# MCS 프론트엔드 및 Server Control Launcher 조사 보고서

본 조사는 제공된 저장소 (syyunfksmma/MCS)에서 Server Control Launcher 와 Manufacturing Cam Management System(MCS) 서비스를 구분하고 사용자 지침에 명시된 기능이 프로토타입 UI에서 얼마나 적용되었는지 분석한 결과이다. 문서와 코드를 검토하여 정성적·정량적 지표를 추출하였다.

## 1. Server Control Launcher와 MCS 서비스 구분

| 구분 | 설명/증거 | 사용 근거 |
| --- | --- | --- |
| **목적** | **Server Control Launcher:** 요구사항 문서는 서버 관리 서비스와 MCS 서비스를 동시에 제어하는 **데스크톱 앱**으로 설명하고 있다. 초기 버전은 Windows 기반으로, ‘Start All’ 버튼을 통해 API 2종과 Web UI 1종을 순차적으로 기동‧중지하고 로그를 실시간으로 표시한다[[1]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Requirements.md#L37-L54). | Launcher는 MCS 서비스 자체가 아니라 **서비스 기동 도구**로 설계되었음. |
|  | **MCS 서비스:** PRD 문서와 구현 계획에서 MCS는 제조 CAM 작업을 위한 **웹 서비스**로, 파일 트리·Routing 관리·검색 등 기능을 제공한다[[2]](https://github.com/syyunfksmma/MCS/blob/main/docs/PRD_MCS.md#L114-L151). ExplorerShell.tsx 파일은 Routing 목록과 그룹을 트리 형태로 보여주고, 상세 보기·파일 다운로드·검색 등을 담당한다. | MCS는 사용자에게 CAM 데이터를 관리·검색하게 하는 **주요 프론트엔드 서비스**임. |
| **UI 구성** | **Launcher:** GUI 디자인 문서는 헤더 영역, 서비스 카드 영역(3개 서비스 상태 표시), 로그 패널, 우측 설정 패널을 포함하는 20/52/28 레이아웃을 제시한다[[3]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Final_GUI_Design.md#L20-L28). Start All/Stop All 버튼과 로그 필터링 등 제어 인터랙션이 정의되어 있다[[4]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Final_GUI_Design.md#L29-L34). | 런처 UI는 서비스 기동·모니터링을 목적으로 하는 관리 패널. |
|  | **MCS UI:** MCS 프론트엔드는 탐색 트리, 리본 메뉴, 검색·필터 레일, 퀵 메뉴, 파일 업로드 패널 등 **대규모 작업 관리 UI**로 구성되어 있다. ExplorerShell 컴포넌트는 데이터를 로드한 뒤 좌측 트리, 중앙 요약/탭, 우측 패널을 배치하고 Ribbon·Hover Menu·Search Filter Rail 등의 컴포넌트를 렌더링한다[[5]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L1961-L1968). | MCS UI는 CAM Routing 데이터를 관리·검색하는 **업무용 인터페이스**임. |
| **내부 서비스 구성** | Launcher 요구사항에 따르면 “Server Management Service”와 “MCS Service Launcher” 두 섹션이 있으며, MCS 관련 API 2종과 Web UI 1종을 실행/중단할 수 있도록 설계되어 있다[[6]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Requirements.md#L44-L54). | MCS 서비스는 런처에서 **하위 프로세스**로 기동된다. |

**요약:** Server Control Launcher는 데스크톱 환경에서 API와 Web UI 프로세스를 시작/중지하고 로그를 실시간 스트리밍하는 **운영 도구**이다. MCS 서비스는 CAM Routing 데이터의 생성‧검색‧승인 등의 업무를 수행하는 **웹 기반 애플리케이션**으로 런처와 별도의 코드베이스에서 개발된다. 런처는 MCS를 포함한 여러 서비스를 기동하며 UI 구성과 기능이 전혀 다르다.

## 2. 사용자 지침 별 UI 기능 구현 현황

### 2.1 프론트엔드 프로토타입 개발 (Teamcenter 패턴 적용)

| 지침 | 실제 구현 | 적용 여부 |
| --- | --- | --- |
| **Teamcenter 패턴 준수 및 React 프론트엔드** | Explorer UI는 Teamcenter X 패턴을 참고한 3‑패널 레이아웃(트리/메인/프리뷰)을 사용한다. ExplorerShell은 Ribbon 컨트롤, 요약 카드, 트리 패널 등으로 구성되어 사용자의 작업 맥락을 제공한다[[5]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L1961-L1968). 전체 앱은 React 17/Ant Design 4를 기반으로 작성되어 있다. | **부분 적용** – Teamcenter 레이아웃/토큰을 차용하였으나 메뉴바·Ribbon 내용은 일부 지침과 다름. |
| **파일 관리, 사용자 Presence, 실시간 갱신을 Supabase로 구현** | 코드 전반에서 Supabase 연결은 발견되지 않았다. 구현 계획 문서는 **ASP.NET Core 8 + SignalR + SQL Server**를 사용해 다중 사용자 편집, 파일 트리 갱신, Presence를 처리하도록 변경하였다[[7]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/CAM_WebService_ImplementationPlan.md#L44-L57). 프론트엔드에서는 SignalR Hub를 통해 실시간 업데이트를 수행할 준비가 되어 있으나 Supabase는 사용되지 않는다. | **미적용** – Supabase 기반 시뮬레이션 대신 SignalR Hub/ASP.NET 코어로 구현 계획이 수정되었다. |
| **기업 표준 UI 컴포넌트 & Siemens 브랜딩 적용** | UI는 Ant Design 컴포넌트를 토대로 하며, 모듈 스타일(module.css)을 통해 색상과 버튼 모양을 커스터마이징하였다. ExplorerRibbon의 버튼과 그룹 라벨은 Teamcenter UX에서 사용하는 Ribbon 패턴과 유사한 색상·배지 스타일을 적용한다[[8]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerRibbon.tsx#L32-L113). 다만 Siemens 브랜딩 지침(폰트, 색상 팔레트)의 적용 여부를 직접 확인할 수 있는 CSS 토큰 정의는 저장소에 없었다. | **부분 적용** – Ant Design 기반 UI에 Teamcenter 느낌을 접목했지만 Siemens 브랜딩 표준을 완전히 준수하는지 확인할 수 없음. |

### 2.2 메뉴바 구성 기능 확인

| 메뉴/요구 기능 | 코드/문서 조사 결과 | 구현 상태 |
| --- | --- | --- |
| **Dashboard** (금일 미할당/할당/완료 현황, 인원별/설비별 작업현황 그래프, 주간/연간/월간 프로그램 작성 및 불량률 KPI) | 프론트엔드 저장소에서 Dashboard 페이지나 관련 그래프 컴포넌트는 발견되지 않았다. API 계약 문서(products\_dashboard\_v2.md)는 제품 대시보드의 JSON 응답 정의만 포함한다. | **미구현** – 대시보드 UI와 KPI 그래프 구현 흔적 없음. |
| **MCS 메뉴 (3D 모델과 CAM 파일 관리, Esprit EDGE API 키 생성/전달, Esprit EDGE 기동)** | Explorer UI는 Routing 코드를 생성·관리하고 파일 트리 및 업로드 패널을 제공한다. MCS에서 3D 모델 뷰어, Esprit EDGE API 키 생성·전달 또는 Esprit EDGE 기동 기능은 코드에서 찾을 수 없었다. | **일부 구현** – Routing 관리와 파일 업로드는 구현되어 있으나 3D 뷰어 및 Esprit EDGE 연동 기능은 미구현. |
| **Server 메뉴 (정해진 폴더 구조에 따른 파일 구조, 검색·로그·REV 관리, STL/SolidWorks 모델 뷰어)** | 저장소에는 Server 메뉴 UI가 존재하지 않았다. Server Control Launcher 문서는 데스크톱 앱에서 개별 API/Web UI 프로세스를 기동하는 기능을 정의하지만, 웹 페이지에서 STL/3D 뷰어를 제공하는 컴포넌트는 없었다. | **미구현** |
| **Option 메뉴 (폴더 구조 설정, 작업 할당 관리, Access Data 소스 매핑, 서버 드라이브 관리, 사용자 관리, Esprit EDGE 버전 관리, 트래픽/서비스 관리)** | 관련 UI 코드가 존재하지 않으며, 옵션 페이지 설계나 구현 계획 문서도 발견되지 않았다. | **미구현** |

### 2.3 MCS 좌측 행렬 리스트 및 필터 레일 (Sprint6 E1)

| 항목 | 실제 구현 | 정량 지표/평가 |
| --- | --- | --- |
| **좌측 행렬 리스트** | TreePanel 컴포넌트는 제품 → 리비전 → Routing 그룹 → Routing의 4단계 트리 구조를 렌더링한다. 사용자는 그룹의 순서를 드래그하여 재정렬하거나 그룹명을 변경하고 삭제 상태 토글을 할 수 있으며, 라우팅을 선택하면 세부 정보 탭과 다운로드 기능이 활성화된다[[9]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L1939-L1954). | **완료** – 트리 구조 및 행렬 리스트 구현. |
| **Filter Rail 정보 구조 & ExplorerShell 배선** | SearchFilterRail.tsx는 제품·그룹·상태 필터를 제공한다. 사용자 선택 값과 이벤트 핸들러를 props로 받아 ExplorerShell과 상태를 동기화하며, 모든 필터를 초기화하는 Reset 버튼을 제공한다[[10]](https://github.com/syyunfksmma/MCS/blob/57a4699ce4e3819b394d46969f24eb9f9048efe5/web/mcs-portal/src/components/explorer/SearchFilterRail.tsx#L37-L92). | **완료** – 제품/그룹/상태 3종 필터 및 리셋 동작 구현. |
| **퀵 액션 토글: 최근 본 Routing, SLA 초과만 보기** | 코드에는 최근 항목/ SLA 초과 토글이 없다. 대신 ExplorerShell의 searchResult는 검색 SLA 타겟, 서버 SLA ms, 클라이언트 관측 ms, 결과 수 등을 계산하여 요약 카드에 표시한다[[11]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2033-L2057). | **미구현** – 필터 레일에 “최근 Routing” 또는 “SLA 초과만 보기” 토글이 없으며 검색 SLA 요약만 제공. |

### 2.4 Ribbon 액션 그룹화 (Sprint6 E2)

| 요구 | 구현 | 평가 |
| --- | --- | --- |
| **세 그룹 (Routing: New/Duplicate/Archive; Approval: Request/Approve/Reject; Add‑in: Queue Job/Retry/Cancel)** | ExplorerRibbon 컴포넌트는 세 개의 그룹을 렌더링하지만, 실제 버튼은 “선택한 Routing 열기/새 Routing”, “Workspace 업로드/다운로드”, “Add‑in 콘솔”만 제공한다[[12]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerRibbon.tsx#L32-L100). 승인 요청·승인/거절, 중복 복제, Archive, Add‑in 작업/Retry/Cancel 기능은 없다. | **부분 구현** – 그룹화 UI는 존재하지만 버튼 항목과 기능이 요구사항과 상이함. 승인과 중복/Archive 등의 작업은 향후 개발 필요. |

### 2.5 Hover Quick Menu (Sprint6 E3 / Sprint7 E3)

| 요구 사항 | 구현 | 정량 지표 |
| --- | --- | --- |
| **Tree Panel 노드 Hover 시 Quick Menu: View Detail, Open Uploads, Approve** | ExplorerHoverMenu.tsx는 마우스 hover 후 0.2 초 지연으로 표시되며, 상세보기(View Detail), 업로드 패널 열기(Open Uploads), 승인(Approve), Pin/Unpin 메뉴를 제공한다. useHoverMenu 훅에서 openDelay 200ms/closeDelay 150ms로 지연과 자동 닫힘을 구현했고, 키보드 접근성을 위해 ESC 키 이벤트와 aria 속성을 설정하였다[[13]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx#L87-L143)[[14]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/hooks/useHoverMenu.ts#L32-L39). | **완료** – 지연시간, 접근성, 승인 버튼 Disabled 처리 등 UI/UX 요건 충족. |
| **SLA 배지 및 Add‑in 상태 배지 표시** | Hover Menu는 Routing 코드, 상태, Pinned 상태, Add‑in Job Status를 배지로 표기하고 SLA 초과 시 ClockCircleOutlined 아이콘과 SLA 지연(ms)을 표시한다[[15]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx#L87-L102). | **완료** – SLA 지연 표시 및 배지 표시 구현. |
| **Keyboard 접근성** | ESC 키를 누르면 메뉴가 닫히며, 메뉴 항목은 role="menuitem"으로 지정되어 키보드 포커스를 받을 수 있다[[16]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx#L79-L166). | **완료**. |

### 2.6 Search & SLA 지표

ExplorerShell의 검색 기능은 React Query 뮤테이션으로 API 호출을 수행하며, 사용자가 검색어를 입력하면 실시간 검색(2자 이상 입력 시 350 ms 후 자동 조회)과 명시적 엔터 검색을 지원한다. 검색 결과에는 productCode·groupName·status 필터가 적용되며, 최근 검색 SLA 지표를 요약 카드로 제공한다.

**정량 지표 예시 (UI 상태)**

* **SLA 타깃:** 1500 ms (문서에서 지정)[[11]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2033-L2057)
* **서버 SLA 값:** searchResult.slaMs – API가 보고한 응답 시간. 샘플 데이터를 로드한 경우엔 0 ms로 표시된다[[11]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2033-L2057).
* **클라이언트 관측 값:** searchResult.observedClientMs – 브라우저 측 측정 시간. 예시에서는 0 ms 또는 실제 측정값으로 표시된다[[11]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2033-L2057).
* **결과 수:** filteredItems.length – 필터를 통과한 검색 결과의 건수[[17]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2059-L2064).
* **아이템 수:** Explorer 요약 카드에서 itemsState.length로 표시되며, 로드된 Routing 데이터의 총 수량을 의미한다[[5]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L1961-L1968).

이러한 지표는 모의 데이터에 기반하므로 실제 환경에서는 API 응답 시간을 측정해 SLA 준수 여부를 판단해야 한다.

## 3. 종합 평가 및 개선 제안

1. **Server Control Launcher와 MCS의 역할 분리:**
2. 런처는 데스크톱 관리 도구로, API/Web UI 프로세스를 기동·중단하고 로그를 모니터링한다. MCS 프론트엔드는 CAM Routing 데이터를 관리하는 웹 서비스이다. 문서와 코드에서 두 컴포넌트가 명확히 분리돼 있으며 기능상 중복이 없다[[1]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Requirements.md#L37-L54).
3. **프론트엔드 적용 상황:**
4. 팀센터 패턴과 리본 UX를 참조하여 기본 레이아웃과 일부 UI 컴포넌트를 구현했지만, **메뉴바(Dashboard/Server/Option)** 및 **Esprit EDGE 연동** 등 요구사항의 주요 기능은 전혀 구현되지 않았다. 향후 이 메뉴들의 설계·개발이 필요하다.
5. 필터 레일, Hover Quick Menu는 상세 요구사항을 대부분 충족하였으나, **최근 Routing/ SLA 초과만 보기** 토글과 **Ribbon의 승인/중복/Archive/Cancel 버튼**은 누락되었다. 다음 스프린트에서 기능을 추가해야 한다.
6. Supabase 대신 ASP.NET Core + SignalR Hub로 실시간 업데이트를 처리하려는 아키텍처 변경이 문서에서 확인되었으므로, 지침의 표현을 최신 계획으로 업데이트할 필요가 있다[[7]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/CAM_WebService_ImplementationPlan.md#L44-L57).
7. **정량 지표:**
8. 검색 SLA 요약 및 데이터 개수 등은 UI에서 실시간으로 확인할 수 있다. 실제 API 통합 시 SLA 목표(1500 ms) 대비 서버/클라이언트 응답 시간을 기록해 KPI 로 활용해야 한다[[11]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2033-L2057).
9. 현재 Sprint 7 Task List는 필터 레일, 리본, Hover Quick Menu 구현을 **모두 완료된 것으로 표시**하고 있어 일정상 Sprint 6/7 목표는 달성되었다[[18]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/sprint/Sprint7_TaskList.md#L77-L79).

### 2025-09-30 업데이트

Sprint 8은 Dashboard KPI 구축 및 SignalR 실시간 Presence 구현, Sprint 9는 Dashboard/MCS/Server/Option 메뉴 확장, Sprint 10은 Ribbon/Filter 토글 및 브랜딩 정합성 확보로 재편되었다.

Option/Server/Esprit API 계약 워크숍을 2025-10-02~10-09에 4회 진행하며, 산출물로 계약 문서와 회의 로그를 생성한다.

SearchFilterRail 토글 UI는 구현되어 있으나 서버 연동이 남아있으므로 Sprint 10 Task List에 동기화 작업을 명시했다.

## 4. 결론

* **Server Control Launcher**와 **MCS 서비스**는 설계 목표와 UI 구성이 크게 다르며, 서로 다른 배포 대상(데스크톱 vs 웹)임을 확인하였다.
* MCS 프론트엔드는 Teamcenter UX를 참조한 기본 뷰와 핵심 상호작용(검색, 트리, Hover Menu)을 구현하여 스프린트 목표를 충족하였다.
* 그러나 사용자 요구사항에 명시된 **대시보드**, **Server/Option 메뉴**, **Esprit EDGE 연동**, **Supabase 기반 실시간 기능** 등은 아직 구현되지 않았다. 또한 Ribbon 구성 일부와 필터 레일의 퀵 액션 기능이 부족한 상태다.
* 차후 스프린트에서는 미구현 기능을 계획하고, 문서와 코드에서 브랜딩 규격 및 최신 아키텍처 변경사항(SignalR Hub)을 반영해 사용자 요구사항과 일치하도록 보완해야 한다.

[[1]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Requirements.md#L37-L54) [[6]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Requirements.md#L44-L54) Server\_Control\_Launcher\_Requirements.md

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Requirements.md>

[[2]](https://github.com/syyunfksmma/MCS/blob/main/docs/PRD_MCS.md#L114-L151) PRD\_MCS.md

<https://github.com/syyunfksmma/MCS/blob/main/docs/PRD_MCS.md>

[[3]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Final_GUI_Design.md#L20-L28) [[4]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Final_GUI_Design.md#L29-L34) Server\_Control\_Launcher\_Final\_GUI\_Design.md

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/design/Server_Control_Launcher_Final_GUI_Design.md>

[[5]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L1961-L1968) [[9]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L1939-L1954) [[11]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2033-L2057) [[17]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx#L2059-L2064) ExplorerShell.tsx

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerShell.tsx>

[[7]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/CAM_WebService_ImplementationPlan.md#L44-L57) CAM\_WebService\_ImplementationPlan.md

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/CAM_WebService_ImplementationPlan.md>

[[8]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerRibbon.tsx#L32-L113) [[12]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerRibbon.tsx#L32-L100) ExplorerRibbon.tsx

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerRibbon.tsx>

[[10]](https://github.com/syyunfksmma/MCS/blob/57a4699ce4e3819b394d46969f24eb9f9048efe5/web/mcs-portal/src/components/explorer/SearchFilterRail.tsx#L37-L92) SearchFilterRail.tsx

<https://github.com/syyunfksmma/MCS/blob/57a4699ce4e3819b394d46969f24eb9f9048efe5/web/mcs-portal/src/components/explorer/SearchFilterRail.tsx>

[[13]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx#L87-L143) [[15]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx#L87-L102) [[16]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx#L79-L166) ExplorerHoverMenu.tsx

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/components/explorer/ExplorerHoverMenu.tsx>

[[14]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/hooks/useHoverMenu.ts#L32-L39) useHoverMenu.ts

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/web/mcs-portal/src/hooks/useHoverMenu.ts>

[[18]](https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/sprint/Sprint7_TaskList.md#L77-L79) Sprint7\_TaskList.md

<https://github.com/syyunfksmma/MCS/blob/a889431a612f3e425d1eba92af1b7c72c5623708/docs/sprint/Sprint7_TaskList.md>