

Tổng Quan Ứng Dụng Chat - Retion UI Chat

Tài liệu mô tả chi tiết cấu trúc và luồng hoạt động của ứng dụng Chat

Cập nhật: 2026-01-17

Mục Lục

1. [OAuth - Xác thực người dùng](#)
2. [Select Page - Chọn trang quản lý](#)
3. [Setting Page - Cài đặt trang](#)
4. [Layout Chat - Giao diện chat](#)

1. OAuth - Xác thực người dùng

1.1 Tổng quan

Module OAuth xử lý toàn bộ luồng **đăng nhập, đăng ký, quên mật khẩu** và **xác thực email** của người dùng.

 **Thư mục:** `src/views/OAuth/`

1.2 Cấu trúc thư mục

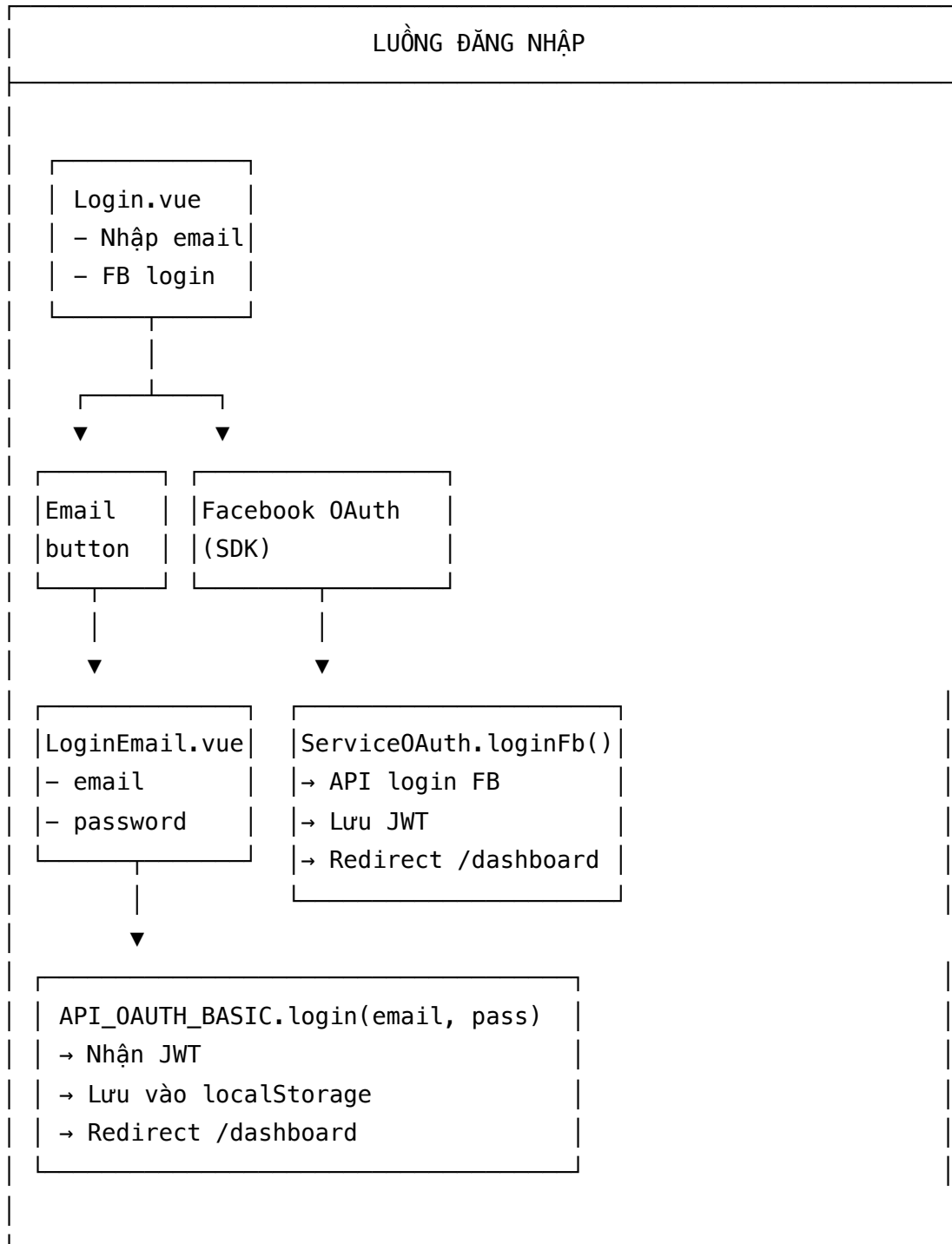
```
src/views/OAuth/  
├─ Login.vue           # Màn hình đăng nhập chính  
├─ LoginEmail.vue      # Đăng nhập bằng email + mật khẩu  
├─ Register.vue        # Màn hình đăng ký  
├─ RegisterDetail.vue  # Form điền thông tin đăng ký  
├─ ForgotPassword.vue  # Quên mật khẩu  
├─ ResetPassword.vue   # Đặt lại mật khẩu  
├─ Alert.vue           # Component thông báo  
├─ AlertError.vue      # Component thông báo lỗi  
├─ InputPassword.vue   # Input password có toggle hiển thị  
├─ NewTo.vue           # Link chuyển sang đăng ký  
├─ GoLogin.vue         # Link chuyển sang đăng nhập  
├─ Or.vue              # Phân cách "hoặc"  
├─ service.ts          # Service xử lý logic OAuth  
├─ store.ts            # Pinia store cho OAuth  
├─ validate.ts         # Validation rules (Yup)  
└─ index.scss          # Styles chung
```

1.3 Routes

Path	Component	Mô tả
/oauth	OAuthV2.vue	Layout wrapper cho OAuth
/oauth/login	Login.vue	Trang đăng nhập chính
/oauth/login-email	LoginEmail.vue	Đăng nhập email + mật khẩu
/oauth/register	Register.vue	Trang đăng ký
/oauth/register-detail	RegisterDetail.vue	Form chi tiết đăng ký
/oauth/forgot-password	ForgotPassword.vue	Quên mật khẩu
/oauth/reset-password	ResetPassword.vue	Đặt lại mật khẩu

1.4 Luồng hoạt động

Đăng nhập





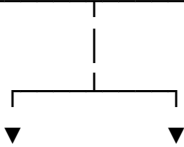
Đăng ký

LUỒNG ĐĂNG KÝ

Register.vue
- Nhập email
- Check email



API_OAUTH_BASIC.checkEmail()
→ Kiểm tra email tồn tại



Email đã
tồn tại
→ Báo lỗi

RegisterDetail.vue
- Email
- Họ, Tên
- Mật khẩu
- Xác nhận mật khẩu



API_OAUTH_BASIC.register()
→ Tạo tài khoản
→ Gửi email xác thực



Redirect → LoginEmail.vue
với thông báo xác thực

1.5 Service chính

File: service.ts

```
/** Các service chính trong composableService() */

class Service0Auth {
  /** Đăng nhập bằng Facebook token */
  async loginFb(access_token: string)

  /** Chuyển hướng trang */
  redirect(to: RouteLocationRaw)
}

/** Xử lý lỗi custom theo message */
function customError(message?: string): string | undefined

/** Toast thông báo với xử lý lỗi đặc biệt */
class CustomToast extends Toast implements IAlert
```

1.6 Validation

File: validate.ts

- VLD_EMAIL : Validate email
- VLD_EMAIL_PASSWORD : Validate email + password
- VLD_EMAIL_REGISTER : Validate form đăng ký đầy đủ

2. Select Page - Chọn trang quản lý

2.1 Tổng quan

Module Select Page cho phép người dùng **xem**, **quản lý** và **chọn** các trang (Facebook, Instagram, Zalo, Website) để chat.

 **Thư mục:** src/views/Dashboard/SelectPage/

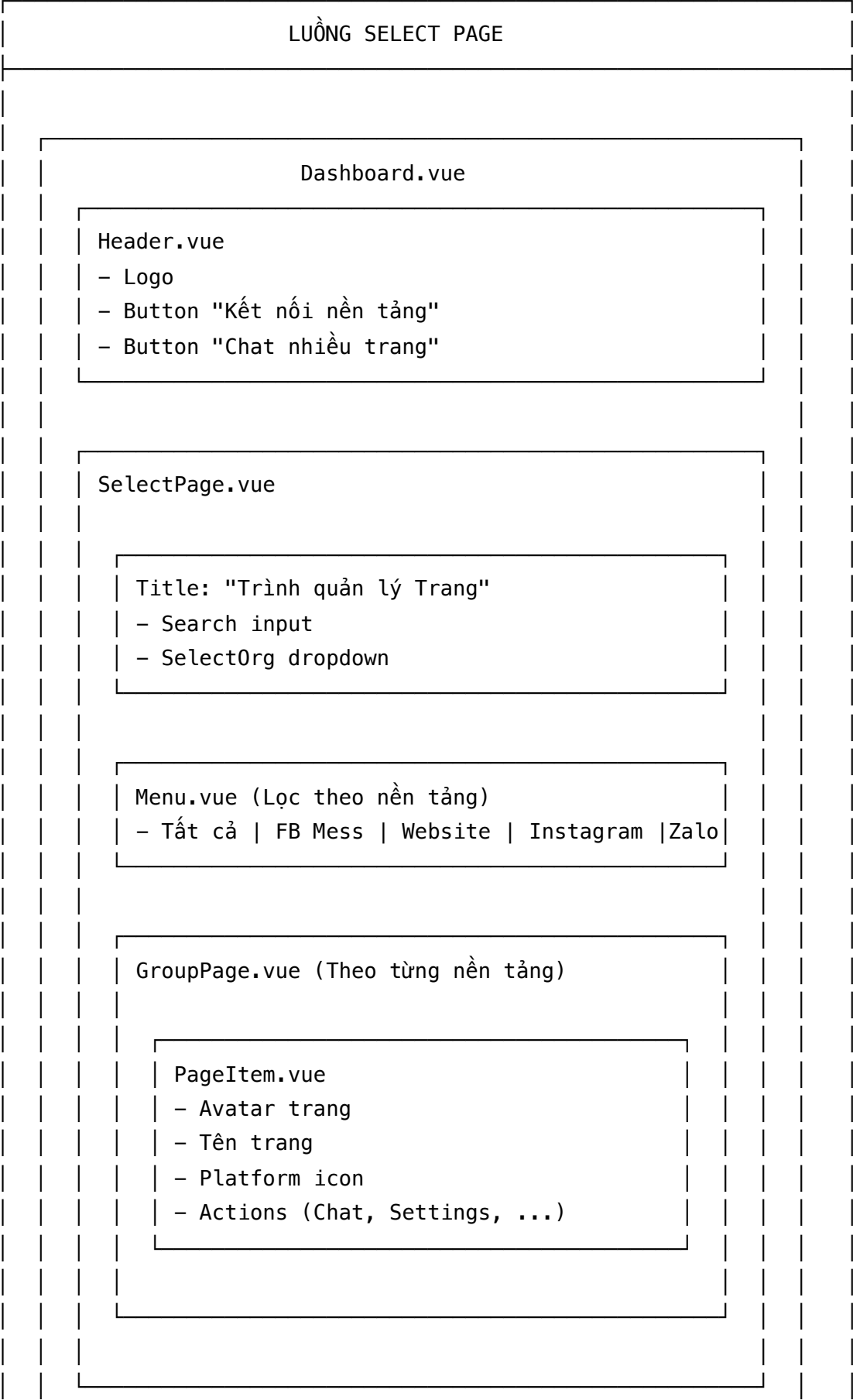
2.2 Cấu trúc thư mục

```
src/views/Dashboard/
├─ Dashboard.vue           # Layout Dashboard chính
├─ SelectPage.vue          # Màn hình quản lý trang
├─ SelectPlatform.vue      # Chọn nền tảng kết nối
├─ ConnectPage.vue         # Modal kết nối trang mới
├─ Header.vue              # Header Dashboard
├─ SelectPage/
│   ├─ Menu.vue            # Menu lọc theo nền tảng
│   ├─ GroupPage.vue       # Nhóm trang theo loại
│   ├─ PageItem.vue        # Item hiển thị 1 trang
│   └─ PageItem/
│       └─ Action.vue       # Actions của trang
│       └─ ...
│   └─ GroupPageAction.vue  # Action bar khi chọn nhóm
├─ SelectGroup.vue         # Chọn nhóm trang
├─ AssignGroup.vue         # Gán nhóm cho trang
├─ EmptyPage.vue           # Trạng thái không có trang
├─ AllOrg/
│   └─ AllOrg.vue          # Xem tất cả tổ chức
│   └─ ...
├─ DropdownPickConnectPlatform.vue # Dropdown chọn nền tảng
├─ symbol.ts               # Provide/Inject keys
└─ type.ts                 # TypeScript types
```

2.3 Routes

Path	Component	Mô tả
/dashboard	Dashboard.vue	Layout Dashboard
/dashboard/select-page	SelectPage.vue	Quản lý trang
/dashboard/select-platform	SelectPlatform.vue	Chọn nền tảng

2.4 Luồng hoạt động



2.5 Các nền tảng hỗ trợ

Nền tảng	Filter Key	Icon
Facebook Messenger	FB_MESS	FacebookIcon
Website Chat	WEBSITE	WebIcon
Instagram	FB_INSTAGRAM	InstagramIcon
Zalo	ZALO	ZaloIcon

2.6 Tính năng chính

1. **Lọc theo nền tảng:** Menu tabs để lọc trang theo loại
2. **Tìm kiếm:** Search input tìm trang theo tên
3. **Chọn tổ chức:** Dropdown chọn tổ chức quản lý
4. **Nhóm trang:** Gom nhóm nhiều trang để chat cùng lúc
5. **Kết nối nền tảng:** Modal kết nối trang mới từ FB/Zalo/etc

2.7 Lifecycle

```
onMounted(async () => {  
  // Load lại info của chatbot user  
  getMeChatbotUser?.()  
  
  // Kích hoạt tự động mở kết nối nền tảng nếu cần  
  triggerConnectPlatform()  
  
  // Lấy toàn bộ dữ liệu tổ chức và trang  
  await getAllOrgAndPage()  
  
  // Xử lý đăng nhập không có page  
  handleLoginWithoutPage()  
})
```


3. Setting Page - Cài đặt trang

3.1 Tổng quan

Module Setting cho phép cấu hình **cài đặt cá nhân**, **cài đặt tổ chức**, **quản lý thanh toán** và **API/Webhook**.

 **Các thư mục liên quan:**

```
src/views/Dashboard/
├── Setting/
│   └── PersonalSetting.vue    # Cài đặt cá nhân
├── Org/
│   ├── Setting.vue          # Cài đặt tổ chức
│   ├── App.vue              # Quản lý ứng dụng
│   ├── Api.vue              # Quản lý API keys
│   ├── Agent.vue            # Trợ lý ảo AI
│   ├── Webhook.vue          # Webhook settings
│   └── Pay/
│       ├── Info.vue         # Thông tin thanh toán
│       └── ReCharge.vue      # Nạp tiền
├── Widget/
│   ├── AllWidget.vue        # Chợ widget
│   ├── InstalledWidget.vue  # Widget đã cài
│   └── MyWidget.vue         # Widget của tôi
└── User.vue                 # Quản lý người dùng
```

3.2 Routes

Path	Component	Mô tả
/dashboard/org/setting	OrgSetting.vue	Cài đặt tổ chức
/dashboard/org/pay	OrgPay.vue	Thanh toán
/dashboard/org/pay/info	OrgPayInfo.vue	Thông tin thanh toán
/dashboard/org/pay/recharge	OrgPayReCharge.vue	Nạp tiền
/dashboard/org/app	OrgApp.vue	Quản lý ứng dụng
/dashboard/org/api	OrgApi.vue	API Keys

Path	Component	Mô tả
/dashboard/org/virtual-assistant	OrgAgent.vue	Trợ lý ảo
/dashboard/org/webhook	OrgWebhook.vue	Webhook
/dashboard/widget/market	AllWidget.vue	Chợ widget
/dashboard/widget/installed	InstalledWidget.vue	Widget đã cài
/dashboard/widget/my-widget	MyWidget.vue	Widget của tôi
/dashboard/user	User.vue	Quản lý người dùng

3.3 Personal Setting

File: PersonalSetting.vue

Các tùy chọn cá nhân:

Setting	Mô tả
is_enable_personal_setting	Bật/tắt ghi đè setting của page
is_hide_page_avatar	Ẩn avatar trang trong danh sách
display_label_type	Kiểu hiển thị nhãn: FULL / ICON / ICON_TOOLTIP

Lưu trữ:

```
// Lưu vào localStorage
setItem('personal_settings', personal_setting.value)
setItem('is_enable_personal_setting', 'yes' | 'no')
```

3.4 Org Setting


Các cài đặt cấp độ tổ chức:

- **Thông tin tổ chức:** Tên, mô tả, logo
- **Thanh toán:** Gói dịch vụ, quota, nạp tiền
- **API & Webhook:** Keys, endpoints
- **Widget:** Quản lý ứng dụng bên thứ 3
- **Trợ lý ảo:** Cấu hình AI chatbot

4. Layout Chat - Giao diện chat

4.1 Tổng quan

Layout Chat là màn hình chính để **nhắn tin, quản lý hội thoại, xem thông tin khách hàng và sử dụng widget.**

 **Thư mục:** `src/views/ChatWarper/`

4.2 Cấu trúc thư mục

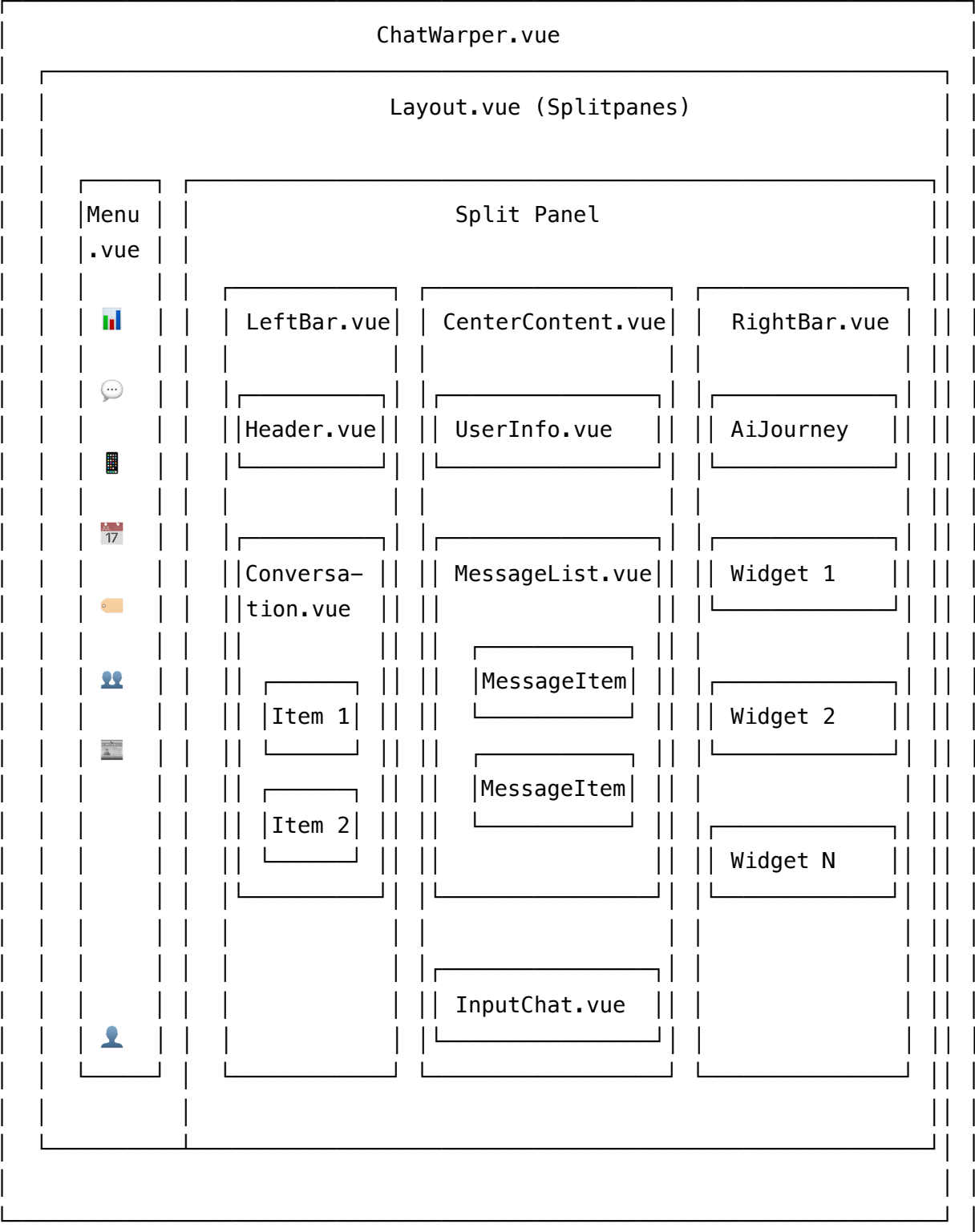
```
src/views/ChatWarper/
├─ ChatWarper.vue          # Layout wrapper + Socket handler
├─ Layout.vue              # Splitpanes layout (trái/phải)
├─ Menu.vue                # Menu sidebar với các bộ lọc
├─ Menu/
│   └─ NavItem.vue         # Item navigation
│   └─ FilterInteract.vue  # Lọc theo tương tác
│   └─ FilterMessage.vue   # Lọc theo tin nhắn
│   └─ FilterPhone.vue     # Lọc theo số điện thoại
│   └─ FilterDate.vue      # Lọc theo ngày
│   └─ FilterTag.vue       # Lọc theo nhãn
│   └─ FilterNotTag.vue    # Lọc không có nhãn
│   └─ FilterStaff.vue     # Lọc theo nhân viên
│   └─ FilterPost.vue      # Lọc theo bài post
│   └─ Attach.vue          # Menu đính kèm
│   └─ User.vue            # User dropdown
└─ Chat/
    ├─ LeftBar.vue         # Cột trái - Danh sách hội thoại
    ├─ LeftBar/
    │   └─ Header.vue      # Header với search + tabs
    │   └─ Conversation.vue # Danh sách hội thoại
    │   └─ Conversation/
    │       └─ Item.vue     # Item hội thoại
    │       └─ ...
    └─ CenterContent.vue   # Cột giữa - Khung chat
        └─ CenterContent/
            └─ UserInfo.vue  # Thông tin user đang chat
            └─ MessageList.vue # Danh sách tin nhắn
                └─ MessageList/
                    └─ MessageItem/ # Các loại tin nhắn
                        └─ ...
            └─ InputChat.vue  # Input gửi tin nhắn
            └─ InputChat/
                └─ TextArea.vue # TextArea input
                └─ Attachment.vue # Đính kèm file
                └─ QuickReply.vue # Trả lời nhanh
                └─ ...
            └─ UserInfo/
                └─ Info.vue    # Chi tiết thông tin
                └─ Note.vue    # Ghi chú
                └─ ...
```

```
|— RightBar.vue          # Cột phải – Widget panel
└─ RightBar/
    |— WidgetSorting.vue  # Sắp xếp widget
    |— PostRightBar.vue   # Right bar cho Post
    └─ PostAnalytic/      # Phân tích bài post
```

4.3 Routes

Path	Component	Mô tả
/ hoặc /chat	ChatWarper.vue	Màn hình chat chính

4.4 Layout Structure



4.5 Components chính

4.5.1 ChatWarper.vue

Chức năng:

- Khởi tạo và quản lý **Socket connection**
- Xử lý các **socket events** (tin nhắn mới, cập nhật hội thoại)
- Gửi **notifications** (web push, sound)
- Validate conversation theo **bộ lọc**
- Load **page info** và **widget tokens**

Lifecycle:

```
onMounted(async () => {  
  // Khởi tạo socket  
  $socket.connect()  
  
  // Lấy thông tin page  
  await $main.getPageInfoToChat()  
  
  // Kiểm tra quyền thông báo  
  checkAllowNoti()  
  
  // Lắng nghe focus/blur  
  window.addEventListener('focus', checkFocusChatTab)  
})  
  
onUnmounted(() => {  
  // Đóng socket  
  $socket.close()  
})
```

4.5.2 Menu.vue

Các bộ lọc:

Bộ lọc	Component	Icon	Mô tả
Menu	Attach.vue	Squares2X2Icon	Menu chính
Tương tác	FilterInteract.vue	InboxIcon	Lọc theo tương tác

Bộ lọc	Component	Icon	Mô tả
Tin nhắn	FilterMessage.vue	ChatDotIcon	Lọc theo tin nhắn
Số điện thoại	FilterPhone.vue	PhoneIcon	Lọc SĐT
Ngày	FilterDate.vue	DateIcon	Lọc theo ngày
Nhãn	FilterTag.vue	TagIcon	Lọc theo nhãn
Không có nhãn	FilterNotTag.vue	TagNotIcon	Lọc không có nhãn
Nhân viên	FilterStaff.vue	UsersIcon	Lọc theo nhân viên
Bài post	FilterPost.vue	NewSpaperIcon	Lọc theo post

4.5.3 LeftBar.vue

Cấu trúc:

- Header.vue : Search input, tabs (ALL/UNREAD/REMIND)
- Conversation.vue : Danh sách hội thoại với infinite scroll

4.5.4 CenterContent.vue

Cấu trúc:

- UserInfo.vue : Avatar, tên, nền tảng, actions
- MessageList.vue : Danh sách tin nhắn (flex-col-reverse)
- InputChat.vue : Textarea, emoji, attachment, quick reply

4.5.5 RightBar.vue

Cấu trúc:

- AiJourney : Lịch trình AI
- **Widget list:** Các widget đã cài đặt
- WidgetSorting.vue : Sắp xếp thứ tự widget
- PostRightBar.vue : Hiển thị khi xem comment post

4.6 Socket Events

```
handleSocketEvent(socket_data: {  
  conversation?: ConversationInfo, // Dữ liệu khách hàng  
  message?: MessageInfo,          // Tin nhắn mới  
  staff?: StaffSocket,             // Dữ liệu nhân viên  
  event?: SocketEvent,             // Tên sự kiện  
  update_message?: MessageInfo,    // Tin nhắn cần cập nhật  
  update_comment?: FacebookCommentPost // Comment cập nhật  
})
```

Các event được xử lý:

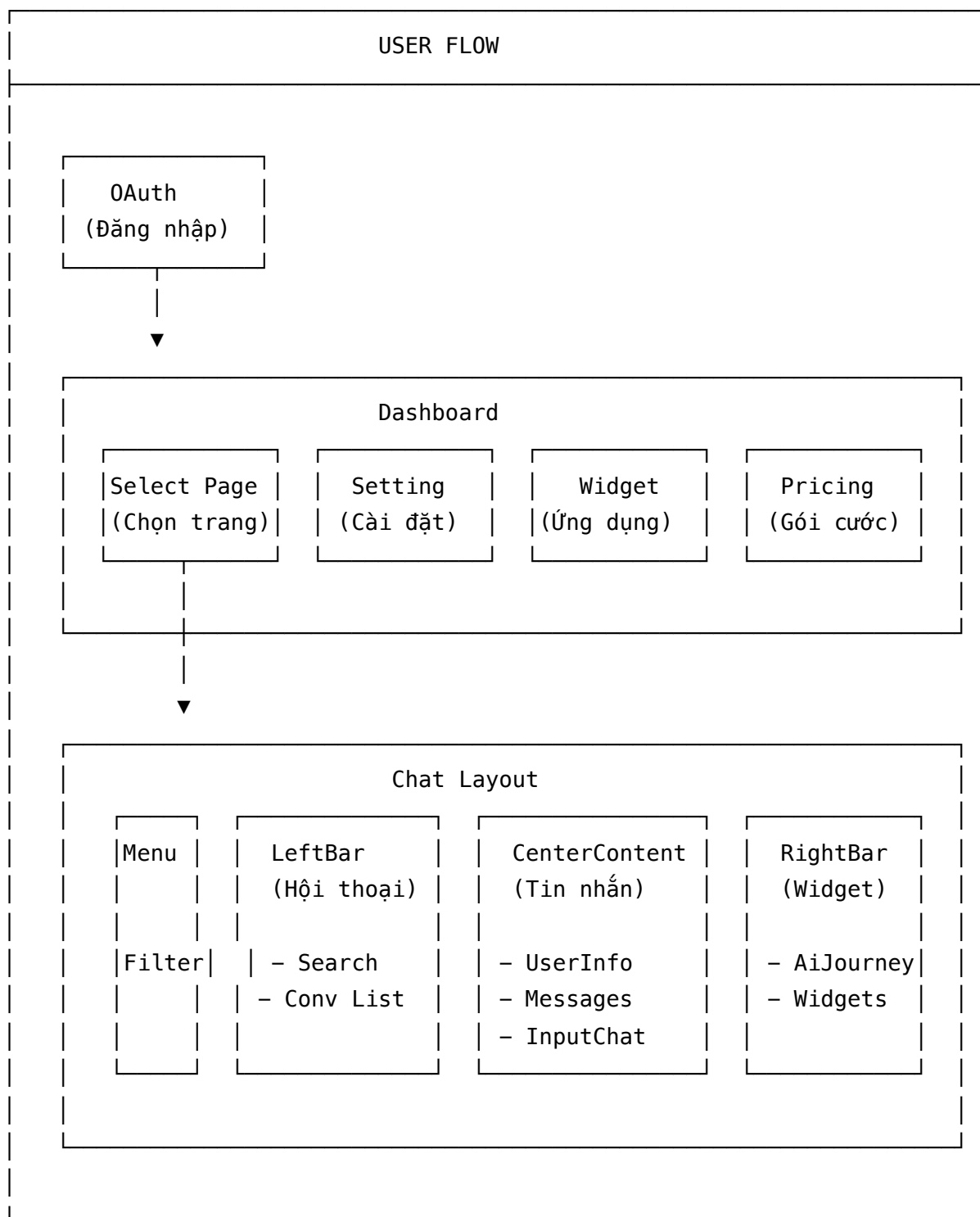
- new_message : Tin nhắn mới
- update_conversation : Cập nhật hội thoại
- update_message : Cập nhật tin nhắn
- staff_typing : Nhân viên đang gõ
- ...

4.7 Splitpanes Layout

File: Layout.vue

```
// Cấu hình  
const MIN = 360 // Độ rộng tối thiểu cột trái (px)  
const MAX = 460 // Độ rộng tối đa cột trái (px)  
  
// Lưu trữ  
$localStorage.setItem('conversation_width', size)
```

Sơ đồ tổng quan





References

- **Routes:** `src/router/routes.ts`
- **Stores:** `src/stores/`
- **API Services:** `src/utils/api/`
- **Components:** `src/components/`

Ghi chú: Tài liệu này mô tả cấu trúc tổng quan của ứng dụng. Mỗi module có thể được mở rộng với documentation chi tiết hơn nếu cần.