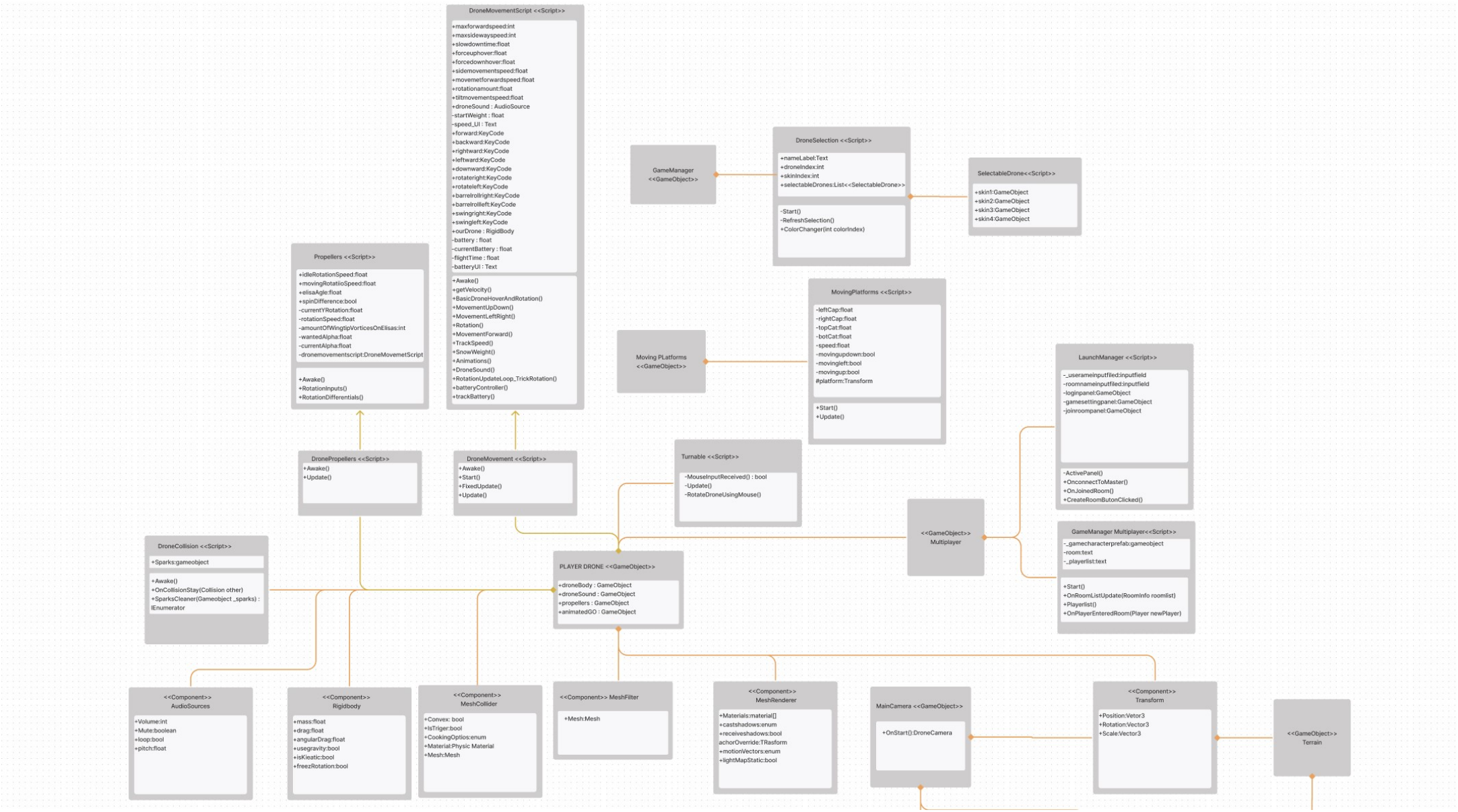
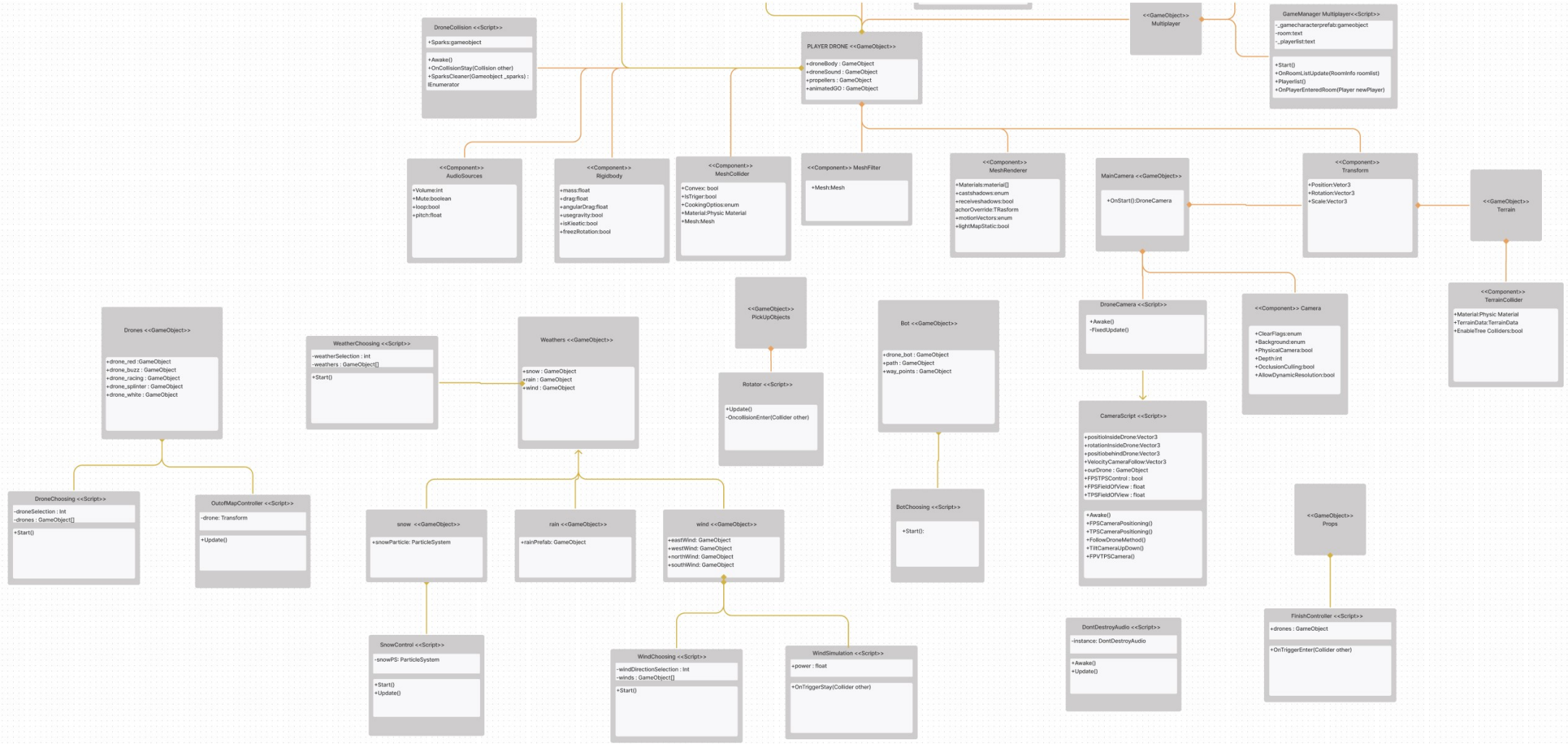
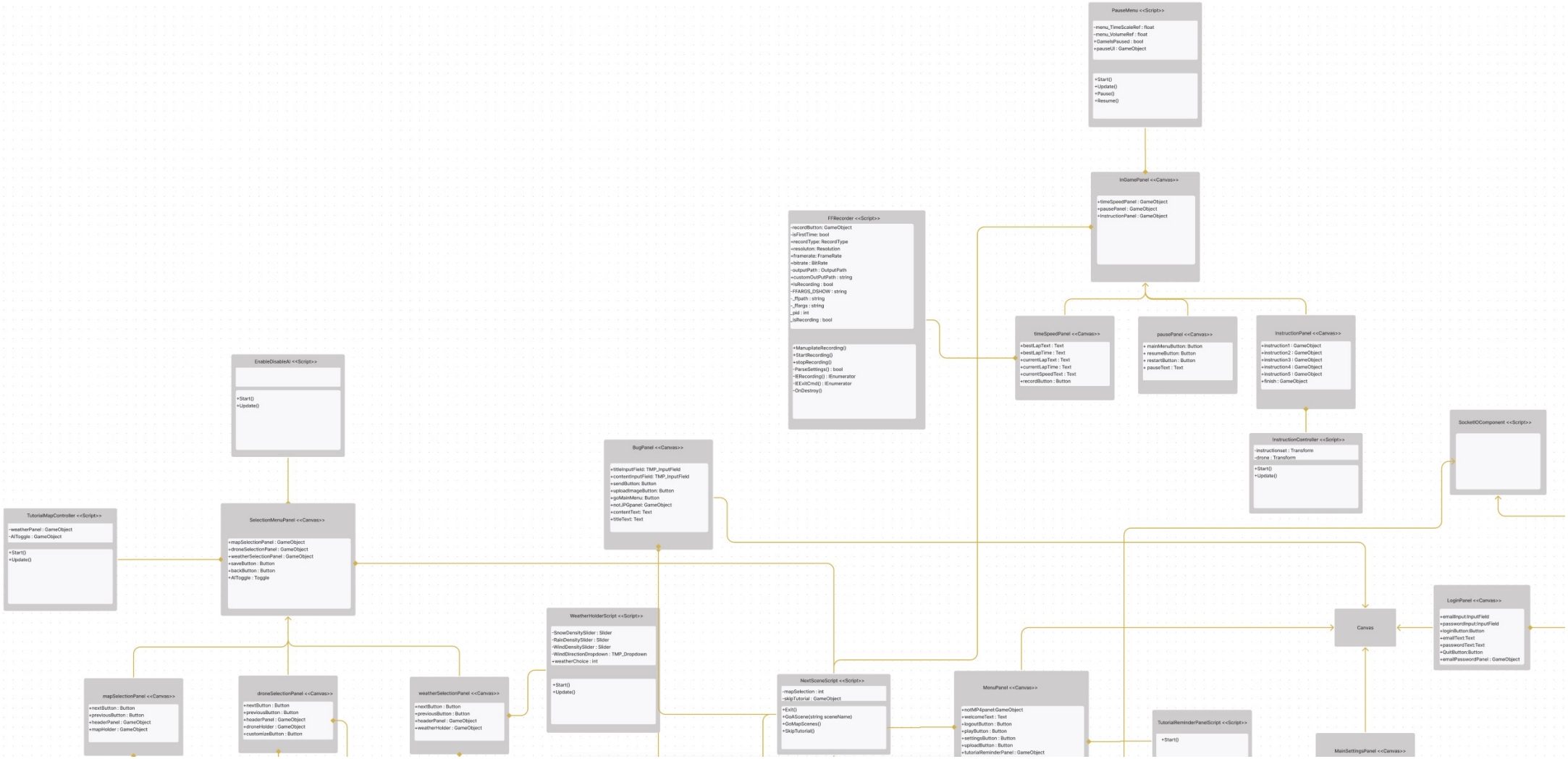
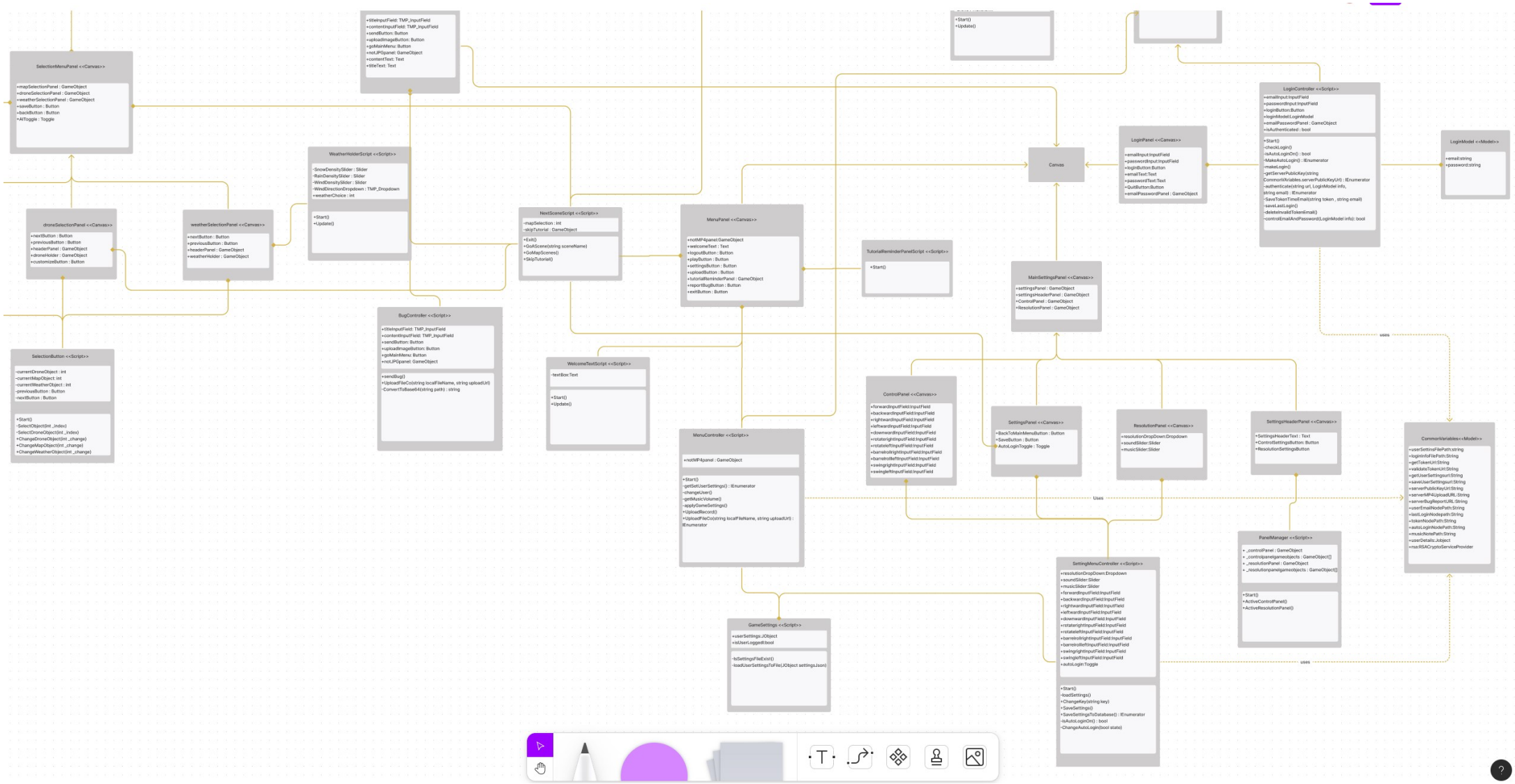
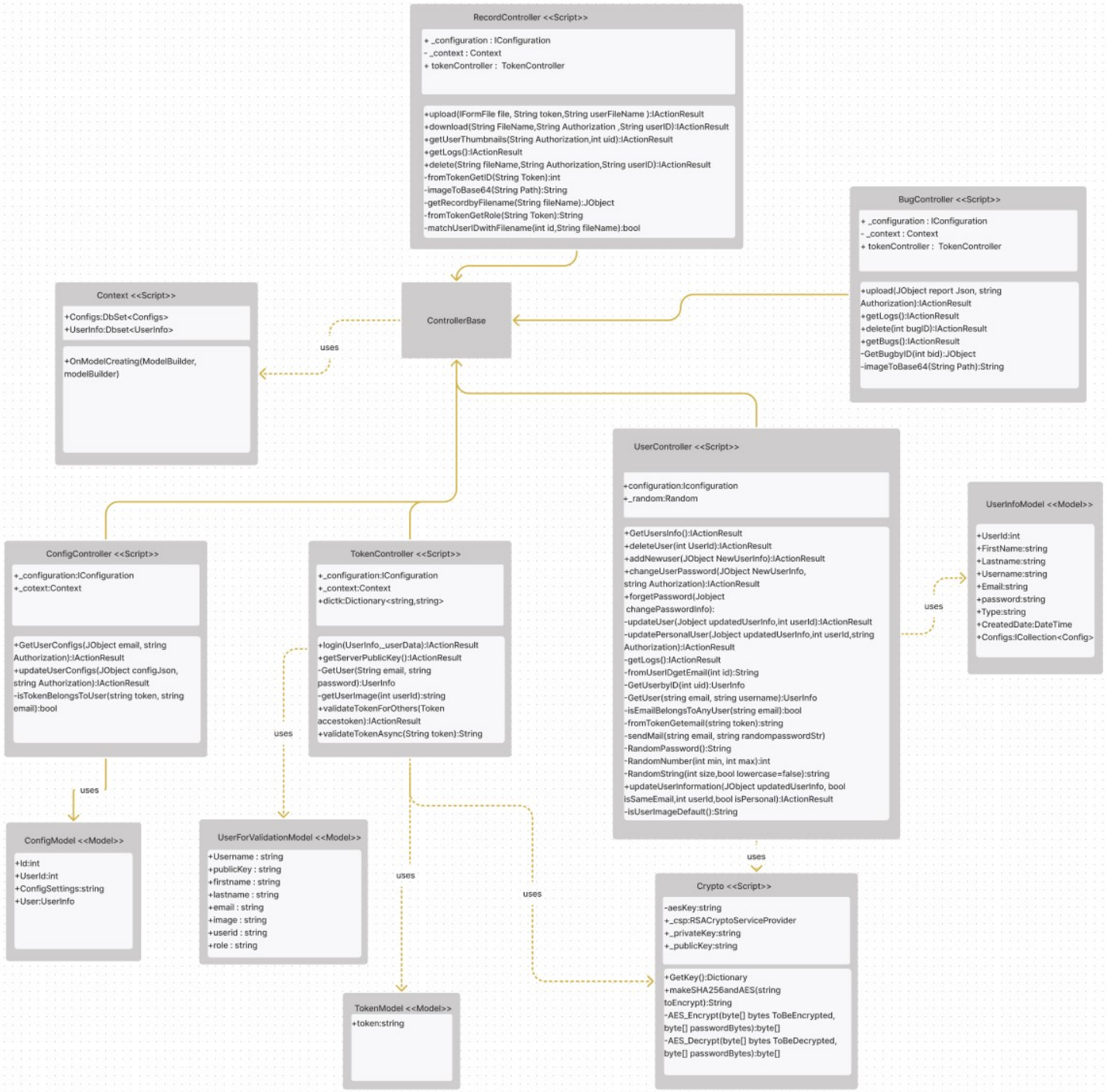
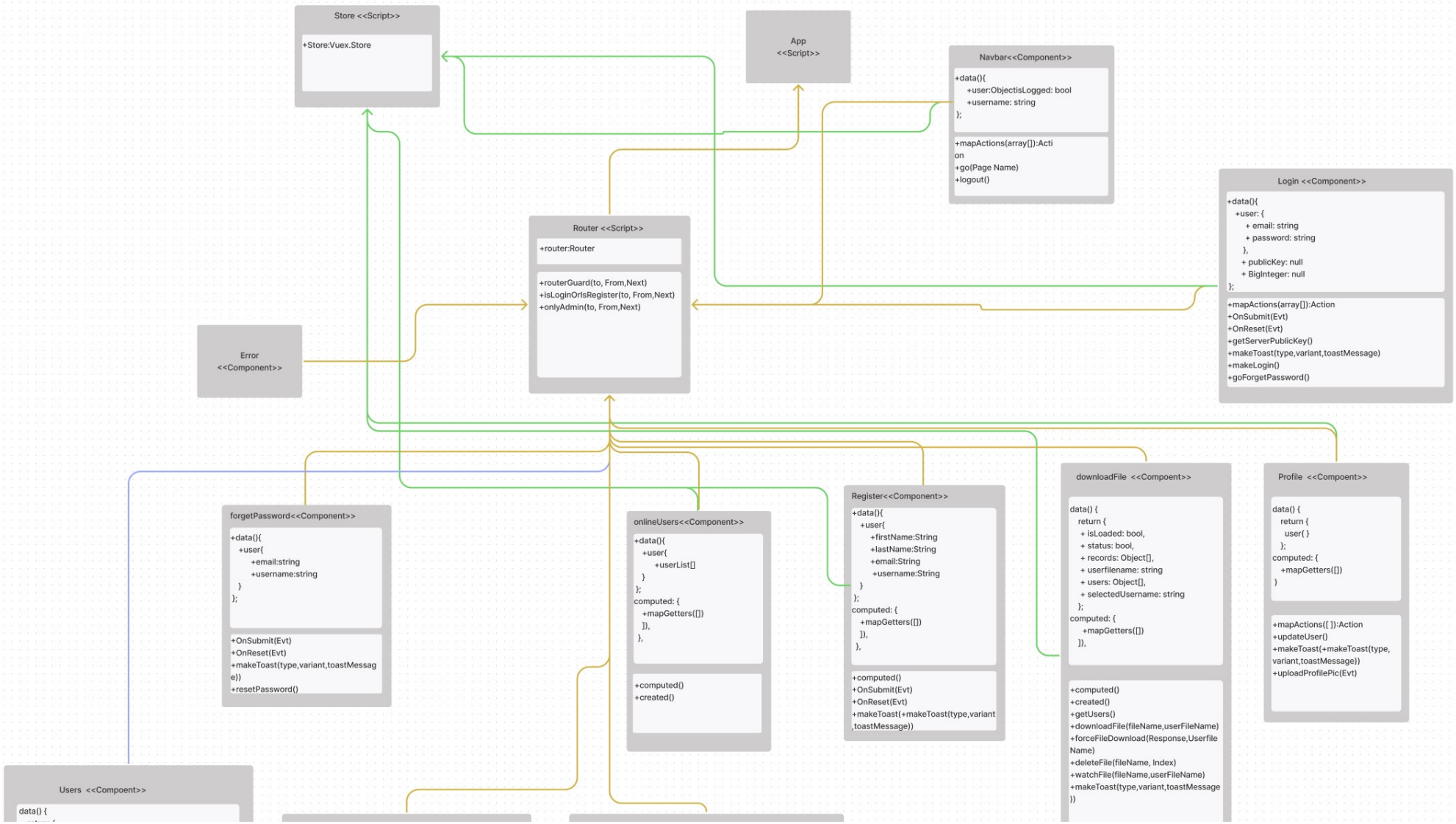

DroneMovementScript <<Script>>
+maxforwardspeed:int
+maxsidewayspeed:int
+slowdowntime:float
+forceuphover:float
+forcedownhover:float
+sidemovementspeed:float
+movemetforwardspeed:float
+rotationamount:float
+tiltmovementspeed:float
+droneSound : AudioSource
-startWeight : float
-speed_UI : Text
+forward:KeyCode
+backward:KeyCode
+rightward:KeyCode
+leftward:KeyCode
+downward:KeyCode
+rotateright:KeyCode
+rotateleft:KeyCode
+barrelrollright:KeyCode
+barrelrollleft:KeyCode
+swingright:KeyCode
+swingleft:KeyCode
+ourDrone : RigidBody
-battery : float
-currentBattery : float
-flightTime : float
-batteryUI : Text
+Awake()
+getVelocity()
+BasicDroneHoverAndRotation()
+MovementUpDown()
+MovementLeftRight()
+Rotation()
+MovementForward()
+TrackSpeed()
+SnowWeight()
+Animations()
+DroneSound()
+RotationUpdateLoop_TrickRotation()
+batteryController()
+trackBattery()
GameManager <<GameObject>>
DroneSelection <<Script>>
+nameLabel:Text
+droneIndex:int
+skinIndex:int
+selectableDrones:List<<SelectableDrone>>
-Start()
-RefreshSelection()
+ColorChanger(int colorIndex)
SelectableDrone<<Script>>
+skin1:GameObject
+skin2:GameObject
+skin3:GameObject
+skin4:GameObject
Propellers <<Script>>
+idleRotationSpeed:float
+movingRotatiioSpeed:float
+elisaAgle:float
+spinDifference:bool
-currentYRotation:float
-rotationSpeed:float
-amountOfWingtipVorticesOnElisas:int
-wantedAlpha:float
-currentAlpha:float
-dronemovementscript:DroneMovemetScript
+Awake()
+RotationInputs()
+RotationDifferentials()
MovingPlatforms <<Script>>
-leftCap:float
-rightCap:float
-topCat:float
-botCat:float
-speed:float
-movingupdown:bool
-movingleft:bool
-movingup:bool
#platform:Transform
+Start()
+Update()
Moving PLatforms <<GameObject>>
LaunchManager <<Script>>
-_userameinputfiled:inputfield
-roomnameinputfiled:inputfield
-loginpanel:GameObject
-gamesettingpanel:GameObject
-joinroompanel:GameObject
-ActivePanel()
+OnconnectToMaster()
+OnJoinedRoom()
+CreateRoomButonClicked()
DronePropellers <<Script>>
+Awake()
+Update()
DroneMovement <<Script>>
+Awake()
+Start()
+FixedUpdate()
+Update()
Turnable <<Script>>
-MouseInputReceived() : bool
-Update()
-RotateDroneUsingMouse()
<<GameObject>> Multiplayer
GameManager Multiplayer<<Script>>
-_gamecharacterprefab:gameobject
-room:text
-_playerlist:text
+Start()
+OnRoomListUpdate(RoomInfo roomlist)
+Playerlist()
+OnPlayerEnteredRoom(Player newPlayer)
DroneCollision <<Script>>
+Sparks:gameobject
+Awake()
+OnCollisionStay(Collision other)
+SparksCleaner(Gameobject _sparks) : IEnumerator
PLAYER DRONE <<GameObject>>
+droneBody : GameObject
+droneSound : GameObject
+propellers : GameObject
+animatedGO : GameObject
<<Component>> AudioSources
+Volume:int
+Mute:boolean
+loop:bool
+pitch:float
<<Component>> Rigidbody
+mass:float
+drag:float
+angularDrag:float
+usegravity:bool
+isKieatic:bool
+freezRotation:bool
<<Component>> MeshCollider
+Convex: bool
+IsTriger:bool
+CookingOptios:enum
+Material:Physic Material
+Mesh:Mesh
<<Component>> MeshFilter
+Mesh:Mesh
<<Component>> MeshRenderer
+Materials:material[]
+castshadows:enum
+receiveshadows:bool
achorOverride:TRasform
+motionVectors:enum
+lightMapStatic:bool
MainCamera <<GameObject>>
+OnStart():DroneCamera
<<Component>> Transform
+Position:Vetor3
+Rotation:Vector3
+Scale:Vector3
<<GameObject>> Terrain

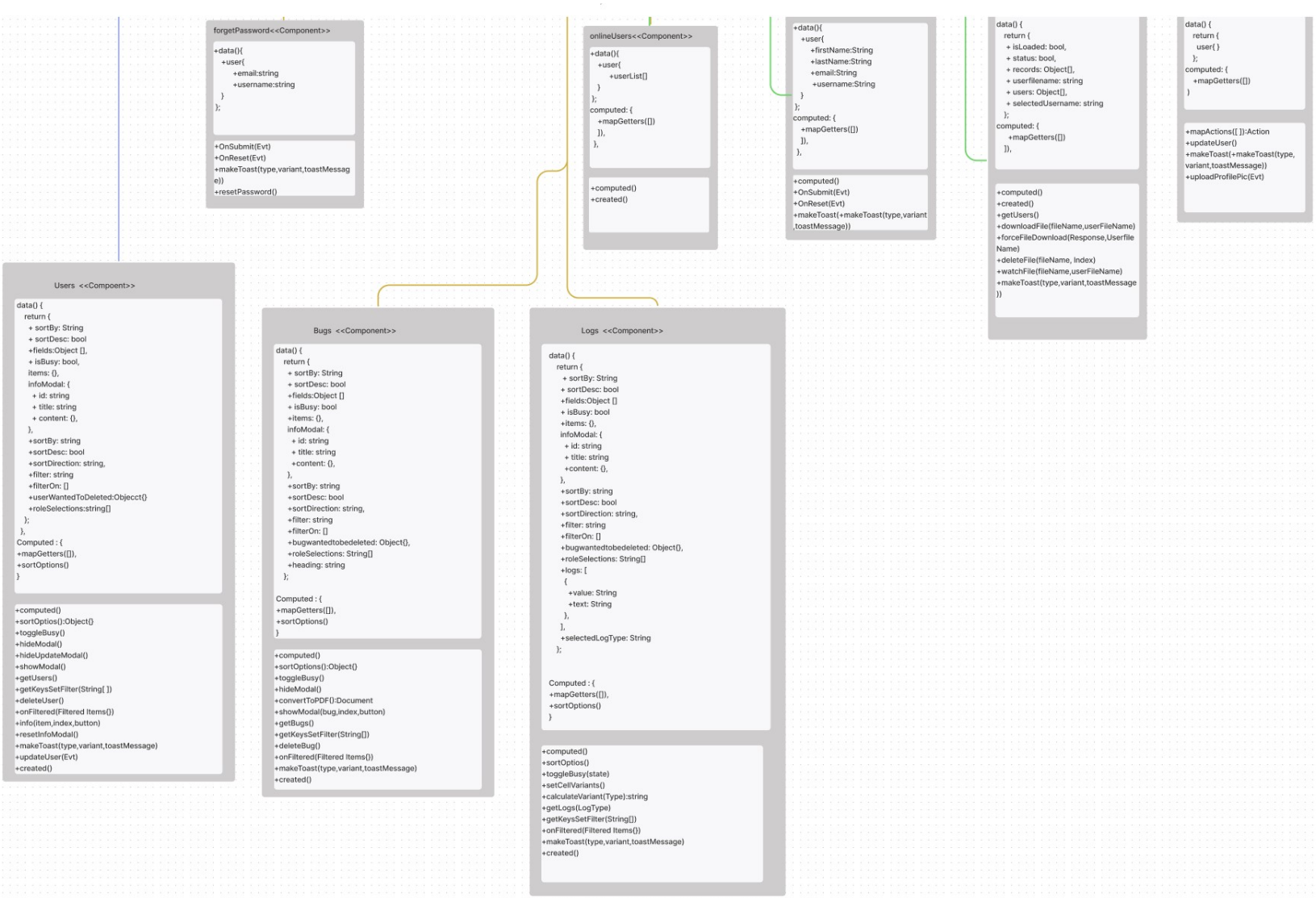

DroneCollision <<Script>>
+Sparks:gameobject
+Awake()
+OnCollisionStay(Collision other)
+SparksCleaner(Gameobject _sparks) : IEnumerator
PLAYER DRONE <<GameObject>>
+droneBody : GameObject
+droneSound : GameObject
+propellers : GameObject
+animatedGO : GameObject
<<GameObject>> Multiplayer
GameManager Multiplayer<<Script>>
-_gamecharacterprefab:gameobject
-room:text
-_playerlist:text
+Start()
+OnRoomListUpdate(RoomInfo roomlist)
+Playerlist()
+OnPlayerEnteredRoom(Player newPlayer)
<<Component>> AudioSources
+Volume:int
+Mute:boolean
+loop:bool
+pitch:float
<<Component>> Rigidbody
+mass:float
+drag:float
+angularDrag:float
+usegravity:bool
+isKieatic:bool
+freezRotation:bool
<<Component>> MeshCollider
+Convex: bool
+IsTriger:bool
+CookingOptios:enum
+Material:Physic Material
+Mesh:Mesh
<<Component>> MeshFilter
+Mesh:Mesh
<<Component>> MeshRenderer
+Materials:material[]
+castshadows:enum
+receiveshadows:bool
achorOverride:TRasform
+motionVectors:enum
+lightMapStatic:bool
MainCamera <<GameObject>>
+OnStart():DroneCamera
<<Component>> Transform
+Position:Vetor3
+Rotation:Vector3
+Scale:Vector3
<<GameObject>> Terrain
<<Component>> TerrainCollider
+Material:Physic Material
+TerrainData:TerrainData
+EnableTree Colliders:bool
Drones <<GameObject>>
+drone_red :GameObject
+drone_buzz : GameObject
+drone_racing : GameObject
+drone_splinter : GameObject
+drone_white : GameObject
WeatherChoosing <<Script>>
-weatherSelection : int
-weathers : GameObject[]
+Start()
Weathers <<GameObject>>
+snow : GameObject
+rain : GameObject
+wind : GameObject
<<GameObject>> PickUpObjects
Rotator <<Script>>
+Update()
-OncollisionEnter(Collider other)
Bot <<GameObject>>
+drone_bot : GameObject
+path : GameObject
+way_points : GameObject
DroneCamera <<Script>>
+Awake()
-FixedUpdate()
<<Component>> Camera
+ClearFlags:enum
+Background:enum
+PhysicalCamera:bool
+Depth:int
+OcclusionCulling:bool
+AllowDynamicResolution:bool
CameraScript <<Script>>
+positioInsideDrone:Vector3
+rotationInsideDrone:Vector3
+positiobehindDrone:Vector3
+VelocityCameraFollow:Vector3
+ourDrone : GameObject
+FPSTPSControl : bool
+FPSFieldOfView : float
+TPSFieldOfView : float
+Awake()
+FPSCameraPositioning()
+TPSCameraPositioning()
+FollowDroneMethod()
+TiltCameraUpDown()
+FPVTPSCamera()
DroneChoosing <<Script>>
-droneSelection : Int
-drones : GameObject[]
+Start()
OutofMapController <<Script>>
-drone: Transform
+Update()
snow <<GameObject>>
+snowParticle: ParticleSystem
rain <<GameObject>>
+rainPrefab: GameObject
wind <<GameObject>>
+eastWind: GameObject
+westWind: GameObject
+northWind: GameObject
+southWind: GameObject
BotChoosing <<Script>>
+Start():
<<GameObject>> Props
SnowControl <<Script>>
-snowPS: ParticleSystem
+Start()
+Update()
WindChoosing <<Script>>
-windDirectionSelection : Int
-winds : GameObject[]
+Start()
WindSimulation <<Script>>
+power : float
+OnTriggerStay(Collider other)
DontDestroyAudio <<Script>>
-instance: DontDestroyAudio
+Awake()
+Update()
FinishController <<Script>>
+drones : GameObject
+OnTriggerEnter(Collider other)

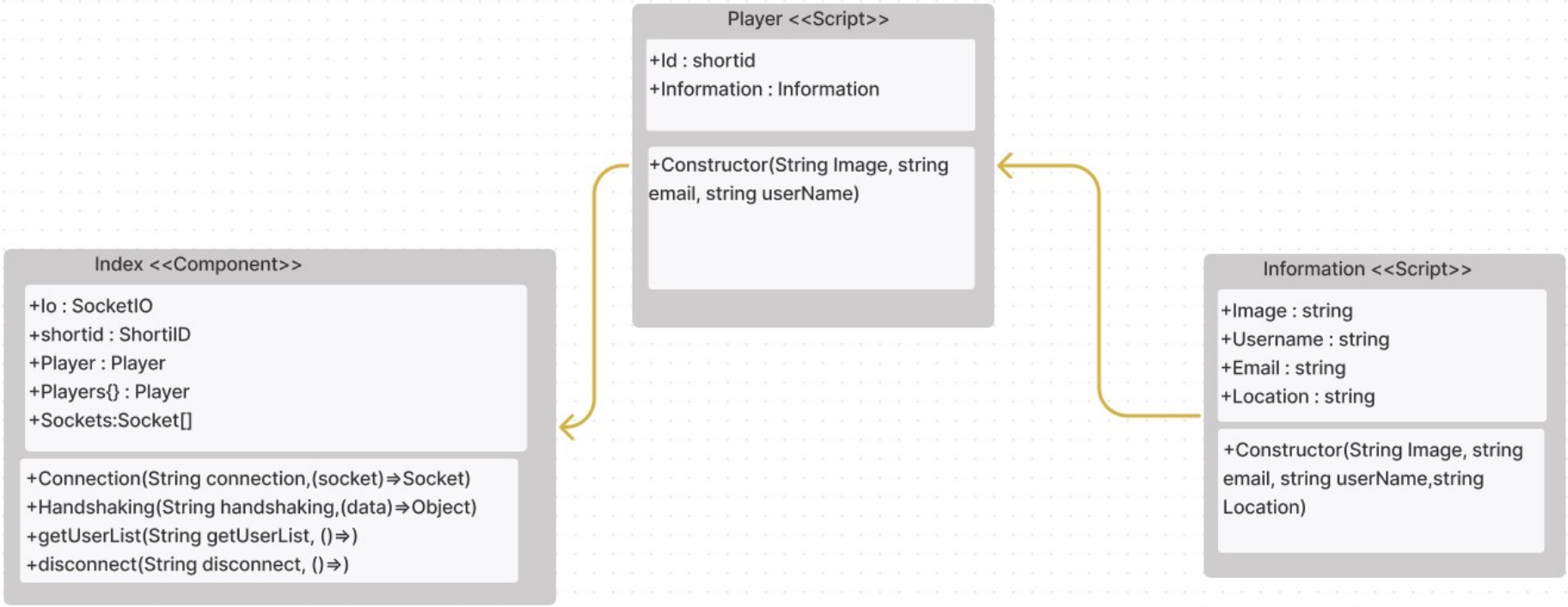

PauseMenu <<Script>>
-menu_TimeScaleRef : float
-menu_VolumeRef : float
+GameIsPaused : bool
+pauseUI : GameObject
+Start()
+Update()
+Pause()
+Resume()
InGamePanel <<Canvas>>
+timeSpeedPanel : GameObject
+pausePanel : GameObject
+InstructionPanel : GameObject
FFRecorder <<Script>>
-recordButton: GameObject
-isFirstTime: bool
+recordType: RecordType
+resoluton: Resolution
+framerate: FrameRate
+bitrate : BitRate
-outputPath : OutputPath
-customOutPutPath : string
+isRecording : bool
-FFAROS_DSHOW : string
-_ffpath : string
-_ffargs : string
_pid : int
_isRecording : bool
+ManupilateRecording()
+StartRecording()
+stopRecording()
-ParseSettings() : bool
-IERecording() : IEnumerator
-IEExitCmd() : IEnumerator
-OnDestroy()
timeSpeedPanel <<Canvas>>
+bestLapText : Text
+bestLapTime : Text
+currentLapText : Text
+currentLapTime : Text
+currentSpeedText : Text
+recordButton : Button
pausePanel <<Canvas>>
+ mainMenuButton: Button
+ resumeButton: Button
+ restartButton : Button
+ pauseText : Text
InstructionPanel <<Canvas>>
+instruction1 : GameObject
+instruction2 : GameObject
+instruction3 : GameObject
+instruction4 : GameObject
+instruction5 : GameObject
+finish : GameObject
InstructionController <<Script>>
-instructionset : Transform
-drone : Transform
+Start()
+Update()
SocketIOComponent <<Script>>
EnableDisableAI <<Script>>
+Start()
+Update()
BugPanel <<Canvas>>
+titleInputField: TMP_InputField
+contentInputField: TMP_InputField
+sendButton: Button
+uploadImageButton: Button
+goMainMenu: Button
+notJPGpanel: GameObject
+contentText: Text
+titleText: Text
TutorialMapController <<Script>>
-weatherPanel : GameObject
-AIToggle : GameObject
+Start()
+Update()
SelectionMenuPanel <<Canvas>>
+mapSelectionPanel : GameObject
+droneSelectionPanel : GameObject
+weatherSelectionPanel : GameObject
+saveButton : Button
+backButton : Button
+AIToggle : Toggle
Canvas
LoginPanel <<Canvas>>
+emailInput:InputField
+passwordInput:InputField
+loginButtonButton
+emailText:Text
+passwordText:Text
+QuitButton:Button
+emailPasswordPanel : GameObject
WeatherHolderScript <<Script>>
-SnowDensitySlider : Slider
-RainDensitySlider : Slider
-WindDensitySlider : Slider
-WindDirectionDropdown : TMP_Dropdown
+weatherChoice : Int
+Start()
+Update()
mapSelectionPanel <<Canvas>>
+nextButton : Button
+previousButton : Button
+headerPanel : GameObject
+mapHolder : GameObject
droneSelectionPanel <<Canvas>>
+nextButton : Button
+previousButton : Button
+headerPanel : GameObject
+droneHolder : GameObject
+customizeButton : Button
weatherSelectionPanel <<Canvas>>
+nextButton : Button
+previousButton : Button
+headerPanel : GameObject
+weatherHolder : GameObject
NextSceneScript <<Script>>
-mapSelection : int
-skipTutorial : GameObject
+Exit()
+GoAScene(string sceneName)
+GoMapScenes()
+SkipTutorial()
MenuPanel <<Canvas>>
+notMP4panel:GameObject
+welcomeText : Text
+logoutButton : Button
+playButton : Button
+settingsButton : Button
+uploadButton : Button
+tutorialReminderPanel : GameObject
TutorialReminderPanelScript <<Script>>
+Start()
MainSettingsPanel <<Canvas>>


SelectionMenuPanel <<Canvas>>
+mapSelectionPanel : GameObject
+droneSelectionPanel : GameObject
+weatherSelectionPanel : GameObject
+saveButton : Button
+backButton : Button
+AiToggle : Toggle
+titleInputField: TMP_InputField
+contentInputField: TMP_InputField
+sendButton: Button
+uploadImageButton: Button
+goMainMenu: Button
+notJPGpanel: GameObject
+contentText: Text
+titleText: Text
+Start()
+Update()
LoginController <<Script>>
+emailInput:InputField
+passwordInput:InputField
+loginButton:Button
+loginModel:LoginModel
+emailPasswordPanel : GameObject
+isAuthenticated : bool
+Start()
-checkLogin()
-isAutoLoginOn() : bool
-MakeAutoLogin() : IEnumerator
-makeLogin()
-getServerPublicKey(string CommonVariables.servePublicKeyUrl) : IEnumerator
-authenticate(string url, LoginModel info, string email) : IEnumerator
-SaveTokenTimeEmail(string token , string email)
-saveLastLogin()
-deleteInvalidTokenEmail()
-controlEmailAndPassword(LoginModel info): bool
LoginModel <<Model>>
+email:string
+password:string
Canvas
LoginPanel <<Canvas>>
+emailInput:InputField
+passwordInput:InputField
+loginButton:Button
+emailText:Text
+passwordText:Text
+QuitButton:Button
+emailPasswordPanel : GameObject
WeatherHolderScript <<Script>>
-SnowDensitySlider : Slider
-RainDensitySlider : Slider
-WindDensitySlider : Slider
-WindDirectionDropdown : TMP_Dropdown
+weatherChoice : int
+Start()
+Update()
droneSelectionPanel <<Canvas>>
+nextButton : Button
+previousButton : Button
+headerPanel : GameObject
+droneHolder : GameObject
+customizeButton : Button
weatherSelectionPanel <<Canvas>>
+nextButton : Button
+previousButton : Button
+headerPanel : GameObject
+weatherHolder : GameObject
NextSceneScript <<Script>>
-mapSelection : int
-skipTutorial : GameObject
+Exit()
+GoAScene(string sceneName)
+GoMapScenes()
+SkipTutorial()
MenuPanel <<Canvas>>
+notMP4panel:GameObject
+welcomeText : Text
+logoutButton : Button
+playButton : Button
+settingsButton : Button
+uploadButton : Button
+tutorialReminderPanel : GameObject
+reportBugbutton : Button
+exitButton : Button
TutorialReminderPanelScript <<Script>>
+Start()
MainSettingsPanel <<Canvas>>
+settingsPanel : GameObject
+settingsHeaderPanel : GameObject
+ControlPanel : GameObject
+ResolutionPanel : GameObject
uses
BugController <<Script>>
+titleInputField: TMP_InputField
+contentInputField: TMP_InputField
+sendButton: Button
+uploadImageButton: Button
+goMainMenu: Button
+notJPGpanel: GameObject
+sendBug()
+UploadFileCo(string localFileName, string uploadUrl)
-ConvertToBase64(string path) : string
SelectionButton <<Script>>
-currentDroneObject : int
-currentMapObject: int
-currentWeatherObject : int
-previousButton : Button
-nextButton : Button
+Start()
-SelectObject(int _index)
-SelectDroneObject(int _index)
+ChangeDroneObject(int _change)
+ChangeMapObject(int _change)
+ChangeWeatherObject(int _change)
WelcomeTextScript <<Script>>
-textBox:Text
+Start()
+Update()
ControlPanel <<Canvas>>
+forwardInputField:InputField
+backwardInputField:InputField
+rightwardInputField:InputField
+leftwardInputField:InputField
+downwardInputField:InputField
+rotaterightInputField:InputField
+rotateleftInputField:InputField
+barrelrollrightInputField:InputField
+barrelrollleftInputField:InputField
+swingrightInputField:InputField
+swingleftInputField:InputField
SettingsPanel <<Canvas>>
+BackToMainMenuButton : Button
+SaveButton : Button
+AutoLoginToggle : Toggle
ResolutionPanel <<Canvas>>
+resolutionDropDown:Dropdown
+soundSlider:Slider
+musicSlider:Slider
SettingsHeaderPanel <<Canvas>>
+SettingsHeaderText : Text
+ControlSettingsButton: Button
+ResolutionSettingsButton
CommonVariables<<Model>>
+userSettinsFilePath:string
+loginInfoFilePath:String
+getTokenUrl:String
+validateTokenUrl:String
+getUserSettingsurl:String
+saveUserSettingsurl:String
+serverPublicKeyUrl:String
+serverMP4UploadURL:String
+serverBugReportURL:String
+userEmailNodePath:String
+lastLoginNodepath:String
+tokenNodePath:String
+autoLoginNodePath:String
+musicNotePath:String
+userDetailsJobject
+rsa:RSACryptoServiceProvider
MenuController <<Script>>
+notMP4panel : GameObject
+Start()
-getSetUserSettings() : IEnumerator
-changeUser()
-getMusicVolume()
-applyGameSettings()
+UploadRecord()
+UploadFileCo(string localFileName, string uploadUrl) : IEnumerator
Uses
PanelManager <<Script>>
+ _controlPanel : GameObject
+ _controlpanelgameobjects : GameObject[]
+ _resolutionPanel : GameObject
+ _resolutionpanelgameobjects : GameObject[]
+Start()
+ActiveControlPanel()
+ActiveResolutionPanel()
GameSettings <<Script>>
+userSettings:JObject
+isUserLoggedI:bool
-IsSettingsFileExist()
-loadUserSettingsToFile(JObject settingsJson)
SettingMenuController <<Script>>
+resolutionDropDown:Dropdown
+soundSlider:Slider
+musicSlider:Slider
+forwardInputField:InputField
+backwardInputField:InputField
+rightwardInputField:InputField
+leftwardInputField:InputField
+downwardInputField:InputField
+rotaterightInputField:InputField
+rotateleftInputField:InputField
+barrelrollrightInputField:InputField
+barrelrollleftInputField:InputField
+swingrightInputField:InputField
+swingleftInputField:InputField
+autoLogin:Toggle
+Start()
-loadSettings()
+ChangeKey(string key)
+SaveSettings()
+SaveSettingsToDatabase() : IEnumerator
-isAutoLoginOn() : bool
-ChangeAutoLogin(bool state)
uses


RecordController <<Script>>
+ _configuration : IConfiguration
- _context : Context
+ tokenController : TokenController
+upload(IFormFile file, String token,String userFileName):IActionResult
+download(String FileName,String Authorization ,String userID):IActionResult
+getUserThumbnails(String Authorization,int uid):IActionResult
+getLogs():IActionResult
+delete(String fileName,String Authorization,String userID):IActionResult
-fromTokenGetID(String Token):int
-imageToBase64(String Path):String
-getRecordbyFilename(String fileName):JObject
-fromTokenGetRole(String Token):String
-matchUserIDwithFilename(int id,String fileName):bool
BugController <<Script>>
+ _configuration : IConfiguration
- _context : Context
+ tokenController : TokenController
+upload(JObject report Json, string Authorization):IActionResult
+getLogs():IActionResult
+delete(int bugID):IActionResult
+getBugs():IActionResult
-GetBugbyID(int bid):JObject
-imageToBase64(String Path):String
Context <<Script>>
+Configs:DbSet<Configs>
+UserInfo:Dbset<UserInfo>
+OnModelCreating(ModelBuilder, modelBuilder)
uses
ControllerBase
UserController <<Script>>
+configuration:Iconfiguration
+_random:Random
+GetUsersInfo():IActionResult
+deleteUser(int UserId):IActionResult
+addNewuser(JObject NewUserInfo):IActionResult
+changeUserPassword(JObject NewUserInfo, string Authorization):IActionResult
+forgetPassword(Jobject changePasswordInfo):
-updateUser(Jobject updatedUserInfo,int userId):IActionResult
-updatePersonalUser(Jobject updatedUserInfo,int userId,string Authorization):IActionResult
-getLogs():IActionResult
-fromUserIDgetEmail(int id):String
-GetUserbyID(int uid):UserInfo
-GetUser(string email, string username):UserInfo
-isEmailBelongsToAnyUser(string email):bool
-fromTokenGetemail(string token):string
-sendMail(string email, string randompasswordStr)
-RandomPassword():String
-RandomNumber(int min, int max):int
-RandomString(int size,bool lowercase=false):string
+updateUserInformation(JObject updatedUserInfo, bool isSameEmail,int userId,bool isPersonal):IActionResult
-IsUserImageDefault():String
uses
UserInfoModel <<Model>>
+UserId:int
+FirstName:string
+Lastname:string
+Username:string
+Email:string
+password:string
+Type:string
+CreatedDate:DateTime
+Configs:ICollection<Config>
ConfigController <<Script>>
+_configuration:IConfiguration
+_cotext:Context
+GetUserConfigs(JObject email, string Authorization):IActionResult
+updateUserConfigs(JObject configJson, string Authorization):IActionResult
-isTokenBelongsToUser(string token, string email):bool
uses
ConfigModel <<Model>>
+Id:int
+UserId:int
+ConfigSettings:string
+User:UserInfo
TokenController <<Script>>
+_configuration:IConfiguration
+_context:Context
+dictk:Dictionary<string,string>
+login(UserInfo,_userData):IActionResult
+getServerPublicKey():IActionResult
-GetUser(String email, string password):UserInfo
-getUserImage(int userId):string
+validateTokenForOthers(Token accestoken):IActionResult
+validateTokenAsync(String token):String
uses
UserForValidationModel <<Model>>
+Username : string
+publicKey : string
+firstname : string
+lastname : string
+email : string
+image : string
+userid : string
+role : string
uses
TokenModel <<Model>>
+token:string
uses
Crypto <<Script>>
-aesKey:string
+_csp:RSACryptoServiceProvider
+_privateKey:string
+_publicKey:string
+GetKey():Dictionary
+makeSHA256andAES(string toEncrypt):String
-AES_Encrypt(byte[] bytes ToBeEncrypted, byte[] passwordBytes):byte[]
-AES_Decrypt(byte[] bytes ToBeDecrypted, byte[] passwordBytes):byte[]


Store <<Script>>
+Store:Vuex.Store
App
<<Script>>
Navbar<<Component>>
+data(){
+user:ObjectisLogged: bool
+username: string
};
+mapActions(array[]):Acti
on
+go(Page Name)
+logout()
Login <<Component>>
+data(){
+user: {
+ email: string
+ password: string
},
+ publicKey: null
+ BigInteger: null
};
+mapActions(array[]):Action
+OnSubmit(Evt)
+OnReset(Evt)
+getServerPublicKey()
+makeToast(type,variant,toastMessage)
+makeLogin()
+goForgetPassword()
Router <<Script>>
+router:Router
+routerGuard(to, From,Next)
+isLoginOrIsRegister(to, From,Next)
+onlyAdmin(to, From,Next)
Error
<<Component>>
forgetPassword<<Component>>
+data(){
+user{
+email:string
+username:string
}
};
+OnSubmit(Evt)
+OnReset(Evt)
+makeToast(type,variant,toastMessag
e))
+resetPassword()
onlineUsers<<Component>>
+data(){
+user{
+userList[]
}
};
computed: {
+mapGetters([])
]),
},
+computed()
+created()
Register<<Component>>
+data(){
+user{
+firstName:String
+lastName:String
+email:String
+username:String
}
};
computed: {
+mapGetters([])
]),
},
+computed()
+OnSubmit(Evt)
+OnReset(Evt)
+makeToast(+makeToast(type,variant
,toastMessage))
downloadFile <<Compoent>>
data() {
return {
+ isLoaded: bool,
+ status: bool,
+ records: Object[],
+ userfilename: string
+ users: Object[],
+ selectedUsername: string
};
computed: {
+mapGetters([])
],
+computed()
+created()
+getUsers()
+downloadFile(fileName,userFileName)
+forceFileDownload(Response,Userfile
Name)
+deleteFile(fileName, Index)
+watchFile(fileName,userFileName)
+makeToast(type,variant,toastMessage
))
Profile <<Compoent>>
data() {
return {
user{ }
};
computed: {
+mapGetters([])
}
+mapActions([ ]):Action
+updateUser()
+makeToast(+makeToast(type,
variant,toastMessage))
+uploadProfilePic(Evt)
Users <<Compoent>>
data() {


forgetPassword<<Component>>
+data(){
+user{
+email:string
+username:string
}
};
+OnSubmit(Evt)
+OnReset(Evt)
+makeToast(type,variant,toastMessage))
+resetPassword()
onlineUsers<<Component>>
+data(){
+user{
+userList[]
}
};
computed: {
+mapGetters([])
]),
},
+computed()
+created()
+data(){
+user{
+firstName:String
+lastName:String
+email:String
+username:String
}
};
computed: {
+mapGetters([])
]),
},
+computed()
+OnSubmit(Evt)
+OnReset(Evt)
+makeToast(+makeToast(type,variant,toastMessage))
data() {
return {
+ isLoaded: bool,
+ status: bool,
+ records: Object[],
+ userfilename: string
+ users: Object[],
+ selectedUsername: string
};
computed: {
+mapGetters([])
],
+computed()
+created()
+getUsers()
+downloadFile(fileName,userFileName)
+forceFileDownload(Response,UserfileName)
+deleteFile(fileName, Index)
+watchFile(fileName,userFileName)
+makeToast(type,variant,toastMessage))
data() {
return {
user{ }
};
computed: {
+mapGetters([])
}
+mapActions([ ]):Action
+updateUser()
+makeToast(+makeToast(type,variant,toastMessage))
+uploadProfilePic(Evt)
Users <<Compoent>>
data() {
return {
+ sortBy: String
+ sortDesc: bool
+fields:Object [],
+ isBusy: bool,
items: {},
infoModal: {
+ id: string
+ title: string
+ content: {},
},
+sortBy: string
+sortDesc: bool
+sortDirection: string,
+filter: string
+filterOn: []
+userWantedToDeleted:Objecct{}
+roleSelections:string[]
};
},
Computed : {
+mapGetters([]),
+sortOptions()
}
+computed()
+sortOptios():Object{}
+toggleBusy()
+hideModal()
+hideUpdateModal()
+showModal()
+getUsers()
+getKeysSetFilter(String[ ])
+deleteUser()
+onFiltered(Filtered Items{})
+info(item,index,button)
+resetInfoModal()
+makeToast(type,variant,toastMessage)
+updateUser(Evt)
+created()
Bugs <<Component>>
data() {
return {
+ sortBy: String
+ sortDesc: bool
+fields:Object []
+ isBusy: bool
+items: {},
infoModal: {
+ id: string
+ title: string
+content: {},
},
+sortBy: string
+sortDesc: bool
+sortDirection: string,
+filter: string
+filterOn: []
+bugwantedtobedeleted: Object{},
+roleSelections: String[]
+heading: string
};
Computed : {
+mapGetters([]),
+sortOptions()
}
+computed()
+sortOptions():Object{}
+toggleBusy()
+hideModal()
+convertToPDF():Document
+showModal(bug,index,button)
+getBugs()
+getKeysSetFilter(String[])
+deleteBug()
+onFiltered(Filtered Items{})
+makeToast(type,variant,toastMessage)
+created()
Logs <<Component>>
data() {
return {
+ sortBy: String
+ sortDesc: bool
+fields:Object []
+ isBusy: bool
+items: {},
infoModal: {
+ id: string
+ title: string
+content: {},
},
+sortBy: string
+sortDesc: bool
+sortDirection: string,
+filter: string
+filterOn: []
+bugwantedtobedeleted: Object{},
+roleSelections: String[]
+logs: [
{
+value: String
+text: String
},
],
+selectedLogType: String
};
Computed : {
+mapGetters([]),
+sortOptions()
}
+computed()
+sortOptios()
+toggleBusy(state)
+setCellVariants()
+calculateVariant(Type):string
+getLogs(LogType)
+getKeysSetFilter(String[])
+onFiltered(Filtered Items{})
+makeToast(type,variant,toastMessage)
+created()


Player <<Script>>
+Id : shortid
+Information : Information
+Constructor(String Image, string email, string userName)
Index <<Component>>
+Io : SocketIO
+shortid : ShortiID
+Player : Player
+Players{} : Player
+Sockets:Socket[]
+Connection(String connection,(socket)⇒Socket)
+Handshaking(String handshaking,(data)⇒Object)
+getUserList(String getUserList, ()⇒)
+disconnect(String disconnect, ()⇒)
Information <<Script>>
+Image : string
+Username : string
+Email : string
+Location : string
+Constructor(String Image, string email, string userName,string Location)