**MỤC LỤC**

**LỜI CẢM ƠN**

**MỞ ĐẦU**

**CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU**

**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**CHƯƠNG 3. KIỂM TRA HOẠT ĐỘNG CỦA HỆ THỐNG**

**KẾT LUẬN**

**PHỤ LỤC**

**TÀI LIỆU THAM KHẢO**

LỜI CẢM ƠNMỞ ĐẦU

## Lý do chọn đề tài

Rò rỉ thông tin về sản phẩm trước khi ra mắt trong những năm trở lại đây không còn là vấn đề quá xa lạ. Có nhiều thương hiệu sử dụng việc này như là một chiêu để tiếp thị, gây sự hiếu kỳ, hấp dẫn khách hàng. Tuy nhiên, đa số tổ chức bị tiết lộ thông tin quan trọng này bởi sự tấn công, đánh cắp dữ liệu từ bên ngoài. Điều này mang lại những hậu quả nặng nề, làm mất lợi thế của doanh nghiệp trên thị trường.

Ở Việt Nam, tình trạng ăn cắp ý tưởng và thông tin sản phẩm xảy ra rất phổ biến. Để bảo vệ ý tưởng kinh doanh và sản phẩm của mình, doanh nghiệp có thể tiến hành đăng ký Sở hữu trí tuệ. Tuy nhiên, Trung tâm bản quyền, Cục sở hữu trí tuệ bộ Khoa học Công nghệ chỉ chấp nhận đăng ký sản phẩm cụ thể, thương hiệu đã hiện có. Đây là một bất lợi đối với doanh nghiệp, bởi nhiều trường hợp đánh cắp thông tin xảy ra khi sản phẩm vẫn chưa được giới thiệu ra thị trường.

Giống như hầu hết các thương hiệu lớn khác, Samsung cũng thường xuyên bị rò rỉ các thông tin sản phẩm từ sớm, thậm chí trước cả thời điểm ra mắt nhiều tháng. Lần gần nhất Samsung chịu ảnh hưởng nặng nề vì các tin đồn và hình ảnh rò rỉ loạt sản phẩm ra mắt hồi đầu tháng 8 năm 2022 như Galaxy Z Fold 3, Galaxy Z Flip 3, Galaxy Buds 2. Mặc dù Samsung đã không ít lần gửi thư nhắc nhở đến các cá nhân, đơn vị lan truyền các hình ảnh của sản phẩm mới, thậm chí xử phạt hành chính với các nhân viên làm rò rỉ thông tin theo quy định tại hợp đồng lao động, nhưng điều này có vẻ không hiệu quả. Các tổ chức, cá nhân có thể lợi dụng việc dùng thử nghiệm thiết bị để đánh cắp thông tin sản phẩm.

Qua thời gian thực tập và làm việc tại Samsung, cá nhân em nhận thấy việc bảo mật thông tin sản phẩm mới còn nhiều mặt hạn chế. Từ đó, em đưa ra giải pháp phát triển ứng dụng tên là *Giải pháp bảo mật thiết bị* để giảm thiểu việc vi phảm bảo mật trong trạng thái dùng thử nghiệm thiết bị mới và nó sẽ được cài đặt trên tất cả các thiết bị mới trước khi đưa đến tay người dùng thử nghiệm. Vì vậy, em lựa chọn đề tài: “Xây dựng ứng dụng bảo mật cho thiết bị di động Samsung khi sử dụng bản dùng thử” là đề tài cho đồ án tốt nghiệp.

## Mục tiêu nghiên cứu

- Phân tích, hiểu nghiệp vụ bài toán thực tế.

- Nghiên cứu về Phát triển phần mềm trên thiết bị di động, Android SDK, Samsung SDK, Google FCM.

- Phân tích và thiết kế hệ thống.

- Phát triển, kiểm thử và vận hành hệ thống đảm bảo hoạt động ổn định.

- Xây dựng báo cáo và tài liệu kèm theo.

## Đối tượng nghiên cứu

- Ngôn ngữ lập trình Java

- Công cụ lập trình Android Studio IDE

- Bộ công cụ phát triển phần mềm Android SDK, Samsung SDK ,Google FCM (Firebase Cloud Message).

## Phạm vi nghiên cứu

Sản phầm của đề tài này sẽ đáp ứng các yêu cầu sau:

- Hiển thị dấu mờ để ngăn chặn rò rỉ màn hình giao diện người dùng.

- Chặn truy cập mạng của các ứng dụng trong danh sách đen (tránh thu thập và chia sẻ thông tin cấu hình của thiết bị).

- Khóa thiết bị bằng Cover SDK/FCM khi điện thoại bị tháo rời/ mất.

## Phương pháp nghiên cứu

- Phương pháp nghiên cứu về mặt lý thuyết:

* Nghiên cứu và tiến hành thu thập các tài liệu có liên quan đến đề tài.
* Tổng hợp các tài liệu đã thu thập và tiến hành phân tích.
* Chọn lọc các tài liệu nghiên cứu để báo cáo đồ án tốt nghiệp.

- Phương pháp nghiên cứu trong thực nghiệm:

* Phân tích các sản phẩm có chức năng tương tự.
* Phân tích và tiến hành xây dựng sản phẩm theo phân tích và yêu cầu thực tế.
* Tìm hiểu ngôn ngữ lập trình Android, công cụ lập trình Android Studio IDE, mô hình MVVM, bộ công cụ phát triển phần mềm Android SDK, Samsung SDK ,Google FCM (Firebase Cloud Message).
* Áp dụng công nghệ vào xây dựng hệ thống.
* Tổng hợp tất các kiến thức đã học hoàn thành báo cáo.

# CHƯƠNG 1. TỔNG QUAN VỀ NỘI DUNG NGHIÊN CỨU

## Giới thiệu chung

Hiện nay, công nghệ di động đang phát triển một cách vượt bậc. Các thiết bị đi động đang trở thành một phần thiết yếu cho tất cả các ngành công nghiệp trên toàn thế giới. Với sự hiện diện của mình, các thiết bị di động đã thúc đẩy hoạt động kinh doanh, doanh thu, lượng người tiêu dùng tăng lên theo cấp số nhân. Các công ty hoạt động trong lĩnh vực công nghệ không ngừng cải tiến sản phẩm cũ và nghiên cứu cho ra mắt sản phẩm mới đón đầu xu hướng. Việc cạnh tranh trở nên gay gắt, các chiêu trò thủ đoạn vô cùng tinh vi. Một trong những hành vi cạnh tranh không lành mạnh, trái pháp luật nhưng lại rất phổ biến hiện nay là ăn cắp ý tưởng.

Sở hữu ý tưởng kinh doanh mới hay những cải tiến trong sản phẩm là một lợi thế vô cùng lớn cho doanh nghiệp, quyết định sự phát triển của một thương hiệu trên thị trường. Ý tưởng về sản phẩm mới bị rò rỉ là mối nguy đối với doanh nghiệp. Các doanh nghiệp khác có thể sao chép ý tưởng và tung sản phẩm ra thị trường sớm hơn, có lợi thế trong hoạt động quảng bá sản phẩm tốt hơn. Doanh nghiệp sẽ đánh mất vị trí người tiên phong và sự chủ động của mình trên thị trường. Bên cạnh đó, thị phần của sản phẩm mới có thể bị ảnh hưởng nghiêm trọng khi ra mắt, khiến doanh thu bán hàng bị sụt giảm. Từ thực tế đó, ứng dụng bảo mật thông tin thiết bị trong trạng thái dùng thử ra đời để giải quyết các vấn đề liên quan đến rò rỉ thông tin sản phẩm mới sắp được ra mắt.

## Cơ sở lý thuyết

### Tổng quan về lập trình di động

Hiện nay có 3 hướng chính xây dựng và phát triển 1 ứng dụng di động là: Native app, Web App và Hybrid app. Mỗi hướng đều có ưu và nhược điểm và kỹ năng riêng.

**1.2.1.1. Native App**

Native app là một ứng dụng di động hoặc máy tính được phát triển và tối ưu hóa cho một nền tảng cụ thể, chẳng hạn như iOS (hệ điều hành của iPhone và iPad) hoặc Android (hệ điều hành của điện thoại thông minh và máy tính bảng sử dụng hệ điều hành Android của Google). Native app được viết bằng ngôn ngữ lập trình đặc thù của nền tảng đó, chẳng hạn Objective-C hoặc Swift cho iOS, hoặc Java hoặc Kotlin cho Android.

Mỗi Native App chỉ chạy được trên một nền tảng và không thể mang sang các nền tảng khác.

* Ưu điểm:
  + Tận dụng được tính năng có sẵn trên thiết bị như: GPS, Camera, thiết bị thu âm...
  + Hiệu năng cao vì code native chạy trực tiếp trên máy.
  + Có thể chạy được ở chế độ online hoặc offline.
* Nhược điểm:
  + Không thể kết hợp nhiều nền tảng. Mỗi một ứng dụng chỉ chạy trên 1 nền tảng nhất định.
  + Mỗi hệ điều hành cân phải viết ứng dụng riêng khó đồng bộ giữa các ứng dụng.
  + Việc bảo trì hay nâng cấp sẽ làm mất nhiều thời gian. Do phải sửa chữa từng app trên từng hệ điều hành.
  + Xây dựng ứng dụng cần dùng các phần mềm riêng biệt theo hệ điều hành (dùng Xcode trên Mac để phát triển ứng dụng IOS, Android Studio để phát triển ứng dụng Android).

**1.2.1.2. Hybrid App**

Hybrid app là một loại ứng dụng di động hoặc máy tính được phát triển bằng cách sử dụng các công nghệ đa nền tảng, cho phép chia sẻ mã nguồn giữa các nền tảng khác nhau, chẳng hạn như iOS, Android, và các nền tảng web. Hybrid app thường được viết bằng các ngôn ngữ web phổ biến như HTML, CSS, và JavaScript.

* Ưu điểm:
  + Chỉ cần có kiến thức về HTML, CSS, JavaScript…
  + Viết một lần dùng được nhiều nơi.
  + Tận dụng được các chức năng của hệ thống.
  + Có thể chạy được ở chế độ offline
* Nhược điểm:
  + Hiệu năng chậm .
  + Không ổn định do khó debug. Framework sẽ dịch code thành code native sửa lỗi khá khó khăn không biết được dịch như thế nào.

**1.2.1.3. Web App**

Web app được phát triển khi đã có sẵn một website đang hoạt động. Ta tạo thêm một trang web riêng cho các thiết bị di động sử dụng HTML, CSS và một số thư viện khác hỗ trợ.

Web app được thiết kế chạy trên nền tảng web hoặc các trình duyệt của thiết bị di động cho phép người dùng thao tác như thao tác trang web giống nhu thao tác ứng dụng.

* Ưu điểm:
  + Có thể chạy trên tất cả trình duyệt của mobile hỗ trợ phiên bản HTML và javascript.
  + Không cần cài đặt trên máy miễn là máy có trình duyệt web.
  + Một phiên bản duy nhất cho tất cả, nên giảm chi phí và thời gian cho phát triển, bảo trì, cũng như nâng cấp sau này.
  + Phiên bản được cập nhật liên tục không cần phải cập nhật trên chợ.
* Nhược điểm:
  + Hiệu năng không được tốt như native app và luôn phải chạy online.
  + Không thể dùng được các tính năng tích hợp của di động: Notification, chụp hình, nghiêng máy, định vị GPS, các sensor…
  + Với một số máy đời cũ, Web app sẽ bị vỡ giao diện, hiển thị sai, hoặc javascript không chạy.

### Giới thiệu công cụ lập trình Android Studio

Android Studio là Môi trường phát triển phần mềm tích hợp (IDE) chính thức để phát triển ứng dụng Android. Nó được ra mắt vào ngày 16 tháng 5 năm 2013 tại hội nghị Google I/O. Android Studio được phát hành miễn phí theo giấy phép Apache Licence 2.0. Android Studio ở giai đoạn truy cập xem trước sớm bắt đầu từ phiên bản 0.1 vào tháng 5 năm 2013, sau đó bước vào giai đoạn beta từ phiên bản 0.8 được phát hành vào tháng 6 năm 2014. Phiên bản ổn định đầu tiên được ra mắt vào tháng 12 năm 2014, bắt đầu từ phiên bản 1.0.

Dựa trên phần mềm IntelliJ IDEA của JetBrains, Android Studio được thiết kế đặc biệt để phát triển ứng dụng Android. Nó hỗ trợ các hệ điều hành Windows, Mac OS X và Linux, và là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools (ADT) dựa trên Eclipse.

### Ngôn ngữ lập trình Java

Java là một ngôn ngữ lập trình hướng đối tượng, dựa trên lớp được thiết kế để có càng ít phụ thuộc thực thi càng tốt. Nó là ngôn ngữ lập trình có mục đích chung cho phép các nhà phát triển ứng dụng viết một lần, chạy ở mọi nơi (WORA), nghĩa là mã Java đã biên dịch có thể chạy trên tất cả các nền tảng hỗ trợ Java mà không cần biên dịch lại. Các ứng dụng Java thường được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java (JVM) nào bất kể kiến trúc máy tính bên dưới. Cú pháp của Java tương tự như C và C++, nhưng có ít cơ sở cấp thấp hơn các ngôn ngữ trên. Java runtime cung cấp các khả năng động (chẳng hạn như phản ánh và sửa đổi mã thời gian chạy) thường không có sẵn trong các ngôn ngữ biên dịch truyền thống.

Java ban đầu được James Gosling tại Sun Microsystems (sau đó đã được Oracle mua lại) phát triển và được phát hành vào năm 1995 như một thành phần cốt lõi của nền tảng Java của Sun Microsystems. Các trình biên dịch Java, máy ảo và thư viện lớp thực thi gốc và tham chiếu ban đầu được Sun phát hành theo giấy phép độc quyền. Kể từ tháng 5 năm 2007, tuân theo các thông số kỹ thuật của Quy trình Cộng đồng Java, Sun đã cấp phép hầu hết các công nghệ Java của mình theo Giấy phép Công cộng GNU. Oracle cung cấp Máy ảo Java HotSpot của riêng mình, tuy nhiên việc triển khai tham chiếu chính thức là OpenJDK JVM, là phần mềm mã nguồn mở miễn phí và được hầu hết các nhà phát triển sử dụng và là JVM mặc định cho hầu hết các bản phân phối Linux.

### Application Framework trong lập trình Android

Application Framework trong Android là một bộ các thư viện và lớp cơ bản được cung cấp bởi hệ điều hành Android để phát triển ứng dụng di động. Nó cung cấp các công cụ và giao diện lập trình ứng dụng (API) cho các nhà phát triển để xây dựng ứng dụng Android.

Các thành phần chính của Application framework trong Android bao gồm:

* **Activities**: Là các thành phần giao diện người dùng của ứng dụng Android. Nó cho phép người dùng tương tác với ứng dụng thông qua các màn hình, các hoạt động (Activity) của ứng dụng.
* **Fragments**: Là một phần của một Activity, được sử dụng để tạo giao diện người dùng đa màn hình. Fragments cho phép phát triển ứng dụng Android linh hoạt hơn, với khả năng chia sẻ logic và giao diện người dùng giữa các màn hình khác nhau.
* **Views**: Là các thành phần giao diện người dùng như TextView, Button, EditText, ListView, RecyclerView, và nhiều hơn nữa. Views cho phép hiển thị và tương tác với dữ liệu trong giao diện người dùng của ứng dụng Android.
* **Intents**: Là cơ chế để giao tiếp giữa các thành phần khác nhau của ứng dụng Android hoặc giữa các ứng dụng khác nhau. Intents cho phép chuyển dữ liệu và thực hiện các hoạt động như mở một Activity mới hoặc gửi một thông báo.
* **Content Providers**: Là các thành phần dùng để quản lý và cung cấp dữ liệu cho ứng dụng Android, chẳng hạn như lưu trữ dữ liệu trong cơ sở dữ liệu SQLite hoặc chia sẻ dữ liệu giữa các ứng dụng khác nhau.
* **Services**: Là các thành phần chạy nền của ứng dụng Android, cho phép thực hiện các hoạt động không liên quan đến giao diện người dùng, chẳng hạn như phát nhạc, tải dữ liệu từ mạng, hoặc xử lý công việc dài hạn.
* **Notification Manager**: Là các thành phần chạy nền của ứng dụng Android, cho phép thực hiện các hoạt động không liên quan đến giao diện người dùng, chẳng hạn như phát nhạc, tải dữ liệu từ mạng, hoặc xử lý công việc dài hạn.

Ngoài ra, Application framework trong Android còn cung cấp các công cụ và giao diện lập trình khác như Resource Manager, Package Manager, Activity Manager, và System Services, giúpngười phát triển xây dựng ứng dụng Android với nhiều tính năng phong phú và đa dạng, bao gồm đồ họa, âm thanh, định vị, mạng, quản lý tài nguyên, quản lý giao diện người dùng, và quản lý vòng đời ứng dụng.

Một số công nghệ và thư viện khác trong Application framework của Android bao gồm:

* **Android UI Toolkit**: Cung cấp các công cụ và thư viện để xây dựng giao diện người dùng trong Android, bao gồm XML layouts, các lớp View, Drawable, Animator, và nhiều tính năng đồ họa và tương tác khác.
* **Android Resource Manager**: Quản lý tài nguyên trong ứng dụng Android như hình ảnh, âm thanh, dữ liệu chuỗi, màu sắc, kiểu chữ, và các tài nguyên khác.
* **Android Location API**: Cung cấp khả năng định vị địa lý cho ứng dụng Android, cho phép định vị vị trí của thiết bị, lấy dữ liệu về vị trí, hoặc theo dõi vị trí của thiết bị.
* **Android Networking API**: Cung cấp các công cụ và thư viện để thực hiện các kết nối mạng trong ứng dụng Android, bao gồm HTTP, WebSocket, gửi và nhận dữ liệu qua mạng.
* **Android Multimedia API**: Cung cấp khả năng xử lý đa phương tiện như hình ảnh, âm thanh, và video trong ứng dụng Android, bao gồm phát lại, ghi âm, xử lý hình ảnh, và các tính năng đa phương tiện khác.
* **Android Database API:** Cung cấp các công cụ và thư viện để quản lý cơ sở dữ liệu trong ứng dụng Android, bao gồm SQLite và Content Providers để lưu trữ và truy vấn dữ liệu.
* **Android Notification API**: Cung cấp khả năng hiển thị thông báo trong Android, bao gồm tạo, quản lý, và tương tác với các thông báo đẩy hoặc thông báo trong hệ thống.

### Google Firebase Cloud Message