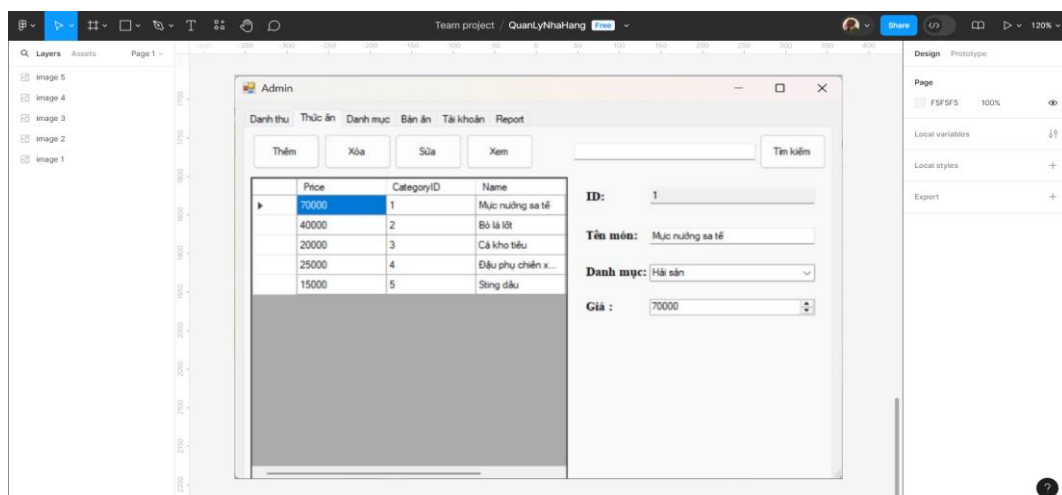
Hình 11 Giao diện doanh thu

Mô tả form doanh thu:

Các Form bao gồm giao diện doanh thu, giao diện món ăn, giao diện danh mục, giao diện bàn ăn, giao diện tài khoản, giao diện report đều thuộc chức năng nâng cao nên chỉ tài khoản có quyền admin (hay loại = 1) mới có thể truy cập được.

Form này cho phép tùy chọn xem các hóa đơn đã thanh toán của nhà hàng theo ngày bắt đầu và kết thúc tương ứng, giúp người quản lý theo dõi tình hình kinh doanh của nhà hàng một cách dễ dàng.

5.1.5 Giao diện thức ăn

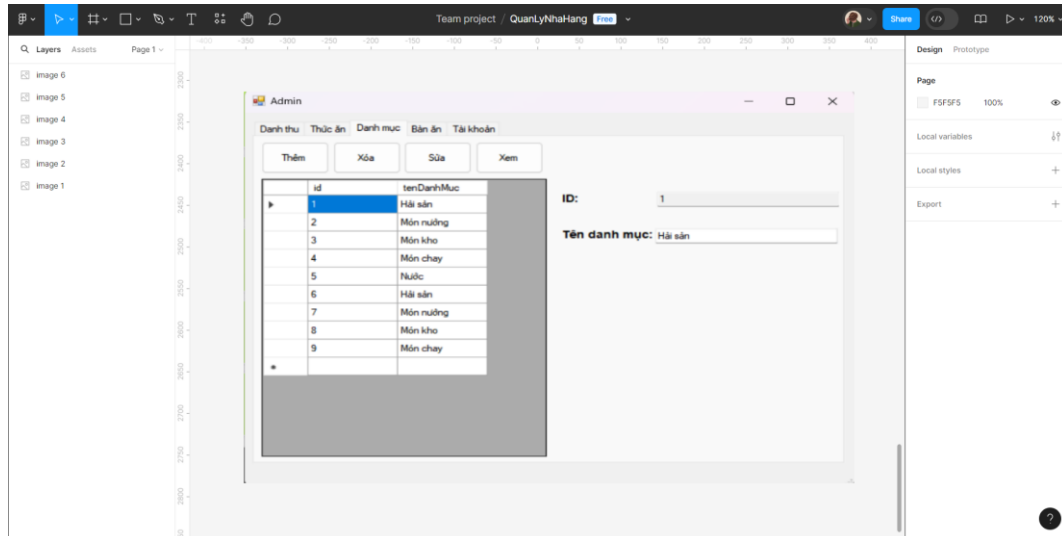


Hình 12 Giao diện thức ăn

Mô tả form thức ăn:

Form có chức năng hiển thị danh sách thông tin từng món ăn bao gồm tên món ăn, giá tiền, cho biết thuộc danh mục nào. Ngoài ra còn cho phép thêm, sửa và xóa giúp việc thay đổi menu của nhà hàng trở nên tiện lợi hơn.

5.1.6 Giao diện danh mục

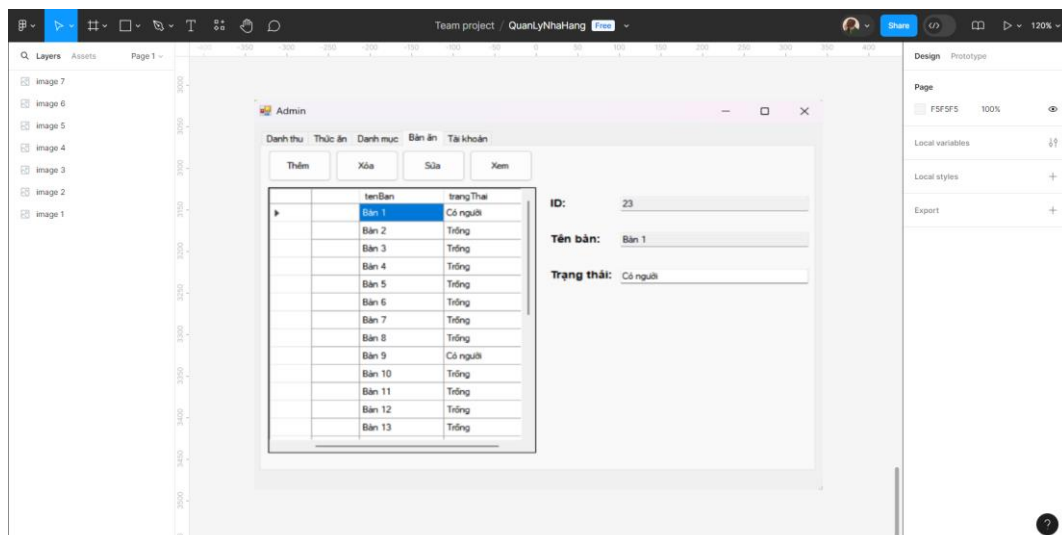


Hình 13 Giao diện danh mục

Mô tả form danh mục:

Tương tự như Form món ăn, Form danh mục hiển thị thông tin và danh sách danh mục, cung cấp khả năng thêm, sửa và xóa danh mục.

5.1.7 Giao diện bàn ăn

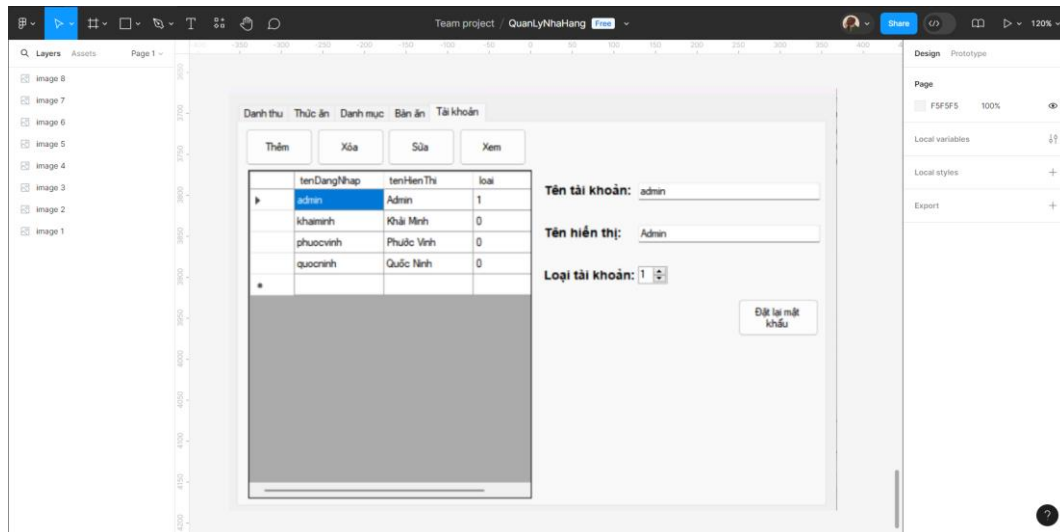


Hình 14 Giao diện bàn ăn

Mô tả form bàn ăn:

Hiển thị danh sách bàn ăn, trạng thái của từng bàn (Trống hoặc Có người).

5.1.8 Giao diện tài khoản

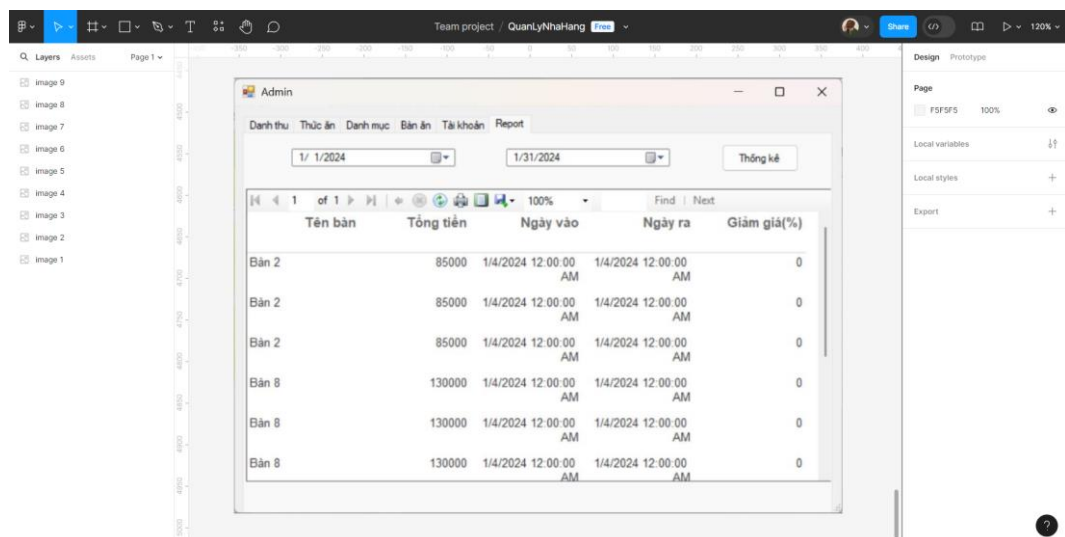


Hình 15 Giao diện tài khoản

Mô tả form tài khoản:

Hiện thị danh sách tài khoản được cho phép, loại tài khoản (quyền). Thêm sửa xóa các tài khoản.

5.1.9 Giao diện Report



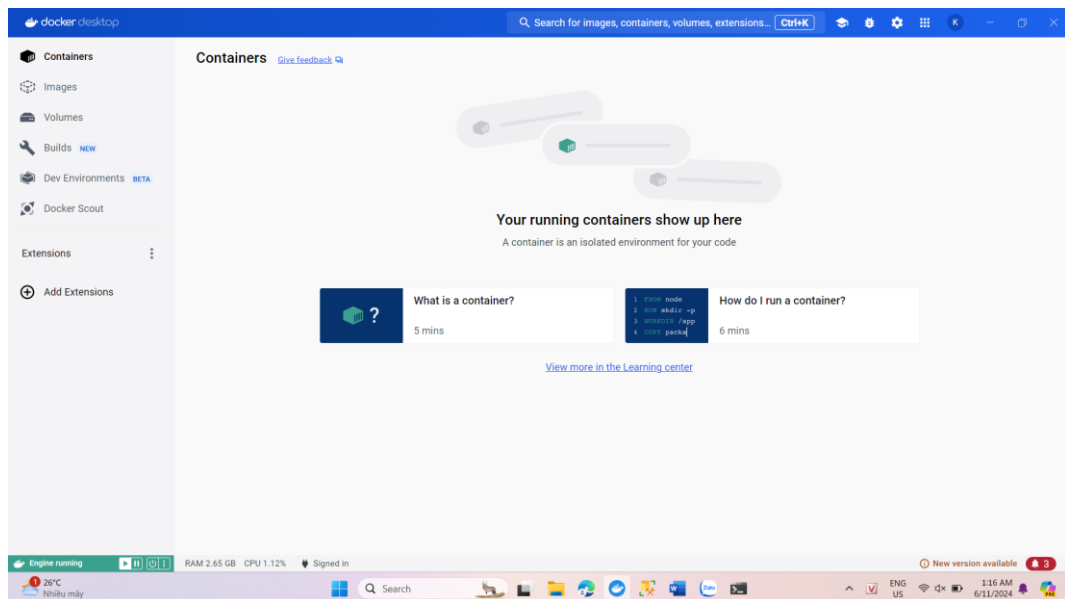
Hình 16 Giao diện Report

Mô tả form Report:

Sử dụng report để in danh sách hóa đơn theo ngày bắt đầu (mặc định là đầu tháng) và ngày kết thúc (mặc định là cuối tháng).

5.2 Triển khai SQL Server với Docker

Bước 1 : Mở Docker Desktop



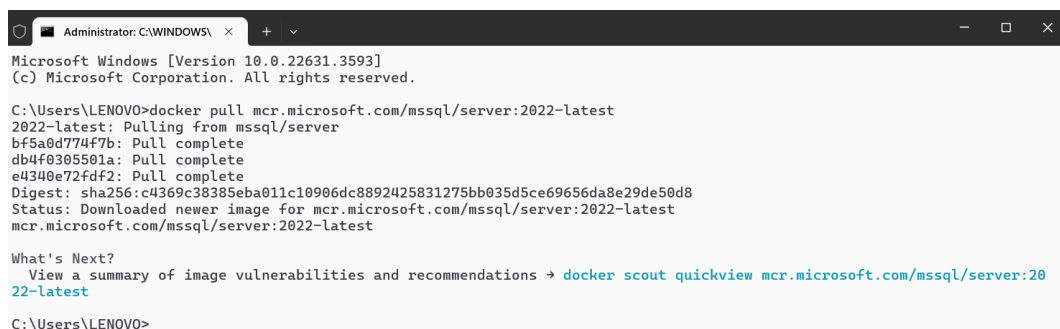
Hình 17 Docker Desktop

Bước 2: Mở Command Prompt (CMD) trên Windows



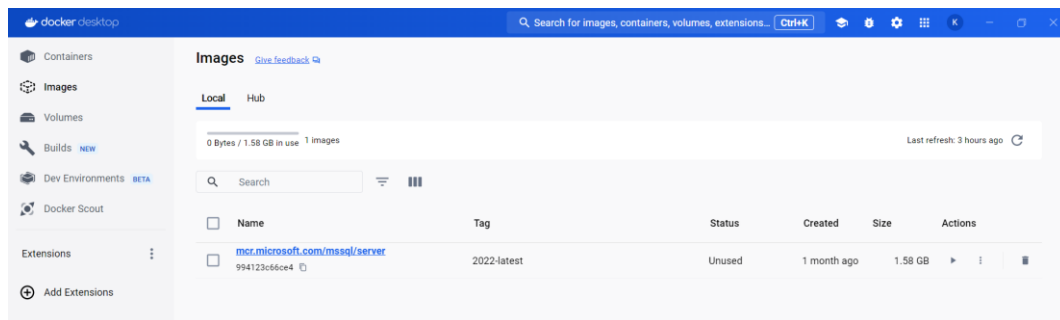
Hình 18 Command Prompt (CMD)

Bước 3: Chạy lệnh docker pull mcr.microsoft.com/mssql/server:2022-latest



Hình 19 Chạy lệnh trên Command Prompt (CMD)

Sau khi chạy xong sẽ tự động pull về 1 images



Hình 20 Docker Images sau khi chạy lệnh

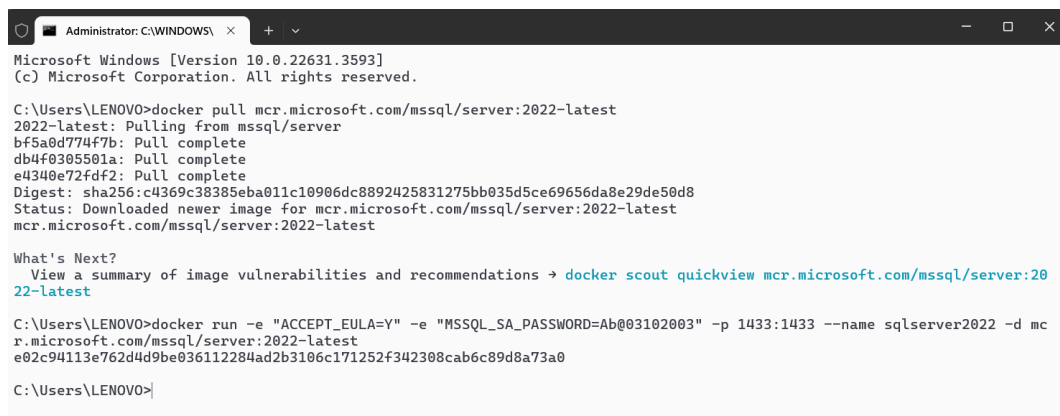
Bước 4 : Chạy lệnh `docker run -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=Ab@03102003" -p 1433:1433 --name sqlserver2022 -d mcr.microsoft.com/mssql/server:2022-latest`

docker run: Lệnh này được sử dụng để tạo và chạy một container mới dựa trên một image Docker.

-e "ACCEPT_EULA=Y": Tham số này thiết lập một biến môi trường trong container. ACCEPT_EULA=Y đồng ý với thỏa thuận cấp phép người dùng cuối của Microsoft (EULA).

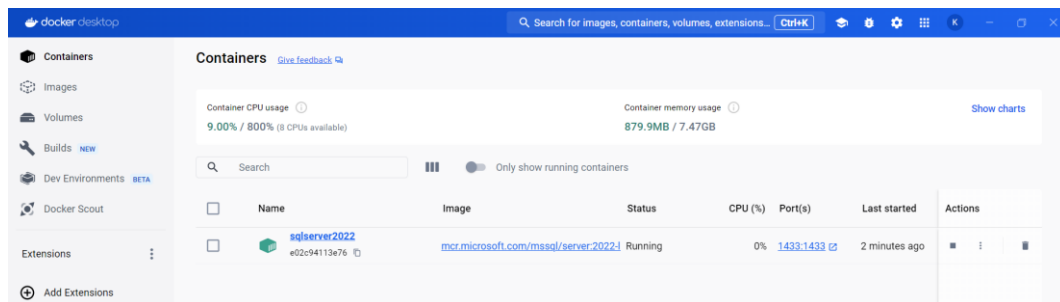
-e "MSSQL_SA_PASSWORD=Ab@03102003": Thiết lập một biến môi trường khác để đặt mật khẩu cho tài khoản sa (System Administrator) trong SQL Server.

-p 1433:1433: Chuyển tiếp cổng từ máy chủ Docker sang container.



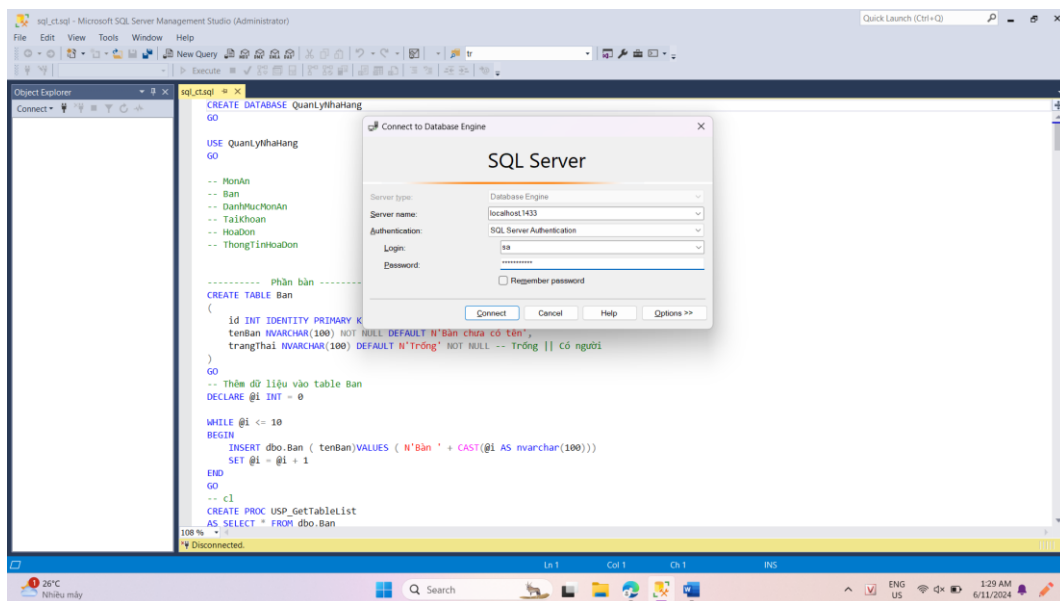
Hình 21 Chạy lệnh trên Command Prompt (CMD)

Sau khi chạy xong lệnh sẽ tạo ra Container



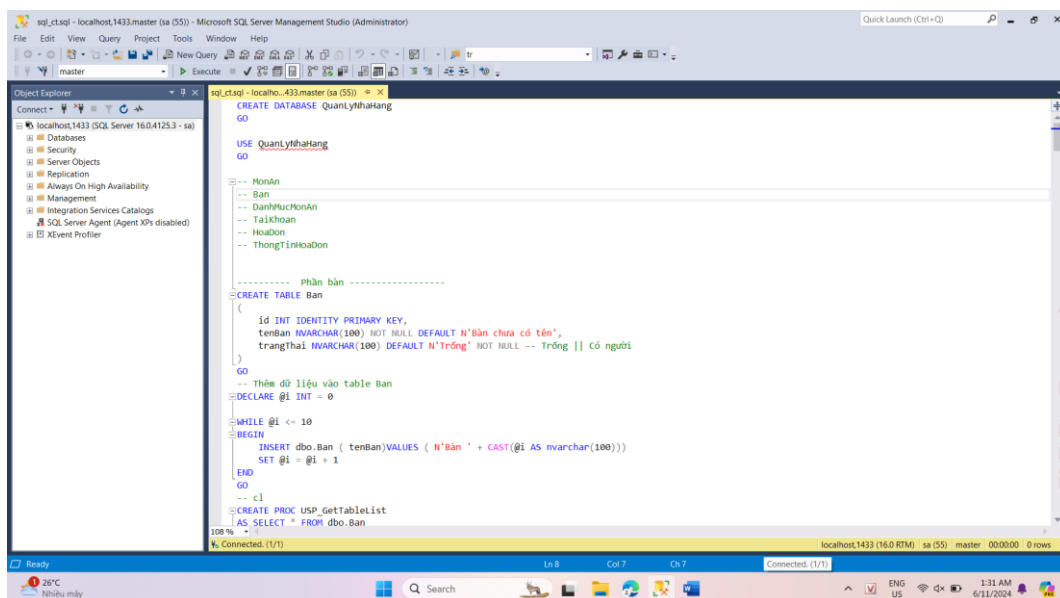
Hình 22 Docker Container

Bước 5 : Đăng nhập vào SQL Server



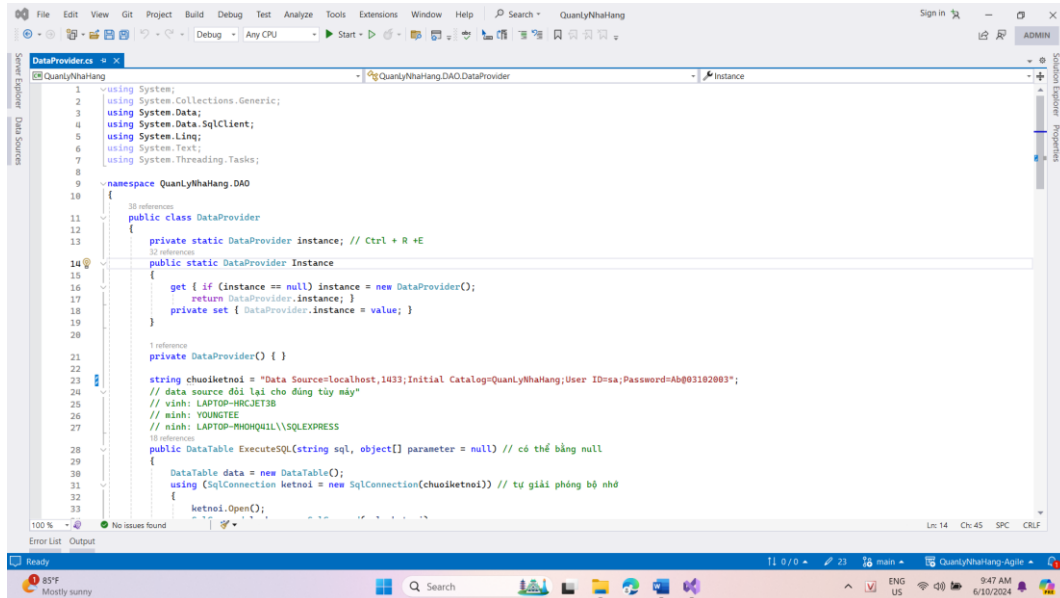
Hình 23 Đăng nhập SQL Server

Khi bấm Connect đăng nhập thành công



Hình 24 Đăng nhập SQL thành công

5.3 Mã nguồn



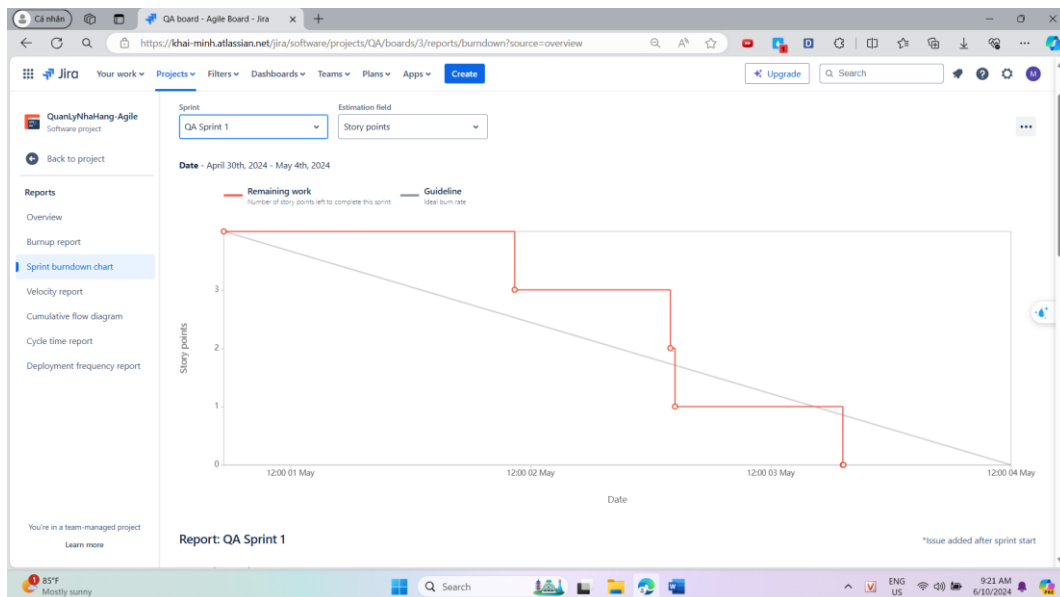
```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.SqlClient;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace QuanLyNhaHang.DAO
10 {
11     public class DataProvider
12     {
13         private static DataProvider instance; // Ctrl + R + E
14         public static DataProvider Instance
15         {
16             get { if (instance == null) instance = new DataProvider();
17                 return DataProvider.instance; }
18             private set { DataProvider.instance = value; }
19         }
20
21         private DataProvider() { }
22
23         string chuoiKetnoi = "Data Source=localhost,1433;Initial Catalog=QuanLyNhaHang;User ID=sa;Password=Ab@03102003";
24         // data source đổi lại cho đúng tùy máy
25         // vnh: LAPTOP-HRCJET3B
26         // nhinh: YOUNGTEE
27         // nhinh: LAPTOP-HMHQ4LL\SQLSERVER
28         public DataTable ExecuteSQL(string sql, object[] parameter = null) // có thể bằng null
29         {
30             DataTable data = new DataTable();
31             using (SqlConnection ketnoi = new SqlConnection(chuoiKetnoi)) // tự giải phóng bộ nhớ
32             {
33                 ketnoi.Open();
34             }
35         }
36     }
37 }
```

Hình 25 Mã nguồn ứng dụng

Link GitHub : <https://github.com/khaiminh03/QuanLyNhaHang-Agile.git>

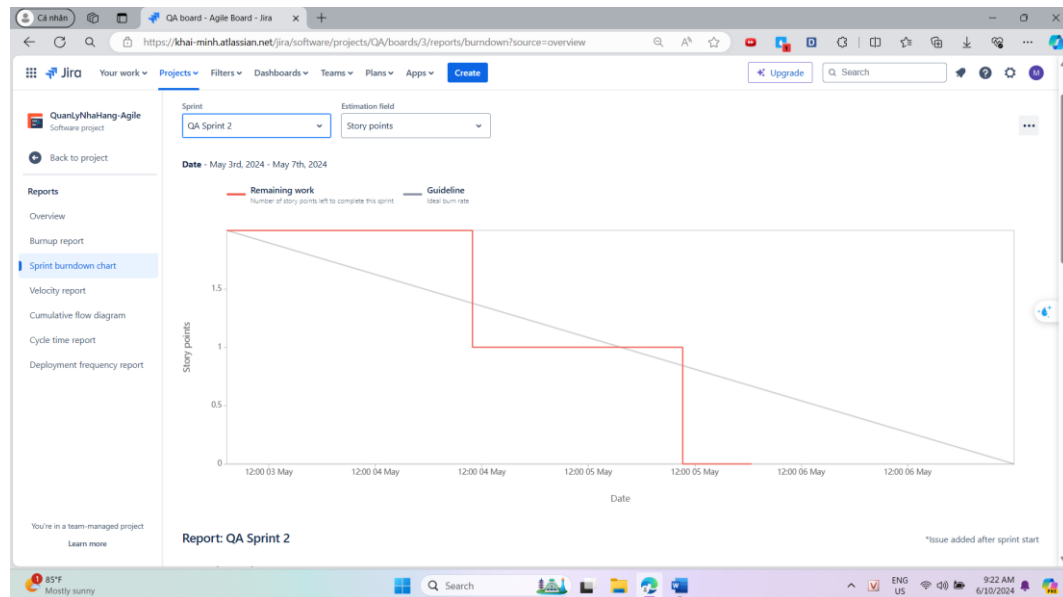
5.4 Biểu đồ Sprint Burndown chart

Sprint 1



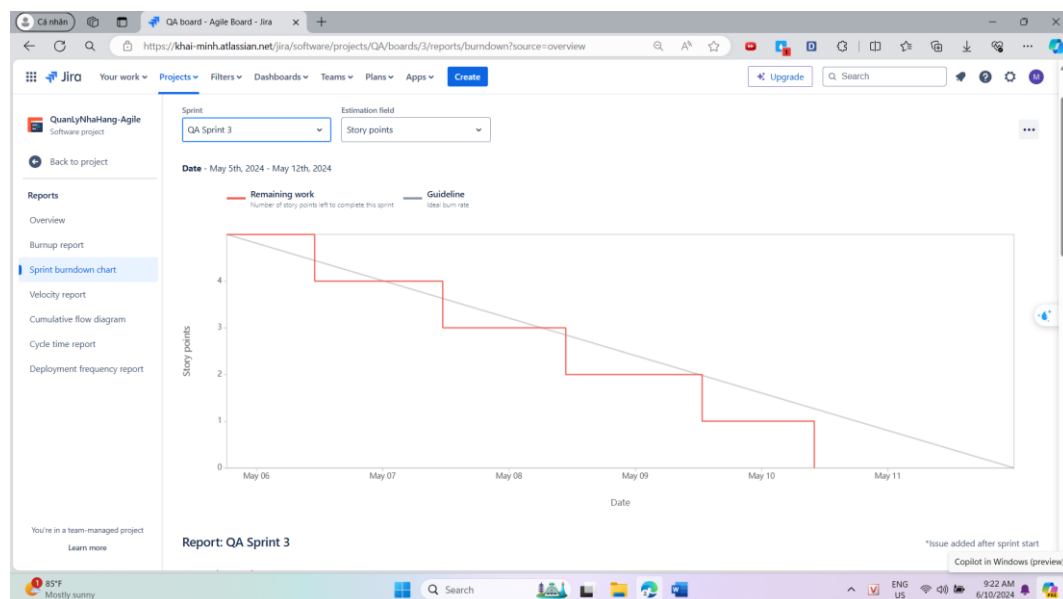
Hình 26 Biểu đồ Burndown chart Sprint 1

Sprint 2



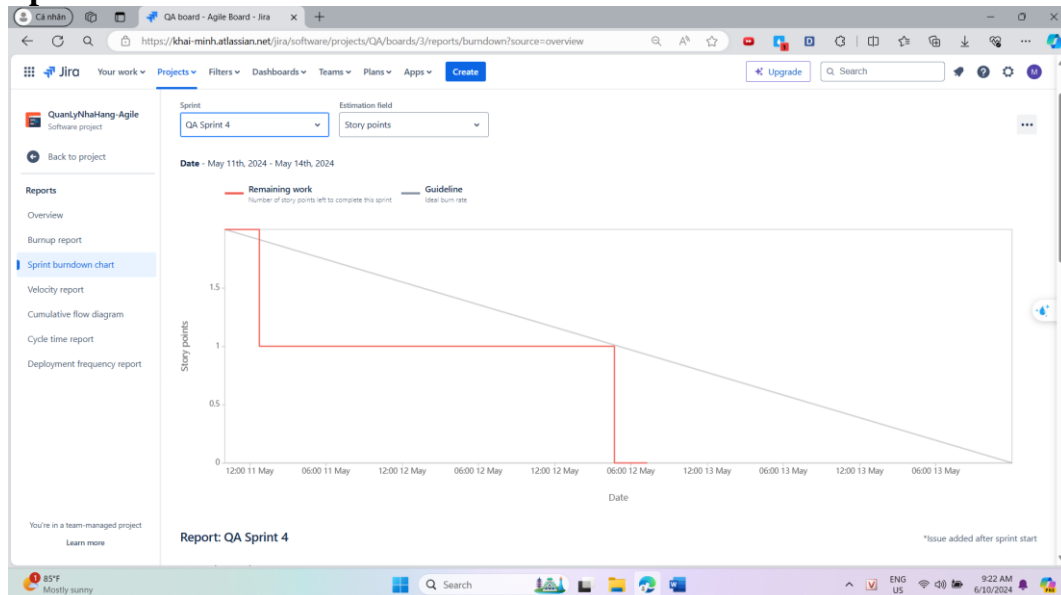
Hình 27 Biểu đồ Burndown chart Sprint 2

Sprint 3



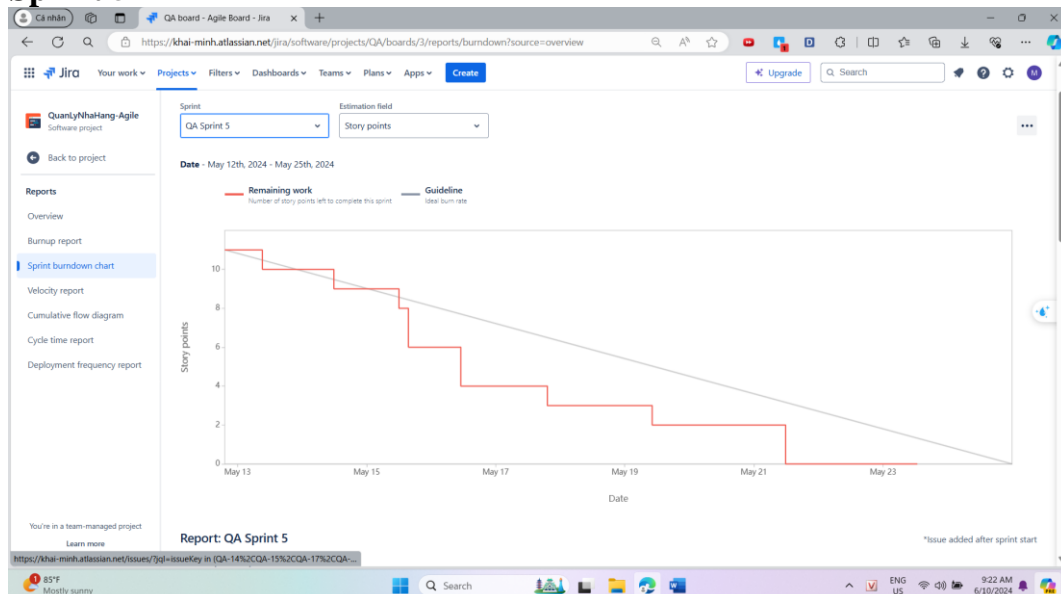
Hình 28 Biểu đồ Burndown chart Sprint 3

Sprint 4



Hình 29 Biểu đồ Burndown chart Sprint 4

Sprint 5



Hình 30 Biểu đồ Burndown chart Sprint 5

CHƯƠNG 6 KẾT LUẬN

3.1 Kết quả đạt được

- Lập được kế hoạch scrum, chia thành các sprint.
- Chia nhánh ứng với từng thành viên trong nhóm.
- Các thành viên push code theo đúng nhánh được phân công.
- Push code lên git và done jira theo lịch trong bảng kế hoạch scrum.
- Hoàn thành các sprint đúng thời gian.
- Triển khai và kết nối đến SQL Server bằng docker.

3.2 Hạn chế

Điểm đánh giá point còn chung chung chưa sát với độ khó thực tế của từng chức năng.

3.3 Hướng phát triển

Phát triển ứng dụng di động đi kèm cho nhân viên nhà hàng để quản lý đơn hàng, đặt bàn và thanh toán từ xa.

Sử dụng phương pháp Agile và phần mềm Jira để lên kế hoạch, phát triển và quản lý quy trình mở rộng quy mô ứng dụng.

TÀI LIỆU THAM KHẢO

- [1] [Agile là gì? Hiểu về mô hình Agile để triển khai hiệu quả \(base.vn\)](#)
- [2] [Agile là gì? Scrum là gì? Các công cụ quản lý dự án theo Agile \(topdev.vn\)](#)
- [3] <https://viblo.asia/p/tim-hieu-ve-mo-hinh-3-lop-3-layer-Qbq5Qmyz5D8>
- [4] <https://plctech.com.vn/winform-la-gi/>
- [5] <https://codegym.vn/blog/c-la-gi-tim-hieu-ve-ngon-ngu-lap-trinh-c/>
- [6] <https://topdev.vn/blog/sql-server-la-gi/>
- [7] [02-Làm việc với Docker Image,Container,kết nối SQL Server - YouTube](#)
- [8] <https://learn.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-ver16&tabs=cli&pivots=cs1-bash>
- [9] https://hub.docker.com/_/microsoft-mssql-server?zarsrc=410&utm_source=zalo&utm_medium=zalo&utm_campaign=zalo