



THE ORIGINALS

FootboFun System Design Document



DECEMBER 1, 2013

Ahmet Abak, Can Glaydın, aęrı Utku Akpak, Doęancan Arabacı, İbrahim Arslan

Table of Contents

1	OVERVIEW.....	3
1.1	Scope.....	3
1.2	Purpose.....	3
1.3	Intended Audience.....	3
2	DEFINITIONS.....	3
3	CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTIONS.....	4
3.1	Software Design in Context.....	4
3.2	Software Design Descriptions within the Life Cycle.....	4
3.2.1	Influences on SDD preparation.....	4
3.2.2	Influences on software life cycle products.....	4
3.2.3	Design verification and design role in validation.....	4
4	DESIGN DESCRIPTION INFORMATION CONTENT.....	4
4.1	Introduction.....	4
4.2	SDD Identification.....	4
4.3	Design Stakeholders and Their Concerns.....	5
4.4	Design Views.....	5
4.5	Design Viewpoints.....	5
4.6	Design Elements.....	5
4.6.1	Design entities.....	5
4.6.2	Design relationships.....	7
4.6.3	Design Constraints.....	7
4.7	Design Rationale.....	8
4.8	Design Languages.....	8
5	DESIGN VIEWPOINTS.....	8
5.1	Introduction.....	8
5.2	Context Viewpoint.....	8
5.2.1	Design Concerns.....	9
5.2.2	Design Elements.....	9
5.3	Composition Viewpoint.....	15
5.3.1	Design Concerns.....	15
5.3.2	Design Elements.....	15
5.4	Logical Viewpoint.....	18
5.4.1	Design Concerns.....	18

5.4.2 Design Elements18

5.5 Interface Viewpoint32

5.5.1 Design Concerns32

5.5.2 Design Element.....32

5.5.3 Example Languages33

5.6 Interaction Viewpoint.....36

5.6.1 Design Concerns36

5.6.2 Design Elements37

1 OVERVIEW

1.1 Scope

This document gives a detailed description of Footbofun Software and the solutions are described by using UML diagrams. It also shows the user interface and class diagrams, which gives the basic structure of the project. Moreover, it gives a clear understanding of how the project will be implemented. The contents of this document do not cover a fully functional system and can include some assumptions which will become certain after the implementation stage.

1.2 Purpose

The purpose of this document is to briefly explain the project design and structure of Footbofun Software. It explains the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. It also illustrates the functional and non-functional requirements which will serve as a base for the implementation of our project.

1.3 Intended Audience

Target audience of this project is the people who is interested in football and who like football simulation games.. Program supplies for stakeholders, a well-designed social-networking football game on cross-platforms, i.e. iOS, Android and Web using the same server.

2 DEFINITIONS

- Footbofun Software: A social-networking football game which is available for mobile phones and on the Internet and which requires the Internet connection and a facebook account.
- Stakeholders: Stakeholders are the organizations or the people who are interested in the software. Organizations which deal with the mobile/The Internet games or the people who are interested in and/or work for mobile/The Internet games development related business are the stakeholders for this project.
- UML Diagram: Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created, by the Object Management Group. UML 2.2 has 14 types of diagrams divided into two categories. Seven diagram types represent structural information, and the other seven represent general types of behavior, including four that represent different aspects of interactions.
- Software Requirements Specification (SRS): A complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software.
- IEEE: Institute of Electrical and Electronics Engineers
- iOS: Apple Mobile Operating System
- Android: Google Mobile Operating System
- TCP: Transmission Control Protocol
- IP: Internet Protocol
- HTTP: Hypertext Transfer Protocol
- IDE: Integrated Development Environment

- MySQL: Open-source relational database management system. It is a sequential query language.

3 CONCEPTUAL MODEL FOR SOFTWARE DESIGN DESCRIPTIONS

This clause establishes a conceptual model for this SDD. The conceptual model includes basic terms and concepts of SDD, the context in which SDD is prepared and used, the stakeholders who use them, and how they are used.

3.1 Software Design in Context

The purpose and intended use of the Footbofun software is to simulate a football player's life using the classes supported by the software. Another purpose of the software is to construct a reference game engine that can be used for the development of the games which has the same type. The project is a multiplayer social network game, which simulates football career, social life and assets of a young football player. There is no visual match simulation but it is possible to track the football match from commentator's output. An event-based simulation which is generated by statistical parameters of teams and players is used. Moreover; this project is going to support cross-platforms, i.e. iOS, Android and Web using the same server. In development process, Java and Flash technologies, native iOS and Android platforms will be used. All in all, those who will use this product will interact each other with different platform.

3.2 Software Design Descriptions within the Life Cycle

3.2.1 Influences on SDD preparation

The influence on SDD preparation is requirements that are considered in the SRS of the project.

3.2.2 Influences on software life cycle products

The SDD influences the content of several major software life cycle work products. It generally influences from SRS and the priorities of the requirements in the project may be changed and shifted in the lifecycle of the product.

3.2.3 Design verification and design role in validation

Verification and validation is the last part of the product lifecycle and SDD part of the project has the most important influence on these phases of the lifecycle of the production.

4 DESIGN DESCRIPTION INFORMATION CONTENT

4.1 Introduction

In this chapter, many subjects such as identification of SDD; identified design stakeholders and design concerns; selected design viewpoints -each with type definitions of its allowed design elements and design languages-; design views, overlays and rationale will be described.

4.2 SDD Identification

This System Design Description report is created at 01.12.2013 and it is first design description created for this software.

Scope of this document is to give a clear understanding of how the project will be implemented.

This SDD is prepared to the STD; IEEE 1016-2009.

Star UML is used for all kinds of diagrams.

4.3 Design Stakeholders and Their Concerns

In this section, we are going to describe the primary stakeholders, secondary stakeholders and their concerns about the project. Stakeholders are the people, test teams and organizations who are involved or interested in software. In this project, our main stakeholder is SNG BILISIM INC.

SNG BILISIM INC is a company which develops social networking games for mobile phones and Web. Their concerns about this project are identified below:

- The GUI of the software shall be clean, simple, and understandable.
- Server part of software should be easily used with similar kind of games.
- To change some features in software, changing server side shall be enough.
- Software should be easily maintained.
- User statistics and users' character information should be placed in seperated databases.
- Software should be adapted to new technologies.
- Server part of software should be scalable.

4.4 Design Views

Design views of this SDD are design rational, contextual, compostion, interface, logical and interaction views.

4.5 Design Viewpoints

A design viewpoint addresses a different perspective to be focused on to effectively encompass requirements that have been previously created and to identify the stakeholders as to which these requirements are relevant. It is defined in the IEEE 1016-2009 Standard as “The specification of the elements and conventions available for constructing and using a design view.” There are six design viewpoints which have been used to address the range of design concerns that have been recognized.

4.6 Design Elements

4.6.1 Design entities

4.6.1.1 *Game Serving System*

This system is called “Game Server”. It is a system which serves main game logic and connections. Users and admins shall be connected through this system.

4.6.1.2 *Data Mining Server System*

“Data Mining” server is another server part of the software. It's a serving system, which aims to let authorized data analysts to work on software and game statistics and generate reports.

4.6.1.3 *Client Systems*

Client side of the software will be developed for iOS, Android and web browsers. Client softwares should be implemented as interface of the server part. Clients will be implemented only for Game Server and they will be available to Facebook users.

4.6.1.4 Game Logic Component

The component called “Game Logic” aims to calculate every computation related to game itself. Game logic component includes User, Character, Statistics, Finance, Match and Social components. These sub-components’ aim is to simplify and organize game logic component.

4.6.1.5 I/O Handling Component

“I/O Handling” component aims to manage connections to/from server. Users and admins will be connected to all other components through I/O Handling component.

4.6.1.6 Database Connection Component

“Database Connection” component is responsible for all necessary connections to/from database. It aims to simplify and organize connections with databases.

4.6.1.7 Game Logic Managers

Game Logic Managers is a general description of subsystems called “Managers” -resides in Game Logic Component-; which aims to connect subsystems in Game Logic Component. These managers are “Football Manager”, “Admin Manager” and “Finance Manager”. They all work as utility/singleton classes and are responsible of different operations types.

4.6.1.8 Apache HTTP Server

“Apache Server” is a framework created by Apache Software Foundation. On server parts of software it shall be used as a HTTP server.

4.6.1.9 Couchbase Database

“Couchbase” database is a data storing system, which has NoSQL type storing option. Because of this feature it is faster than MySQL databases, but its not possible to make any queries. It will be used to store users’ character informations and general game informations.

4.6.1.10 MySQL Database

“MySQL” database is a common way to store data. It’s possible to make efficient queries. MySQL database will be used to store action of users and software. These datas will be calculated by data analysts through data mining server.

4.6.1.11 Google Json Library

“GSON” is a library created by Google. With help of this library its possible to store any kind of objects as Json strings. Sending and receiving objects will be accomplished as Json strings.

4.6.1.12 Facebook Graph API

“Facebook Graph API” is a general library created by Facebook. With this API it is possible to fetch general information of users from Facebook and authorize login through Facebook. The software will be available only for Facebook users. (No Facebook authorization is needed for Admins and Data Analysts). All operations related to Facebook will be accomplished through this API.

4.6.1.13 AIDE

“AIDE” is a Integrated Development Environment created by Android. It stands for Android IDE. All necessary features of this IDE will be used for Android client of the software. All necessary Java libraries in AIDE shall be used.

4.6.1.14 Native iOS Frameworks

iOS frameworks for iOS client software will be used by developer team, when necessary.

4.6.1.15 Java Program

“Java” is a programming language. Developer team shall use Java for all server parts of software and Android client of the software.

4.6.1.16 Objective-C Program

“Objective-C” is a programming language created by Apple. Developer team shall use Objective-C for iOS client side of the software.

4.6.2 Design relationships

As described above, main systems related to the software are Game Server, Data Mining Server, Client Systems, Database Connection Component, I/O Handler Component, Game Logic Components, Databases and Game Logic Managers.

I/O Handler Component is responsible to correspond Game Server with Client Systems.

Database Connection Component is responsible to correspond Game Logic Managers with database.

Game logic managers (Football, Finance, Admin) are responsible to correspond Game Logic Components to each other.

Other less important relationships between classes of components will be explained in Logical Viewpoint part.

4.6.3 Design Constraints

All design entities described above should be implemented. While implementing those design entities relationships between them should be provided.

Software must be implemented fully object oriented and reusable.

The classes should be well defined, separated and understandable.

4.7 Design Rationale

The software includes very important amount of parameters and different classes. This leads to a very complicated software architecture. This design documentation aims to group, organize and simplify all classes as sub-systems and systems. All subsystem and systems should be created, so every component should have a set of specific operations on related datas. Components' operations shall never affect each other directly. They may affect each other through manipulating datas with help of Manager sub-systems.

This design also aims to create a generic approach to similar kind of software. Because of this, the design differs game logic from other general server operations. With changing game logic system and reimplementing necessary I/O handlers; it is possible to create a similar game software.

4.8 Design Languages

While creating this documentation and designing the software, UML is used. To use UML, StarUML program is chosen.

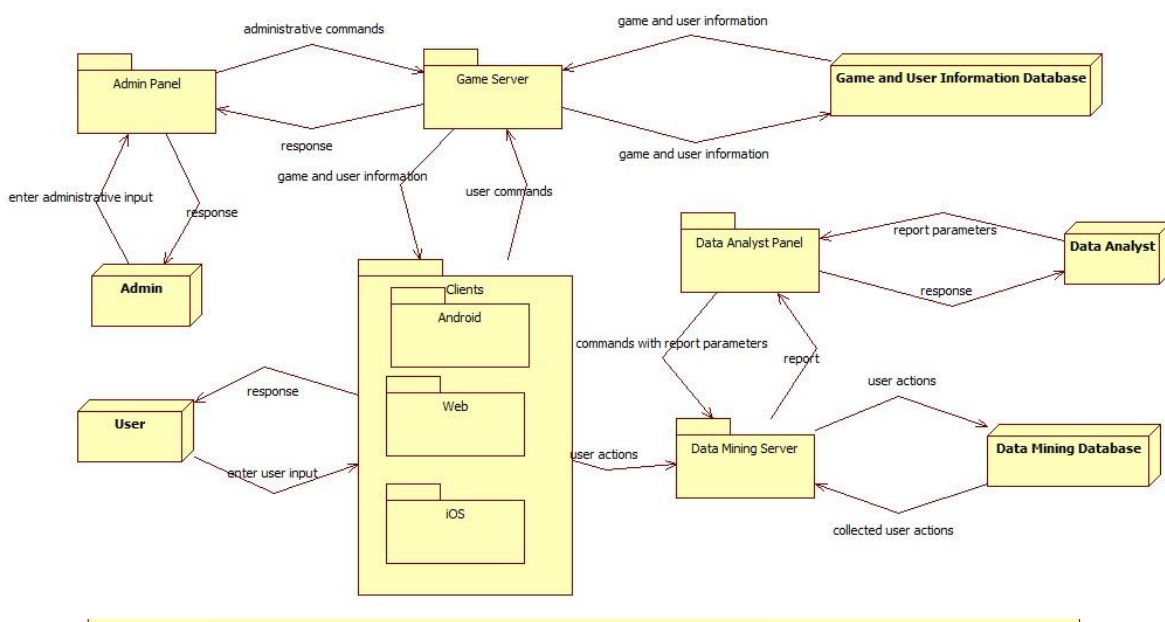
5 DESIGN VIEWPOINTS

5.1 Introduction

In this part, using visual and modelling programs, the design viewpoints of FootboFun Software is illustrated. With the help of UML tools, the content of viewpoints are demonstrated.

5.2 Context Viewpoint

A Context Data Flow Diagram is used to provide clear separation of actors and their role within this system.



5.2.1 Design Concerns

There are three main service categories concerning to our product. Each of these categories are related with their actors. The main service categories are User, Administrator and Data Analyst actions.

Users are people who will play the game. They are going to interact with different clients of the final product. Administrators are controllers of the game who can change the game parameters, ban users and create events and promotions. Data analysts are people who are going to use data mining module to generate reports about user actions.

5.2.2 Design Elements

Design entities: Design entities are actors and actions of the game.

Actors of the product are Users, Administrators, and Data Analysts.

5.2.2.1 User Functions

User functions section explains what a user can do in the game.

5.2.2.1.1 Login

When user starts the application on a mobile device, he should see login screen first. Login screen should clarify that user can only login via Facebook. Once user logged in with his Facebook account, it should be saved on device and after that login screen won't be available again.

For a user, who reaches the application on a web browser should launch the game from Facebook only. There won't be any login screen, since users are already logged in.

5.2.2.1.2 Managing Living Area

There are 6 main screens (Football player's life's main parts to manage.) in our game.

In living area section user shall see his ambit, house and car. And also he shall interact with those.

5.2.2.1.2.1 Rest

The character in the game should have an energy attribute, which goes down after matchs, trainings and some events such as, going out at night. From house image, he shall let rest his character.

Depending on confort level of the house, the character should rest faster or slower.

5.2.2.1.2.2 Have a Party

With interacting the house, a user shall throw a party in his house. Parties have a regeneration time, which is maximum 18 hours. He can reduce this time by some products from shop.

After the party, user's morale should be higher but energy should be lower.

If user has no girlfriend during this party, he can meet one.

5.2.2.1.2.3 Single Training

If the user has necessary equipment in his house, he shall train himself. Training should reduce his energy but also increase his stamina level.

Single trainings won't be effective as much as team trainings.

Single training should have a cooldown time, which is maximum 3 hours. He can reduce this time by some products from shop.

5.2.2.1.2.4 Buy/Sell Car

When user clicks/touches his car in front of his house, he should be redirected to car shop. In car shop he can buy a new car.

Buying a better car should increase his morale.

The user can have only one car at a time. If he buys a new one, the old car should be automatically sold.

5.2.2.1.2.5 Recreational Activity

Recreation activities includes watching TV, reading a book, playing console games etc.

When user choose to spend recreation time, his morale should be higher. His energy won't be effected.

Recreation time has a cooldown time, which is maximum 3 hours. He can reduce this time by some products from shop.

5.2.2.1.2.6 Buy/Sell Furniture

When user interacts with his furniture part, he should be redirected to furniture shop. In there, he can buy a new furniture set.

Buying a better furniture set should increase effect of party, rest and training events.

The user can have only one furniture set at a time. If he buys a new one, the old set should be automatically sold.

5.2.2.1.2.7 Buy/Sell House

The user shall buy a new house any time. He can be redirected to house shop from living area screen. New house can result change in ambit change. There are 3 available ambits:Suburbs, Skyscrapers and Mansions.

A better house should increase his morale regeneration and he should have more furniture set options.

The user can have only one house at a time. If he buys a new one, the old house should be automatically sold.

5.2.2.1.3 Managing Social Life

In social life section, user should see relation status of his family, his friends from team and his girlfriend if he has one. Moreover, he'll have a chance to go out or make charity work from this screen.

5.2.2.1.3.1 Date With Girlfriend

If he has a girlfriend, he shall go out with his girlfriend. After he chooses to date with her, he'll be able to choose which place to go. Going better places should increase his morale and relationship status with his girlfriend.

There should be a cooldown time, which is maximum 3 hours. He can reduce this time by some products from shop.

5.2.2.1.3.2 Buy a Gift to Girlfriend

User can buy a gift to his girlfriend at any time. Depending on value of the gift, his relation status with his girlfriend should be increased. If he buys a cheap gift, the girlfriend may be upset, which results to reduce of relation status.

5.2.2.1.3.3 Break Up With Girlfriend

When you have a girlfriend, it's impossible to meet a new one. That's why when user decides to change his girlfriend, first he should break up with his old one.

Breaking up with girlfriend should decrease player's morale.

5.2.2.1.3.4 Go Out With Family

By interacting with family, the user can choose to go out with his family. This event should increase both player's morale and his relation status with his family.

Going to more expensive places should effect increase amount. He can meet a new girlfriend in such events.

5.2.2.1.3.5 Go Out With Team Friends

By interacting with family, the user can choose to go out with his team friends. This event should increase both player's morale and his relation status with his team friends.

A better relationship with his teammates should result in more chance to get pass from them.

Going to more expensive places should effect increase amount. He can meet a new girlfriend in such events.

5.2.2.1.3.6 Charity Work

From socail life screen, the user should be available to make some charity works.

Attending to charity works, should increase morale and fans of the player. He can meet a new girlfriend in such events.

5.2.2.1.4 Managing Club Life

In club life screen, user should see information about his team. All events, related to team should be handled in this screen.

5.2.2.1.4.1 Team Training

The user is able make one training match in a day. This match should affect his chances to be in line up.

Training should decrease energy of the player.

After training he may gain some skill points.

5.2.2.1.4.2 Interact With Club Managers

User shall interact with club manager. He shall ask for better facilities and a better contract. There should be a cool down time, which is maximum 3 hours. He can reduce this time by some products from shop.

Club administartion should answer the player immediately.

5.2.2.1.4.3 Choose Game Type

The user should have the opportunity to choose his game style. He can choose between defensive/offensive, passive/aggressive, selfish/sharing and hard-working/lazy.

Every game style should affect matches differently. It should be possible to read explanations of game styles in this screen.

5.2.2.1.5 Manager Room Functions

In manager room, the user should see his own manager. He can have managers with different levels and abilities. Better managers can find better teams or do better investments.

In this room, the user can manage sponsors, investments and transfer options.

5.2.2.1.5.1 Open Transfer Menu

The user shall see transfer menu from this scene. In transfer menu, he can do following: Inspect a team, request for transfer.

5.2.2.1.5.2 Inspect Team

The user shall search for all teams in his world. He can see empty slots of a team, team level and basic information of the related team.

5.2.2.1.5.3 Request Transfer

While inspecting teams, if the users thinks that he can transfer to a team; he'll be able to make a transfer request to that team. If his manager is good enough, he should receive a contract from that team in same day.

5.2.2.1.5.4 Accept/Decline Sponsor Proposals

In manager room scene, he can see contracts from possible sponsorship agreements. He can read conditions of the contract and decide to accept or decline the contract.

5.2.2.1.5.5 Invest

The player's manager should manage his investments. The player should see possible and ongoing investments belong to him. On possible investments, there should be a risk possibility value, which shows possible success rate of the investment. Based on this, user can choose to invest or not.

Investments' success rate also depend on player's manager's skills. A better manager should result in better investments and more profit.

5.2.2.1.5.6 Undo an Investment

The player should see list of his current investments. From this list he can undo his investment. After that, the investment won't be visible to player anymore.

5.2.2.1.6 Manage Character

In this section, the player should see his own character, his statistics, skill points and his trophies.

5.2.2.1.6.1 Buy Clothes/Accessory

By interacting character's clothes/accessory he shall go to clothes/accessory shop. Here can buy new items, which increase his morale and fans.

5.2.2.1.6.2 Buy Energy Drink

The user should see amount of energy drinks, he owns. If he wants to buy more, he can go to shop and get more energy drink.

Energy drinks are used to increase energy amount of the player.

5.2.2.1.6.3 Use Energy Drink

The user should see his energy level and amount of energy drinks in character section. He can choose to use one energy drink to increase his energy.

5.2.2.1.6.4 Change Face Style/Hair Style

The user shall go to hairdresser and change his hair and beard.

Changing his style to a better style should result in morale and fan increase.

5.2.2.2 Admin Functions

Admin functions section explains what an admin can do in the game system.

5.2.2.2.1 Add/Remove Tournaments

Admins can directly interact with server and add/remove tournaments. This gives the administrators the ability to create custom tournaments and promotions. For instance, the administrator can create a limited

size tournament or lower the price of an object for a limited period while the game is active. The increases the administrator control over the game and also increases the interaction inside the game.

5.2.2.2.2 Change Game Parameters

This action gives data analyst ability to generate dynamic reports based on their specifications. These reports in going to be generated using all the automatically stored actions of all the users playing the game. The accuracy and information content of reports based on the number of players using the game.

5.2.2.2.3 Ban User

If an admin is sure that a user cheats, he should be able to ban him with his unique ID.

5.2.2.3 Data Analyst Functions

Data analyst functions section explains what a data analyst can do in the game system.

5.2.2.3.1 Generate Data Mining Report

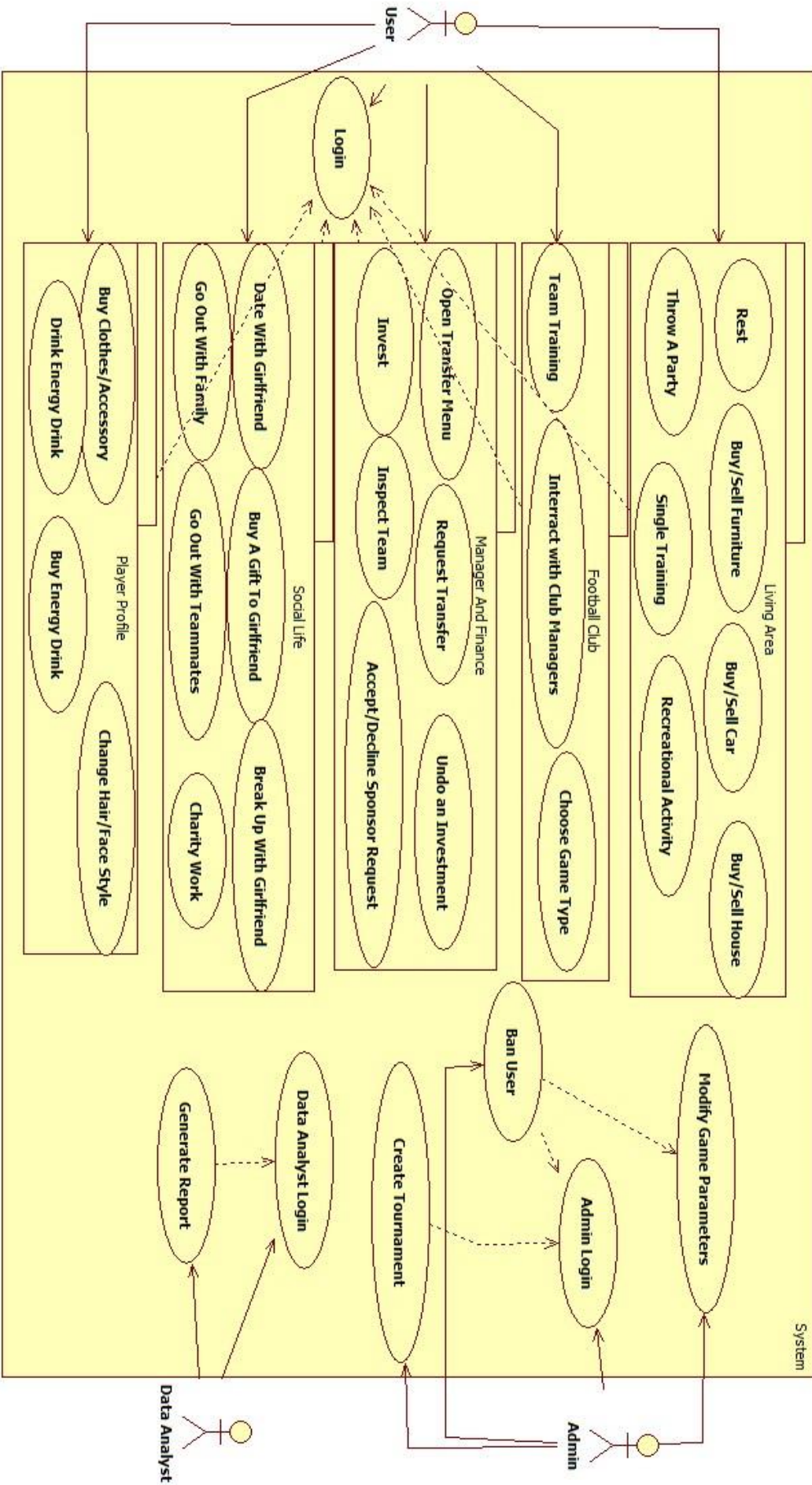
This action gives data analyst ability to generate dynamic reports based on their specifications. These reports in going to be generated using all the automatically stored actions of all the users playing the game. The accuracy and information content of reports based on the number of players using the game.

Design Relationships:

User can send action inputs to clients. In clients these actions will be interpreted and required messages will be sent to both game and data mining servers. Game server will compute the resulting output of user commands received from clients and game and user information stored in related database. The resulting output will be sent back to clients. Data mining server will process the messages coming from clients and store them in the data mining database.

Administrators can send action inputs to admin panel. In admin panel these actions will be interpreted and required messages will be sent to the game server. These messages will modify game and user information database. This will also affect the general flow of the game.

Data analysts can send report parameters to data analyst panel. In data analyst panel reports will be generated with these parameters, and information stored in related database.



5.3 Composition Viewpoint

5.3.1 Design Concerns

Composition viewpoint will help to manage software process. In this part, main work packages and packages inside those will be identified.

There will be there main work packages in this software. Namely: Clients, Data Mining Server and Game Server. Game server shall be implemented first and be ready before starting to Client implementations.

There are smaller components in each main work package and they are described in below component diagram.

5.3.2 Design Elements

Design Entities:

Main design entities are Game Server, Data Mining Server and Clients.

Game Server consist of two subsystems, which are Game Logic and General Server Components. Game Logic subsystem consists of Team, Character, Statistcis, User, Finance, Match and Social Life components. General Server Components are I/O Handler, Database Manager, Administration Manager and Scheduler.

Data Mining Server consist of 4 smaller components. They are I/O Handler, Action Collection Manager, Database Manager and Report Generation component.

Clients are interfaces of Game Server, which shall run on Web Browsers, iOS and Android devices.

Design Relationships:

Smaller components of Game Server are connected each other with Manager singletons. These Managers are Football Life, Finance, Social Life and Admin managers. Managers shall provide communication between other components of game server. The managers will be also responsible of connecting I/O Handler, Database Manager, Scheduler and Administration Manager with Game Server's components.

I/O handler of Data Mining Server is responsible to manage connections of this server. According to coming commands it should interact with Action Collection and Report Generation components. These two components will be connected with Database Manager.

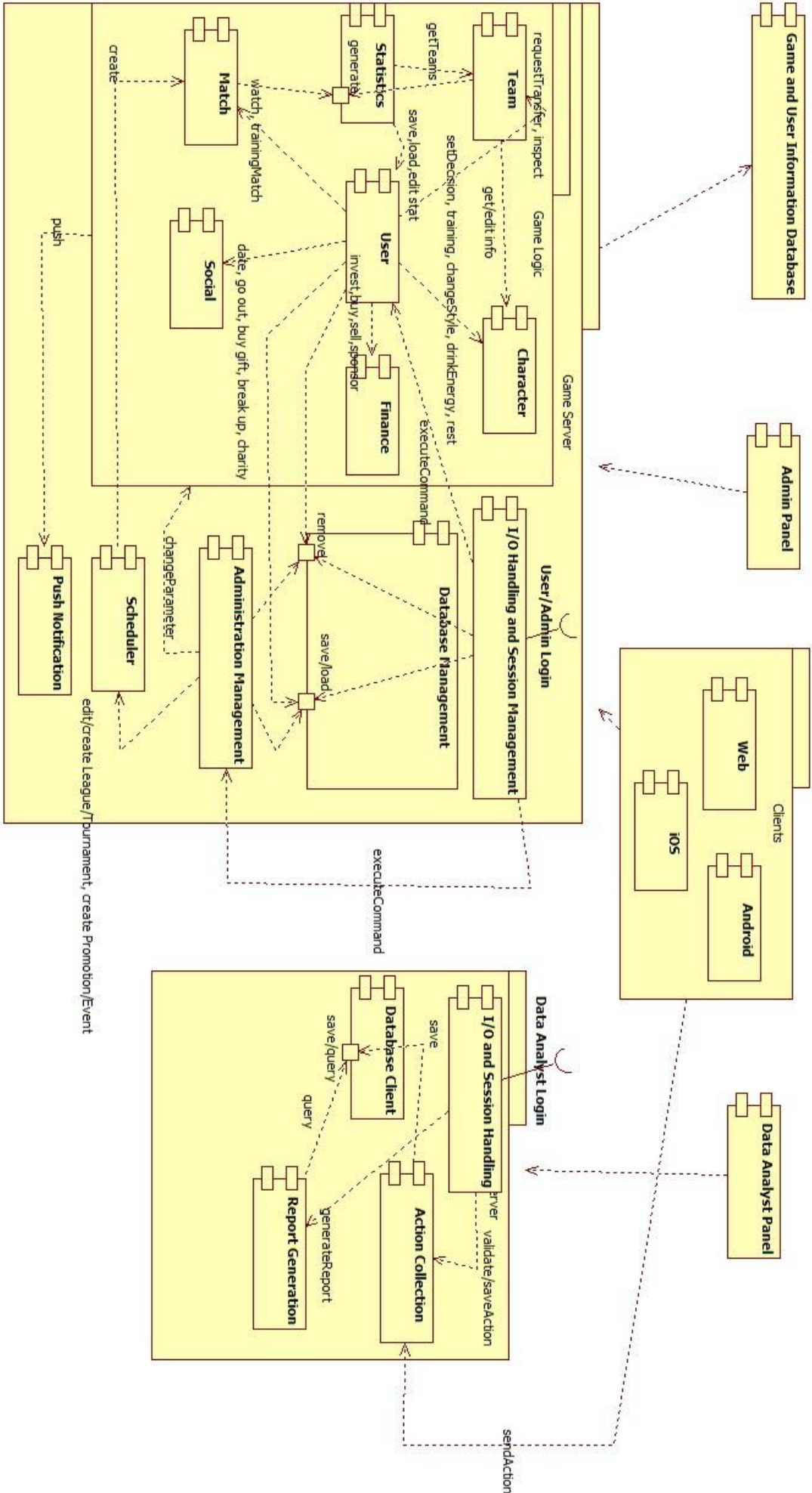
Clients are connected to Game Server through I/O Handler component of Game Server. Incoming commands shall be delivered to related Managers.(Football Life, Finance, Administration, Social)

Design Attributes:

Game Server's aim is to simulate game world and connect users to each other, letting admins to manage game world. To accomplish this, game server has smaller components. Each component is capable of computing related operations. And also each component holds necessary information.

Data Mining Server's aim is to collect and compute statistics received from users. With computing them it should generate detailed reports. To simplify data mining server system, there are helper components. They are able to do different operations over different datas.

Clients shall connect users with game world. User should be able to use every operation defined in server.



5.4 Logical Viewpoint

5.4.1 Design Concerns

Logical viewpoint will identify all classes and relations between classes. Aim of logical viewpoint is to clarify and simplify the system design and lead development team.

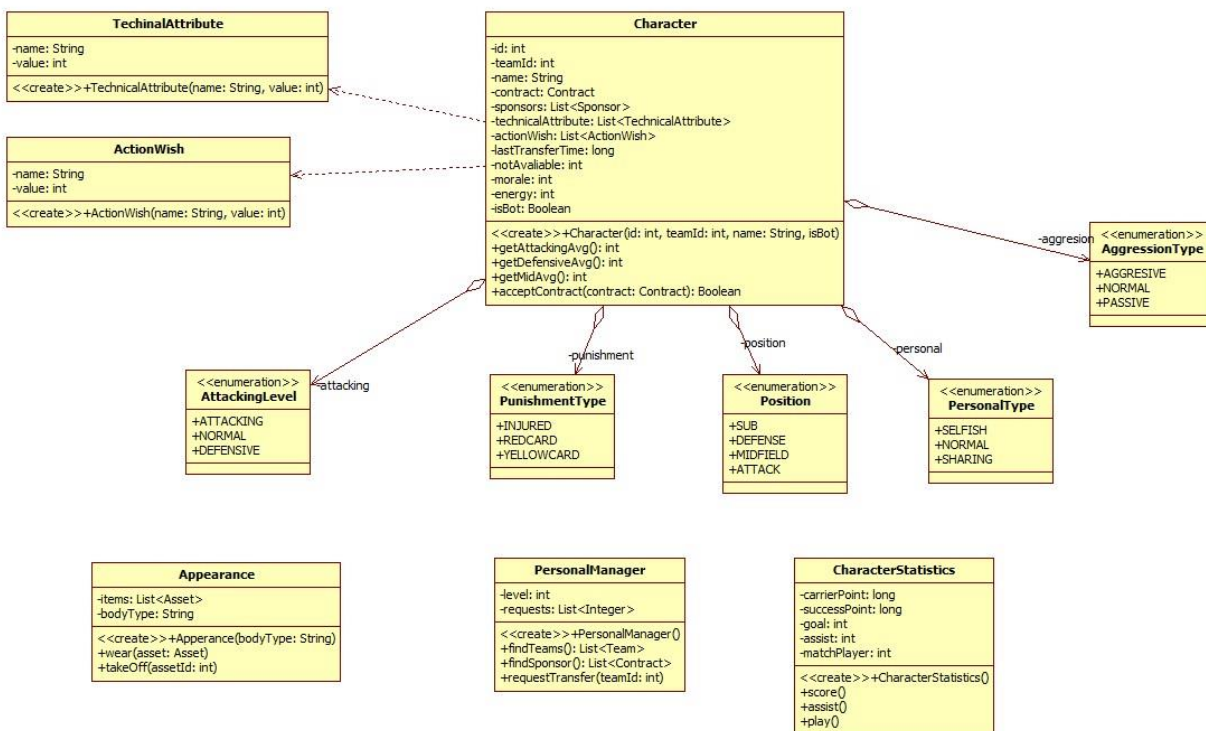
There are too many components in design of the software. Since it is impossible to show the design in one diagram, all components will be explained differently. Main correspondences and associations will be described for each component or class.

5.4.2 Design Elements

Relationships of all classes provided below will be explained in related classes' attributes.

5.4.2.1 Character Component

This component's responsibility is to hold and manipulate all information related to users' football players in game. There are 6 classes and 5 enumerations to handle this duty.



Entities:

5.4.2.1.1 Character

This class will hold and organize all information of a user's football player. This class will be used by Finance, Social Life, Football Life and Admin managers. It will use/create/hold all other classes inside this component.

Through Character class, all 6 classes can be connected with Finance, Football Life and Social Life managers. These three managers should be able to manipulate data stored in described classes. Any method in the classes can call Managers' methods.

5.4.2.1.2 TechnicalAttribute:

The class has two attributes, which will hold related character's skill. For example, it can store "Shooting" and 75. Complete information will be given in game design document.

5.4.2.1.3 ActionWish:

ActionWish class will show user's wish to be present in related actions. Some of these actions might be "Shooting", "Heading", "Freekick" and so. This information will be given in game design documentation.

5.4.2.1.4 Appearance:

This class will store assets (accessories, hairstyles, clothes), which are related to character's appearance. Using this class, user will be able to wear and take off an asset.

5.4.2.1.5 PersonalManager:

Every user will have a personal manager, who organizes character's football and financial life. According to level of the personal manager he can find more suitable teams, sponsorships and transfer opportunities.

5.4.2.1.6 CharacterStatistics:

This class will hold two success points gained from every action in game. Career points are related to character's football life and success points are related to whole life of the character.

5.4.2.1.7 AttackingLevel (Enum):

Attacking level can be set by user, through football manager. According to attacking level value, the character will act differently in match simulations.

5.4.2.1.8 PunishmentType (Enum):

The enumeration will show punishment status of a character. He can be injured, punished by yellow card or punished by red card. Team class will use this information while setting a team strategy.

5.4.2.1.9 Position (Enum):

The position enumeration on which position the player is assigned by his team. Team class will use this information

5.4.2.1.10 PersonalType (Enum):

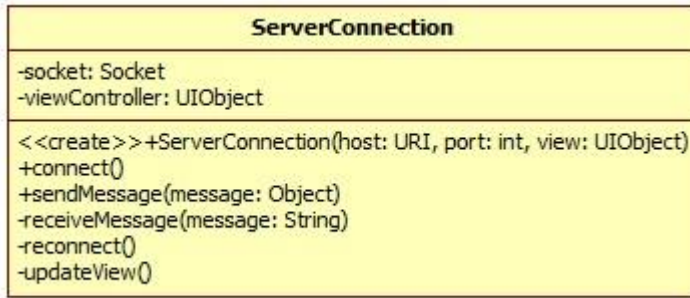
PersonalType will clarify if the character is willing to play selfish, sharing or normal. This information will be used by match simulation and action classes.

5.4.2.1.11 AggressionType (Enum):

This enumeration will hold if a character decides to play aggressive, passive or normal. This information will be used by match simulation and action classes.

5.4.2.2 Client Component

This component's aim is to handle server connections and update view in client side. There is only one class to handle this. There will be additional helper classes if necessary.



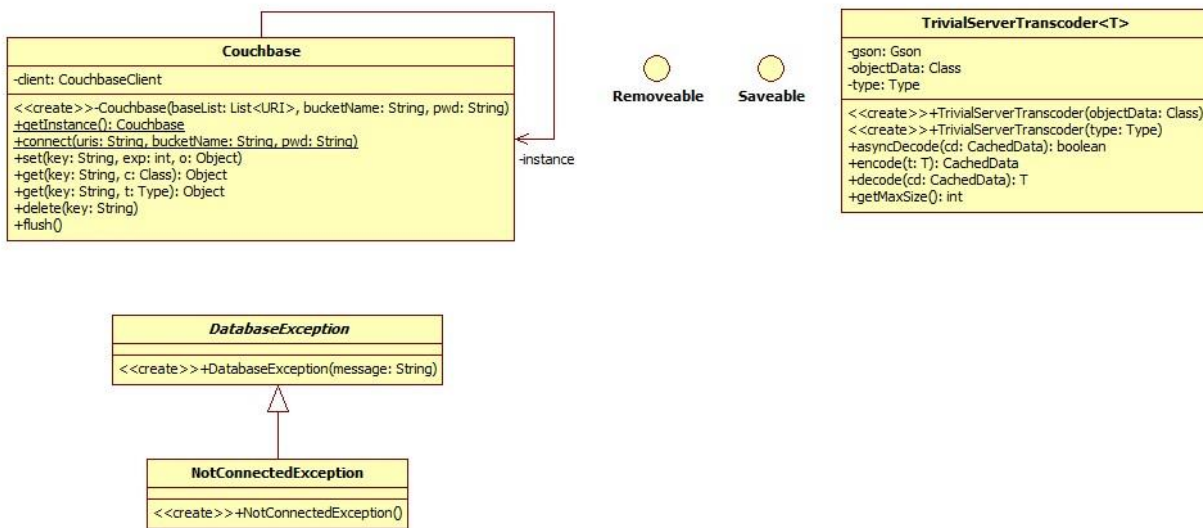
Entities:

5.4.2.2.1 *ServerConnection*

This class will create a socket connection between client side and server. Then with sending necessary messages, it will receive a page view and show that to user. It will also send user’s interaction with this page view.

5.4.2.3 *Database Component*

This component’s duty is to connect and transfer data between database and server. There are 2 main classes and 2 exception classes to handle this operations.



Entities:

5.4.2.3.1 *CouchBase*

This class will provide information and configuration methods for couchbase database. It is a singleton. This singleton will be used by all components, who need to communicate with database.

5.4.2.3.2 *DatabaseException*

When an error in database occurs while reading, writing or removing this class will handle exception. It will be used by methods, which tries to connect with databases.

5.4.2.3.3 NotConnectedException

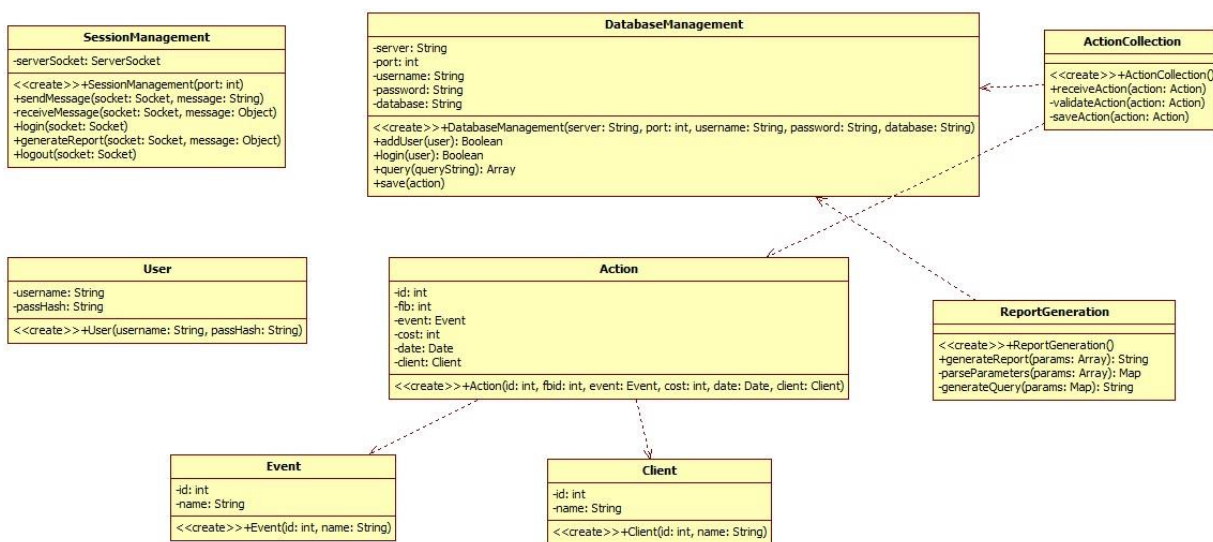
This extended class is a kind of exception related to database exceptions.

5.4.2.3.4 ServerTranscoder<T>

ServerTranscoder class would encode and decode message between server and database. It will be used by all methods which connects with database.

5.4.2.4 Data Mining Server

This server’s aim is to collect data from users, handle database connections and generate reports for data analysts with given parameters. There are 8 classes to achieve these operations.



Entities:

5.4.2.4.1 SessionManagement

This class manages connection between data analyst panel and data mining server. It handles all incoming and out-going messages through a socket connection. It will be used by all other classes in this component.

5.4.2.4.2 User

User will hold very basic information of a data analyst. It will be used in login and authentication parts of database manager for data mining server.

5.4.2.4.3 Action

Action class holds user’s actions to game. It uses event class and client class as members.

5.4.2.4.4 Event

An event is a class which is a member of action class. It will be a basic information about user’s interactions with game. They will be achieved from game clients.

5.4.2.4.5 Client

This class holds information about client devices for related action. It will be achieved from clients through sessioning management class.

5.4.2.4.6 DatabaseManagement

This class manages connections between MySQL Data Mining database and data mining server. It is used by ActionCollection and ReportGeneration classes.

5.4.2.4.7 ActionCollection

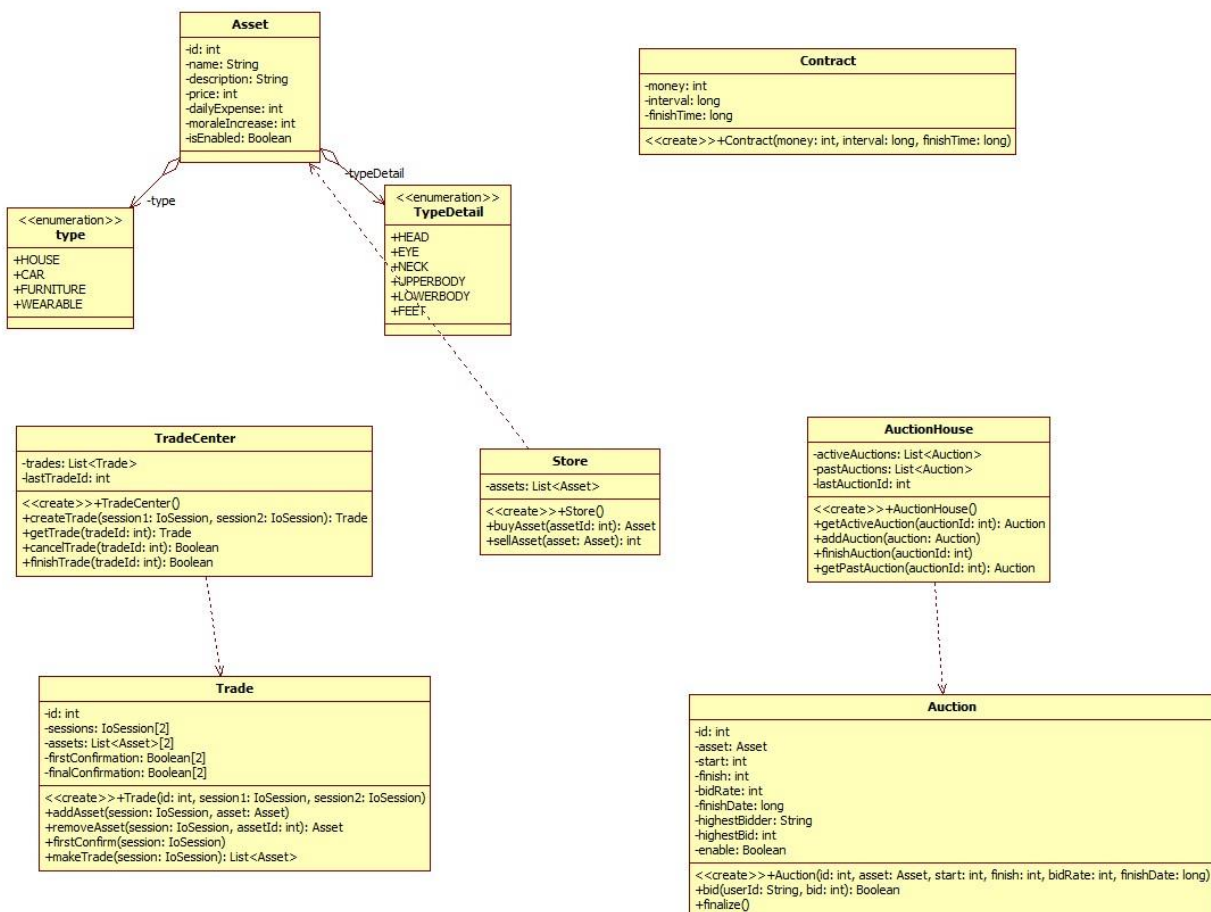
Action collection is a manager, which stores, loads and removes actions from/to data mining database. It will use DatabaseManagement class and action class.

5.4.2.4.8 ReportGeneration

The aim of this class is to generate reports with information recieved from database and parameters recieved from data analysts.

5.4.2.5 Finance Component

Finance component shall store information of all kinds of contracts (sponsorships, player-team contrats), trade center and assests available for trade and lastly auction information and auction items. Users will reach this information and interact with assrest available through finance manager.



Entities:

5.4.2.5.1 Asset

This class is representation of any kind of item, which a user can buy, use, sell and equip. It has a daily expense member which shows how much money would a user lose from that asset. And there is a member which specifies how much moral increase would be happen for related character.

5.4.2.5.2 Type (Enum)

Type enumeration will clarify type of an asset. It can be house, car, furniture or wearable. This information will be used by asset class directly.

5.4.2.5.3 TypeDetail (Enum)

The enumeration class will hold additional information of related type asset. It will also be used by asset class directly.

5.4.2.5.4 Contract

Contract is a class which is used by sponsorship agreements and team-player contracts. It shall show how much money the character earns from this contract and when contract ends.

5.4.2.5.5 TradeCenter

TradeCenter class is a manager class which organizes, creates and computes trades between two sessions. This sessions can be computers or humans.

5.4.2.5.6 Store

This class has information of all available assets. Finance manager can use this class to buy an asset to a related character. It can also use Store class for selling assets.

5.4.2.5.7 Trade

Trade class is responsible to manage a single trade operation between two sessions. It will wait for getting two sessions' decides and computes the result. According to result it will set assets information through Finance Manager.

5.4.2.5.8 AuctionHouse

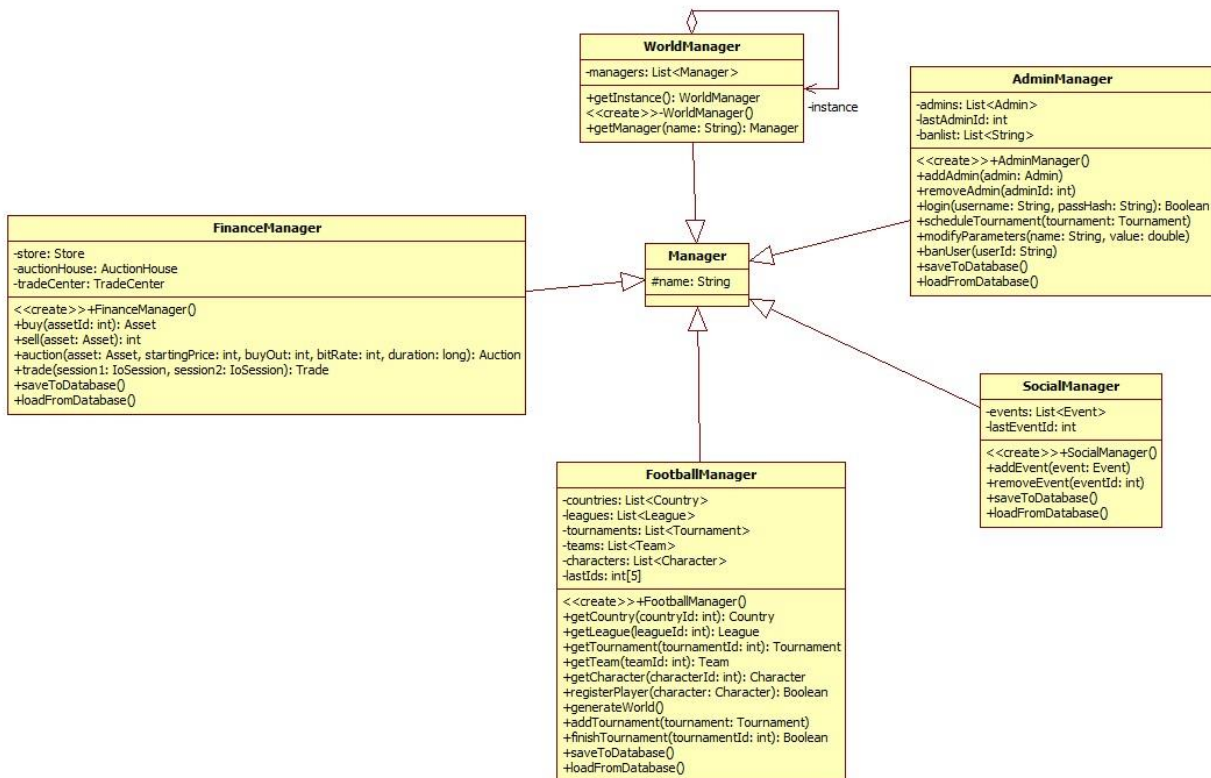
AuctionHouse class is responsible to manage active auctions. Auctions are created and finalized here. This class is used by Finance Manager.

5.4.2.5.9 Auction

Auction class holds the information of auction.

5.4.2.6 Managers

Manager classes are singletons available to other game logic components. Their aim is to handle communication between these components. Every command shall receive to I/O handler, then sent to one of the managers and lastly managers will interact with other components and compute the command. There are 5 main manager classes extended from one base Manager class.



Entities:

5.4.2.6.1 [Manager](#)

Manager class is extended by all other classes in this component. It is responsible to connect other components to each other. It holds the name information.

5.4.2.6.2 [WorldManager](#)

This class' aim is to let other components to reach desired manager. This is a singleton.

5.4.2.6.3 [FinanceManager](#)

It holds information related to all kind of finance. (Auctions, trades, store) It organizes all finance operations.

5.4.2.6.4 [FootballManager](#)

This class holds all the information related to football.(leagues, tournaments, teams, simulation). It organizes everything related to this information.

5.4.2.6.5 [AdminManager](#)

This class provides all game information to system administrators.Administrator Registry and log in processes are managed by this class.

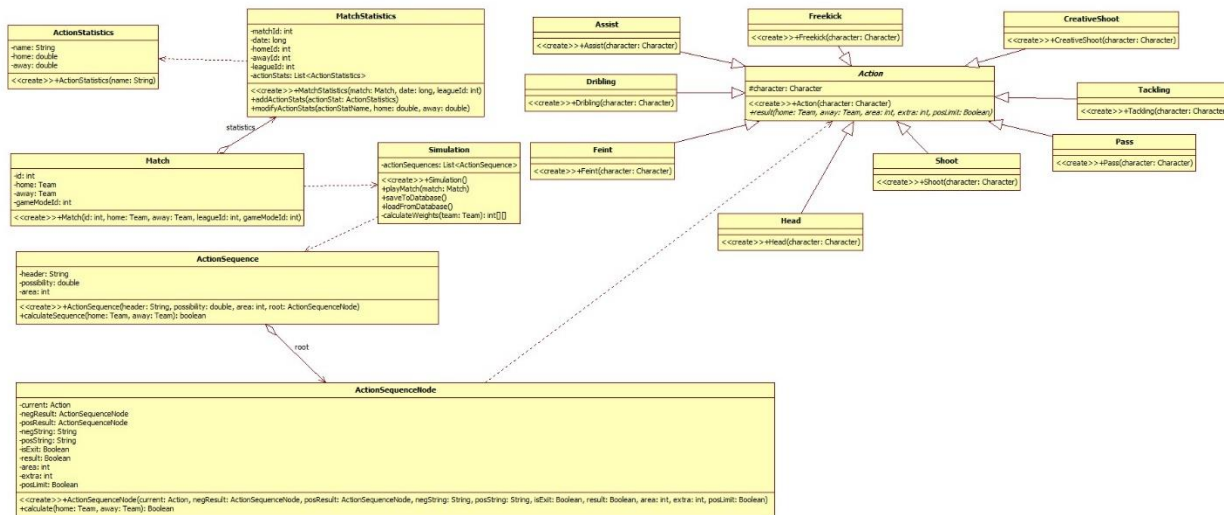
5.4.2.6.6 [SocialManager](#)

This class manages all social events in the game(girlFriendAction, goOutAction, charity works). All social events are created here.

Relationships:

5.4.2.7 Match Component

Match component will be created to simulate football matches between given teams and generate statistics for both team and character classes. There are 7 base classes in this component. One of the classes will be extended by another 9 classes. This will be explained in Action Class section.



Entities:

5.4.2.7.1 Match

This class gets 2 teams as a parameter and then it will be used by match simulation class.

5.4.2.7.2 Simulation

This class lets matches to be simulated. First this class calculates teams' power attributes for related areas, then according to this calculations it gets a randomly chosen action sequence and simulates the match with calculating these action sequences.

5.4.2.7.3 MatchStatistics

This class is responsible for calculating related match statistics. It uses action statistics to create match statistics.

5.4.2.7.4 ActionStatistics

This class is responsible for calculating related action statistics. It is also used to create match statistics.

5.4.2.7.5 ActionSequence

This class includes reference to first node of action sequence node decision tree. It also holds basic information about this action sequence. They are area and possibility.

5.4.2.7.6 ActionSequenceNode

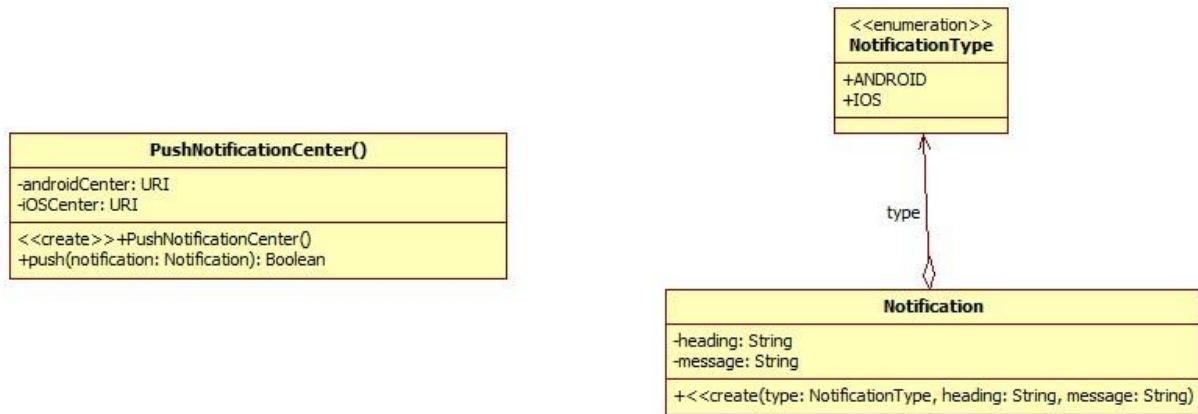
With references of this class, a decision tree can be created. This tree refers to a position in a football match. To calculate if a position is resulted as a goal or not, each action sequence node in the decision tree will be calculated. This class is also responsible for creating commentary strings.

5.4.2.7.7 Action

This class is used to define all 9 actions used for creating action sequences.(Pass, Feint, Freekick, Assits, Dribling, Head, Tackling).

5.4.2.8 Push Notification Component

Push notification component will be responsible of sending notifications to desired clients. These notifications will include promotions and news of game world.



Entities:

5.4.2.8.1 Notification

This class notifies user about important game actions.

5.4.2.8.2 NotificationType

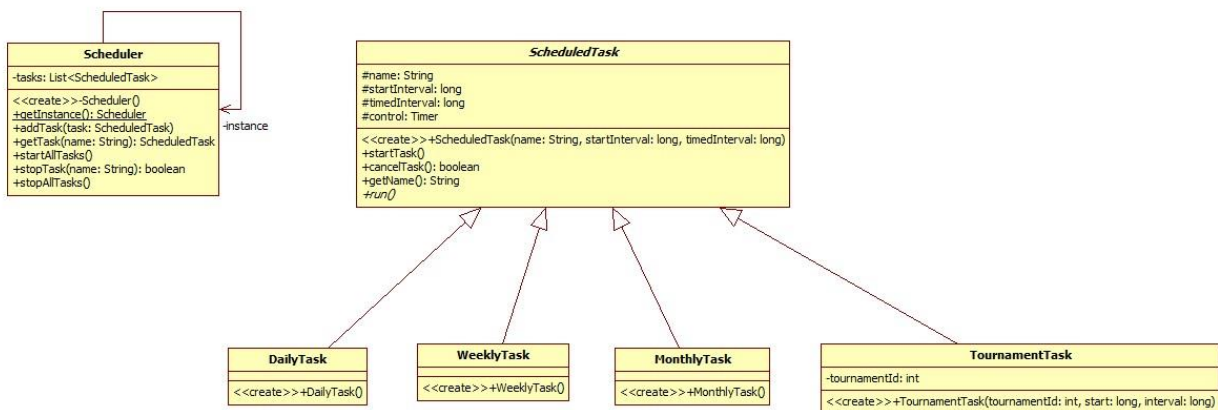
This class identifies the notification types.

5.4.2.8.3 PushNotificationCenter

This class manages all notifications for all clients.

5.4.2.9 Scheduler

This class can be thought as a generic and more efficient timer class. It will schedule daily, weekly, monthly and tournament server tasks. It can schedule a tournament match or a game independent/server related task. This component contains 6 classes.



Entities:

5.4.2.9.1 Scheduler

This class schedules all timer events.

5.4.2.9.2 ScheduledTask

This class is a base class which holds information of task.

5.4.2.9.3 DailyTask

This class is an extended class which holds information of daily task.

5.4.2.9.4 WeeklyTask

This class is an extended class which holds information of weekly task.

5.4.2.9.5 MonthlyTask

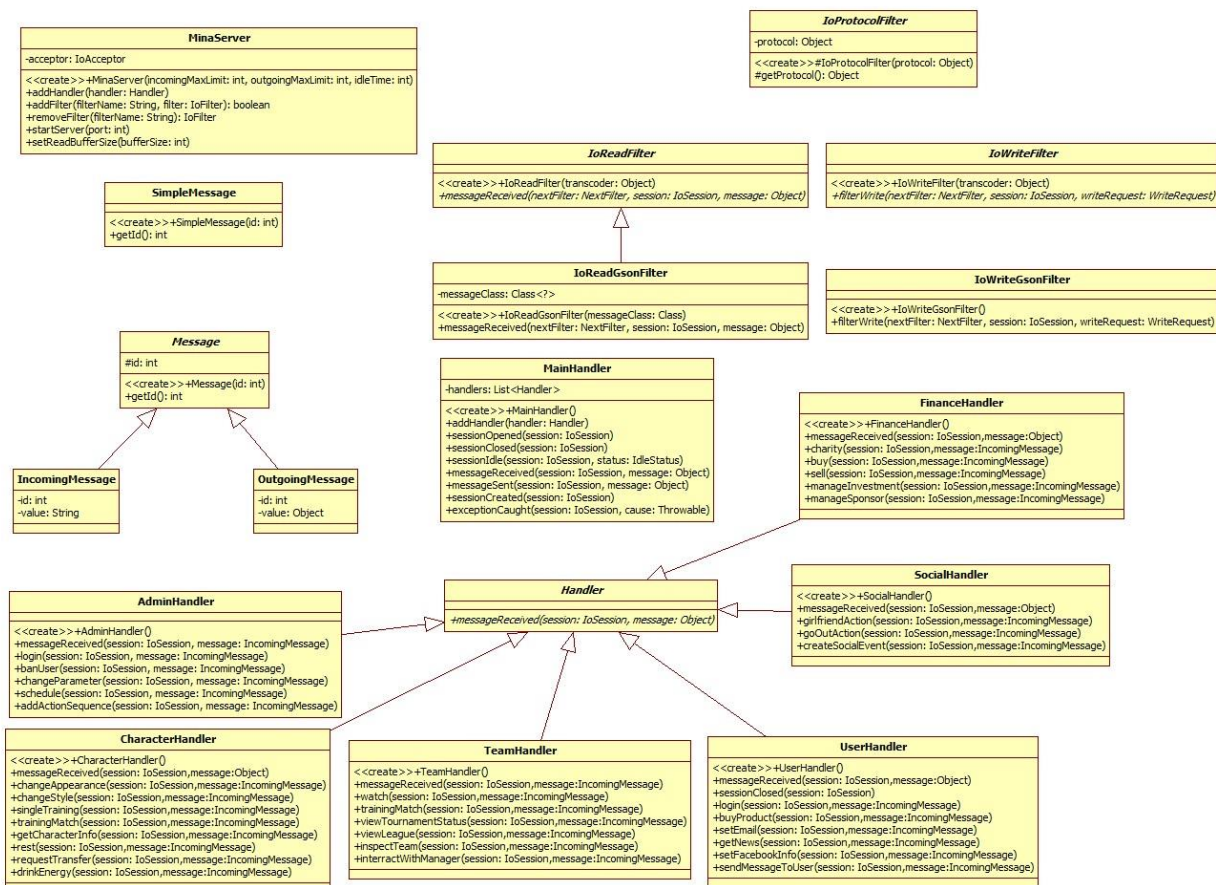
This class is an extended class which holds information of monthly task.

5.4.2.9.6 TournamentTask

This class is an extended class which holds information of tournament task.

5.4.2.10 Session Manager

This manager component's responsibility is to manage connections to server. All kind of users of game server (admin, user) will be first operated here. This component will receive commands from users and notify related manager. There are 18 classes in this component.



Entities:

5.4.2.10.1 MinaServer

This class handle client sessions.

5.4.2.10.2 IOProtocolFilter

This class is used to filter incoming and outgoing messages to/from mina server.

5.4.2.10.3 SimpleMessage

It is a base protocol message.

5.4.2.10.4 Message

It is a more detailed protocol message.

5.4.2.10.5 IncomingMessage

It is an extended message which holds incoming message information.

5.4.2.10.6 OutgoingMessage

It is an extended message which holds outgoing message information.

5.4.2.10.7 IOReadGsonFilter

It filters json typed IO messages.

5.4.2.10.8 IOWriteFilter

It filters outgoing messages.

5.4.2.10.9 IOReadFilter

It filters incoming messages.

5.4.2.10.10 IOWriteGsonFilter

It filters outgoing json typed messages.

5.4.2.10.11 MainHandler

It handles all sessions' operations.

5.4.2.10.12 Handler

This is a abstract class for games' handler classes. Following classes are extended from this.

FinanceHandler

SocialHandler

UserHandler

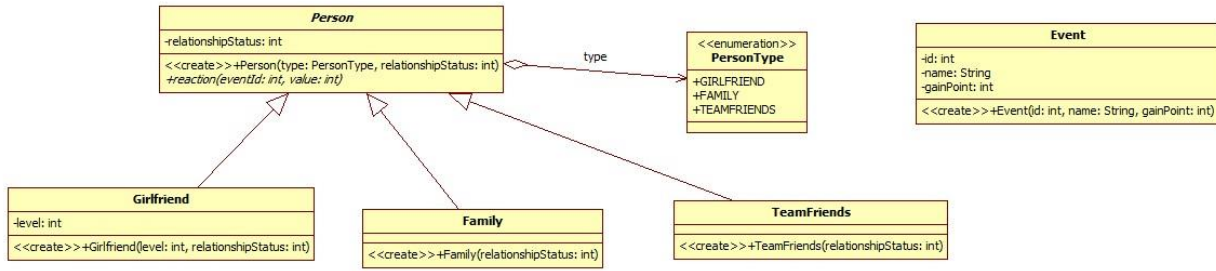
TeamHandler

CharacterHandler

AdminHandler

5.4.2.11 Social Life Component

This component aims to store social life information of a user. When user operates on his/r social life, social life manager will manipulate classes in this components. Afterwards, user can reach his social life information. This component also computes responses of people in user's social life.



Entities:

5.4.2.11.1 Event

It describes a social event and hold information about it.

5.4.2.11.2 Person

It is an abstract class of actors in social life.

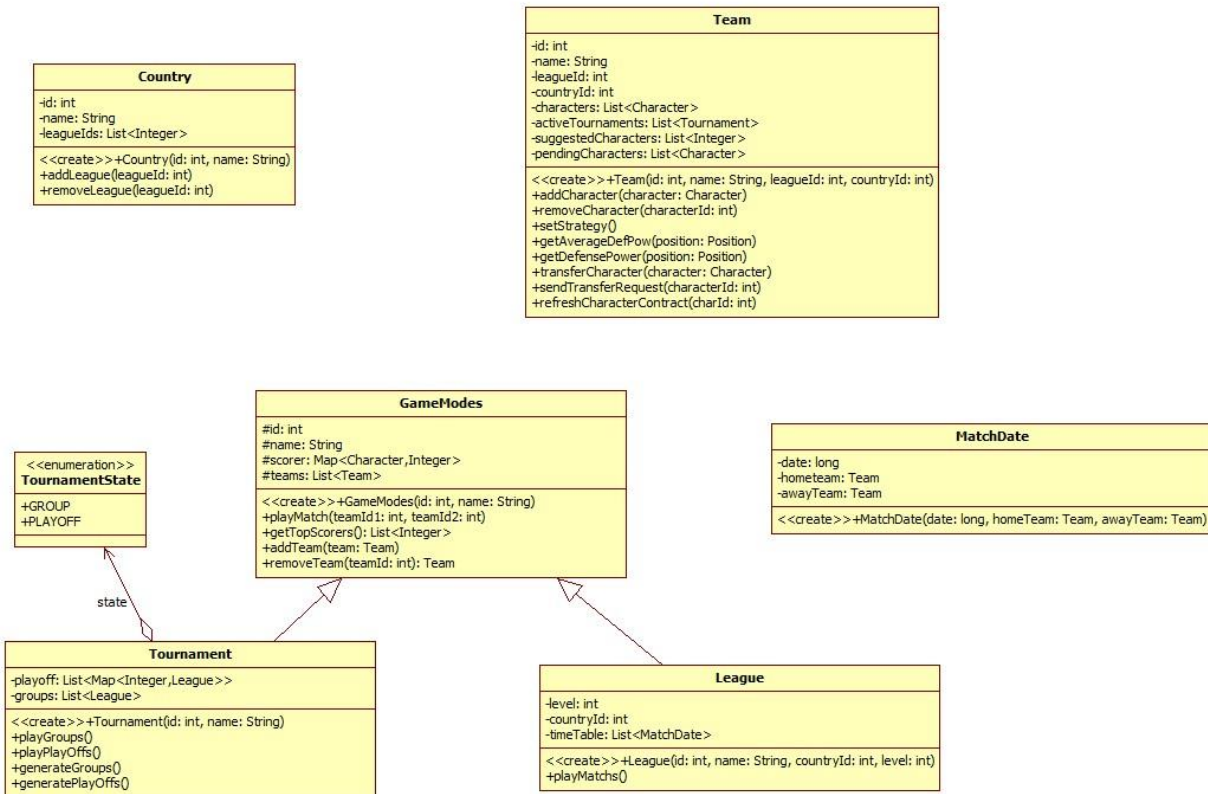
GirlFriend

Family

TeamFriends

5.4.2.12 Team Component

Team Component will store all teams information around the game world. It will hold country, league, tournament and math fixture informations. Football life manager will use this component to simulate matches. This component contains 7 classes.



Entities:

5.4.2.12.1 Team

This class holds information about the team. It can set team strategy.

5.4.2.12.2 GameModes

It is a base class which holds information about tournaments and leagues.

5.4.2.12.3 League

League holds teams and using football manager it simulates a league.

5.4.2.12.4 Tournament

Tournament holds teams and using football manager it simulates tournaments. It is formed from leagues.

5.4.2.12.5 TournamentState

It holds a state of tournament which can be groups or playoffs.

5.4.2.12.6 Country

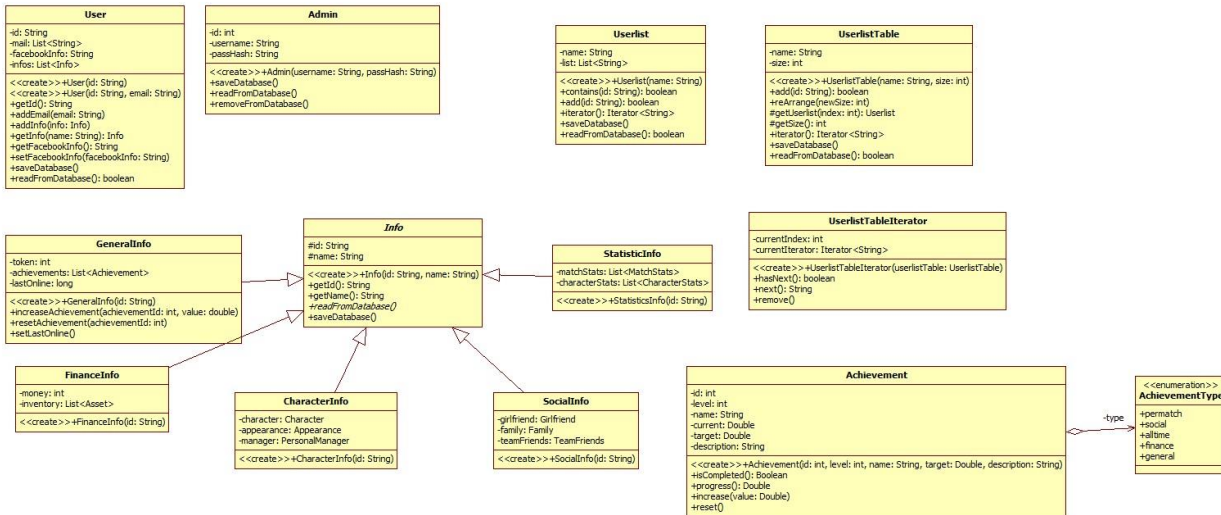
It holds leagues.

5.4.2.12.7 MatchDate

It is a scheduled task to adjust the times of matches.

5.4.2.13 User Component

User component will store all basic information of users and admins. From this basic information a user or admin can reach more detailed information described in other components. There are 12 classes and 1 enum class in this component.



Entities:

5.4.2.13.1 User

It is a representation of a real time persons' account in game world.

5.4.2.13.2 Admin

It is a user like class for administrators.

5.4.2.13.3 Userlist

This holds the userlist information.

5.4.2.13.4 UserlistTableIterator

It is used to iterate over user list.

5.4.2.13.5 Achievement

It is used to define challenges and reward users in a game.

5.4.2.13.6 AchievementType

It is used for classification of achievements.

5.4.2.13.7 Info

It is a general information object which is available for users and connects to other information objects. These information objects are listed below;

GeneralInfo

FinanceInfo

CharacterInfo

SocialInfo

StatisticInfo

5.5 Interface Viewpoint

5.5.1 Design Concerns

- The application server is developed in java so JVM is restricted.
- This application's server and client communicate over internet.
- Also the IP port on the server must be available to clients and also TCP drivers must be installed on server.
- For server stability, a cloud server is needed to run server. Each client should connect to internet to use application and
- All clients and server must be communicate with Facebook graph api. Besides these, user interfaces need user actions and
- These actions are provided from a healthy running touch screen for mobile devices and mouse for web clients. Another concern is that
- Web client users should install a proper flash player to their own computers to run web client.

5.5.2 Design Element

External Interfaces:

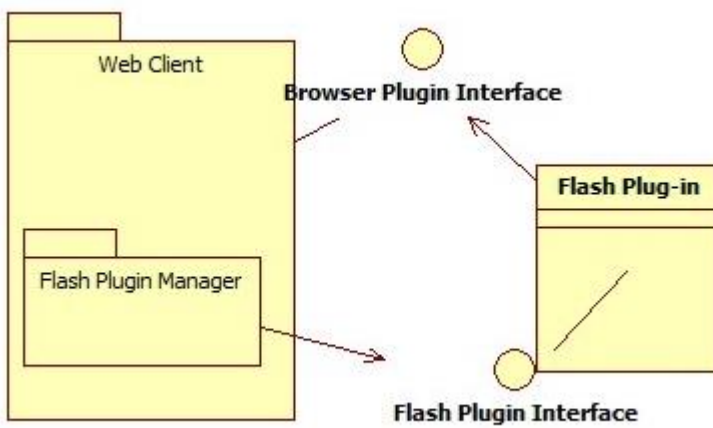
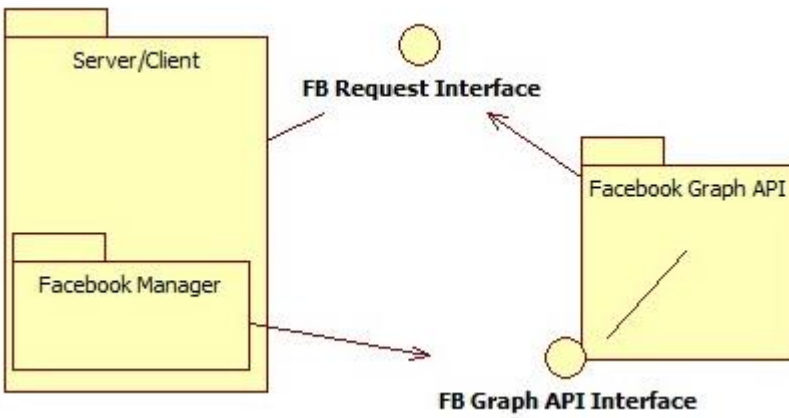
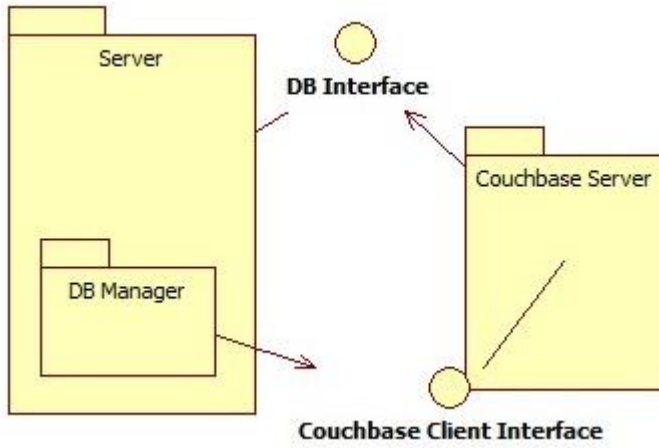
- TCP/IP Port Interface: for basic communication between server and clients
- Mobile Device Touch Interface: for getting touch event and sends it to mobile client app.
- Mouse Device Interface: for getting mouse events by OS I/O interfaces and sends it to web client app.
- Couchbase Client Interface: for connecting server to no-sql database
- Facebook Grap API Interface: for sending facebook requests and getting responses on server side.
- Flash Player Plugin Interface: for running web client on a web browser which support flash player

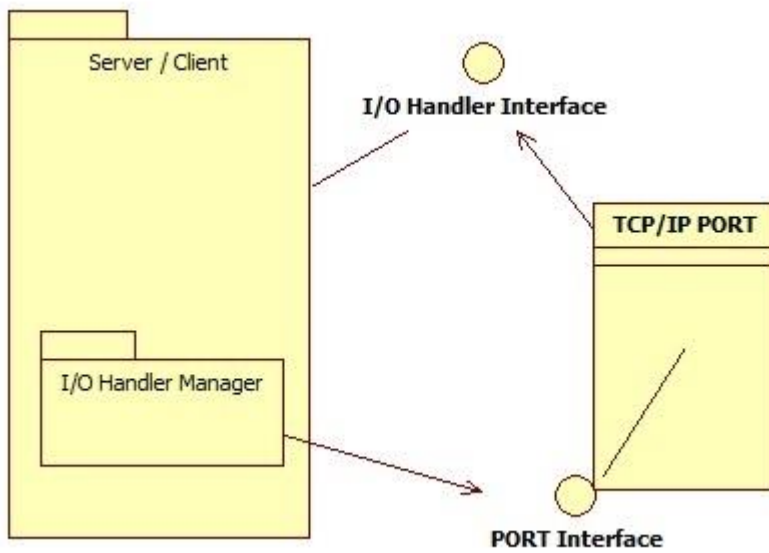
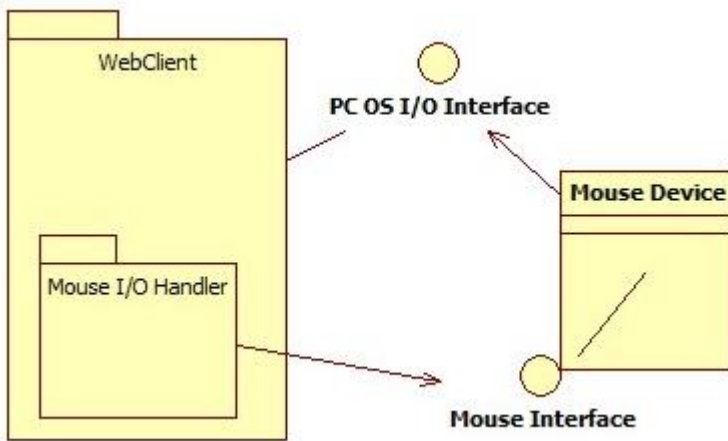
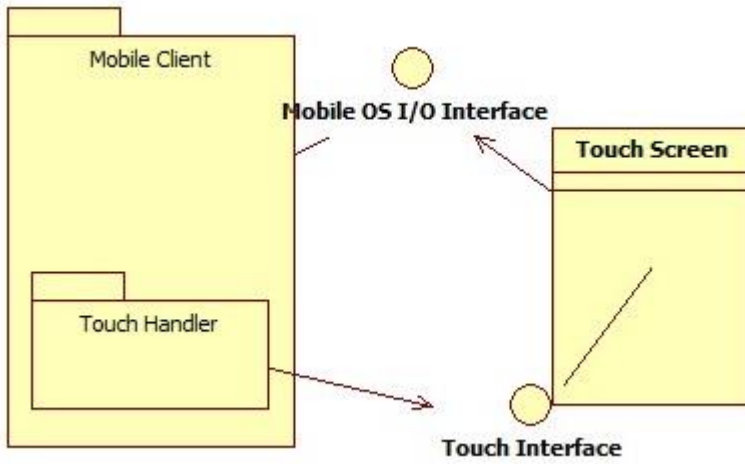
Internal Interfaces:

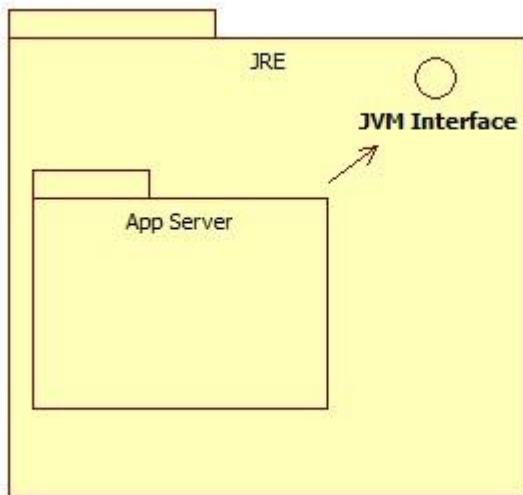
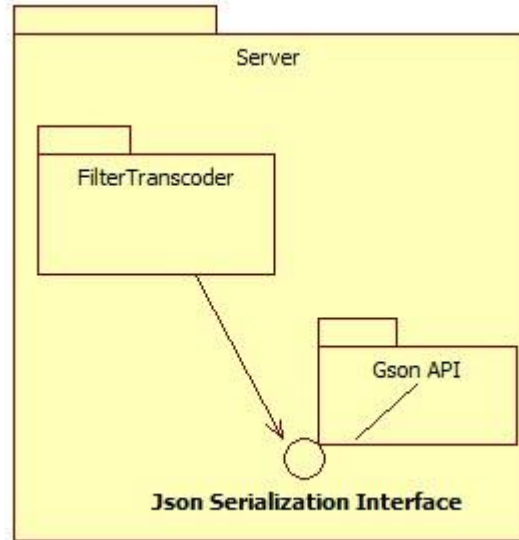
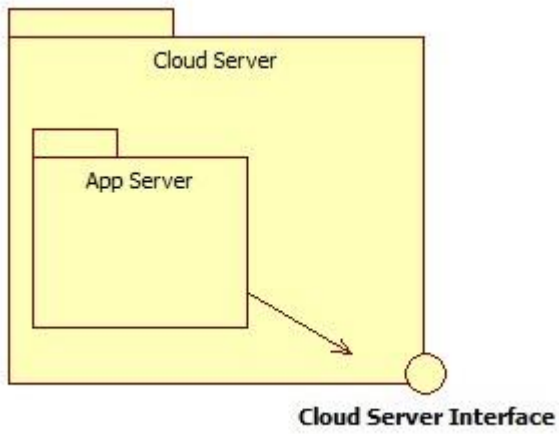
- Cloud Server Application Interface: for running app server constantly and making it reachable on the internet
- JVM Run-time platform interface: for running app server which is developed in java
- Mina Session Interface: for handling clients which connects to server as sessions.
- Gson Serialization interface: for serializing objects to json form.

5.5.3 Example Languages

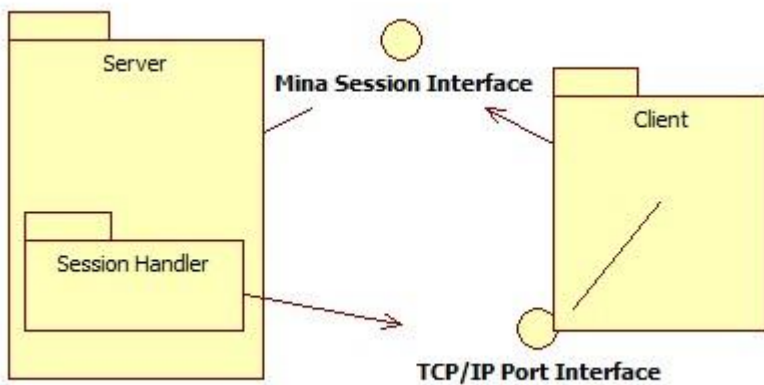
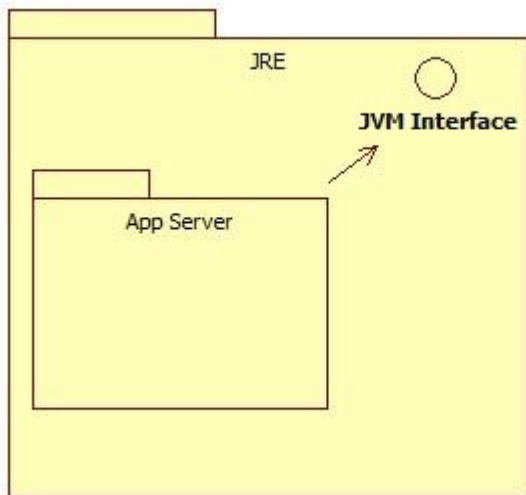
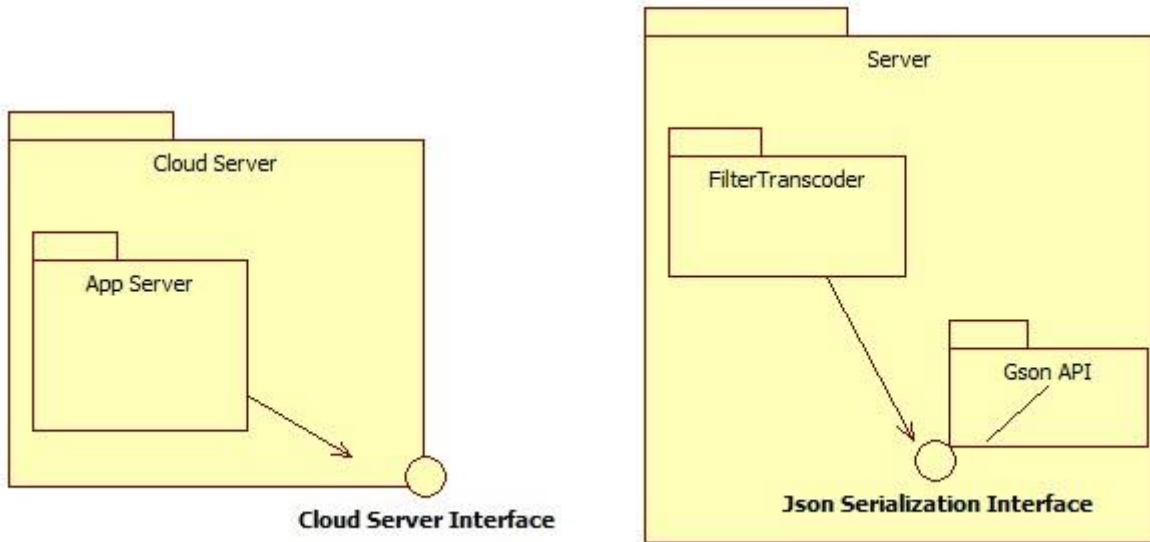
External Interfaces







Internal Interfaces



5.6 Interaction Viewpoint

5.6.1 Design Concerns

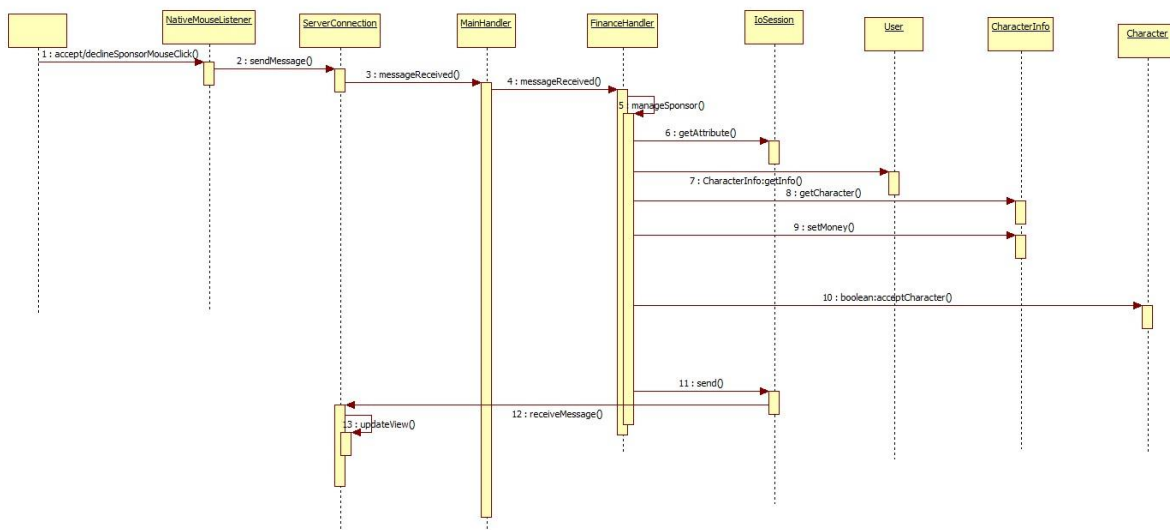
In the products design we have used two important patterns. Singleton pattern is used for the most used and most universal classes amount the server and for utility classes we also used the same singleton pattern.

In client we have used model-view-controller architecture. Although we haven't mentioned any specific classes, in the model section we have used identical models of server classes and for view part we have defined a server connection class. View part of the MVC architecture are native classes of the specific clients.

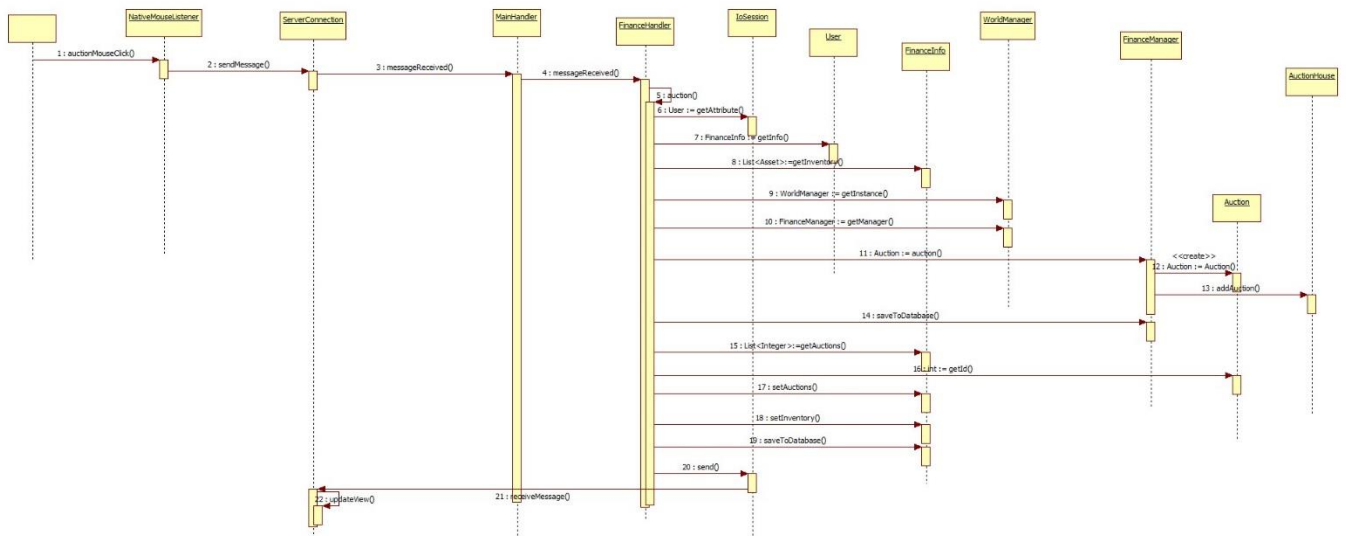
The main concern for used patterns are that they need to be implemented correctly. Also, the implementation should decouple classes as much as possible.

Another concern is to correctly transmit message between objects of the server. Since this is a multithreaded application, All the mutual function need to take this under consideration and use necessary mechanisms.

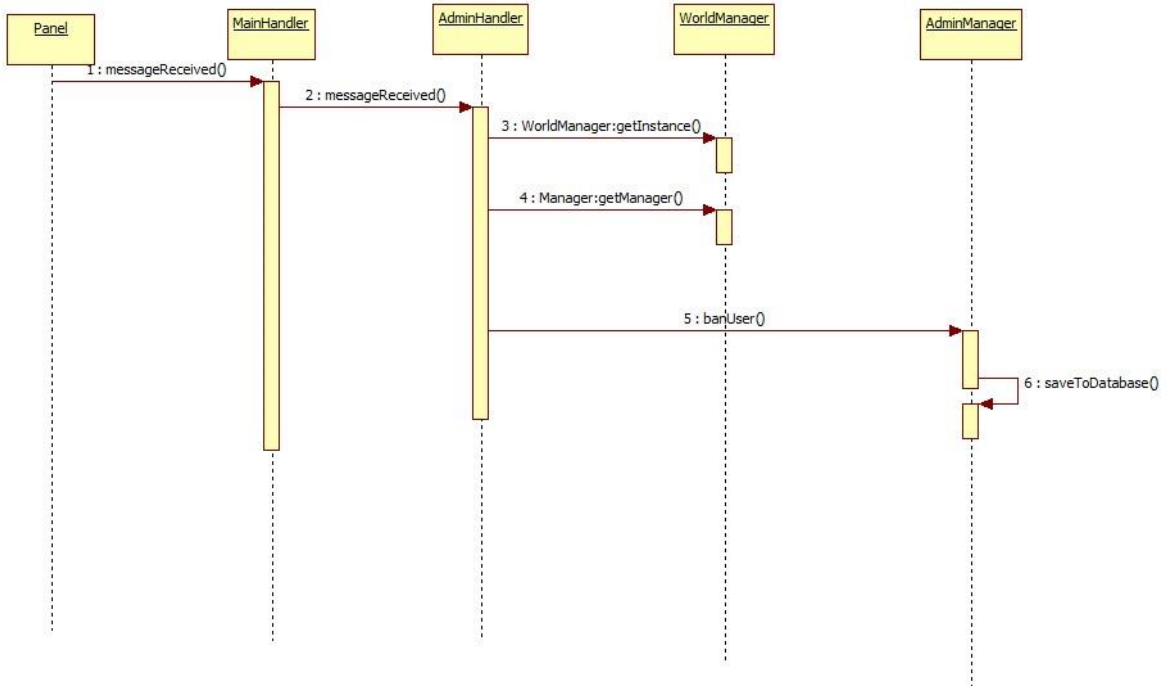
5.6.2 Design Elements



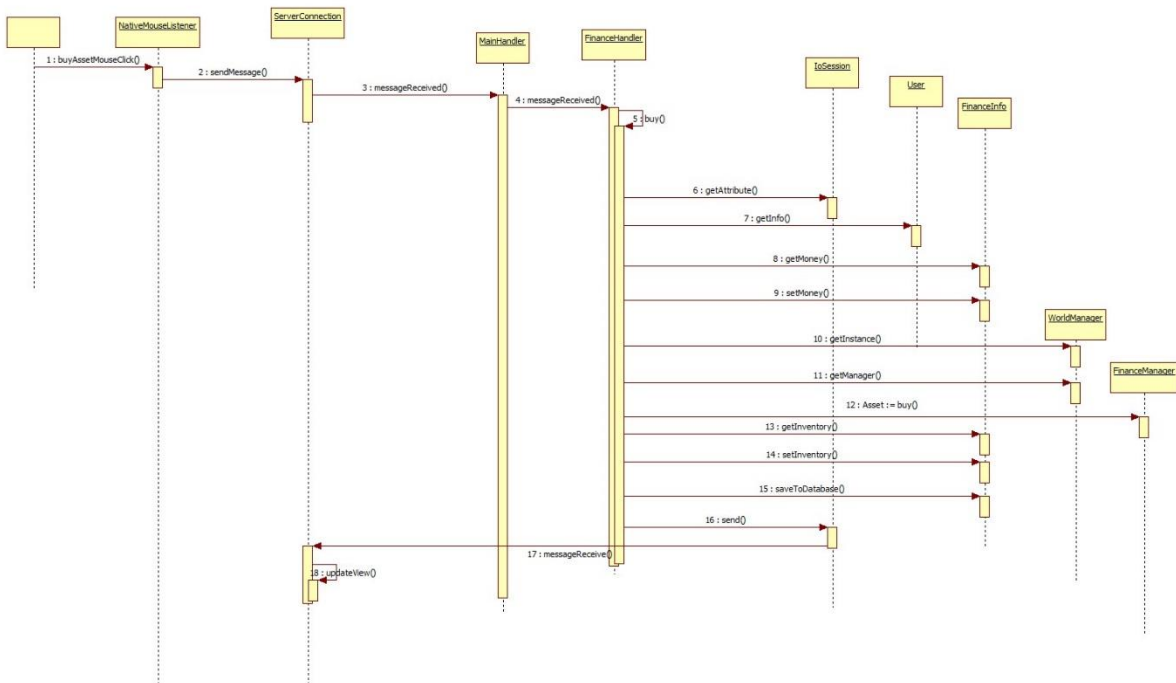
This sequence diagram shows the use case of Accept/Reject Sponsor.



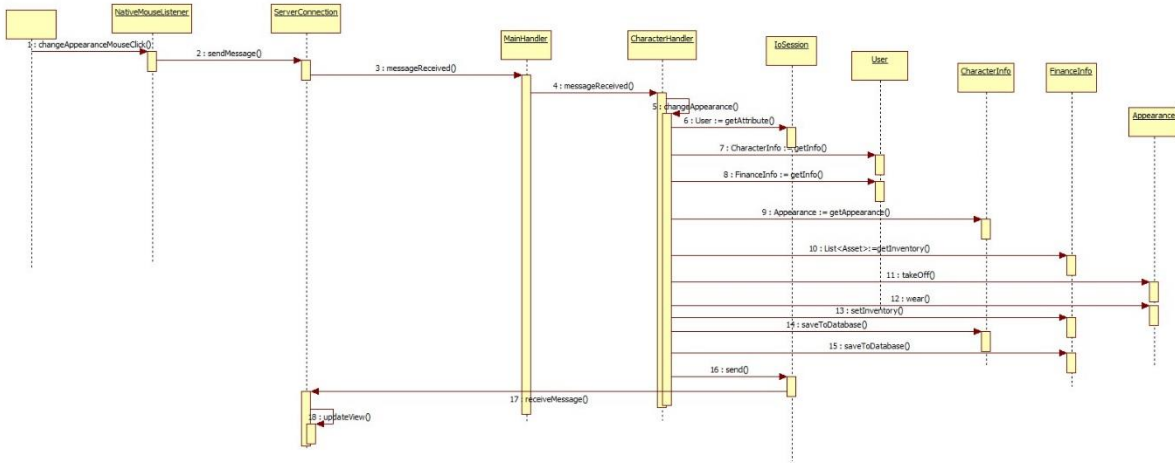
This sequence diagram shows the use case of Auction.



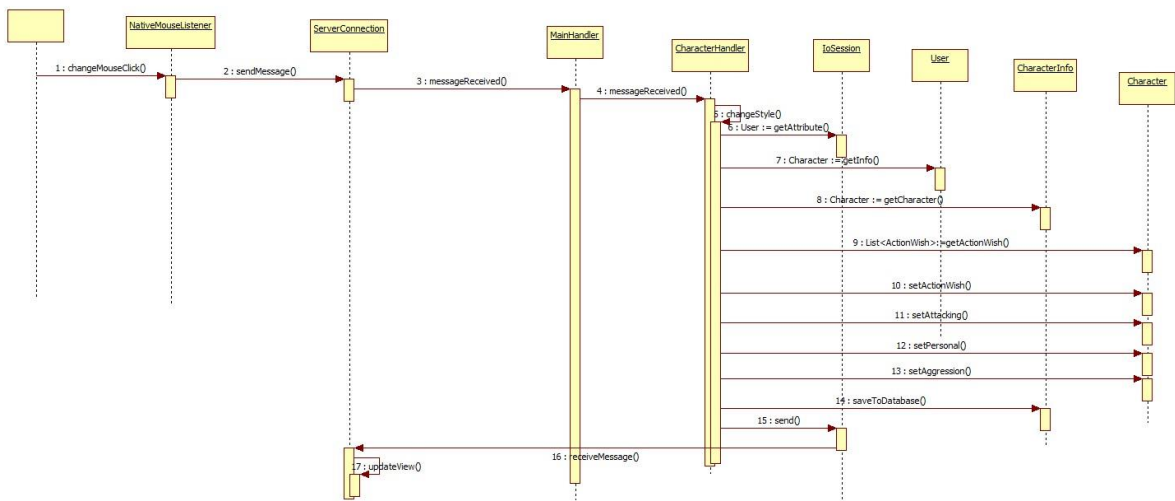
This sequence diagram shows the use case of Ban User.



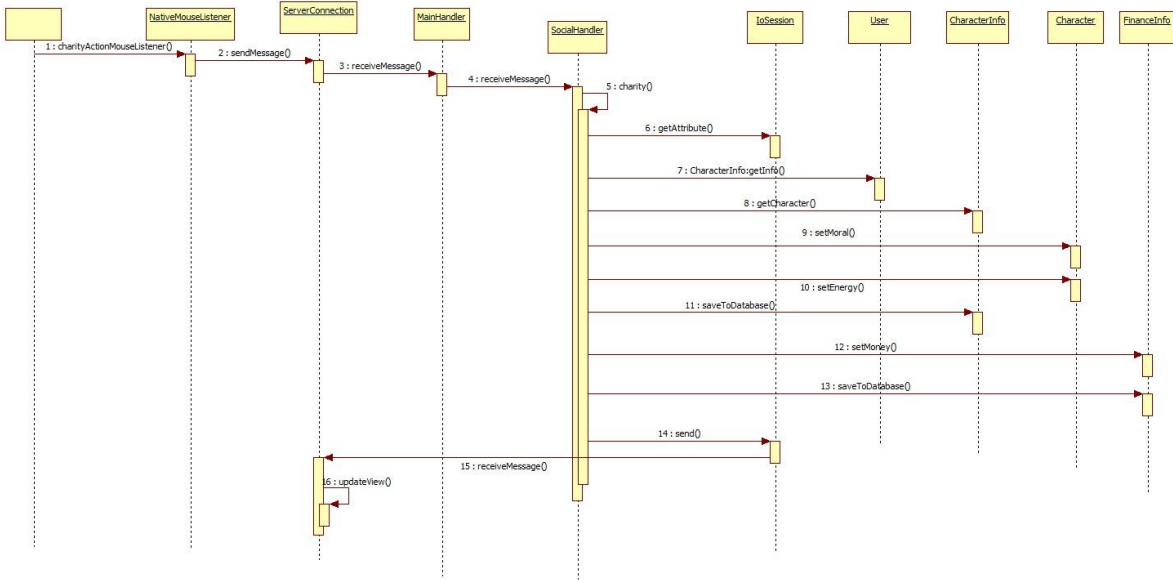
This sequence diagram shows all buy related use cases.



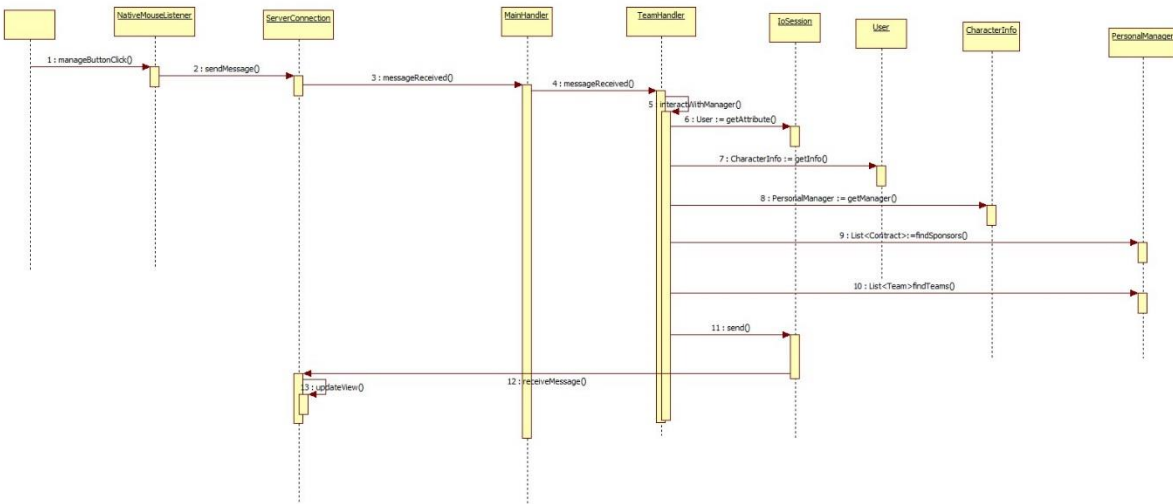
This sequence diagram shows the use case of change Hair/Face Style.



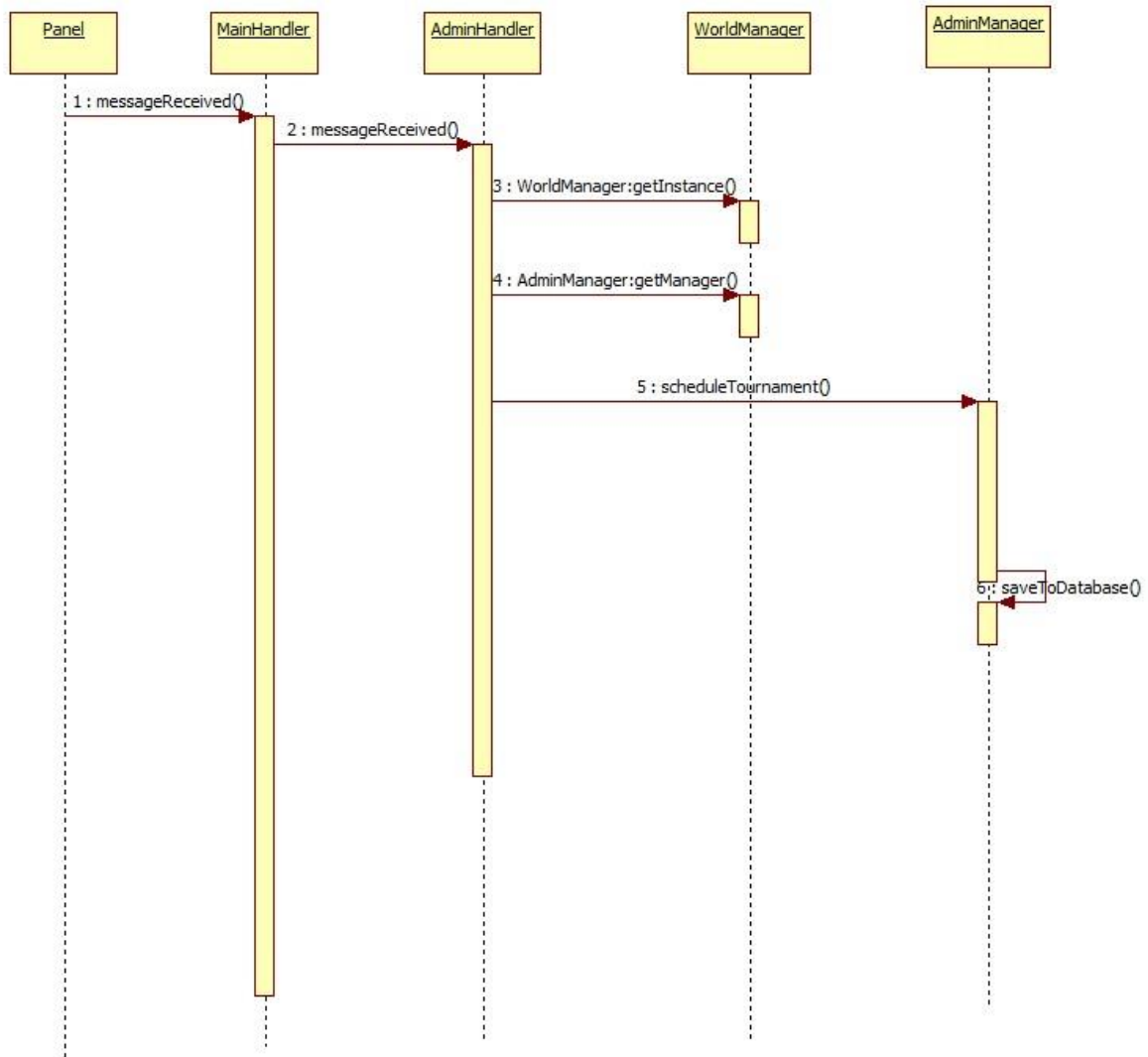
This sequence diagram shows the use case of Choose Game Type.



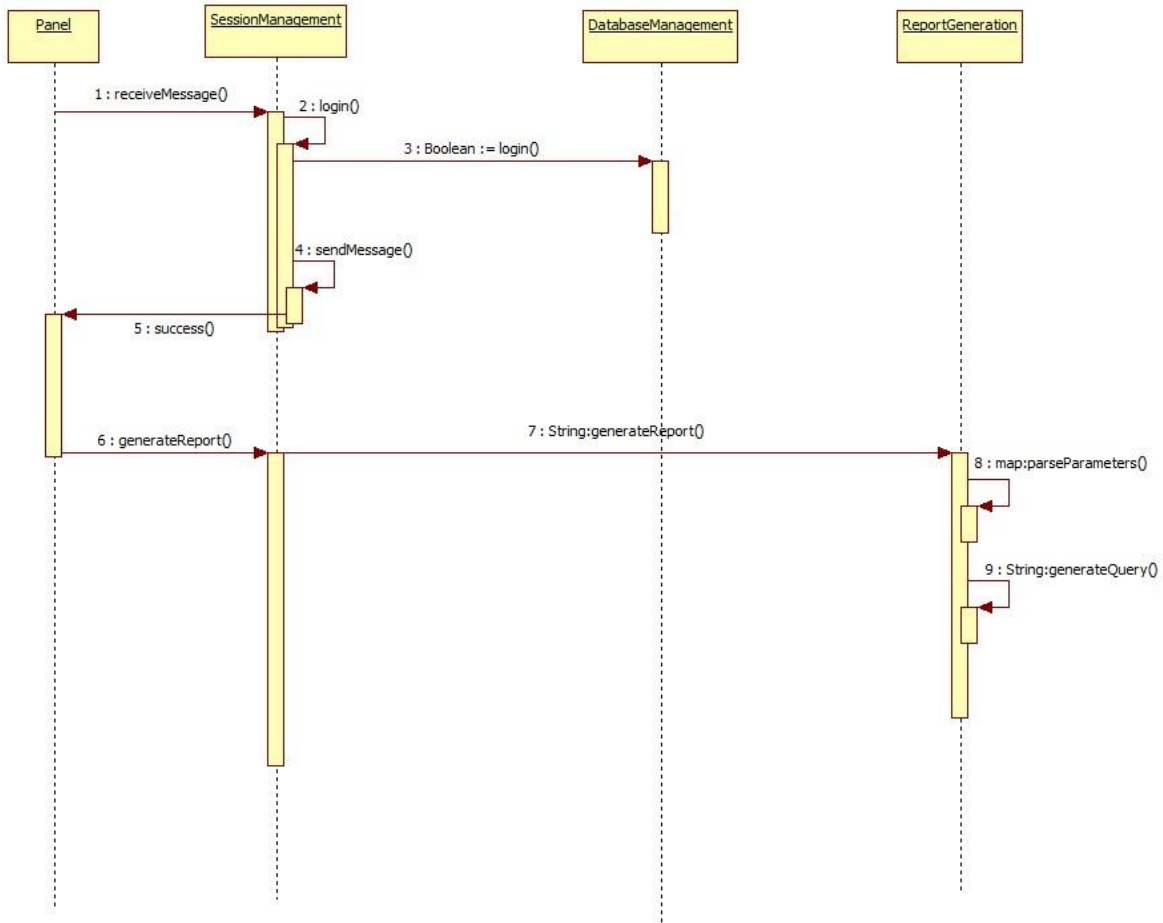
This sequence diagram shows the use case of Charity Work



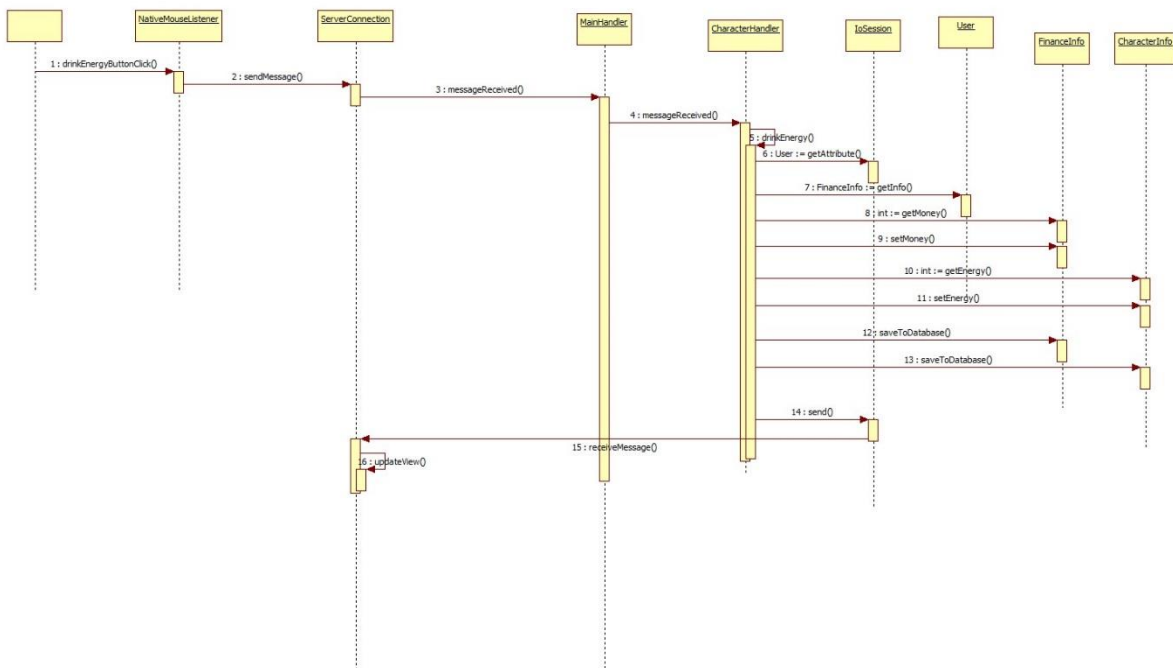
This sequence diagram shows the use case of Interact With Club Manager.



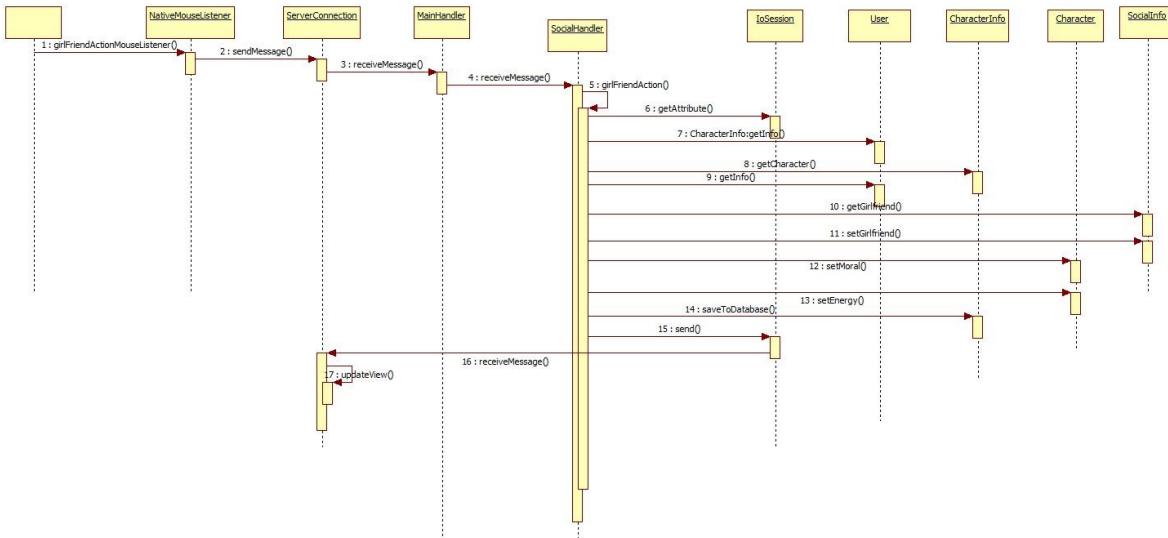
This sequence diagram shows the use case of Create Tournament.



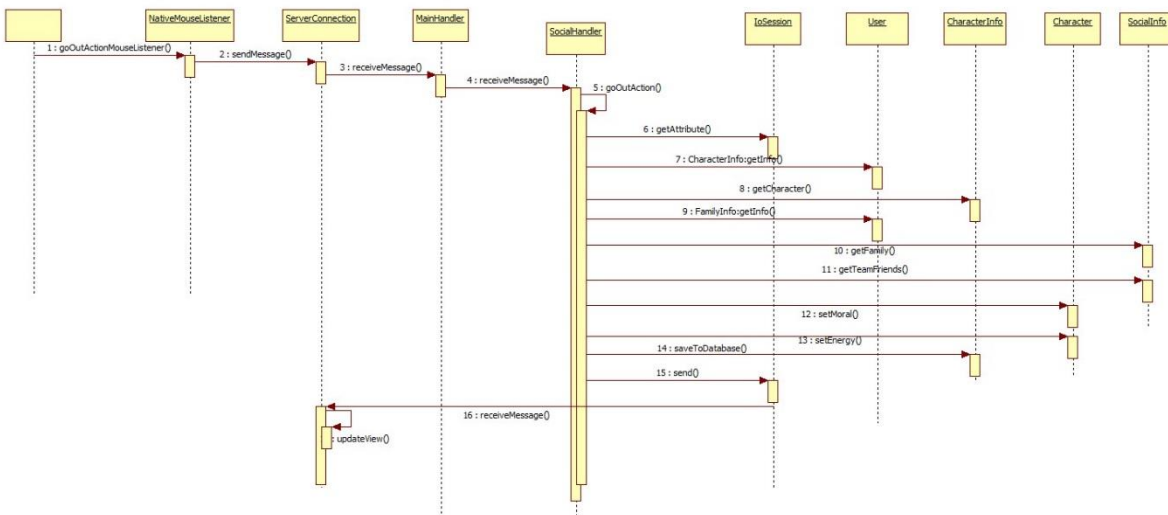
This sequence diagram shows the use case of Generate Report and Login of the Data Analyst.



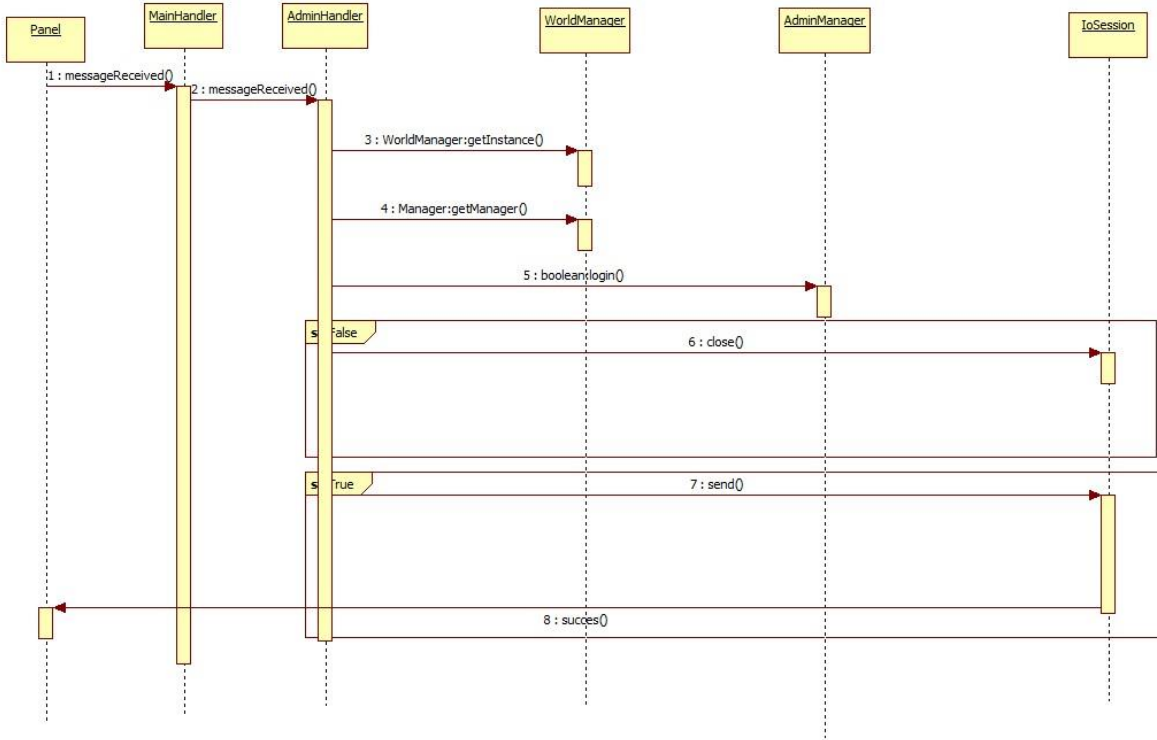
This sequence diagram shows the use case of Drink Energy.



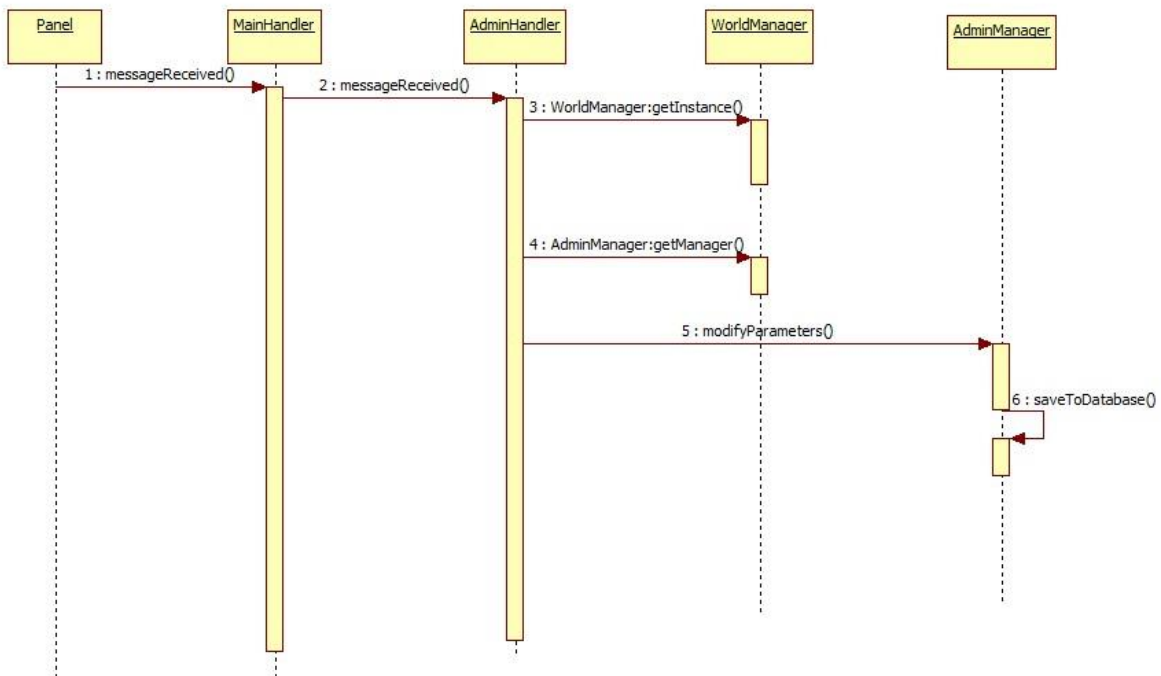
This sequence diagram shows the use cases of all the activities with Girlfriend.



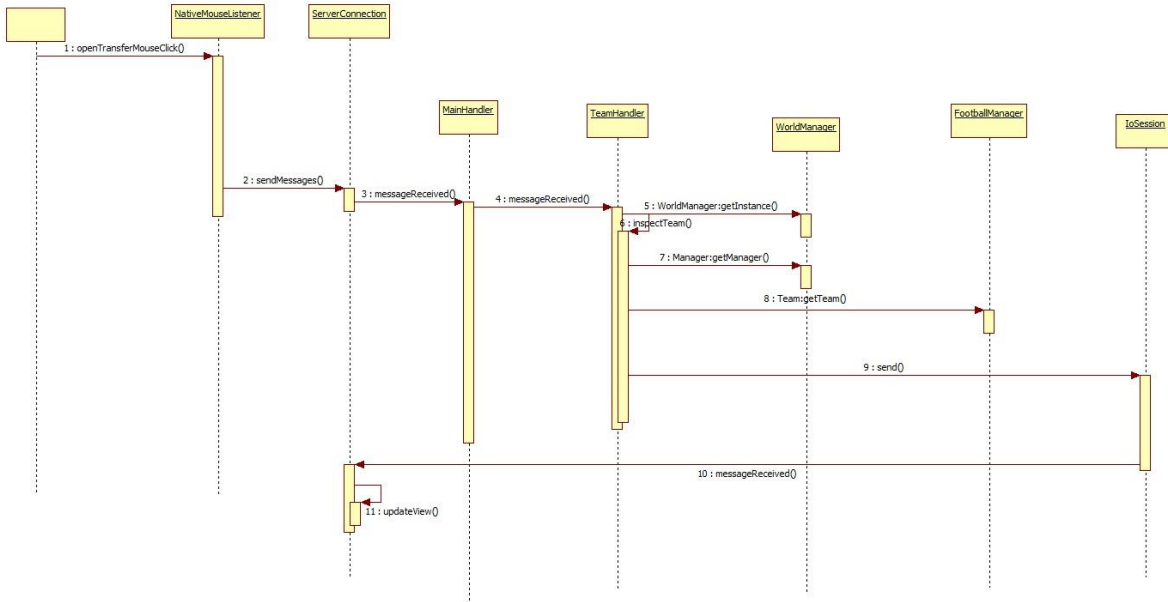
This sequence diagram shows the use cases of all the Go Out Activities.



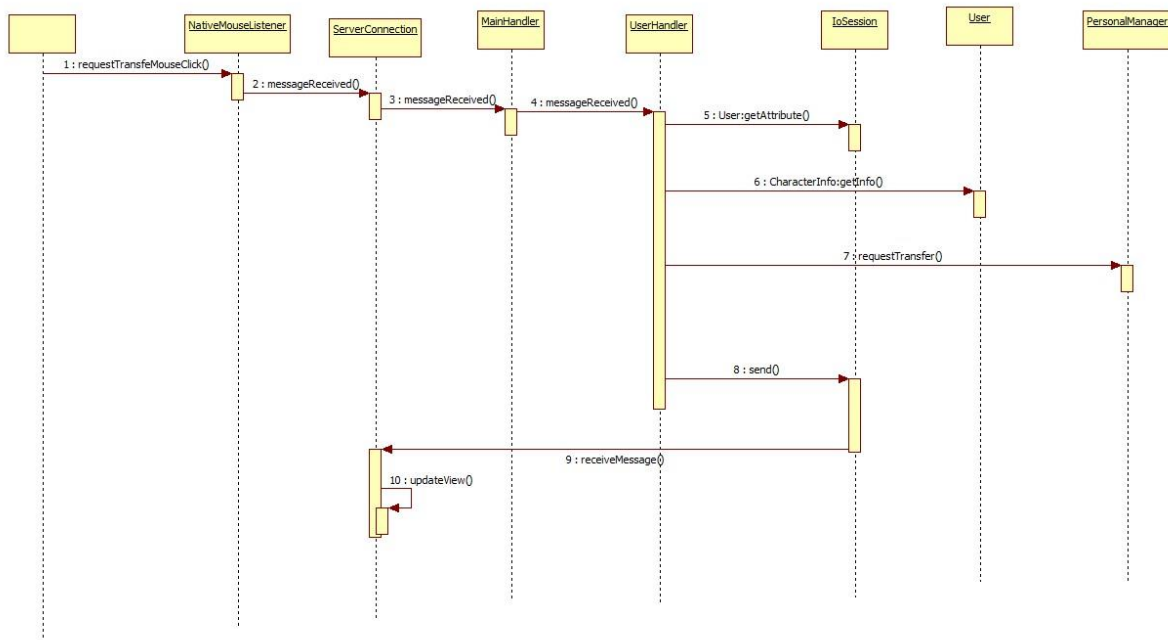
This sequence diagram shows the use case of Administration Login.



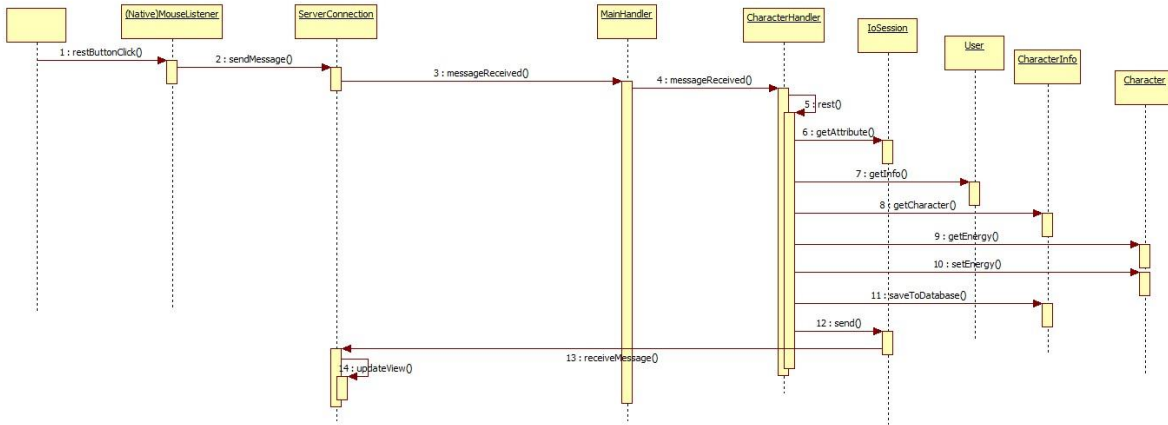
This sequence diagram shows the use case of modify Parameters.



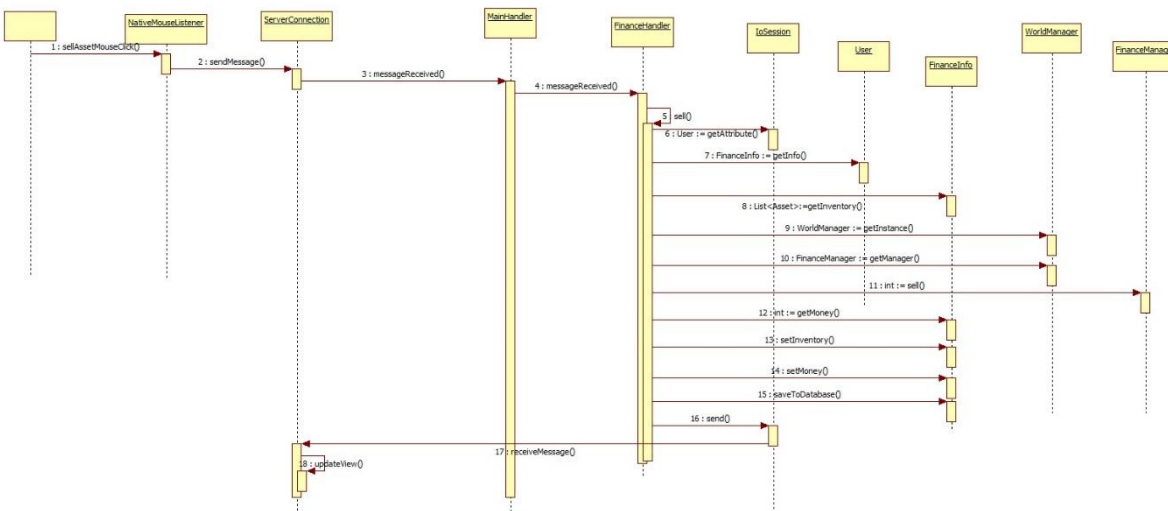
This sequence diagram shows the use case of Open Transfer Menu.



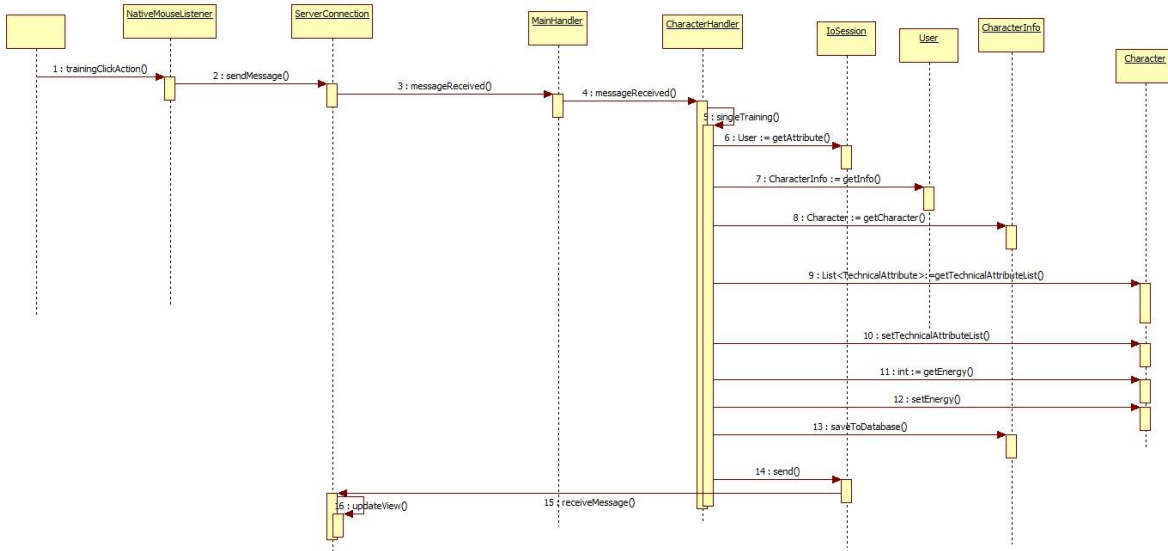
This sequence diagram shows the use case of Request Transfer.



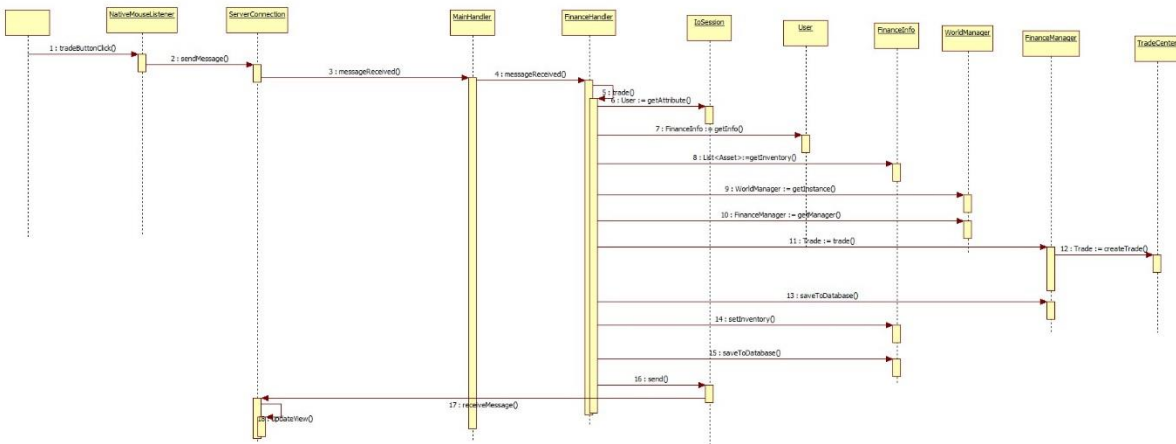
This sequence diagram shows the use case of Rest.



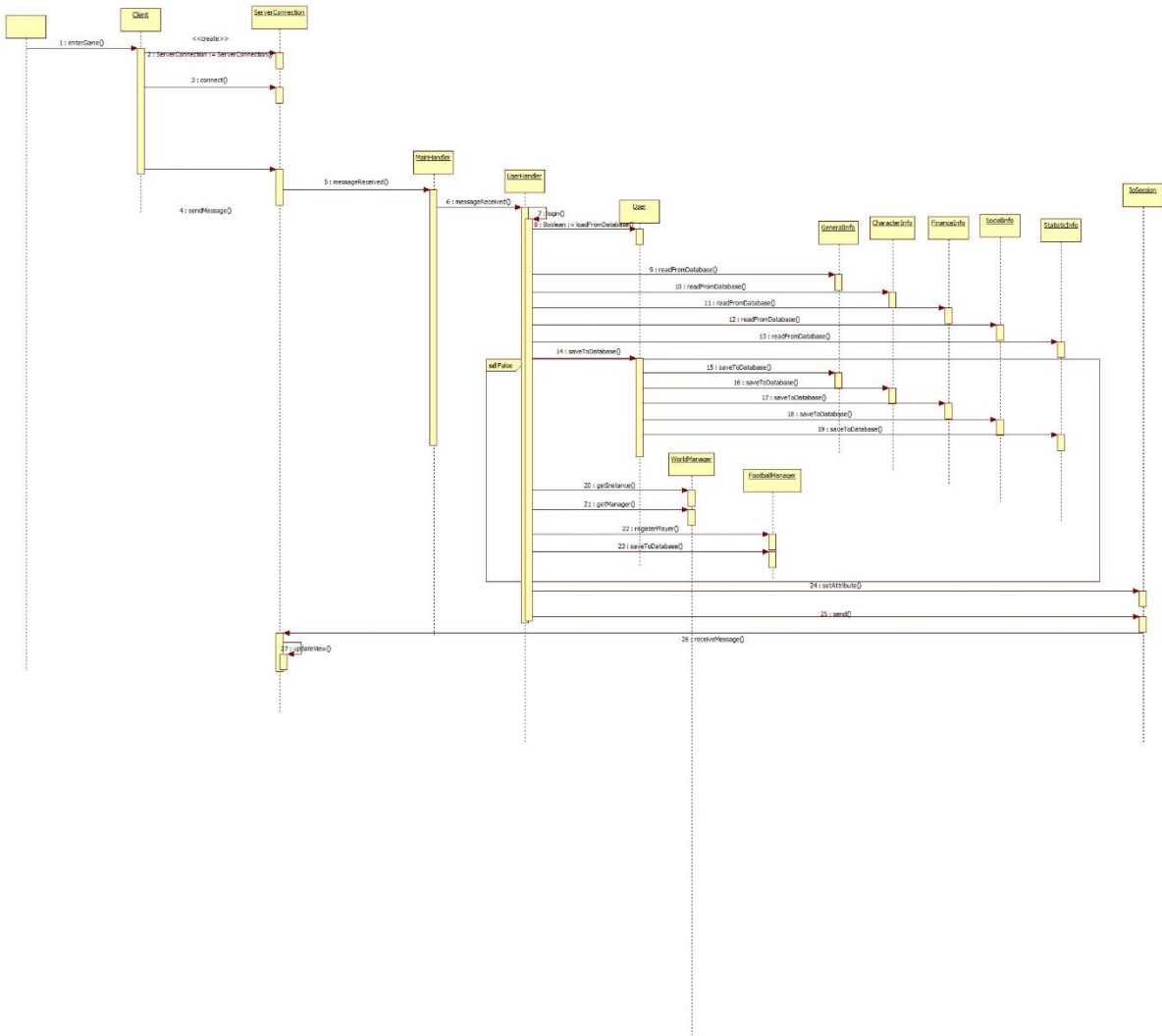
This sequence diagram shows the use case of all sell related actions.



This sequence diagram shows the use case of Single Training.



This sequence diagram shows the use case of Trade.



This sequence diagram shows the use case of Auction.