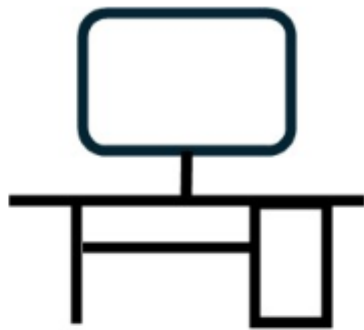


3. Design

Desco (Desk decorate)



DESCO

(Student No, Name, E-mail)

22313525, 박인석, ibsq123@yu.ac.kr

[Revision history]

Revision date	Version #	Description	Author
05/31/2025	1.00	First draft	박인석

= Contents =

1. Introduction	
2. Class diagram	
3. Sequence diagram	
4. State machine diagram	
5. Implementation requirements	
6. Glossary	
7. References	

1. Introduction

- Summarize the contents of this document.
- Describe the important points of your design.
- 12pt, 160%.

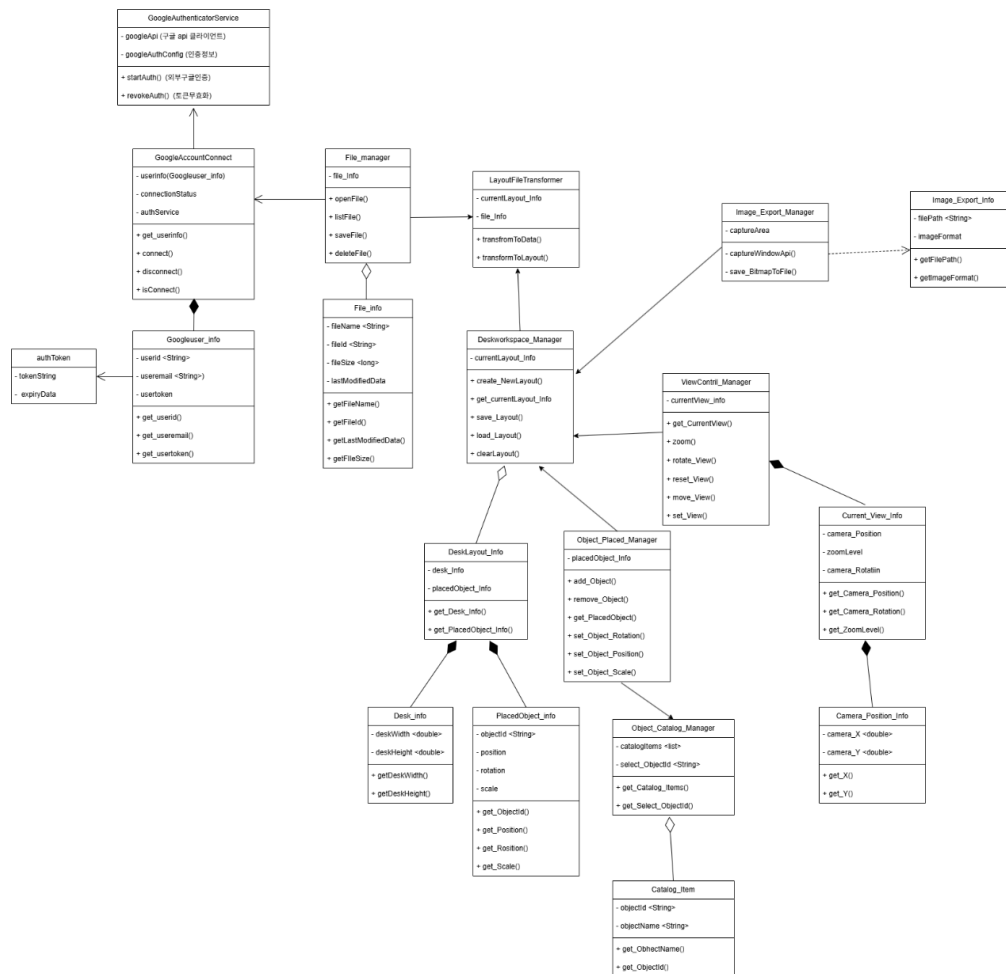
본 문서는 'Desco (Desk decorate)' 애플리케이션의 시스템 디자인을 설명합니다.

클래스 다이어그램을 통해 시스템의 정적 구조와 주요 구성 요소 간의 관계를 정의합니다. 시퀀스 다이어그램은 Google Drive 연동, 파일 저장 및 불러오기, 오브젝트 배치 및 수정, 뷰 관리, 이미지 내보내기 등 주요 기능에 대한 객체 간의 동적 상호작용 흐름을 상세히 보여줍니다. 스테이트 머신 다이어그램은 시스템 및 주요 객체의 생명주기 동안 발생할 수 있는 다양한 상태 변화를 모델링합니다. 또한, 시스템 구현을 위한 운영 환경 요구사항과 본 문서에서 사용된 용어에 대한 정의를 포함합니다.

핵심 디자인 원칙은 모듈화된 아키텍처를 기반으로 하며, Google Drive API를 활용한 클라우드 기반 데이터 저장 및 사용자 인증 기능을 통합하는 데 중점을 둡니다. 이는 사용자에게 직관적인 데스크 꾸미기 경험을 제공하는 동시에, 데이터의 영속성과 접근성을 확보하기 위한 것입니다.

2. Class diagram

- Draw a class diagram.
- Describe each class in detail (attributes, methods, others) (table type).
- 12pt, 160%.



Class name	explanation
GoogleAuthenticator Service	<p>Google 인증</p> <p>Attributes:</p> <ul style="list-style-type: none"> - googleApi : Google의 인증 서버 및 리소스 서버와 실제 HTTP 통신을 수행하는 저수준 클라이언트 객체 (예: 토큰 요청, 토큰 무효화 요청 전송) - googleAuthConfig (인증 정보): 클라이언트 ID, 비밀번호, 요청 등 Google 인증에 필요한 설정 정보 <p>Methods:</p> <ul style="list-style-type: none"> + startAuth(): String 외부 Google 인증 플로우를 시작 (로그인 화면 표시 등). 인증 URL을 생성하여 반환

	<p>+ getUserAuthData (authorizationCode:String): 구글 api 에서 받은 authorizationCode를 사용하여 Google로부터 액세스 토큰을 받고, 해당 토큰으로 사용자 프로필 정보 까지 가져 와 UserAuthBundle 형태로 반환</p> <p>+ revokeAuth(token:AuthToken): 전달받은 인증 토큰을 Google 서버에 무효화 요청.</p>
GoogleAccountConnect	<p>Google 인증</p> <p>Attributes:</p> <ul style="list-style-type: none"> - userinfo (Googleuser_info): 현재 연결된 사용자의 정보 객체. - connectionStatus (ConnectionStatus): 현재 계정 연결 상태 (예: Connected, Disconnected, Error). <p>Methods:</p> <ul style="list-style-type: none"> + get_userinfo(): Googleuser_info: 현재 연결된 사용자 정보 객체를 반환. + connect(): String Google 계정 연결 프로세스를 시작 (authService.startAuth() 호출). + disconnect(): void: Google 계정 연결 해제 프로세스를 시작 (authService.revokeAuth() 호출) + isConnect(): boolean: 현재 계정이 연결되어 있는지 여부를 반환. - handleAuthSuccess(authData : AuthData): 인증 성공시 전달받은 bundle의 데이터로 this.userInfo를 생성/업데이트하고, Connected로 설정
	<p>Google 인증</p> <p>Attributes:</p> <ul style="list-style-type: none"> - userid (String): 사용자의 Google 고유 ID.

<p>Googleuser_info</p>	<ul style="list-style-type: none"> - useremail (String): 사용자의 Google 이메일 주소. - usertoken (AuthToken): Google API 호출에 사용될 인증 토큰 객체 <p>Methods:</p> <ul style="list-style-type: none"> + get_userid(): String: 사용자 ID 반환. + get_useremail(): String: 사용자 이메일 반환. + get_usertoken(): AuthToken: 인증 토큰 객체 반환.
<p>AuthToken</p>	<p>Google 인증</p> <p>Attributes:</p> <ul style="list-style-type: none"> - tokenString (String): 실제 액세스 토큰 문자열. - expiryData (Date): 토큰 만료 일시. <p>Methods:</p> <ul style="list-style-type: none"> + getTokenString(): String: 액세스 토큰 문자열 반환. + getExpiryData(): Date: 만료 일시 반환. + isExpired(): boolean: 토큰이 만료되었는지 확인.
<p>File_manager</p>	<p>파일 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - file_Info_List (List<File_info>): 관리하는 파일 정보 목록 - googleDriveApi Google Drive API와 직접 통신하기 위한 클라이언트 객체. <p>Methods:</p> <ul style="list-style-type: none"> + openFile(fileId:String): Google Drive에서 fileId에 해당하는 파일의 내용을 다운로드하고, LayoutFileTransformer 호출 DeskLayout_Info 객체로 변환한 후 반환 + listFile(): List<File_info>: Google Drive의 파일 목록을

	<p>조회하여 각 파일의 메타데이터를 담은 File_Info 객체들의 리스트로 반환</p> <p>+ saveFile(transfromToData(DeskLayout_Info), fileName:String): boolean: 전달받은 DeskLayout_Info 객체를 LayoutFileTransformer를 사용하여 저장 가능한 데이터 형태로 변환한 후, 해당 데이터를 fileName으로 Google Drive에 파일로 저장</p> <p>+ deleteFile(fileId:String): boolean: Google Drive에서 fileId에 해당하는 파일을 삭제하고, 성공 여부를 boolean으로 반환</p>
File_info (파일 정보)	<p>파일 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - fileName (String): 파일 이름. - fileId (String): Google Drive 상의 파일 고유 ID. - fileSize (long): 파일 크기 (바이트 단위). - lastModifiedData (Date): 마지막 수정 일시. <p>Methods:</p> <ul style="list-style-type: none"> + getFileName(): String: 파일 이름 반환. + getFileId(): String: 파일 ID 반환. + getLastModifiedData(): Date: 마지막 수정 일시 반환. + getFileSize(): long: 파일 크기 반환.
LayoutFileTransformer	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <p>Methods:</p> <ul style="list-style-type: none"> + transformToData(layoutInfo : DeskLayout_Info): FileData: DeskLayout_Info 객체를 파일

	<p>저장용 FileData로 변환. // JSON</p> <p>+ transformToLayout(data : FileData): DeskLayout_Info: FileData를 DeskLayout_Info 객체로 변환.</p>
Deskworkspace_Manager	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - currentLayout_Info (DeskLayout_Info): 현재 작업 중인 레이아웃 객체. <p>Methods:</p> <ul style="list-style-type: none"> + create_NewLayout(): 새 레이아웃 생성. + get_currentLayout_Info(): 현재 레이아웃 반환. + load_DeskLayout() void: 외부에서 로드된 레이아웃으로 현재 레이아웃 설정. + clearLayout(): void: 현재 레이아웃 초기화
DeskLayout_Info	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - desk_Info (Desk_info): 책상 정보 객체 - placedObjects (List<PlacedObject_info>): 포함하는 배치된 오브젝트 목록 <p>Methods:</p> <ul style="list-style-type: none"> + get_Desk_Info(): Desk_info: 책상 정보 반환 + get_PlacedObject_Info(): List<PlacedObject_info>: 배치된 오브젝트 목록 반환

<p>Desk_info</p>	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - deskWidth (double): 책상 너비 - deskHeight (double): 책상 높이 <p>Methods:</p> <ul style="list-style-type: none"> + getDeskWidth(): double: 너비 반환 + getDeskHeight(): double: 깊이 반환
<p>PlacedObject_info</p>	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - objectId (String): 배치된 인스턴스의 고유 ID - position (Vector3): 3D 위치 - rotation (vector3): 3D 회전 - scale (Vector3): 3D 크기 - catalogItem (CatalogItemInfo): 이 배치된 오브젝트의 원본이 되는 카탈로그 아이템 정보에 대한 참조 <p>Methods:</p> <ul style="list-style-type: none"> + get_ObjectId(): String: 인스턴스 ID 반환 + get_CatalogItem(): Catalog_Item: 카탈로그 아이템 반환 + get_Position(): Vector3: 위치 반환 + get_Rotation(): 회전 반환 + get_Scale(): 크기 반환.
	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - placedObject_Info (PlacedObject_info): 현재 선택된 배

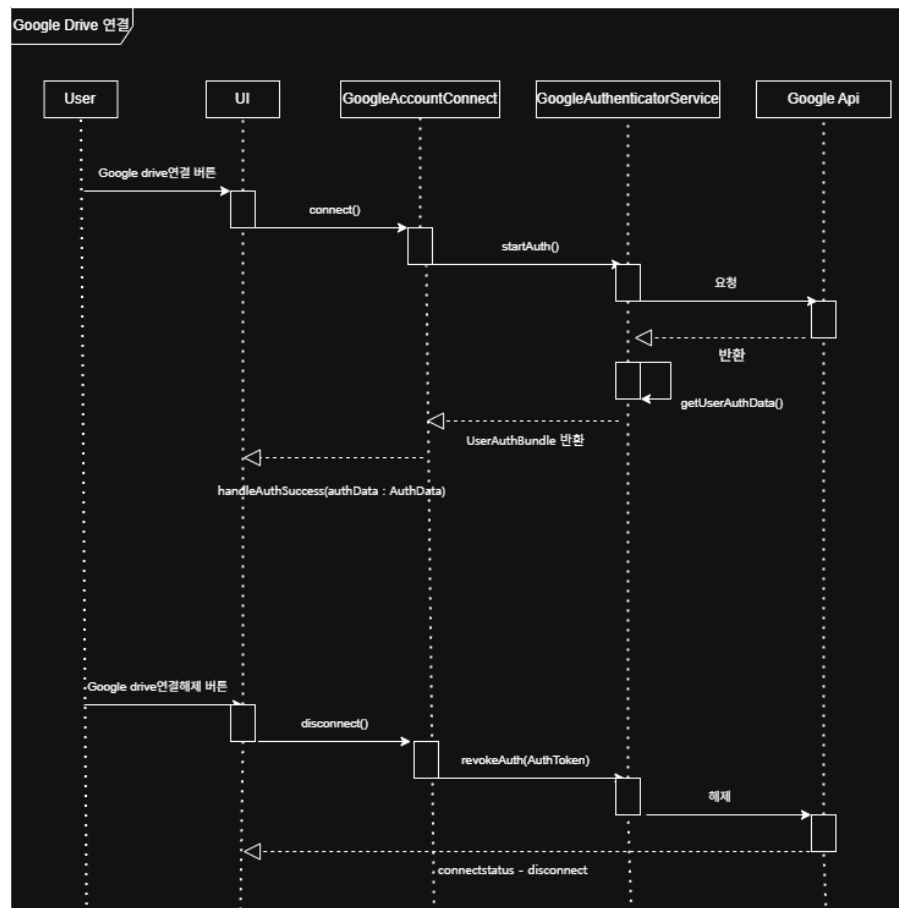
Object_Placed_Manager	<p>치 오브젝트</p> <p>Methods:</p> <ul style="list-style-type: none"> + add_Object (catalogObjectId: String, position: PositionInfo): PlacedObject_info: objectCatalogManager로부터 catalogObjectId에 해당하는 기본 오브젝트를 더함 + remove_Object(objectId): void: 레이아웃에서 오브젝트 삭제. + get_PlacedObject(objectId): PlacedObject_info: 배치된 오브젝트 반환 + set_Object_Rotation(objectId, rotation): void: 오브젝트 회전 설정. + set_Object_Position(objectId, position): void: 오브젝트 위치 설정. + set_Object_Scale(objectId, scale): void: 오브젝트 크기 설정 + selectObject(objectId: String): void objectId에 해당하는 객체를 찾아 selectedObjectInfo로 설정
Object_Catalog_Manager	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - catalogItems (List<Catalog_Item>): 카탈로그 아이템 목록 - selectedCatalogItemId (String): 현재 선택된 카탈로그 아이템 ID <p>Methods:</p> <ul style="list-style-type: none"> + get_CatalogItems(): List<Catalog_Item>: 카탈로그 목록 반환 + get_selectedCatalogItemId ()String: 선택된 카탈로그 아이템 ID 반환

Catalog_Item	<p>오브젝트/데스크 관리 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - catalogId (String): 이 카탈로그 아이টে을 고유하게 식별하는 ID. - objectName (String): 카탈로그 아이টে 이름. <p>Methods:</p> <ul style="list-style-type: none"> + get_ObjectName(): String: 이름 반환. + get_CatalogId(): String: 카탈로그 아이টে의 고유 ID를 반환
ViewContril_Manager	<p>View 관련 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - currentView_info (Current_View_Info): 현재 뷰 정보 객체 <p>Methods:</p> <ul style="list-style-type: none"> + get_CurrentView(): Current_View_Info: 현재 뷰 정보 반환 + zoom(factor:double): void: 줌 조절 + rotate_View(Vector3d) void: 뷰 회전 + move_View(Vector3d): void: 뷰 이동 + reset_View(): void: 뷰 초기화
Current_View_Info	<p>View 관련 도메인</p> <p>Attributes:</p> <ul style="list-style-type: none"> - camera_Position (Vector3d): 카메라 위치 - zoomLevel (double): 줌 레벨. - camera_Rotation (Vector3d): 카메라 회전

	Methods: + get_Camera_Position(): 카메라 위치 반환. + get_Camera_Rotation(): double: 카메라 회전 반환. + get_ZoomLevel(): double: 줌 레벨 반환.
Image_Export_Manager	이미지 다운로드 도메인 Attributes: - captureArea (): 캡처 영역 Methods: + captureAndSaveImage(exportInfo : Image_Export_Info): boolean: 화면 캡처 후 파일로 저장. - captureWindowApi(): PlatformBitmap: 윈도우 API로 캡처 - save_BitmapToFile(PlatformBitmap, filePath:String, ImageFormat): boolean 비트맵을 파일로 저장
Image_Export_Info	이미지 다운로드 도메인 Attributes: - filePath (String): 저장될 파일 경로 - imageFormat (ImageFormat): 이미지 형식 Methods: + get_FilePath(): String: 파일 경로 반환 + get_ImageFormat(): ImageFormat: 이미지 형식 반환

3. Sequence diagram

- Draw sequence diagrams for the whole functions of your system (this is related to the use cases you made in the Conceptualization phase).
- Explain each sequence diagram.
- 12pt, 160%.



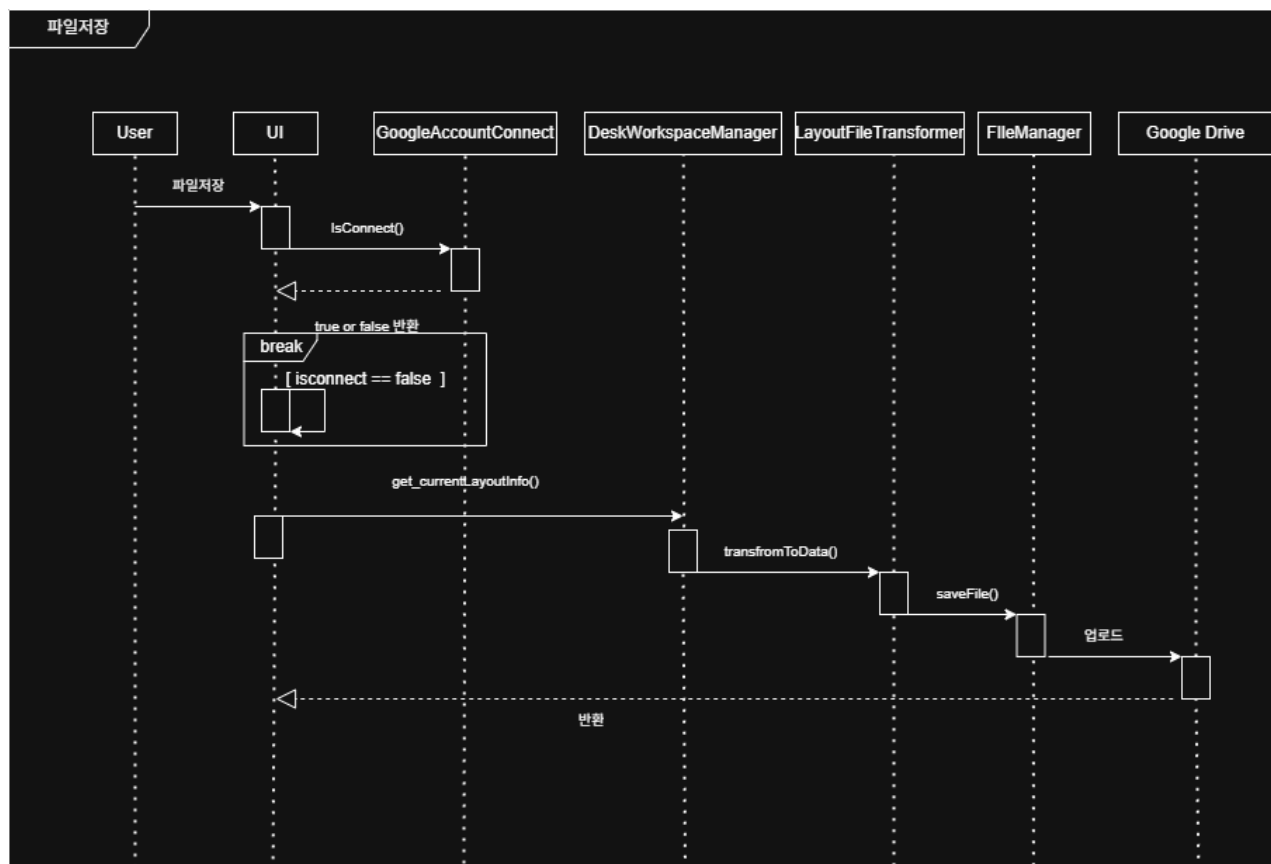
Google Drive 연결:

사용자가 Desco 애플리케이션 내에서 "Google Drive 연결" 버튼을 누르면, UI는 계정 연결 관리자(GoogleAccountConnect)에게 연결 시작을 요청합니다. 계정 연결 관리자는 실제 인증 처리를 담당하는 인증 서비스(GoogleAuthenticatorService)를 호출하여 인증 절차를 개시합니다. 인증 서비스는 Google API 서버와 통신하여 사용자를 Google 로그인 및 권한 동의 페이지로 안내하고, 사용자가 성공적으로 인증을 마치면 Google API로부터 인증 코드를 받습니다. 이 인증 코드를 사용하여 인증 서비스는 다

시 Google API와 통신해 최종적으로 액세스 토큰 및 사용자 정보가 담긴 데이터 묶음(UserAuthBundle)을 획득합니다. 이 데이터 묶음은 계정 연결 관리자에게 전달되어 사용자 정보 저장 및 연결 상태 업데이트에 사용되며, 최종적으로 UI는 연결 성공 상태를 사용자에게 알리고 화면을 갱신합니다.

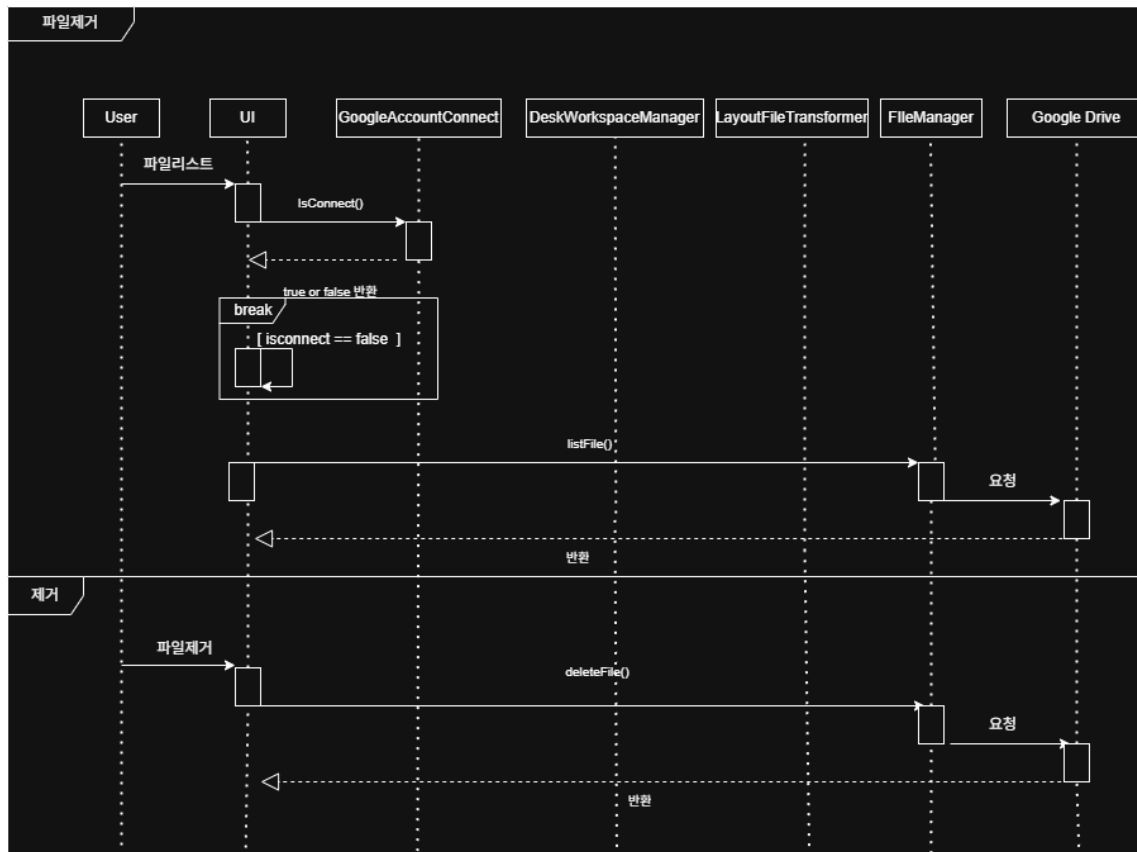
Google Drive 연결 해제:

사용자가 "Google Drive 연결 해제" 버튼을 누르면, UI는 계정 연결 관리자에게 연결 해제를 요청합니다. 계정 연결 관리자는 현재 저장된 사용자의 인증 토큰을 인증 서비스에 전달하며 토큰 무효화를 요청합니다. 인증 서비스는 이 토큰을 가지고 Google API 서버에 해당 토큰을 더 이상 사용하지 않도록 무효화 요청을 보냅니다. Google API로부터 토큰 무효화 성공 응답을 받으면, 인증 서비스는 이 결과를 계정 연결 관리자에게 알립니다. 계정 연결 관리자는 저장된 사용자 정보를 삭제하거나 초기화하고 연결 상태를 '연결 해제됨'으로 변경한 후, 이 결과를 UI에 전달합니다. UI는 사용자에게 연결 해제 상태를 알리고 화면을 갱신합니다.



파일 저장 (Google Drive에 저장)

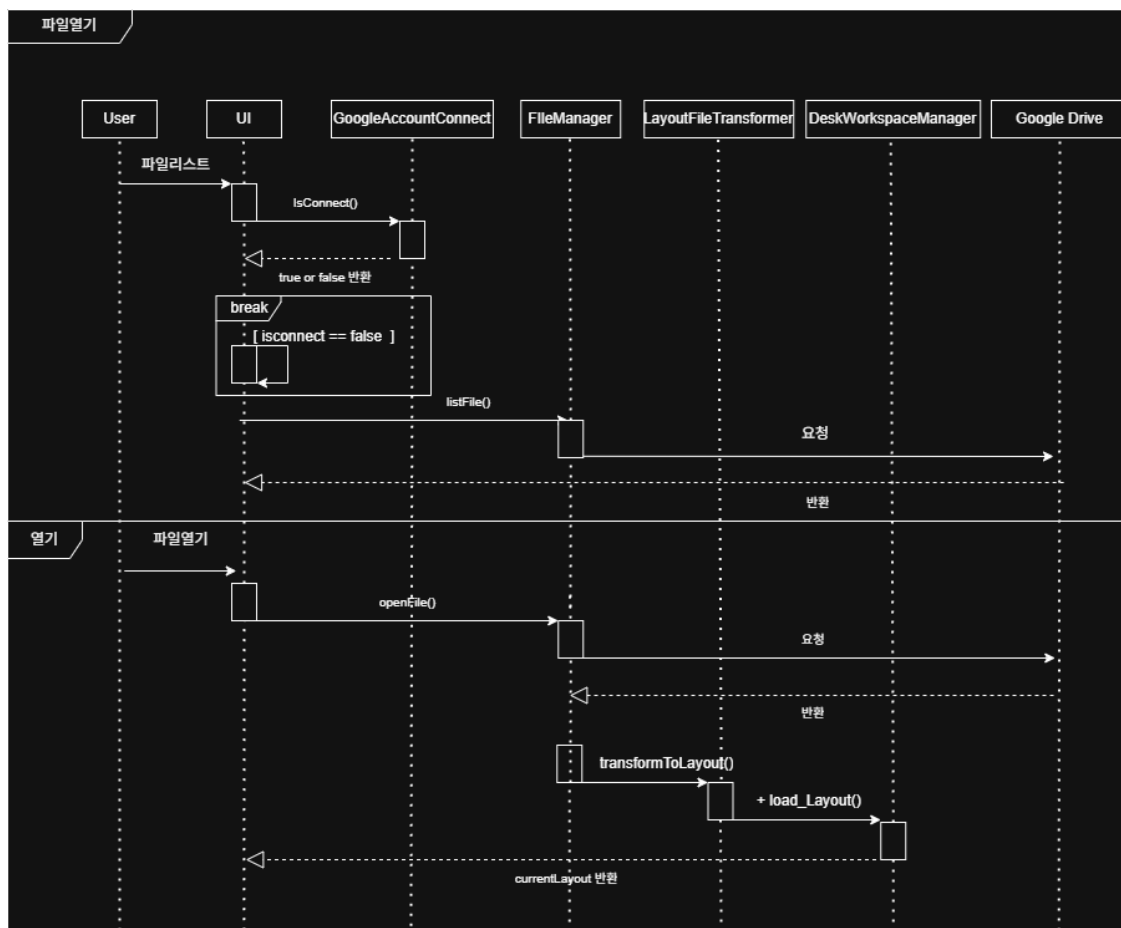
사용자가 Desco 애플리케이션 내에서 "파일 저장" 기능을 실행하면, UI는 먼저 GoogleAccountConnect에게 Google Drive 계정 연결 여부를 문의합니다. GoogleAccountConnect는 연결 상태를 UI에 알리고, 연결되어 있지 않다면 UI는 파일 저장 절차를 중단합니다. 계정이 연결된 상태라면, UI는 DeskWorkspaceManager에게 현재 작업 중인 레이아웃 데이터를 요청합니다. DeskWorkspaceManager로부터 레이아웃 데이터를 전달받은 UI는 이 데이터를 저장 가능한 형태로 변환하고자 LayoutFileTransformer에게 데이터 변환을 지시합니다. LayoutFileTransformer가 레이아웃 정보를 적절한 데이터 형식으로 변환하여 UI에게 돌려주면, UI는 이 변환된 데이터와 함께 파일 저장 명령을 FileManager에게 내립니다. FileManager는 전달받은 데이터를 사용하여 Google Drive에 실제 파일 저장을 시도하고, Google Drive로부터 파일 저장 작업의 결과를 응답받습니다. 최종적으로 FileManager는 이 저장 결과를 UI에게 전달하며, UI는 해당 결과를 사용자에게 알리고 화면을 갱신합니다.



파일 제거 (Google Drive에서 제거)

사용자가 Desco 애플리케이션 내에서 Google Drive에 저장된 특정 파일을 제거하고자 할 때, 먼저 UI는 사용자가 "파일 리스트"를 요청하면 GoogleAccountConnect에게 Google Drive 계

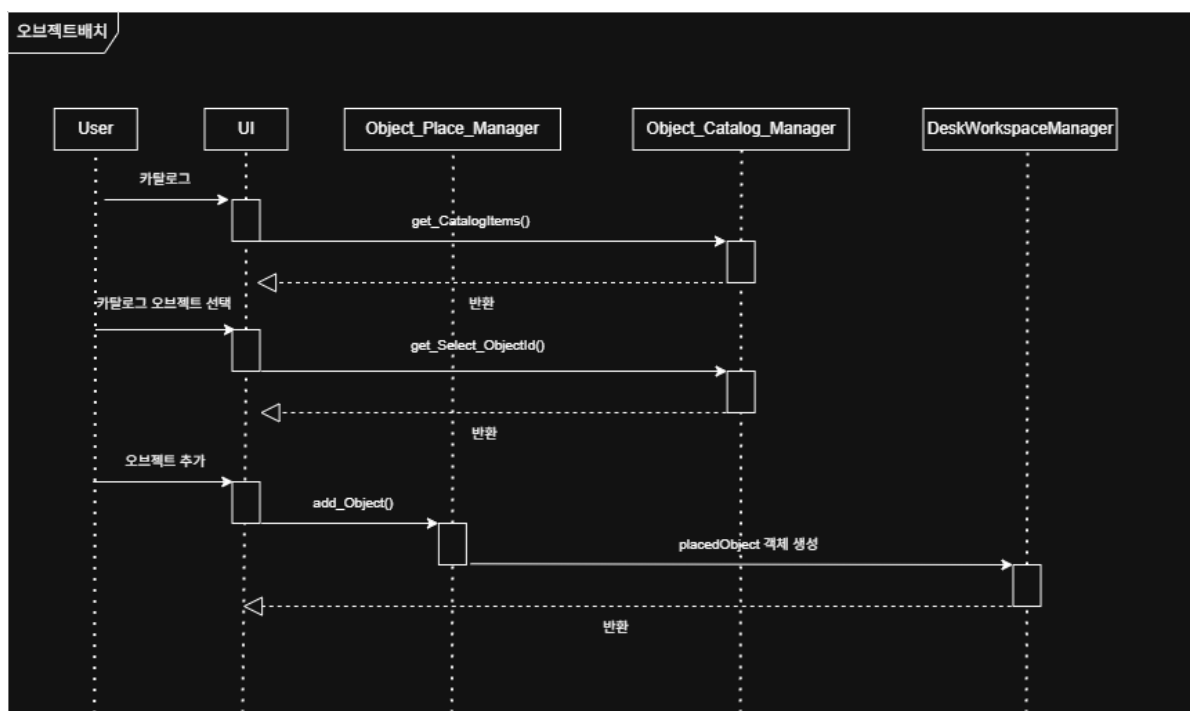
정 연결 여부를 문의합니다. GoogleAccountConnect는 연결 상태를 UI에 알리고, 연결되어 있지 않다면 UI는 파일 목록 조회 및 제거 절차를 중단합니다. 계정이 연결된 상태라면, UI는 파일 목록을 가져오기 위해 FileManager에게 Google Drive의 파일 목록을 요청합니다. Google Drive로부터 파일 목록을 반환받은 FileManager는 이 정보를 UI에 전달하고, UI는 사용자에게 파일 목록을 보여줍니다. 사용자가 목록에서 특정 파일을 선택하여 "파일 제거"를 실행하면, UI는 선택된 파일 정보와 함께 제거 명령을 FileManager에게 전달합니다. FileManager는 이 요청을 받아 Google Drive에 해당 파일의 실제 삭제를 요청합니다. Google Drive는 파일 삭제 작업을 수행하고 그 결과를 FileManager에게 응답하며, FileManager는 최종적으로 파일 제거 결과를 UI에게 반환합니다. UI는 이 결과를 사용자에게 알리고 화면을 갱신합니다.



파일 열기 (Google Drive에서 열기)

사용자가 Desco 애플리케이션 내에서 Google Drive에 저장된 파일을 열어 작업 공간에 로드하고자 할 때, 먼저 UI는 사용자가 "파일 리스트"를 요청하면 GoogleAccountConnect에게 Google Drive 계정 연결 여부를 문의합니다. GoogleAccountConnect는 연결 상태를 UI에 알

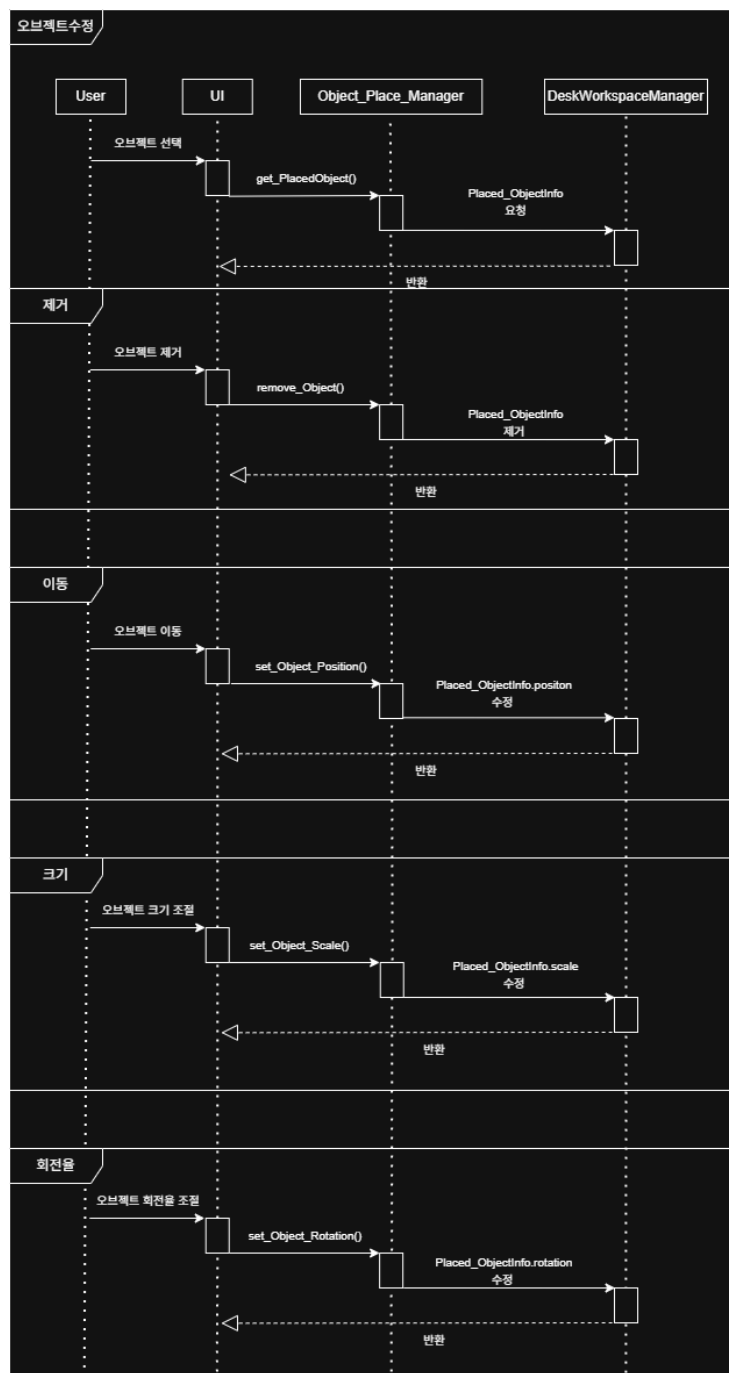
리고, 연결되어 있지 않다면 UI는 파일 목록 조회 및 열기 절차를 중단합니다. 계정이 연결된 상태라면, UI는 파일 목록을 가져오기 위해 FileManager에게 Google Drive의 파일 목록을 요청합니다. Google Drive로부터 파일 목록을 반환받은 FileManager는 이 정보를 UI에 전달하고, UI는 사용자에게 파일 목록을 보여줍니다. 사용자가 목록에서 특정 파일을 선택하여 "파일 열기"를 실행하면, UI는 선택된 파일 정보와 함께 열기 명령을 FileManager에게 전달합니다. FileManager는 이 요청을 받아 Google Drive에 해당 파일 데이터의 다운로드를 요청합니다. Google Drive는 파일 데이터를 반환하고, FileManager는 이 원본 파일 데이터를 LayoutFileTransformer에게 전달하여 Desco 애플리케이션이 이해할 수 있는 레이아웃 객체로 변환하도록 지시합니다. LayoutFileTransformer가 데이터를 성공적으로 변환하면, 이 변환된 레이아웃 객체는 DeskWorkspaceManager에게 전달되어 현재 작업 공간에 로드됩니다. 작업 공간에 레이아웃이 성공적으로 로드되면, DeskWorkspaceManager는 이 결과를 LayoutFileTransformer를 거쳐 FileManager에게 전달하고, 최종적으로 FileManager는 현재 로드된 레이아웃 정보를 UI에게 반환합니다. UI는 로드된 레이아웃을 사용자에게 보여주고 화면을 갱신합니다.



오브젝트 배치

사용자가 Desco 애플리케이션 내에서 새로운 오브젝트를 데스크 위에 배치하고자 할 때, 먼저 UI는 사용자가 "카탈로그"를 요청하면 Object_Catalog_Manager에게 카탈로그 아이템 목록을 요청합니다. Object_Catalog_Manager는 아이템 목록을 UI에 반환하고, 사용자가 "카탈로그

오브젝트 선택"을 통해 특정 아이템을 선택하면 UI는 다시 Object_Catalog_Manager에게 선택된 오브젝트의 ID를 요청하여 전달받습니다. 이후 사용자가 "오브젝트 추가" 명령을 실행하면, UI는 선택된 오브젝트 ID와 배치 위치 등의 정보를 Object_Place_Manager에게 전달하며 오브젝트 추가를 지시합니다. Object_Place_Manager는 이 정보를 바탕으로 DeskWorkspaceManager에게 실제 placedObject 객체 생성을 요청합니다. DeskWorkspaceManager는 새로운 오브젝트를 작업 공간에 생성하고 그 결과를 Object_Place_Manager에게 반환하며, 최종적으로 Object_Place_Manager는 배치 결과를 UI에게 전달하여 사용자에게 알려주고 화면을 갱신합니다.



오브젝트 수정

사용자가 Desco 애플리케이션 내에서 데스크 위에 배치된 오브젝트를 수정하고자 할 때, 먼저 "오브젝트 선택"을 통해 UI는 Object_Place_Manager에게 특정 오브젝트의 정보를 요청합니다. Object_Place_Manager는 DeskWorkspaceManager에게 해당 Placed_ObjectInfo를 요청하여 전달받고, 이 정보를 UI에 반환하여 사용자가 조작할 대상을 확정합니다.

오브젝트 제거

UI는 Object_Place_Manager에게 해당 오브젝트 제거를 지시하고, Object_Place_Manager는 DeskWorkspaceManager에게 Placed_ObjectInfo 제거를 요청합니다. "오브젝트 이동" 시, UI는 Object_Place_Manager에게 새 위치 정보와 함께 이동을 지시하며, Object_Place_Manager는 DeskWorkspaceManager에게 해당 Placed_ObjectInfo의 위치 수정을 요청합니다.

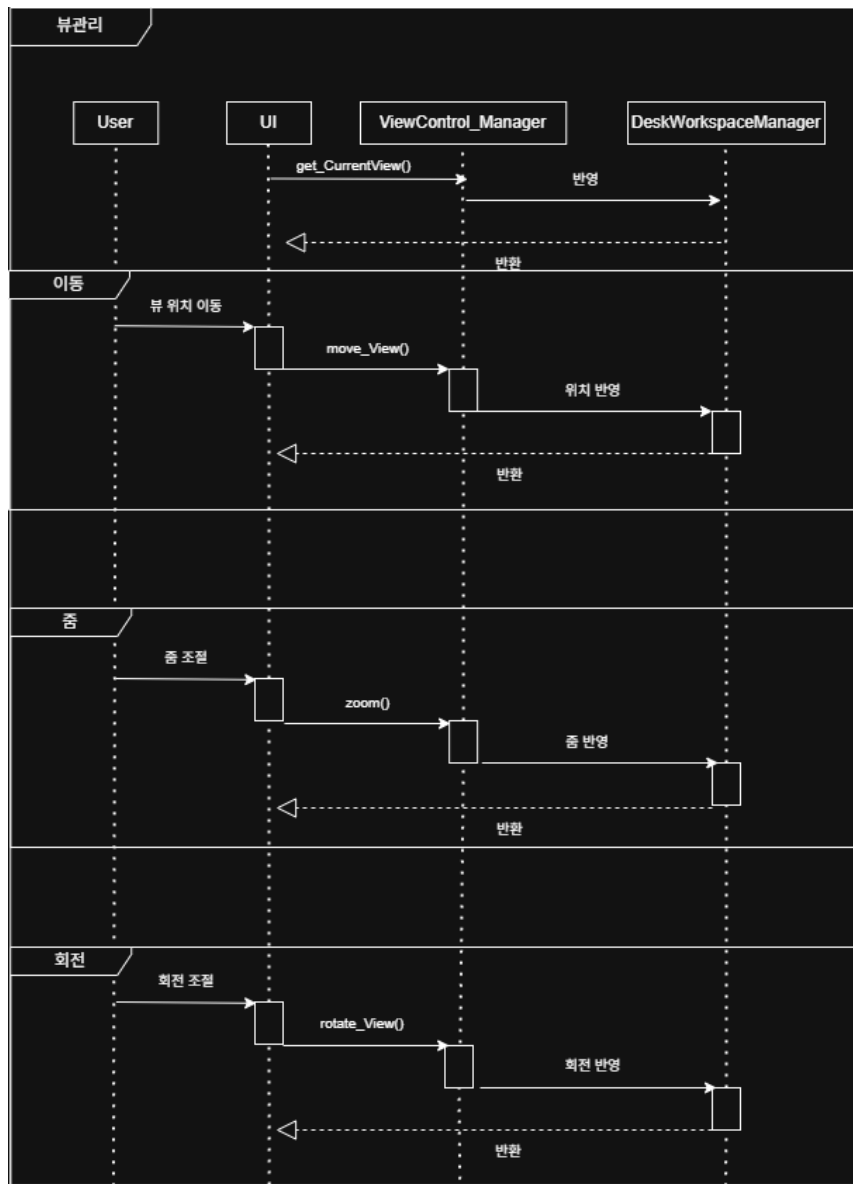
오브젝트 크기 조절

UI는 Object_Place_Manager에게 새 크기 정보와 함께 조절을 지시하고, Object_Place_Manager는 DeskWorkspaceManager에게 해당 Placed_ObjectInfo의 크기 수정을 요청합니다.

오브젝트 회전을 조절

UI는 Object_Place_Manager에게 새 회전 정보와 함께 조절을 지시하며, Object_Place_Manager는 DeskWorkspaceManager에게 해당 Placed_ObjectInfo의 회전 수정을 요청합니다.

각 경우에 DeskWorkspaceManager는 요청된 수정 작업을 수행하고 그 결과를 Object_Place_Manager에게 반환하며, 최종적으로 Object_Place_Manager는 수정 결과를 UI에게 전달하여 사용자에게 알리고 화면을 갱신합니다.



뷰 관리

사용자가 Desco 애플리케이션의 작업 공간 뷰를 관리하고자 할 때, 먼저 UI는 초기 뷰 상태를 파악하기 위해 ViewControl_Manager에게 현재 뷰 정보를 요청, ViewControl_Manager는 현재 뷰 상태를 DeskWorkspaceManager로부터 전달받아 UI에 반환합니다.

뷰 위치 이동

UI는 ViewControl_Manager에게 뷰 이동을 지시하고, ViewControl_Manager는 변경된 위치를 DeskWorkspaceManager에 반영하도록 요청합니다

줌 조절

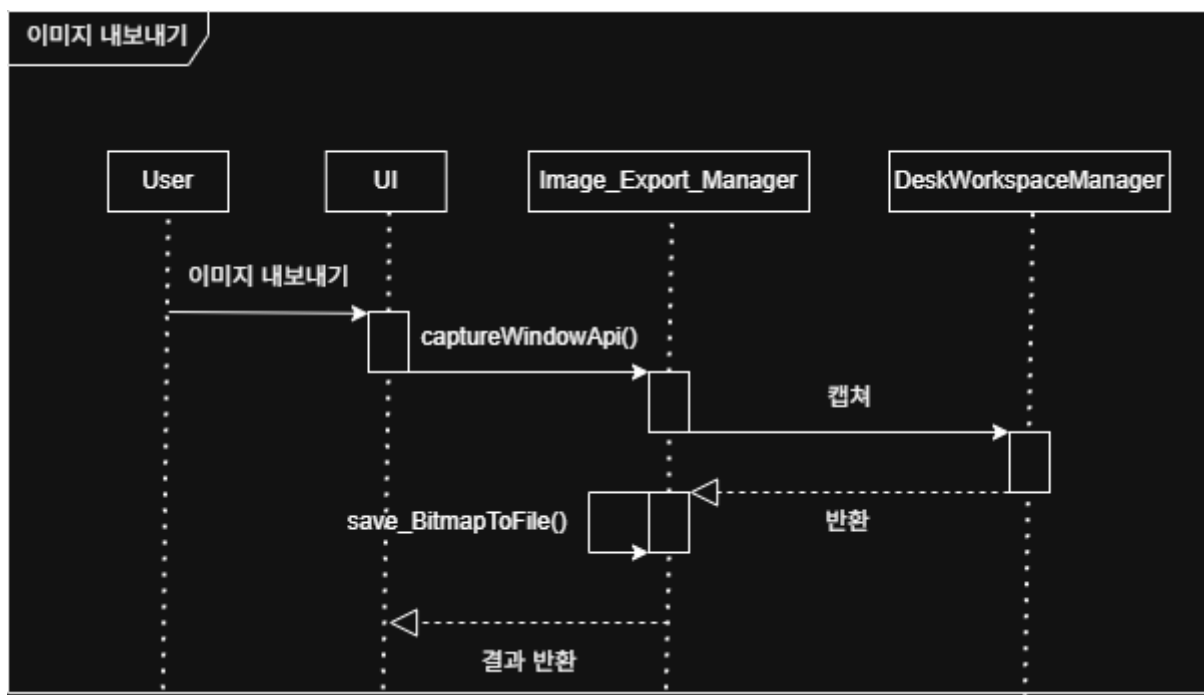
UI는 ViewControl_Manager에게 줌 레벨 변경을 지시하며, ViewControl_Manager는 새로운 줌

레벨을 DeskWorkspaceManager에 반영하도록 요청합니다.

회전 조절

UI는 ViewControl_Manager에게 뷰 회전을 지시하고, ViewControl_Manager는 변경된 회전값을 DeskWorkspaceManager에 반영하도록 요청합니다.

각 경우에 DeskWorkspaceManager는 요청된 뷰 변경 사항을 작업 공간에 적용하고 그 결과를 ViewControl_Manager에게 반환하며, 최종적으로 ViewControl_Manager는 이 결과를 UI에게 전달하여 사용자에게 변경된 뷰를 보여주고 화면을 갱신합니다.



이미지 내보내기

사용자가 Desco 애플리케이션 내에서 현재 작업 공간의 뷰를 이미지 파일로 내보내고자 할 때, UI는 사용자의 "이미지 내보내기" 요청을 받아 Image_Export_Manager에게 윈도우 캡처 API 호출을 지시합니다. Image_Export_Manager는 DeskWorkspaceManager에게 현재 작업 공간 뷰의 "캡처"를 요청하고, DeskWorkspaceManager는 해당 뷰의 이미지 데이터를 반환합니다. 이 캡처된 데이터를 전달받은 Image_Export_Manager는 내부적으로 비트맵 데이터를 파일로 저장하는 로직을 수행합니다. 최종적으로 Image_Export_Manager는 이미지 저장 작업의 결과를 UI에게 반환하며, UI는 이 결과를 사용자에게 알립니다.

- Desco(데스코) : 해당 책상 꾸미기 어플리케이션 이름
- 작업공간 : 사용자가 책상을 배치하고 오브젝트를 꾸미는 주요 3D 인터페이스 영역
- Desk : 사용자가 작업 공간 내에 생성하고 오브젝트를 배치하는 기준 가구 모델

- Object : 책상 위에 배치할 수 있는 모든 가상의 아이템
- Sequence Diagram : 객체 간의 동적 상호작용을 시간적 개념으로 모델링하여 나타낸 다이어그램
- StateMachine Diagram : 객체 LifeTime동안의 변화될 수 있는 모든 상태를 정의해 둔 다이어그램
- API (Application Programming Interface) : 응용 프로그램이나 소프트웨어 구성 요소들이 서로 상호작용하고 특정 기능이나 데이터에 접근할 수 있도록 미리 정의해 둔 규칙, 함수, 프로토콜 등의 집합으로 이루어진 인터페이스. 내부 구현의 복잡성을 몰라도 해당 기능을 활용할 수 있게 해줌

7. References

- Describe all of your references (book, paper, technical report etc).
- 12pt, 160%.

-강의자료 : Structural Modeling II, Behavior Modeling I, II

-참조자료 : [Design] Examples