|  |  |
| --- | --- |
|  | * + 1. **ĐẠI HỌC KHOA HỌC TỰ NHIÊN**     2. **KHOA CÔNG NGHỆ THÔNG TIN** |
| * + 1. **MÔN QUẢN LÝ QUY TRÌNH PHẦN MỀM** |

* + 1. **Test plan**
    2. **ỨNG DỤNG TỪ ĐIỂN CHO HỆ DIỀU HÀNH ADROID**

Phiên bản 1.0

TP HỒ CHÍ MINH 2013

**Bảng nghi nhận thay đổi tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả** | **Tác giả** |
| 05/10/2013 | 1.0 | Soạn thảo tài liệu | Hà Thanh Toàn, Phan Phước Thông và các thành viên nhóm lớn 2 |
|  |  |  |  |
|  |  |  |  |

**Mục lục**

[1. Giới thiệu 4](#_Toc369592566)

[1.1 Mục đích tài liệu 4](#_Toc369592567)

[1.2 Tổng quan dự án 4](#_Toc369592568)

[1.3 Đối tượng độc giả 5](#_Toc369592569)

[1.4 Phạm vi test 5](#_Toc369592570)

[1.5 Tài liệu dự án 6](#_Toc369592571)

[2. Yêu cầu kiểm thử 7](#_Toc369592572)

[2.1 Yêu cầu chức năng 7](#_Toc369592573)

[2.2 Yêu cầu hệ thống **Error! Bookmark not defined.**](#_Toc369592574)

[2.3 Yêu cầu phi chức năng 7](#_Toc369592575)

[3. Chiến lược kiểm thử 8](#_Toc369592576)

[3.1 Các loại kiểm thử 10](#_Toc369592577)

[3.2 Công cụ kiểm thử 13](#_Toc369592578)

[4. Nguồn lực 14](#_Toc369592579)

[4.1 Nhân sự 14](#_Toc369592580)

[4.2 Hệ thống 14](#_Toc369592581)

[5. Thời gian kiểm thử 15](#_Toc369592582)

[6. Thông tin và tài liệu kết quả 15](#_Toc369592583)

[6.1 Ghi chú kiểm thử (Test log) 15](#_Toc369592584)

[6.2 Tổng hợp báo cáo lỗi 15](#_Toc369592585)

1. Giới thiệu

## Mục đích tài liệu

* + 1. Mục đích của tài liệu này là:
* Lên kế hoạch kiểm thử
* Mô tả chi tiết các phương pháp kiểm thử sẽ áp dụng vào dự án
* Phạm vi kiểm thử
* Yêu cầu kiểm thử
* Nguồn lực kiểm thử
* Thời gian sẽ kiểm thử.

## Tổng quan dự án

* + 1. Vấn đề tra từ điển giấy gặp phải nhiều khó khăn do người dùng phải mang theo quyển từ điển bên người mới có thể tra từ, như vậy thì khá bất tiện do quyển từ điển thì thường rất dày và nặng. Còn nếu như sử dụng quyển từ điển nhỏ thì sẽ không đủ vốn từ để tra. Việc xây dựng phần mềm từ diển trên Android giúp cho việc tra cứu từ vựng trở nên đơn giản, không cần phải mang theo 1 quyển từ điển bên người, giúp người dùng tra cứu một cách nhanh chóng và chính xác.
    2. Phần mềm bao gồm những chức năng sau:
* Tra từ: Người dùng nhập vào từ cần tra và bấm tìm kiếm, phần mềm sẽ tra cứu và đưa ra kết quả cho người dùng
* Tra chéo: Người dùng trong quá trình xem nghĩa của từ mà gặp phải một từ nào khó hiểu có thể bấm vào từ đó để tra tiếp mà không cần phải nhập lại.
* Xem Lịch sử: Cho phép người dùng xem lại lịch sử những từ đã tra, chọn vào một từ xuất hiện trong lịch sử để tra tiếp.
* Xóa lịch sử: Người dùng còn có thể xóa lịch sử sau một thời gian sử dụng.
* Xem từ yêu thích: Cho phép người dùng xem lại những từ đã đánh dấu yêu thích, và chọn vào từ đó để tra cứu
* Thêm từ yêu thích: Người dùng có thể chọn một từ nào đó để thêm vào danh sách yêu thích.
* Xóa danh sách từ yếu thích: Người dùng còn có thể xóa danh sách sau một thời gian sử dụng.
* Tra từ bằng giọng nói: Người dùng đọc lên từ muốn tra cứu và phần mềm sẽ phân tích và đưa ra gợi ý để người dùng chọn.

## Đối tượng độc giả

* + 1. Tài liệu này dành cho cả khách hàng và đội ngủ phát triển.

## Phạm vi test

Các giai đoạn test: Unit test.

Các kiểu test:

* Functional Testing
* Performance Testing

Các tính năng và chức năng sẽ test:

* Tra từ
* Tra chéo
* Xem Lịch sử
* Xóa lịch sử
* Xem từ yêu thích
* Xóa từ yêu thích
* Thêm từ yêu thích
* Tra từ bằng giọng nói

Các tính năng không test:

* Feed back
* About
* Rate

## Ràng buộc

Môi trường test cần phải tương thích tốt với hệ điều hành Android phiên bản 4.1.2

Công cụ để test cần có: Máy thật để test sau khi release sản phẩm

Ràng buộc về nguồn lực: trong quá trình test thì phải luôn có ít nhất là 1 đến 2 người luôn bám sát tiết độ, vừa phát triển vừa test để giảm bớt rũi ro trong quá trình phát triển

Quá trình test không thể để đến cuối dự án sau khi đã phát triển xong mới test mà là test song song với developers

## Liệt kê mạo hiểm

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mạo hiểm | Phương pháp khắc phục, phòng ngừa | Mức độ ảnh hưởng |
| 1 | Không có thiết bị thật | Mượn bạn bè | Thấp |
| 2 | Ứng dụng không tương thích với nhiều phiên bản Android |  | Cao |
| 3 |  |  |  |

## Tài liệu dự án

Dannh sách các tài liệu đã có của dự án

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tài liệu | Đã có/được tạo | Đã nhận | Người tạo/Nguồn | Ghi chú |
| Tài liệu mô tả yêu cầu | x | x |  |  |
| Tài liệu mô tả chức năng | x | x |  |  |
| Tài liệu kế hoạch dự án |  |  |  |  |
| Tài liệu phân tích thiết kế |  |  |  |  |
| Tài liệu hướng dẫn sử dụng |  |  |  |  |

1. Yêu cầu kiểm thử

## Yêu cầu chức năng

* Tra từ bằng giọng nói
* Tra từ bằng cách nhập
* Phát âm từ
* Tra từ chéo
* Thêm từ vào danh sách yêu thích
* Xóa từ trong danh sách yêu thích
* Xem danh sách từ yêu thích
* Xem lịch sử
* Xóa lịch sử
* Từ trong ngày

## Yêu cầu phi chức năng

* Ứng dụng tra từ nhanh (sau khi nhập tối đa trong 3s sẽ có kết quả trả về)
* Giao diện đơn giản dể sử dụng, thân thiện với người dùng
* Giọng phát âm chuẩn
* Gợi ý: Hiển thị những từ gần giống với từ tra bằng giọng nói

1. Chiến lược kiểm thử

## Kỹ thuật:

### Functional Test

Đối với chu trình sự kiện của mỗi UC, sẽ xác định một tập các giao dịch đại diện cho mỗi hành động của tác nhân khi thực hiện UC.

Tối thiểu phải có 2 TC cho mỗi giao dịch, một TC để phản ánh điều kiện tích cực và một phản ánh điều kiện tiêu cực (không được chấp nhận)

Trong giai đoạn đầu tiện, các UC sẽ được test, theo hình thức sau:

Việc kiểm tra và đánh giá việc thực hiện mỗi TC sẽ được thực hiện theo phương pháp sau:

Thực hiện Test script (Mỗi test script có được thực hiện thành công như mong muốn không?)

Phương pháp kiểm tra Object Data (tiến hành trong các test script) sẽ được dùng để kiểm tra sự hiển thị của các màn hình chính và dữ liệu được xác định được nắm bắt/hiển thị bởi mục tiêu test trong khi thực hiện test.

### Performance Test:

Với mỗi UC, xác định một tập các giao dịch, như định nghĩa trong tài liệu phân tích workload, sẽ được tiến hành và thực hiện bằng Rational Suite PerformanceStudio và Rational Robot (GUI scripts)

Test script dùng để thực hiện mỗi giao dịch sẽ bao gồm bộ đếm thời gian tương tự để đo thời gian phản hồi, ví dụ tổng thời gian giao dịch (như định nghĩa trong tài liệu phân tích workload), và các hoạt động giao dịch chính hoặc thời gian xử lý.

Test script sẽ thực hiện các workload trong 1 giờ (trừ phi được ghi chú khác trong tài liệu phân tích workload).

Kiểm tra và đánh giá việc thực hiện mỗi thực hiện test (của một workload) bao gồm:

Thực hiện test được theo dõi bằng biểu đồ trạng thái (để xác định rằng việc test và workload được thực hiện như mong muốn)

Thực hiện test script (mỗi test script có được thực hiện thành công như mong đợi không?)

Ghi nhận và đánh giá thời gian phản hồi đã định nghĩa bằng các báo cáo sau:

* Performance Percentile
* Response Time

## Điều kiện hoàn thành

Tất cả các TC có trong kế hoạch đều đã được thực hiện

Tất cả các lỗi được xác định phải được ghi nhận vào một giải pháp đã thỏa thuận (All identified defects have been addressed to an agreed upon resolution)

Tất cả các TC có trong kế hoạch đã được thực hiện lại và toàn bộ các lỗi mở đã được ghi nhận như đã thỏa thuận và không có lỗi mới nào được phát hiện

Hoặc

Toàn bộ các TC đặt mức ưu tiên cao đều đã được thực hiện

Toàn bộ các lỗi tìm thấy đều được ghi nhận vào một giải pháp đã thỏa thuận

Toàn bộ các lỗi có trọng số 1 và 2 đều được giải quyết

Tất cả các TC có mức ưu tiên cao đều đã được thực hiện lại và toàn bộ các lỗi mở đã được ghi nhận như đã thỏa thuận và không có lỗi mới nào được phát hiện

## Các vấn đề đặc biệt

- Cơ sở dữ liệu test yêu cầu người thiết kế hoặc quản trị CSDL hỗ trợ để tạo mới, cập nhật và làm tươi dữ liệu test

- Việc test hiệu suất hệ thống sử dụng máy chủ trong mạng hiện tại (có hỗ trợ cả các giao dịch khác không thuộc việc test). Việc test sẽ phải được lập lịch vào những giờ không còn các giao dịch khác trên mạng.

- Mục tiêu test phải đồng nhất với hệ thống hợp lệ (hoặc giả lập đồng bộ) để việc test chức năng có thể được tiến hành và thực hiện

- Việc test có thể bị dừng khi số lỗi vượt quá norm, ...

- Cán bộ test có thể dừng test khi lập trình viên không thực hiện unit test, ...

## Các loại kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| Giai đoạn/ Kiểu test | Unit test | Inergration test | System Test |
| Giai đoạn 1  Function, User Interface | X | X | X |
| Giai đoạn 2  Performance test | X | X |  |
| Giai đoạn 3  Intergrity, Structure |  |  | X |

## 3.1.1 Functional Testing

Mục đích của test chức năng là tập trung vào các yêu cầu test có thể được lưu vết trực tiếp trong các UC hoặc các chức năng và qui tắc nghiệp vụ. Mục tiêu của kiểu test này là kiểm tra tính đúng đắn của các dữ liệu, qui trình và báo cáo cũng như việc thực hiện đúng những qui tắc nghiệp vụ. Kiểu test này dựa vào kỹ thuật black box, tức là kiểm tra ứng dụng và các xử lý nội tại bằng cách tương tác với ứng dụng thông qua giao diện người sử dụng và phân tích các kết quả hoặc đầu ra. Bảng sau liệt kê một số gợi ý đối với mỗi ứng dụng:

|  |  |
| --- | --- |
| Mục đích test: | Đảm bảo mục tiêu test đúng đắn của chức năng, bao gồm định hướng, dữ liệu đầu vào, xử lý và dữ liệu nhận được |
| Cách thực hiện: | Thực hiện mỗi UC, chu trình UC hoặc chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để kiểm tra:  -    Kết quả mong đợi với dữ liệu hợp lệ.  -    Lỗi thích hợp hoặc thông báo hiển thị khi dữ liệu không hợp lệ.  -    Mỗi qui tắc nghiệp vụ đều được áp dụng đúng |
| Điều kiện hoàn thành: | -   Toàn bộ kế hoạch test đã được thực hiện.  -   Toàn bộ các lỗi phát hiện ra đã được ghi nhận. |
| Các vấn đề đặc biệt: | Xác định hoặc mô tả các vấn đề (nội bộ hoặc bên ngoài) ảnh hưởng đến việc test chức năng |

## 3.1.2 Test giao diện người sử dụng (User Interface Testing)

Test giao diện người dùng (UI) kiểm tra các tương tác của người dùng với phần mềm. Mục tiêu của test UI là để đảm bảo rằng giao diện người dùng cung cấp cho người sử dụng cách truy cập và sử dụng thích hợp thông qua các chức năng trong mục tiêu test. Ngoài ra, test UI còn để đảm bảo rằng các đối tượng trong phạm vi chức năng UI giống như mong đợi và phù hợp với tổ chức hoặc chuẩn ngành

|  |  |
| --- | --- |
| Mục đích test: | Kiểm tra:     Việc sử dụng thông qua mục tiêu test phản ánh đúng các chức năng và yêu cầu nghiệp vụ, bao gồm màn hình đến màn hình, trường đến trường và sử dụng các phương pháp truy cập (phím tabs, di chuột, tổ hợp phím)      Các đối tượng và thuộc tính màn hình như menus, size, position, state, và tập tring vào việc tương thích với chuẩn |
| Cách thực hiện: | Tạo ra và chỉnh sửa test cho mỗi màn hình để kiểm tra việc sử dụng đúng cách và tình trạng các đối tượng cho mỗi màn hình và đối tượng của ứng dụng |
| Điều kiện hoàn thành: | Mỗi màn hình được kiểm tra thành công đúng với phiên bản kiểm tra hoặc phạm vi chấp nhận được |
| Các vấn đề đặc biệt: | Không phải toàn bộ các thuộc tính của các đối tượng đều truy cập được |

### 3.1.3 Test hiệu suất (Performance testing)

Performance profiling là một dạng test hiệu suất trong đó thời gian phản hồi, tỷ lệ giao dịch và các yêu cầu phụ thuộc thời gian khác được đo đạc và đánh giá. Mục đích của Performance Profiling là kiểm tra các yêu cầu về hiệu suất có đạt được hay không. Performance profiling là tiến hành và thực hiện để mô tả sơ lược và điều chỉnh các hành vi hiệu suất của mục tiêu test như một hàm của các điều kiện ví dụ workload hoặc cấu hình phần cứng.

Chú ý: Các giao dịch dưới đây tham chiếu đến “các giao dịch nghiệp vụ logic”. Các giao dịch này được định nghĩa như xác định các UC mà tác nhân của hệ thống hy vọng được thực hiện bằng cách sử dụng mục tiêu test, như thêm mới hoặc sửa một hợp đồng

|  |  |
| --- | --- |
| Mục đích test: | Kiểm tra các biểu hiện về hiệu suất cho các giao dịch hoặc chức năng nghiệp vụ đã thiết kế theo những điều kiện sau:      workload bình thường đã biết trước (normal anticipated workload)      workload xấu đã biết trước (anticipated worst case workload) |
| Cách thực hiện: |     Sử dụng các thủ tục test cho test chức năng và chu trình nghiệp vụ      Chỉnh sửa file dữ liệu để tăng số lượng các giao dịch hoặc scripts để tăng số tương tác xảy ra trong mỗi giao dịch      Scripts phải được chạy trên một máy (trường hợp tốt nhất để đánh giá người dùng đơn lẻ, giao dịch đơn lẻ) và phải lặp lại trên nhiều máy trạm (ảo hoặc thực, xem các vấn đề đặc biệt dưới đây) |
| Điều kiện hoàn thành: |     Giao dịch đơn lẻ hoặc người dùng đơn lẻ: Thực hiện thành công test script không có lỗi và trong phạm vi mong đợi hoặc thời gian phản hồi cho mỗi giao dịch      Nhiều giao dịch hoặc nhiều người dùng: Thực hiện thành công test script không có lỗi và trong thời gian chấp nhận được |
| Các vấn đề đặc biệt: | Việc test hiệu suất toàn diện bao gồm phải có một workload nền trên máy chủ.  Có một số phương pháp để thực hiện, bao gồm:     “Drive transactions” trực tiếp đến máy chủ, thường trong các form gọi SQL.      Tạo các người dùng ảo để giả lập nhiều máy trạm, thường là vài trăm. Sử dụng công cụ Remote Terminal Emulation để thực hiện việc load này, kỹ thuật này còn được dùng để load giao dịch trên mạng      Sử dụng nhiều người dùng, mỗi người chạy một test script để load lên hệ thống  Test hiệu suất phải được thực hiện trên máy chuyên dụng hoặc thời gian chuyên dùng. Điều đó cho phép việc tính toán được đầy đủ và chính xác.  Cơ sở dữ liệu sử dụng để test hiệu suất phải có kích thước thực tế hoặc đo bằng nhau |

* 1. **Môi trường test**

## Môi trường sẽ xây dựng để thực hiện test qua các giai đoạn Unit test, Intergration test, System test, Acceptance test: Eclipse

## Với mỗi giai đoạn, cần xác định các yếu tố để xây dựng môi trường test như thế nào, sử dụng như môi trường mà chương trình sẽ chạy thật hay tạo môi trường giả lập gần giống với môi trường chạy thật của chương trình. Các yếu tố về môi trường như:

## - Khi test chạy chương trình bằng bản dịch chạy trên bản dịch.

## - Các database sẽ sử dụng chung với database phát triển.

## - Có thể sử dụng mang internet để máy ảo kết nối với internet.

## - Sau khi gần hoàn thành sản phẩm sẽ test thử trên máy thật.

## Công cụ kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
|  | Công cụ | Nhà sản xuất/Tự phát triển | Phiên bản |
| Quản lý hoạt động kiểm thử | Eclipse | Nguồn mở | ADT |
| Kiểm thử tự đông | Robotium | Nguồn mở |  |
| Kiểm soát lỗi |  |  |  |
| Kiểm thử chức năng | JUnit |  |  |
| Kiểm thử hoạt động |  |  |  |
| Quản lý dự án |  |  |  |
| Các công cụ quản trị (nếu có) | Github |  |  |
| Cơ sở dữ liệu | Sqlite |  |  |

1. Nguồn lực

## Nhân sự

|  |  |  |
| --- | --- | --- |
| Nhân sự | Số lượng tối thiểu có | Ghi chú |
| Quản lý kiểm thử (Test manager) Test Project Manager | 1 | Quản lý hoạt động kiểm thử  Nhiệm vụ:   * Hướng dẫn kỉ thuật * Sử dụng và quản lý nguồn lực * Báo cáo quản lý |
| Nhân viên kiểm thử (Test ter) | 2 | Thực hiện việc kiểm thử  Nhiệm vụ:   * Tiến hành kiểm thử * Viết các ghi chú kết quả kiểm thử (test log) * Viết tài liệu báo cáo kiểm thử |

## Hệ thống

|  |  |
| --- | --- |
| * + 1. Tài nguyên | * + 1. Tên/Kiểu/Loại |
| Máy thật để test ứng dụng | * + 1. Thiết bị |

1. Thời gian kiểm thử

|  |  |  |
| --- | --- | --- |
| Nhiệm vụ | Ngày bắt đầu | Ngày kết thúc |
| Kế hoạch | 14/10/2013 | 16/10/2013 |
| Thực hiện kiểm thử | 17/10/2013 | 6/11/2013 |
| Đánh giá kết quả kiểm thử | 7/11/2013 | 14/11/2013 |

1. Thông tin và tài liệu kết quả

## Ghi chú kiểm thử (Test log)

## Tổng hợp báo cáo lỗi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Sản phẩm | Ngày bàn giao | Người bàn giao | Người nhận bàn giao |
|  | <Test cases> |  |  |  |
|  | <Test procedures> |  |  |  |
|  | <Defect log> |  |  |  |
|  | <Defect reports> |  |  |  |