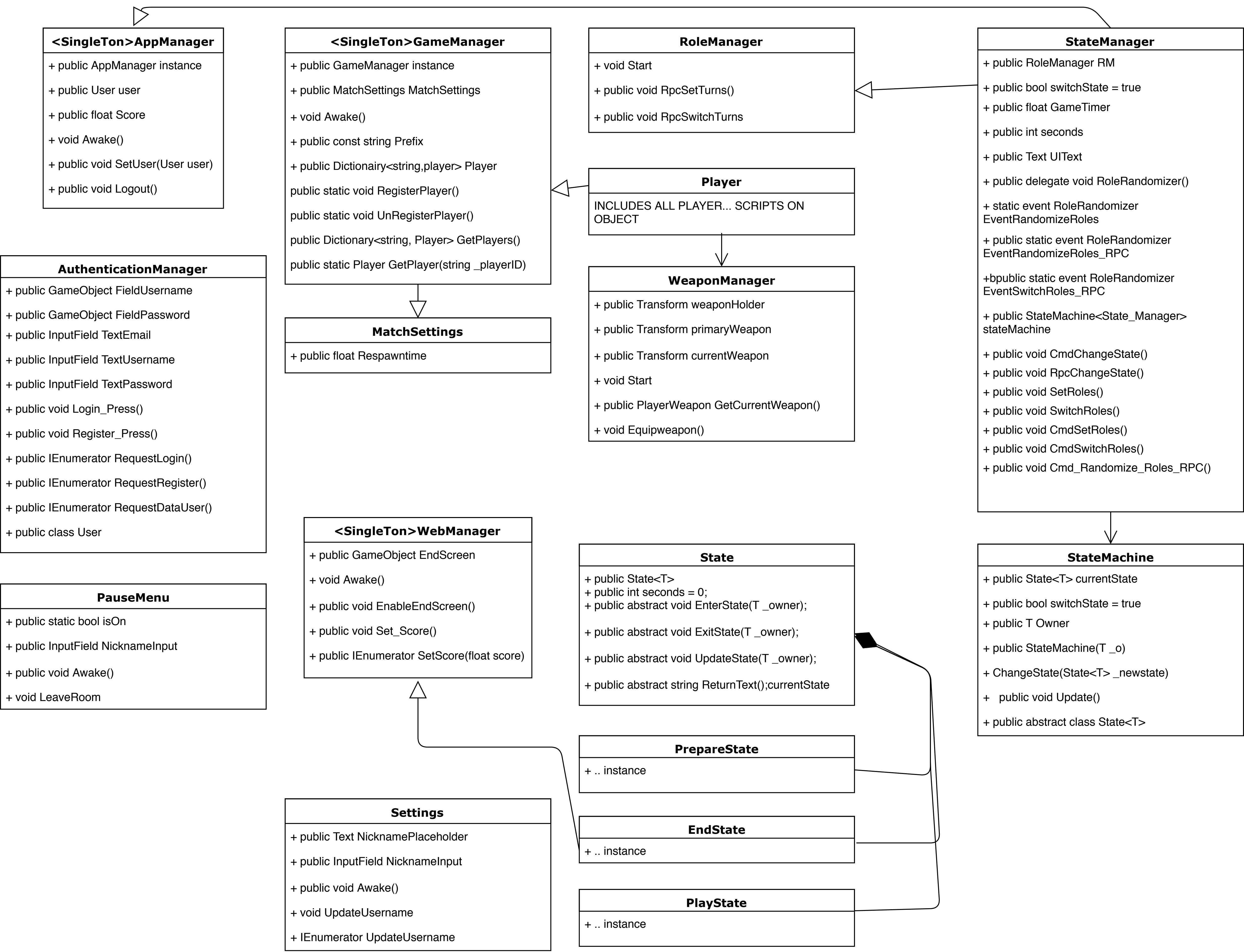
<SingleTon>AppManager
+ public AppManager instance
+ public User user
+ public float Score
+ void Awake()
+ public void SetUser(User user)
+ public void Logout()
<SingleTon>GameManager
+ public GameManager instance
+ public MatchSettings MatchSettings
+ void Awake()
+ public const string Prefix
+ public Dictionairy<string,player> Player
public static void RegisterPlayer()
public static void UnRegisterPlayer()
public Dictionary<string, Player> GetPlayers()
public static Player GetPlayer(string _playerID)
RoleManager
+ void Start
+ public void RpcSetTurns()
+ public void RpcSwitchTurns
StateManager
+ public RoleManager RM
+ public bool switchState = true
+ public float GameTimer
+ public int seconds
+ public Text UIText
+ public delegate void RoleRandomizer()
+ static event RoleRandomizer EventRandomizeRoles
+ public static event RoleRandomizer EventRandomizeRoles_RPC
+bpublic static event RoleRandomizer EventSwitchRoles_RPC
+ public StateMachine<State_Manager> stateMachine
+ public void CmdChangeState()
+ public void RpcChangeState()
+ public void SetRoles()
+ public void SwitchRoles()
+ public void CmdSetRoles()
+ public void CmdSwitchRoles()
+ public void Cmd_Randomize_Roles_RPC()
Player
INCLUDES ALL PLAYER... SCRIPTS ON OBJECT
WeaponManager
+ public Transform weaponHolder
+ public Transform primaryWeapon
+ public Transform currentWeapon
+ void Start
+ public PlayerWeapon GetCurrentWeapon()
+ void Equipweapon()
MatchSettings
+ public float Respawntime
AuthenticationManager
+ public GameObject FieldUsername
+ public GameObject FieldPassword
+ public InputField TextEmail
+ public InputField TextUsername
+ public InputField TextPassword
+ public void Login_Press()
+ public void Register_Press()
+ public IEnumerator RequestLogin()
+ public IEnumerator RequestRegister()
+ public IEnumerator RequestDataUser()
+ public class User
<SingleTon>WebManager
+ public GameObject EndScreen
+ void Awake()
+ public void EnableEndScreen()
+ public void Set_Score()
+ public IEnumerator SetScore(float score)
State
+ public State<T>
+ public int seconds = 0;
+ public abstract void EnterState(T _owner);
+ public abstract void ExitState(T _owner);
+ public abstract void UpdateState(T _owner);
+ public abstract string ReturnText();currentState
StateMachine
+ public State<T> currentState
+ public bool switchState = true
+ public T Owner
+ public StateMachine(T _o)
+ ChangeState(State<T> _newstate)
+ public void Update()
+ public abstract class State<T>
PauseMenu
+ public static bool isOn
+ public InputField NicknameInput
+ public void Awake()
+ void LeaveRoom
PrepareState
+ .. instance
EndState
+ .. instance
PlayState
+ .. instance
Settings
+ public Text NicknamePlaceholder
+ public InputField NicknameInput
+ public void Awake()
+ void UpdateUsername
+ IEnumerator UpdateUsername