📊 Dashboard Module - Tài liệu chi tiết

> **Mô tả chi tiết Dashboard bao gồm Select Page, Connect Page, Org Settings, Widgets**
>
> Cập nhật: 2026-01-19

📋 Mục Lục

1. Tổng quan
2. Cấu trúc thư mục
3. Routes & Navigation
4. Dashboard Layout
5. Select Page Module
6. Connect Page Module
7. Organization Settings
8. Widget Management
9. Stores & State Management
10. Composables

1. Tổng quan

1.1 Mục đích

Dashboard là **trung tâm quản lý** của ứng dụng, cho phép:

- Quản lý các **trang (Pages)** được kết nối (Facebook, Zalo, Instagram, Website)
- **Kết nối nền tảng** mới
- Cài đặt **tổ chức (Organization)**
- Quản lý **Widget/Ứng dụng** bên thứ 3
- Xem thông tin **thanh toán & gói dịch vụ**

1.2 Tech Stack

Công nghệ	Mục đích
Vue 3 Composition API	Framework
Pinia	State management
vue-i18n	Đa ngôn ngữ
Heroicons	Icons
TailwindCSS	Styling

2. Cấu trúc thư mục

```
src/views/Dashboard/
├── Dashboard.vue              # Layout wrapper chính
├── Header.vue                 # Header với nav buttons
│
├── SelectPage/                # Module quản lý trang
│   ├── SelectPage.vue         # Trang chính quản lý pages
│   ├── 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 chế độ chọn nhóm
│   ├── SelectGroup.vue        # Dropdown chọn nhóm
│   ├── AssignGroup.vue        # Gán nhóm cho trang
│   ├── EmptyPage.vue          # Trạng thái không có trang
│   ├── DropdownPickConnectPlatform.vue  # Dropdown nền tảng
│   ├── AllOrg/                # Xem tất cả tổ chức
│   │   ├── AllOrg.vue
│   │   └── ...
│   ├── symbol.ts              # Provide/Inject keys
│   └── type.ts                # TypeScript types
│
├── ConnectPage/               # Module kết nối nền tảng
│   ├── ConnectPage.vue        # Modal kết nối chính
│   ├── Menu.vue               # Menu chọn nền tảng
│   ├── ActivePage.vue         # Danh sách trang đã kết nối
│   ├── Facebook.vue           # Kết nối Facebook
│   ├── Instagram.vue          # Kết nối Instagram
│   ├── Zalo.vue               # Kết nối Zalo
│   ├── Website.vue            # Tạo Website chat
│   ├── Member.vue             # Quản lý thành viên
│   └── symbol.ts
│
├── Org/                       # Cài đặt tổ chức
│   ├── Org.vue                # Layout wrapper
│   ├── Setting.vue            # Cài đặt chung
│   ├── App.vue                # Quản lý ứng dụng
│   ├── Api.vue                # API Keys
│   ├── Agent.vue              # Trợ lý ảo AI
│   ├── Webhook.vue            # Webhook settings
│   ├── Pay/                   # Thanh toán
```

```
│   │   ├── Info.vue          # Thông tin gói
│   │   └── ReCharge.vue      # Nạp tiền
│   └── ReChargeBtn.vue       # Button nạp tiền
│
├── Widget/                   # Quản lý Widget
│   ├── Widget.vue            # Layout wrapper
│   ├── AllWidget.vue         # Chợ widget
│   ├── InstalledWidget.vue   # Widget đã cài
│   └── MyWidget.vue          # Widget của tôi
│
├── Pricing/                  # Bảng giá
│   └── Pricing.vue
│
├── Setting/                  # Cài đặt cá nhân
│   └── PersonalSetting.vue
│
├── SelectPlatform.vue        # Chọn nền tảng kết nối
├── Download.vue              # Tải ứng dụng
├── User.vue                  # Quản lý người dùng
├── Noti.vue                  # Thông báo
├── SkeletonGroupPage.vue     # Loading skeleton
├── composables/              # Composable functions
│   └── usePageManager.ts
└── symbol.ts                 # Provide/Inject keys
```

3. Routes & Navigation

3.1 Route Configuration

```
{
  path: '/dashboard',
  redirect: '/dashboard/select-page',
  component: Dashboard,
  children: [
    { path: 'select-page', component: SelectPage },
    { path: 'select-platform', component: SelectPlatform },
    { path: 'pricing', component: Pricing },
    {
      path: 'widget',
      redirect: '/dashboard/widget/market',
      component: Widget,
      children: [
        { path: 'market', component: AllWidget },
        { path: 'installed', component: InstalledWidget },
        { path: 'my-widget', component: MyWidget },
      ],
    },
    {
      path: 'org',
      redirect: '/dashboard/org/setting',
      component: Org,
      children: [
        { path: 'setting', component: OrgSetting },
        {
          path: 'pay',
          redirect: '/dashboard/org/pay/info',
          component: OrgPay,
          children: [
            { path: 'info', component: OrgPayInfo },
            { path: 'recharge', component: OrgPayReCharge },
          ],
        },
        { path: 'app', component: OrgApp },
        { path: 'api', component: OrgApi },
        { path: 'virtual-assistant', component: OrgAgent },
        { path: 'webhook', component: OrgWebhook },
      ],
```

```
    },
    { path: 'noti', component: Noti },
    { path: 'download', component: Download },
    { path: 'user', component: User },
  ],
}
```

3.2 Route Map

```
/dashboard
    │
    ├── /select-page          → Quản lý trang (default)
    ├── /select-platform      → Chọn nền tảng kết nối
    ├── /pricing              → Bảng giá
    │
    ├── /widget
    │   ├── /market           → Chợ widget
    │   ├── /installed        → Đã cài đặt
    │   └── /my-widget        → Widget của tôi
    │
    ├── /org
    │   ├── /setting          → Cài đặt tổ chức
    │   ├── /pay
    │   │   ├── /info         → Thông tin thanh toán
    │   │   └── /recharge     → Nạp tiền
    │   ├── /app              → Ứng dụng
    │   ├── /api              → API Keys
    │   ├── /virtual-assistant → Trợ lý ảo
    │   └── /webhook          → Webhook
    │
    ├── /noti                 → Thông báo
    ├── /download             → Tải ứng dụng
    └── /user                 → Quản lý người dùng
```

4. Dashboard Layout

4.1 Dashboard.vue

Mục đích: Layout wrapper chính cho Dashboard

Cấu trúc:

```vue
<template>
  <div class="flex flex-col h-full p-3 gap-3">
    <!-- Header với các nút điều khiển -->
    <header class="flex-shrink-0">
      <template #right>
        <!-- Button "Kết nối nền tảng" -->
        <button @click="toggleDropdown">
          <PlusCircleIcon />
          {{ $t('Kết nối nền tảng') }}
          <ChevronDownIcon />
        </button>

        <!-- Button "Chat nhiều trang" -->
        <button @click="toggleModelGroupPage()">
          <SquaresPlusIcon />
          {{ $t('Gom nhóm trang') }}
        </button>

        <!-- Button Nạp tiền (chỉ hiện trong /org) -->
        <ReChargeBtn v-if="$route.path.includes('/dashboard/org/')" />
      </template>
    </header>

    <!-- Content area -->
    <div class="overflow-hidden h-full">
      <RouterView />
    </div>

    <!-- Modals -->
    <DropdownPickConnectPlatform ref="..." />
    <ConnectPage
      ref="..."
      @done="reloadPageData()"
    />
  </div>
</template>
```

Logic chính:

```
class Main {
  /** Vào chế độ chat nhiều trang */
  toggleModelGroupPage() {
    // Reset danh sách trang đã chọn nếu đang ở chế độ nhiều tổ chức
    if (orgStore.is_selected_all_org) pageStore.selected_page_id_list = {}

    // Toggle chế độ
    selectPageStore.toggleGroupPageMode()
  }

  /** Ẩn hiện modal kết nối nền tảng */
  toggleModalConnectPage(key?: string) {
    pageManagerStore.connect_page_ref?.toggleModal?.(key)
  }

  /** Có hiển thị các nút của trang chọn page không */
  isShowSelectPageButton() {
    return (
      $route.path.includes('select-page') &&
      (!selectPageStore.is_group_page_mode || !size(pageStore.active_page_list))
    )
  }
}
```

4.2 Header.vue

Cấu trúc:

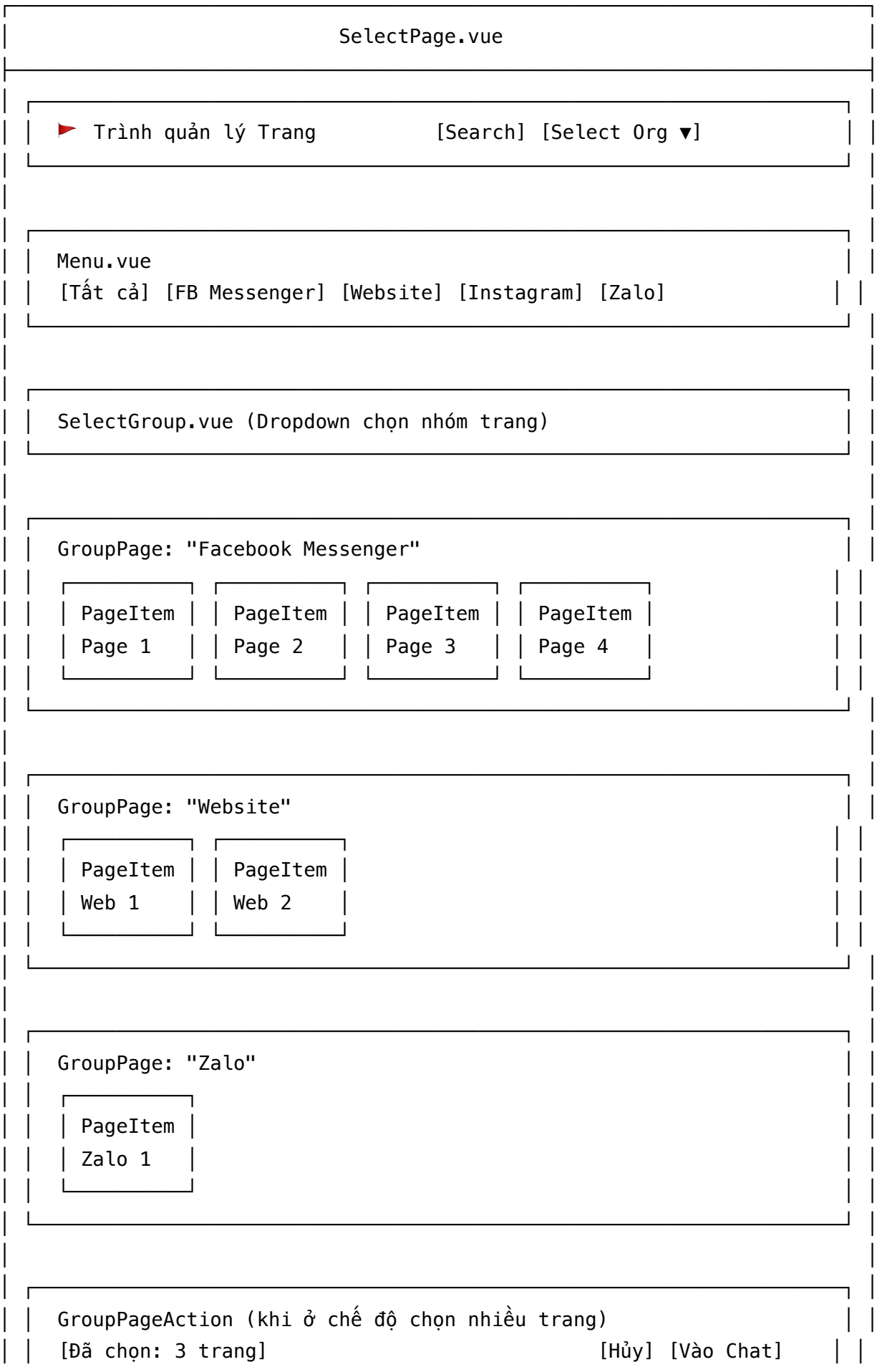
```vue
<template>
  <header class="flex justify-between items-center">
    <!-- Left: Logo/Back button -->
    <div class="flex items-center gap-3">
      <RouterLink to="/chat">
        <Logo />
      </RouterLink>
      <div class="text-xl font-semibold">{{ pageTitle }}</div>
    </div>

    <!-- Right: Slot cho buttons -->
    <div class="flex items-center gap-2">
      <slot name="right" />
    </div>
  </header>
</template>
```

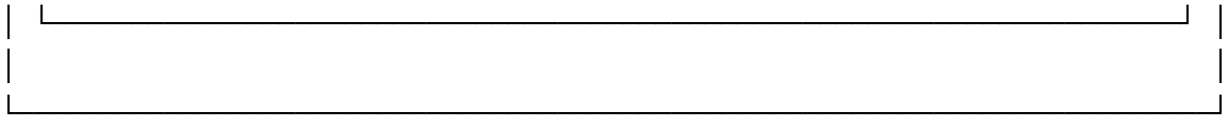
5. Select Page Module

5.1 SelectPage.vue

Mục đích: Màn hình quản lý tất cả các trang đã kết nối

UI Layout:


SelectPage.vue
Trình quản lý Trang
[Search] [Select Org ▼]
Menu.vue
[Tất cả] [FB Messenger] [Website] [Instagram] [Zalo]
SelectGroup.vue (Dropdown chọn nhóm trang)
GroupPage: "Facebook Messenger"
PageItem Page 1
PageItem Page 2
PageItem Page 3
PageItem Page 4
GroupPage: "Website"
PageItem Web 1
PageItem Web 2
GroupPage: "Zalo"
PageItem Zalo 1
GroupPageAction (khi ở chế độ chọn nhiều trang)
[Đã chọn: 3 trang]
[Hủy] [Vào Chat]



Lifecycle:

```
onMounted(async () => {
  // Load lại info của chatbot user
  getMeChatbotUser?.()

  // Kích hoạt tự động mở kết nối nền tảng nếu URL có ?connect_page=
  triggerConnectPlatform()

  // Lấy toàn bộ dữ liệu tổ chức và trang
  await getALlOrgAndPage()

  // Xử lý đăng nhập không có page
  handleLoginWithoutPage()
})
```

5.2 Menu.vue

Mục đích: Tabs lọc theo nền tảng

Tabs config:

Tab	Value	Icon
Tất cả nền tảng	`ALL_PLATFORM`	-
Facebook Messenger	`FB_MESS`	FacebookIcon
Website	`WEBSITE`	WebIcon
Instagram	`FB_INSTAGRAM`	InstagramIcon
Zalo	`ZALO`	ZaloIcon

5.3 GroupPage.vue

Mục đích: Hiển thị nhóm trang theo nền tảng

Props:

```
const $props = withDefaults(
  defineProps<{
    title: string // Tiêu đề nền tảng
    icon: Component // Icon component
    filter: string // Filter key (FB_MESS, WEBSITE, etc.)
    tab?: string // Tab mở modal khi empty
    advancedFilter?: (page: PageData) => boolean // Lọc nâng cao
  }>(),
  {}
)
```

Logic chính:

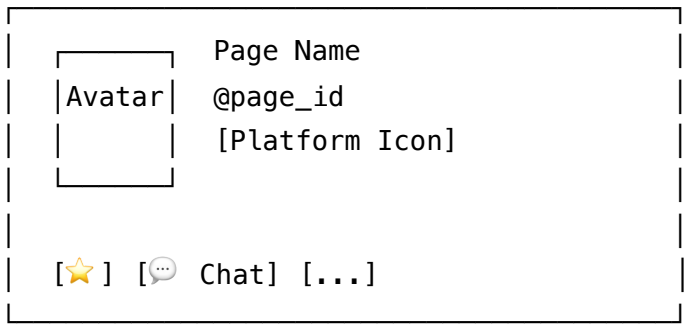
```
/** Danh sách page sau khi lọc */
const active_page_list = ref<PageData[]>()

/** Computed: toàn bộ page của group này được chọn? */
const is_select_all_page = computed({
  get() {
    let count_selected_page = 0
    let total_page_of_group = 0

    loopPageOfGroup(page => {
      total_page_of_group++
      if (pageStore.isSelectedPage(page?.fb_page_id)) count_selected_page++
    })

    if (!total_page_of_group) return false
    return count_selected_page === total_page_of_group
  },
  set(newValue) {
    loopPageOfGroup(page => {
      pageStore.setPageSelected(page?.fb_page_id, newValue)
    })
  },
})

/** Sắp xếp page gắn sao lên đầu */
function getListPage() {
  let pages = sortListPage()?.filter(
    page => IS_RECENT || page?.page?.type?.includes($props.filter)
  )

  if ($props.advancedFilter) pages = pages?.filter($props.advancedFilter)

  if (IS_RECENT) pages = pages?.slice(0, 4)

  active_page_list.value = pages
}

/** Vào chế độ gộp trang nhanh */
function quickGroupPage() {
  pageStore.selected_page_id_list = {}

  active_page_list.value?.forEach(page => {
    const PAGE_ID = page?.page?.fb_page_id
```

```
    if (!PAGE_ID) return
    set(pageStore.selected_page_id_list, PAGE_ID, true)
  })

  goToChat()
}
```

5.4 PageItem.vue

Mục đích: Hiển thị 1 trang với avatar, tên và actions

UI:



Actions:

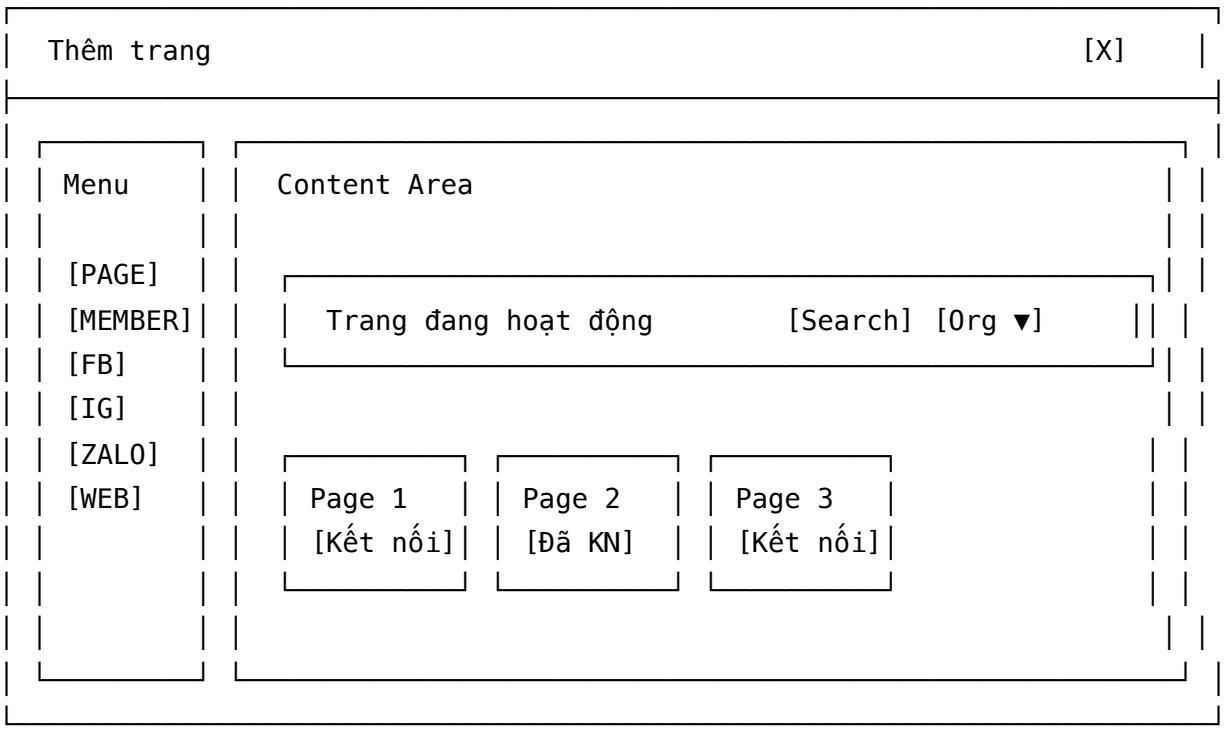
- ⭐ Gắn sao/Bỏ sao
- 💬 Vào Chat
- 🔄 Refresh token

6. Connect Page Module

6.1 ConnectPage.vue (Modal)

Mục đích: Modal kết nối nền tảng mới hoặc cấp lại quyền

UI Structure:



Menu Options:

Key	Component	Description
PAGE	ActivePage.vue	Danh sách trang đã kết nối
MEMBER	Member.vue	Quản lý thành viên tổ chức
FB_MESS	Facebook.vue	Kết nối Facebook Messenger
FB_INSTAGRAM	Instagram.vue	Kết nối Instagram
ZALO	Zalo.vue	Kết nối Zalo OA
WEBSITE	Website.vue	Tạo Website chat widget

Logic:

```
/** Ẩn hiện modal */
function toggleModal(key?: string) {
  // Tự động chọn menu
  if (key) connectPageStore.selectMenu(key)

  // Toggle modal
  modal_connect_page_ref.value?.toggleModal()
}

class Main {
  /** Reset store khi đóng modal */
  resetStore() {
    connectPageStore.is_hidden_menu = false
  }
}

// Expose cho parent component
defineExpose({ toggleModal })

// Provide cho children
provide(KEY_TOGGLE_MODAL_FUNCT, toggleModal)
```

6.2 Platform Connect Components

Facebook.vue

```
// Sử dụng Facebook SDK để lấy access token
// → Gọi API kết nối trang
// → Reload danh sách trang
```

Zalo.vue

```
// Redirect đến Zalo OAuth
// → Callback sau khi user authorize
// → Lưu access token & kết nối
```

Website.vue

```
// Form tạo website widget
// - Tên website
// - Domain (whitelist)
// - Cấu hình widget
// → API tạo page type WEBSITE
```

7. Organization Settings

7.1 Structure

```
/dashboard/org
├── /setting          → Cài đặt tổ chức
├── /pay
│   ├── /info         → Thông tin gói & quota
│   └── /recharge     → Nạp tiền
├── /app              → Quản lý ứng dụng
├── /api              → API Keys
├── /virtual-assistant → Trợ lý ảo AI
└── /webhook          → Webhook endpoints
```

7.2 Personal Settings

File: `Setting/PersonalSetting.vue`

Các tùy chọn:

Setting	Type	Description
`is_enable_personal_setting`	boolean	Bật ghi đè setting của page
`is_hide_page_avatar`	boolean	Ẩn avatar trang
`display_label_type`	enum	Kiểu hiển thị nhãn

Display Label Types:

- `FULL` : Hiển thị đầy đủ text + màu

- `ICON` : Chỉ hiển thị dot màu
- `ICON_TOOLTIP` : Dot màu + tooltip

Storage:

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

8. Widget Management

8.1 Widget Routes

Path	Component	Description
`/dashboard/widget/market`	`AllWidget.vue`	Chợ widget
`/dashboard/widget/installed`	`InstalledWidget.vue`	Đã cài đặt
`/dashboard/widget/my-widget`	`MyWidget.vue`	Widget của tôi

8.2 Widget Types

- **System Widgets:** Widget mặc định của hệ thống
- **Third-party Widgets:** Widget từ nhà phát triển bên thứ 3
- **Custom Widgets:** Widget do user tự tạo

9. Stores & State Management

9.1 Related Stores

Store	File	Purpose
`usePageStore`	`stores/page.ts`	Quản lý danh sách trang
`useSelectPageStore`	`stores/selectPage.ts`	State của Select Page

Store	File	Purpose
`useConnectPageStore`	`stores/connectPage.ts`	State của Connect Page modal
`useOrgStore`	`stores/org.ts`	Thông tin tổ chức
`usePageManagerStore`	`stores/pageManager.ts`	Refs & state quản lý

9.2 usePageStore

```
// Các state chính
{
  all_page_list: [],           // Tất cả trang
  active_page_list: [],        // Trang đang hoạt động
  selected_page_id_list: {},   // Map trang đã chọn (multi-select)
  map_orgs: {},                // Map trang theo tổ chức
}

// Methods
isSelectedPage(pageId)         // Check trang được chọn chưa
setPageSelected(pageId, value) // Chọn/bỏ chọn trang
countActivePage()              // Đếm số trang active
```

9.3 useSelectPageStore

```
{
  current_menu: 'ALL_PLATFORM',  // Tab đang chọn
  search: '',                    // Tìm kiếm
  is_loading: false,             // Loading state
  is_group_page_mode: false,     // Chế độ chọn nhóm trang
}

// Methods
toggleGroupPageMode()            // Toggle chế độ
selectMenu(key)                  // Chọn tab
```

10. Composables

10.1 usePageManager.ts

File: `composables/usePageManager.ts`

```
export function usePageManager() {
  const pageStore = usePageStore()
  const selectPageStore = useSelectPageStore()
  const pageManagerStore = usePageManagerStore()

  /** Toggle dropdown kết nối nền tảng */
  function toggleDropdown() {
    pageManagerStore.ref_dropdown_pick_connect_platform?.toggleDropdown()
  }

  /** Reload data trang sau khi kết nối */
  async function reloadPageData() {
    await getAllOrgAndPage()
  }

  /** Lấy tất cả org và page */
  async function getALlOrgAndPage() {
    // API call...
  }

  /** Sắp xếp danh sách trang (starred first) */
  function sortListPage() {
    return orderBy(
      pageStore.active_page_list,
      [page => page?.page?.is_starred],
      ['desc']
    )
  }

  /** Điều hướng vào chat */
  function goToChat() {
    $router.push('/chat')
  }

  /** Toggle modal kết nối */
  function toggleModalConnectPage(key?: string) {
    pageManagerStore.connect_page_ref?.toggleModal?.(key)
  }

  return {
    toggleDropdown,
    reloadPageData,
    getALlOrgAndPage,
```

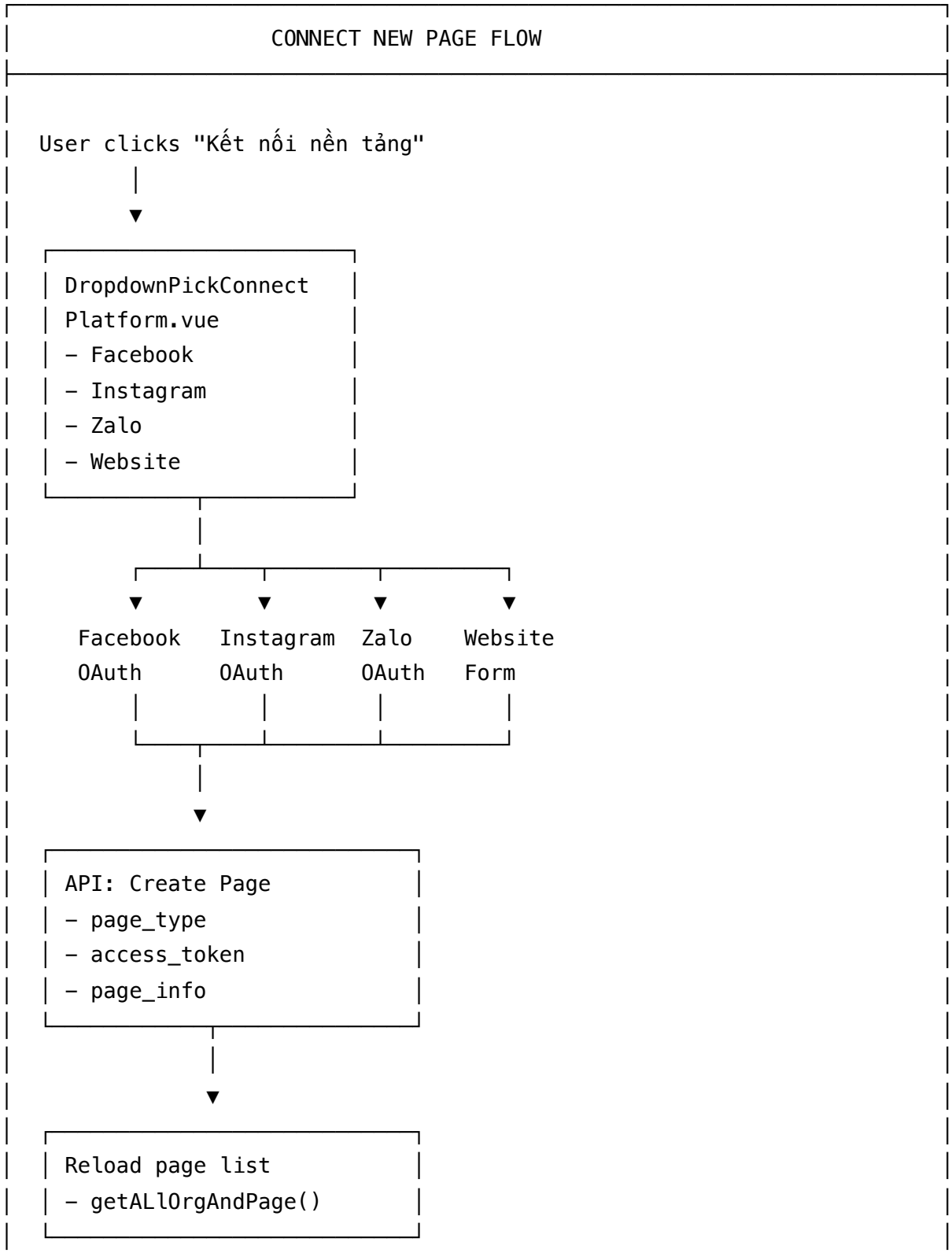
```
    sortListPage,
    goToChat,
    toggleModalConnectPage,
  }
}
```

10.2 symbol.ts (Provide/Inject)

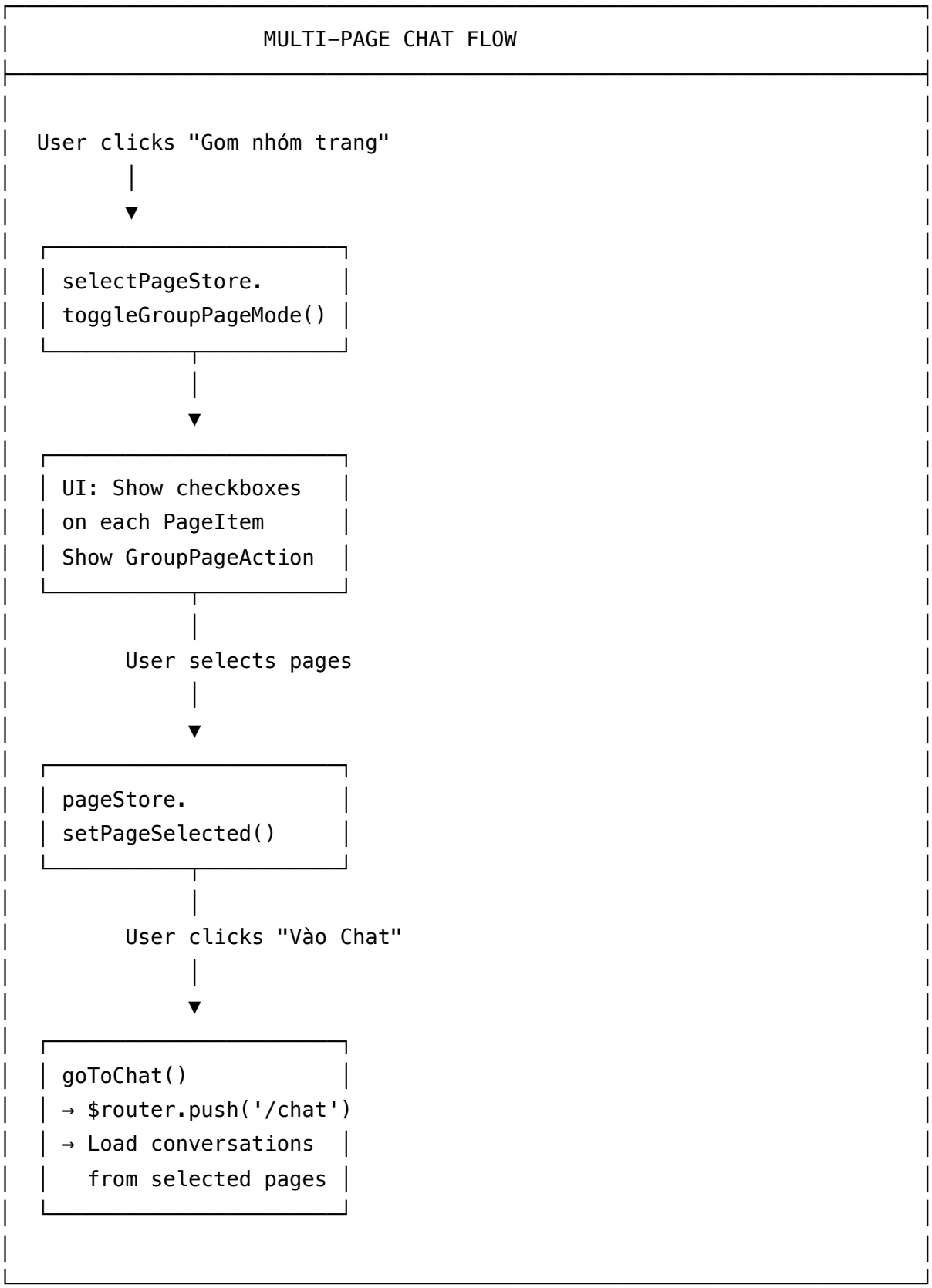
```
// Dashboard level
export const KEY_GET_CHATBOT_USER_FUNCT = Symbol('getMeChatbotUser')

// SelectPage level
export const KEY_ADVANCE_SELECT_AGE_FUNCT = Symbol('advanceSelectPage')

// ConnectPage level
export const KEY_TOGGLE_MODAL_FUNCT = Symbol('toggleModal')
```

Flow Diagrams

Kết nối trang mới



Vào Chat nhiều trang



📎 Related Files

- **Layout:** `src/views/Dashboard.vue`
- **Composables:** `src/views/Dashboard/composables/`
- **Stores:** `src/stores/page.ts` , `src/stores/org.ts`
- **APIs:** `src/utils/api/N4Service/Page.ts`

> **Note:** Dashboard là trung tâm quản lý chính, kết nối với nhiều module khác như Chat, Settings, Widget thông qua Pinia stores và provide/inject pattern.