

ĐỒ ÁN MÔN HỌC
CÔNG CỤ VÀ MÔI TRƯỜNG PHÁT TRIỂN PHẦN MỀM

ĐỀ TÀI:
ỨNG DỤNG ÔN THI GIẤY PHÉP LÁI XE

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn: **ThS. Nguyễn Hữu Trung**

Sinh viên thực hiện:

MSSV:	Họ và tên:	Lớp:
2011068866	Đàng Anh Min	20DTHC4
2011064200	Kiều Thị Hà Duyên	20DTHB2
2011065095	Phan Thị Anh	20DTHB1

TP. Hồ Chí Minh, 2024

LỜI CAM ĐOAN

Trong quá trình thực hiện Đồ án môn Công cụ và môi trường phát triển phần mềm, chuyên ngành “Công nghệ phần mềm” do khả năng tiếp thu thực tế còn nhiều hạn chế, kiến thức chưa sâu rộng. Mặc dù nhóm chúng em đã cố gắng hết sức nhưng chắc chắn bài báo cáo này khó tránh khỏi những thiếu sót, nhưng những gì được trình bày trong báo cáo này là sự thể hiện những kết quả đạt được dưới sự hướng dẫn của giảng viên – thầy Nguyễn Hữu Trung đã nhiệt tình hướng dẫn, khuyến khích và tạo điều kiện cho chúng em hoàn thành tốt đề tài của mình.

Chúng em xin cam đoan rằng nội dung trình bày trong báo cáo đồ án, những số liệu và kết quả nghiên cứu là trung thực, hoàn toàn được thực hiện không sao chép bất kỳ nguồn nào khác. Ngoài ra, trong bài báo cáo có sử dụng một số nguồn tài liệu tham khảo đã được trích dẫn nguồn và chú thích rõ ràng. Nếu không đúng sự thật, chúng em xin chịu mọi trách nhiệm.

Sinh viên đồng thực hiện

Đàng Anh Min

Kiều Thị Hà Duyên

Phan Thị Anh

MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	1
1.1. NGÔN NGỮ LẬP TRÌNH	1
1.1.1. <i>Java</i>	1
1.1.2. <i>XML (Xtensible Markup Language)</i>	1
1.2. MÔI TRƯỜNG PHÁT TRIỂN PHẦN MỀM.....	2
1.2.1. <i>Giới thiệu về Android</i>	2
1.2.2. <i>Android Studio</i>	3
1.2.3. <i>Github</i>	4
1.2.4. <i>Figma</i>	4
1.2.5. <i>Bug</i>	5
CHƯƠNG 2. UI/UX.....	7
2.1. GIỚI THIỆU VỀ FIGMA	7
2.2. CÁC BƯỚC THỰC HIỆN	7
2.2.1. <i>Đăng nhập và tạo dự án</i>	7
2.2.2. <i>Tạo Wireframe và Bố cục ban đầu:</i>	8
2.2.3. <i>Thiết kế giao diện người dùng</i>	9
CHƯƠNG 3. GITHUB.....	20
3.1. GIỚI THIỆU	20
3.2. CÁC BƯỚC TRIỂN KHAI.....	20
CHƯƠNG 4. CÔNG CỤ QUẢN LÝ DỰ ÁN (JIRA).....	24
4.1. GIỚI THIỆU	24
4.2. CÁC TÍNH NĂNG	24
4.2.1. <i>Tạo project</i>	24
4.2.2. <i>Tạo board</i>	24
4.2.3. <i>Tạo issues</i>	25
4.2.4. <i>Tìm issues trong Jira</i>	25
4.2.5. <i>Workflow trong Jira</i>	25
4.2.6. <i>Tạo Workflow</i>	25
4.3. HƯỚNG DẪN SỬ DỤNG	26
4.3.1. <i>Cách tạo một Jira site</i>	26
4.3.2. <i>Các thao tác với project</i>	28
4.3.3. <i>Các thao tác với issue</i>	30
4.3.4. <i>Các thao tác với Workflow</i>	34

DANH MỤC HÌNH ẢNH, ĐỒ THỊ

Hình 1.1. Kiến trúc của Android	3
Hình 2.1. Đăng nhập tài khoản Figma.....	8
Hình 2.2. Tạo file thiết kế mới	8
Hình 2.3. Sử dụng Frame.....	9
Hình 2.4. Sử dụng các công cụ như khung hình, hình chữ nhật và văn bản	9
Hình 2.5. Tùy chỉnh màu sắc	10
Hình 2.6. Hình ảnh trang chủ	10
Hình 2.7. Hình ảnh trang lý thuyết.....	11
Hình 2.8. Hình ảnh trang biển báo cấm.....	11
Hình 2.9. Hình ảnh trang biển báo hiệu lệnh.....	12
Hình 2.10. Hình ảnh trang chỉ dẫn	12
Hình 2.11. Hình ảnh trang biển báo nguy hiểm	13
Hình 2.12. Hình ảnh trang biển báo phụ	13
Hình 2.13. Hình ảnh trang mẹo ghi nhớ lý thuyết.....	14
Hình 2.14. Hình ảnh trang mẹo ghi nhớ biển báo	14
Hình 2.15. Hình ảnh trang mẹo câu hỏi sa hình	15
Hình 2.16. Hình ảnh trang mẹo thực hành A1,A2.....	15
Hình 2.17. Hình ảnh trang kinh nghiệm thực hành A1,A2	16
Hình 2.18. Hình ảnh trang mẹo thực hành B1,B2	16
Hình 2.19. Hình ảnh trang kinh nghiệm thực hành B1,B2.....	17
Hình 2.20. Hình ảnh giao diện trang thi sát hạch	17
Hình 2.21. Hình ảnh thi sát hạch không đạt	18
Hình 2.22. Hình ảnh thi sát hạch đạt	18
Hình 2.23. Hình ảnh trang lịch sử bài thi	19
Hình 2.24. Hình ảnh đáp án bài thi đã thi.....	19
Hình 3.1. Tạo thư mục repository trên server.	20
Hình 3.2. Chọn Git Bash here ở thư mục cần tải lên để thực hiện các bước kế tiếp.....	21
Hình 3.3. Sử dụng câu lệnh git init.....	21
Hình 3.4. Sử dụng câu lệnh git add README.md	21
Hình 3.5. Sử dụng câu lệnh “git commit -m “first commit””	22
Hình 3.6. Sử dụng câu lệnh “git branch -M main”.....	22
Hình 3.7. Sử dụng câu lệnh git remote add origin.	22
Hình 3.8. Hình ảnh push lên thành công.	23
Hình 4.1. Get it free.	26
Hình 4.2. Tạo tài khoản bằng email	27
Hình 4.3. Chọn template Kanban hoặc Scrum	28
Hình 4.4. Tạo một issue mới trên Jira Software.....	28
Hình 4.5. Xem danh sách project	29
Hình 4.6. Cài đặt project.....	29

Hình 4.7. Sửa tên project Jira	30
Hình 4.8. Xóa project trên Jira	30
Hình 4.9. Xem danh sách issue	31
Hình 4.10. Hình ảnh không thấy mục “issue” trên side bar	31
Hình 4.11. Bật issue navigator	32
Hình 4.12. Tạo issue mới.....	32
Hình 4.13. Thông tin issue hiển thị sau khi tạo một issue mới	34
Hình 4.14. Hướng dẫn tạo Workflow	35
Hình 4.15. Tạo mới Workflow	35

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Ngôn ngữ lập trình

1.1.1. Java

Ngôn ngữ lập trình Java đã trở thành một trong những ngôn ngữ quan trọng và phổ biến nhất trong lĩnh vực phát triển phần mềm. Với tính đa nền tảng, Java cho phép ứng dụng chạy trên nhiều hệ điều hành mà không cần sửa đổi mã nguồn, nhờ vào Java Virtual Machine (JVM). Được thiết kế hướng đối tượng, Java thúc đẩy việc sử dụng khái niệm đối tượng, giúp xây dựng mã nguồn linh hoạt và dễ bảo trì.

Java còn nổi tiếng với tính an toàn và bảo mật cao, sử dụng thuật toán thu thập rác để quản lý bộ nhớ một cách tự động. Sự hỗ trợ rộng rãi cho nhiều loại ứng dụng và nền tảng, cùng với hiệu suất đáng kể và cộng đồng lập trình viên lớn, làm cho Java trở thành lựa chọn hàng đầu trong việc xây dựng ứng dụng di động, web, và các hệ thống phức tạp. Đồng thời, sự tích hợp của Java trong các framework như Spring và Struts cũng giúp nó đóng vai trò quan trọng trong phát triển ứng dụng web.

1.1.2. XML (Xtensible Markup Language)

XML, hay ngôn ngữ đánh dấu mở rộng, là một ngôn ngữ đánh dấu linh hoạt được sử dụng để lưu trữ và truyền thông tin dưới dạng văn bản. Với cấu trúc mở và đóng thẻ, XML làm cho tài liệu trở nên dễ đọc và ghi cho cả con người và máy tính. Điều này cũng cho phép XML mở rộng một cách linh hoạt, vì người dùng có thể tự định nghĩa các thẻ và thuộc tính theo nhu cầu cụ thể của họ.

XML độc lập với tài liệu và hỗ trợ Unicode, cho phép sử dụng nhiều ngôn ngữ và bộ ký tự khác nhau. Nó thường được tích hợp với các ngôn ngữ khác như HTML trong việc mô tả cấu trúc dữ liệu trên web. XML không chỉ được sử dụng trong truyền thông dữ liệu giữa máy chủ và máy khách, mà còn là một công cụ quan trọng trong lưu trữ cấu hình và mô tả dữ liệu cấu trúc. Điều này làm cho XML trở thành một tiêu chuẩn quan trọng và được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

1.2. Môi trường phát triển phần mềm

1.2.1. Giới thiệu về Android

1.2.1.1. Khái niệm

Android là một hệ điều hành di động (OS - Operating System) phổ biến được phát triển dựa trên nền tảng Linux và được thiết kế đặc biệt dành cho các nhà phát triển thiết bị, các nhà mạng, lập trình viên có thể tiếp cận nhanh chóng, điều chỉnh tự do trong mã nguồn mở đó.

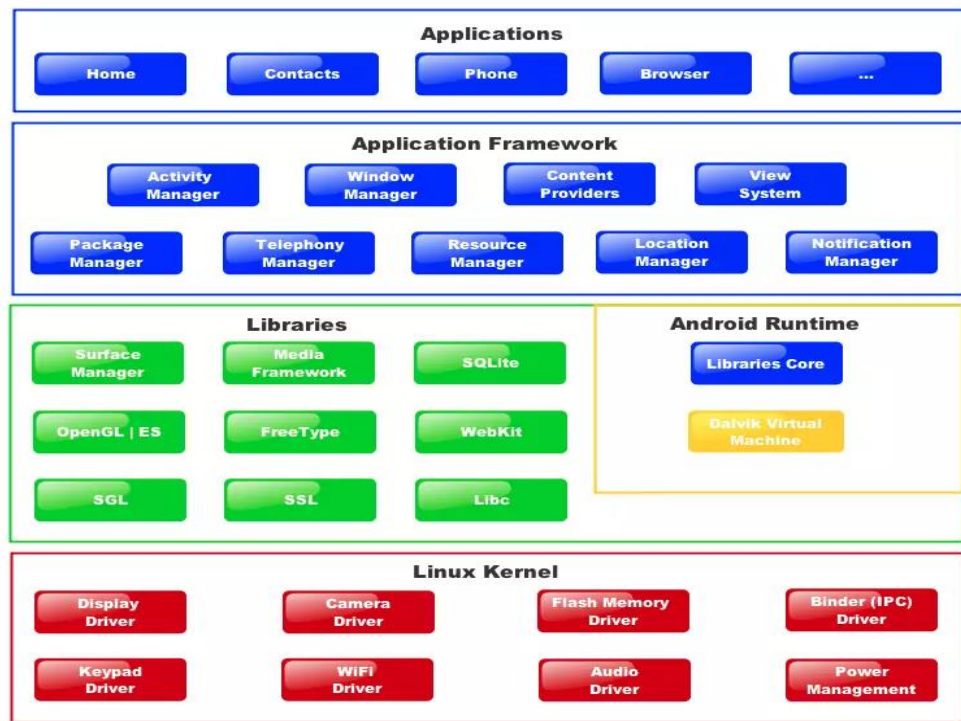
Hiện hệ điều hành Android tích hợp trên điện thoại di động, máy tính bảng, smart tivi, smartwatch, cùng nhiều thiết bị thông minh trong nhà khác.

Android được xây dựng để cho phép các nhà phát triển để tạo ra các ứng dụng di động hấp dẫn, tận dụng tất cả tính năng một chiếc điện thoại đã cung cấp. Nó được xây dựng để được thực sự mở. Android được xây dựng trên mã nguồn mở Linux Kernel. Hơn nữa, nó sử dụng một máy ảo tùy chỉnh được thiết kế để tối ưu hóa bộ nhớ và tài nguyên phần cứng trong một môi trường di động.

1.2.1.2. Kiến trúc của Android

Kiến trúc của Android được thiết kế mở và linh hoạt, giúp hỗ trợ sự đa dạng của nhiều loại thiết bị di động và đảm bảo tính tương thích giữa các ứng dụng và thiết bị khác nhau.

Hệ điều hành Android là 1 ngăn xếp các thành phần phần mềm, được chia thành 5 phần và 4 lớp chính như trong hình bên dưới:



Hình 1.1. Kiến trúc của Android

1.2.2. Android Studio

Android Studio là một môi trường phát triển tích hợp (IDE) được Google phát triển đặc biệt cho việc xây dựng ứng dụng Android. Với giao diện người dùng thân thiện, Android Studio giúp nhà phát triển tập trung vào quá trình phát triển mà không gặp khó khăn trong việc quản lý dự án. Đặc biệt, tích hợp sâu với Gradle Build System giúp tự động hóa quá trình xây dựng ứng dụng và quản lý các phụ thuộc.

Công cụ Visual Layout Editor của Android Studio cho phép nhà phát triển thiết kế giao diện người dùng một cách trực quan, chỉ cần kéo và thả các thành phần mà không cần viết mã nguồn. Trình soạn thảo mã nguồn thông minh giúp tăng tốc quá trình viết mã với tính năng autocompletion và đánh giá lỗi.

Android Studio cung cấp Android Emulator để kiểm thử ứng dụng trên nhiều loại thiết bị và phiên bản Android khác nhau. Ngoài ra, nó hỗ trợ kết nối và kiểm thử trực tiếp trên thiết bị thật.

Với tích hợp sẵn của Firebase, nhà phát triển có thể dễ dàng tích hợp các tính năng như phân tích, xác thực người dùng, lưu trữ dữ liệu, và nhiều dịch vụ khác. Đồng

thời, Android Studio hỗ trợ các hệ thống quản lý phiên bản như Git để giúp nhóm phát triển làm việc hiệu quả và theo dõi lịch sử thay đổi của mã nguồn. Tóm lại, Android Studio là công cụ mạnh mẽ và linh hoạt, đóng vai trò quan trọng trong quá trình phát triển ứng dụng Android.

1.2.3. Github

GitHub là một nền tảng quản lý mã nguồn mở phổ biến nhất hiện nay, cung cấp một loạt các công cụ và tính năng quan trọng cho việc phát triển phần mềm. Trong GitHub, các dự án được tổ chức trong các repository, nơi chứa toàn bộ mã nguồn, lịch sử thay đổi và các nhánh phát triển. Mỗi lần thay đổi được thực hiện, người dùng có thể tạo commit để ghi lại các thay đổi và mô tả chúng.

Các nhánh (branches) cho phép nhà phát triển làm việc song song trên các tính năng và sửa lỗi mà không làm ảnh hưởng đến nhánh chính (master branch). Sau khi hoàn thành, các thay đổi có thể được đề xuất thông qua Pull Request, nơi người khác có thể xem xét và thảo luận trước khi thực hiện việc merge.

Ngoài ra, GitHub còn cung cấp các công cụ bổ sung như wiki để tạo tài liệu và dự án để quản lý công việc và tiến độ của dự án. Các vấn đề, sự cố và cải tiến được đề xuất thông qua các Issue, tạo nên một môi trường cộng đồng sôi động và hỗ trợ nhau trong việc phát triển phần mềm.

1.2.4. Figma

Figma là một công cụ thiết kế giao diện người dùng và cộng tác trực tuyến, được sử dụng rộng rãi trong ngành thiết kế sản phẩm số. Được phát triển bởi Figma, Inc., Figma cung cấp một nền tảng đa nền tảng và trực tuyến cho các nhà thiết kế, giúp họ tạo ra các giao diện người dùng đẹp và tương tác một cách dễ dàng.

Một số đặc điểm chính của Figma bao gồm:

- Giao diện trực quan: Figma có một giao diện sử dụng dễ dàng và trực quan, cho phép người dùng tạo và chỉnh sửa các thiết kế một cách nhanh chóng.

- Cộng tác đa người dùng: Figma cho phép nhiều người dùng làm việc trên cùng một thiết kế cùng một lúc, từ xa, thông qua tính năng cộng tác trực tuyến.
- Thiết kế đa nền tảng: Figma hỗ trợ thiết kế cho nhiều nền tảng, bao gồm web, di động và máy tính để bàn, giúp người dùng tạo ra giao diện thân thiện với người dùng trên mọi thiết bị.
- Thư viện và component: Figma cho phép người dùng tạo ra các thư viện và component tái sử dụng, giúp tăng tốc quá trình thiết kế và duy trì tính nhất quán trong toàn bộ dự án.
- Prototype và testing: Figma cung cấp tính năng prototype để tạo ra các bản demo tương tác của sản phẩm, cũng như tính năng testing để thu thập ý kiến từ người dùng.

Figma đã trở thành một công cụ quan trọng trong quy trình thiết kế sản phẩm số, được sử dụng bởi các nhóm thiết kế lớn và nhỏ trên khắp thế giới. Đặc biệt, tính linh hoạt và khả năng cộng tác trực tuyến của Figma đã làm cho nó trở thành lựa chọn hàng đầu cho các dự án phát triển phần mềm và sản phẩm trực tuyến.

1.2.5. Bug

Bug là một thuật ngữ phổ biến trong lĩnh vực công nghệ thông tin, đặc biệt là trong phát triển phần mềm. Một bug là một lỗi hoặc hành vi không mong muốn trong phần mềm mà gây ra các vấn đề hoặc nguy cơ cho hoạt động bình thường của chương trình. Đây có thể là kết quả của lỗi lập trình, thiết kế không chính xác hoặc các yếu tố khác như môi trường hoạt động không như dự kiến.

Bug có thể xuất hiện ở mọi giai đoạn trong quá trình phát triển phần mềm, từ việc viết mã nguồn ban đầu, kiểm thử, triển khai cho đến khi ứng dụng hoạt động trên môi trường sản phẩm. Các bug có thể làm ứng dụng gặp sự cố, crash, hoặc làm giảm hiệu suất hoạt động của chương trình.

Quản lý bug là một phần quan trọng của quy trình phát triển phần mềm, và các nhà phát triển thường sử dụng các công cụ quản lý bug để ghi lại, theo dõi và giải quyết các vấn đề. Quy trình kiểm thử phần mềm thường bao gồm việc phát hiện và báo cáo các bug, sau đó là sửa chữa chúng trước khi phần mềm được phát hành.

Mặc dù bug có thể gây ra rắc rối và trở ngại trong quá trình phát triển, nhưng việc phát hiện và sửa chữa chúng là một phần quan trọng của việc cải thiện chất lượng phần mềm và cung cấp trải nghiệm người dùng tốt hơn.

CHƯƠNG 2. UI/UX

2.1. Giới thiệu về Figma

Figma là một công cụ thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) đa nền tảng và trực tuyến, phổ biến trong ngành công nghiệp phát triển phần mềm và thiết kế sản phẩm số. Với giao diện trực quan và dễ sử dụng, Figma cho phép người dùng tạo, chỉnh sửa và chia sẻ các thiết kế một cách linh hoạt và hiệu quả, mà không cần phải cài đặt phần mềm hay định cụ thể vị trí làm việc.

Công cụ này cũng nổi bật với khả năng cộng tác đa người dùng, cho phép các nhóm làm việc trên cùng một thiết kế cùng một lúc, từ xa. Điều này giúp tăng cường sự hợp tác giữa các thành viên trong đội ngũ thiết kế và phát triển. Hơn nữa, Figma hỗ trợ thiết kế cho nhiều nền tảng, bao gồm web, di động và máy tính để bàn, giúp đảm bảo rằng các thiết kế có thể được xem và chỉnh sửa trên mọi thiết bị.

Ngoài ra, Figma cung cấp các tính năng mạnh mẽ như tạo thư viện và component tái sử dụng, tạo ra các bản prototype tương tác để kiểm tra trải nghiệm người dùng, và các công cụ testing để thu thập phản hồi từ người dùng. Tất cả những điều này giúp Figma trở thành một công cụ không thể thiếu trong quy trình thiết kế sản phẩm số, từ việc xây dựng wireframe và mockup cho đến việc phát triển và triển khai sản phẩm cuối cùng.

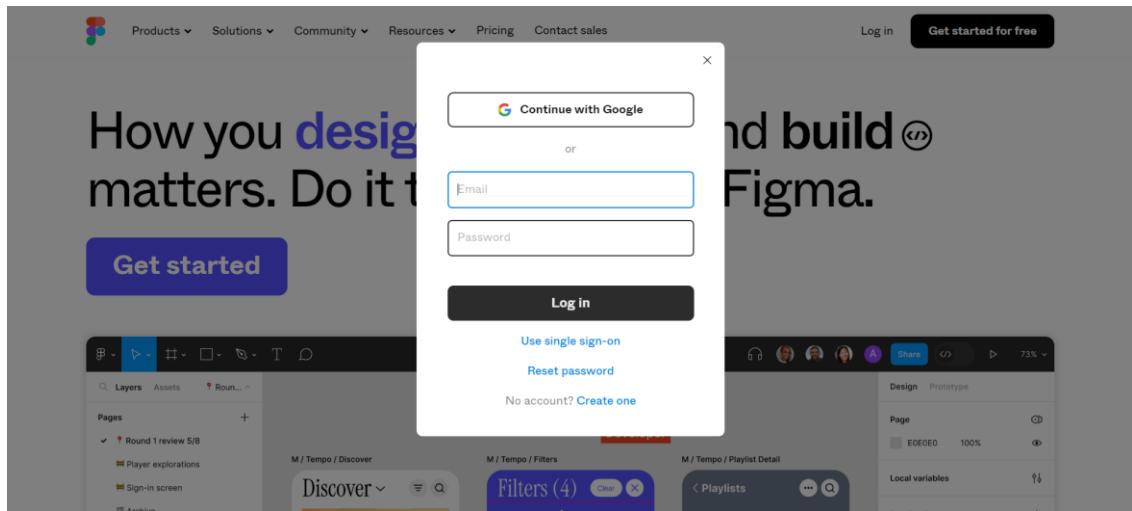
Link Figma:

<https://www.figma.com/file/78W6XmSRGv7JhU4xrRC9n/%C4%90%E1%BB%93-%C3%A1n--C%C3%B4ng-c%E1%BB%A5-v%C3%A0-ph%C3%A1t-tri%E1%BB%83n-ph%E1%BA%A7n-m%E1%BB%81m?type=design&node-id=0%3A1&mode=dev&t=luqKUbytK37iELgz-1>

2.2. Các bước thực hiện

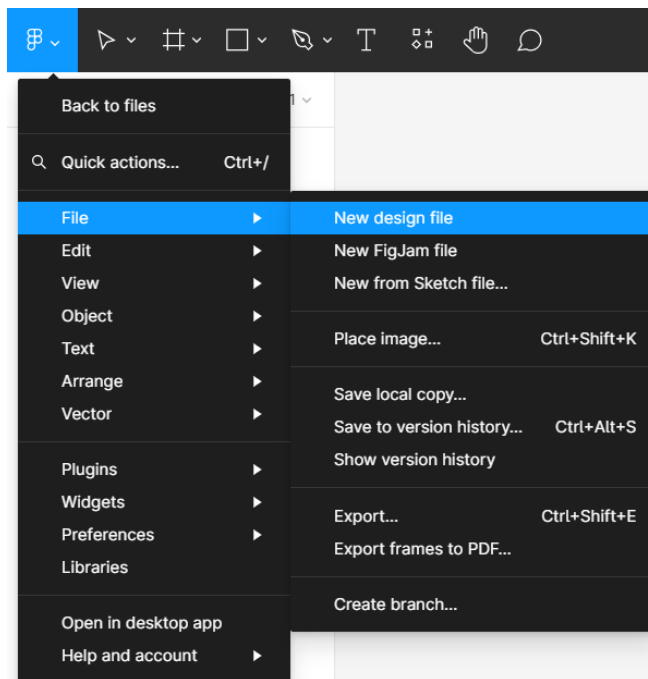
2.2.1. Đăng nhập và tạo dự án

Đăng nhập vào tài khoản Figma hoặc đăng ký nếu bạn chưa có tài khoản.



Hình 2.1. Đăng nhập tài khoản Figma

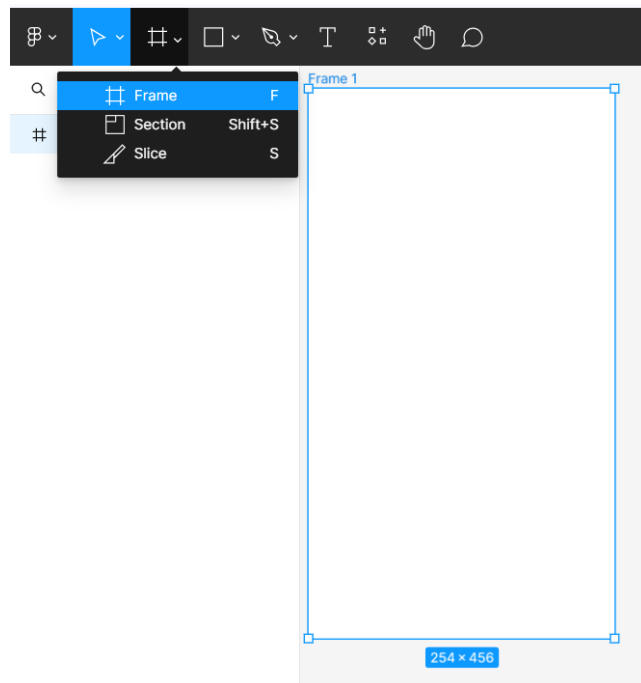
Tạo dự án mới bằng cách nhấp vào “File” và chọn “New design file”.



Hình 2.2. Tạo file thiết kế mới

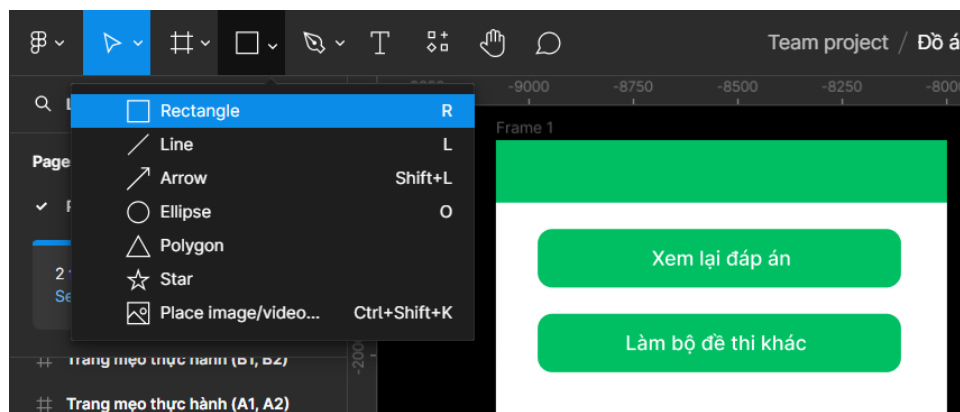
2.2.2. Tạo Wireframe và Bố cục ban đầu:

Bắt đầu với việc tạo wireframe để định hình cấu trúc tổng thể của ứng dụng.



Hình 2.3. Sử dụng Frame

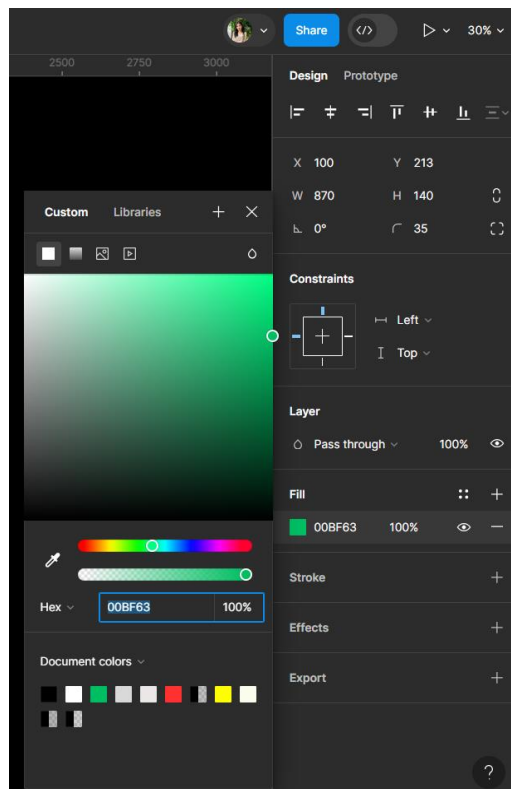
Sử dụng các công cụ như khung hình, hình chữ nhật và văn bản để tạo ra bố cục ban đầu của trang web hoặc ứng dụng di động.



Hình 2.4. Sử dụng các công cụ như khung hình, hình chữ nhật và văn bản

2.2.3. Thiết kế giao diện người dùng

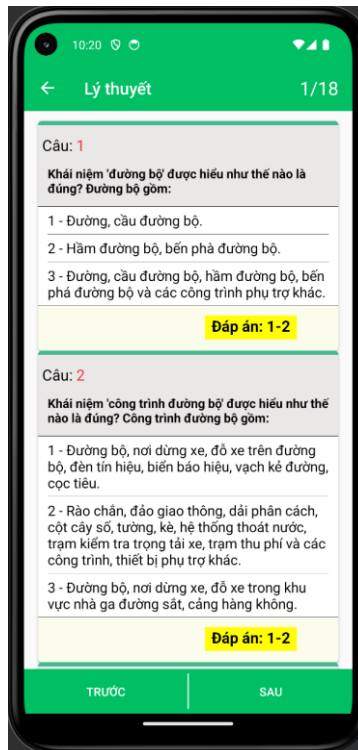
Bắt đầu từ wireframe, bổ sung các yếu tố giao diện như nút, ô nhập liệu, menu, v.v. Sử dụng các công cụ vẽ và tùy chỉnh để tạo ra các phần tử giao diện với hình dạng, kích thước và màu sắc phù hợp.



Hình 2.5. Tùy chỉnh màu sắc



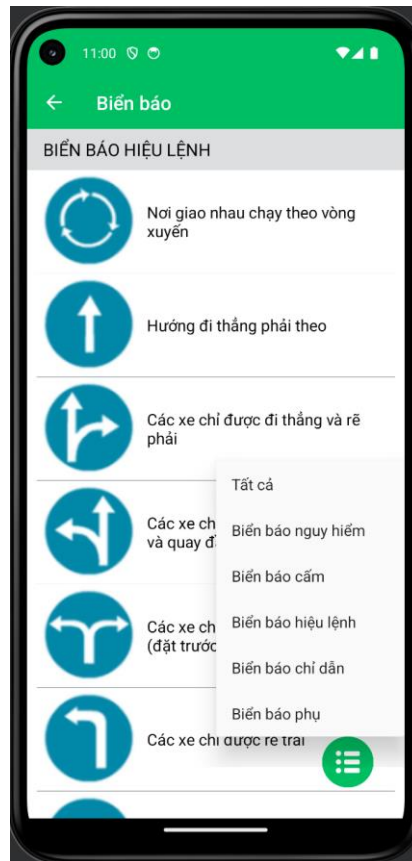
Hình 2.6. Hình ảnh trang chủ



Hình 2.7. Hình ảnh trang lý thuyết



Hình 2.8. Hình ảnh trang biển báo cấm



Hình 2.9. Hình ảnh trang biển báo hiệu lệnh



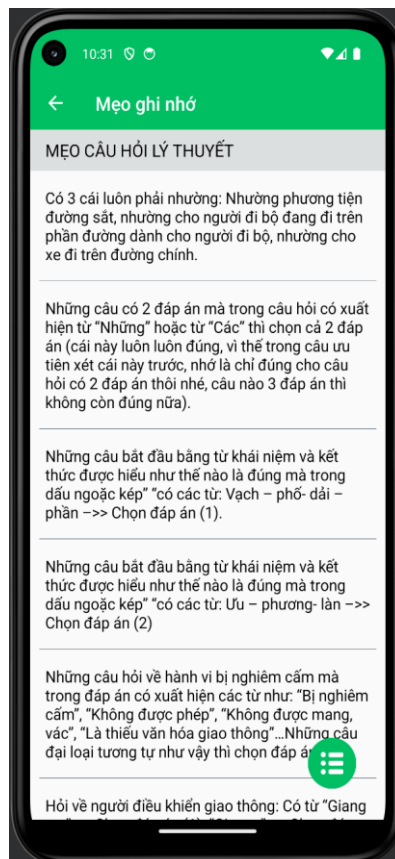
Hình 2.10. Hình ảnh trang chỉ dẫn



Hình 2.11. Hình ảnh trang biển báo nguy hiểm



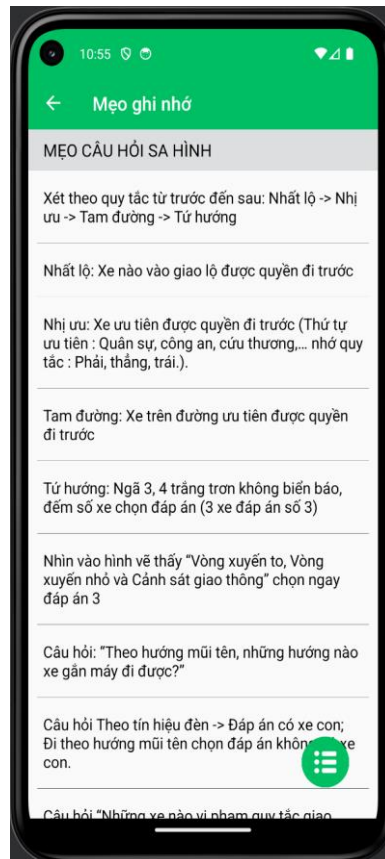
Hình 2.12. Hình ảnh trang biển báo phụ



Hình 2.13. Hình ảnh trang mẹo ghi nhớ lý thuyết



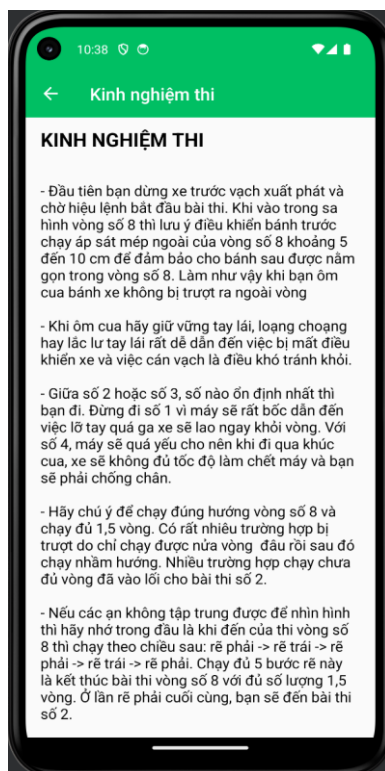
Hình 2.14. Hình ảnh trang mẹo ghi nhớ biển báo



Hình 2.15. Hình ảnh trang mẹo câu hỏi sa hình



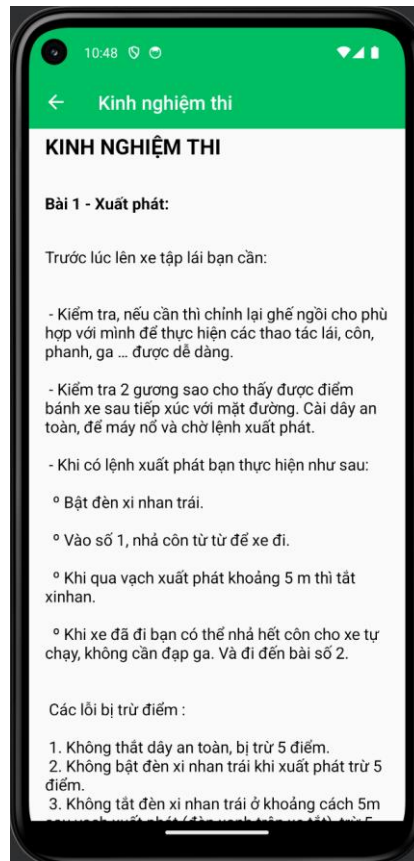
Hình 2.16. Hình ảnh trang mẹo thực hành A1,A2



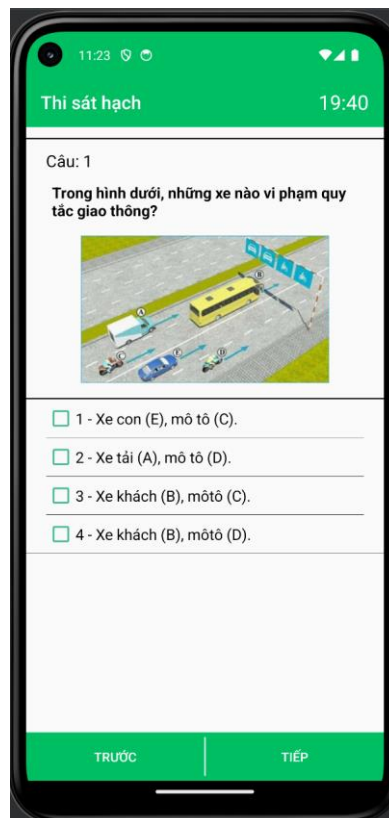
Hình 2.17. Hình ảnh trang kinh nghiệm thực hành A1,A2



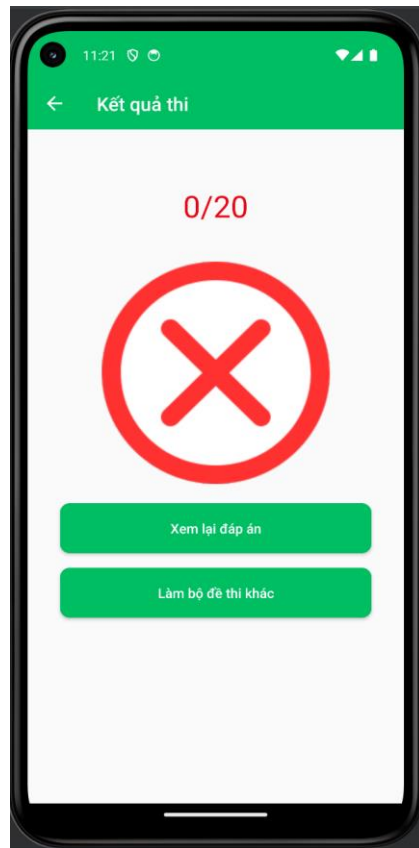
Hình 2.18. Hình ảnh trang mẹo thực hành B1,B2



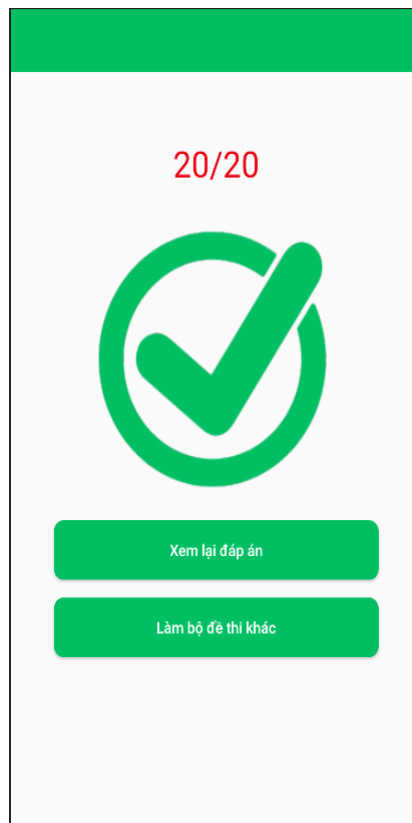
Hình 2.19. Hình ảnh trang kinh nghiệm thực hành B1,B2



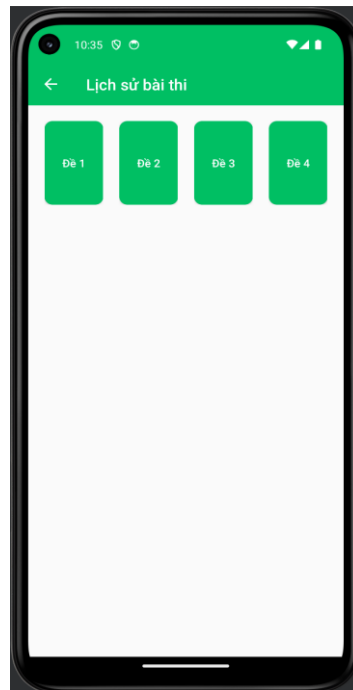
Hình 2.20. Hình ảnh giao diện trang thi sát hạch



Hình 2.21. Hình ảnh thi sát hạch không đạt



Hình 2.22. Hình ảnh thi sát hạch đạt



Hình 2.23. Hình ảnh trang lịch sử bài thi



Hình 2.24. Hình ảnh đáp án bài thi đã thi

CHƯƠNG 3. GITHUB

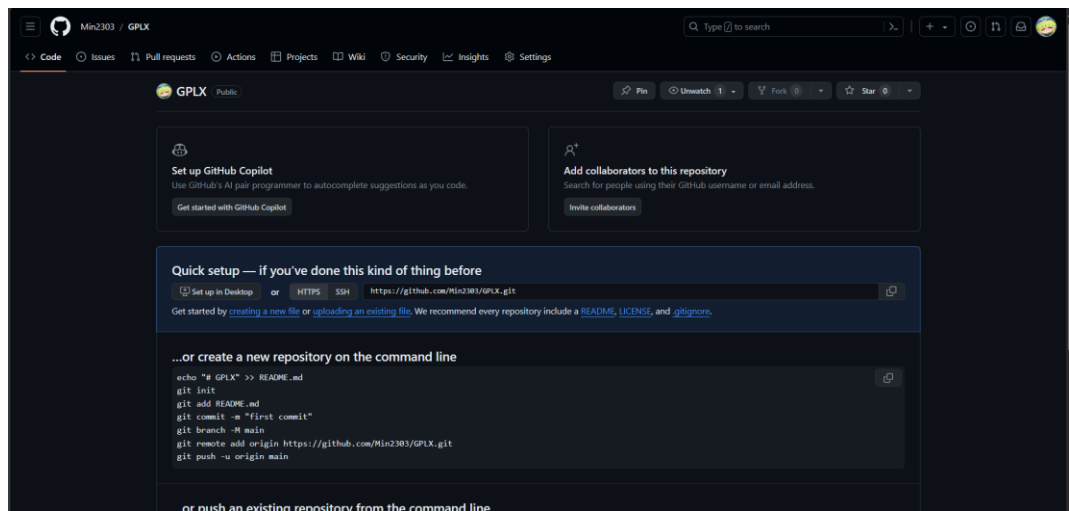
3.1. Giới thiệu

Dự án này sử dụng Git làm công cụ quản lý mã nguồn, với mã nguồn được lưu trữ trên GitHub, cũng như các nền tảng khác như GitLab và Bitbucket.

Link Github dự án: https://github.com/hduyen1511/App_OnThiGPLX.git

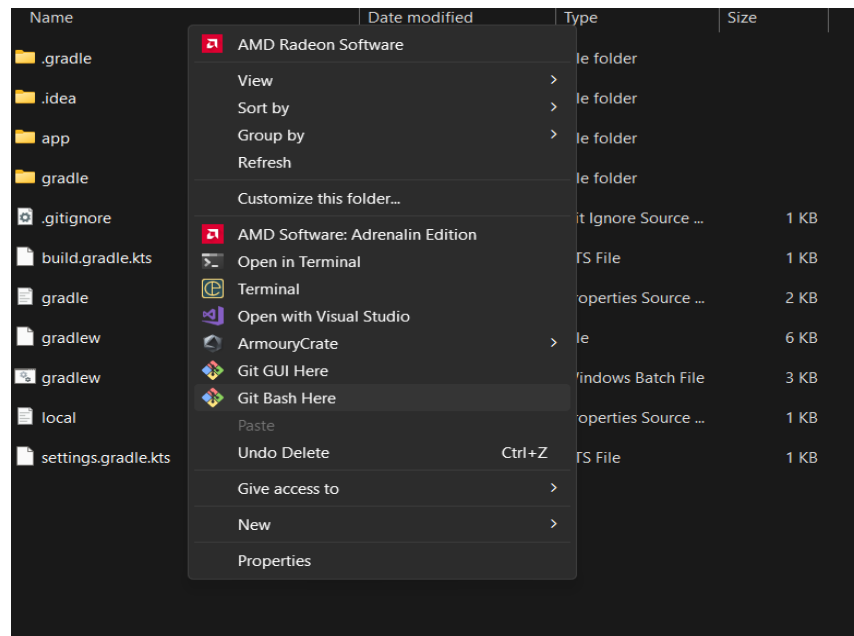
3.2. Các bước triển khai

Tạo thư mục repository trên server.



Hình 3.1. Tạo thư mục repository trên server.

Chọn Git Bash here ở thư mục cần tải lên để thực hiện các bước kế tiếp.



Hình 3.2. Chọn Git Bash here ở thư mục cần tải lên để thực hiện các bước kế tiếp.

Sử dụng câu lệnh git init - Khởi tạo một kho lưu trữ Git mới trong thư mục hiện tại. Điều này tạo ra một thư mục con mới tên là '.git' chứa tất cả các tệp cần thiết cho kho lưu trữ Git và thiết lập thư mục hiện tại làm thư mục làm việc cho kho lưu trữ mới.

```

MINGW64:/d/DoAnChuyenNganh/GPLX/GPLX
ASUS@LAPTOP-FTE5I6H5 MINGW64 /d/DoAnChuyenNganh/GPLX/GPLX
$ git init
Initialized empty Git repository in D:/DoAnChuyenNganh/GPLX/GPLX/.git/
ASUS@LAPTOP-FTE5I6H5 MINGW64 /d/DoAnChuyenNganh/GPLX/GPLX (master)
$ |

```

Hình 3.3. Sử dụng câu lệnh git init

Sử dụng câu lệnh “ git add README.md “ để thêm tệp 'README.md' vào khu vực staging (hoặc index). Khu vực staging là nơi bạn chuẩn bị các tệp cần được commit vào kho lưu trữ. Bằng cách này, 'README.md' được đánh dấu để ghi lại các thay đổi trong lần commit tiếp theo.

```

ASUS@LAPTOP-FTE5I6H5 MINGW64 /d/DoAnChuyenNganh/GPLX/GPLX (master)
$ git add README.md
fatal: pathspec 'README.md' did not match any files

```

Hình 3.4. Sử dụng câu lệnh git add README.md

Sử dụng câu lệnh “git commit -m "first commit"” để thực hiện một commit với các tệp đã được thêm vào khu vực staging, ở đây là 'README.md'. Lần

commit này được gắn với thông điệp "first commit". Một commit là một bản ghi của các thay đổi trong kho lưu trữ, cho phép bạn theo dõi lịch sử sửa đổi

```
ASUS@LAPTOP-FTE5I6H5 MINGW64 /d/DoAnChuyenNganh/GPLX/GPLX (master)
$ git commit -m "first commit"
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        .idea/
        app/
        build.gradle.kts
        gradle.properties
        gradle/
        gradlew
        gradlew.bat
        settings.gradle.kts

nothing added to commit but untracked files present (use "git add" to track)
```

Hình 3.5. Sử dụng câu lệnh “git commit -m “first commit””

Sử dụng câu lệnh “git branch -M main” để Đổi tên nhánh hiện tại thành ‘main’. Trong nhiều trường hợp, Git bắt đầu với nhánh mặc định là ‘master’, và câu lệnh này đảm bảo rằng nhánh chính được gọi là ‘main’.

```
ASUS@LAPTOP-FTE5I6H5 MINGW64 /d/DoAnChuyenNganh/GPLX/GPLX (master)
$ git branch -M main
```

Hình 3.6. Sử dụng câu lệnh “git branch -M main”

Sử dụng câu lệnh “git remote add origin <https://github.com/Min2303/GPLX.git>” để Thêm một remote repository mới có tên là ‘origin’ vào kho lưu trữ Git của bạn. Địa chỉ URL chỉ định địa điểm của remote repository trên GitHub. Remote này là nơi bạn sẽ đẩy các thay đổi từ kho lưu trữ cục bộ của mình.

```
ASUS@LAPTOP-FTE5I6H5 MINGW64 /d/DoAnChuyenNganh/GPLX/GPLX (main)
$ git remote add origin https://github.com/Min2303/GPLX.git
```

Hình 3.7. Sử dụng câu lệnh git remote add origin.

Dùng câu lệnh “git push -u origin main” để Đẩy các thay đổi từ nhánh ‘main’ của kho lưu trữ cục bộ lên nhánh ‘main’ của kho lưu trữ remote tại ‘origin’ (GitHub trong trường hợp này). Cờ ‘-u’ nghĩa là thiết lập nhánh ‘main’ trên ‘origin’ như nhánh upstream mặc định cho nhánh ‘main’ cục bộ, điều này giúp cho các lần

push sau có thể đơn giản chỉ cần gọi `git push` mà không cần chỉ định nhánh hay remote.

The screenshot shows a GitHub repository for user 'hduyen1511'. The commit history table lists several files, with the most recent commit being 'Update README.md' by 'c6a67f3 · now' with 3 commits. Below the table, the README file is displayed, containing a list of team members.

File	Commit Message	Time
.idea	Do an chuyen nganh	3 weeks ago
app	Do an chuyen nganh	3 weeks ago
gradle/wrapper	Do an chuyen nganh	3 weeks ago
.gitignore	Do an chuyen nganh	3 weeks ago
README.md	Update README.md	now
build.gradle.kts	Do an chuyen nganh	3 weeks ago
gradle.properties	Do an chuyen nganh	3 weeks ago
gradlew	Do an chuyen nganh	3 weeks ago
gradlew.bat	Do an chuyen nganh	3 weeks ago
settings.gradle.kts	Do an chuyen nganh	3 weeks ago

README

Thành Viên

Kiều Thị Hà Duyên _ 2011064200

Đàng Anh Min _ 2011068866

Phan Thị Anh _ 2011065095

Hình 3.8. Hình ảnh push lên thành công.

CHƯƠNG 4. CÔNG CỤ QUẢN LÝ DỰ ÁN (JIRA)

4.1. Giới thiệu

Jira là một hệ thống quản lý dự án và theo dõi công việc mạnh mẽ, được phát triển bởi Atlassian và rất phổ biến trong cộng đồng phát triển phần mềm và quản lý dự án. Với Jira, người dùng có thể tạo, gán và theo dõi các công việc, nhiệm vụ và vấn đề một cách có cấu trúc.

Nền tảng này cung cấp các công cụ linh hoạt cho việc quản lý dự án, từ việc tổ chức công việc và sprints đến việc lập lịch và theo dõi tiến độ. Không chỉ dừng lại ở việc quản lý công việc, Jira còn là nơi lưu trữ và quản lý tất cả các vấn đề, lỗi và yêu cầu tính năng của dự án. Điều này giúp đảm bảo rằng mọi thành viên trong nhóm đều có cái nhìn tổng quan và có thể làm việc hiệu quả hơn. Bằng cách tùy chỉnh quy trình làm việc và sử dụng các công cụ báo cáo và phân tích, Jira giúp nhóm làm việc cải thiện hiệu suất và tối ưu hóa quy trình làm việc của mình.

Tóm lại, Jira là một công cụ quan trọng giúp tăng cường sự hợp tác và hiệu suất làm việc trong các dự án phát triển phần mềm và quản lý dự án.

4.2. Các tính năng

4.2.1. Tạo project

Khi bạn bắt đầu một dự án mới trong Jira, việc đầu tiên là tạo một project. Mỗi project đại diện cho một lĩnh vực công việc cụ thể, chẳng hạn như phát triển phần mềm, quản lý dự án, hoặc tiếp thị. Trong quá trình tạo project, bạn có thể chọn loại project phù hợp, thiết lập quyền truy cập cho các thành viên của nhóm, và tùy chỉnh cài đặt theo nhu cầu cụ thể của dự án. Việc này giúp định rõ phạm vi và mục tiêu của dự án từ đầu và tạo ra một nền tảng cơ sở để quản lý công việc một cách hiệu quả.

4.2.2. Tạo board

Boards trong Jira là giao diện hiển thị và quản lý các công việc dựa trên các loại khác nhau, như scrum board, kanban board, hay agile board. Khi tạo một board mới, bạn có thể chọn loại board phù hợp với cách làm việc của nhóm và tùy chỉnh cài đặt để

phản ánh các quy trình và quy tắc làm việc. Boards cung cấp một cách trực quan để theo dõi công việc, phản hồi nhanh chóng và điều chỉnh quy trình làm việc theo thời gian, giúp đảm bảo sự linh hoạt và hiệu quả trong quản lý dự án.

4.2.3. Tạo issues

Trong Jira, mọi công việc, vấn đề, nhiệm vụ, lỗi hoặc yêu cầu tính năng đều được biểu diễn dưới dạng "issues". Việc tạo issues mới cho phép bạn mô tả chi tiết vấn đề hoặc công việc cần thực hiện, gán chúng cho các thành viên trong nhóm, đặt ưu tiên và thiết lập hạn chót. Điều này giúp đảm bảo rằng mọi người trong nhóm đều biết rõ các công việc cần hoàn thành và tiến độ của chúng.

4.2.4. Tìm issues trong Jira

Trong quá trình quản lý dự án, việc tìm kiếm và truy cập các issues quan trọng là một phần không thể thiếu. Jira cung cấp một công cụ tìm kiếm mạnh mẽ cho phép người dùng lọc và tìm kiếm issues dựa trên nhiều tiêu chí khác nhau như trạng thái, người gán, ưu tiên, và nhiều hơn nữa. Bạn có thể sử dụng các từ khóa, cú pháp tìm kiếm nâng cao, hoặc sử dụng các bộ lọc được xây dựng sẵn để nhanh chóng định vị và truy cập các issues một cách dễ dàng.

4.2.5. Workflow trong Jira

Workflow là một phần quan trọng của quản lý dự án trong Jira, vì nó xác định các quy trình và quy tắc mà các issues phải đi qua từ khi chúng được tạo ra đến khi hoàn thành. Jira cung cấp các workflow mặc định cho các loại dự án phổ biến như phát triển phần mềm, quản lý dự án, và tiếp thị. Tuy nhiên, bạn cũng có thể tùy chỉnh hoặc tạo mới các workflow để phản ánh quy trình làm việc cụ thể của dự án. Bằng cách xác định các trạng thái, quy tắc di chuyển, và hành động tự động, bạn có thể định rõ các bước cần thiết để hoàn thành một công việc và tối ưu hóa quy trình làm việc của nhóm.

4.2.6. Tạo Workflow

Một trong những điểm mạnh của Jira là khả năng tạo và tùy chỉnh các workflow theo nhu cầu cụ thể của dự án. Bằng cách sử dụng giao diện người dùng thân thiện, bạn có thể tạo mới hoặc sửa đổi các workflow một cách dễ dàng. Bạn có thể xác định các

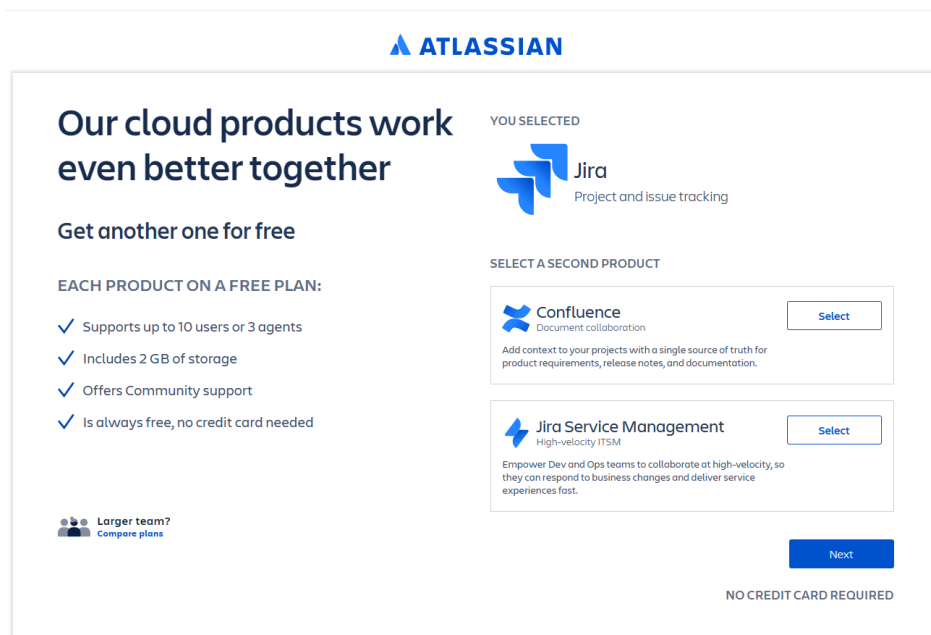
trạng thái, quy tắc chuyển đổi, và hành động tự động mà issues sẽ trải qua trong quy trình làm việc. Điều này giúp bạn tùy chỉnh quy trình làm việc để phản ánh đúng nhu cầu và luồng công việc của nhóm.

4.3. Hướng dẫn sử dụng

4.3.1. Cách tạo một Jira site

Các bước để tạo Jira site như sau:

- Truy cập vào đường link [Jira | Issue & Project Tracking Software | Atlassian](https://jira.atlassian.com)
- Chọn “Get it free” trên góc phải màn hình. Bạn sẽ thấy màn hình dưới đây. Chọn “Next”.



Hình 4.1. Get it free.

- Tạo tài khoản Atlassian. Bạn có thể tạo tài khoản bằng cách nhập email làm việc hoặc qua tài khoản Google của bạn.



Welcome back, 4200_Kiều

Free for up to 10 users - no credit card required.

Work email

kduyen1511@gmail.com

[Sign in with a different Atlassian account](#)

Your site

kduyen1511-1713168011309

.atlassian.net



I agree to the [Atlassian Customer Agreement](#), which incorporates by reference the [AI Product-Specific Terms](#), and acknowledge the [Privacy Policy](#).

Agree and start now

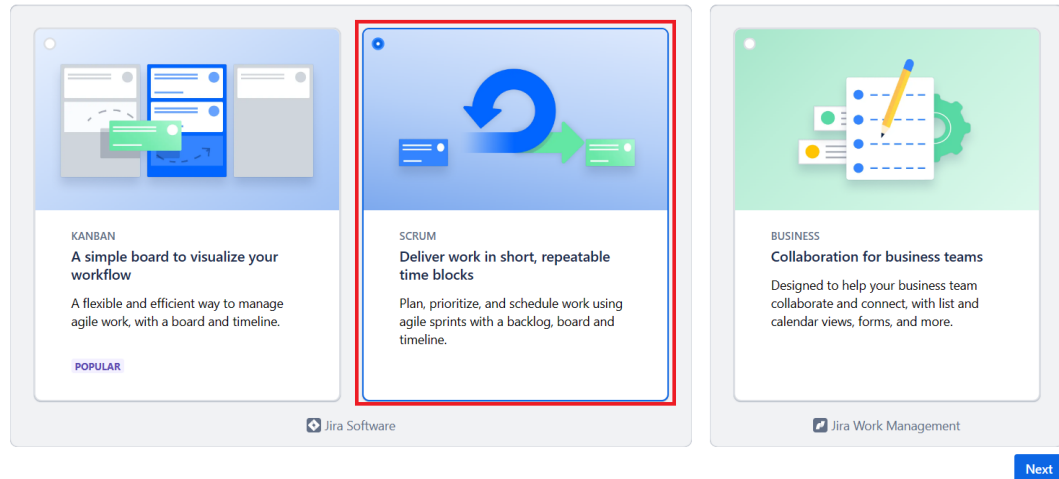


Hình 4.2. Tạo tài khoản bằng email

- Sau khi tạo site, bạn sẽ được yêu cầu tạo project (dự án) đầu tiên. Bạn có thể chọn Kanban hoặc Scrum, loại team-managed (quản lý nhóm) hoặc company-managed (quản lý công ty) tùy theo nhu cầu sử dụng.

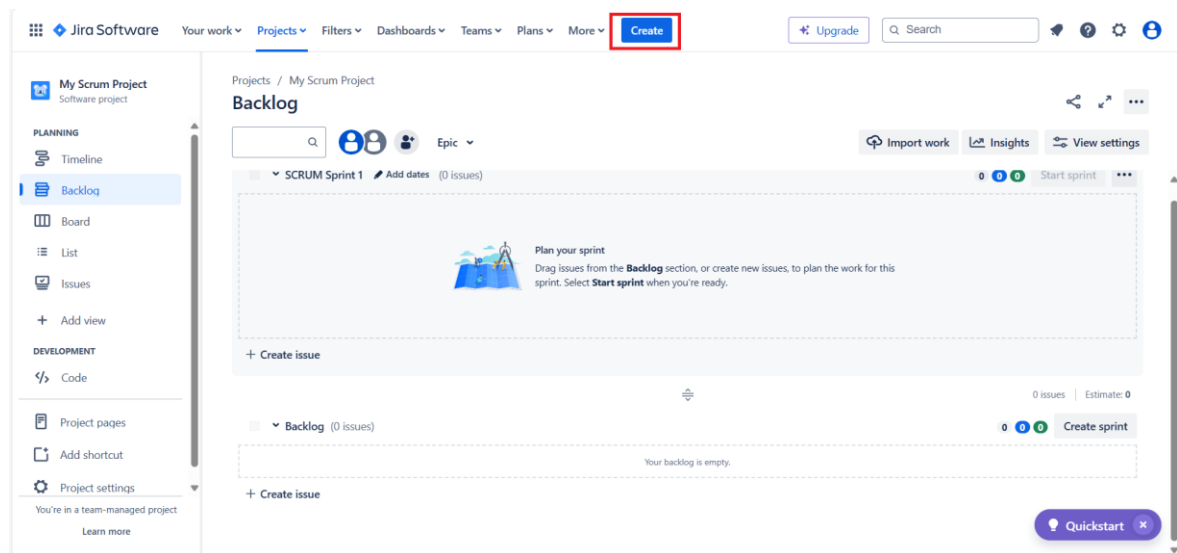
Select a template for your first project

If you're not sure what to choose, don't worry. You can quickly create a new project if this one's not right for you.



Hình 4.3. Chọn template Kanban hoặc Scrum

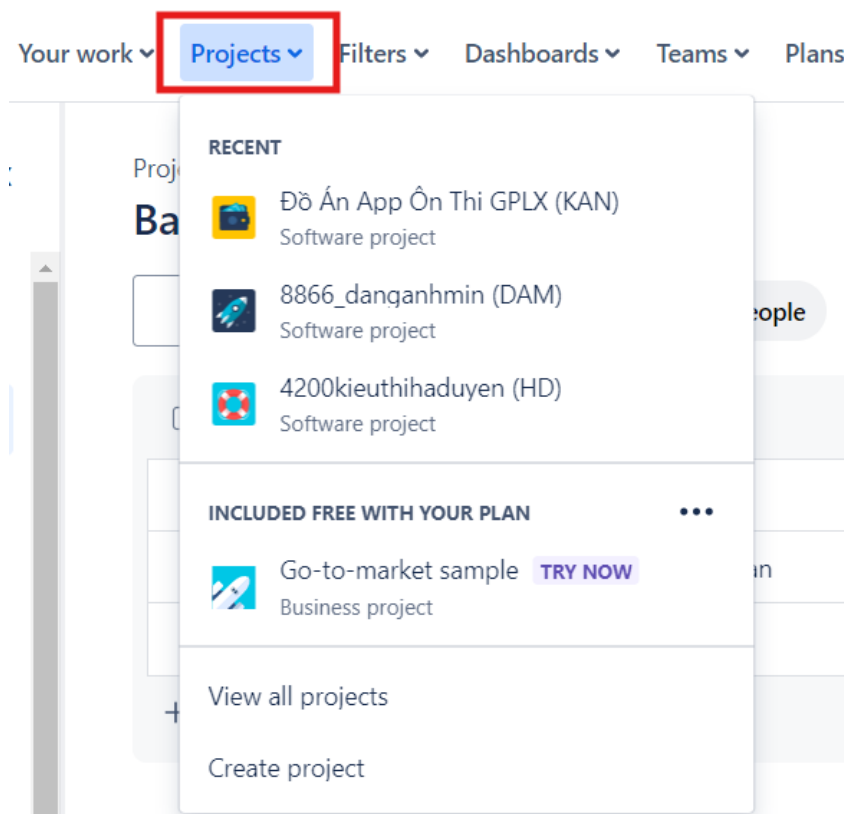
- Vậy là bạn đã tạo thành công một Jira site và có thể thử tạo các issue (vấn đề, công việc) cần xử lý bằng cách ấn vào nút “Create” như trong hình dưới đây.



Hình 4.4. Tạo một issue mới trên Jira Software

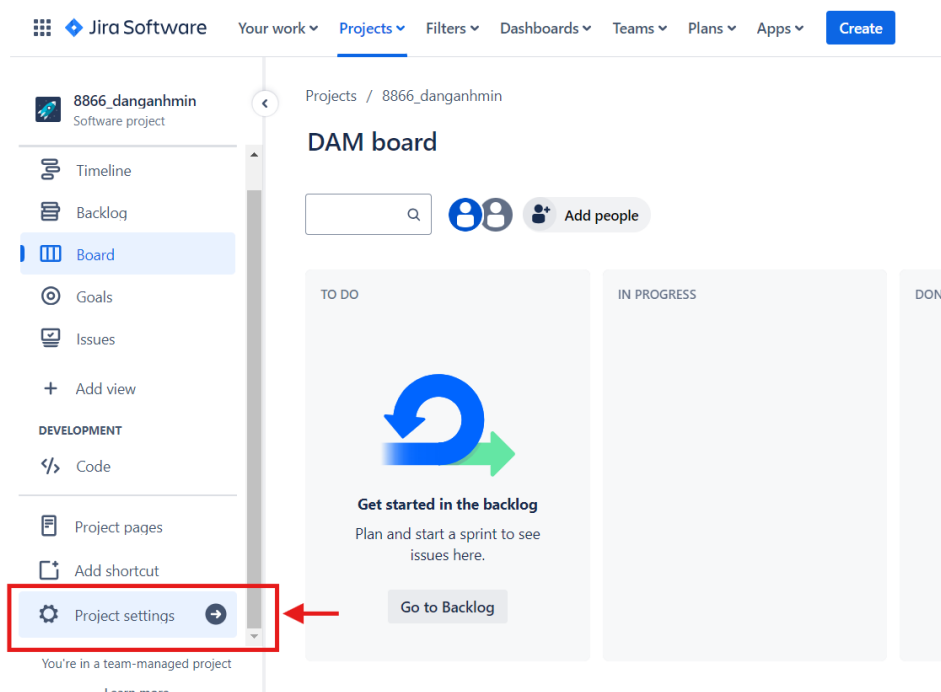
4.3.2. Các thao tác với project

Xem danh sách project hiện tại: Ấn vào “Project” trên thanh menu như trong hình. Từ mục project trong menu bar sẽ xổ xuống danh sách một số project mà bạn đang tham gia. Bạn cũng có thể chọn “View all projects” để xem tất cả project hoặc chọn “Create project” để tạo một project mới.



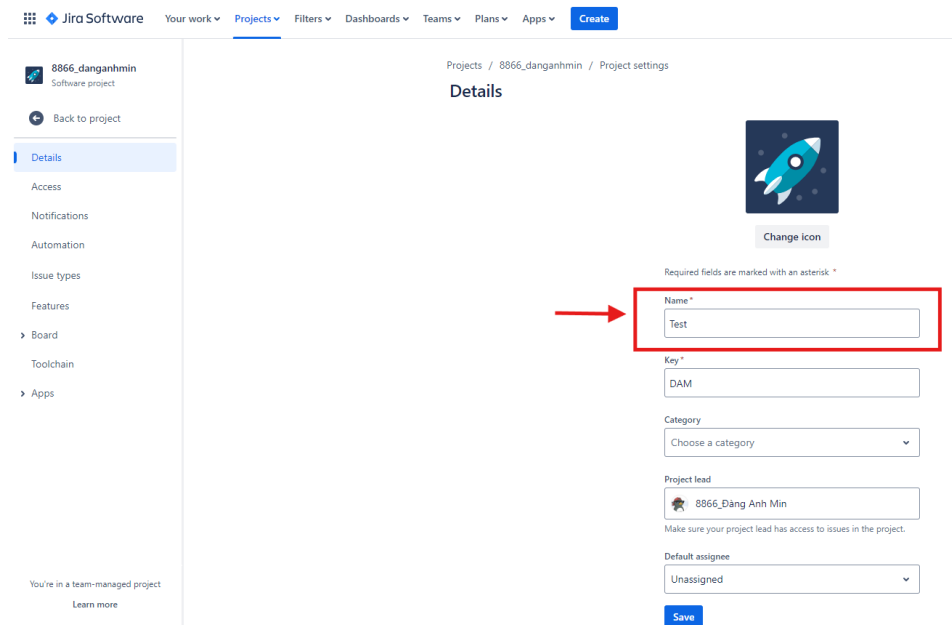
Hình 4.5. Xem danh sách project

Sửa tên, xóa project: Có thể thay đổi một số cài đặt cơ bản của project tại mục Project Setting.



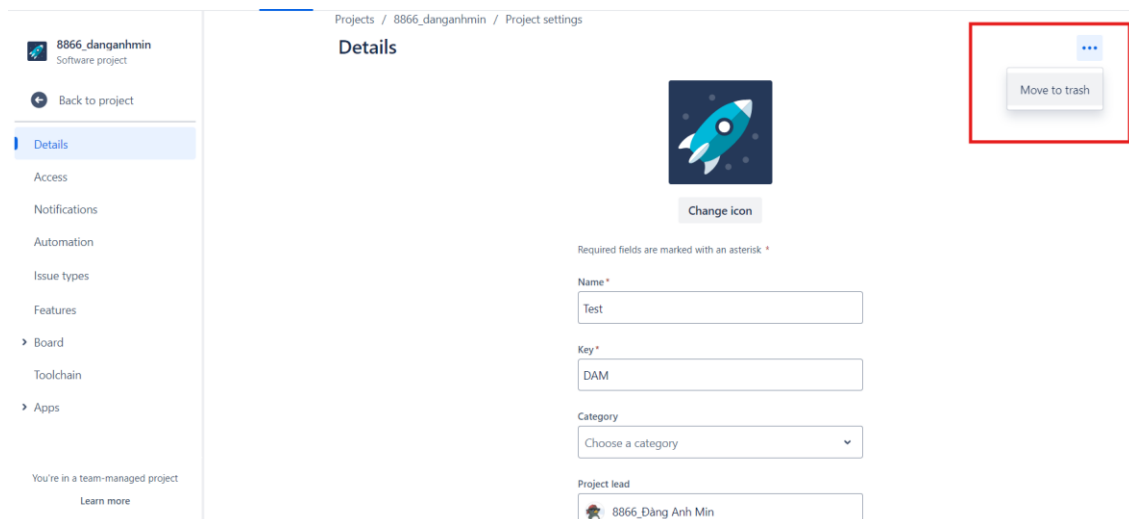
Hình 4.6. Cài đặt project

Sau đó sửa các thông tin cơ bản như Name (tên dự án), icon dự án, category (phân loại) tại mục Details.



Hình 4.7. Sửa tên project Jira

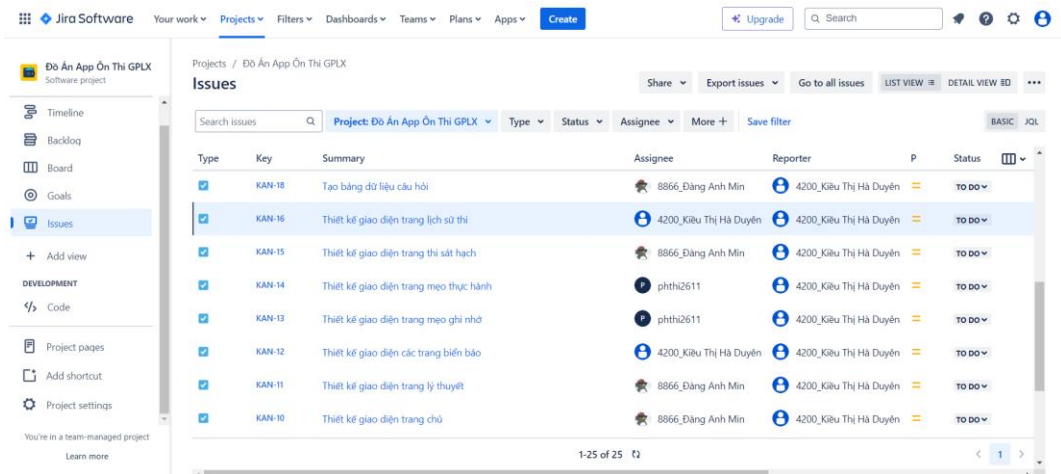
Hoặc xóa project bằng cách click vào dấu ba chấm (“...”) tại góc phải màn hình và chọn “Move to trash”.



Hình 4.8. Xóa project trên Jira

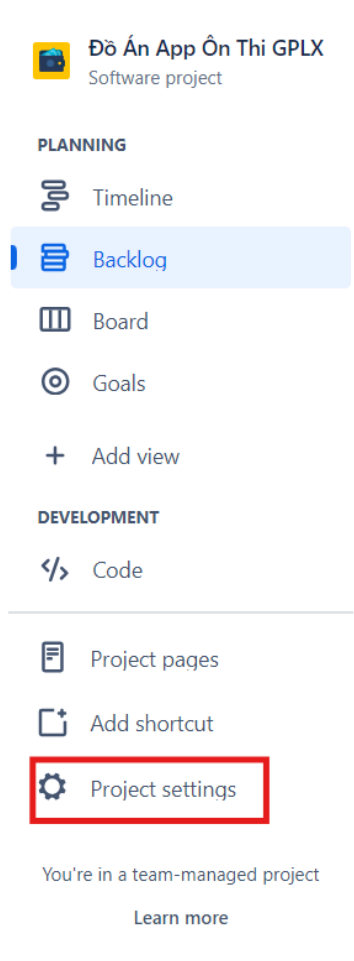
4.3.3. Các thao tác với issue

Xem danh sách các issue: Trên side bar bên trái, click vào “issue” để xem danh sách các issue.

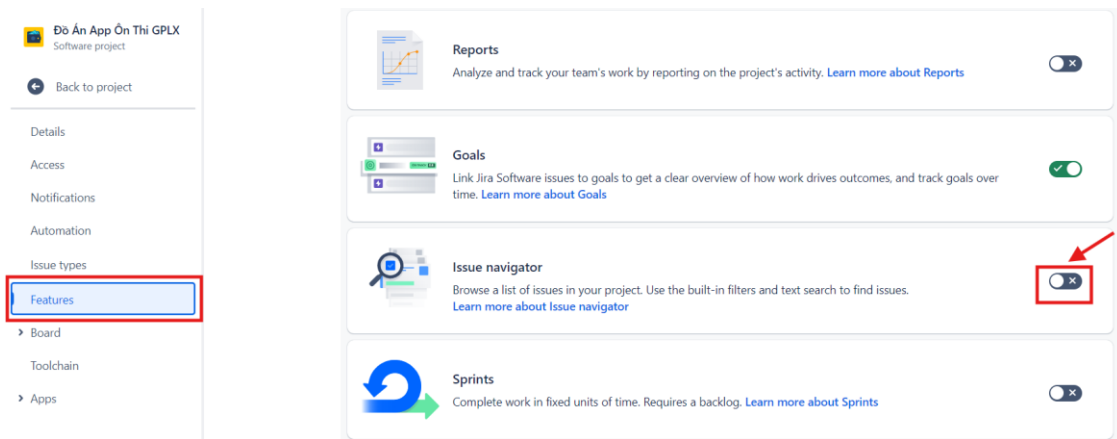


Hình 4.9. Xem danh sách issue

Nếu không thấy mục “issue” trên side bar, hãy bật tính năng “issue navigation” bằng cách chọn “Project setting” → “Feature” và bật để button ở mục “Issue navigation” chuyển thành màu xanh. Tại đây, bạn cũng có thể bật tính năng tạo báo cáo của project bằng cách chuyển button ở mục “Report” thành màu xanh.



Hình 4.10. Hình ảnh không thấy mục “issue” trên side bar



Hình 4.11. Bật issue navigator

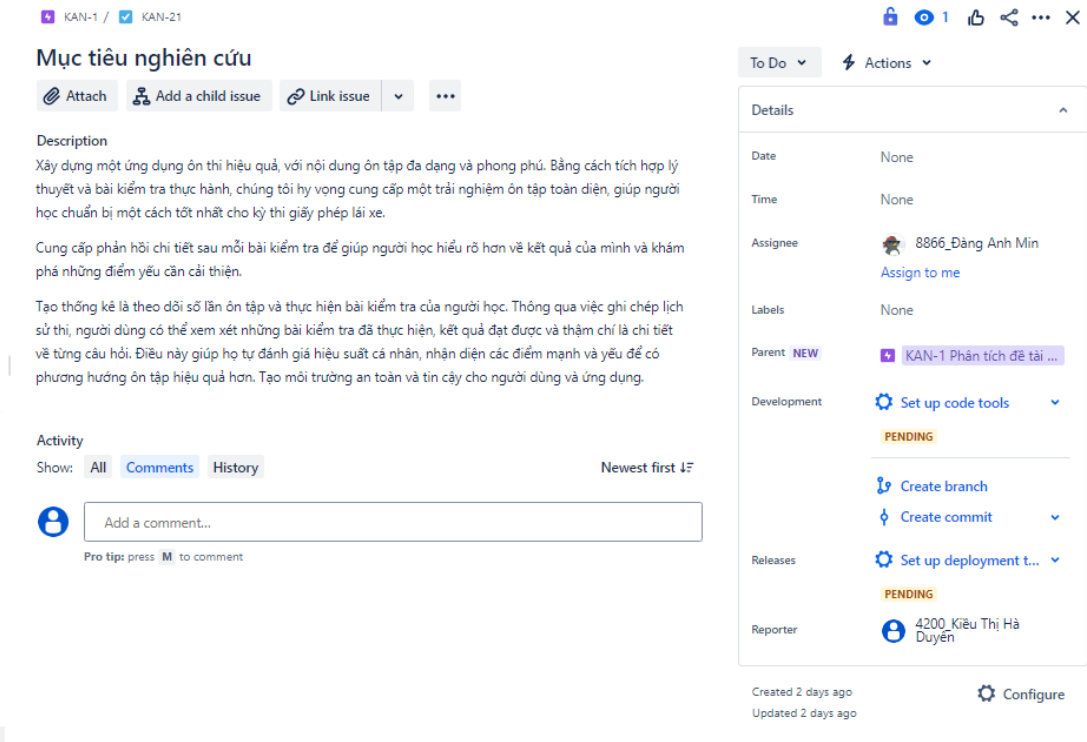
Tạo issue mới: Như đã nói ở trên, ấn vào nút “Create” để tạo một issue mới, màn hình tạo issue như dưới đây sẽ xuất hiện.

Hình 4.12. Tạo issue mới

Chú ý: Các mục có dấu (*) là các trường bắt buộc.

- Tùy theo customize Jira của công ty mà các mục hiển thị khi tạo mới một issue sẽ khác. Tuy nhiên, về căn bản sẽ có các mục dưới đây:
- Project*: Chọn dự án mà issue này thuộc về.
- Issue Type*: Chọn loại issue. Các loại issue có thể có là: bug, task, epic,... (Loại issue này còn phụ thuộc vào customize Jira của công ty đang sử dụng để phù hợp với mô hình quản lý).
- Summary*: Mô tả tóm tắt cho issue mà bạn muốn tạo.
- Description*: Mô tả chi tiết về issue.
- Assignee*: Người thực hiện, giải quyết issue.
- Labels: Chọn nhãn hoặc đặt nhãn mới để dễ thống kê, tìm kiếm dễ dàng hơn. Labels có thể được đặt theo tên chức năng, tên nghiệp vụ, hoặc đối tượng khách hàng,...
- Reporter: Người kiểm duyệt. Hệ thống đặt mặc định là người tạo issue nhưng bạn có thể thay đổi sang người khác.
- Attachments: Tập thông tin đính kèm nếu có.
- Link issue: Lựa chọn issue có liên quan và mối quan hệ với issue đó.

Sau khi điền các thông tin liên quan đến issue, ấn “Create”. Thông tin trên một issue sẽ hiển thị với các mục bạn đã điền.



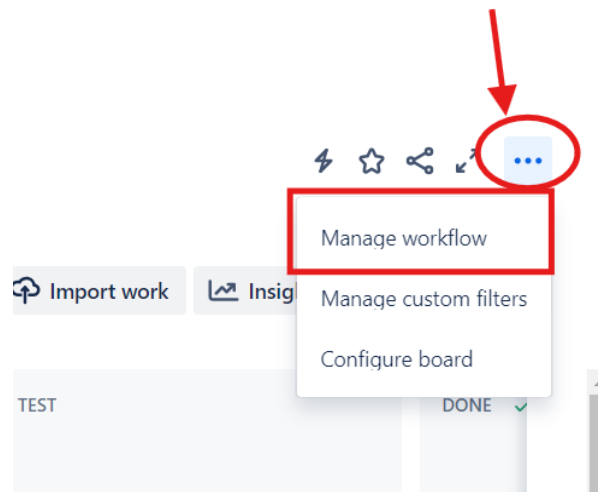
Hình 4.13. Thông tin issue hiển thị sau khi tạo một issue mới

Khi click vào một issue từ board, screen thông tin issue sẽ hiện ra như trên. Tại đây, bạn có thể sửa các thông tin liên quan đến issue mà mình đã điền từ trước. Hoặc thêm các field (trường) khác để mô tả issue được đầy đủ hơn bằng cách ấn vào “Configure” ở góc dưới bên trái của screen.

4.3.4. Các thao tác với Workflow

Cách tạo Workflow mới trong Jira tương đối đơn giản:

- Bắt đầu bằng cách nhấp vào biểu tượng ba chấm.
- Chọn “Manage workflow”
- Sử dụng các nút To-do status và transition để xác định các status và transition, và thêm chúng vào workflow của bạn. Bạn có thể xác định bao nhiêu thành phần này tùy thích.



Hình 4.14. Hướng dẫn tạo Workflow



Hình 4.15. Tạo mới Workflow

TÀI LIỆU THAM KHẢO

[1] Báo cáo đồ án tốt nghiệp lập trình Android.

https://www.academia.edu/39362717/Báo_cáo_lập_trình_Android_Đồ_án_tốt_nghiệp

[2] Hệ điều hành Android là gì?

<https://dienmaycholon.vn/kinh-nghiem-mua-sam/he-dieu-hanh-android-la-gi>.

[3] Kiến trúc của hệ điều hành Android.

<https://viblo.asia/p/kien-truc-cua-he-dieu-hanh-android-PaLGDYdaelX>.

[4] "SQLite là gì?" Viblo. <https://viblo.asia/p/sqlite-la-gi-E375zVVR5GW>

[5] Kiến trúc của hệ điều hành Android.

<https://viblo.asia/p/sqlite-la-gi-E375zVVR5GW>.

[6] Jira là gì? Hướng dẫn sử dụng các tính năng cơ bản của Jira Software cho người mới bắt đầu? <https://www.dssolution.vn/vi/jira-la-gi-huong-dan-su-dung-cac-tinh-nang-co-ban-cua-jira-software-cho-nguoi-moi-bat-dau/>

[7] Tổng quan về hệ điều hành Android.

<https://dinhnt.com/series/bai-1-tong-quan-ve-he-ieu-hanh-android-378/>

[8] Nguyễn Thị Mỹ Cúc, Bùi Thị Kim Vàng, Cao Đoàn Như Tuấn (2021). Thiết kế app di động luyện thi trắc nghiệm môn tiếng anh. Trường Đại Học Thủ Dầu Một.