

요구사항ID	요구사항명	요구사항 상세내역	상태	출처	비고
REQ-UI-001	업무 화면 로딩 프레임 - Tab View	<p>■ Tab View Layout 동작 방식</p> <ul style="list-style-type: none"> - 여러 개 화면을 오픈할 수 있지만, 한 번에 1개의 화면만을 볼 수 있음 - Tab View Layout 하단에 화면을 선택할 수 있는 탭(프로그램명 표시)이 화면 오픈 개수만큼 생성됨 - 오픈하려는 화면이 기존에 열려 있는 경우, 해당 화면으로 포커스가 이동함(새로고침은 하지 않음) - Tab View 탭 영역의 개수가 화면에 표시될 수 있는 개수를 초과하는 경우에는 탭 영역 이동(좌, 우)버튼을 통해서 이 버튼을 클릭해서 탭 영역을 이동함 <p>■ Tab View Layout 공통 버튼</p> <p>(1) 개별 Window 영역</p> <ul style="list-style-type: none"> - 화면 닫기 버튼 : 해당 화면을 닫음 <p>(2) 전체 Window 컨트롤 영역</p> <ul style="list-style-type: none"> - 탭 영역 이동 버튼 : 열린 탭의 개수가 전체 Tab 컨트롤 영역을 초과할 경우 탭 영역을 좌우로 이동시킴 - 전체 화면 닫기 버튼 : 열려 있는 전체 화면을 닫음 - 전체 오픈된 화면 목록 조회 버튼 : 리스트 형태로 전체 오픈된 화면 목록을 표시하고 선택하는 기능 제공 			
REQ-UI-002	업무 화면 로딩 프레임 - Multi View	<p>■ Multi View Layout 동작 방식</p> <ul style="list-style-type: none"> - 여러 개 화면을 오픈하며, 여러 개의 화면을 분할해서 볼 수 있는 기능을 제공함 - Multi View Layout 하단에 화면을 선택할 수 있는 탭(프로그램명 표시)이 화면 오픈 개수만큼 생성됨 - 오픈하려는 화면이 기존에 열려 있는 경우, 해당 화면으로 포커스가 이동함(새로고침은 하지 않음) - Multi View 탭 영역의 개수가 화면에 표시될 수 있는 개수를 초과하는 경우에는 탭 영역 이동(좌, 우)버튼이 표시되고 이 버튼을 클릭해서 탭 영역을 이동함 <p>■ Multi View Layout 공통 버튼</p> <p>(1) 개별 Window 영역</p> <ul style="list-style-type: none"> - 화면 닫기 버튼 : 해당 화면을 닫음 <p>(2) 전체 Window 컨트롤 영역</p> <ul style="list-style-type: none"> - 탭 영역 이동 버튼 : 열린 탭의 개수가 전체 Tab 컨트롤 영역을 초과할 경우 탭 영역을 좌우로 이동시킴 - 화면 분할 버튼 : 좌우, 상하, 계단식, 비독만 형태로 열린 화면을 분할해서 보여줌 - 전체 화면 닫기 버튼 : 열려 있는 전체 화면을 닫음 			
REQ-UI-003	업무 화면 로딩 프레임 - Single View	<p>■ Single View Layout 동작 방식</p> <ul style="list-style-type: none"> - 한 번에 1개의 화면만 열림 - 메뉴를 클릭해서 새로운 화면을 열면 기존 화면은 닫히고, 새로운 화면이 열림 			
REQ-UI-004	전체 메뉴 표시 & 검색 기능	<p>■ 전체 메뉴 표시</p> <ul style="list-style-type: none"> - Top 영역 : 사용자에게 권한이 있는 전체 메뉴를 Pull-Down 최상단 Top 영역에 보여줌 - Left 영역 : 사용자에게 권한이 있는 전체 메뉴를 Tree 최상단 Left 영역에 보여줌 <p>■ 메뉴 검색 기능</p> <ul style="list-style-type: none"> - 전체 메뉴를 검색할 수 있는 기능 - 자동완성(AutoComplete) 기능을 제공하는 입력박스로 입력된 키워드가 포함된 메뉴명이 타이핑 시 하단에 표시됨 - 메뉴 검색 후 메뉴를 선택하면, 해당 메뉴에 링크된 프로그램이 열림 <p>■ 메뉴 접기 & 펼치기 & 메뉴 숨김 기능</p> <ul style="list-style-type: none"> - 접기 및 펼치기 버튼 클릭을 통해서 Left 메뉴 영역에 표시된 전체 메뉴를 접거나 펼침 - Left 영역의 전체 메뉴를 숨기거나 보이도록 하는 토글 버튼을 제공 			
REQ-UI-005	즐거찾기 메뉴 표시	<p>■ 즐거찾기 메뉴 표시</p> <ul style="list-style-type: none"> - 즐겨찾기로 등록된 메뉴 목록을 화면에 표시하고, 메뉴 클릭 시 화면을 오픈하는 기능 - 즐겨찾기 메뉴는 브라우저를 종료하고 다시 접속할 때도 데이터가 유지되어야 하므로 DB에 저장되어야 함 - Left 메뉴 영역에 탭을 추가해서 즐겨찾기 메뉴를 표시함 			
REQ-UI-006	최근 열린 메뉴 표시 및 삭제	<p>■ 최근 열린 메뉴 표시 및 삭제</p> <ul style="list-style-type: none"> - 최근에 열린 메뉴 목록을 화면에 표시하고, 메뉴 클릭 시 화면을 오픈하는 기능 - 최근 열린 화면 정보는 DB에 저장되지 않기 때문에 화면 새로고침 시 모두 삭제됨 			
REQ-UI-007	로그인 사용자 환경설정 및 로그아웃 처리	<p>■ 로그인 사용자 정보 표시</p> <ul style="list-style-type: none"> - 현재 로그인 중인 사용자의 정보를 메인 화면 내에 표시함 <p>■ 업무 화면 로딩 프레임 변경</p> <ul style="list-style-type: none"> - 업무 화면 로딩 프레임을 Tab View 또는 Multi View로 설정하는 기능 <p>■ 로그아웃</p> <ul style="list-style-type: none"> - Logout 메뉴를 클릭하면 로그아웃 처리를 하고 로그인 페이지로 이동함 			
REQ-UI-008	업무화면 타이틀 영역 공통 기능	<p>■ 업무화면 타이틀 영역 공통 기능 목록</p> <p>(1) 메뉴 명 표시</p> <ul style="list-style-type: none"> - 현재 오픈된 프로그램의 메뉴 명을 표시함 <p>(2) 메뉴 위치 표시</p> <ul style="list-style-type: none"> - 현재 메뉴 명을 제외한 상위 메뉴 위치 정보를 표시함 <p>(3) 즐겨찾기 등록/삭제 토글 버튼</p> <ul style="list-style-type: none"> - 현재 포커스된 화면을 즐겨찾기로 등록함 <p>(4) 화면 정보 조회 버튼</p> <ul style="list-style-type: none"> - 현재 로딩 중인 업무 화면의 소스 경로를 표시함 <p>(5) 화면 초기화 버튼</p> <ul style="list-style-type: none"> - 업무 화면을 최초 로딩한 상태로 초기화 시킴 			
REQ-UI-009	알림, 확인, 토스트 메시지 창	<p>■ 알림 메시지 창</p> <ul style="list-style-type: none"> - 사용자 정의 메시지 및 [확인] 버튼을 표시함 - [확인] 버튼 클릭 후, 실행해야할 코드가 있다면 Callback 함수로 전달 <p>■ 확인 메시지 창</p> <ul style="list-style-type: none"> - [확인] 버튼 클릭 후, 실행해야할 코드를 Callback 함수를 호출하고 true 값을 전달 - [취소] 버튼 클릭 시에는 Callback 함수를 호출하고 false 값을 전달 <p>■ 토스트 메시지 창</p> <ul style="list-style-type: none"> - 사용자 확인이 반드시 필요하지 않은 메시지는 Toast 메시지 창을 통해서 메시지 보여줌 - 정해진 시간 동안(기본 3초) 메시지를 보여주고 자동으로 사라짐 <p>■ Native Alert과 Confirm 사용 시 주의사항</p> <ul style="list-style-type: none"> - 화면 내에서 사용되는 Alert과 Confirm 창은 Layer Alert과 Confirm을 이용해야 함 - UI 동작 시 불가피하게 동기 방식의 Alert과 Confirm이 필요한 경우에만 한정해서 Native Alert과 Confirm의 사용을 허용함 			
REQ-UI-010	Layer(WFrame) 팝업 창	<p>■ Layer(WFrame) 팝업 창 동작 방식</p> <ul style="list-style-type: none"> - 화면 소스(웹스퀘어 XML)를 팝업 Layer 안에 동적으로 로딩함 - iframe을 사용하지 않고, wframe(scope=true)을 사용하여 동적으로 DOM을 생성하고, 소스 내 Javascript 코드를 실행함 - 팝업 종료 시, DOM과 팝업으로 오픈한 소스 내 Javascript 코드에서 생성한 객체를 삭제함 - 팝업 종료 후, 실행해야할 코드가 있다면 Callback 함수로 전달 - Layer 팝업은 팝업을 오픈한 웹 브라우저 영역을 벗어나는 위치로 이동할 수 없음 - 팝업 오픈 시 전체 프레임 Modal 방식으로 오픈되어 팝업 화면 이외의 다른 UI 영역으로 접근 불가 - 팝업이 오픈 된 상태에서 다른 업무 화면 접근이 필요한 경우 Window 팝업 방식으로 오픈해야 함 <p>■ Layer(WFrame) 팝업 오픈 대상 화면</p> <ul style="list-style-type: none"> - 동일한 도메인 내 시스템의 화면 오픈 시 사용함 - 타 도메인 내 시스템의 화면은 오픈할 수 없음 - 타 도메인 시스템의 화면을 오픈하려고 할 경우 Window 팝업 방식으로 오픈해야 함 			
REQ-UI-011	Window 팝업 창	<p>■ Window 팝업 창 동작 방식</p> <ul style="list-style-type: none"> - 웹 브라우저의 Window Popup(window.open API)을 이용해서 팝업을 오픈함 - Window 팝업은 별도의 웹 브라우저 윈도우로 실행되기 때문에, 팝업을 오픈한 웹 브라우저 영역을 벗어나는 위치로 이동할 수 있음 - 듀얼 모니터를 사용하는 경우 보조 모니터 영역으로 팝업 창 이동이 가능함 - Window 팝업을 오픈한 부모 화면이 종료되어도, Window 팝업이 자동으로 종료되지 않음 <p>■ Window 팝업 오픈 대상 화면</p> <ul style="list-style-type: none"> - 타 도메인 내 시스템의 화면을 오픈 시 사용함 - 동일한 도메인 내 시스템 업무 화면을 보조 모니터로 이동해서 사용하고자 할 때도 사용 가능함 			

REQ-UI-012	시스템 에러 발생 시 메시지 처리	<p>■ Error 메시지 표시</p> <ul style="list-style-type: none"> - 서비스 호출 과정에서 서비스 응답 결과로 에러 코드와 메시지를 전송 시, 자동으로 Error 메시지 팝업 창을 오픈해서 에러 내용을 사용자에게 보여줌 <p>■ Error 메시지 팝업 창</p> <ul style="list-style-type: none"> - 서비스 호출 과정에서 서비스 응답 결과로 에러 코드와 메시지를 전송 시, 에러 내용을 표시하는 용도로 사용함 			
REQ-UI-013	Grid 데이터 조회 - 데이터 Paging 처리 방식	<p>■ Paging 처리 방식 구분</p> <ul style="list-style-type: none"> - 검색 조건과 페이지 번호가 변경될 때마다 매번 서버에서 데이터를 재조회하는 방식 <p>■ 데이터 Paging 처리 동작 순서</p> <ol style="list-style-type: none"> (1) 사용자가 [조회] 버튼을 클릭하거나, 페이지 번호를 클릭한다. (2) 조회할 페이지 번호와 검색 조건을 포함해서 데이터 조회 서비스를 호출한다. (3) 응답 받은 데이터를 데이터 저장소 객체(DataList)에 저장한다. (기존에 저장된 데이터는 삭제됨) (4) 응답 받은 페이지 정보를 이용해서 현재 페이지 번호, 표시할 페이지 수, 전체 페이지 수 표시를 위한 UI 요소의 상태를 업데이트 한다. <p>■ Paging 처리를 위한 정보</p> <ul style="list-style-type: none"> - 서비스 요청 시 : 데이터 검색 조건과 페이지 정보(시작 계시물 번호, 마지막 계시물 번호) - 서비스 응답 시 : 검색 결과 데이터(Row 데이터)와 전체 Row 정보(초조 조회 시에만 전달), 처리 결과(상태코드, 메시지) 			
REQ-UI-014	Grid 데이터 조회 - 데이터 Scrolling 처리 방식	<p>■ Scrolling 처리 방식 구분</p> <ol style="list-style-type: none"> (1) 전체 데이터 로딩 - 검색 조건에 해당하는 전체 데이터를 한 번에 로딩해서 클라이언트(웹 브라우저)에서 처리하는 방식 (2) 데이터 Appending 처리 - ScrollEnd 이벤트가 발생될 때마다 남은 데이터가 있을 경우, 서버에서 데이터를 요청해서 추가하는 방식 <p>■ 전체 데이터 로딩 처리 동작 순서</p> <ol style="list-style-type: none"> (1) 사용자가 검색조건을 입력하고 [조회] 버튼을 클릭한다. (2) 조회 조건에 해당하는 전체 데이터를 모두 서버에서 받아옴 (3) 결과 내 재 검색을 실행해도 서버에서 데이터를 재 요청하지 않고, 클라이언트에서 로딩한 데이터를 필터링해서 보여준다. <p>■ ScrollEnd 이벤트를 이용한 Grid 데이터 Appending 처리 동작 순서</p> <ol style="list-style-type: none"> (1) 사용자가 [조회] 또는 [다음 데이터 가져오기]버튼을 클릭하거나, Grid의 Scroll Bar를 드래그해서 맨 마지막 Row로 포커스를 이동(ScrollEnd Event 발생)시킨다. (2) 조회할 페이지 번호와 검색 조건을 포함해서 데이터 조회 서비스를 호출한다. (3) 응답 받은 데이터를 데이터 저장소 객체(DataList)에 추가(Appending)한다. (기존에 저장된 데이터는 유지됨) 			
REQ-UI-015	Grid 데이터 저장 처리	<p>■ 별도 입력 화면을 통한 데이터 추가/수정 처리</p> <ul style="list-style-type: none"> - 데이터 Paging 처리 방식에서는 데이터의 추가/수정 시 별도의 팝업 창을 통해서 데이터를 입력해야 함 - 데이터 저장에 완료된 후에는 Callback 함수를 통해서 데이터를 재조회해야 함 <p>■ Grid 데이터 편집 기능을 이용한 다 건 데이터 저장 처리</p> <ul style="list-style-type: none"> - Grid 내 Row별 데이터 상태를 보여주기 위한 컬럼이 추가됨 - 삭제하려고 하는 Row의 CheckBox를 선택하면 Row 상태가 삭제(D)로 바뀜 - 신규 추가한 Row를 다시 삭제하면 RowStatus 컬럼이 삽입 후 삭제(V)로 바뀌면서, 화면에서도 해당 Row가 사라지고, 서버에도 데이터를 전송하지 않음 - 신규 Row 추가 시, 포커스된 Row가 있다면 해당 Row 바로 뒤에 추가되고, 없다면 제일 첫번째 Row에 추가됨 <p>■ 데이터 업데이트를 위한 Grid의 컬럼 정의</p> <ol style="list-style-type: none"> (1) 첫번째 컬럼(RowStatus): 각 Row의 상태를 보여줌 <ul style="list-style-type: none"> - RowStatus : 조회(R), 신규(C), 수정(U), 삭제(D), 신규 추가후 삭제(V) (2) 두번째 컬럼(CheckBox) : Grid Row 삭제 처리를 위한 선택용 컬럼 (3) 세번째 이후 컬럼들 : 실제 데이터 컬럼들 			
REQ-UI-016	데이터 유효성 검사(Validation)	<p>■ 데이터 입력 시 입력키 값 제어 항목</p> <ul style="list-style-type: none"> - 펄스케이 컴포넌트 속성의 기능으로 제공하는 입력허용문자(allowChar), 입력제외문자(ignoreChar), 최대입력자리수-글자수기준(maxlength), 최대 입력 자리 수 - 바이트 기준(maxByteLength) 속성 설정을 통해서, 데이터 입력 시 입력키 값을 제어함 - 허용되지 않는 문자와 글자수 초과 입력 자체가 되지 않음 <p>■ 단건(TableLayout) 데이터 유효성 검사</p> <ul style="list-style-type: none"> - Table 형태 레이아웃으로 입력되는 단 건 데이터에 대한 데이터 유효성 검사(필수입력, 최소 자리수, 사용자 정의 함수) - 유효성 검사에 실패한 컬럼이 있다면, 에러 메시지를 띄워주고, 해당 컴포넌트로 포커스를 이동함 <p>■ 다건(Grid) 데이터 유효성 검사</p> <ul style="list-style-type: none"> - Grid 형태 레이아웃으로 입력되는 다건 데이터에 대한 데이터 유효성 검사(필수입력, 최소자리수, 사용자 정의 함수) - 데이터 저장을 위한 서비스 호출 전에 데이터 유효성 검사를 호출해야 함 - 유효성 검사에 실패한 컬럼이 있다면, 에러 메시지를 띄워주고, 해당 Grid의 Cell로 포커스를 이동함 <p>■ 유효성 검사 항목 포커스 이동</p> <ul style="list-style-type: none"> - 클라이언트 사이드 유효성 검사 과정에서 발생한 에러에 대해서는 포커스 기능이 제공되지만, 서버 사이드 유효성 검사 과정에서 발생한 에러에 대해서는 에러 알림 메시지 기능만 제공하고 항목 포커스 이동 기능은 제공하지 않음 			
REQ-UI-017	첨부 파일 업로드 & 다운로드	<p>■ 첨부 파일 업로드 기능 요건</p> <ol style="list-style-type: none"> (1) [추가] 버튼 <ul style="list-style-type: none"> - 파일 탐색창을 통해서 업로드할 첨부 파일을 선택함 - Chrome 및 IE10 이상의 웹 브라우저에서는 Ctrl 키를 누르고 파일을 선택하면 여러 개의 파일을 선택할 수 있음 - 제한된 파일 개수 및 용량을 초과할 경우 경고창을 띄움 (2) [삭제] 체크박스 <ul style="list-style-type: none"> - 삭제하려는 파일 목록의 체크박스를 클릭함 - 아직 서버에 전송되지 않은 신규 업로드 파일을 체크할 경우, 파일 목록에서 바로 삭제됨 (4) 파일 정보 및 업로드 진행 정보 <ul style="list-style-type: none"> - 업로드 진행 상태 - 현재 업로드한 파일 수 - 총 업로드 가능한 파일 수 - 현재 업로드한 파일 용량 - 총 업로드 가능한 파일 용량 (5) 첨부파일 다운로드 <ul style="list-style-type: none"> - 서버에 업로드된 파일명을 클릭하면, 해당 파일에 대한 다운로드를 수행함 - 웹 브라우저 자체 기능을 통해서 파일의 다운로드 상태를 확인할 수 있음 			
REQ-UI-018	서비스 중복 호출 방지	<p>■ 버튼 중복 클릭 방지 처리</p> <ul style="list-style-type: none"> - 버튼 연속 클릭 시 onClick 이벤트가 중복 호출되지 않도록 함 <p>■ Progress Bar Layer 처리</p> <ul style="list-style-type: none"> - 서비스 중복 호출 등으로 인한 장애를 유발할 수 있는 서비스 호출 시에는 Progress Bar Layer 처리 - Progress Bar Layer가 처리가 되면 화면 내 다른 UI 요소에 클릭이 불가능함 - 주요 대상 서비스 : 업무 화면 내 메인 데이터 조회 및 데이터 업데이트 서비스 			
REQ-UI-019	공통 코드	<p>■ 공통 코드 처리 프로세스</p> <ol style="list-style-type: none"> (1) 화면 오픈 시 화면 내에서 필요한 공통 코드 로딩을 위한 서비스를 호출함 (2) 서버에서 전달받은 공통 코드 데이터를 저장하기 위한 데이터 저장 객체(DataList) 동적으로 생성해서 저장함 (3) 데이터 저장 객체와 화면 UI 컴포넌트를 바인딩 처리함 <p>■ 공통 코드 표현 방식</p> <ul style="list-style-type: none"> - 공통 코드 그룹의 데이터가 많지 않은 경우, TableLayout 형태에서는 SelectBox, Radio, CheckBox, MultiSelect, CheckComboBox 등을 이용해서 표시하고, Grid에서는 SelectBox, CheckComboBox를 이용해서 표시함 - SelectBox, Radio는 단 건 선택이 가능하고, CheckComboBox와 MultiSelect, CheckBox는 다 건 선택이 가능함 - 공통 코드 그룹의 데이터가 많거나 코드 조회 기능이 필요한 경우, 공통 코드 팝업 창을 이용해서 표시함 			
REQ-UI-020	공통 메시지	<p>■ 공통 메시지 처리</p> <ul style="list-style-type: none"> - 화면과 UI 공통 기능에서 출력되는 공통 메시지(KEY, VALUE)를 로딩하고 KEY값에 의해서 메시지를 가져오는 기능 			
REQ-UI-021	권한 제어	<p>■ 권한 제어 프로세스</p> <ol style="list-style-type: none"> (1) 로그인 화면에서 초기 데이터 로딩 시 각 프로그램 별 사용자 권한 정보 로딩을 위한 서비스를 호출해서 저장함 (2) 글로벌 영역에 저장된 프로그램 별 권한 데이터를 이용해서 화면의 버튼을 제어함(권한이 없는 버튼들을 숨김 처리함) 			
REQ-UI-022	화면 간 데이터 전달	<p>■ 화면 간 데이터 전달 유형</p> <ul style="list-style-type: none"> - 동일한 Window(Tab) 내에서 다른 화면으로 전환하면서 데이터 전달 - 다른 Window(Tab)를 오픈하면서 데이터 전달 - 팝업 화면을 오픈하면서 데이터 전달 			

REQ-UI-023	History Back 기능	■ History Back 동작 - 업무 화면이 열릴 때마다 브라우저의 History에 기록이 되어, 브라우저의 [Back], [Forward], [Refresh] 버튼이 클릭 되었을 때 이전, 다음 페이지 이동 및 새로고침이 동작해야 함 ■ Tab View, Multi View History Back 동작 - [Back] 버튼 클릭 시 현재 열려 있는 화면 중에서 바로 이전에 열렸던 화면으로 이동함 - [Forward] 버튼 클릭 시 현재 열려 있는 화면 중에서 이전 화면으로 이동하기 전 화면으로 이동함 - [Refresh] 버튼을 클릭 시 현재 열려 있는 화면은 새로고침 되고, 메인 화면을 제외한 나머지 화면은 닫힘 ■ Single View History Back 동작 - [Back] 버튼 클릭 시에 바로 이전에 열렸던 화면으로 이동함 - [Forward] 버튼 클릭 시에는 [Back] 버튼에 의해서 이전 화면으로 이동하기 전 화면으로 이동함 - [Refresh] 버튼을 클릭 시 현재 열린 화면이 새로고침 되어야 함			
REQ-UI-024	화면 종료 시 데이터 저장 여부 체크	■ Tab View, Multi View 화면 닫기 동작 - 화면 닫기 버튼을 클릭하면, 현재 화면 내에 저장하지 않은 데이터 변경이 존재할 경우, 사용자에게 데이터를 저장하지 않고 화면을 종료할 것인지 묻는 Confirm 창을 오픈함 - 전체 화면 닫기 버튼을 클릭하면, 저장하지 않은 데이터 변경이 없는 화면은 모두 닫고, 저장하지 않은 데이터 변경이 있는 화면의 경우 Confirm 창을 오픈함 - Confirm 창에서 [확인] 버튼을 클릭하면 화면을 닫지 않고, [취소] 버튼을 클릭하면 화면을 닫음 ■ Single View 화면 닫기 동작 - 기존에 화면이 열린 상태에서 다른 화면을 오픈할 때, 기존 화면이 닫히게 되는데, 이때 열려 있던 기존 화면에서 저장하지 않은 데이터 변경이 있을 경우, 화면을 종료할 것인지 묻는 Confirm 창을 오픈함 - Confirm 창에서 [확인] 버튼을 클릭하면 화면을 닫지 않고, [취소] 버튼을 클릭하면 화면을 닫음			
REQ-UI-025	단축키 기능	■ 화면 별 단축키 설정 및 동작 - 업무 화면 단위로 함수 및 버튼 클릭 이벤트에 대해서 단축키를 지정한 후, 해당 업무 화면에 단축키가 실행되면 해당 함수 및 이벤트가 실행됨			
REQ-UI-026	시스템 글로벌 데이터 제어 함수	■ 시스템 글로벌 데이터 대상 데이터 - 사용자 로그인 정보 - 메뉴 정보 - 즐겨찾기 정보 - 전체 화면 공유 데이터			
REQ-UI-027	서비스 통신 관련 공통 함수	■ 서비스 통신 전후 공통 처리 (1) 서비스 통신 전 처리 - 공통적으로 전송해야 할 데이터를 추가해서 전송 - 서버에 전송할 데이터는 JSON 포맷으로 전달함 (2) 서비스 통신 후 처리 - 응답 결과 데이터를 DataCollection에 저장 - 에러 발생 시, 에러 코드 및 메시지를 에러 메시지 창으로 표시함 ■ 서비스 통신 관련 공통 함수 목록 - 동적으로 Submission 객체를 생성하는 함수			
REQ-UI-028	객체 및 컴포넌트 제어 관련 함수	■ 객체 및 컴포넌트 제어 관련 함수 목록 - 컴포넌트 class 속성 Toggle 함수 - 현재 Scope 내에서 문자열에 해당하는 컴포넌트 객체를 찾아서 반환하는 함수 - 동적으로 컴포넌트를 생성하는 함수 - JSON 객체가 맞는지 검사하는 함수 - 배열 객체가 맞는지 검사하는 함수 - 문자열을 JSON 객체로 변환하는 함수 - 일정 간격으로 함수를 실행하기 위한 Interval, Timeout 설정 관련 함수			
REQ-UI-029	GridView 컴포넌트 제어 관련 공통 함수	■ GridView 컴포넌트 제어 관련 공통 함수 목록 - GridView와 바인딩된 데이터 객체에 데이터(Excel, CSV) 파일 내 데이터를 업로드(Import)하는 함수 (화면 내 로딩된 데이터에 한정하며, 데이터 용량이 큰 경우 WAS에서 OutOfMemory Error가 발생할 수 있으므로 데이터 용량을 고려하여 업로드 행수를 제한해야 함) - GridView와 바인딩된 데이터 객체에 데이터를 파일(Excel, CSV)로 다운로드(Export)하는 함수(파일의 데이터 용량이 큰 경우 WAS에서 OutOfMemory Error가 발생할 수 있으므로 데이터 용량을 고려하여 다운로드 행수를 제한해야 함) - GridView내 특정 CheckBox 입력타입 컬럼과 RowStatus 상태를 동기화 시키는 함수			
REQ-UI-030	DataCollection 제어 관련 공통 함수	■ DataCollection 제어 관련 공통 함수 목록 - DataMap 객체 데이터 초기화 함수 - 동적으로 DataList 객체를 생성하는 함수 - 동적으로 DataMap 객체를 생성하는 함수 - 특정 컴포넌트에 바인딩된 DataCollection(DataList, DataMap) 정보를 반환하는 함수 - 특정 컴포넌트에 바인딩된 DataCollection(DataList, DataMap)의 컬럼 이름을 반환하는 함수 - 모든 DataCollection Broadcast 상태를 조회 및 설정하는 함수			
REQ-UI-031	시스템 & 이벤트 제어 관련 공통 함수	■ 시스템 & 이벤트 제어 관련 공통 함수 목록 - 현재 서버 일시를 가져오는 함수 - 지정된 영역(Division)안에서 엔터키가 입력되면 특정 함수를 실행시켜주는 함수 - 클라이언트 환경이 모바일인지 여부를 검사하는 함수 - 웹 브라우저 내 콘솔 창에 로그 기록 함수			
REQ-UI-032	날짜 관련 공통 함수	■ 날짜 관련 공통 함수 목록 - 현재 클라이언트 일시를 가져오는 함수 - 문자열에 날짜 형식 포맷을 적용하는 함수 - 특정 일시에서 지정된 기간(월, 시, 분)을 더하거나 빼는 함수 - 특정 일시에서 지정된 기간(월, 시, 분)을 더하거나 빼는 함수 - 입력된 날짜가 날짜 형식에 맞는지 검사하는 함수 - 두 날짜의 차이를 계산하는 함수 - 날짜 형식이 맞는지 검사하는 함수 - 양력 문자열을 음력 문자열로 변환하는 함수			
REQ-UI-033	문자열 관련 공통 함수	■ 문자열 관련 공통 함수 목록 - 문자열 타입인지 검사하는 함수 - 문자열 내 특정 문자열을 치환하는 함수 - 문자열이 Empty 상태(undefined or null or "")인지를 검사하는 함수 - 주민번호 적합성을 검사하는 함수 - 사업자번호 적합성을 검사하는 함수 - 전화번호 적합성을 검사하는 함수 - 이메일 적합성을 검사하는 함수 - 문자열의 Byte수를 반환하는 함수 - 문자열에 주민번호 형식 포맷을 적용하는 함수 - 문자열에 전화번호 형식 포맷을 적용하는 함수 - 문자열에 시간 포맷을 적용하는 함수 - 세자리마다 콤마를 표시하는 함수 - 소문자를 대문자로 변환하는 함수 - 문자열 뒤에 알맞은 조사('은' 또는 '는')를 붙여주는 함수 - 특정한 문자로 주어진 길이 만큼 채우는 함수 - 문자열이 전체가 한글인지 여부를 검사하는 함수 - 문자열안에 한글이 포함되어 있는지 여부를 검사하는 함수 - JSON 객체를 문자열 타입으로 변환하는 함수			
REQ-UI-034	숫자 관련 공통 함수	■ 숫자 관련 공통 함수 목록 - 올림 처리 함수 - 반올림 처리 함수 - 내림 처리 함수 - 정수 여부를 검사하는 함수 - 홀수 여부를 검사하는 함수 - 문자열을 정수형으로 변환하는 함수 - 문자열을 실수형으로 변환하는 함수			