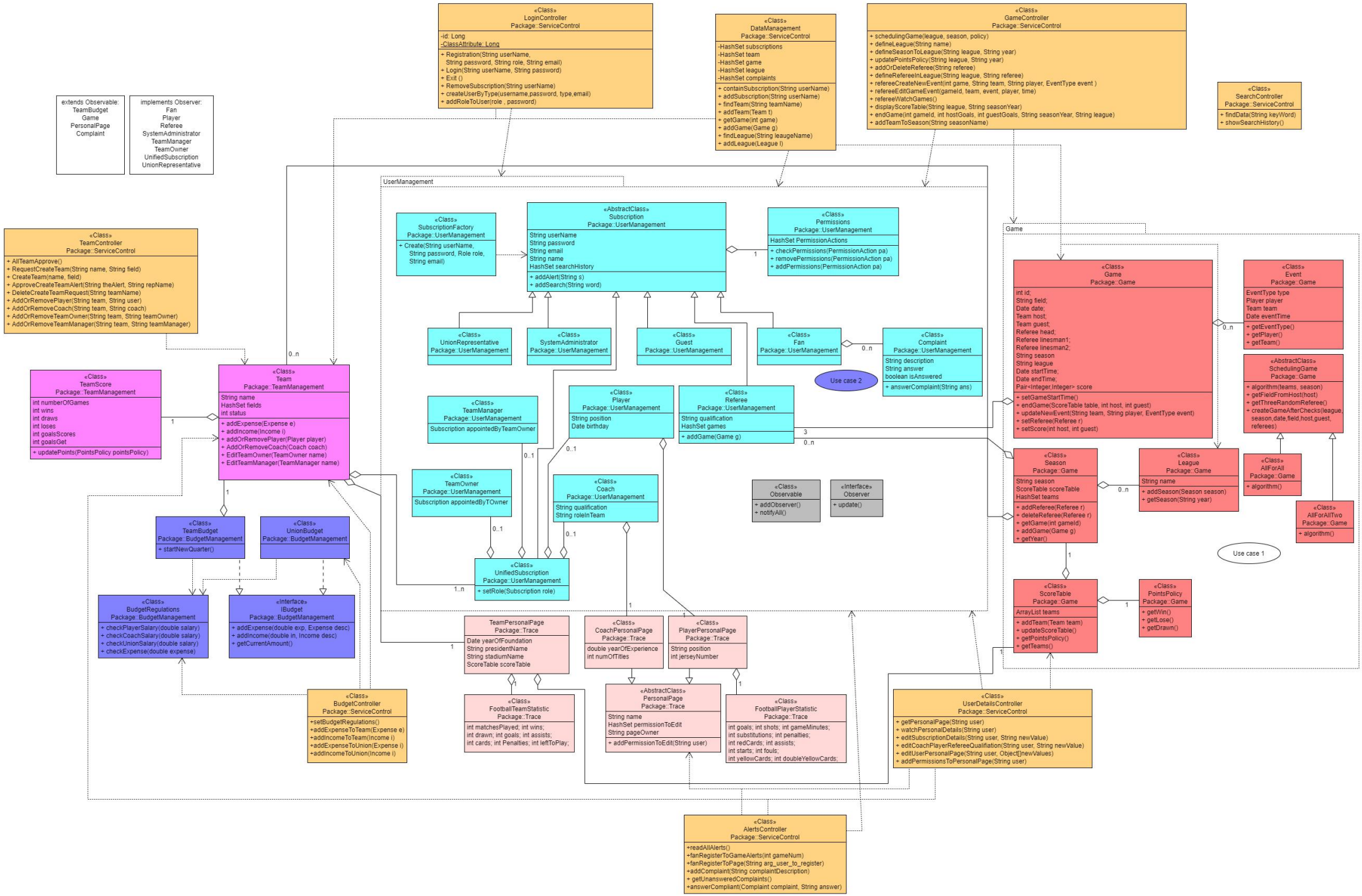
«Class»
LoginController
Package::ServiceControl
-id: Long
-ClassAttribute: Long
+ Registration(String userName, String password, String role, String email)
+ Login(String userName, String password)
+ Exit ()
+ RemoveSubscription(String userName)
+ createUserByType(username,password, type,email)
+ addRoleToUser(role , password)
«Class»
DataManagement
Package::ServiceControl
-HashSet subscriptions
-HashSet team
-HashSet game
-HashSet league
-HashSet complaints
+ containSubscription(String userName)
+ addSubscription(String userName)
+ findTeam(String teamName)
+ addTeam(Team t)
+ getGame(int game)
+ addGame(Game g)
+ findLeague(String leaugeName)
+ addLeague(League l)
«Class»
GameController
Package::ServiceControl
+ schedulingGame(league, season, policy)
+ defineLeague(String name)
+ defineSeasonToLeague(String league, String year)
+ updatePointsPolicy(String league, String year)
+ addOrDeleteReferee(String referee)
+ defineRefereeInLeague(String league, String referee)
+ refereeCreateNewEvent(int game, String team, String player, EventType event)
+ refereeEditGameEvent(gameId, team, event, player, time)
+ refereeWatchGames()
+ displayScoreTable(String league, String seasonYear)
+ endGame(int gameId, int hostGoals, int guestGoals, String seasonYear, String league)
+ addTeamToSeason(String seasonName)
«Class»
SearchController
Package::ServiceControl
+ findData(String keyWord)
+ showSearchHistory()
extends Observable:
TeamBudget
Game
PersonalPage
Complaint
implements Observer:
Fan
Player
Referee
SystemAdministrator
TeamManager
TeamOwner
UnifiedSubscription
UnionRepresentative
UserManagement
«Class»
TeamController
Package::ServiceControl
+ AllTeamApprove()
+ RequestCreateTeam(String name, String field)
+ CreateTeam(name, field)
+ ApproveCreateTeamAlert(String theAlert, String repName)
+ DeleteCreateTeamRequest(String teamName)
+ AddOrRemovePlayer(String team, String user)
+ AddOrRemoveCoach(String team, String coach)
+ AddOrRemoveTeamOwner(String team, String teamOwner)
+ AddOrRemoveTeamManager(String team, String teamManager)
«Class»
SubscriptionFactory
Package::UserManagement
+ Create(String userName, String password, Role role, String email)
«AbstractClass»
Subscription
Package::UserManagement
String userName
String password
String email
String name
HashSet searchHistory
+ addAlert(String s)
+ addSearch(String word)
«Class»
Permissions
Package::UserManagement
HashSet PermissionActions
+ checkPermissions(PermissionAction pa)
+ removePermissions(PermissionAction pa)
+ addPermissions(PermissionAction pa)
«Class»
UnionRepresentative
Package::UserManagement
«Class»
SystemAdministrator
Package::UserManagement
«Class»
Guest
Package::UserManagement
«Class»
Fan
Package::UserManagement
«Class»
Complaint
Package::UserManagement
String description
String answer
boolean isAnswered
+ answerComplaint(String ans)
Use case 2
«Class»
TeamScore
Package::TeamManagement
int numberOfGames
int wins
int draws
int loses
int goalsScores
int goalsGet
+ updatePoints(PointsPolicy pointsPolicy)
«Class»
Team
Package::TeamManagement
String name
HashSet fields
int status
+ addExpense(Expense e)
+ addIncome(Income i)
+ addOrRemovePlayer(Player player)
+ AddOrRemoveCoach(Coach coach)
+ EditTeamOwner(TeamOwner name)
+ EditTeamManager(TeamManager name)
«Class»
TeamManager
Package::UserManagement
Subscription appointedByTeamOwner
«Class»
Player
Package::UserManagement
String position
Date birthday
«Class»
Referee
Package::UserManagement
String qualification
HashSet games
+ addGame(Game g)
«Class»
TeamOwner
Package::UserManagement
Subscription appointedByTOwner
«Class»
Coach
Package::UserManagement
String qualification
String roleInTeam
«Class»
Observable
+ addObserver()
+ notifyAll()
«Interface»
Observer
+ update()
«Class»
UnifiedSubscription
Package::UserManagement
+ setRole(Subscription role)
«Class»
TeamBudget
Package::BudgetManagement
+ startNewQuarter()
«Class»
UnionBudget
Package::BudgetManagement
«Class»
BudgetRegulations
Package::BudgetManagement
+ checkPlayerSalary(double salary)
+ checkCoachSalary(double salary)
+ checkUnionSalary(double salary)
+ checkExpense(double expense)
«Interface»
IBudget
Package::BudgetManagement
+ addExpense(double exp, Expense desc)
+ addIncome(double in, Income desc)
+ getCurrentAmount()
«Class»
BudgetController
Package::ServiceControl
+setBudgetRegulations()
+addExpenseToTeam(Expense e)
+addIncomeToTeam(Income i)
+addExpenseToUnion(Expense i)
+addIncomeToUnion(Income i)
TeamPersonalPage
Package::Trace
Date yearOfFoundation
String presidentName
String stadiumName
ScoreTable scoreTable
«Class»
CoachPersonalPage
Package::Trace
double yearOfExperience
int numOfTitles
«Class»
PlayerPersonalPage
Package::Trace
String position
int jerseyNumber
«Class»
FootballTeamStatistic
Package::Trace
int matchesPlayed; int wins;
int drawn; int goals; int assists;
int cards; int Penalties; int leftToPlay;
«AbstractClass»
PersonalPage
Package::Trace
String name
HashSet permissionToEdit
String pageOwner
+ addPermissionToEdit(String user)
«Class»
FootballPlayerStatistic
Package::Trace
int goals; int shots; int gameMinutes;
int substitutions; int penalties;
int redCards; int assists;
int starts; int fouls;
int yellowCards; int doubleYellowCards;
«Class»
UserDetailsController
Package::ServiceControl
+ getPersonalPage(String user)
+ watchPersonalDetails(String user)
+ editSubscriptionDetails(String user, String newValue)
+ editCoachPlayerRefereeQualifiation(String user, String newValue)
+ editUserPersonalPage(String user, Object[]newValues)
+ addPermissionsToPersonalPage(String user)
«Class»
AlertsController
Package::ServiceControl
+readAllAlerts()
+fanRegisterToGameAlerts(int gameNum)
+fanRegisterToPage(String arg_user_to_register)
+addComplaint(String complaintDescription)
+ getUnansweredComplaints()
+answerCompliant(Complaint complaint, String answer)
Game
«Class»
Game
Package::Game
int id;
String field;
Date date;
Team host;
Team guest;
Referee head;
Referee linesman1;
Referee linesman2;
String season
String league
Date startTime;
Date endTime;
Pair<Integer,Integer> score
+ setGameStartTime()
+ endGame(ScoreTable table, int host, int guest)
+ updateNewEvent(String team, String player, EventType event)
+ setReferee(Referee r)
+ setScore(int host, int guest)
«Class»
Event
Package::Game
EventType type
Player player
Team team
Date eventTime
+ getEventType()
+ getPlayer()
+ getTeam()
«AbstractClass»
SchedulingGame
Package::Game
+ algorithm(teams, season)
+ getFieldFromHost(host)
+ getThreeRandomReferee()
+ createGameAfterChecks(league, season,date,field,host,guest, referees)
«Class»
Season
Package::Game
String season
ScoreTable scoreTable
HashSet teams
+ addReferee(Referee r)
+ deleteReferee(Referee r)
+ getGame(int gameId)
+ addGame(Game g)
+ getYear()
«Class»
League
Package::Game
String name
+ addSeason(Season season)
+ getSeason(String year)
«Class»
AllForAll
Package::Game
+ algorithm()
«Class»
AllForAllTwo
Package::Game
+ algorithm()
Use case 1
«Class»
ScoreTable
Package::Game
ArrayList teams
+ addTeam(Team team)
+ updateScoreTable()
+ getPointsPolicy()
+ getTeams()
«Class»
PointsPolicy
Package::Game
+ getWin()
+ getLose()
+ getDrawn()