**Bộ Giáo Dục Và Đào Tạo**

**Trường Đại Học Ngoại Ngữ - Tin Học Thành Phố Hồ Chí Minh**

**Khoa Công Nghệ Thông Tin**

A yellow oval with red text

Description automatically generated

**MÔN HỌC: BẢO ĐẢM CHẤT LƯỢNG PHẦN MỀM**

**ĐỀ TÀI: WEBSITE BÁN VÉ MÁY BAY**

**Giáo viên hướng dẫn:** ThS. Tiếu Phùng Mai Sương

**Thành viên:**

1. Trương Tấn Nhã – 20DH111319
2. Lê Thị Thu Thảo – 20DH111870
3. Trần Nguyễn Kim Ngân – 20DH110947

*TP. Hồ Chí Minh, tháng 12 năm 2023*

LỜI CẢM ƠN

Để hoàn thiện được bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến ThS. Tiếu Phùng Mai Sương – người cô đã hỗ trợ cho chúng em rất nhiều trong quá trình làm bài. Cô đã đưa ra rất nhiều ý kiến để chúng em khắc phục những thiếu sót và hoàn thành tốt bài làm cũng như là nộp đúng thời hạn đã được đề ra trước đó.

Thêm vào đó, trong quá trình học tập và trao đổi dưới sự hướng dẫn của cô, chúng em không chỉ có được rất nhiều kiến thức bổ ích, mà còn được truyền thêm sự say mê và thích thú đối với bộ môn Bảo đảm chất lượng phần mềm.

Mặc dù đã rất cố gắng hoàn thiện bài với tất cả sự nỗ lực của nhóm thì đây vẫn là một môn học có tính thực tế rất cao, ngoài ra còn nhiều hạn chế về mặt thời gian cũng như lượng kiến thức, bài làm của nhóm em chắc chắn không thể tránh khỏi những sai sót nên rất mong sẽ nhận được những ý kiến đóng góp quý báu của cô để báo cáo của chúng em hoàn hiện hơn và có thể làm tốt hơn nữa trong những lần về sau.

Một lần nữa chúng em xin chân thành cảm ơn cô và luôn mong nhận được sự đóng góp của cô.

Cuối lời, chúng em xin kính chúc cô dồi dào sức khỏe, hạnh phúc để tiếp tục dìu dắt nhiều thế hệ học trò đến với những bến bờ tri thức.

Trân trọng!

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc153599956)

[MỤC LỤC ii](#_Toc153599957)

[DANH MỤC VIẾT TẮT iv](#_Toc153599958)

[DANH MỤC HÌNH ẢNH v](#_Toc153599959)

[DANH MỤC BẢNG BIỂU vi](#_Toc153599960)

[BẢNG PHÂN CÔNG CÔNG VIỆC vii](#_Toc153599961)

[I. GIỚI THIỆU CHUNG 1](#_Toc153599962)

[II. LÝ THUYẾT 2](#_Toc153599963)

[1. Bảo đảm chất lượng phần mềm 2](#_Toc153599964)

[2. Automation Testing Tools 2](#_Toc153599965)

[2.1. Giới thiệu về Automation Testing 2](#_Toc153599966)

[2.2. Giới thiệu chung về Telerik Test Studio 3](#_Toc153599967)

[2.3. Các tính năng của Telerik Test Studio 3](#_Toc153599968)

[2.4. Ưu điểm của Telerik Test Studio 4](#_Toc153599969)

[2.5. Nhược điểm của Telerik Test Studio 4](#_Toc153599970)

[III. KIỂM THỬ WEBSITE BÁN VÉ MÁY BAY 5](#_Toc153599971)

[1. Đặc tả hệ các chức năng hệ thống (System Functions) 5](#_Toc153599972)

[1.1. Kiểm thử các chức năng trong phần mềm 5](#_Toc153599973)

[1.2. Các Use Case 5](#_Toc153599974)

[1.3. Đặc tả các chức năng mà Case Actor sử dụng 6](#_Toc153599975)

[1.4. Sơ đồ Use Case tổng quát 8](#_Toc153599976)

[1.5. Thiết kế cơ sở dữ liệu 11](#_Toc153599977)

[1.6. Sơ đồ Sequence 12](#_Toc153599978)

[1.7. Sơ đồ Activity 13](#_Toc153599979)

[2. Xác định mục tiêu, các chuẩn chất lượng cần đạt của hệ thống 15](#_Toc153599980)

[2.1. Mục tiêu 15](#_Toc153599981)

[2.2. Các chuẩn cần đạt của hệ thống 15](#_Toc153599982)

[3. Phạm vi kiểm thử của hệ thống (Scope) 16](#_Toc153599983)

[3.1. Các chức năng cần kiểm thử 16](#_Toc153599984)

[3.2. Các chức năng không cần kiểm thử 16](#_Toc153599985)

[4. Thiết kế Test Cases/ Test Checklists 17](#_Toc153599986)

[4.1. Kiểm thử chức năng 17](#_Toc153599987)

[4.2. Kiểm thử phi chức năng 43](#_Toc153599988)

[4.3. Kiểm thử các yếu tố khác 45](#_Toc153599989)

[5. Thực thi kiểm thử trên Telerik và báo cáo Buglist 58](#_Toc153599990)

[5.1. Kết quả sau khi đã record 58](#_Toc153599991)

[5.2. Kết quả sau khi thực thi Automation Test 60](#_Toc153599992)

[5.3. Kết quả trả về từ file log 61](#_Toc153599993)

[5.4. Thực thi chuyển Step thành code C# 62](#_Toc153599994)

[5.5. Thêm danh sách các Step vào test list 63](#_Toc153599995)

[5.6. Thông báo kết quả sau khi chạy test list 64](#_Toc153599996)

[5.7. Báo cáo thống kê kết quả của các step sau khi chạy test list 65](#_Toc153599997)

[5.8. Thực hiện tìm kiếm/ nhận diện Element có trong web 66](#_Toc153599998)

[5.9. Thực hiện test performance dựa trên test Script 67](#_Toc153599999)

[5.10. Hiển thị thông tin sau khi test Performance 68](#_Toc153600000)

[5.11. Hiển thị thông tin chi tiết Performance 69](#_Toc153600001)

[5.12. So sánh Performance từ các trình duyệt khác nhau 70](#_Toc153600002)

[5.13. Thực thi test Check List 71](#_Toc153600003)

[5.14. Lên lịch cho Check List 72](#_Toc153600004)

[5.15. Xuất kết quả báo cáo sang file Excel sau khi test list 73](#_Toc153600005)

[6. Test Report 75](#_Toc153600006)

[IV. KẾT LUẬN 76](#_Toc153600007)

[1. Ưu điểm 76](#_Toc153600008)

[2. Nhược điểm 76](#_Toc153600009)

[3. Giải pháp mở rộng (nâng cao) 76](#_Toc153600010)

[4. Kết quả đạt được 77](#_Toc153600011)

[5. Kết quả chưa đạt được 77](#_Toc153600012)

[6. Hướng phát triển 77](#_Toc153600013)

[TÀI LIỆU THAM KHẢO 78](#_Toc153600014)

DANH MỤC VIẾT TẮT

DANH MỤC HÌNH ẢNH

[Hình 1. 1. Use Case: Tổng quát 8](#_Toc153599803)

[Hình 1. 2. Use Case: Quản trị hệ thống 8](#_Toc153599804)

[Hình 1. 3. Use Case: Quản lý tài chính 9](#_Toc153599805)

[Hình 1. 4. Use Case: Quản lý chuyến bay 9](#_Toc153599806)

[Hình 1. 5. Use Case: Quản lý vé máy bay 10](#_Toc153599807)

[Hình 1. 6. ERD Diagram 11](#_Toc153599808)

[Hình 1. 7. Quy trình hệ thống bán vé máy bay 12](#_Toc153599809)

[Hình 1. 8. Quy trình đăng nhập 13](#_Toc153599810)

[Hình 1. 9. Quy trình đặt vé 14](#_Toc153599811)

[Hình 5. 1. Sau khi record 58](#_Toc153599654)

[Hình 5. 2. Sau khi chạy automation test 60](#_Toc153599655)

[Hình 5. 3. File log 61](#_Toc153599656)

[Hình 5. 4. Chuyển Step thành C# 62](#_Toc153599657)

[Hình 5. 5. Thêm Step vào Test List 63](#_Toc153599658)

[Hình 5. 6. Sau khi chạy Test List 64](#_Toc153599659)

[Hình 5. 7. Thống kê 65](#_Toc153599660)

[Hình 5. 8. Tìm kiếm/ nhận diện Element 66](#_Toc153599661)

[Hình 5. 9. Test Performance 67](#_Toc153599662)

[Hình 5. 10. Hiển thị thông tin test 68](#_Toc153599663)

[Hình 5. 11. Hiển thị chi tiết 69](#_Toc153599664)

[Hình 5. 12. So sánh khác nhau 70](#_Toc153599665)

[Hình 5. 13. Test Check List 71](#_Toc153599666)

[Hình 5. 14. Lên lịch test 72](#_Toc153599667)

[Hình 5. 15. Xuất ra Excel 73](#_Toc153599668)

DANH MỤC BẢNG BIỂU

[Bảng 1. 1. Bảng đặc tả chức năng 6](#_Toc153599874)

BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ tên** | **Công việc** |
| 1 | Trương Tấn Nhã | Login – Register |
| Home Page |
| Person Detail |
| 2 | Lê Thị Thu Thảo | Ticket – History |
| Flight Route |
| Viết báo cáo |
| 3 | Trần Nguyễn Kim Ngân | Schedule A Flight |
| OCR |
| Update Excel |

1. GIỚI THIỆU CHUNG

Công nghệ thông tin đang ngày càng phát triển, kéo theo đó là hệ thống mạng và các phần mềm cũng gia tăng về số lượng theo quy mô rộng và chất lượng phần mềm theo chiều sâu. Nhưng cũng từ đó đã nảy sinh ra nhiều vấn đề về lỗi hỏng hóc phần mềm không đáng có gây ra các ảnh hưởng nghiêm trọng đến xã hội, kinh tế, …

Những lỗi này có thể do tự bản thân phần mềm bị hỏng do không được kiểm duyệt kĩ lưỡng trước khi đưa cho người dùng cuối hay cũng có thể do có người cố tình phá hoại nhằm đánh cắp thông tin cá nhân như số tài khoản ngân hàng, số điện thoại, danh bạ hay tin nhắn, …

Những vấn đề nan giải và cấp thiết này ngày càng có xu hướng mở rộng trong các năm gần đây, điển hình như ở Hàn Quốc khi họ trải cuộc tấn công internet vào tháng 1 năm 2014, dữ liệu lên đến 100 triệu thẻ tín dụng đã bị các tin tặc lấy đi, gây ra thiệt hại nặng nề cho kinh tế hay như càng có nhiều loại virus phá hoại mới xuất hiện, tấn công vào các lỗ hỏng bảo mật phần mềm làm tê liệt nhiều hệ thống phần mềm và phần cứng. Từ đây ta dễ dàng nhận ra là mặc dù phần mềm phát triển ngày càng phức tạp nhưng vấn đề chất lượng vẫn là một dấu hỏi lớn cần xem xét cẩn thận.

Do đó yêu cầu đặt ra là cần có công tác kiểm thử phần mềm thật kĩ lưỡng nhằm ngăn chặn các lỗi hay hỏng hóc còn tiềm tàng bên trong phần mềm mà ta chưa kịp nhận ra và đó cũng chính là lí do nhóm em lựa chọn đề tài kiểm thử Website bán vé máy bay để nghiên cứu, tìm hiểu và đề ra các giải pháp mới nhằm cải tiến các quy trình kiểm thử sao cho đảm bảo chất lượng nhất.

1. LÝ THUYẾT
   1. Bảo đảm chất lượng phần mềm

Là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm.

Trong kỹ thuật kiểm thử không chỉ giới hạn ở việc thực hiện một chương trình hoặc ứng dụng với mục đích đi tìm các lỗi phần mềm (bao gồm các lỗi và các thiếu sót) mà còn là một quá trình phê chuẩn và xác minh một chương trình máy tính/ ứng dụng/ sản phẩm nhằm:

* Đáp ứng được mọi yêu cầu hướng dẫn khi thiết kế và phát triển phần mềm.
* Thực hiện công việc đúng như kỳ vọng.
* Có thể triển khai được với những đặc tính tương tự.
* Và đáp ứng được mọi nhu cầu của các bên liên quan.

Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất nhưng trong Agile (là một tập hợp các phương pháp phát triển phần mềm linh hoạt dựa trên việc lặp đi lặp lại và gia tăng giá trị) thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.

* 1. Automation Testing Tools
     1. Giới thiệu về Automation Testing

*Automation Testing* được thực hiển bởi công cụ kiểm thử để thực hiện các bộ testcase. Phần mềm tự động có thể nhập dữ liệu test vào hệ thống được thử nghiệm, so sánh kết quả mong muốn và kết quả thực tế, đưa ra báo cáo chi tiết. Kiểm thử tự động cũng đòi hỏi đầu tư đáng kể vào chi phí và nguồn lực. Chu kỳ phát triển nối tiếp sẽ yêu cầu thực hiện các bộ test lặp lại nhiều lần. Việc sử dụng công cụ kiểm thử tự động, nó có thể ghi lại bộ test này và chạy lại khi được yêu cầu. Khi bộ test được chạy tự động, sự can thiệp của con người là không cần thiết. Mục đích của tự động là làm giảm số lượng testcase được chạy thủ công và không hẳn sẽ loại bỏ kiểm thử thủ công.

Kiểm thử phần mềm tự động *rất quan trọng* vì:

* Việc kiểm thử thủ công tất cả các quy trình, các lĩnh vực, các kịch bản dường như rất tốn thời gian và chi phí.
* Thật khó để kiểm thử thủ công các trang web đa ngôn ngữ.
* Kiểm thử tự động không yêu cầu có sự can thiệp từ người kiểm thử, có thể chạy qua đêm mà không cần giám sát.
* Kiểm thử tự động giúp tăng tốc độ thực thi kiểm thử.
* Kiểm thử tự động giúp tăng độ bao phủ.
* Kiểm thử thủ công có thể trở nên nhàm chán và dễ bị lỗi.

Các trường hợp kiểm thử tự động được lựa chọn *sử dụng theo các tiêu chí* sau để tăng tự động hóa:

* Rủi ro cao – các trường hợp kiểm tra quan trọng.
* Các trường hợp được thực thi lại nhiều lần.
* Các trường hợp rất tẻ nhạt hoặc khó để kiểm tra thủ công.
* Các trường hợp tiêu tốn nhiều thời gian.
  + 1. Giới thiệu chung về Telerik Test Studio

Telerik Test Studio là một công cụ tự động hóa kiểm tra phổ biến được sử dụng rộng rãi trong lĩnh vực phát triển phần mềm. Công cụ này được phát triển bởi Telerik, một công ty chuyên về phát triển công cụ và giải pháp phát triển ứng dụng.

Telerik Test Studio được thiết kế để hỗ trợ tự động hóa việc kiểm tra chất lượng ứng dụng web và máy tính để bàn một cách hiệu quả, giúp tăng cường độ tin cậy và hiệu suất của ứng dụng.

* + 1. Các tính năng của Telerik Test Studio
* Sản phẩm đa dạng: Telerik cung cấp một loạt các sản phẩm và công cụ phát triển phù hợp cho nhiều nền tảng, bao gồm ứng dụng web, ứng dụng di động, ứng dụng máy tính để bàn, và nền tảng điện toán đám mây.
* Hỗ trợ cho nhiều ngôn ngữ lập trình: Các sản phẩm của Telerik thường hỗ trợ nhiều ngôn ngữ lập trình như C#, JavaScript, và Java, giúp phát triển viên lựa chọn ngôn ngữ phù hợp với dự án của họ.
* Công cụ phát triển giao diện người dùng: Telerik cung cấp các công cụ giúp phát triển viên xây dựng giao diện người dùng đẹp và tương tác, bao gồm các thành phần UI và thư viện giao diện dựa trên nhiều nền tảng.
* Tích hợp và mở rộng: Các sản phẩm Telerik thường có khả năng tích hợp và mở rộng, cho phép phát triển viên tùy chỉnh và mở rộng tính năng theo nhu cầu cụ thể của dự án.
* Cộng đồng và hỗ trợ: Telerik có một cộng đồng phát triển lớn và cung cấp tài liệu học tập và hỗ trợ kỹ thuật cho người dùng.
  + 1. Ưu điểm của Telerik Test Studio
* Đa dạng sản phẩm: Được biết đến với nhiều sản phẩm và công cụ chất lượng, Telerik giúp phát triển viên xây dựng ứng dụng đa dạng trên nhiều nền tảng một cách hiệu quả.
* Giao diện người dùng tốt: Các công cụ phát triển giao diện người dùng của Telerik giúp tạo ra giao diện đẹp và tương tác, giúp ứng dụng thu hút người dùng.
* Hỗ trợ đa nền tảng: Telerik hỗ trợ phát triển ứng dụng cho nhiều nền tảng, bao gồm web, di động, và máy tính để bàn.
* Tích hợp và mở rộng: Các công cụ của Telerik có khả năng tích hợp với các môi trường phát triển phổ biến và cho phép tùy chỉnh theo nhu cầu cụ thể.
  + 1. Nhược điểm của Telerik Test Studio
* Chi phí cao: Các sản phẩm và công cụ của Telerik thường có chi phí sử dụng, điều này có thể tạo áp lực cho các dự án với nguồn lực hạn chế.
* Cần học hỏi và đào tạo: Một số công cụ Telerik có độ khó trong việc học sử dụng, và việc đào tạo nhóm phát triển có thể đòi hỏi thời gian.
* Tốn nhiều hiệu suất và tài nguyên: Một số sản phẩm Telerik có thể đòi hỏi tài nguyên máy tính cao, đặc biệt khi xây dựng ứng dụng phức tạp.
* Phụ thuộc vào môi trường phát triển: Sử dụng các công cụ Telerik có thể yêu cầu phát triển viên tuân theo mô hình phát triển và công nghệ cụ thể mà Telerik hỗ trợ.
* Không hỗ trợ đa nền tảng, ví dụ như MacOS.

1. KIỂM THỬ WEBSITE BÁN VÉ MÁY BAY
   1. Đặc tả hệ các chức năng hệ thống (System Functions)
      1. Kiểm thử các chức năng trong phần mềm

* Khách hàng:
* Đăng ký/ đăng nhập.
* Tìm kiếm chuyến bay.
* Lựa chọn loại ghế trên chuyến bay (First Class, Business Class, Economy Class).
* Xem tất cả các chương trình khuyến mãi.
* Đặt vé qua website.
* Theo dõi lịch trình bay.
* Hủy vé.
* Đóng góp ý kiến hoặc tư vấn hỗ trợ.
* Quản lý:
* Kiểm tra danh sách vé được đặt dựa trên lịch sử đặt vé.
* Cập nhật vé mới được bán vào hệ thống.
* Cập nhật vé máy bay lên hệ thống.
* Tra cứu lịch sử vé máy bay đã được đặt.
* Cập nhật tình trạng vé máy bay.
* Cập nhật chuyến bay lên hệ thống.
* Cập nhật tình trạng chuyến bay lên hệ thống.
* Muốn xem danh sách chuyến bay.
* Xem lịch sử cập nhật chuyến bay.
  + 1. Các Use Case
* Khách hàng:
* Đặt vé máy bay trực tuyến để tiết kiệm thời gian.
* Tìm kiếm các chuyến bay có sẵn để đặt vé trực tuyến.
* Thanh toán vé trực tuyến để tiết kiệm thời gian.
* Tra cứu mã vé để xem lại lịch sử đặt vé.
* Admin:
* Quản lý thông tin chuyến bay để xem, thêm hoặc chỉnh sửa các chuyến bay.
* Quản lý thông tin tuyến bay để xem, thêm, chỉnh sửa hoặc xóa các tuyến bay.
* Xem thông tin chi tiết các chuyến bay đã hoàn thành để có thể quản lý nguồn nhân lực và kế hoạch sản xuất.
* Quản lý thông tin của khách hàng để xem, chỉnh sửa thông tin khách hàng.
* Quản lý thông tin vé máy bay để xem, thêm, chỉnh sửa hoặc xóa các vé máy bay.
* Xem thông tin chi tiết hóa đơn của khách hàng để quản lý kế hoạch sản xuất.
* Quản lý các chương trình khuyến mãi để có thể quản lý kế hoạch sản xuất.
* Quản lý nhân viên để xem, chỉnh sửa thông tin các nhân viên.
  + 1. Đặc tả các chức năng mà Case Actor sử dụng

Bảng 1. . Bảng đặc tả chức năng

|  |  |
| --- | --- |
| **Use case** | **Mô tả** |
| Đặt vé máy bay trực tuyến | - Khách hàng đăng nhập vào hệ thống. |
| - Khách hàng tìm kiếm các chuyến bay có sẵn. |
| - Khách hàng chọn chuyến bay và đặt vé trực tuyến. |
| - Hệ thống xác nhận đặt vé thành công và cung cấp mã vé. |
| - Khách hàng thanh toán vé trực tuyến. |
| - Hệ thống cung cấp thông tin hóa đơn cho khách hàng. |
| Tra cứu mã vé | - Khách hàng nhập mã vé để tra cứu. |
| - Hệ thống hiển thị thông tin chi tiết về vé và lịch sử đặt vé của khách hàng. |
| Xem thông tin chuyến bay | - Nhân viên bán vé đăng nhập vào hệ thống. |
| - Nhân viên bán vé tìm kiếm thông tin chi tiết về chuyến bay để giải đáp thắc mắc của khách hàng. |
| - Nhân viên bán vé xem thông tin chi tiết về các chuyến bay đã được đặt để quản lý nguồn nhân lực. |
| - Nhân viên bán vé xem thông tin chi tiết về các chuyến bay đã hoàn thành để quản lý nguồn nhân lực và kế hoạch sản xuất. |
| Xem thông tin hóa đơn khách hàng | - Nhân viên bán vé đăng nhập vào hệ thống. |
| - Nhân viên bán vé xem thông tin chi tiết về hóa đơn của khách hàng để hỗ trợ và giải đáp thắc mắc. |
| Xem chương trình khuyến mãi | - Nhân viên bán vé và admin đăng nhập vào hệ thống. |
| - Nhân viên bán vé và admin xem thông tin chi tiết về các chương trình khuyến mãi để hỗ trợ khách hàng và quản lý kế hoạch sản xuất. |
| Quản lý thông tin chuyến bay | - Admin đăng nhập vào hệ thống. |
| - Admin xem, thêm, chỉnh sửa thông tin chuyến bay. |
| Quản lý thông tin tuyến bay | - Admin đăng nhập vào hệ thống. |
| - Admin xem, thêm, chỉnh sửa, xóa thông tin tuyến bay. |
| Quản lý thông tin khách hàng | - Admin đăng nhập vào hệ thống. |
| - Admin xem, chỉnh sửa thông tin khách hàng. |
| Quản lý thông tin vé máy bay | - Admin đăng nhập vào hệ thống. |
| - Admin xem, thêm, chỉnh sửa, xóa thông tin vé máy bay. |
| Quản lý thông tin hóa đơn | - Admin đăng nhập vào hệ thống. |
| - Admin xem thông tin chi tiết về hóa đơn của khách hàng để quản lý kế hoạch sản xuất. |
| Quản lý chương trình khuyến mãi | - Admin đăng nhập vào hệ thống. |
| - Admin xem, thêm, chỉnh sửa thông tin chương trình khuyến mãi. |
| Quản lý nhân viên | - Admin đăng nhập vào hệ thống. |
| - Admin xem, chỉnh sửa thông tin các nhân viên. |

* + 1. Sơ đồ Use Case tổng quát

A diagram of a diagram

Description automatically generated

Hình 1. . Use Case: Tổng quát

* + - 1. Sơ đồ chi tiết: Quản trị hệ thống

A diagram of a diagram

Description automatically generated with medium confidence

Hình 1. . Use Case: Quản trị hệ thống

* + - 1. Sơ đồ chi tiết: Quản lý tài chính

A diagram of a diagram

Description automatically generated

Hình 1. . Use Case: Quản lý tài chính

* + - 1. Sơ đồ chi tiết: Quản lý chuyến bay

A diagram of a diagram

Description automatically generated with medium confidence

Hình 1. . Use Case: Quản lý chuyến bay

* + - 1. Sơ đồ chi tiết: Quản lý vé máy bay

A diagram of a network

Description automatically generated

Hình 1. . Use Case: Quản lý vé máy bay

* + 1. Thiết kế cơ sở dữ liệu

A picture containing text, diagram, plan, parallel

Description automatically generated

Hình 1. . ERD Diagram

* + 1. Sơ đồ Sequence

A diagram of a project

Description automatically generated

Hình 1. . Quy trình hệ thống bán vé máy bay

* + 1. Sơ đồ Activity

A diagram of a diagram

Description automatically generated

Hình 1. . Quy trình đăng nhập

A diagram of a chat

Description automatically generated with medium confidence

Hình 1. . Quy trình đặt vé

* 1. Xác định mục tiêu, các chuẩn chất lượng cần đạt của hệ thống
     1. Mục tiêu
* Kiểm thử xem giao diện người dùng của trang web có hoạt động như mong đợi không và có đáp ứng được nhu cầu của khách hàng không.
* Kiểm thử chức năng thêm, xóa, chỉnh sửa thông tin, tìm kiếm, có hoạt động như mong đợi mà không có bất kỳ lỗi nào không.
* Kiểm thử chức năng đăng nhập bằng tài khoản đã được phân quyền trước đó, sau khi đăng nhập thì người dùng có sử dụng được đúng theo role đã chia không.
* Đảm bảo mục tiêu test đúng đắn của các chức năng, bao gồm định hướng, dữ liệu đầu vào, xử lý và dữ liệu nhận được.
  + 1. Các chuẩn cần đạt của hệ thống
* *Tiêu chuẩn ISO 9126:* tiêu chuẩn quốc tế để đánh giá chất lượng các phần mềm doanh nghiệp.
* *Tiêu chí 1: Tính chức năng*
* Tính phù hợp: cung cấp các chức năng cần thiết cho việc quản lý nhân viên (thêm, xóa, sửa, tìm kiếm).
* Tính chính xác: đưa ra các kết quả đúng đắn khi người dùng thực hiện một thao tác bất kỳ trên hệ thống.
* Tính an toàn: yêu cầu người dùng phải đăng nhập khi sử dụng hệ thống.
* *Tiêu chí 2: Tính tin cậy*
* Khả năng chịu lỗi: khả năng của phần mềm hoạt động ổn định tại một mức độ nào đó khi đang trong trường hợp có lỗi.
* *Tiêu chí 3: Tính khả chuyển*
* Khả năng thích nghi: Cho phép hệ thống có thể được thực thi trên các trang nền tảng website khác nhau.
* Khả năng cài đặt được: Phần mềm dễ dàng cài đặt trên nhiều máy tính khác nhau, vẫn giữ đúng các chuẩn, quy ước, quy định.
  1. Phạm vi kiểm thử của hệ thống (Scope)
     1. Các chức năng cần kiểm thử

Các yêu cầu chức năng thực hiện test:

* Đăng nhập.
* Đăng kí.
* Lên lịch chuyến bay.
* Quản lý nhân viên.
* Tạo đường bay.
* Khuyến mãi.
  + 1. Các chức năng không cần kiểm thử
* Đăng xuất.
* Logo.
  1. Thiết kế Test Cases/ Test Checklists
     1. Kiểm thử chức năng
        1. Schedule A Flight







* + - 1. Login, Register









* + - 1. Home Page













* + - 1. Person Detail











* + - 1. Booking Ticket













* + - 1. History Booking





* + - 1. Flight Route







* + 1. Kiểm thử phi chức năng
       1. Schedule A Flight



* + - 1. Login, Register



* + - 1. Home Page



* + 1. Kiểm thử các yếu tố khác
       1. Schedule A Flight





* + - 1. Login, Register







* + - 1. Home Page







* + - 1. Person Detail







* + - 1. Booking Ticket











* + - 1. History Booking

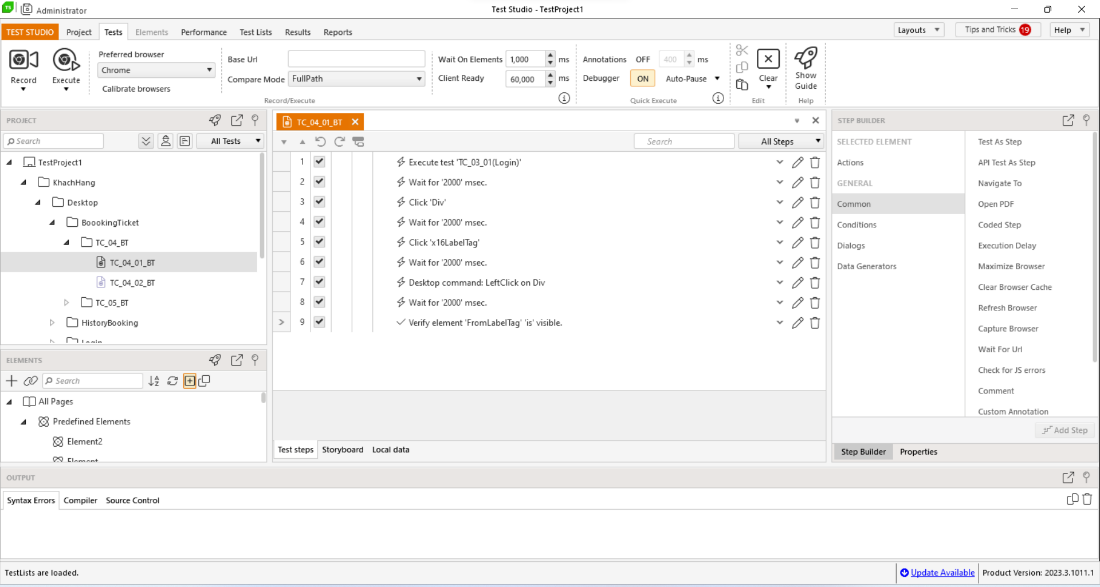




* + - 1. Flight Route

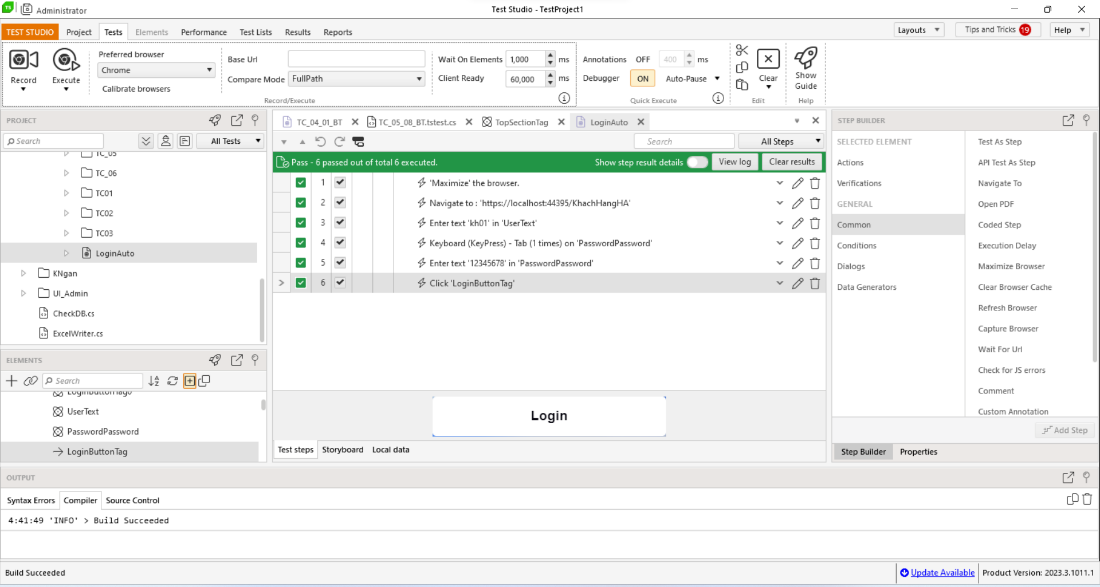


* 1. Thực thi kiểm thử trên Telerik và báo cáo Buglist
     1. Kết quả sau khi đã record



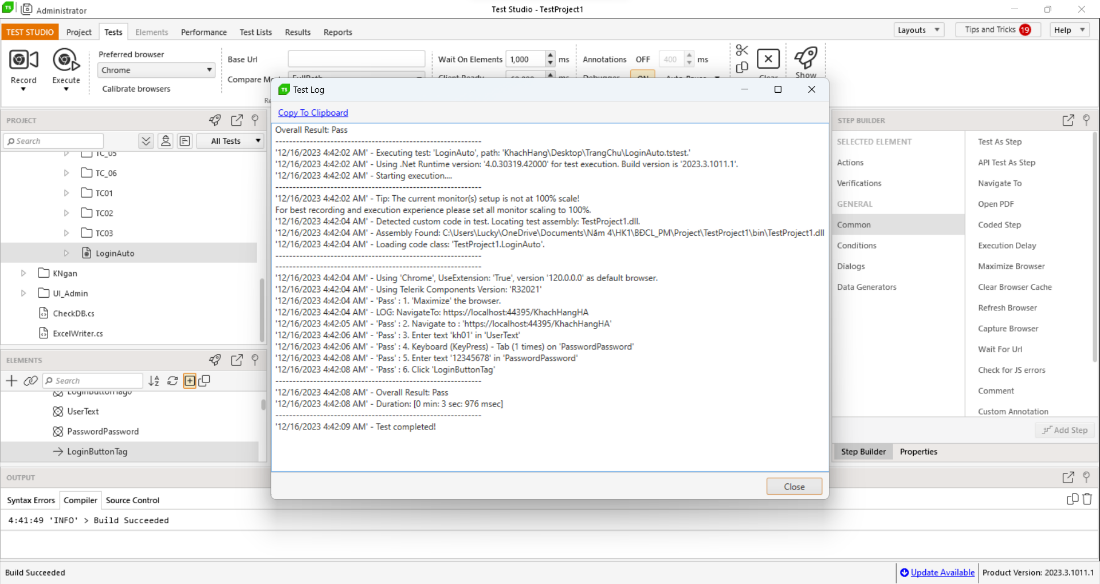
Hình 5. . Sau khi record

* + 1. Kết quả sau khi thực thi Automation Test



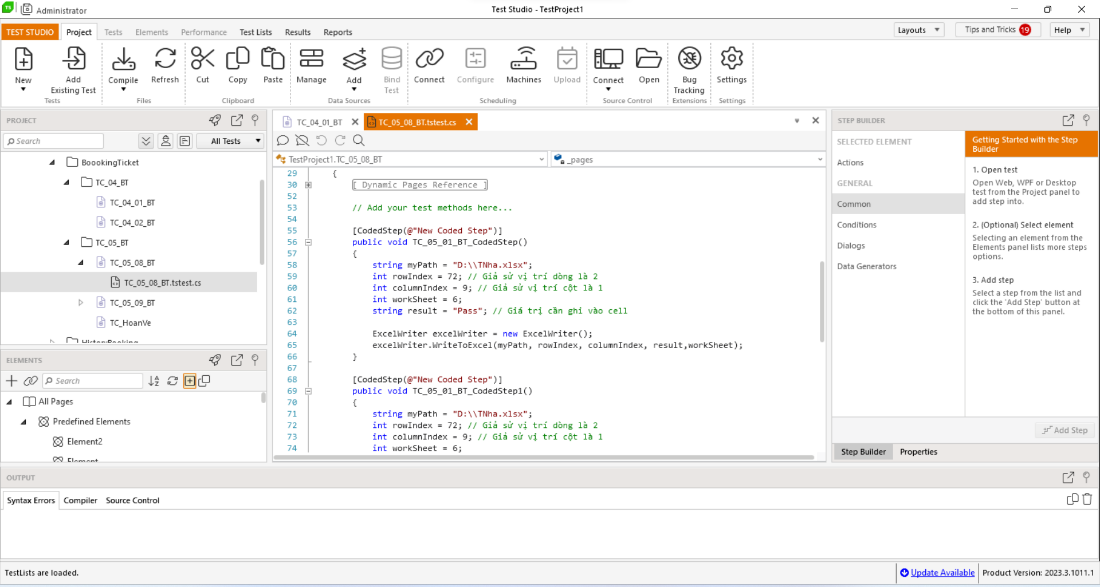
Hình 5. . Sau khi chạy automation test

* + 1. Kết quả trả về từ file log



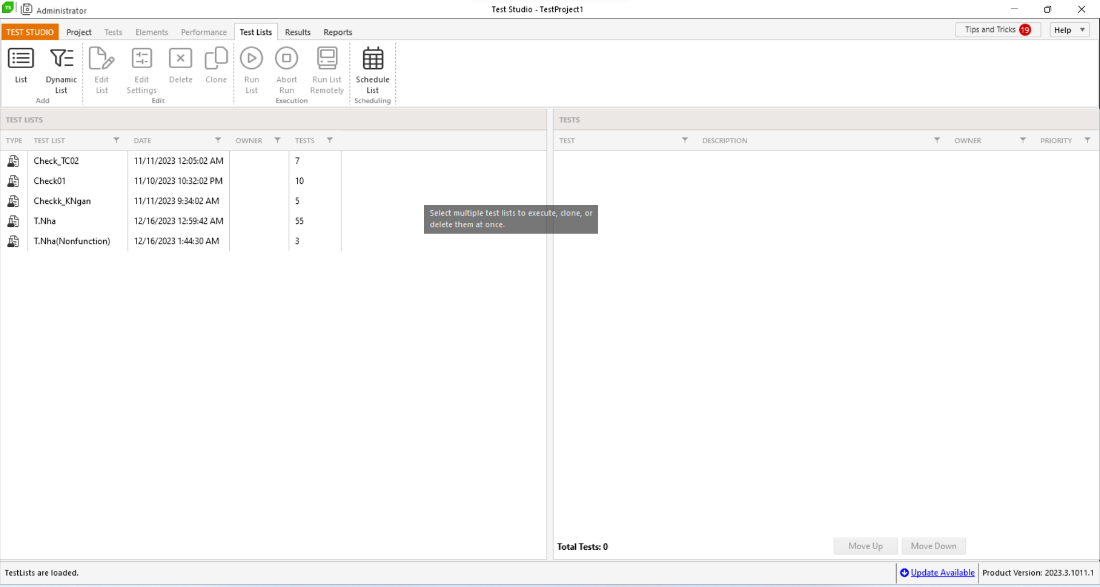
Hình 5. . File log

* + 1. Thực thi chuyển Step thành code C#



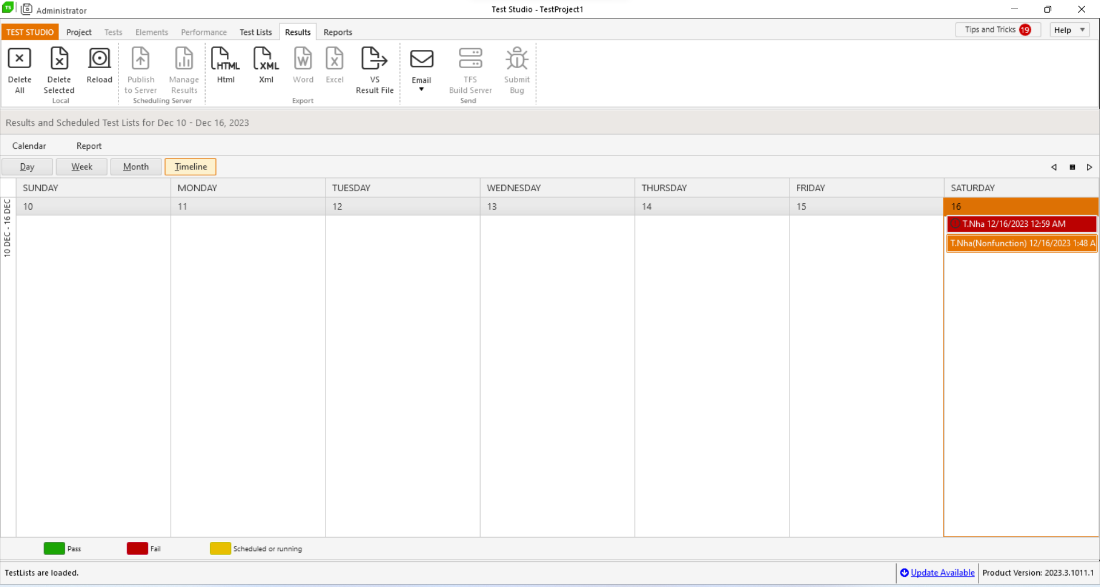
Hình 5. . Chuyển Step thành C#

* + 1. Thêm danh sách các Step vào test list



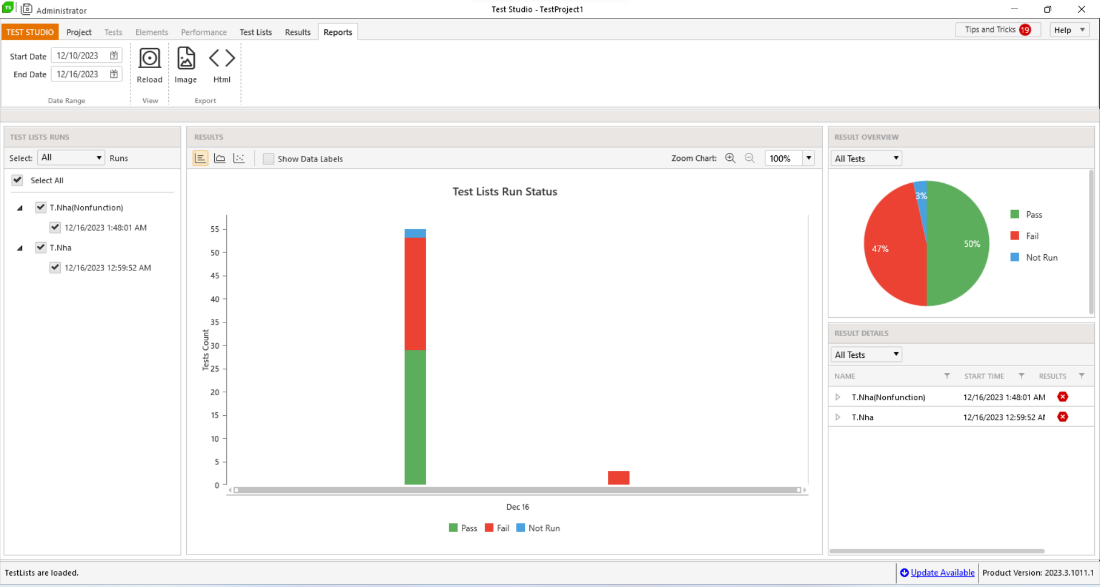
Hình 5. . Thêm Step vào Test List

* + 1. Thông báo kết quả sau khi chạy test list



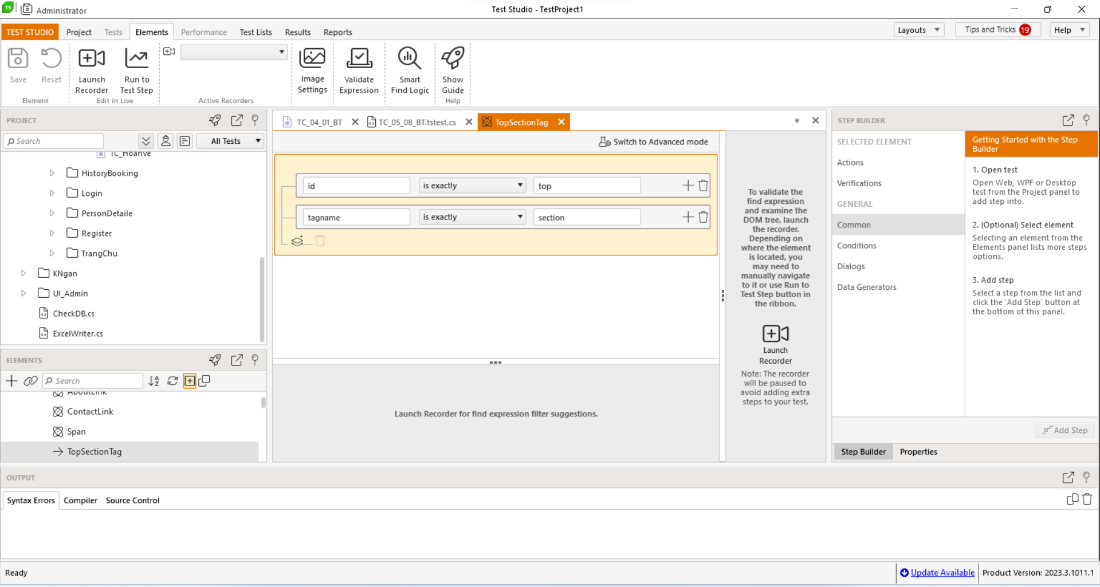
Hình 5. . Sau khi chạy Test List

* + 1. Báo cáo thống kê kết quả của các step sau khi chạy test list



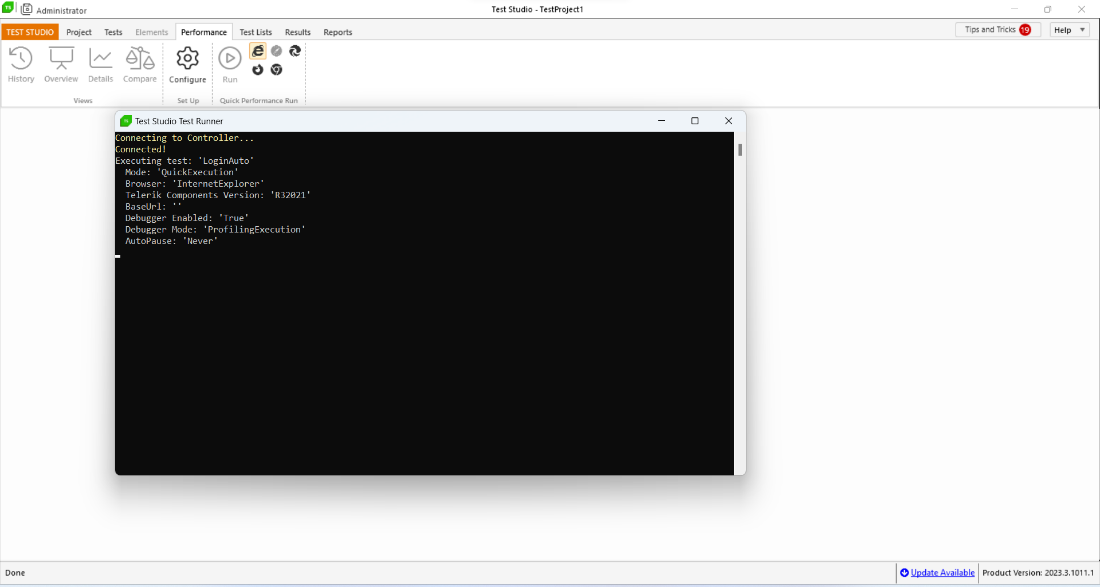
Hình 5. . Thống kê

* + 1. Thực hiện tìm kiếm/ nhận diện Element có trong web



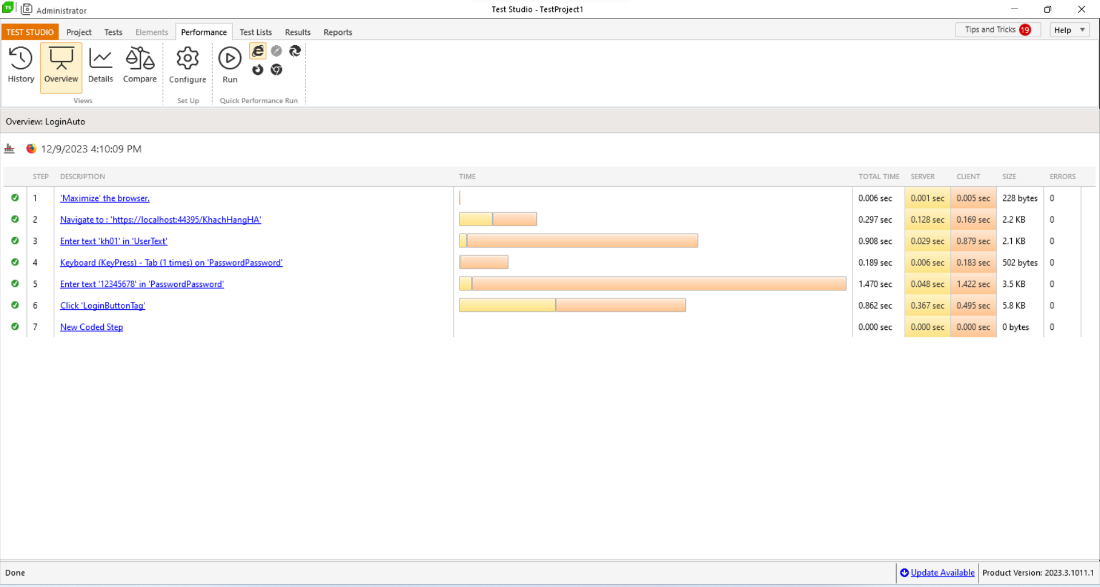
Hình 5. . Tìm kiếm/ nhận diện Element

* + 1. Thực hiện test performance dựa trên test Script



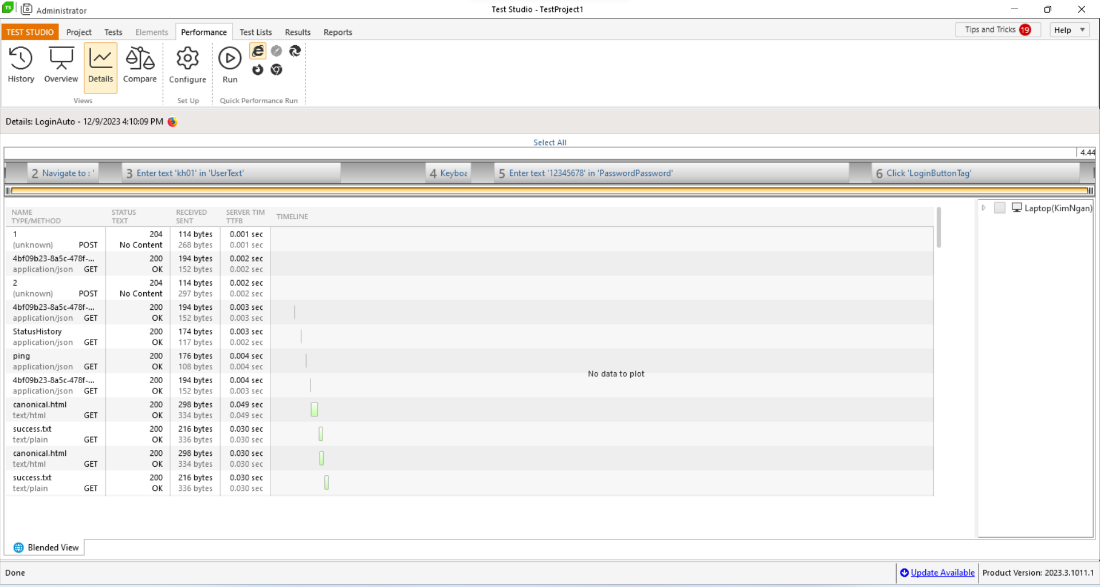
Hình 5. . Test Performance

* + 1. Hiển thị thông tin sau khi test Performance



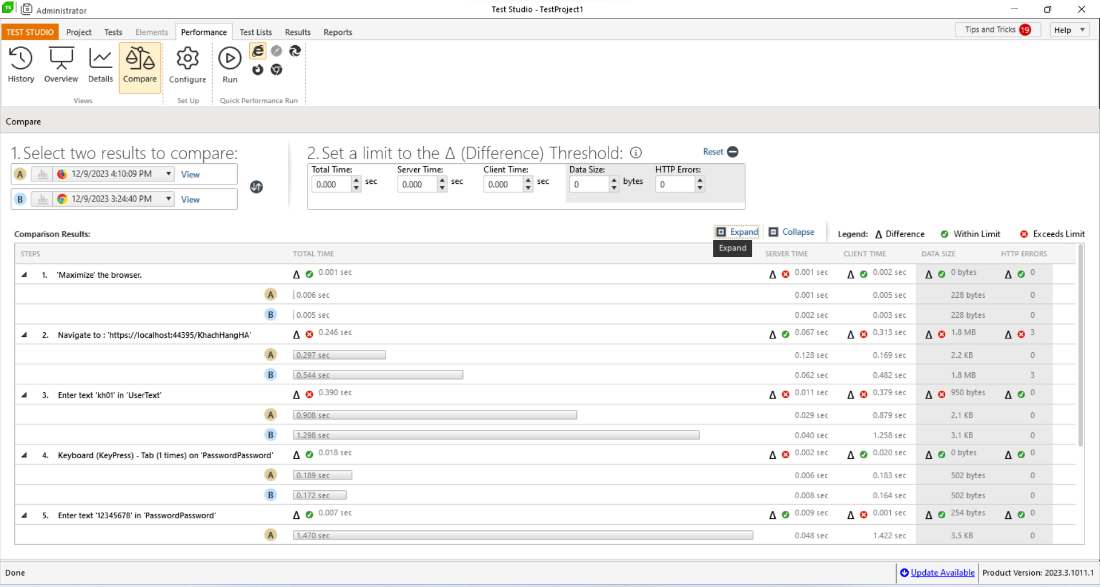
Hình 5. . Hiển thị thông tin test

* + 1. Hiển thị thông tin chi tiết Performance



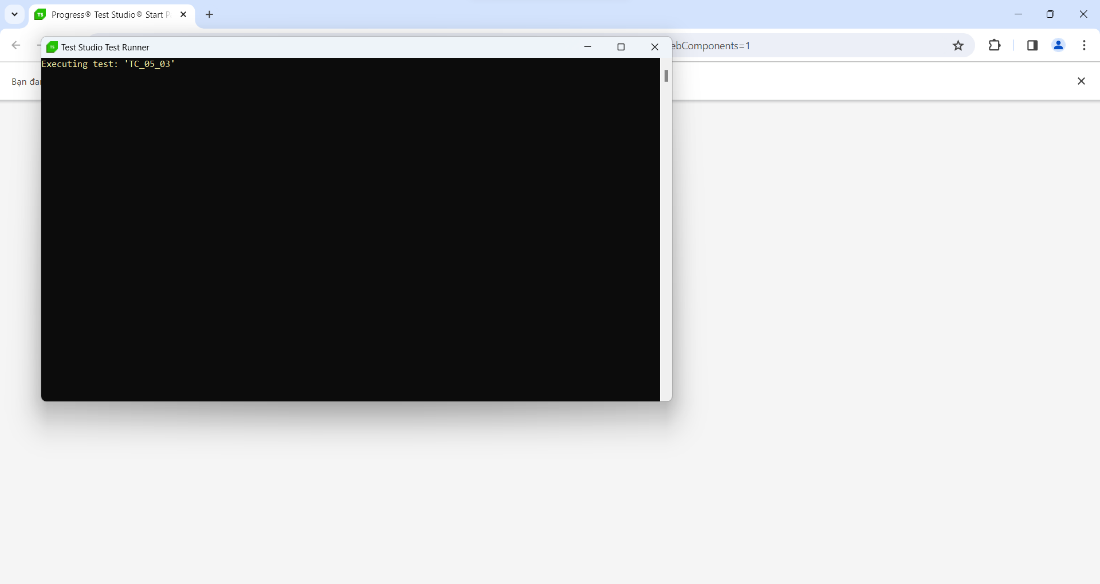
Hình 5. . Hiển thị chi tiết

* + 1. So sánh Performance từ các trình duyệt khác nhau



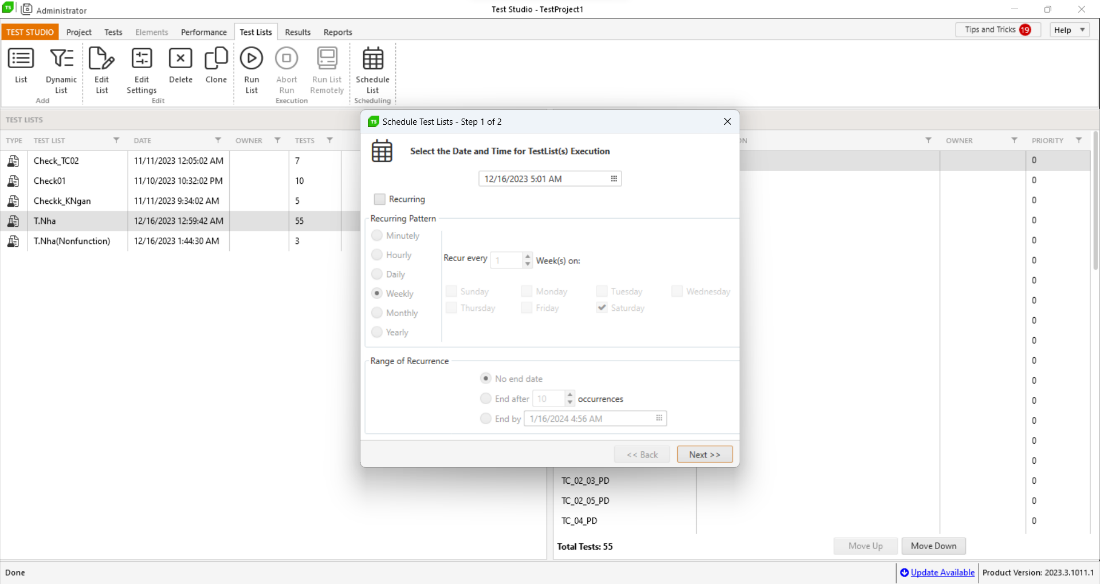
Hình 5. . So sánh khác nhau

* + 1. Thực thi test Check List



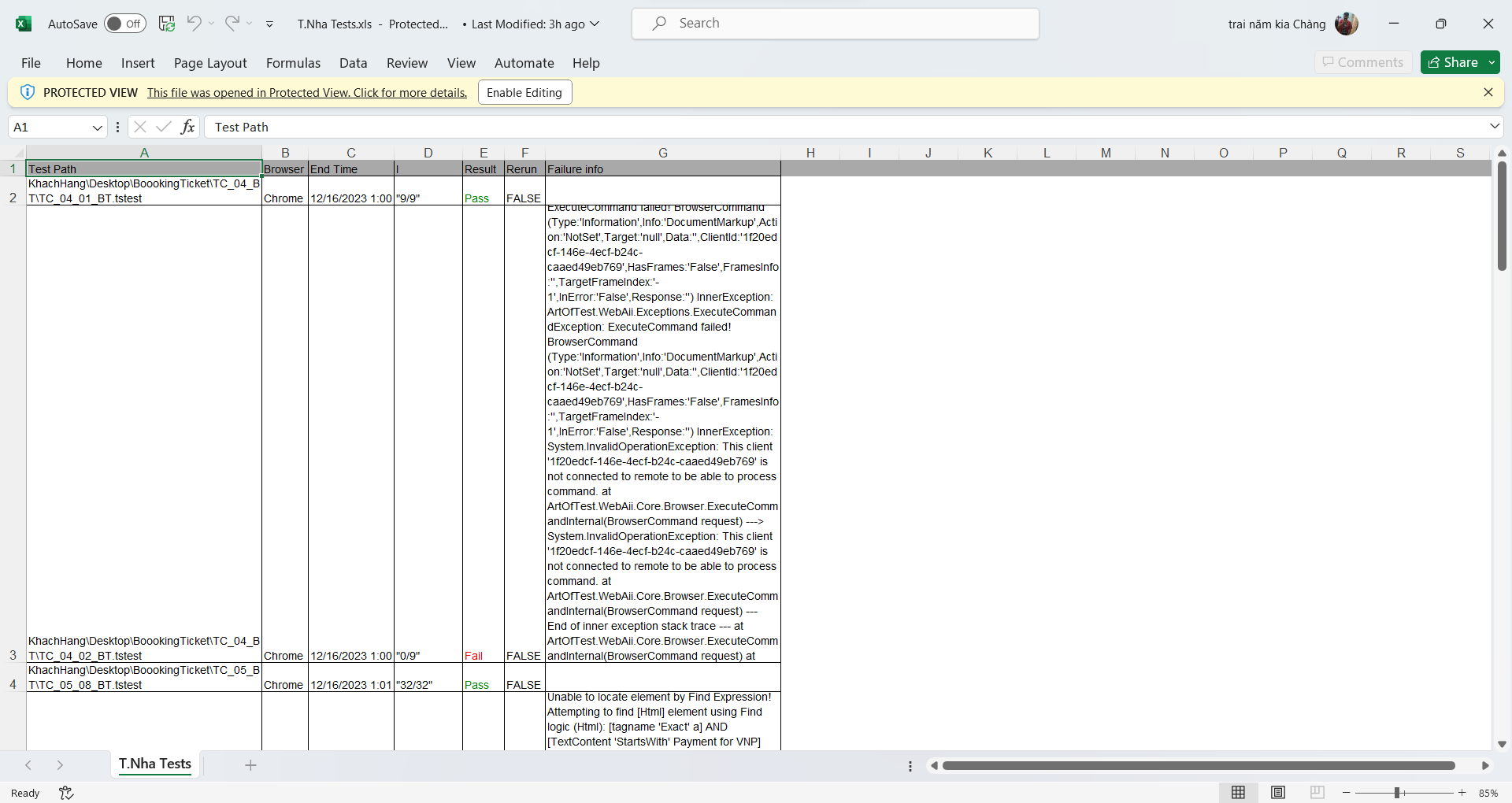
Hình 5. . Test Check List

* + 1. Lên lịch cho Check List



Hình 5. . Lên lịch test

* + 1. Xuất kết quả báo cáo sang file Excel sau khi test list



Hình 5. . Xuất ra Excel

* 1. Test Report



1. KẾT LUẬN
   1. Ưu điểm

* Hệ thống đáp ứng được nhiều nhu cầu và có nhiều chức năng.
* Hệ thống có thể chạy ổn định trên window.
* Tốc độ xử lý trang web, reponsive khá ổn.
* Độ bảo mật của web được bảo đảm ngay cả khi test performance.
* Lên lịch tự động chạy chương trình kiểm tra.
* Thống kê báo cáo rõ ràng minh bạch và có thể xuất file đa dạng.
* Linh hoạt trong việc trao đổi dữ liệu giữa các máy, có thể sử dụng git để lưu trữ thông tin kiểm thử về dự án.
  1. Nhược điểm
* Chương trình có tính chuyên nghiệp chưa cao, chưa giải quyết được trọn vẹn những vấn đề nảy sinh trong quá trình quản lý.
* Chức năng chưa thực sự toàn vẹn, thiếu sót nhiều.
* Số lượng truy cập vào trang web thiếu tính ổn định.
* Vẫn còn mắc một số lỗi nhất định.
* Khó học với nguồn tài liệu khan hiếm.
* Cộng đồng phát triển không lớn.
* Chức năng tìm Element và OCR chưa đủ nhận dạng rõ ràng.
* Chi phí sử dụng phần mềm khá cao so với các phần mềm kiểm thử khác.
* Khi viết Script hệ thống không tự động thêm thư viện mà phải làm thủ công.
  1. Giải pháp mở rộng (nâng cao)
* Kiểm tra thường xuyên phần mềm bằng Tự động hóa.
* Dự án nên cử một người chủ chốt của mình làm công việc bảo trì sau khi dự án kết thúc giai đoạn phát triển.
* Chuẩn hóa thao tác bảo trì và thiết bị môi trường bảo trì.
* Thiết kế phần mềm tối ưu nhất sao cho dễ dàng bảo trì mỗi khi phát sinh các lỗi (thường là lỗi phức tạp).
* Quản lý tốt các tài liệu, dữ liệu, chương trình nguồn, dữ liệu thử, …
* Tổ chức test case theo từng dạng module dễ kiểm soát
* Xây dựng lớp kết nối đến cơ sở dữ liệu trực tiếp có thể can thiệp kiểm tra dữ liệu ở bất cứ step nào.

* 1. Kết quả đạt được
* Tìm hiểu và nắm được lý thuyết lẫn thực hành để áp dụng vào đồ án.
* Nắm được tổng quan về kiểm thử phần mềm: các phương pháp, kỹ thuật kiểm thử phần mềm và các vấn đề liên quan; tìm hiểu và nắm được phương pháp thiết kế test case trong kiểm thử phần mềm và áp dụng phương pháp đó với phần mềm thực tế.
* Tìm hiểu về các tool test cụ thể là công cụ Telerik Test Studio.
* Hiểu sâu rộng hơn về môi trường automation test, hạn chế lượng test case đáng kể khi có cho mọi thứ tự động hóa.
* Nắm bắt và rút kinh nghiệm cho những lần xây dựng dự án website lần sau.
* Kiểm thử là rất quan trọng trong quá trình sản xuất phần mềm, đảm bảo chất lượng phần mềm. Sự áp dụng với kiến thức tìm hiểu được mới chỉ dừng lại ở một trang web nhỏ và đơn giản.
  1. Kết quả chưa đạt được
* Chưa thể test toàn diện tất cả các chức năng của website.
* Chưa thể vận dụng hết công cụ tài nguyên có sẵn mà phần mềm cung cấp.
* Chưa tận dụng tối đa và tối ưu hóa trong việc viết testcase
  1. Hướng phát triển
* Thực hiện kiểm thử trên mô hình bài toán phần mềm rộng hơn, phức tạp hơn.
* Tìm hiểu và nghiên cứu thêm về các công cụ kiểm thử tự động, kiểm thử website, kiểm thử tải, kiểm thử cơ sở dữ liệu, …
* Đi đến kiểm thử Restful API, kiểm tra độ xác thực người dùng.
* Tập trung xây dựng các phương thức lớp và ứng dụng Design Pattern phục vụ cho việc viết Test Script hiệu quả hơn, tối ưu hóa lượng tải công việc

**TÀI LIỆU THAM KHẢO**

HAYES, Linda G., 2004. *The Automated Testing Handbook* [online]. Software Testing Institute. Available from: http://books.google.ie/books?id=-jangThcGIkC&printsec=frontcover&dq=Automated+Testing+Unleashed:+The+Complete+Handbook,+5++NEW+BOOKS&hl=&cd=1&source=gbs\_api

ANGELOV, Anton, 2023. Most Exhaustive WebDriver Locators Cheat Sheet. *Automate The Planet* [online]. 26 July 2023. Available from: https://www.automatetheplanet.com/selenium-webdriver-locators-cheat-sheet/

Dotnet-Bot, [no date]. SqlConnection Class (System.Data.SqlClient). *Microsoft Learn* [online]. Available from: https://learn.microsoft.com/en-us/dotnet/api/system.data.sqlclient.sqlconnection?view=dotnet-plat-ext-8.0

The Selenium Browser Automation Project, [no date]. *Selenium* [online]. Available from: https://www.selenium.dev/documentation/

NUnit Documentation Site | NUnit Docs, [no date]. [online]. Available from: https://docs.nunit.org/

Telerik, [no date]. Getting Started - Progress Test Studio. [online]. Available from: https://docs.telerik.com/teststudio/testing-framework/getting-started