**BỘ** **GIÁO** **DỤC** **VÀ** **ĐÀO** **TẠO**



**TRƯỜNG** **ĐẠI** **HỌC** **NÔNG** **LÂM** **TP** **HCM**

**KHOA** **CÔNG** **NGHỆ** **THÔNG** **TIN**

**LUẬN** **VĂN** **TỐT** **NGHIỆP**

**NGHIÊN CỨU GIẢI PHÁP, CÔNG NGHỆ REACT NATIVE, NODEJS ĐỂ XÂY DỰNG PHẦN MỀM THANH TOÁN VÍ ĐIỆN TỬ.**

**Ngành** **:** **Niên** **khóa** **:** **Lớp** **:** **Sinh** **viên** **thực** **hiện** **:**

**Công** **nghệ** **thông** **tin** **2016** **-** **2020**

**DH16DT**

**Nguyễn Hoàng Việt**

**Nguyễn Lê Khang**

Thành phố Hồ Chí Minh, tháng 5 năm 2020

**BỘ** **GIÁO** **DỤC** **VÀ** **ĐÀO** **TẠO**



**TRƯỜNG** **ĐẠI** **HỌC** **NÔNG** **LÂM** **TP** **HCM**

**KHOA** **CÔNG** **NGHỆ** **THÔNG** **TIN**

**LUẬN** **VĂN** **TỐT** **NGHIỆP**

**NGHIÊN CỨU GIẢI PHÁP, CÔNG NGHỆ REACT NATIVE, NODEJS ĐỂ XÂY DỰNG PHẦN MỀM THANH TOÁN VÍ ĐIỆN TỬ.**

**Giảng** **viên** **hướng** **dẫn:** TS. Nguyễn Văn Dũ

**Sinh** **viên** **thực** **hiện:** Nguyễn Hoàng Việt

Nguyễn Lê Khang

Thành phố Hồ Chí Minh, tháng 5 năm 2020

ĐỂ XÂY DỰNG PHẦN MỀM THANH TOÁN VÍ ĐIỆN TỬ.

NGHIÊN CỨU GIẢI PHÁP, CÔNG NGHỆ REACT NATIVE , NODEJS

2020

NĂM

**CÔNG** **TRÌNH** **HOÀN** **TẤT** **TẠI**

**TRƯỜNG** **ĐẠI** **HỌC** **NÔNG** **LÂM** **TP** **HCM**

Cán bộ hướng dẫn: TS. Nguyễn Văn Dũ

Cán bộ phản biện: ThS. Lê Phi Hùng

Luận văn cử nhân được bảo vệ tại HỘI ĐỒNG CHẤM LUẬN VĂN CỬ NHÂN TRƯỜNG ĐẠI HỌC NÔNG LÂM TP. HCM ngày 25 tháng 05 năm 2020

**BỘ** **GIÁO** **DỤC** **VÀ** **ĐÀO** **TẠO** **CỘNG** **HOÀ** **XÃ** **HỘI** **CHỦ** **NGHĨA** **VIỆT** **NAM**

**TRƯỜNG** **ĐH** **NÔNG** **LÂM** **TP.** **HCM** **Độc** **lập - Tự** **do - Hạnh** **phúc**

**NHIỆM** **VỤ** **LUẬN** **VĂN** **CỬ** **NHÂN**

Họ tên sinh viên: **Nguyễn Hoàng Việt**

Ngày tháng năm sinh: 18/08/1998

Chuyên ngành: Công nghệ Thông tin

Họ tên sinh viên: **Nguyễn Lê Khang**

Ngày tháng năm sinh: 08/11/1998

Chuyên ngành: Công nghệ Thông tin

Phái: **Nam**

Nơi sinh: Bến Tre

Ngành: Công nghệ Thông tin

Phái: **Nam**

Nơi sinh: Bình Thuận

Ngành: Công nghệ Thông tin

I. TÊN ĐỀ TÀI: **Nghiên cứu giải pháp, công nghệ React native, NodeJS để xây dựng phần mềm thanh toán Ví điện tử.**

II. NHIỆM VỤ VÀ NỘI DUNG

Nghiên cứu giải pháp, công nghệ React native, NodeJS để xây dựng phần mềm thanh toán Ví điện tử.

III. NGÀY GIAO NHIỆM VỤ: **31/07/2019**

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: **25/05/2020**

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: **TS. Nguyễn Văn Dũ**

Ngày / / Ngày / / Ngày / /

CÁN BỘ HƯỚNG DẪN

(Ký và ghi rõ họ tên)

CÁN BỘ PHẢN BIỆN

(Ký và ghi rõ họ tên)

KHOA CNTT

(Ký và ghi rõ họ tên

LỜI CẢM ƠN

Để hoàn thành chuyên đề báo cáo thực tập này trước tiên em xin gửi đến các quý thầy, cô giáo trường Đại học Nông Lâm Thành phố Hồ Chí Minh lời cảm ơn chân thành và sâu sắc nhất.

Đặc biệt, em xin gởi đến TS. Nguyễn Văn Dũ – người đã tận tình hướng dẫn, giúp đỡ em hoàn thành chuyên đề báo cáo thực tập này lời cảm ơn sâu sắc nhất.

Em xin chân thành cảm ơn sự giúp đỡ của ban lãnh đạo, cô chú, anh chị và toàn thể nhân viên công ty cổ phần JITS đã tạo điều kiện cho em tìm hiểu cung cấp các số liệu và tài liệu cho em trong quá trình thực tập và làm chuyên đề.

Vì kiến thức bản thân còn hạn chế, trong quá trình thực tập, hoàn thiện chuyên đề này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ thầy, cô cũng như quý công ty.

Em xin chân thành cảm ơn.

Thành phố Hồ Chí Minh, tháng 5 năm 2020

Nguyễn Hoàng Việt

Nguyễn Lê Khang

TÓM TẮT

Đề tài nghiên cứu **“Nghiên cứu giải pháp, công nghệ React Native, NodeJS để xây dựng phần mềm thanh toán Ví điện tử”** đã được tiến hành từ tháng 7 năm 2019 đến tháng 5 năm 2020. Luận văn này đã nghiên cứu về giải pháp, công nghệ React native, NodeJS để xây dựng phần mềm thanh toán Ví điện tử. Kiến thức sau khi thực hiện luận văn này sẽ giúp sinh viên bắt đầu trên con đường chuyên sâu.

Kết quả đạt được sau nghiên cứu nàylà bộ tài liệu nghiên cứu công nghệ React Native và NodeJS. Phần mềm ví điện tử với các chức năng liên quan đến tài khoản, nhận/chuyển tiền, thanh toán trực tuyến, ... trên nền tảng di động.

Từ khóa: React Native, NodeJS, Wallet, …

**ABSTRACT**

The research topic "Research solutions, React native and NodeJS technologies to develop e-wallet payment software" has been conducted from July 2019 to May 2020. This thesis has studied the solutions, React native and NodeJS technologies, to build e-wallet payment software. Knowledge after implementing this thesis will help students get started on an in-depth path.

Moreover, this thesis can be considered as a useful reference document for students in studying and applying React Native and NodeJS technologies for developing mobile applications.

Key words: React Native, NodeJS, Wallet, …

DANH SÁCH CÁC HÌNH

[Hình 1: Kiến trúc client – server. 4](#_Toc41116508)

[Hình 2: Kiến trúc ứng dụng NodeJS. 5](#_Toc41116509)

[Hình 3: Kiến trúc ứng dụng React Native. 6](#_Toc41116510)

[Hình 4: Kiến trúc phầm mềm React Native 7](#_Toc41116511)

[Hình 5: Kiến trúc phầm mềm NodeJS 8](#_Toc41116512)

[Hình 6: Bắt đầu với React Native 12](#_Toc41116513)

[Hình 7: Demo Class Component 13](#_Toc41116514)

[Hình 8: Demo Function Component 14](#_Toc41116515)

[Hình 9: Một ví dụ về việc sử dụng nhiều View trên ứng dụng Android và iOS 15](#_Toc41116516)

[Hình 10: Thành Phần Cơ Bản React Native 16](#_Toc41116517)

[Hình 11: Sơ đồ các thành phần React 17](#_Toc41116518)

[Hình 12: Sơ đồ hoạt động JWT 20](#_Toc41116519)

[Hình 13: Cấu trúc message ISO 8583 21](#_Toc41116520)

[Hình 14: Mô tả Message Type Indicator (1) 21](#_Toc41116521)

[Hình 15: Mô tả Message Type Indicator (2) 22](#_Toc41116522)

[Hình 16: Mô tả Message Type Indicator (3) 22](#_Toc41116523)

[Hình 17: Mô tả Message Type Indicator (4) 23](#_Toc41116524)

[Hình 18: Ví dụ về Message Type Indicator 23](#_Toc41116525)

[Hình 19 Một lĩnh vực được trình xuất hiện trong các bitmap 25](#_Toc41116526)

[Hình 20: Ví dụ bitmap 26](#_Toc41116527)

[Hình 21: Định nghĩa yếu tố dữ liệu 27](#_Toc41116528)

[Hình 22: Loại data element 28](#_Toc41116529)

[Hình 23: Giới thiệu JWT 29](#_Toc41116530)

[Hình 24: Áp dụng JWT 31](#_Toc41116531)

[Hình 25: JWT header 31](#_Toc41116532)

[Hình 26: JWT Payload 32](#_Toc41116533)

[Hình 27: JWT Signature 33](#_Toc41116534)

[Hình 28: Kiến trúc Rate Limiting 34](#_Toc41116535)

[Hình 29: Thuật toán Sliding Window 35](#_Toc41116536)

[Hình 30: Cách JWT hoạt động 36](#_Toc41116537)

[Hình 31: Tự động logout 30s khi thoát ứng dụng 36](#_Toc41116538)

[Hình 32: Chỉ đăng nhập được 1 thiết bị 37](#_Toc41116539)

[Hình 33: Bcryt 38](#_Toc41116540)

[Hình 34: Lượt đồ Class Diagram 41](#_Toc41116541)

[Hình 35: Usecase hệ thống 44](#_Toc41116542)

[Hình 36: Lượt đồ Usecase nhóm chức năng màng hình chính 45](#_Toc41116543)

[Hình 37: Lượt đồ Usecase nhóm chức năng tài khoản 49](#_Toc41116544)

[Hình 38: Luồng sự kiện nhóm chức năng tài khoản 50](#_Toc41116545)

[Hình 39: Luồng sự kiện nhóm chức năng quên mật khẩu 50](file:///C:\Users\16130\OneDrive\Desktop\SRS_All_FINAL_05.docx#_Toc41116546)

[Hình 40: Usecase nhóm chức năng nhận và chuyển tiền 61](#_Toc41116547)

[Hình 41: Flow nạp tiền vào ví 62](#_Toc41116548)

[Hình 42: Flow thêm tài khoản ngân hàng 67](#_Toc41116549)

[Hình 43: Flow chuyển tiền 70](#_Toc41116550)

[Hình 44: Usecase nhóm chức năng thanh toàn trực tiếp 76](#_Toc41116551)

[Hình 45: Flow liên kết chức năng nạp tiền điện thoại 77](#_Toc41116552)

[Hình 46: Lượt đồ chức năng mua thẻ điện thoại 84](#_Toc41116553)

[Hình 47: Flow chức năng mua vé xem phim 90](#_Toc41116554)

[Hình 48:Flow chức năng nạp tiền vào game 96](#_Toc41116555)

[Hình 49: lượt đồ chức năng thanh toán học phí 103](#_Toc41116556)

[Hình 50: Lượt đồ chức năng thanh toán hóa đơn (điện/nước) 109](#_Toc41116557)

MỤC LỤC

[LỜI CẢM ƠN I](#_Toc41121740)

[TÓM TẮT II](#_Toc41121741)

[DANH SÁCH CÁC HÌNH III](#_Toc41121742)

[MỤC LỤC V](#_Toc41121743)

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc41121744)

[1.1. Lý do chọn đề tài 1](#_Toc41121745)

[1.2. Mục đích đề tài 2](#_Toc41121746)

[1.3. Sản phẩm của đề tài 2](#_Toc41121747)

[1.4. Hiện trạng và khó khăn thực tế 2](#_Toc41121748)

[1.5. Thuật ngữ, từ viết tắt 3](#_Toc41121749)

[CHƯƠNG 2: MÔ TẢ TỔNG THỂ 4](#_Toc41121750)

[2.1. Các kiến trúc 4](#_Toc41121751)

[2.1.1. Kiến trúc hệ thống. 4](#_Toc41121752)

[2.1.2. Kiến trúc ứng dụng 4](#_Toc41121753)

[2.1.3. Kiến trúc phầm mềm 7](#_Toc41121754)

[2.2. Bối cảnh ví điện tử 8](#_Toc41121755)

[2.3. Chức năng sản phẩm 9](#_Toc41121756)

[2.4. Các đặc điểm người sử dụng 9](#_Toc41121757)

[2.5. Các ràng buộc 9](#_Toc41121758)

[2.6. Các giả định 10](#_Toc41121759)

[CHƯƠNG 3: TÓM TẮT REACT NATIVE 12](#_Toc41121760)

[3.1. Giới thiệu 12](#_Toc41121761)

[3.1.1. Điều kiện tiên quyết 12](#_Toc41121762)

[3.1.2. Ví dụ tương tác 12](#_Toc41121763)

[3.1.3. Function Components và Class Components 13](#_Toc41121764)

[3.2. Core Components và Native Components 14](#_Toc41121765)

[3.2.1. Views và mobile development 15](#_Toc41121766)

[3.2.2. Native Components 15](#_Toc41121767)

[3.2.3. Core Components 16](#_Toc41121768)

[3.3. React Fundamentals 17](#_Toc41121769)

[CHƯƠNG 4: TÓM TẮT NODEJS 18](#_Toc41121770)

[4.1. Giới Thiệu 18](#_Toc41121771)

[4.2. Những ứng dụng nên viết bằng Nodejs 18](#_Toc41121772)

[4.3. Những kết luận sai lầm về Nodejs 19](#_Toc41121773)

[4.4. Lý do để tự tin sử dụng Nodejs 19](#_Toc41121774)

[4.5. Những công ty lớn nào đang sử dụng Nodejs. 19](#_Toc41121775)

[CHƯƠNG 5: CÁC CÔNG NGHỆ BẢO MẬT 20](#_Toc41121776)

[5.1. ISO 8583 20](#_Toc41121777)

[5.1.1. Khái niệm về chuẩn ISO 8583 20](#_Toc41121778)

[5.1.2. Cấu trúc message ISO 8583 20](#_Toc41121779)

[5.2. JWT 29](#_Toc41121780)

[5.2.1. Header 31](#_Toc41121781)

[5.2.2. Payload 31](#_Toc41121782)

[5.2.3. Signature 33](#_Toc41121783)

[5.3. Rate Limiting 33](#_Toc41121784)

[5.3.1. Rate Limiting là gì? 33](#_Toc41121785)

[5.3.2. Tại sao API cần phải có Rate Limiting? 33](#_Toc41121786)

[5.3.3. Rate Limiting hoạt động như thế nào? 34](#_Toc41121787)

[5.3.4. Thuật toán sử dụng trong Rate Limiting 35](#_Toc41121788)

[5.3.5. Tự động logout 30s khi đang trạng thái chờ 36](#_Toc41121789)

[5.3.6. Chỉ đăng nhập được 1 thiết bị 36](#_Toc41121790)

[5.3.7. Bcrypt 37](#_Toc41121791)

[CHƯƠNG 6: YÊU CẦU CHỨC NĂNG 39](#_Toc41121792)

[6.1. Lượt đồ tổng quan 39](#_Toc41121793)

[6.1.1. Class Diagram 39](#_Toc41121794)

[6.1.2. Database diagram 42](#_Toc41121795)

[6.1.3. Usecase hệ thống 44](#_Toc41121796)

[6.2. Nhóm chức năng màn hình chính 45](#_Toc41121797)

[6.2.1. Chức năng quản lí thẻ/tài khoản 45](#_Toc41121798)

[6.2.2. Chức năng thông báo 46](#_Toc41121799)

[6.2.3. Chức năng cài đặt 47](#_Toc41121800)

[6.3. Nhóm chức năng tài khoản 49](#_Toc41121801)

[6.3.1. Chức năng đăng nhập 50](#_Toc41121802)

[6.3.2. Chức năng đăng ký 52](#_Toc41121803)

[6.3.3. Chức năng quên mật khẩu 54](#_Toc41121804)

[6.3.4. Chức năng lịch sử giao dịch 56](#_Toc41121805)

[6.3.5. Chức năng xem và thay đổi thông tin 58](#_Toc41121806)

[6.3.6. Chức năng thay đổi mật khẩu 59](#_Toc41121807)

[6.4. Nhóm chức năng nhận và chuyển tiền qua mạng 61](#_Toc41121808)

[6.4.1. Chức năng nạp tiền vào ví 62](#_Toc41121809)

[6.4.2. Chức năng thêm tài khoản ngân hàng 67](#_Toc41121810)

[6.4.3. Chức năng chuyển tiền 70](#_Toc41121811)

[6.5. Nhóm chức năng thanh toán trực tuyến 76](#_Toc41121812)

[6.5.1. Chức năng nạp tiền điện thoại 76](#_Toc41121813)

[6.5.2. Chức năng mua thẻ điện thoại 84](#_Toc41121814)

[6.5.3. Chức năng mua vé xem phim 90](#_Toc41121815)

[6.5.4. Chức năng nạp tiền vào game 96](#_Toc41121816)

[6.5.5. Chức năng thanh toán học phí 103](#_Toc41121817)

[6.5.6. Chức năng thanh toán hóa đơn (điện/nước) 108](#_Toc41121818)

[CHƯƠNG 7: YÊU CẦU PHI CHỨC NĂNG CỦA PHẦN MỀM 116](#_Toc41121819)

[7.1. Yêu cầu bảo mật 116](#_Toc41121820)

[7.2. Yêu cầu sao lưu 116](#_Toc41121821)

[7.3. Các yêu cầu về tính sử dụng (Usability) 116](#_Toc41121822)

[7.3.1. Các yêu cầu về tính ổn định (Reliability) 117](#_Toc41121823)

[7.3.2. Tính sẵn sàng (Availability) 117](#_Toc41121824)

[7.3.3. Tính chính xác 117](#_Toc41121825)

[7.4. Các yêu cầu về hiệu năng (Performance) 117](#_Toc41121826)

[7.5. Các yêu cầu về tính hỗ trợ (Supportability) 117](#_Toc41121827)

[7.6. Các ràng buộc thiết kế (Design Constraints) 117](#_Toc41121828)

[7.7. Giao tiếp (Interfaces) 118](#_Toc41121829)

[7.7.1. Giao tiếp người dùng (User interfaces) 118](#_Toc41121830)

[7.7.2. Giao tiếp phần cứng (Hardware interfaces) 118](#_Toc41121831)

[7.7.3. Giao tiếp phần mềm (Software interfaces) 118](#_Toc41121832)

[7.7.4. Giao tiếp bên ngoài 118](#_Toc41121833)

[7.7.5. Giao tiếp bên trong 118](#_Toc41121834)

[7.7.6. Giao tiếp truyền thông (Communications interfaces) 119](#_Toc41121835)

[7.7.7. Các tiêu chuẩn áp dụng 119](#_Toc41121836)

[7.7.8. Các yêu cầu khác 119](#_Toc41121837)

[CHƯƠNG 8: KẾT QUẢ ĐẠT ĐƯỢC 120](#_Toc41121838)

[TÀI LIỆU THAM KHẢO 122](#_Toc41121839)

# GIỚI THIỆU

## Lý do chọn đề tài

Thế kỷ 21 cũng có thể được hiểu là một kỷ nguyên của cuộc cách mạng kỹ thuật số nơi công nghệ bao trùm vào hầu hết mọi khía cạnh của cuộc sống của chúng ta. Với sự phát minh của công nghệ di động vào năm 1973, qua nhiều năm với sự tiến bộ trong công nghệ khoa học và tích hợp các lĩnh vực CNTT khác nhau vào các thiết bị di động đã dẫn đến sự ra đời của điện thoại thông minh vào năm 1992. Khái niệm dữ liệu cá nhân hỗ trợ (PDA) ra đời khi mà điện thoại di động được cho là có tiềm năng trong việc thực hiện nhiều nhiệm vụ bổ sung khác nhau của các cá nhân ngoài việc cung cấp kết nối. Thêm vào đó, với sự bùng nổ của Internet và phạm vi tương tác lớn hơn giữa nhiều người dùng khác nhau đã tạo ra nhiều cơ hội cho các doanh nghiệp trong các lĩnh vực học tập, thương mại, giải trí… . Một thay đổi quan trọng như vậy là sự ra mắt của thương mại điện tử. Với nhiều loại cửa hàng trực tuyến cung cấp đa dạng một loạt các sản phẩm cho người dùng thông qua Internet thực sự là một thay đổi lớn cho các thị trường hiện có trên toàn cầu. Người tiêu dùng không còn giới hạn về ranh giới địa lý nơi mà bất kỳ loại hàng hóa nào ở bất cứ đâu trên toàn cầu, có thể được truy cập tại bất kỳ thời điểm nào thông qua Internet.

Tương tự, ý tưởng thanh toán dựa trên thiết bị di động được coi là một bước tiến lớn khác trên thị trường toàn cầu như một phương tiện trao đổi. Bằng chứng lớn nhất cho thấy là trong suốt những năm qua, với sự thay đổi về hình thức công nghệ và kiến ​​thức, con người đã tìm cách phát triển các phương tiện trao đổi đến giai đoạn mà nơi bán và mua hàng hóa và dịch vụ đã được thực hiện càng dễ dàng và thuận tiện càng tốt. Từ các hình thức giao dịch truyền thống, hệ thống chuyển dần thẻ nhựa (ATM) và chuyển thành thanh toán qua điện thoại di động là một bằng chứng rõ ràng về cách con người sử dụng kiến ​​thức trong việc tạo ra sự khác biệt về tạo ra cách trao đổi hàng hóa và dịch vụ trong nước và ngoài nước.

Các tùy chọn nổi về thương mại điện tử nổi bật nhất hiện có là:

* Thẻ tín dụng ( Credit Card): Thẻ tín dụng giống như vay một khoản nợ ngắn hạn mà người tiêu dùng phải trả lại (thường trong vòng 30 ngày), nếu không người tiêu dùng sẽ bị tính lãi.
* Thẻ ghi nợ ( Debit Card): Sử dụng tiền thẻ ghi nợ là sử dụng tiền từ tài khoản của bạn. Với thẻ ghi nợ, người tiêu dùng chỉ có thể chi tiêu số tiền có sẵn cho họ.
* Internet Banking: Một hệ thống thanh toán điện tử cho phép khách hàng ngân hàng cụ thể hoặc bất kỳ tổ chức tài chính nào khác để thực hiện giao dịch thông qua Ngân hàng hoặc trang web của tổ chức tài chính.
* Ví điện tử: Đây là một tính năng điện tử cho phép bạn sử dụng các chế độ thanh toán di động khác mà không chuyển thông tin của bạn đến các trang web của bên thứ ba. Thanh toán ví điện tử được phân kênh thông qua các cổng thanh toán giữ cho thông tin được mã hóa hoàn toàn.

Ví điện tử là một hình thức thanh toán mới cùng với các ưu điểm quá nhiều so với nhược điểm của nó và nhược điểm hoàn toàn có thể khắc phục được. Đầu năm 2019, một trong những ứng dụng Ví điện tử phổ biến của Việt Nam là MoMo công bố trong năm 2018 họ đã tiếp cận được tới gần 10 triệu người dùng cùng với khối lượng giao dịch tăng gấp 3 lần so với năm 2017. Đây là con số gây bất ngờ lớn khi mới cuối tháng 12/2017, MoMo còn công bố số lượng người dùng của mình là hơn 5 triệu khách hàng. Qua đó ta thấy sự tiện lợi của Ví điện tử và số lượng người dùng Ví điện tử ngày càng tăng để đáp ứng nhu cầu đó Ví điện tử cần ra đời.

Một số Ví điện tử phổ biến hiện nay: Ngân lượng, BaoKim, Payoo, Mobivi, MoMo, SohaPay, 123pay…. Các loại Ví điện tử này được phát hành bởi các công ty trong nước và sử dụng phổ biến trong nước. Ví điện tử quốc tế: PayPal (phổ biến nhất), AlertPay, WebMoney, LiqPay, Moneybookers…. Ưu điểm của Ví điện tử này là**:** Mang đến cho khách hàng sự tiện lợi và an toàn, tiết kiệm thời gian làm việc, di chuyển. Dễ dàng thực hiện thanh toán, chuyển và nhận tiền. Có thể thực hiện các truy vấn về tài khoản, biết được sự biến động trong tài khoản của mình nhanh nhất. Thực hiện thanh toán bằng Ví điện tử vô cùng đơn giản bởi bạn chỉ cần có một chiếc điện thoại di động và có một tài khoản là có thể thực hiện được các giao dịch của mình.Nhược điểm của Ví điện tử:Dễ dàng bị mất tài khoản do máy tính, điện thoại thường xuyên truy cập vào website không đáng tin cậy.

## Mục đích đề tài

Mô tả về cấu trúc cơ sở dữ liệu được sử dụng để lưu trữ dữ liệu trong hệ thống thông qua các lượt đồ. Tài liệu dùng làm đầu vào của việc thiết kế, xây dựng và system test.

Phân tích, xây dựng ứng dụng Ví điện tử: Phân tích hệ thống, xây dựng một hệ thống mà một ví điện tử cơ bản cần phải có và hiện thực một số chức năng của ví điện tử với các nhóm chức năng cần đạt được:nhóm chức năng tài khoản, nhóm chức năng nhận và chuyển tiền**,** nhóm chức năng thanh toán trực tiếp.

Nghiên cứu các công nghệ để xây dựng Ví điện tử:React Native: để xây dựng một giao diện đơn giản, NodeJS: để xây dựng web server ( theo dạng REST APIs).

Nghiên cứu và áp dụng các công nghệ bảo mật trong việc trao đổi thông tin qua mạng như: JWT, ISO-8583, tự động logout 30s khi không sử dụng ứng dụng, chỉ đăng nhập được 1 thiết bị, mã hóa password, 1 IP gọi API 15 lần/1 phút.

## Sản phẩm của đề tài

Tài liệu đặc tả các chức năng của phần mềm và tóm tắt nghiên cứu công nghệ React Native và NodeJS.

Ứng dụng Ví điện tử trên nền tảng Android và IOS bao gồm các nhóm chức năng:

* Nhóm chức năng tài khoản : chức năng đăng nhập, chức năng đăng ký, chức năng quên mật khẩu, chức năng lịch sử giao dịch, chức năng xem và thay đổi thông tin, chức năng thay đổi mật khẩu.
* Nhóm chức năng nhận và chuyển tiền : chức năng nạp tiền vào ví, chức năng thêm tài khoản ngân hàng, chức năng chuyển tiền
* Nhóm chức năng thanh toán trực tiếp : chức năng nạp tiền điện thoại, chức năng mua thẻ điện thoại, chức năng mua vé xem phim, chức năng nạp tiền vào game, chức năng thanh toán học phí, chức năng thanh toán hóa đơn (điện/nước).

## Hiện trạng và khó khăn thực tế

Việt Nam đang là một trong những quốc gia có mức tăng trưởng thanh toán điện tử cao nhất thế giới với tốc độ khoảng 35%/năm. Điều này dẫn tới sự phát triển nhanh và rất mạnh của các loại hình thương mại điện tử. Nổi bật trong số đó là cuộc đua đầy sôi động của các ví điện tử trong vài năm gần đây.

Ví điện tử là một hình thức lưu giữ tiền trên mạng nên đòi một hệ thống bảo mật cao, an toàn. Trong quá trình xây dựng gặp nhiều khó khăn về việc liên kết hợp tác với Ngân hàng, nguồn API các hệ thống cung cấp đòi hỏi phải là một đối tác chiến lược. Vì vậy hầu hết các chức năng của hệ thống điều hoạt động trên API giả lập với sự hướng dẫn của công ty JITS.

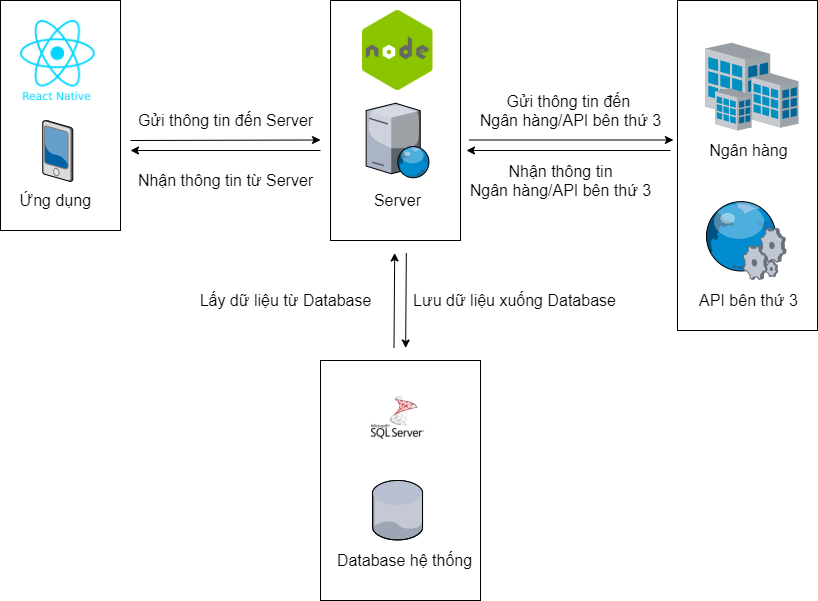
## Thuật ngữ, từ viết tắt

| Thuật ngữ | Định nghĩa | Ghi chú |
| --- | --- | --- |
| CSDL | Cơ sở dữ liệu |  |
| SC | Screen |  |
| TK | Tài khoản |  |
| MSSV | Mã số sinh viên |  |
| CMND | Chứng minh nhân dân |  |
| JITS | Công ty JUST-IN-TIME SOLUTIONS |  |

# MÔ TẢ TỔNG THỂ

## Các kiến trúc

### Kiến trúc hệ thống.



*Hình 1: Kiến trúc client – server.*

Mô tả kiến trúc : Ứng dụng gửi request đến cho server, sau khi nhận được request từ ứng dụng server sẽ lấy thông tin từ database hoặc gửi thông tin đến ngân hàng/API bên thứ 3 dựa vào chức năng mà ứng dụng yêu cầu, sau đó sẽ gửi thông tin trả về từ database hoặc thông tin từ ngân hàng/ API bên thứ 3 tương ứng với chức năng đã yêu cầu về ứng dụng cho người dùng.

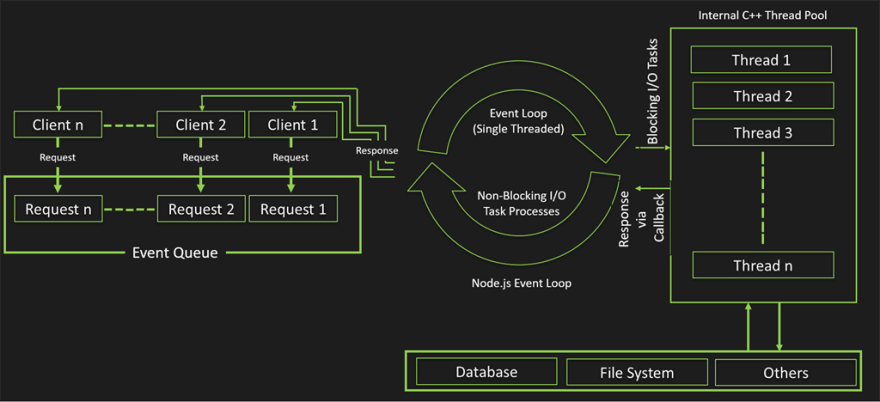
### Kiến trúc ứng dụng

#### Kiến trúc bất đồng bộ NodeJS

Khi chạy 1 chương trình bất kì, máy tính đều thực hiện các công việc tương tự nhau, hệ điều hành sẽ launch chương trình từ ổ cứng đó vào ram, khởi tạo 1 vùng nhớ cho việc chạy chương trình(heap, stack), khi chạy chương trình sẽ tương tác với các thành bên ngoài gọi là I/O(đọc file từ ổ cứng, giao tiếp với chuột/bàn phím, gọi tới và đợi kết quả từ các web server, ...).

Từng thành phần ở trên sẽ ảnh hưởng tới tổng thể hiệu năng của chương trình(bottle-neck). Nếu chương trình sử dụng nhiều tính toán như encode/decode, hashing, ... thì tổng thể hiệu năng phụ thuộc nhiều vào cpu, nên gọi đó là cpu-bound, nếu chương trình gọi tới nhiều web service hay database thì sẽ phụ thuộc vào tốc độ kết nối hay đáp ứng của IO, nên gọi đó là IO-bound, tương tự với memory-bound/cache-bound.

Với đặc thù của ứng dụng web phía client là gọi tới nhiều web service nên sẽ bị hạn chế rất nhiều ở IO. Nên giải quyết IO hẳn nhiên là ưu tiền hàng đầu của JS và giải pháp được JS chọn là cơ chế bất động bộ bằng event-loop.

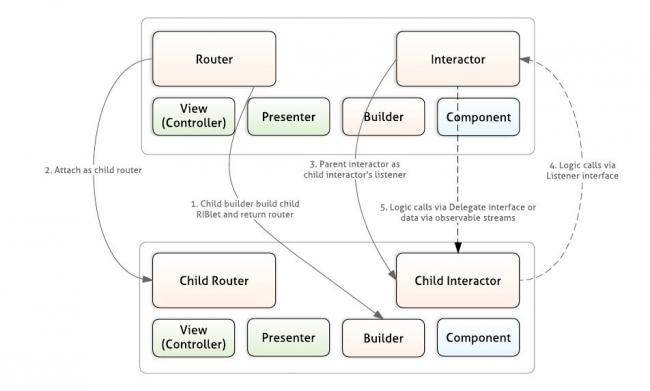


Hình : Kiến trúc ứng dụng NodeJS.

#### React Native

Ứng dụng RN truy cập trực tiếp đến toàn bộ APIs của ngôn ngữ Native. Như vậy một ứng dụng được xây dựng bằng react native cho hiệu năng và cảm nhận tương tự như một ứng dụng native.

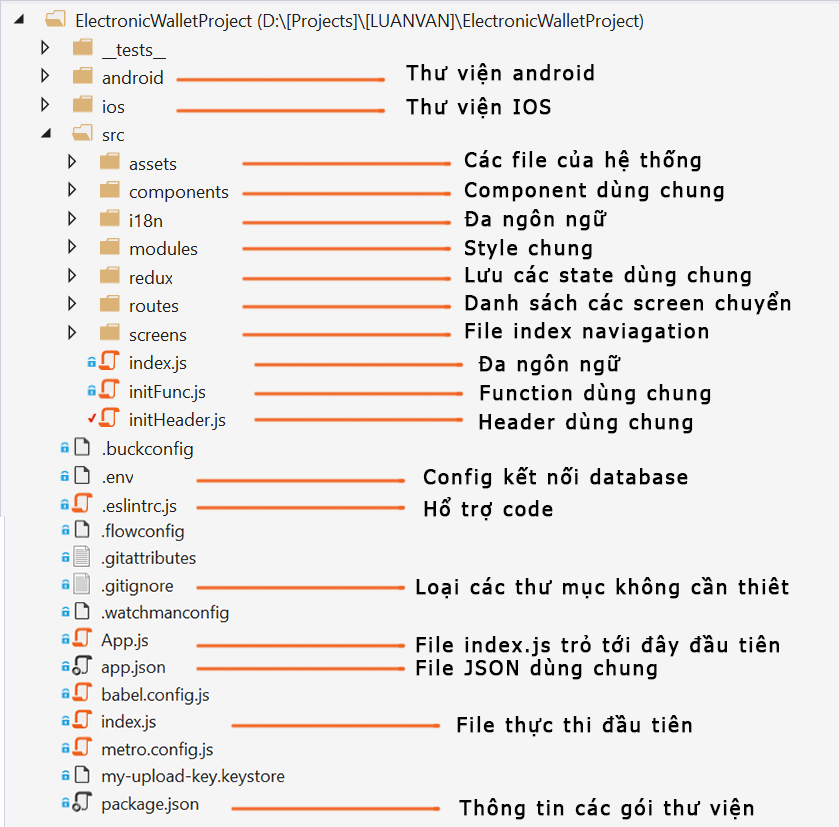
Đầu tiên, có thể dễ dàng giả định rằng React Native sẽ biên dịch ngôn ngữ JS code trực tiếp sang ngôn ngữ native tương ứng. Nhưng điều này thật sự khó khăn vì Objective C hay Java đều là những ngôn ngữ khá chặt chẽ. Thay vì thế, RN hoạt động một cách thông minh hơn. Về bản chất RN được xem như là một tập hợp các thành phần React (React components), mỗi component thế hiện tương ứng với view hay component nào đó của native.



Hình : Kiến trúc ứng dụng React Native.

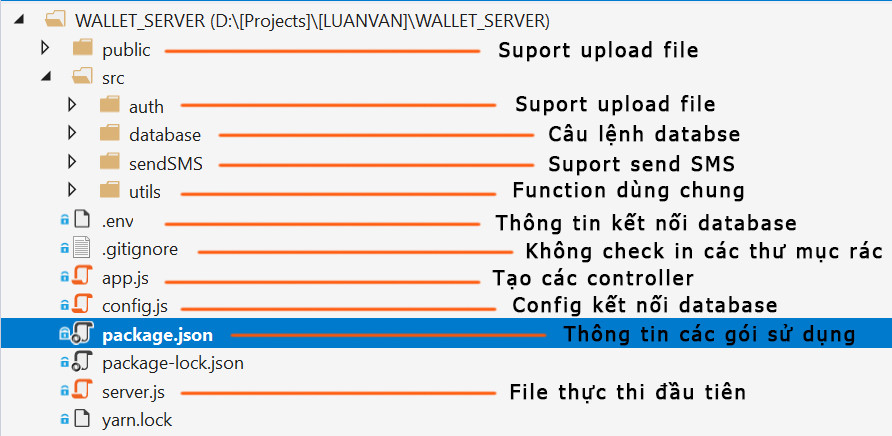
### Kiến trúc phầm mềm

#### React Native



Hình : Kiến trúc phầm mềm React Native

#### Nodejs



Hình : Kiến trúc phầm mềm NodeJS

## Bối cảnh ví điện tử

Trong bối cảnh thương mại điện tử phát triển như hiện nay, ví điện tử là một hình thức thanh toán mới, mang lại nhiều lợi ích và sự tiện lợi cho người sử dụng. Tuy nhiên, người tiêu dùng và doanh nghiệp Việt Nam chưa thực sự mặn mà với dịch vụ này mặc dù hiện nay đã có tới 25 doanh nghiệp cung cấp dịch vụ ví điện tử với số lượng lớn, hình thức đa dạng cả website và các ứng dụng trên di động. Bằng phương pháp thống kê mô tả, thống kê so sánh, nghiên cứu này đi sâu vào phân tích thực trạng sử dụng ví điện tử tại Việt Nam thời gian qua thông qua việc phân tích tình hình phát hành, đặc điểm các loại ví điện tử của các doanh nghiệp, tình hình sử dụng thực tế. Nghiên cứu cũng đi sâu làm rõ những tồn tại, hạn chế và những yếu tố làm cản trở sự phát triển hình thức thanh toán này nhằm đề xuất một số giải pháp thúc đẩy phát triển hình thức thanh toán ví điện tử ở Việt Nam trong thời gian tới.

## Chức năng sản phẩm

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Vai trò |
| 1 | Nhận và chuyển tiền dễ dàng qua mạng | Thực hiện thanh toán nhanh chóng, thuận tiện |
| 2 | Dùng để chi trả trực tuyến | Giảm sự quản lý giao dịch thông qua thẻ ngân hàng |
| 3 | Lưu trữ tiền trên internet | Giảm bớt sợ lưu thông của tiền mặt, hạn chế lạm phát |

## Các đặc điểm người sử dụng

Thời gian qua, nhiều ví điện tử tỏ rõ tham vọng bành trướng tại thị trường Việt Nam bằng việc kêu gọi các nhà đầu tư rót vốn, mở rộng thị phần. Trong năm 2018, số lượng khách hàng sử dụng ví điện tử đã đạt gần 10 triệu người dùng. Ông Phạm Thành Đức, Tổng Giám đốc ví điện tử, cho biết kế hoạch năm 2019 sẽ tăng gấp đôi lượng khách hàng này.

Người dùng ví điện tử chủ yếu là tầng lớp trẻ, sinh viên, người lao động.

Theo thống kê của Ngân hàng Nhà nước, thị trường Việt Nam vào năm 2013 có 1,84 triệu người sử dụng ví điện tử, và dự báo đến năm 2020 sẽ đạt 10 triệu người dùng. Tuy nhiên, vào thời điểm đầu năm 2019 đã đạt lượng 10 triệu người đăng kí sử dụng dịch vụ.

## Các ràng buộc

Người Sử Dụng sẽ được xem là đương nhiên chấp nhận và chịu sự ràng buộc của những Điều Khoản Chung này và được coi như là Hợp đồng giữa J\_Service và Người Sử Dụng khi Người Sử Dụng sử dụng bất kỳ phần nào của Sản Phẩm/Dịch Vụ và trong suốt quá trình sử dụng Sản Phẩm/Dịch Vụ của J\_Service.

J\_Service có quyền thay đổi những Điều Khoản Chung này hoặc bất kỳ tính năng nào của Sản Phẩm/Dịch Vụ vào bất kỳ thời điểm nào. Thay đổi đó sẽ có hiệu lực ngay lập tức sau khi công bố thay đổi của các Điều Khoản Chung hoặc tính năng trên trang web. Người Sử Dụng đồng ý đánh giá những Điều Khoản Chung này định kỳ để đảm bảo rằng họ được cập nhật đối với bất kỳ các thay đổi hoặc sửa đổi đối với những Điều Khoản Chung này. Việc Người Sử Dụng tiếp tục sử dụng Sản Phẩm/Dịch Vụ sẽ được xem là Người Sử Dụng chấp nhận hoàn toàn các Điều Khoản Chung được thay đổi.

Người Sử Dụng đảm bảo rằng Người Sử Dụng đã hiểu rõ các hướng dẫn và quy trình sử dụng Sản Phẩm/Dịch Vụ của J\_Service và những thay đổi, bổ sung (nếu có) của J\_Service.

Nếu Người Sử Dụng không đồng ý chịu sự ràng buộc của các Điều Khoản Chung hiện hành, Người Sử Dụng không được sử dụng Sản Phẩm/Dịch Vụ và sẽ ngừng sử dụng Sản Phẩm/Dịch Vụ ngay lập tức.

## Các giả định

“Dịch Vụ Khách Hàng” nghĩa là dịch vụ chăm sóc khách hàng của J\_Service, được cung cấp theo số điện thoại (08) 39917XXX hoặc 1900545XXX;

“Điểm Giao Dịch” là các điểm được Nhà Cung Cấp ủy quyền để Người Sử Dụng thực hiện các Dịch Vụ Tại Điểm Giao Dịch. Danh sách Điểm Giao Dịch được liệt kê tại đây.

“Giao Dịch” nghĩa là bất kỳ giao dịch nào liên quan đến việc sử dụng Sản Phẩm/Dịch Vụ được cung cấp bởi J\_Service của Người Sử Dụng.

“Giấy Tờ Tùy Thân Hợp Lệ” nghĩa là (a) chứng minh nhân dân/thẻ căn cước công dân có hiệu lực, hoặc (b) hộ chiếu hợp lệ, hoặc (c) giấy phép lái xe hợp lệ;

“J\_Service” nghĩa là Công Ty Cổ Phần Dịch Vụ Just in time, một công ty được thành lập hợp lệ và hoạt động theo pháp luật của nước Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam.

“Người Sử Dụng” nghĩa là cá nhân/tổ chức sử dụng Sản Phẩm/Dịch Vụ của J\_Service.

“Quy Định Pháp Luật” bao gồm toàn bộ các qui định của Việt Nam như Luật, Bộ luật, Pháp lệnh, Nghị định, Thông tư, quy chuẩn, quy tắc, quy định hành chính, và các quy định có hiệu lực pháp luật khác theo thời gian.

“Sản Phẩm/Dịch Vụ” bao gồm Dịch Vụ Tại Điểm Giao Dịch và Dịch Vụ Trên Ứng Dụng.

“Tài Khoản ” nghĩa là tài khoản tiền trên di động do Người Sử Dụng duy trì được sử dụng cho mục đích thực hiện Các Giao Dịch;

“Ứng Dụng ” (Ví Điện Tử): là Ứng dụng ví điện tử do J\_Service cung cấp, quản lý và được Ngân hàng Nhà nước Việt Nam cấp phép triển khai dưới thương hiệu. Đây là tài khoản điện tử trên hệ thống J\_Service mà chủ tài khoản có thể sử dụng và quản lý thông qua nhiều hình thức khác nhau, bao gồm nhưng không giới hạn bởi ứng dụng trên điện thoại di động, website, SIMCARD...

# TÓM TẮT REACT NATIVE

Nhiều người khác nhau sử dụng React Native: từ những nhà phát triển iOS tiên tiến đến những người mới bắt đầu React, cho đến những người bắt đầu lập trình lần đầu tiên trong sự nghiệp. Những tài liệu này được viết cho tất cả người học, bất kể mức độ kinh nghiệm hoặc nền tảng của họ.

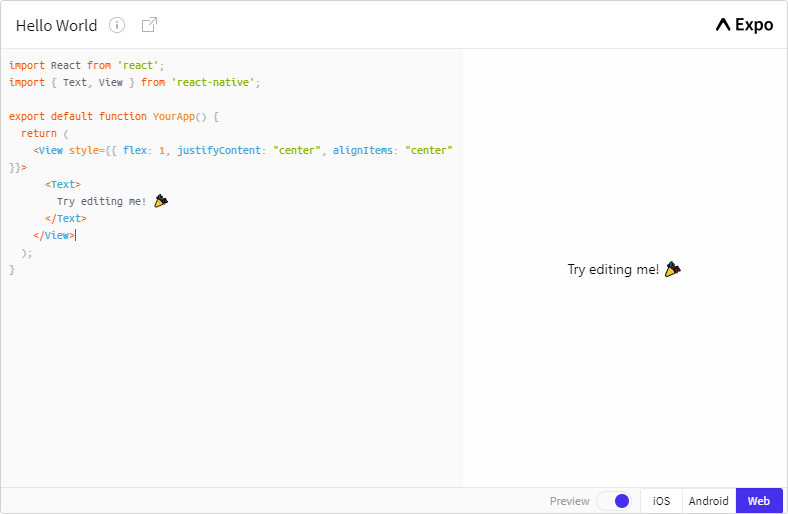
## Giới thiệu

### Điều kiện tiên quyết

React Native là một framework mã nguồn mở được sáng tạo bởi Facebook. Nó được sử dụng để phát triển ứng dụng di động Android và IOS bằng các cho phép các nhà phát triển sử dụng React cùng với môi trường ứng dụng gốc (native).

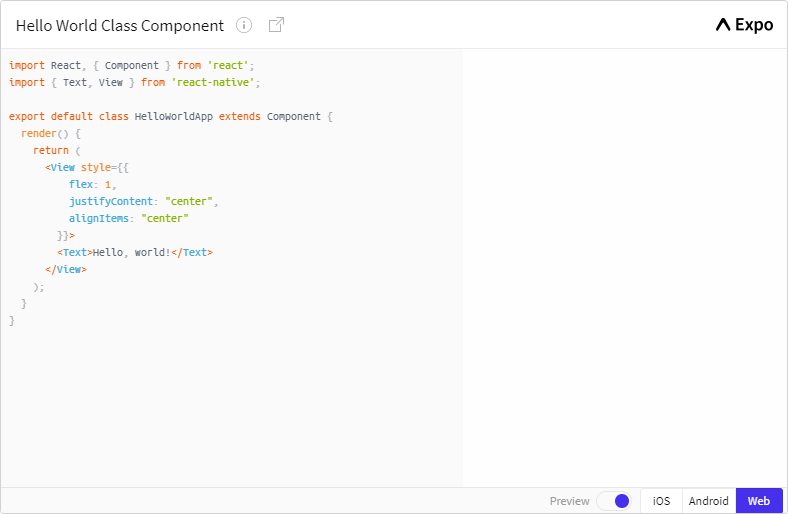
Để làm việc với React Native, bạn sẽ cần có hiểu biết về các nguyên tắc cơ bản của JavaScript

### Ví dụ tương tác

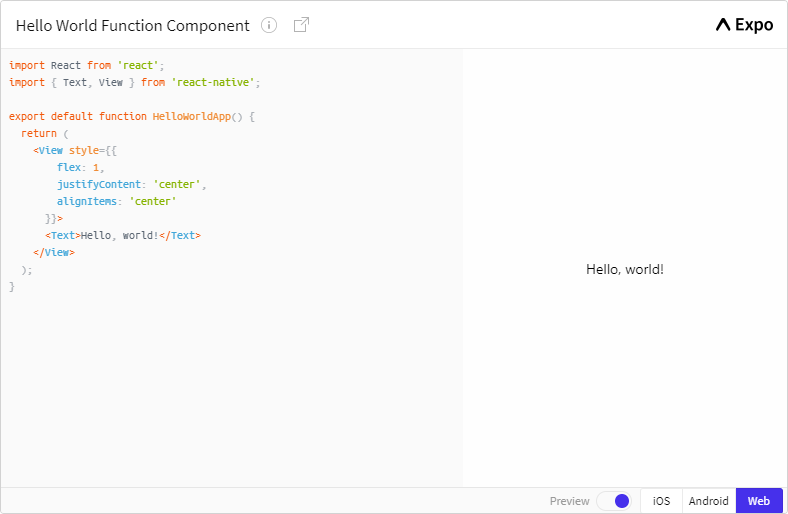


Hình : Bắt đầu với React Native

### Function Components và Class Components

Với React, bạn có thể tạo các thành phần bằng cách sử dụng các Class hoặc các Function. Ban đầu, các thành phần Class là các thành phần duy nhất có thể có các State. Nhưng kể từ khi API Hooks được giới thiệu, bạn có thể thêm State và nhiều thứ khác vào Function Components.

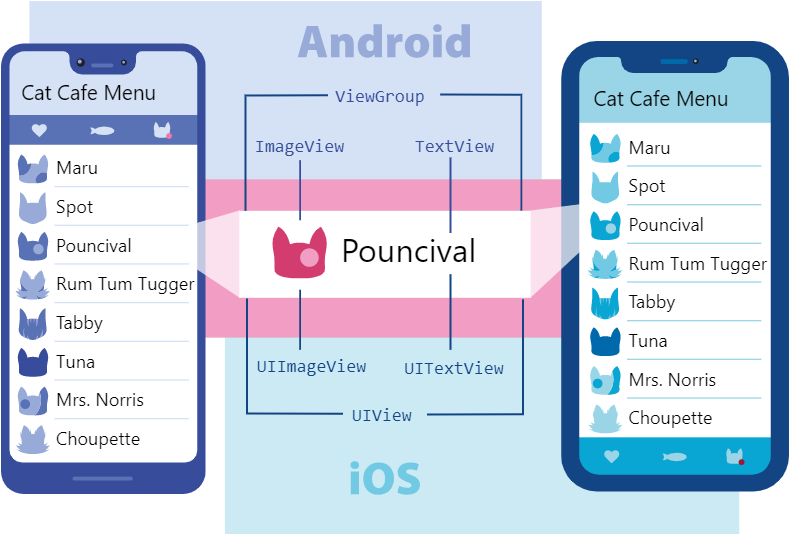
Hình : Demo Class Component



Hình : Demo Function Component

## Core Components và Native Components

React Native là một framework mã nguồn mở để xây dựng các ứng dụng Android và iOS bằng React. Với React Native sử dụng JavaScript để truy cập vào các API và cũng dùng để mô tả ngoại hình và các thay đổi trên UI của bạn bằng việc sử dụng React Components.



Hình : Một ví dụ về việc sử dụng nhiều View trên ứng dụng Android và iOS

### Views và mobile development

Trong phát triển Android và iOS, View là khối giao diện người dùng cơ bản: một thành phần hình chữ nhật nhỏ trên màn hình có thể được sử dụng để hiển thị văn bản, hình ảnh hoặc phản hồi đầu vào của người dùng. Ngay cả các yếu tố hình ảnh nhỏ nhất của một ứng dụng, như một dòng văn bản hoặc nút, cũng là các dạng View. Một số loại View có thể chứa các View khác. Nó xem tất cả là một khối!

### Native Components

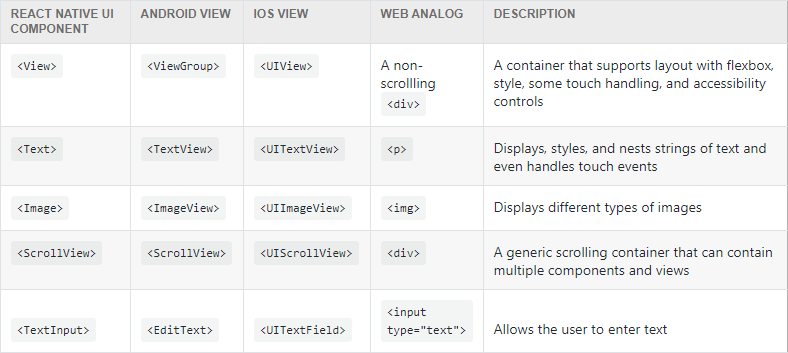
Khi phát triển ứng dụng Android, ta viết các View bằng Kotlin hoặc Java; trong phát triển ứng dụng iOS, ta sử dụng Swift hoặc Objective-C. Với React Native, ta có thể gọi các View này với JavaScript bằng các thành phần React. Khi chạy, React Native tạo các View Android và iOS tương ứng cho các thành phần đó. Do các thành phần React Native được hỗ trợ bởi các chế độ giống như Android và iOS, nên các ứng dụng React Native trông, cảm nhận và hoạt động như bất kỳ ứng dụng nào khác.

React Native cho phép bạn xây dựng các Component riêng cho mình phù hợp với các nhu cầu của ứng dụng. React Native cũng có một hệ sinh thái đóng góp để xây dựng thành nhiều bộ thư viện.

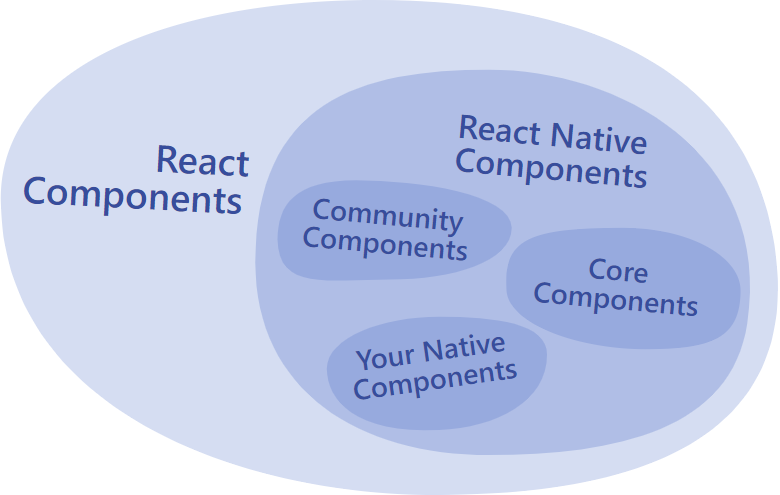
React Native cũng bao gồm một tập hợp các thành phần native (gốc) cần thiết, sẵn sàng sử dụng mà bạn có thể sử dụng để bắt đầu xây dựng ứng dụng của mình.

### Core Components

React Native có nhiều thành phần cốt lõi cho mọi thứ. Bạn có thể tìm thấy tất cả chúng được ghi lại trong phần API. Bạn sẽ làm việc chủ yếu với các thành phần cốt lõi sau:



Hình : Thành Phần Cơ Bản React Native



Hình : Sơ đồ các thành phần React

## React Fundamentals

React Native chạy trên React, một thư viện mã nguồn mở phổ biến để xây dựng giao diện người dùng với JavaScript. Để tận dụng tối đa React Native, nó giúp hiểu chính React. Phần này có thể giúp bạn bắt đầu hoặc có thể phục vụ như một khóa bồi dưỡng.

Gồm các thành phần cơ bản sau:

* Components
* JSX
* Props
* State

# TÓM TẮT NODEJS

## Giới Thiệu

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

## Những ứng dụng nên viết bằng Nodejs

Rõ ràng, không phải mới là Nodejs làm gì cũng tốt, ví dụ như một ứng dụng cần tính ổn định cao, logic phức tạp thì các ngôn ngữ PHP hay Ruby… vẫn là sự lựa chọn tốt hơn. Còn dưới đây là những ứng dụng có thể và nên viết bằng Nodejs:

**Websocket server:** Các máy chủ web socket như là Online Chat, Game Server…

**Fast File Upload Client:** là các chương trình upload file tốc độ cao (với file dung lượng vừa và nhỏ).

**Ad Server:** Các máy chủ quảng cáo.

**Cloud Services:** Các dịch vụ đám mây.

**RESTful API:** đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.

**Any Real-time Data Application:** bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

## Những kết luận sai lầm về Nodejs

Thứ nhất, Nodejs là một nền tảng (platform), không phải Web Framework, cũng không phải ngôn ngữ lập trình.

Thứ hai, Nodejs không hỗ trợ đa luồng, nó là một máy chủ đơn luồng.

Và một điều nữa, Nodejs không dành cho người mới tinh mà chưa biết gì về lập trình, vì như đã nói ở trên, Nodejs không phải là ngôn ngữ lập trình, để học được Nodejs thì bạn cần phải biết về Javascript, kỹ thuật lập trình, một số giao thức…v..v

## Lý do để tự tin sử dụng Nodejs

Các ứng dụng Nodejs được viết bằng **javascript**, ngôn ngữ này là một ngôn ngữ khá thông dụng. Theo tác giả của ngôn ngữ Javascript, Ryan Dahl: “Javascript có những đặc tính mà làm cho nó rất khác biệt so với các ngôn ngữ lập trình động còn lại, cụ thể là nó không có khái niệm về đa luồng, tất cả là đơn luồng và hướng sự kiện.”

Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.

Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí. Các bạn có thể thấy cộng đồng Nodejs lớn như thế nào tại đây, các package đều hoàn toàn free

## Những công ty lớn nào đang sử dụng Nodejs.

Amazon, Ebay, Linkedin, Microsoft, Paypal, trello, Uber và còn nhiều cái tên nổi tiếng khác nữa. Theo như Paypal thì sử dụng Nodejs làm giảm thời gian đáp ứng lên tới 35%

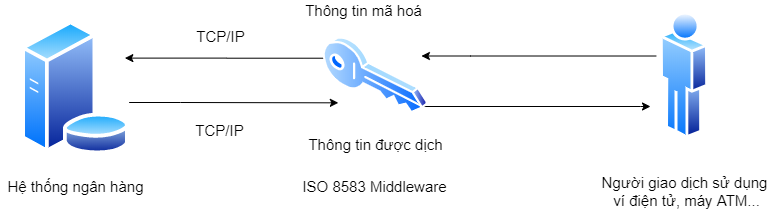
Còn về phía Linkedin chuyển từ Ruby sang sử dụng Nodejs để xử lý các truy cập từ mobile, và con số Server sử dụng giảm từ 30 còn 3, nghĩa là giảm gần 90%.

# CÁC CÔNG NGHỆ BẢO MẬT

## ISO 8583

Ví điện tử giao dịch với ngân hàng bằng các thông điệp theo chuẩn ISO8583 trên nền TCP/IP

Trong đó ứng dụng truyền đi thông tin đi tới Middleware là hệ thống server, ở đấy các thông tin sẽ được mã hoá theo giao thức ISO 8583 các gói tin trên nền TCP/IP tới hệ thống ngân hang. Sau đó ngân hàng sẽ truyền lại các gói tin thông qua giao thức ISO 8583, server sẽ dịch ngược lại thông tin và gửi về kết quả cho người dung.



Hình : Sơ đồ hoạt động JWT

### Khái niệm về chuẩn ISO 8583

Tiêu chuẩn ISO 8583 là tiêu chuẩn tài chính cho các giao dịch trong ngân hàng có tạo ra các thông điệp(messages) - trao đổi thông điệp với các đặc tính kỹ thuật được quy định thuộc chuẩn quốc tế là các tiêu chuẩn cho các hệ thống trao đổi các giao dịch điện tử.

### Cấu trúc message ISO 8583

Cấu trúc của message ISO 8583 gồm: Thông điệp chỉ thị (Message Type Indicator), Các Bitmaps, Các yếu tố dữ liệu kèm theo.

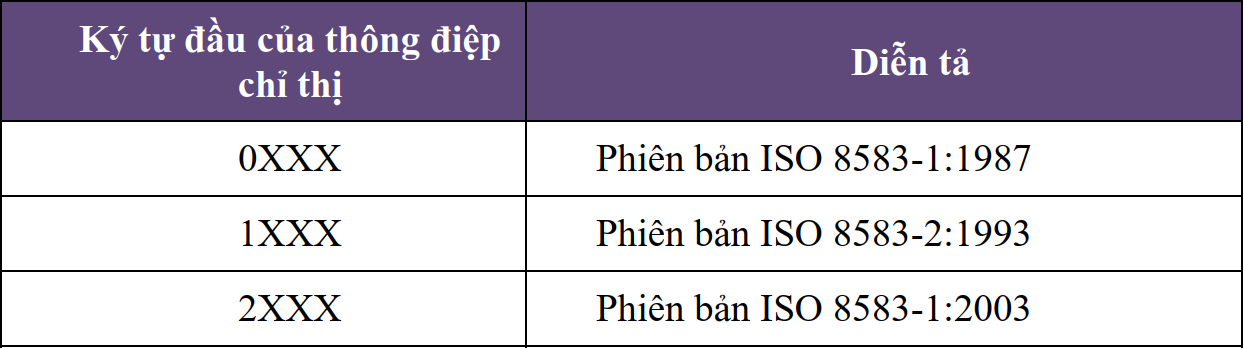


Hình 13: Cấu trúc message ISO 8583

#### Thông điệp chỉ thị (Message Type Indicator)

Là một mã gồm 4 kí tự, nó chứa thông tin về các kiểu message hiện hành, bao  
gồm những thông tin về phiên bản ISO, chức năng của message và ai đã gửi chúng.

Ký tự đầu trong 4 ký tự của thông điệp chỉ thị chứa thông tin về kiểu ISO hiện  
hành:



Hình : Mô tả Message Type Indicator (1)

Ký tự thứ 2 trong 4 ký tự của thông điệp chỉ thị xác định mục đích của thông  
điệp:



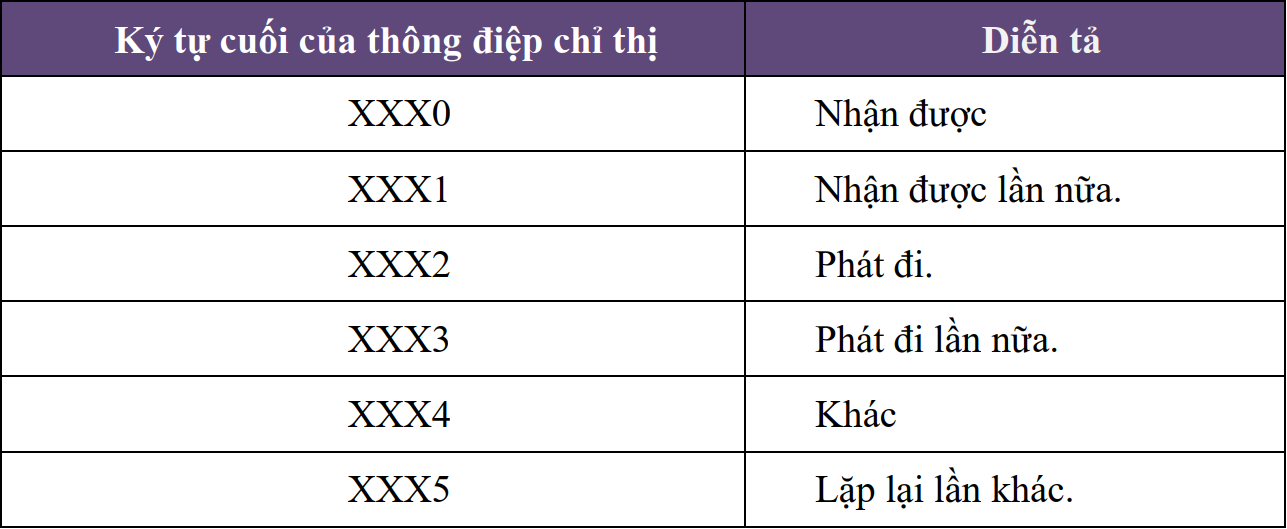
Hình : Mô tả Message Type Indicator (2)

Ký tự thứ 3 trong 4 ký tự của thông điệp chỉ thị xác định chức năng của thông điệp:



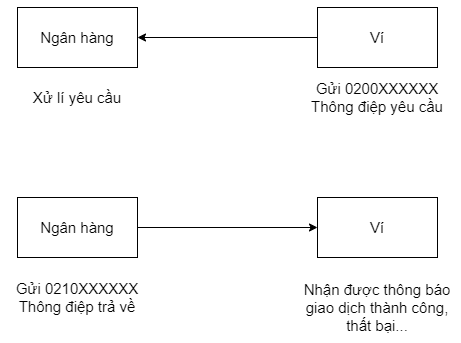
Hình : Mô tả Message Type Indicator (3)

Ký tự cuối trong 4 ký tự của thông điệp chỉ thị xác định xuất xứ của thông  
điệp:



Hình : Mô tả Message Type Indicator (4)

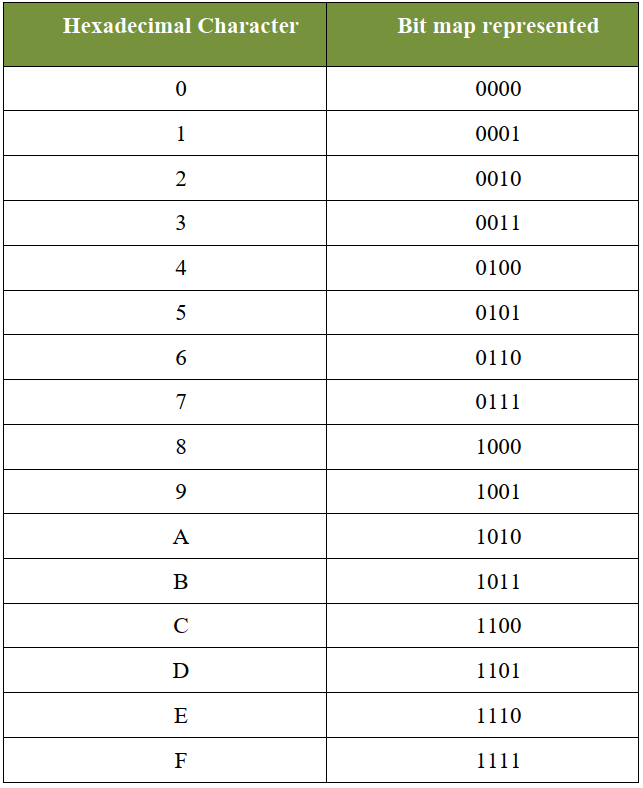
Ví dụ minh hoạ: Thông điệp tài chính mang mã định danh 02XX: khi ở trạng thái này nó sẽ bắt đầu với mã là 0200 từ nơi yêu cầu và sẽ trả lời lại nơi gửi một message với tiêu đề 0210 và chỉ ra rằng đó là thông điệp trả lời từ những yêu cầu trước đó.



Hình : Ví dụ về Message Type Indicator

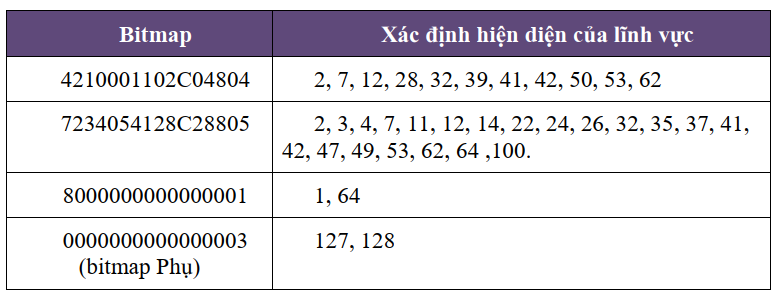
#### Bitmap

Trong chuẩn ISO 8583, một bipmap là một lĩnh vực hoặc lĩnh vực phụ nào đó đã được quy định. Trong 1 thông điệp có thể có tới 3 Bitmap. Trong 1 thông điệp có ít nhất 1 bitmap, gọi là bitmap chính, nó chỉ ra phần tử dữ liệu hiện diện từ 1 tới 64.  
Một bitmap phụ cũng có thể được đưa ra, bitmap này chỉ ra sự hiện diện của các phần tử khác từ 65 tới 128. Tương tự, một bitmap thứ 3 có thể được sử dụng để chỉ sự hiện diện hoặc vắng mặt trong lĩnh vực 129 tới 192, mặc dù những dữ liệu này rất ít gặp. Các bitmap có thể đƣợc truyền bằng 8 bytes dữ liệu nhị phân hoặc 16 kí tự hex  
0-9, A-F trong các kí tự ASCII hoặc bộ kí tự EBCDIC.

Một lĩnh vực được trình bày khi có xuất hiện các bitmap tương ứng trong thông  
điệp, giá trị là 1 khi có mặc là 0 khi vắng mặt các lĩnh vực tương ứng

Hình Một lĩnh vực được trình xuất hiện trong các bitmap

.

****

Hình : Ví dụ bitmap

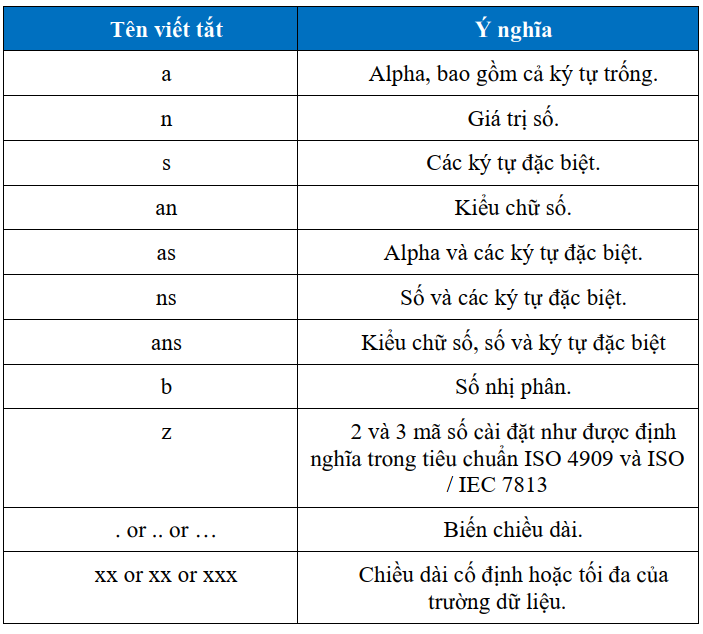
Giải thích bitmap thuộc lĩnh vực 4210001102C04804 :

* 01000010 = 42x (Đếm từ phía bên trái, Vị trí thứ 2 và thứ 7 là bit 1, nên có sự hiện
* diện của lĩnh vực 2 và 7)
* 00010000 = 10x (hiện diện của lĩnh vực 12)
* 00000000 = 00x (Không có lĩnh vực nào)
* 00010001 = 11x (hiện diện của lĩnh vực 28 và 32)
* 00000010 = 02x (hiện diện của lĩnh vực 39)
* 11000000 = C0x (hiện diện của lĩnh vực 41 và 42)
* 01001000 = 48x (hiện diện của lĩnh vực 50 và 53)
* 00000100 = 04x (hiện diện của lĩnh vực 62)

#### Yếu tố dữ liệu:

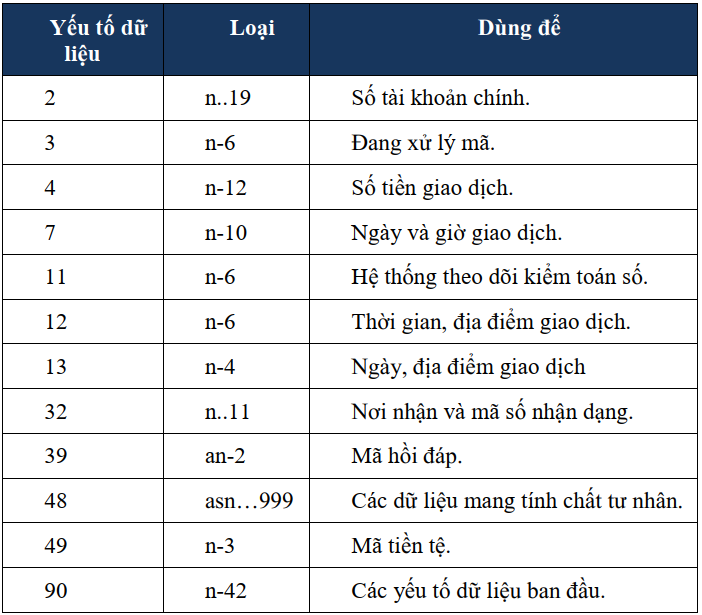
Là các yếu tố dữ liệu mang tính chất cá nhân như các lĩnh vực triển khai, các  
thông tin giao dịch. Hiện có đến 128 yếu tố dữ liệu đầu được xác định trong chuẩn  
ISO 8583:1987 và lên đến 192 yếu tố dữ liệu trong phiên bản sau này. Các phiên  
bản năm 1993 mới được thêm vào các định nghĩa, xóa một số tuy nhiên định dạng  
vẫn không thay đổi.

Trong khi đó mỗi yếu tố dữ liệu mang một yếu tố xác định ý nghĩa và định dạng,  
mỗi yếu tố dữ liệu được diễn tả trong một định dạng chuẩn mà xác định nội dung  
được cho phép của các trường (Số, nhị phân,...) và các lĩnh vực (Hoặc các biên cố  
định) theo bảng sau:



Hình : Định nghĩa yếu tố dữ liệu

Ngoài ra, trong mỗi trường có thể cố định hoặc biển đổi chiều dài. Nếu có thay  
đổi theo độ dài thì độ dài của trường sẽ được chỉ ra bởi biến báo độ dài

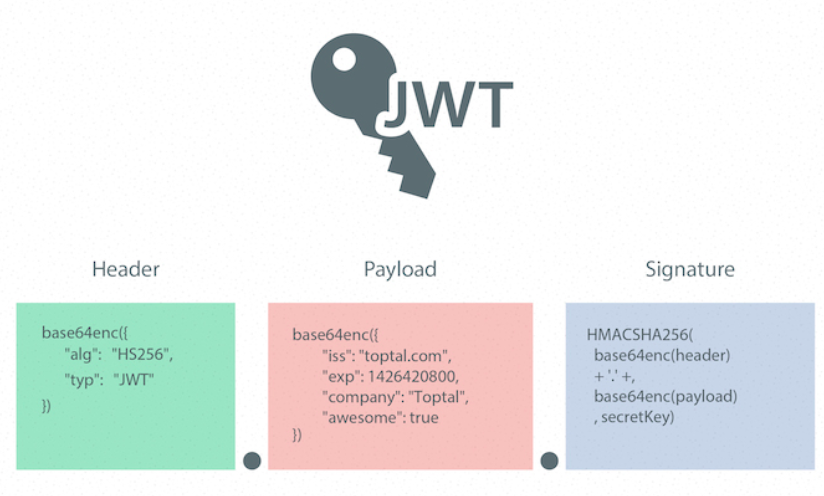


Hình : Loại data element

* Ví dụ về thông điệp ISO 8583:
* Ví dụ về thông điệp về tài chính được gửi đi: 0200823A0000000000000400000000000000042009061390000109061304200420001
* Ví dụ về thông điệp về tài chính được trả về:  
  0210823A00000200000004800000000000000420090613900001090613042004200000103112

## JWT

JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

****

Hình : Giới thiệu JWT

Nói một cách khác, JWT là sự kết hợp (bởi dấu .) một Object Header dưới định dạng JSON được encode base64, một payload object dưới định dạng JSON được encode base64 và một Signature cho URI cũng được mã hóa base64 nốt.

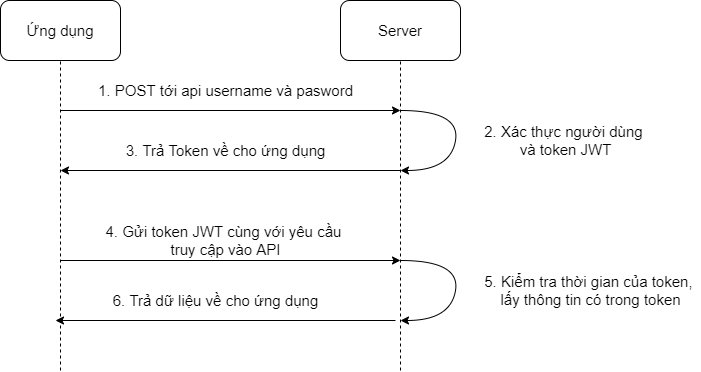
Một trong những tình huống ứng dụng JWT thường gặp, đó là:

* Xác thực: Tình huống thường gặp nhất, khi user logged in, mỗi request tiếp đó đều kèm theo chuỗi token JWT, cho phép người dùng có thể truy cập đường dẫn, dịch vụ và tài nguyên được phép ứng với token đó. Single Sign On cũng là một chức năng có sử dụng JWT một cách rộng rãi, bởi vì chuỗi JWT có kích thước đủ nhỏ để đính kèm trong request và sử dụng ở nhiều hệ thống thuộc các domain khác nhau.
* Trao đổi thông tin: JSON Web Token cũng là một cách hữu hiệu và bảo mật để trao đổi thông tin giữa nhiều ứng dụng, bởi vì JWT phải được ký (bằng cặp public / private key), bạn có thể chắc rằng người gửi chính là người mà họ nói rằng họ là (nói tóm tắt hơn là không hoặc khó để mạo danh bằng JWT), ngoài ra, chữ ký cũng được tính toán dựa trên nội dung của header và nội dung payload, nhờ đó, bạn có thể xác thực được nội dung là nguyên bản, chưa được chỉnh sửa hoặc can thiệp. Tuy nhiên, một lưu ý hết sức quan trọng là do cấu trúc của JWT đơn giản nên JWT có thể dễ dàng bị decode, do vậy, bạn không nên dùng JWT để transfer các thông tin nhạy cảm.

**Cách JWT hoạt động**

* Ở đây, mình ví dụ cụ thể ứng dụng của JWT trong bài toán Authenticate (xác thực) Trong việc xác thực, khi user đăng nhập thành công (Browser sẽ post username và mật khẩu về Server), Server sẽ trả về một chuỗi JWT về Browser, và Token JWT này cần được lưu lại trong Browser của người dùng (thường là LocalStorage hoặc Cookies), thay vì cách truyền thống là tạo một session trên Server và trả về Cookie.
* Bất cứ khi nào mà User muốn truy cập vào Route được bảo vệ (mà chỉ có User đã đăng nhập mới được phép), Browser sẽ gửi token JWT này trong Header Authorization, Bearer schema của request gửi đi.

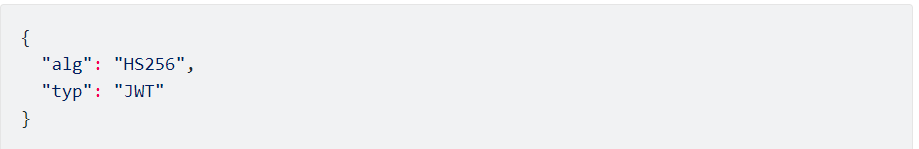
Đây là cách mà stateless (phi trạng thái) authentication làm việc, trạng thái của user không được lưu trong bộ nhớ của Server mà được đóng gói hẳn vào trong JWT. Server sẽ kiểm tra Token JWT này có hợp lệ hay không (Bởi vì JWT có tính chất self-contained, mọi thông tin cần thiết để kiểm tra JWT đều đã được chứa trong Token JWT). Do tính chất stateless nên chúng ta không còn phải lo lắng về domains nào được sử dụng cho API của bạn, như không còn gặp rắc rối với CORS (Cross-Origin Resource Sharing) vì nó không sử dụng cookies. Sơ đồ dưới đây cho thấy quá trình này



Hình : Áp dụng JWT

### Header

Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).



Hình : JWT header

### ****Payload****

Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.

Reserved claims: Đây là một số metadata được định nghĩa trước, trong đó một số metadata là bắt buộc, số còn lại nên tuân theo để JWT hợp lệ và đầy đủ thông tin: iss (issuer), iat (issued-at time) exp (expiration time), sub (subject), aud (audience), jti (Unique Identifier cho JWT, Can be used to prevent the JWT from being replayed. This is helpful for a one time use token.) ... Ví dụ:



Hình : JWT Payload

#### **Public Claims**

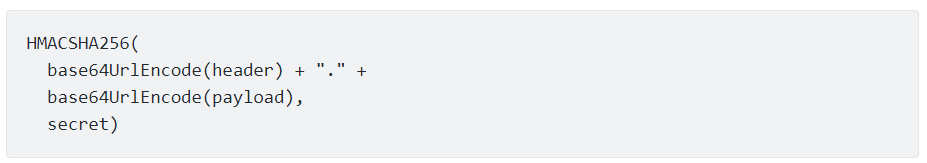
Claims được cộng đồng công nhận và sử dụng rộng rãi.

#### **Private Claims**

Claims tự định nghĩa (không được trùng với Reserved Claims và Public Claims), được tạo ra để chia sẻ thông tin giữa 2 parties đã thỏa thuận và thống nhất trước đó.

### ****Signature****

Chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:



Hình : JWT Signature

## Rate Limiting

### Rate Limiting là gì?

Giả sử như hệ thống của chúng ta nhận được hàng nghìn request nhưng mà trong số đó chỉ xử lí được trăm request/s chẳng hạn, và số request còn lại thì bị lỗi (do CPU hệ thống đang quá tải không thể xử lí được).

Để giải quyết được vấn đề này thì cơ chế Rate Limiting đã ra đời. Mục đích của nó chỉ cho phép nhận 1 số lượng request nhất định trong 1 đơn vị thời gian. Nếu quá sẽ trả về response lỗi.

1 số ví dụ hay gặp trong Rate Limiting như:

* Mỗi người dùng chúng ta chỉ cho phép gửi 100 request/s. Nếu vượt quá thì sẽ trả về response lỗi.
* Mỗi người dùng chỉ cho phép nhập sai thẻ credit 3 lần trong 1 ngày
* Mỗi địa chỉ IP chỉ có thể tạo được 2 account trong 1 ngày.

**Về cơ bản thì rate limiter sẽ cho phép gửi bao nhiêu request trong 1 khoảng thời gian nào đó. Nếu vượt quá sẽ bị block lại.**

### Tại sao API cần phải có Rate Limiting?

Rate Limiting sẽ giúp chúng ta hạn chế được 1 số việc sau:

* Tấn công DOS (Denial of Service) đến hệ thống
* Brute force password trong hệ thống (quyét kiểu vét cạn)
* Brute force thông tin thẻ credit card (quyét kiểu vét cạn)

Những cuộc tấn công này nhìn có vẻ như đến từ người dùng thực, nhưng thực tế nó được tạo ra bởi bot. Do vậy mà nó rất khó bị phát hiện và dễ dàng làm service của chúng ta bị sập.

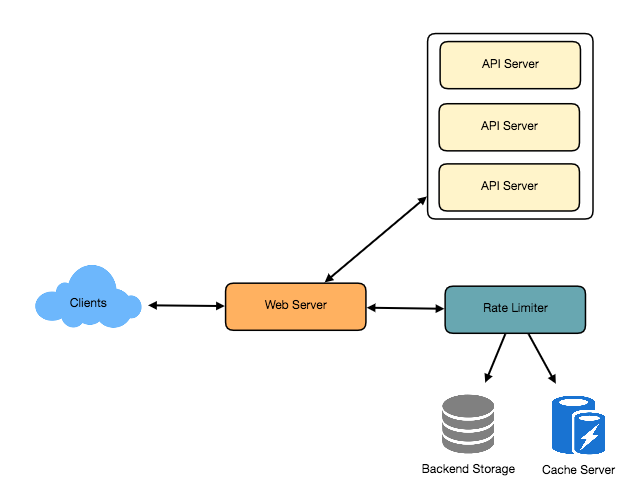
Ngoài ra, Rate Limiting cũng mang lại cho chúng ta 1 số lợi ích sau:

* Bảo mật: Không cho phép nhập sai password quá nhiều lần
* Doanh thu: Với mỗi plan sẽ có rate limit khác nhau. Nếu muốn dùng nhiều hơn thì cần mua lên plan đắt tiền hơn.

### Rate Limiting hoạt động như thế nào?

Rate Limiting có 1 process tên là Throttling dùng để kiểm soát tình trạng sử dụng API của người dùng trong 1 khoảng thời gian nhất định.

Kiến trúc Rate Limiting:



Hình : Kiến trúc Rate Limiting

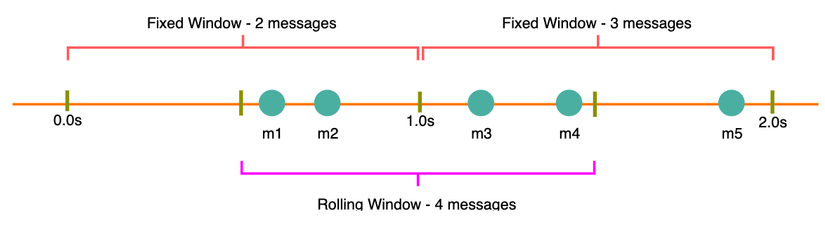
Rate Limiting sẽ đảm nhiệm việc quyết định xem request nào sẽ được gửi đến API Server, request nào sẽ bị block.

Mỗi khi Client gửi request đến Web Server, Web Server sẽ gọi đến Rate Limiter để check xem ngươi dùng này đã vượt quá số Rate Limiting hay chưa? Nếu chưa thì sẽ gửi request đó đến API Server.

### Thuật toán sử dụng trong Rate Limiting

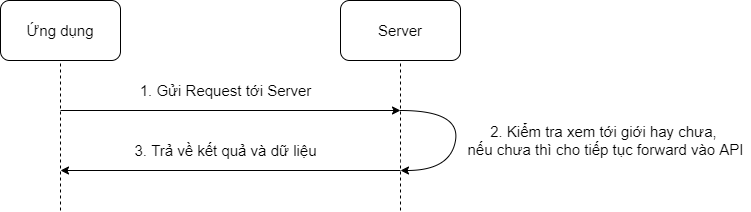
#### Thuật toán Sliding Window (Rolling Window)

Trong thuật toán này thì thời gian sẽ được tính từ phần tử thời gian mà yêu cầu được thực hiện cộng với độ dài của time window.



Hình : Thuật toán Sliding Window

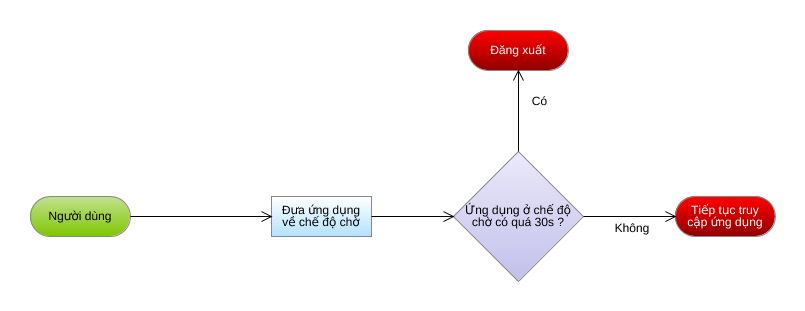
Như trong ví dụ bên trên, giả sử như m1 và m2 được gửi từ giây thứ 0.3s. Khi đó từ 0.3s này sẽ bắt đầu tính. Trong khoảng 0.3s tiếp theo chúng ta thấy có 2 request tiếp theo được gửi đến đó là m3 và m4. Như vậy trong khoảng thời gian 0.6s này đã có 4 request được gửi là m1,m2,m3,m4. Nếu Rate Limiting đang cài đặt là 2 thì khi đó m3,m4 sẽ bị block lại. Giả sử như thời điểm gửi m5 là lúc 1.1s chẳng hạn. Thì khi đó lúc này rate limiting sẽ reset lại thành từ 0. Và lại lặp lại quá trình đó, và m5 sẽ được tính thành 1 lần.



Hình : Cách JWT hoạt động

### Tự động logout 30s khi đang trạng thái chờ

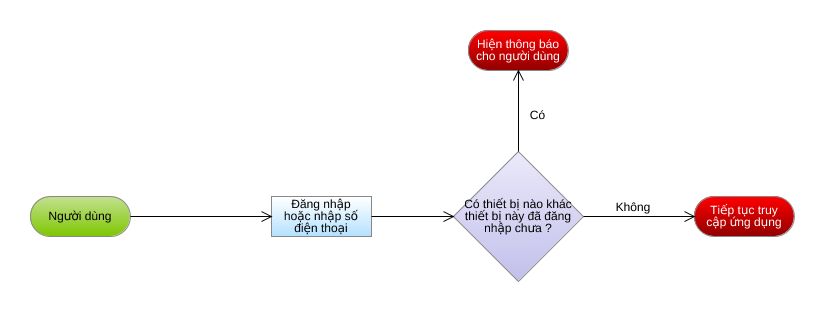
Khi ứng dụng vào chế độ chờ ( chế độ chạy ngầm), ứng dụng sẽ tính thời gian, nếu thời gian chờ vượt quá 30s, ứng dụng sẽ chuyển tới màn hình đăng nhập và bắt người dung đăng nhập lại



Hình : Tự động logout 30s khi thoát ứng dụng

### Chỉ đăng nhập được 1 thiết bị

Người dung chỉ được phép đăng nhập trên 1 thiết bị, khi người dùng dùng 1 thiết bị khác đăng nhập, ứng dụng sẽ hiện lên thiết bị trước đã truy cập tài khoản của mình để người dùng xác thực xem đó có phải là mình hay không. Nếu đó là người dùng, người dùng có thể xoá thiết bị trước đó để đăng nhập trên thiết bị của mình, nếu không người dùng có thể liên lạc với bên hỗ trợ của app.



Hình : Chỉ đăng nhập được 1 thiết bị

### Bcrypt

Bcrypt là một thuật toán băm có thể mở rộng với phần cứng (thông qua số vòng có thể định cấu hình). Sự chậm chạp và nhiều vòng của nó đảm bảo rằng kẻ tấn công phải triển khai số tiền lớn và phần cứng để có thể bẻ khóa mật khẩu của bạn. Thêm vào đó mỗi salt mật khẩu và bạn có thể chắc chắn rằng một cuộc tấn công hầu như không khả thi trừ khi người cố gắng đánh cắp đang sử dụng một máy tính rất mạnh.

Dựa vào những phương pháp trên, một biện pháp được đưa ra như là sự kết hợp của "băm", "salt", "streching". Đó chính là bcrypt. bcrypt sử dụng một thuât toán mã hóa được gọi là "blowfish" . Kết quả về từ hàm băm mật khẩu bao gồm giá trị băm, số lần salt và streching sẽ được set lại thành một giá trị gọi là Modular Crypt Format. Sau đây là kết quả của việc băm một chuỗi ký tự "吾輩は猫である":

$2a$10$4bZqo1S5J5yQk23.n/Vl4OATcXA/e7DnhRaTR7Ico4H5fXPKBcadm

│ │ │ │

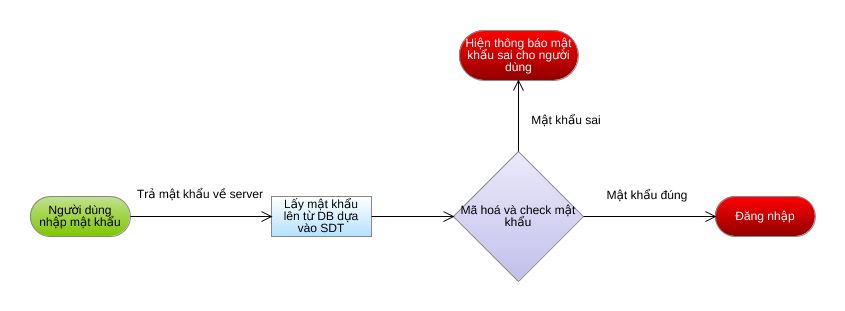
│ │ │ └ ハッシュ値 (184bit; Radix-64 31桁)

│ │ └ salt値 (128bit; Radix-64 22桁)

│ └ コスト (ストレッチング回数 = 2^n; この場合2^10 = 1024回)

└ ハッシュ関数のバージョン (2 = bcrypt; a = 改訂版)

bcrypt trả về một chuỗi duy nhất với tất cả các tham số cần thiết thiết lập cũng như sức mạnh của nó như một chức năng băm mật khẩu. Vì vậy trên phương diện database chỉ cần một cột kiểu chuỗi có thể lưu được giá trị băm này, đây là một điểm lợi rất lớn của bcryt. Ngoài ra, bcrypt hỗ trợ đầu vào của mỗi ký tự ở định dạng UTF-8 như là giá trị băm của chuỗi ký tiếng nhật trong ví dụ trên. Vì vậy nó có thể băm mật khẩu mà không giới hạn ký tự cho các ký tự cả chữ và số...

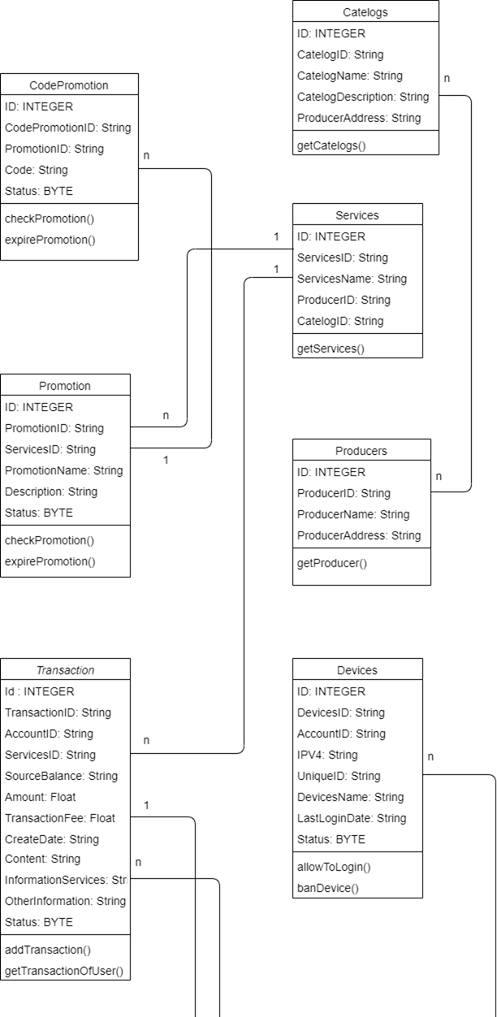


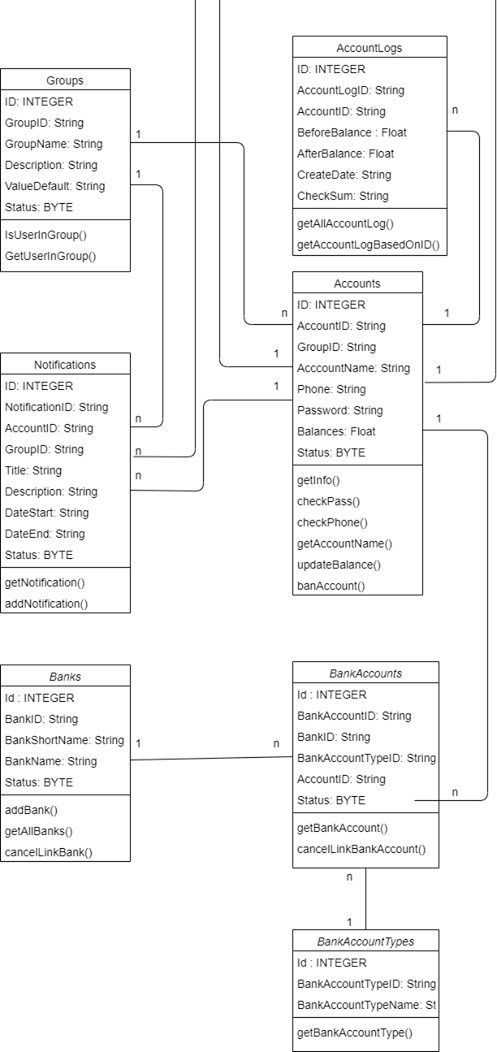
Hình : Bcryt

# YÊU CẦU CHỨC NĂNG

## Lượt đồ tổng quan

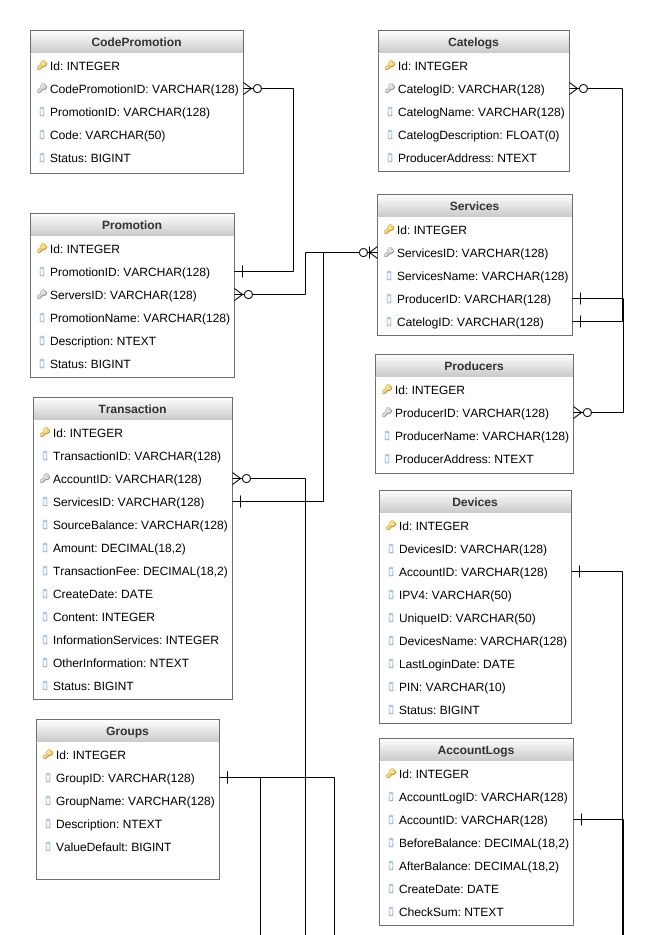
### Class Diagram

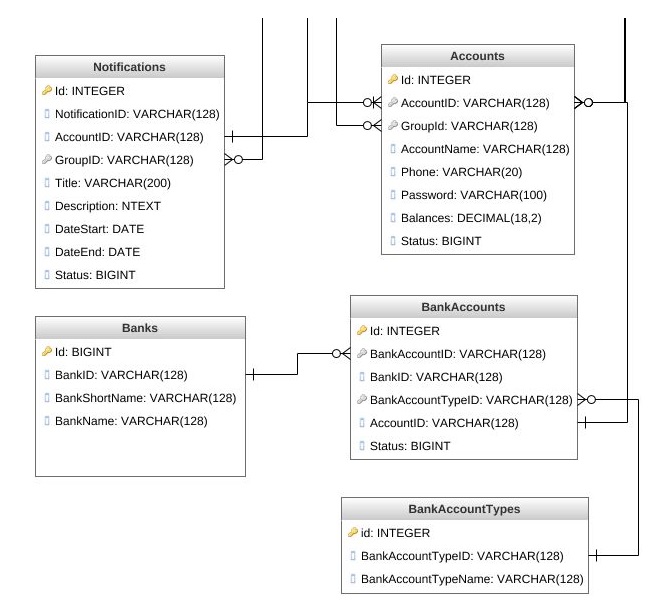




Hình : Lượt đồ Class Diagram

### Database diagram



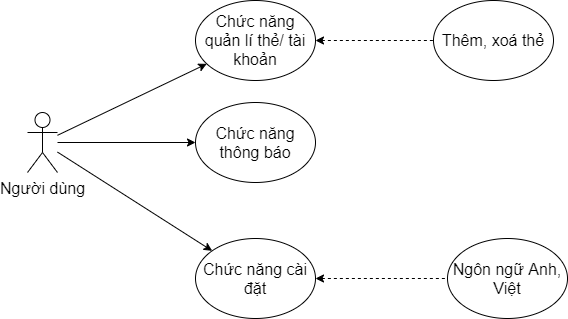


Hình : Database diagram

### Usecase hệ thống

Hình : Usecase hệ thống

## Nhóm chức năng màn hình chính



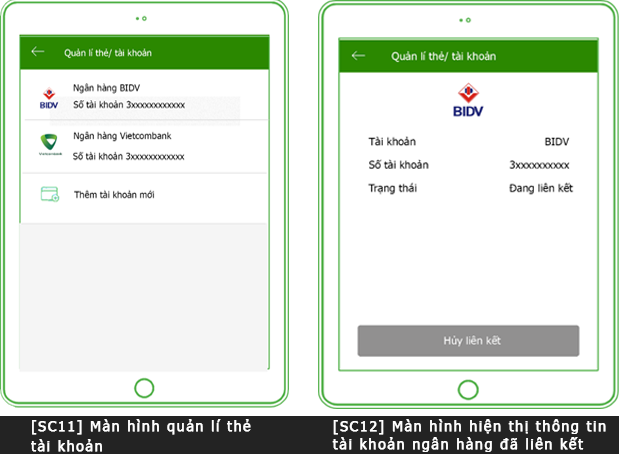
Hình : Lượt đồ Usecase nhóm chức năng màng hình chính

### Chức năng quản lí thẻ/tài khoản

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng quản lí thẻ/ tài khoản |
| Trình trạng trước | Đã đăng nhập |
| Trình trạng sau | N/A |

#### Hình ảnh



### Chức năng thông báo

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng xem thông báo. |
| Trình trạng trước | Đã đăng nhập. |
| Trình trạng sau | N/A |

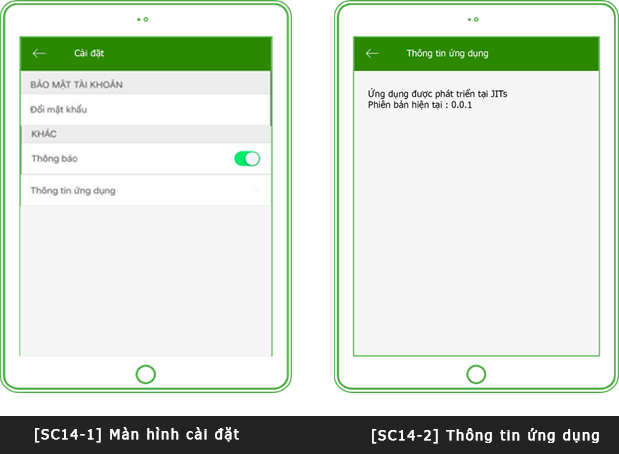
#### Untitled-1Hình ảnh

### Chức năng cài đặt

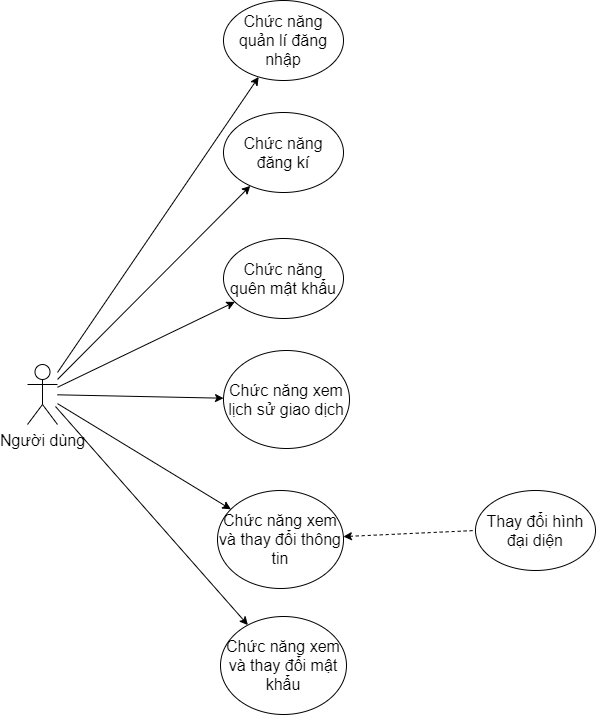
#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng chỉnh sửa một số cài đặt của phần mềm |
| Trình trạng trước | Đã đăng nhập |
| Trình trạng sau | N/A |

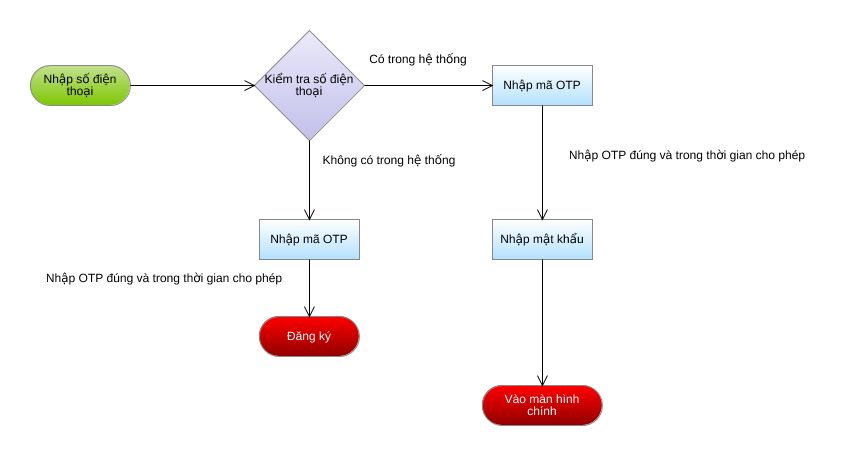
#### Hình ảnh



## Nhóm chức năng tài khoản



Hình : Lượt đồ Usecase nhóm chức năng tài khoản



Hình 39: Nhóm chức năng tài khoản****

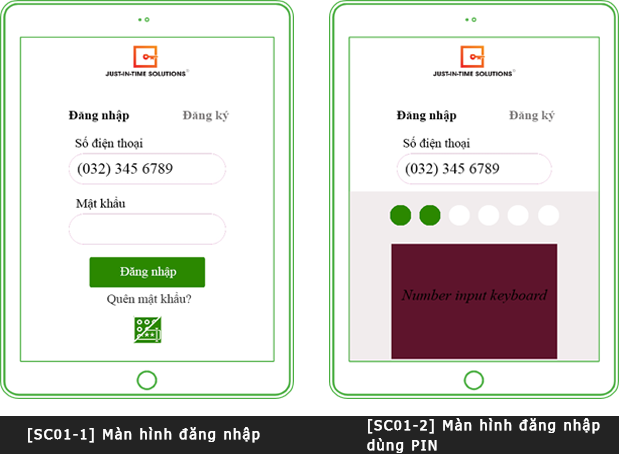
Hình : Chức năng quên mật khẩu

### Chức năng đăng nhập

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng bất kỳ đăng nhập vào hệ thống. |
| Trình trạng trước | N/A |
| Trình trạng sau | Nếu đăng nhập thành công lần đầu, người dùng chuyển đến trang thay đổi mật khẩu. Những lần đăng nhập sau người dùng chuyển đến trang chính. Nếu không thành công, trạng thái của hệ thống không đổi. |

#### Hình ảnh



#### Bắt lỗi nhập dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi nhập | Ví dụ | Thông báo |
| 1 | Số điện thoại chứa ký tự | 0321515aa54 | Số điện thoại phải là số |
| 2 | Số điện thoại rỗng |  | Số điện thoại không được rỗng |
| 3 | Số điện thoại không đủ 10 số | 031212122252225  1201220 | Số điện thoại không đúng dịnh dạng |
| 4 | Kết hợp số điện thoại và mật khẩu không đúng |  | Mật khẩu không đúng |
| 5 | Kết hợp số điện thoại và PIN không đúng |  | PIN không đúng |

### Chức năng đăng ký

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng bất kỳ đăng ký mới một tài khoản. |
| Trình trạng trước | N/A |
| Trình trạng sau | Nếu đăng ký thành công, dữ liệu sẽ được ghi lại và chuyển người dùng đến trang bổ sung thông tin. |

#### Hình ảnh



#### Bắt lỗi nhập dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi nhập | Ví dụ | Thông báo |
| 1 | Số điện thoại chứa ký tự | 0321515aa54 | Số điện thoại phải là số |
| 2 | Số điện thoại rỗng |  | Số điện thoại không được rỗng |
| 3 | Số điện thoại không đủ 10 số | 031212122252225  1201220 | Số điện thoại không đúng dịnh dạng |
| 4 | OTP không dủ 6 số | 1234567  1234 | OTP sai định dạng |
| 5 | OTP không đứng |  | OTP không chính xác |

### Chức năng quên mật khẩu

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng bất kỳ nhập thông tin để khôi phục mật khẩu. |
| Trình trạng trước | N/A |
| Trình trạng sau | Nếu thông tin là đúng, mật khẩu được thay đổi ngẫu nhiên và một tin nhắn chứa mật khẩu được gửi đến người dùng. |

#### Hình ảnh



#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC03] Số điện thoại | TextInput | Nhận số điện thoại người dùng nhập vào, tự động format |
| 2 | [SC03] Gửi OTP | Button | Yêu cầu hệ thống gửi mã OTP vào SĐT trong TextInput SĐT |
| 3 | [SC04-1] Số CMND | TextInput | Nhận số CMND từ người dùng |
| 4 | [SC04-1] Email | TextInput | Nhận email người dùng nhập vào |
| 5 | [SC04-1] Xác nhận | Button | Hiện popup thông báo, nếu thông tin đúng thực hiện tạo lại mật khẩu cho người dùng và gửi bằng tin nhắn đến số điện thoại. |

#### Bắt lỗi nhập dữ liệu

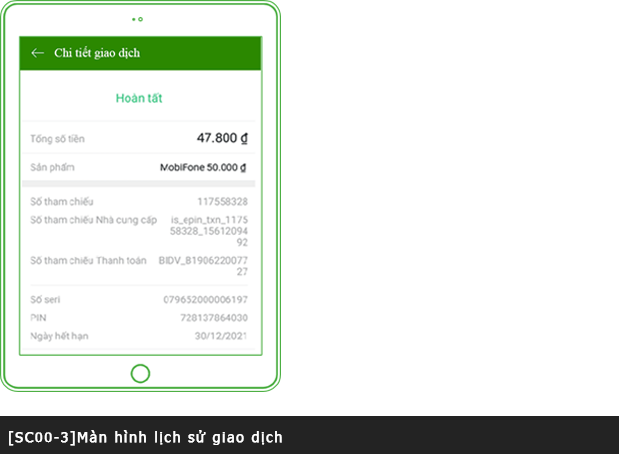
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi nhập | Ví dụ | Thông báo |
| 1 | Số điện thoại chứa ký tự | 0321515aa54 | Số điện thoại phải là số |
| 2 | Số điện thoại rỗng |  | Số điện thoại không được rỗng |
| 3 | Số điện thoại không đủ 10 số | 031212122252225  1201220 | Số điện thoại không đúng dịnh dạng |
| 4 | OTP không dủ 6 số | 1234567  1234 | OTP sai định dạng |
| 5 | OTP không đứng |  | OTP không chính xác |
| 6 | Số chứng minh không đủ 9 hoặc 12 ký tự số | 878985  Âs65656 | Số chứng minh nhân dân không hợp lệ |
| 7 | Email không đúng dịnh dạng | áhaj@  ânsajsn | Email không hợp lệ |
| 8 | Số chứng minh hoặc email không trùng với dữ liệu của người dùng |  |  |

### Chức năng lịch sử giao dịch

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Xem các giao dịch đã thực hiện |
| Trình trạng trước | Người dùng đã đăng nhập thành công vào hệ thống |
| Trình trạng sau | N/A |

#### Hình ảnh



#### Mô tả chi tiết các component

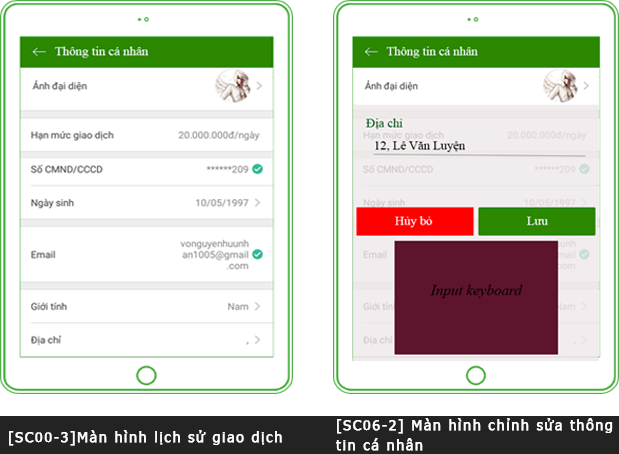
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC00-3]Danh sách các giao dịch | ListItem | Hiện danh sách các giao dich được xấp sếp theo từng tháng. Click chuyển sang [SC05] |
| 2 | [SC05] Thông tin chi tiết các giao dịch | View | Hiện popup thông tin chi tiết các giao dịch |

### Chức năng xem và thay đổi thông tin

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng xem và thay đổi thông tin. |
| Trình trạng trước | Người dùng đã đăng nhập |
| Trình trạng sau | Thông tin người dùng được cập nhật lên hệ thống nếu có thay đổi |

#### Hình ảnh



#### Mô tả chi tiết các component

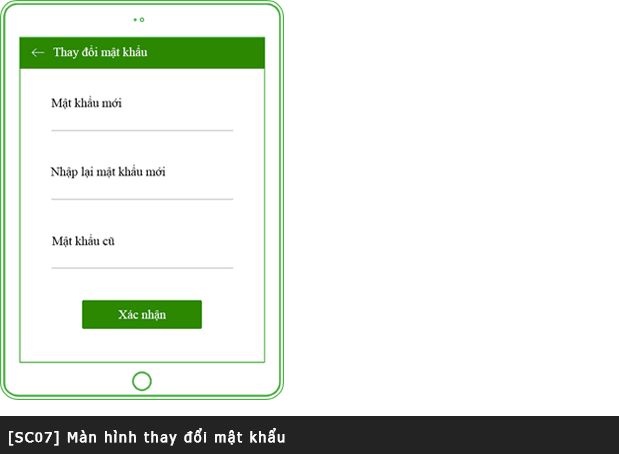
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC06-1] Thông tin người dùng | View | Hiện thông tin người dùng, Click hiện các popup tương ứng [SC06-2] |
| 2 | [SC06-2] Thông tin người dùng đã nhập | TextInput | Hiện thông tin cũ mà người dùng đã nhập. Click vào để thay đổi thông tin |
| 3 | [SC06-2] Lưu | Button | Lưu thông tin và đóng popup |

### Chức năng thay đổi mật khẩu

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng thay đổi mật khẩu |
| Trình trạng trước | Người dùng đã đăng nhập vào hệ thống |
| Trình trạng sau | Lưu lại mật khẩu mới vào hệ thống |

#### Hình ảnh



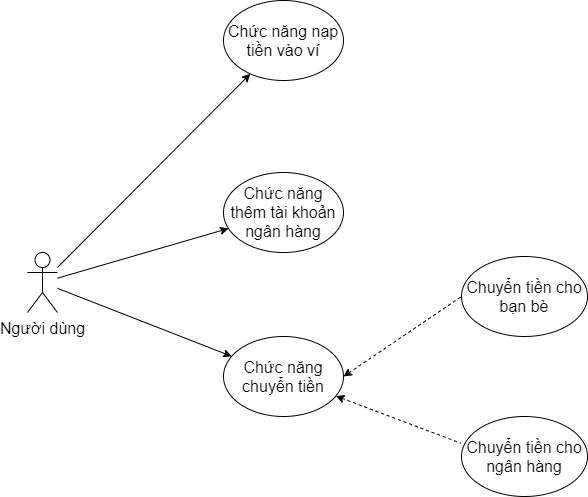
#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC07] Nhập mật khẩu mới | TextInput | Nhập mật khẩu muốn thay đổi |
| 2 | [SC07] Nhập lại mật khẩu | TextInput | Nhập lại mật khẩu mới |
| 3 | [SC07] Mật khẩu cũ | TextInput | Nhập mật khẩu củ |

#### Bắt lỗi nhập dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi nhập | Ví dụ | Thông báo |
| 1 | Mật khẩu mới rỗng |  | Mật khẩu không được rỗng |
| 2 | Mật khẩu mới không bao gồm ký tự thường và hoa và ký số và ký tự đặc biệt không đủ ít nhất 8 ký tự | 12345678  1Aa! | Mật khẩu ít nhất 8 ký tự phải bao gồm ký tự thường, IN HOA, số và ký tự dặc biệt. |
| 3 | Mật khẩu và nhập lại mật khẩu không khớp |  | Mật khẩu nhập lại không khớp |
| 4 | Mật khẩu cũ không đúng |  | Mật khẩu cũ không chính xác |

## Nhóm chức năng nhận và chuyển tiền qua mạng



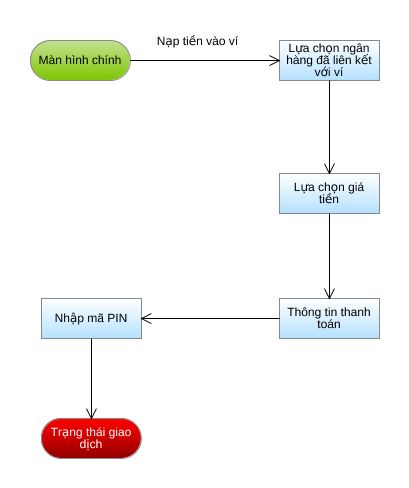
Hình : Usecase nhóm chức năng nhận và chuyển tiền

### Chức năng nạp tiền vào ví

#### Mô tả chung

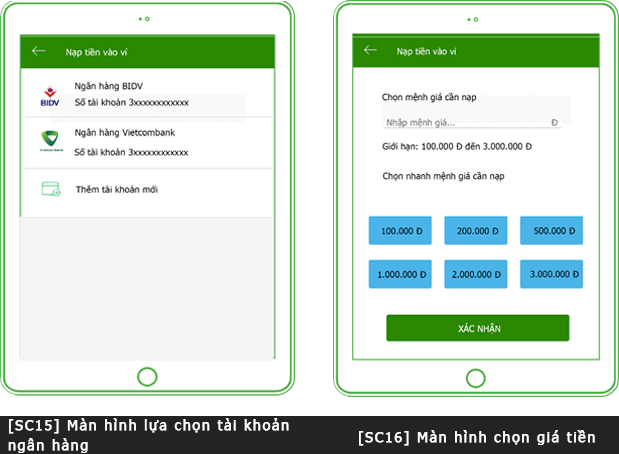
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng nạp tiền vào ví. |
| Trình trạng trước | Đã đăng nhập. |
| Trình trạng sau | Nếu nạp tiền thành công thì số tiền người dùng nạp sẽ từ tài khoản ngân hàng chuyển vào ví, nếu không thì số tiền trong tài khoản ngân hàng và trong ví vẫn giữ nguyên. |

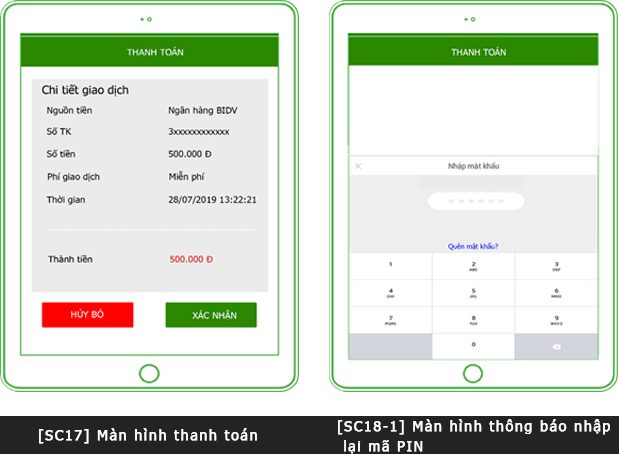
#### Liên kết các màn hình

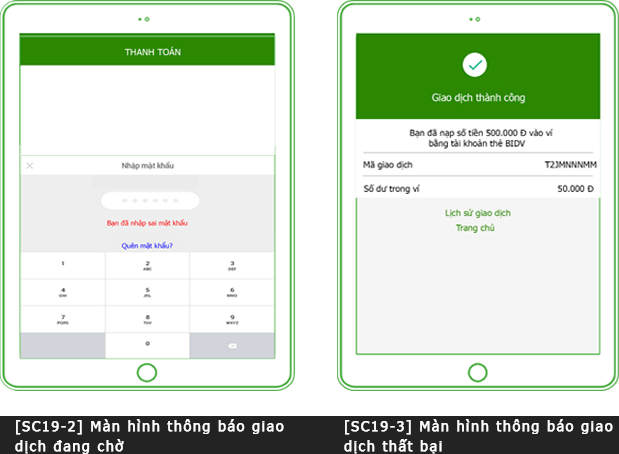


Hình 42: Flow nạp tiền vào ví

#### Hình ảnh







#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC15] Các button của các ngân hàng đã nhập và STK | ListItem | Bấm vào để chọn tài khoản ngân hàng đã có sẵn trong hệ thống, chuyển tới màn hình [SC16] |
| 2 | [SC16] Nhập mệnh giá | Input | Nơi nhập mệnh giá mà người dùng muốn nạp vào ví. |
| 3 | [SC16] Giới hạn | Text | Hiện thị giới hạn mệnh giá tiền, cho người dùng biết giới hạn tiền mà người dùng có thể nhập được. |
| 4 | [SC16] Danh sách các mệnh giá | ButtonGroup | Bấm vào để chọn nhanh mệnh giá |
| 5 | [SC17] Thanh toán | TouchableHighlight | Tiến hành xác nhận và chuyển sang màn hình [SC18] |
| 6 | [SC18] Phím số 1-9 | GroupButton | Dùng để nhập vào và check mã PIN |
| 7 | [SC19] Thông tin giao dịch | Text | Cho biết nội dung giao dịch đó là gì |

#### Bắt lỗi nhập dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | Lỗi không kết nối được với máy chủ. | Không nhận được phản hồi từ máy chủ. | Lỗi không thể kết nối được máy chủ. |
| 2 | Lỗi xác thực với ngân hàng. | Khi nhấn vào tài khoản ngân hàng thì ngân hàng không trả về phản hồi. | Lỗi xác thực với ngân hàng. |
| 3 | Lỗi nhập sai cú pháp. | “chshjacs” | Lỗi nhập kí tự |
| 4 | Lỗi mức tiền nhập vào không nằm trong giới hạn. | “50000” khi thông tin giới hạn là từ 100000 đến 1000000 | Số tiền phải trong mức giới hạn |
| 5 | Lỗi kết nối với ngân hàng. | Khi nhấn vào thanh toán thì khi kết nối với ngân hàng thì ngân hàng không trả về phản hồi. | Chuyển sang [SC19-3] với nội dung giao dịch: lỗi khi giao dịch |
| 6 | Số tiền còn lại trong tài khoản không đủ. | Chuyển 100000Đ nhưng trong thẻ chỉ còn 50000Đ | Chuyển sang [SC19-3] với nội dung giao dịch: Số tiền còn lại trong tài khoản không đủ. |

### Chức năng thêm tài khoản ngân hàng

#### Mô tả chung

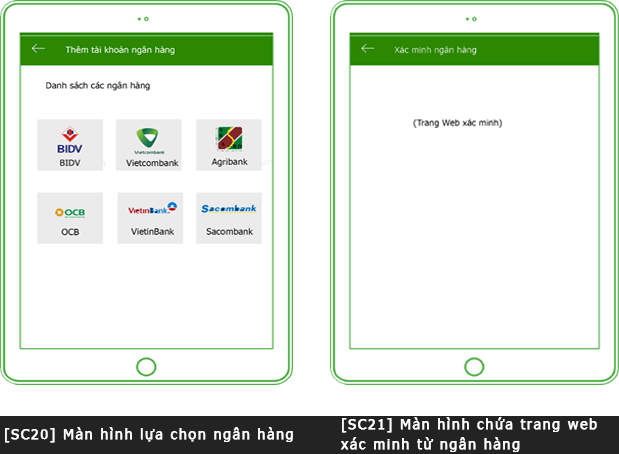
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng thêm tài khoản ngân hàng liên kết |
| Trình trạng trước | Đã đăng nhập. |
| Trình trạng sau | Nếu thêm tài khoản ngân hàng thành công thì sẽ thêm một tài khoản ngân hàng của người dùng trong database |

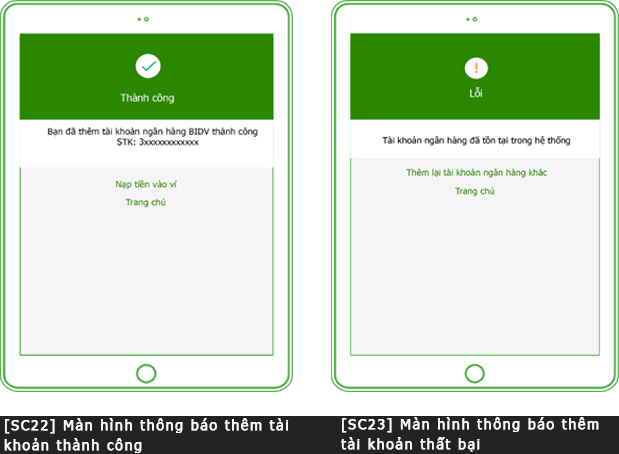
#### Liên kết các màn hình



Hình 43: Flow thêm tài khoản ngân hàng

#### Hình ảnh





#### Mô tả chi tiết các component

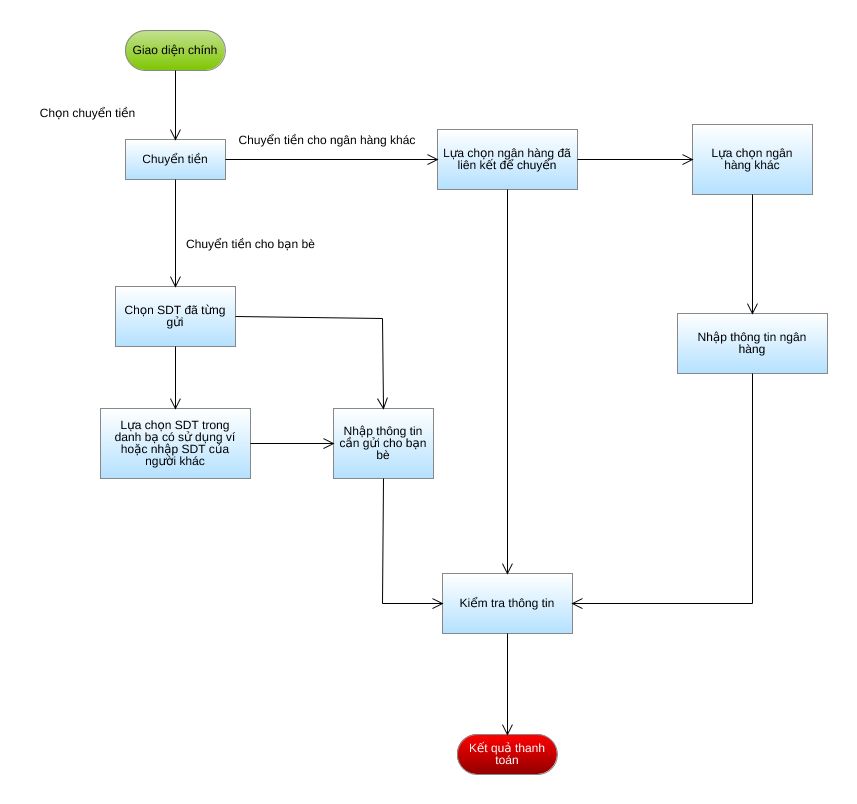
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC20] Nút trở về | Button | Trở về [SC15] |
| 2 | [SC20] Danh sách các ngân hàng mà phần mềm liên kết | ListItem | Danh sách các ngân hàng mà phần mềm liên kết, chuyển sang [SC21] |
| 3 | [SC22] Nạp tiền vào ví | TouchableHightlight | Trang xác minh |
| 4 | [SC22] Trang chủ | TouchableHightlight | Trở về [SC00-1] |
| 5 | [SC23] Thêm lại tài khoản ngân hàng khác | TouchableHightlight | Trở về [SC15] |
| 6 | [SC23] Trang chủ | TouchableHightlight | Trở về [SC00-1] |

### Chức năng chuyển tiền

#### Mô tả chung

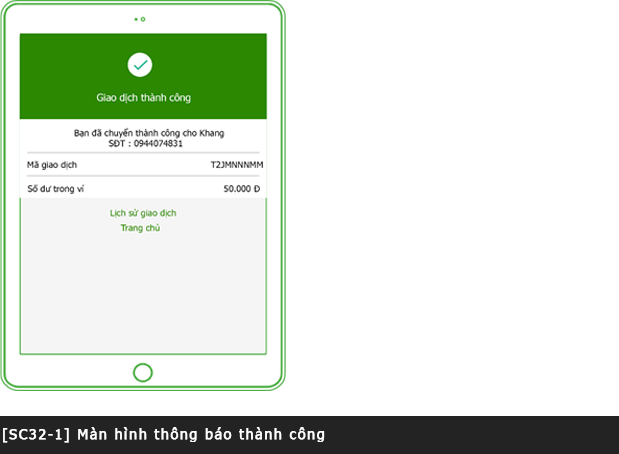
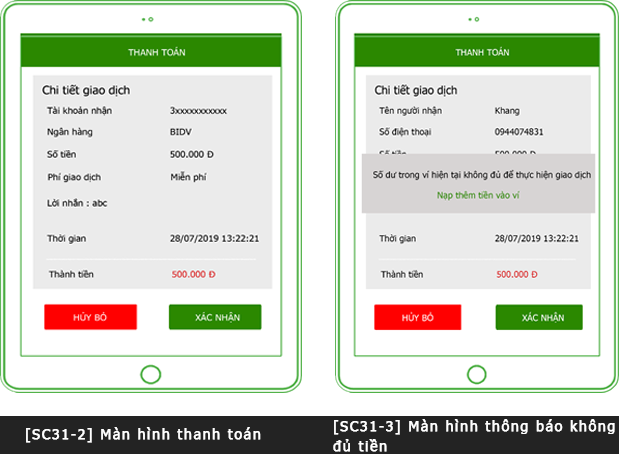
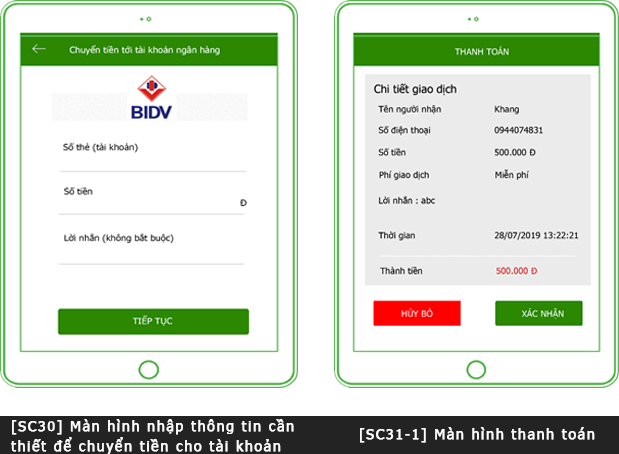
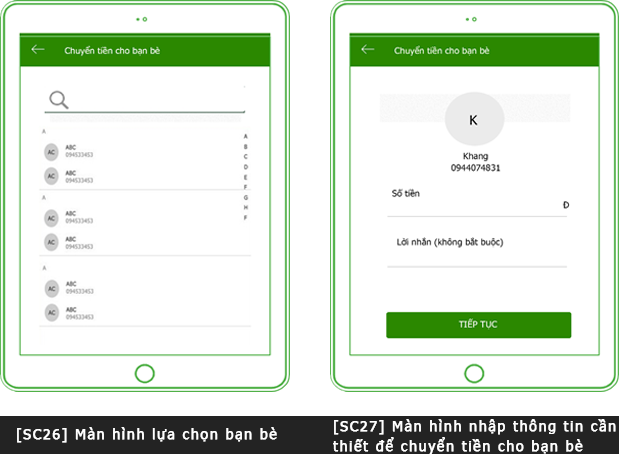
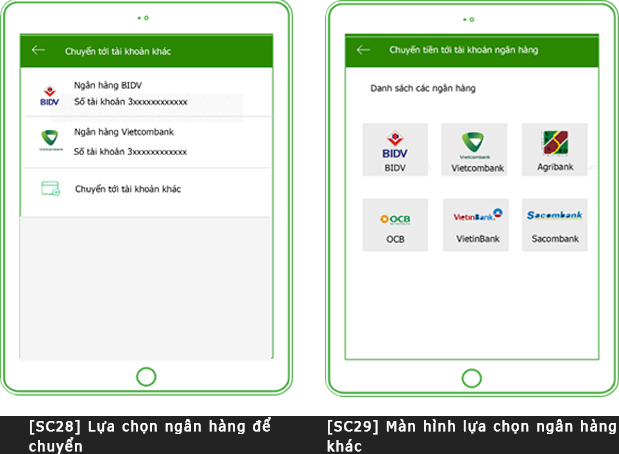
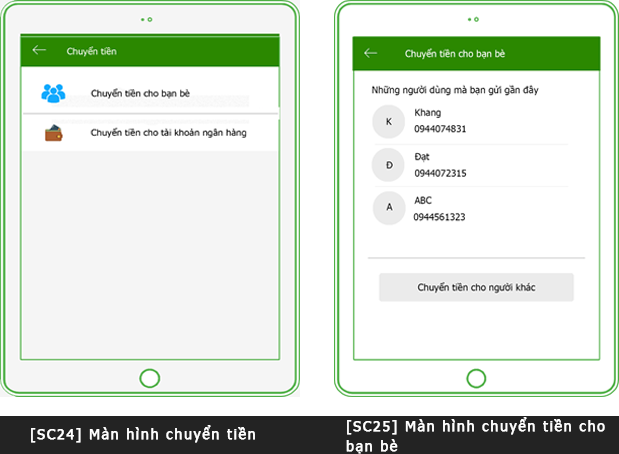
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng chuyển tiền từ ví qua tài khoản khác hoặ qua ngân hàng khác |
| Trình trạng trước | Đã đăng nhập. |
| Trình trạng sau | Nếu chuyển tiền thành công thì số tiền người dùng từ ví sẽ chuyển tới tài khoản ngân hàng hoặc ví tương ứng |

#### Liên kết các màn hình



Hình 44: Flow chuyển tiền

#### Hình ảnh



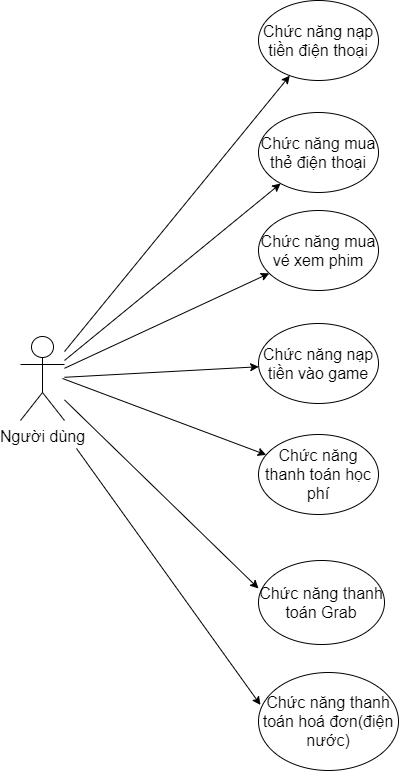
#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Màn hình chuyển trang |
| 1 | [SC25] Danh sách các người dùng đã được chuyển | ListItem | Chuyển sang [SC27] |
| 2 | [SC26] Danh sách bạn bè có tham gia ví | ListItem | Chuyển sang [SC27] |
| 3 | [SC27] Số tiền | Input | Số tiền người dùng cần chuyển |
| 4 | [SC27] Lời nhắn | Input | Lời nhắn của người dùng cần chuyển |
| 5 | [SC30] Số thẻ | Input | Số thẻ ngân hàng cần gửi |

#### Bắt lỗi nhập dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | Danh sách các số điện thoại trùng với SDT trong danh bạ | Không thể kết nối tới máy chủ. | Không thể kết nối tới máy chủ. |
| 2 | Số tiền lỗi nhập dữ liệu | “nnnkk” | Lỗi nhập kí tự. |
| 3 | Danh sách các ngân hàng liên kết lỗi kết nối với máy chủ. | Không thể kết nối tới máy chủ. | Không thể kết nối tới máy chủ. |
| 4 | Thanh toán không đủ số tiền. | Chuyển 100000Đ nhưng trong thẻ chỉ còn 50000Đ | Số dư trong tài khoản không đủ. |
| 5 | Thanh toán lỗi kết nối với ngân hàng. | Khi nhấn vào thanh toán thì khi kết nối với ngân hàng thì ngân hàng không trả về phản hồi. | Lỗi khi giao dịch. |

## Nhóm chức năng thanh toán trực tuyến



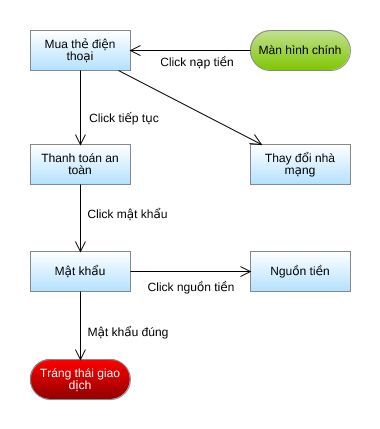
Hình : Usecase nhóm chức năng thanh toàn trực tiếp

### Chức năng nạp tiền điện thoại

#### Mô tả chung

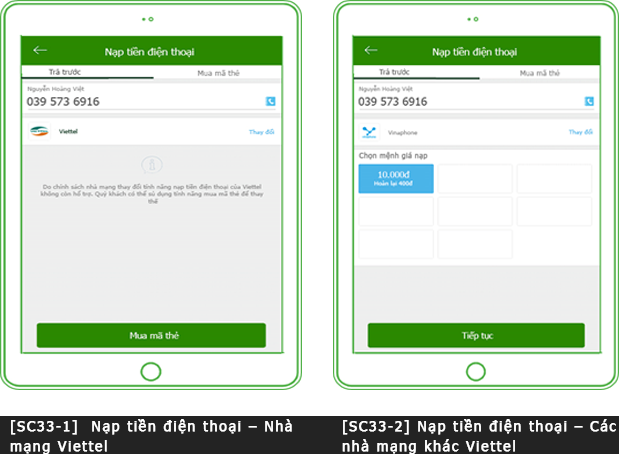
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng nạp tiền điện thoại |
| Trình trạng trước | Đã đăng nhập vào hệ thống, tài khoản trong ví/ngân hàng còn đủ tiền thanh toán |
| Trình trạng sau | Thông báo trình trạng giao dịch (thành công/thất bại) cập nhật lịch sử giao dịch |

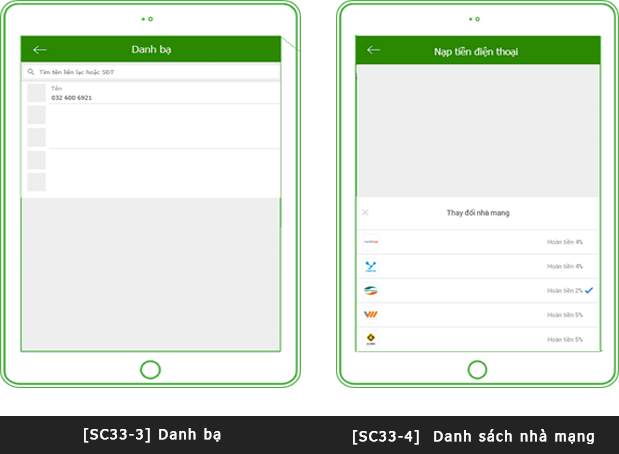
#### Liên kết các màn hình

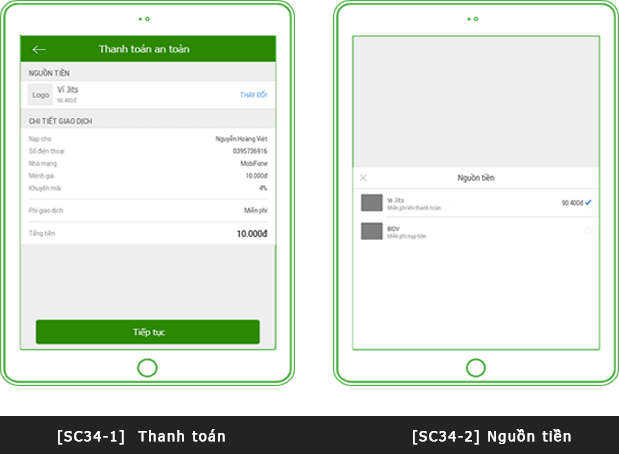


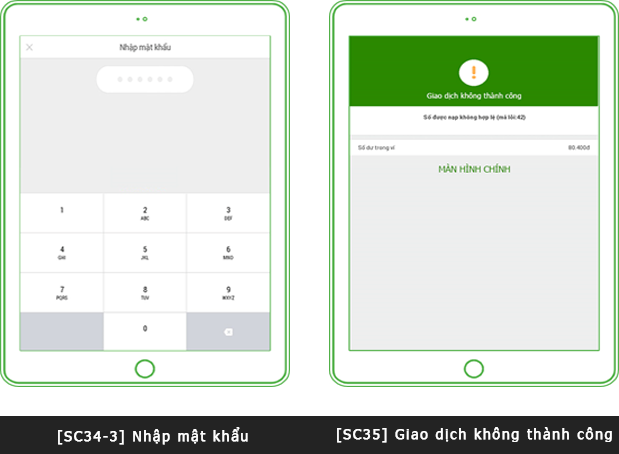
Hình 46: Flow liên kết chức năng nạp tiền điện thoại

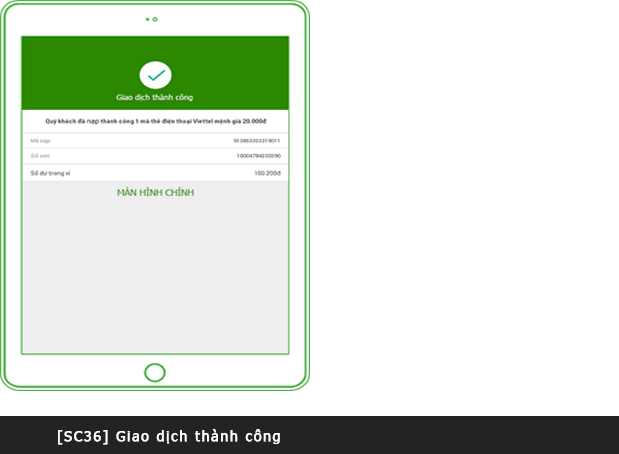
#### Hình ảnh











#### Dòng sự kiện click chuyển trang

|  |  |  |
| --- | --- | --- |
| STT | Component | Màn hình chuyển trang |
| 1 | [SC33-1] [SC33-2] Nút trở về | [[SC00-1]](#_Hình_ảnh) |
| 2 | [SC33-1] Mua mã thẻ | [[SC37-1]](#_Liên_kết_các_6) |
| 3 | [SC33-1] [SC33-2] SĐT | [[SC33-3]](#_Liên_kết_các_6) |
| 4 | [SC33-1] [SC33-2] Item nhà mạng | [[SC33-4]](#_Liên_kết_các_6) |
| 5 | [SC33-2] Tiếp tục | [[SC34-1]](#_Liên_kết_các_6) |
| 6 | [SC33-3] Item SĐT | [[SC33-1]](#_Liên_kết_các_6) |
| 7 | [SC33-4] Item nhà mạng | [[SC33-1]](#_Liên_kết_các_6) |
| 8 | [SC34-1] Item nguồn tiền | [[SC34-2]](#_Liên_kết_các_6) |
| 9 | [SC34-1] Tiếp tục | [[SC34-3]](#_Liên_kết_các_6) |
| 10 | [SC34-3] Nhập đúng mật khẩu, đủ tiền | [[SC36]](#_Liên_kết_các_6) |
| 11 | [SC34-3] Nhập đúng mật khẩu, thiếu tiền | [[SC35]](#_Liên_kết_các_6) |
| 12 | [SC36] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |
| 13 | [SC35] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |

#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC33-1] Trả trước | Button | Giao diện nạp tiền điện thoại |
| 2 | [SC33-1] Mua mã thẻ | Button | Giao diện mua mã thẻ. Click hiện [SC37-1] |
| 3 | [SC33-1] SĐT | TextInput | Click nhập SĐT. Click hiện [SC33-3] |
| 4 | [SC33-1] Item nhà mạng | View | Hiện nhà mạng đang sử dụng, nếu là nhà mạng Viettel thì không thực hiện được chức năng nạp tiền điện thoại, các nhà mạng khác thì được. Click hiện [SC33-4] |
| 5 | [SC33-1] Mua mã thẻ | Button | Click để mua thẻ. Click hiện [SC37-1] |
| 6 | [SC33-2] Danh sách mệnh giá | ListItem | Danh sách các mệnh giá nạp thẻ. Click để chọn mệnh giá cần nạp |
| 7 | [SC33-3] Danh sách các SĐT trong danh bạ | ListItem | Hiện danh sách các SĐT có trong danh bạ. Click trở về [SC33-1] |
| 8 | [SC33-4] Danh sách nhà mạng | ListItem | Hiện danh sách nhà mạng. Click chuyển về [SC33-1] |
| 9 | [SC34-1] Nguồn tiền | View | Click hiện danh sách các nguồn tiền [SC34-2] |
| 10 | [SC34-2] Danh sách các nguồn tiền | ListItem | Hiện danh sách các nguồn tiền. Click Item chuyển về [SC34-1] |
| 11 | [SC34-3] Nhập mật khẩu | TextInput | Nhập mật khẩu. Nếu đúng và tài khoản còn đủ tiền thanh toán chuyển về [SC36], nếu sai SĐT hoặc tài khoản không đủ tiền chuyển về [SC35] và thông báo lỗi |

#### Bắt lỗi nhập dữ liệu

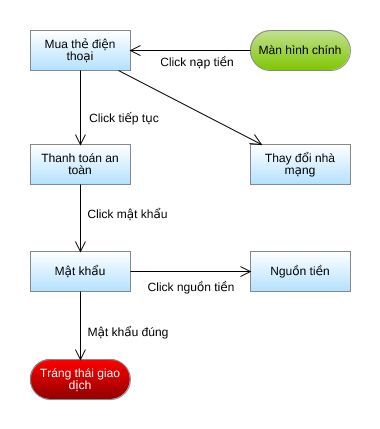
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | SĐT bỏ trống | “” | SĐT không được để trống |
| 2 | SĐT sai với nhà mạng đã chọn | Nhập SĐT Viettel mà chọn nhà mạng ViN/Aphone | Số điện thoại không đúng với nhà mạng đã chọn |
| 3 | SĐT không tồn tại, thiếu số, dư số | 032 666 6666,  032 555 XXX,  032 777 XXXXX | SĐT không hợp lệ |
| 4 | Tài khoản ví hoặc Ngân hàng liên kết không đủ tiền | Nạp thẻ 100k mà trong ví còn 20k | Không đủ tiền để thực hiện giao dịch xin vui lòng nạp thêm |
| 5 | Mật khẩu nhập sai | MK: 123321  MK nhập: 123123 | Sai mật khẩu xin vui lòng nhập lại |

### Chức năng mua thẻ điện thoại

#### Mô tả chung

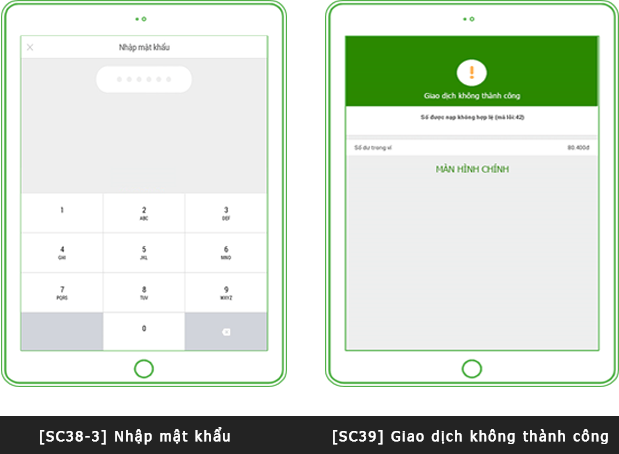
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng mua thẻ điện thoại |
| Trình trạng trước | Đã đăng nhập vào hệ thống, tài khoản trong ví/ngân hàng còn đủ tiền thanh toán |
| Trình trạng sau | Thông báo trình trạng giao dịch (thành công/thất bại) cập nhật lịch sử giao dịch |

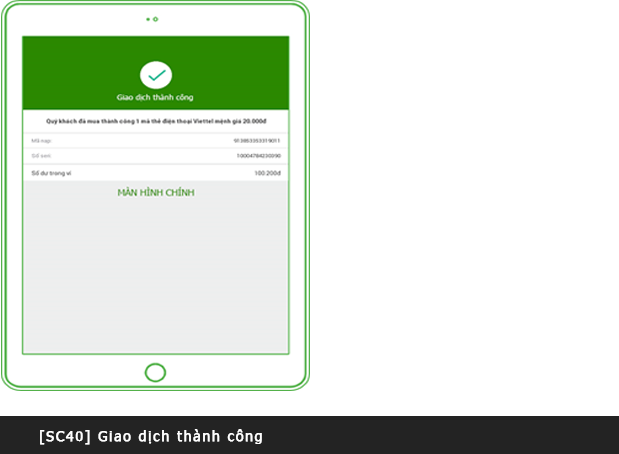
#### Liên kết các màn hình



Hình 47: Lượt đồ chức năng mua thẻ điện thoại

#### Hình ảnh12





#### Dòng sự kiện click chuyển trang

|  |  |  |
| --- | --- | --- |
| STT | Component | Màn hình chuyển trang |
| 1 | [SC37-1] Nút trở về | [[SC00-1]](#_Hình_ảnh) |
| 2 | [SC37-1] Trả trước | [[SC33-1]](#_Liên_kết_các_5) |
| 3 | [SC37-1] SĐT | [[SC37-2]](#_Liên_kết_các_5) |
| 4 | [SC37-1] Tiếp tục | [[SC38-1]](#_Liên_kết_các_5) |
| 5 | [SC38-1] Item nhà mạng | [[SC38-2]](#_Liên_kết_các_5) |
| 6 | [SC38-1] Tiếp tục | [[SC38-3]](#_Liên_kết_các_5) |
| 7 | [SC38-2] Chọn nguồn tiền | [[SC38-1]](#_Liên_kết_các_5) |
| 8 | [SC38-3] Nhập sai mật khẩu, đủ tiền | [[SC40]](#_Liên_kết_các_5) |
| 9 | [SC38-3] Nhập đúng mật khẩu, thiếu tiền | [[SC39]](#_Liên_kết_các_5) |

#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC37-1] Nút trở về | Button | Trở về [SC00-1] |
| 2 | [SC37-1] Trả trước | Button | Giao diện nạp tiền điện thoại. Click hiện [SC33-1] |
| 3 | [SC37-1] Mua mã thẻ | Button | Giao diện mua mã thẻ |
| 4 | [SC37-1] Item nhà mạng | View | Click hiện [SC37-2] |
| 5 | [SC37-1] Tiếp tục | Button | Click để mua thẻ. Click hiện [SC38-1] |
| 6 | [SC37-2] Danh sách mệnh giá | ListItem | Danh sách các mệnh giá nạp thẻ. Click để chọn mệnh giá cần nạp. Click item chuyển sang [SC37-1] |
| 7 | [SC38-1] Nguồn tiền | View | Click hiện danh sách các nguồn tiền [SC38-2] |
| 8 | [SC38-1] Tiếp tục | Button | Click chuyển qua [SC38-3] |
| 9 | [SC38-2] Danh sách các nguồn tiền | ListItem | Hiện danh sách các nguồn tiền. Click Item chuyển về [SC38-1] |
| 10 | [SC38-3] Nhập mật khẩu | TextInput | Nhập mật khẩu. Nếu đúng và tài khoản còn đủ tiền thanh toán chuyển về [SC40], nếu sai hoặc tài khoản không đủ tiền chuyển về [SC39] |
| 11 | [SC39] Màn hình chính | Button | Click trở về [SC00-1] |

#### Bắt lỗi nhập dữ liệu

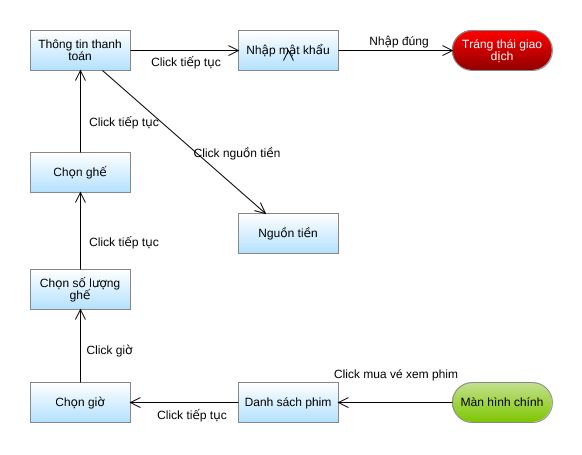
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | SĐT bỏ trống | “” | SĐT không được để trống |
| 2 | SĐT sai với nhà mạng đã chọn | Nhập SĐT Viettel mà chọn nhà mạng ViN/Aphone | Số điện thoại không đúng với nhà mạng đã chọn |
| 3 | SĐT không tồn tại, thiếu số, dư số | 032 666 6666,  032 555 XXX,  032 777 XXXXX | SĐT không hợp lệ |
| 4 | Tài khoản ví hoặc Ngân hàng liên kết không đủ tiền | Nạp thẻ 100k mà trong ví còn 20k | Không đủ tiền để thực hiện giao dịch xin vui lòng nạp thêm |
| 5 | Mật khẩu nhập sai | MK: 123321  MK nhập: 123123 | Sai mật khẩu xin vui lòng nhập lại |

### Chức năng mua vé xem phim

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng mua vé xem phim |
| Trình trạng trước | Đã đăng nhập vào hệ thống, tài khoản trong ví/ngân hàng còn đủ tiền thanh toán |
| Trình trạng sau | Thông báo trình trạng giao dịch (thành công/thất bại) cập nhật lịch sử giao dịch |

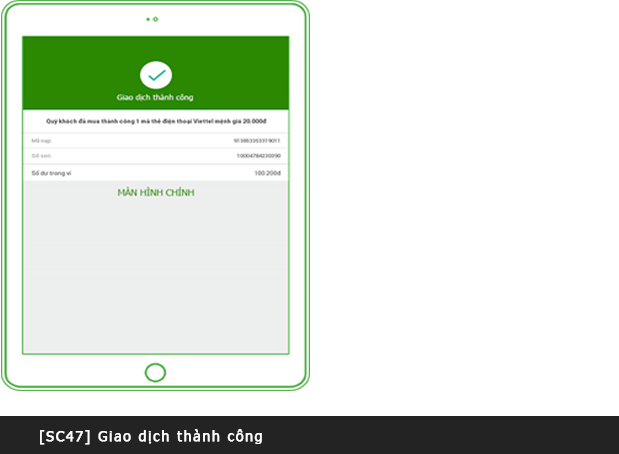
#### Liên kết các màn hình



Hình 48: Flow chức năng mua vé xem phim

#### Hình ảnh4143





#### Dòng sự kiện click chuyển trang

|  |  |  |
| --- | --- | --- |
| STT | Component | Màn hình chuyển trang |
| 1 | [SC41] Nút trở về | [[SC00-1]](#_Hình_ảnh) |
| 2 | [SC41] Chọn Item phim | [[SC42]](#_Liên_kết_các_4) |
| 3 | [SC42] Chọn giờ xem | [[SC43]](#_Liên_kết_các_4) |
| 4 | [SC43] Thêm số lượng ghế và nhấn tiếp tục | [[SC44]](#_Liên_kết_các_4) |
| 5 | [SC44] Chọn ghế và nhấn tiếp tục | [[SC45-1]](#_Liên_kết_các_4) |
| 6 | [SC45-1] Tiếp tục | [[SC45-3]](#_Liên_kết_các_4) |
| 7 | [SC45-2] Chọn nguồn tiền thanh toán | [[SC45-1]](#_Liên_kết_các_4) |
| 8 | [SC45-3] Nhập sai mật khẩu, thiếu tiền | [[SC46]](#_Liên_kết_các_4) |
| 9 | [SC45-3] Nhập đúng mật khẩu, thiếu tiền | [[SC47]](#_Liên_kết_các_4) |
| 10 | [SC46] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |
| 11 | [SC47] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |

#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC41] Nút trở về | Button | Trở về [SC00-1] |
| 2 | [SC41] Chọn Item phim | View | Click phim tương ứng muốn mua vé. Click chuyển sang [SC42] |
| 3 | [SC42] Thứ | Text | Các ngày mua vé trong tuần |
| 4 | [SC42] Rạp phim | Text | Tên và địa chỉ rạp chiếu phim |
| 5 | [SC43] Thông tin phim, giá tiền | View | Thông tin phim |
| 6 | [SC43] Tiếp tục | Button | Nếu đã tăng số lượng thì chuyển đến giao diện chọn ghế [SC44]. Nếu chưa tăng số lượng không cho click |
| 7 | [SC43] Thêm số lượng ghế | Icon | Click tăng số lượng |
| 8 | [SC44] Chọn ghế | ListItem | Danh sách các ghế với các màu khác nhau. Màu cam: ghế vip, màu xám đơn: ghế thường, màu xám cặp: ghế đôi, màu xanh: ghế đã chọn |
| 9 | [SC44] Tiếp tục | Button | Nếu đã chọn ghế click chuyển sang [SC44] |
| 10 | [SC45-1] Nguồn tiền | View | Chọn nguồn tiền thanh toán. Click chuyển sang [SC45-2] |
| 11 | [SC45-1] Tiếp tục | Button | Danh sách các mệnh giá nạp thẻ. Click để chọn mệnh giá cần nạp. Click item chuyển sang [SC37-1] |
| 12 | [SC45-1] Thông tin chung | View | Thông tin vé, tiền, người mua, … |
| 13 | [SC45-2] Chọn nguồn tiền thanh toán | View | Click chọn nguồn tiền chuyển về [SC45-1] |
| 14 | [SC45-2] Nhập sai mật khẩu, thiếu tiền | Button | Click chuyển qua [SC38-3] |
| 15 | [SC45-2] Nhập đúng mật khẩu, thiếu tiền | ListItem | Hiện danh sách các nguồn tiền. Click Item chuyển về [SC38-1] |
| 16 | [SC45-3] Nhập mật khẩu | TextInput | Nhập mật khẩu. Nếu đúng và tài khoản còn đủ tiền thanh toán chuyển về [SC47], nếu sai hoặc tài khoản không đủ tiền chuyển về [SC46] |
| 17 | [SC46] Màn hình chính | Text | Click trở về [SC00-1] |
| 18 | [SC47] Màn hình chính | Text | Click trở về [SC00-1] |

#### Bắt lỗi nhập dữ liệu

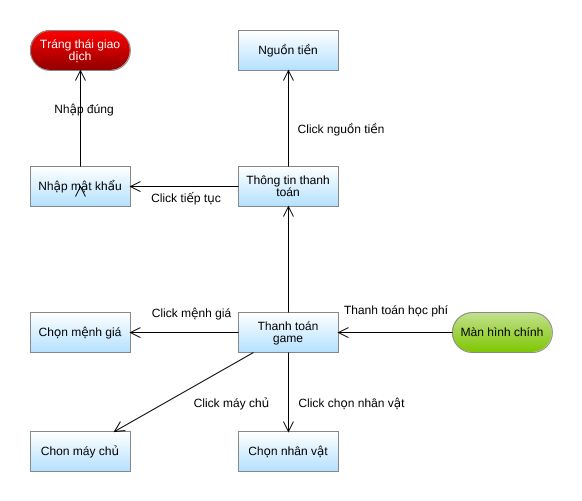
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | Tài khoản không đủ tiền | Thanh toán vé 100k mà TK chỉ còn 20k | Tài khoản của bạn không đủ để thực hiện giao dịch, xin vui lòng nạp thêm. |

### Chức năng nạp tiền vào game

#### Mô tả chung

|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng nạp tiền vào game |
| Trình trạng trước | Đã đăng nhập vào hệ thống, tài khoản trong ví/ngân hàng còn đủ tiền thanh toán |
| Trình trạng sau | Thông báo trình trạng giao dịch (thành công/thất bại) cập nhật lịch sử giao dịch |

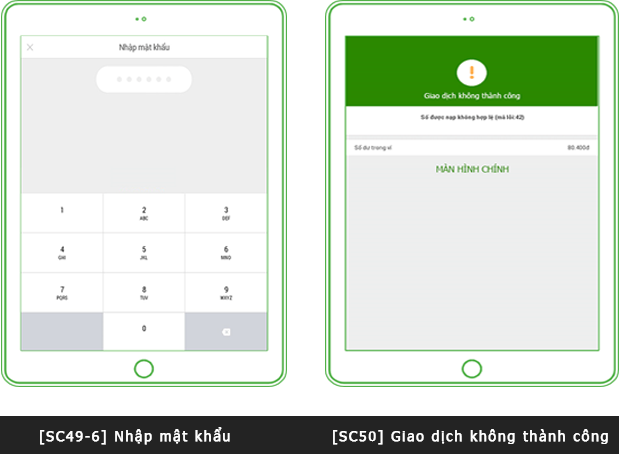
#### Liên kết các màn hình

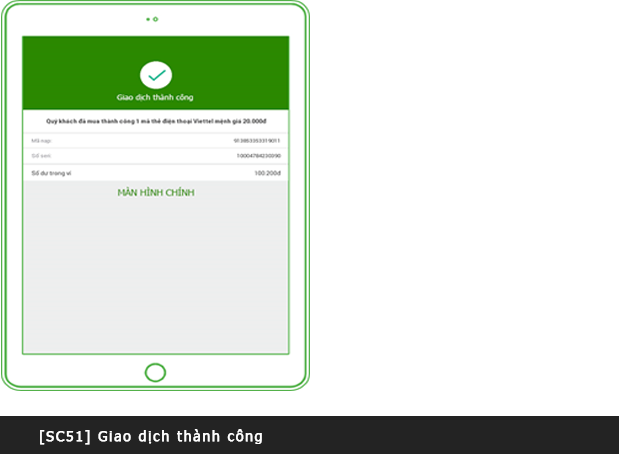


Hình 49:Flow chức năng nạp tiền vào game

#### Hình ảnh4849







#### Dòng sự kiện click chuyển trang

|  |  |  |
| --- | --- | --- |
| STT | Component | Màn hình chuyển trang |
| 1 | [SC48] Nút trở về | [[SC00-1]](#_Hình_ảnh) |
| 2 | [SC48] Chọn Item game | [[SC49-1]](#_Liên_kết_các_3) |
| 3 | [SC49-1] Chọn nhân vật | [[SC49-4]](#_Liên_kết_các_3) |
| 4 | [SC49-1] Chọn máy chủ | [[SC49-3]](#_Liên_kết_các_3) |
| 5 | [SC49-1] Chọn mệnh giá | [[SC49-2]](#_Liên_kết_các_3) |
| 6 | [SC49-1] Chọn nguồn tiền | [[SC49-5]](#_Liên_kết_các_3) |
| 7 | [SC49-1] Chọn xác nhận | [[SC49-6]](#_Liên_kết_các_3) |
| 8 | [SC49-2] Chọn item mệnh giá | [[SC49-1]](#_Liên_kết_các_3) |
| 9 | [SC49-3] Chọn item server | [[SC49-1]](#_Liên_kết_các_3) |
| 10 | [SC49-4] Chọn item nhân vật | [[SC49-1]](#_Liên_kết_các_3) |
| 11 | [SC49-2] Chọn item nguồn tiền | [[SC49-2]](#_Liên_kết_các_3) |
| 12 | [SC49-6] Mật khẩu đúng, đủ tiền thanh toán | [[SC49-6]](#_Liên_kết_các_3) |
| 13 | [SC49-6] Mật khẩu đúng, không đủ tiền thanh toán | [[SC49-6]](#_Liên_kết_các_3) |
| 14 | [SC50] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |
| 15 | [SC49] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |

#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC48] Nút trở về | Button | Trở về [SC00-1] |
| 2 | [SC48] Chọn Item game | ListItem | Danh sách các game, gồm có danh sách game nổi bật và danh sách tất cả các game. Click chuyển sang [SC49-1] |
| 3 | [SC49-1] Chọn nhân vật | View | Chọn nhanh vật cần nạp tiền. Click chuyển sang [SC49-4] |
| 4 | [SC49-1] Chọn máy chủ | View | Chọn máy chủ cần nạp tiền. Click chuyển sang [SC49-1] |
| 5 | [SC49-1] Chọn mệnh giá | View | Chọn mệnh giá cần nạp tiền. Click chuyển sang [SC49-2] |
| 6 | [SC49-1] Chọn nguồn tiền | View | Chọn nguồn tiền cần thanh toán. Click chuyển sang [SC49-5] |
| 7 | [SC49-1] Chọn xác nhận | Button | Click để thanh toán chuyển sang [SC49-6] nếu đầy đủ thông tin. Thông tin chưa đầy đủ không cho chuyển. |
| 8 | [SC49-2] Chọn item mệnh giá | ListItem | Chọn mệnh giá cần thanh toán. Click chuyển sang [SC49-1] |
| 9 | [SC49-3] Chọn item server | Button | Chọn server cần thanh toán. Click chuyển sang [SC49-1] |
| 10 | [SC49-4] Chọn item nhân vật | View | Chọn nhân vật cần thanh toán. Click chuyển sang [SC49-1] |
| 11 | [SC49-2] Chọn item nguồn tiền | Button | Chọn nguồn tiền cần thanh toán. Click chuyển sang [SC49-1] |
| 12 | [SC49-6] Nhập mật khẩu | TextInput | Nếu mật khẩu đúng, đủ tiền thanh toán chuyển đến [SC51]. Nếu mật khẩu đúng không đủ tiền [SC50]. Nếu sai mật khẩu không cho chuyển. Lỗi khác chuyển về [SC50] |
| 14 | [SC50] Màn hình chính | Text | Click chuyển qua [SC38-3] |
| 15 | [SC49] Màn hình chính | Text | Hiện danh sách các nguồn tiền. Click Item chuyển về [SC38-1] |

#### Bắt lỗi nhập dữ liệu

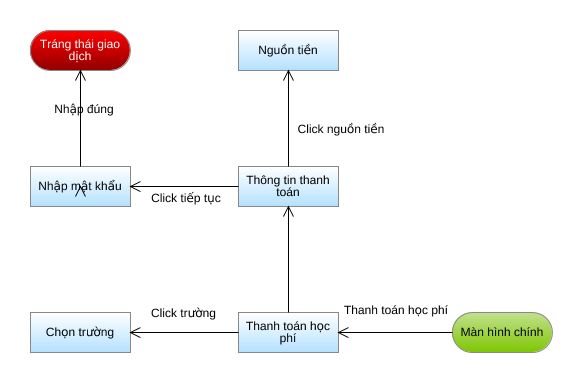
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | Tài khoản bỏ trống | “” | Tài khoản không được để trống |
| 2 | Tài khoản không tồn tại | TK: xxx123xxx không có trong hệ thống | Tài khoản không tồn tại trong hệ thống |
| 3 | Tài khoản không đủ 6 ký tự | Nhập TK: xxx | Tài khoản không đủ 6 ký tự |
| 4 | Ví/Tài khoản liên kết không đủ tiền thanh toán | Thanh toán hóa đơn 100k mà TK chỉ còn 20k | Tài khoản của bạn không đủ để thực hiện giao dịch, xin vui lòng nạp thêm |

### Chức năng thanh toán học phí

#### Mô tả chung

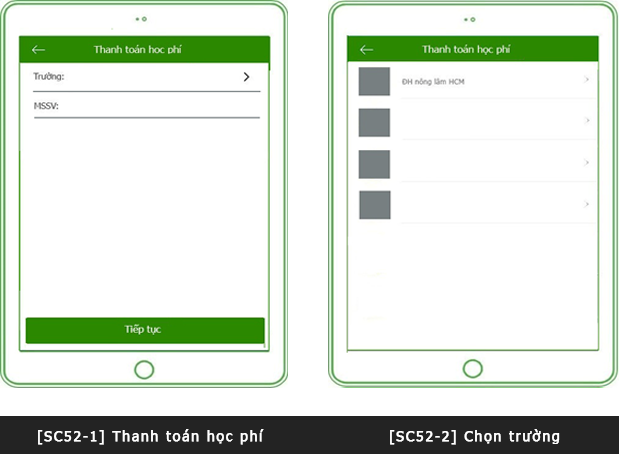
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng thanh toán học phí |
| Trình trạng trước | Đã đăng nhập vào hệ thống, tài khoản trong ví/ngân hàng còn đủ tiền thanh toán |
| Trình trạng sau | Thông báo trình trạng giao dịch (thành công/thất bại) cập nhật lịch sử giao dịch |

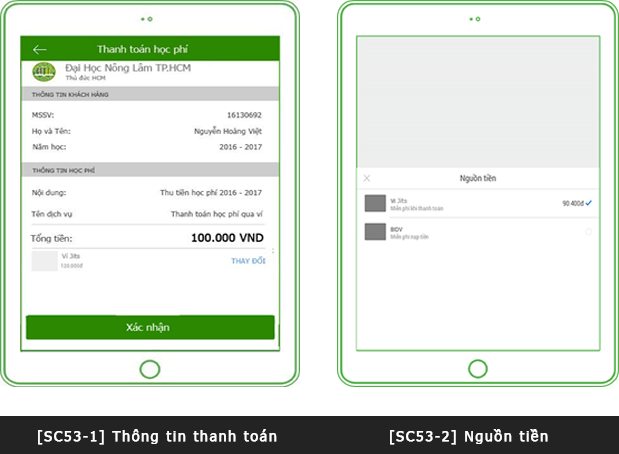
#### Liên kết các màn hình

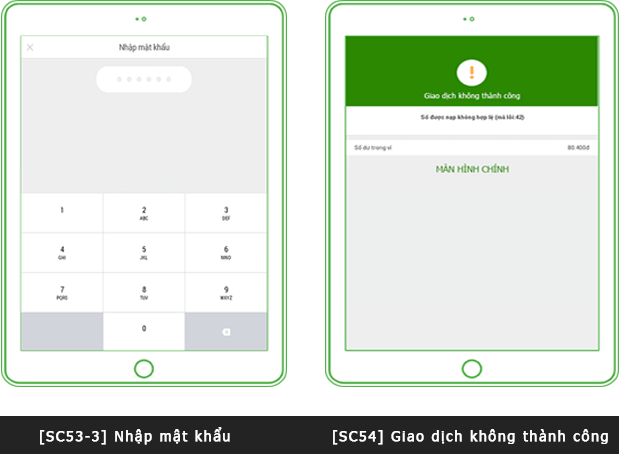


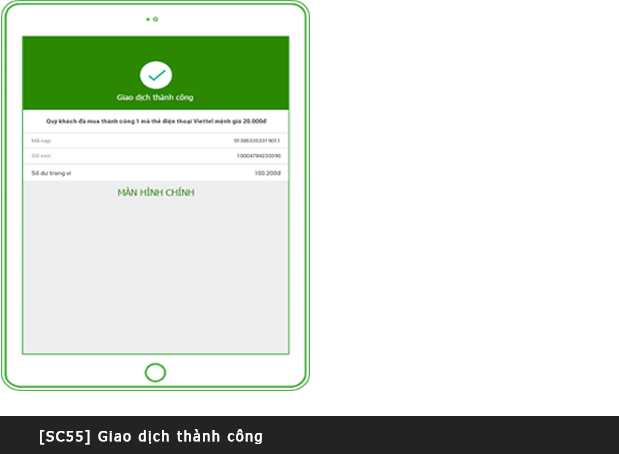
Hình 50: lượt đồ chức năng thanh toán học phí

#### Hình ảnh









#### Dòng sự kiện click chuyển trang

|  |  |  |
| --- | --- | --- |
| STT | Component | Màn hình chuyển trang |
| 1 | [SC52-1] Nút trở về | [[SC00-1]](#_Hình_ảnh_1) |
| 2 | [SC52-1] Chọn trường | [[SC52-2]](#_Liên_kết_các_2) |
| 3 | [SC52-1] Tiếp tục | [[SC53-1]](#_Liên_kết_các_2) |
| 4 | [SC52-2] Chọn item trường | [[SC52-1]](#_Liên_kết_các_2) |
| 5 | [SC53-1] Thay đổi nguồn tiền | [[SC53-2]](#_Liên_kết_các_2) |
| 6 | [SC53-1] Xác nhận | [[SC53-3]](#_Liên_kết_các_2) |
| 7 | [SC53-2] Chọn item nguồn tiền | [[SC53-1]](#_Liên_kết_các_2) |
| 8 | [SC53-3] Mật khẩu đúng, đủ tiền thanh toán | [[SC55]](#_Liên_kết_các_2) |
| 9 | [SC53-3] Mật khẩu đúng, không đủ tiền thanh toán | [[SC54]](#_Liên_kết_các_2) |
| 10 | [SC54] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |
| 11 | [SC55] Màn hình chính | [[SC00-1]](#_Hình_ảnh) |

#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC52-1] Nút trở về | Button | Trở về [SC00-1] |
| 2 | [SC52-1] Chọn trường | View | Chọn trường cần thanh toán học phí. Click chuyển sang [SC52-2] |
| 3 | [SC52-1] Tiếp tục | Button | MSSV tồn tại trong trường đã chọn thì chuyển đến [SC53-1]. Nếu không thì không cho chuyển |
| 4 | [SC52-2] Chọn item trường | View | Chọn 1 trường cần thanh toán. Click chuyển sang [SC52-1] |
| 5 | [SC53-1] Thay đổi nguồn tiền | View | Thay đổi nguồn tiền thanh toán. Click chuyển sang [SC53-2] |
| 6 | [SC53-1] Xác nhận | View | Chuyển đế [SC53-3] |
| 7 | [SC53-2] Chọn item nguồn tiền | ListItem | Chọn nguồn tiền cần thanh toán. Click chuyển sang [SC49-1] |
| 8 | [SC53-3] Nhập mật khẩu | TextInput | Nếu mật khẩu đúng, đủ tiền thanh toán chuyển đến [SC55]. Nếu mật khẩu đúng không đủ tiền [SC54]. Nếu sai mật khẩu không cho chuyển. Lỗi khác chuyển về [SC54] |
| 9 | [SC54] Màn hình chính | Text | Chuyển đến [SC00-1] |
| 10 | [SC55] Màn hình chính | Text | Chuyển đến [SC00-1] |

#### Bắt lỗi nhập dữ liệu

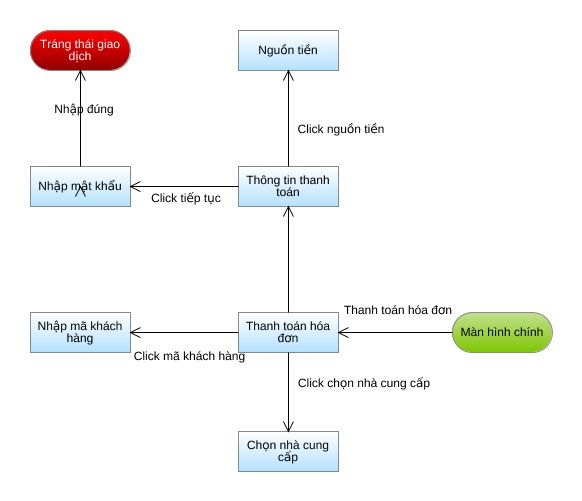
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | Tài khoản không đủ tiền | Thanh toán hóa đơn 100k mà TK chỉ còn 20k | Tài khoản của bạn không đủ để thực hiện giao dịch, xin vui lòng nạp thêm |
| 2 | Chưa nhập MSSV | “” | Trường này không được để trống |
| 3 | MSSV Không tồn tại | Không tồn tại trong CSDL | Mã số sinh viên không tồn tại |
| 4 | MSSV Không đúng định dạng | ĐH Nông Lâm, có mã bắt đầu K42 là 16XX, K41 là 15XX | Bạn nhập không đúng định dạng |

### Chức năng thanh toán hóa đơn (điện/nước)

#### Mô tả chung

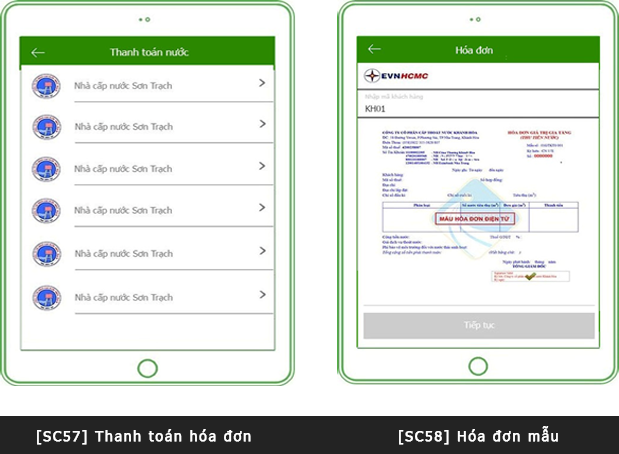
|  |  |
| --- | --- |
| Mô tả chức năng | Cho phép người dùng thanh toán hóa đơn tiền điện/nước |
| Trình trạng trước | Đã đăng nhập vào hệ thống, tài khoản trong ví/ngân hàng còn đủ tiền thanh toán |
| Trình trạng sau | Thông báo trình trạng giao dịch (thành công/thất bại) cập nhật lịch sử giao dịch |

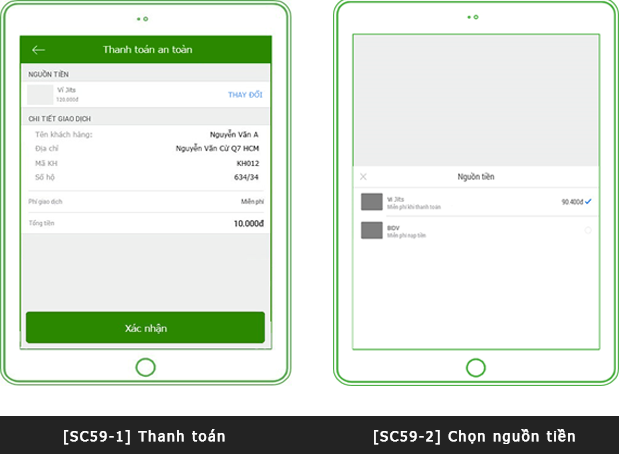
#### Liên kết các màn hình



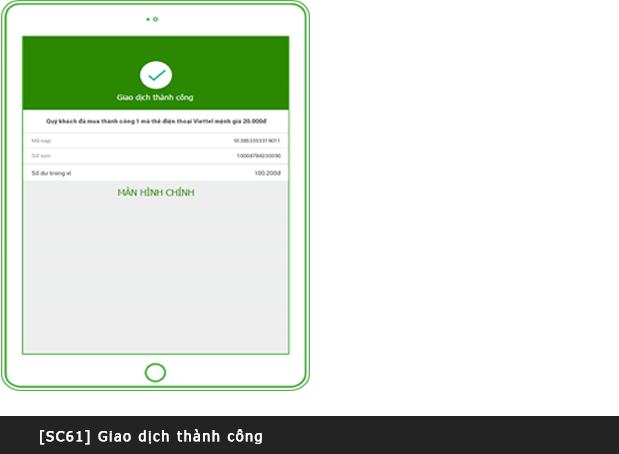
Hình 51: Lượt đồ chức năng thanh toán hóa đơn (điện/nước)

#### Hình ảnh









#### Dòng sự kiện click chuyển trang

|  |  |  |
| --- | --- | --- |
| STT | Component | Màn hình chuyển trang |
| 1 | [SC57] Nút trở về | [[SC00-1]](#_Hình_ảnh) |
| 2 | [SC57] Danh sách các nhà cung cấp | [[SC58]](#_Liên_kết_các) |
| 3 | [SC58] Tiếp tục | [[SC59-1]](#_Liên_kết_các) |
| 4 | [SC59-1] Click chọn item nguồn tiền | [[SC59-2]](#_Liên_kết_các) |
| 5 | [SC59-1] Xác nhận | [[SC59-3]](#_Liên_kết_các) |
| 6 | [SC59-2] Chọn nguồn tiền thanh toán | [[SC59-1]](#_Liên_kết_các) |
| 7 | [SC59-3] Nhập mật khẩu đúng, đủ tiền | [[SC61]](#_Liên_kết_các) |
| 8 | [SC59-3] Nhập mật khẩu đúng, không đủ tiền | [[SC60]](#_Liên_kết_các) |
| 9 | [SC60] Màn hình chính | [[SC57]](#_Liên_kết_các) |
| 10 | [SC61] Màn hình chính | [[SC57]](#_Liên_kết_các) |

#### Mô tả chi tiết các component

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Component | Type | Mô tả |
| 1 | [SC57] Nút trở về | Button | Trở về [SC00-1] |
| 2 | [SC57] Danh sách các nhà cung cấp | ListItem | Chọn nhà cung cấp thanh toán. Click chuyển sang [SC58] |
| 3 | [SC58] Tiếp tục | Button | Nhập mã khách hàng nếu đúng với hãng cung cấp thì chuyển sang [SC59-1]. Nếu sai thì thông báo không tồn tại và không cho chuyển |
| 4 | [SC59-1] Click chọn item nguồn tiền | View | Chọn nguồn tiền cần thanh toán. Click chuyển sang [SC59-2] |
| 5 | [SC59-1] Xác nhận | Button | Chuyển đến [SC59-3] |
| 6 | [SC59-2] Chọn nguồn tiền thanh toán | ListItem | Chon nguồn tiền cân thanh toán. Click chuyển sang [SC59-1] |
| 7 | [SC59-3] Nhập mật khẩu | TextInput | Nếu mật khẩu đúng, đủ tiền thanh toán chuyển đến [SC61]. Nếu mật khẩu đúng không đủ tiền [SC60]. Nếu sai mật khẩu không cho chuyển. Lỗi khác chuyển về [SC60] |

#### Bắt lỗi nhập dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Lỗi | Ví dụ | Thông báo |
| 1 | Mã khách hàng bỏ trống | “” | Mã khách hàng không được để trống |
| 2 | Mã khách hàng không đủ số, dư số | 032 xxx xxx | Không giống với định dạng của 1 mã khách hàng |
| 3 | Tài khoản không đủ tiền | Thanh toán hóa đơn 100k mà TK chỉ còn 20k | Tài khoản của bạn không đủ để thực hiện giao dịch, xin vui lòng nạp thêm |

# YÊU CẦU PHI CHỨC NĂNG CỦA PHẦN MỀM

## Yêu cầu bảo mật

* Xác thực hai lớp: bằng [1] Mật khẩu do chính bạn tự đặt và [2] Mã xác thực OTP được gửi tới số điện thoại đăng ký.
* Tự động khóa ứng dụng khi bạn không thao tác trong vòng 5 phút hoặc ngay khi tắt ứng dụng. Ngoài ra, bạn có thể cài đặt tự động khóa ngay khi đóng màn hình ứng dụng.
* Bảo mật đường truyền internet chuẩn SSL/TLS được cấp chứng chỉ bởi tổ chức bảo mật toàn cầu GlobalSign, giúp bảo vệ dữ liệu toàn vẹn khi giao dịch trong môi trường mạng.
* Tokenization: không lưu thông tin số thẻ quốc tế của bạn mà sử dụng dãy ký tự đặc biệt được mã hóa từ số thẻ và không thể giải mã.
* Hệ thống bảo mật thông minh của có khả năng nhận diện các giao dịch bất thường để ngay lập tức chặn giao dịch, khóa tài khoản, đồng thời cảnh báo tới bạn.

## Yêu cầu sao lưu

* Dữ liệu lưu trong hệ thống được sao lưu dự phòng tự động 24/24 bằng một hệ thống song hành tránh mất mát dữ liệu. Dữ liệu hệ thống có thể kết xuất ra các thiết bị lưu trữ ngoài và phục hồi khi cần thiết.

## Các yêu cầu về tính sử dụng (Usability)

* Hệ thống cho phép truy cập dữ liệu thời gian thực. Các tác vụ thực hiện tức thời trong thời gian ngừng cho phép chấp nhận dưới 30s.
* Hệ thống đảm bảo phục vụ truy cập online khoảng 10000 người cùng một lúc.
* Cung cấp một giao diện thân thiện phù hợp với quy trình nghiệp vụ hiện đang vận hành. Hệ thống đơn giản trong cài đặt và quản lý.

### Các yêu cầu về tính ổn định (Reliability)

### Tính sẵn sàng (Availability)

* Tỷ lệ phần trăm sẵn sàng là 95%.
* Thời gian trung bình giữa hai sự cố được tính bằng giờ.
* Thời gian trung bình phải sửa chữa khi hệ thống bị lỗi, cho phép hệ thống không làm việc 8h.

### Tính chính xác

* Các chức năng phải hoạt động chính xác 100%.
* Khi xảy ra các sự cố làm ngừng vận hành hệ thống, hệ thống phải đảm bảo phục hồi 90% trong vòng 1h và 100% trong vòng 24h.
* Hệ thống gây trung bình 1 lỗi/tháng trong 3 tháng vận hành đầu tiên. 1 lỗi/năm trong 3 năm vận hành tiếp theo và 0 lỗi/năm trong các năm vận hành tiếp theo. Lỗi chấp nhận là lỗi trung bình không gây tổn hại trầm trọng hệ thống và có thể phục hồi 95% hiệu quả.

## Các yêu cầu về hiệu năng (Performance)

* Các tác vụ thực hiện tức thời trong thời gian ngừng cho phép chấp nhận dưới 10s.
* Hệ thống có thể hỗ trợ tối đa 1000 giao dịch trong cùng 1 lúc.
* Hệ thống có thể hỗ trợ tối đa 10000 người đăng nhập online cùng một lúc.

## Các yêu cầu về tính hỗ trợ (Supportability)

* Hệ thống được hỗ trợ 24/24 trong vòng 1 năm miễn phí sau khi hệ thống vận hành chính thức. Sau một năm tiến hành thu phí theo thỏa thuận.
* Hệ thống phản hồi trong vòng tối đa 24h.

## Các ràng buộc thiết kế (Design Constraints)

* Hệ thống được xây dựng trên mã nguồn mở.
* Sử dụng React Native, Redux và NodeJS.
* Web server được sử dụng là NodeJS.
* Có thể chạy được trên cả Android và IOS.
* Hệ quản trị cơ sở dữ liệu là SQL Server.
* Công cụ phát triển là Sublime Text, Github.
* Phân tích và thiết kế được thực hiện theo chuẩn UML.
* Các công cụ hỗ trợ không tính bản quyền, thư viện hỗ trợ khác phải là mã nguồn mở.
* Hệ thống được thiết kế theo hướng có khả năng phát triển trong tương lai với việc thêm bớt các module, hoặc tích hợp hệ thống vào một hệ thống khác dễ dàng.

## Giao tiếp (Interfaces)

### Giao tiếp người dùng (User interfaces)

* Giao diện điện thoại.
* Giao diện mang tính hiện đại, có tính thẩm mỹ.
* Font chữ người dùng tự chọn trong cài đặc điện thoại.
* Giao diện thiết kế trên màn hình độ phân giải tối thiểu 800x600, chế độ màu tối thiểu high color (16 bits).
* Ngôn ngữ sử dụng trong toàn bộ hệ thống là tiếng việt.
* Định dạng ngày được sử dụng trong hệ thống là dd/mm/yyyy.
* Định dạng số được sử dụng trong hệ thống là 000.000. 000.

### Giao tiếp phần cứng (Hardware interfaces)

* Phần mềm chạy trên máy với cấu hình tối thiểu: Với Android là 600 MHz, 1GB Ram, với IOS là IPhone 5 trở lên.
* Máy phải có tối thiểu dư 500MB dung lượng, tốc độ mạng 256 Kbps.
* Với Android: Android 7 trở lên, IOS: IOS 9 trở lên.

### Giao tiếp phần mềm (Software interfaces)

* Phần mềm yêu cầu kết nối định vị, máy ảnh.

### Giao tiếp bên ngoài

* Hệ thống giao tiếp với địa chỉ web được API của các nhà dịch vụ cung cấp.

### Giao tiếp bên trong

* N/A.

### Giao tiếp truyền thông (Communications interfaces)

* Hệ thống vận hành tại máy chủ và được truy cập trực tiếp từ internet.
* Các yêu cầu tài liệu người dùng và hỗ trợ trực tuyến
* Tài liệu hướng dẫn sử dụng và các tính năng được có sẵn trong phần mô tả trên chợ ứng dụng.
* Hỗ trợ hệ thống giúp đỡ trực tuyến cho người dùng cuối. Hệ thống giúp đỡ trực tuyến được tích hợp trực tiếp vào hệ thống.
* Các yêu cầu pháp lý, Bảng quyền và những ghi chú khác.
* Hệ thống sử dụng toàn bộ open source từ bên ngoài.
* Sau khi hệ thống vận hành, toàn bộ source code phát triển hệ thống được chuyển giao và thuộc quyền quản lý của công ty JITS.

### Các tiêu chuẩn áp dụng

* Hệ thống sử dụng chuẩn Oauth để xác thực người dùng khi đăng nhập và đăng ký

### Các yêu cầu khác

* N/A

# KẾT QUẢ ĐẠT ĐƯỢC

**Ưu điểm :**

Xây dựng được một hệ thống CSDL phù hợp, đáp ứng được các nguyên tắc về yêu cầu thiết kế, CSDL có thế mở rộng trong tương lai. Tài liệu đáp ứng được việc triển khai dự án trong thực tế.

Phân tích ứng dụng Ví điện tử thông qua việc tìm hiểu thông tin qua mạng, qua các Ví điện tử đang được sử dụng phổ biến ở Việt Nam như: Momo, ZaloPay. Hiện thực được nhiều chức năng mà một Ví điện tử cần phải có:

* Nhóm chức năng tài khoản: cho phép người dùng tạo tài khoản, cập nhật thông tin cá nhân, đăng nhập, đổi mật khẩu, cập nhật thông tin cá nhân, đăng xuất.
* Nhóm chức năng nhận và chuyển tiền: cho phép người dùng liên kết tài khoản ngân hàng, nhận/chuyển tiền từ ví này qua ví khác hoặc qua ngân hàng khác.
* Nhóm chức năng thanh toán trực tiếp: cho phép người dùng mua/nạp thẻ điện thoại, thanh toán hóa đơn tiền điện/nước, nạp tiền vào game, mua vé xem phim, thanh toán tiền Grab

Nghiên cứu được những nguyên lý cơ bản, nâng cao của framework React Native và NodeJS để ứng dụng vào việc xây dựng hệ thống Ví điện tử. Bên cạnh đó cũng áp dụng thành công nhiều thư viện bên ngoài để thực hiện nhiều chức năng khác nhau

Đã áp dụng được các biện pháp bảo mật mà một hệ thống Ví điện tử cần phải có như: ISO-8583, JWT, …

Thông qua quá trình thực tập tại công ty JITS chúng em được hỗ trợ về cách thức thực hiện một hệ thống thế nào là đúng nhất. Vì vậy hệ thống chúng em đã xây dựng và phát triển hoàn toàn có thể triển khai được trong thực tế.

Trong tương lai khi hệ thống được triển khai thực tế, Server cần phải được nâng cấp và tổ chức lại phù hợp với số lượng người dung mà hệ thống phải đáp ứng. Nghiên cứu chuyên sâu về React Native, NodeJS để optimize hệ thống chạy nhanh hơn.

Đề tài nghiên cứu này hỗ trợ chúng em có được những kiến thức về ngôn ngữ JavaScript là một ngôn ngữ phổ biến hiện nay, giúp chúng em có được nền tảng giúp ích cho việc tìm một công việc tốt sau khi ra trường.

**Hạn chế :**

Đề tài này còn sử dụng quá nhiều API giả lập ( các hệ thống ngân hàng, các chức năng thanh toán…), ngoài liên kết với 2 API thật là mua thẻ điện thoại và gửi tin nhắn OTP về điện thoại người dùng, hệ thống không sử dụng thêm API thật nào.

Chưa nghiên cứu được cách liên kết với các đại lý cổng thanh toán trực tuyến, cách liên kết với các nhà cung cấp dịch vụ để thanh toán ( thanh toán tiền điện : EVN…, thanh toán học phí,…)

|  |  |
| --- | --- |
| Tên tài liệu | Ghi chú |
| ‘React Native’ | <https://reactnative.dev/> |
| <https://react-native-elements.github.io/react-native-elements/> |
| “Redux” | <https://redux.js.org/basics/basic-tutorial> |
| “Momo” | <https://momo.vn/> |
| <https://momo.vn/huong-dan/lien-ket-ngan-hang-ctgr67> |
| <https://momo.vn/huong-dan/lien-ket-vi-momo-scb> |
| “I18n” | <https://www.npmjs.com/package/i18n> |
| “Axios” | <https://github.com/axios/axios> |
| “jwt” | <https://jwt.io/> |
| “ISO 8583” | <https://vi.wikipedia.org/wiki/ISO_8583> |
| <https://viblo.asia/p/cau-chuyen-ve-fintech-va-cac-giai-phap-thanh-toan-p2-iso8583-XL6lAv8J5ek> |
| <https://www.iso.org/obp/ui/#iso:std:iso:8583:-1:en> |
| <https://www.codeproject.com/Articles/100084/Introduction-to-ISO> |
| NodeJS | <https://dev.to/mnhattt/kin-trc-x-l-bt-ng-b-trong-nodejs-nnf> |
| <https://techtalk.vn/xu-li-bat-dong-bo-song-song-trong-node-js.html> |
| <https://nodejs.org/en/> |

# TÀI LIỆU THAM KHẢO