ACE System High-Level Design

Team 6

November 5, 2007

Contents

1 Introduction

This document contains all relevant information about the high-level design of the ACE System. It first presents the global architecture of the system and then focus on the specific parts that are considered most important to the high-level design. All possible use-cases are listed and the most important ones are explored more thoroughly. This document also describes in a general fashion the main component classes of the system. Finally, all tasks are assigned to different members of our team.

2 Architecture

The ACE system will adopt the client-server architecture. The main reasons behind the decisions are presented in the following list:

- ***usability:*** the server side of the system will export a well defines set of services accessible through a dedicated protocol. Knowledge of this protocol is enough to make use of the system's services.
- ***flexibility:*** modifications to internal components can be done transparently as long as the protocol is not changed. E.g., a new DBMS can be installed without the clients ever noticing it.
- ***interoperability:*** anyone with knowledge of the protocol (and appropriate authentication information) can use the system's services and write custom client-side software.
- ***scalability:*** any parameter of the server can be improved/extended without affecting the other parts.

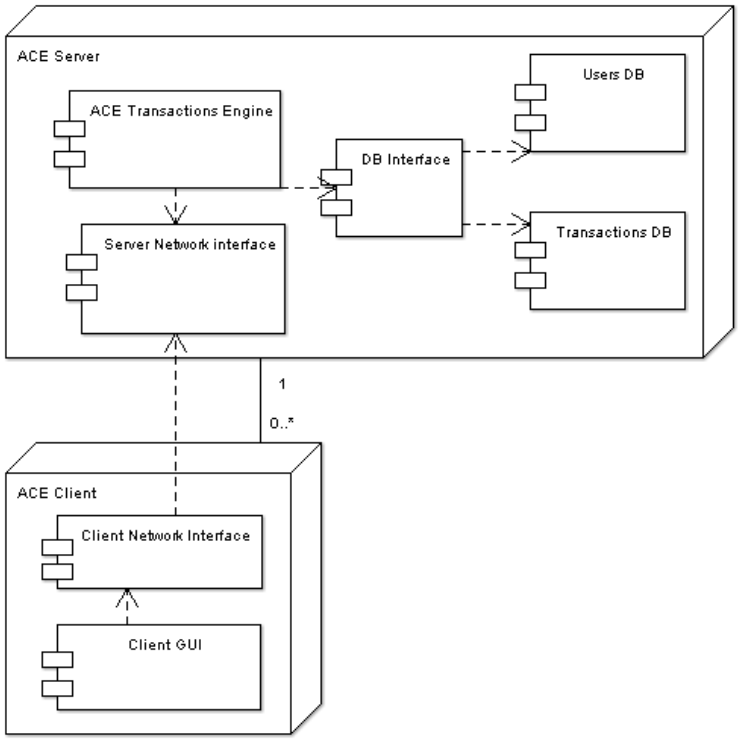

Figure 1: The deployment diagram of the ACE system showing the main

As it can be seen in Figure 1, the system is composed of a server which interacts with one or more clients.

The ACE Server sub-system contains the following components:

- **ACE Transactions Engine:** the component that actually realizes the core functionality of the system. It is responsible for carrying the actual financial transactions. This component uses the rest of the components.
- **Users DB** and **Transactions DB:** store all the information about the users and transactions.
- **DB Interface:** wraps the functionality of the DB components in easy to use methods. If the server needs to store data using different DBMS back-ends (e.g. MySQL, Oracle, Berkley DB XML, etc.) this component will provide a consistent interface to all of them.
- **Server Network Interface:** responsible for the communication with the clients and for the network message exchange. It follows a standard protocol that consists of private-key encrypted pre-defined messages. There are certain messages that the server will accept from clients through an encrypted channel.

On the other side, the ACE Client sub-system contains the following components:

- **Client GUI:** the collection of all the graphical tools available to the end-user that display data in various formats but also allows him or her to interact with the system.
- **Client Network Interface:** responsible for packaging the GUI commands into network messages according to the ACE protocol.

3 Use-case Model

3.1 Actors

There are three actors in ACE system:

1. **End-User**
   Role: Interacts with an ACE client to place a various orders or view information about his account and the ACE market. Here, "Client" denotes any application that connects to the server using the MessageServer class, thus following the standard protocol recognized by the Server.
   Participates in use-cases: End-User related use-cases (*see the end-user use-case diagram*)
2. **Administrator**
   Role: Manages user accounts and the currencies of the ACE system.
   Participates in use-cases: Administrator related use-cases (*see the administrator use-case diagram*)
3. **ACE Database**
   Role: Records all the data of the ACE system, including user accounts and information, all orders that were placed, default leverage ratio, and interest rate.
   Participates in use-cases: All use-cases

3.2 Use-case Diagrams

In Figure 2 and Figure 3 presents the use-case diagrams for ACE system. They have been split up into two figures for layout purposes.

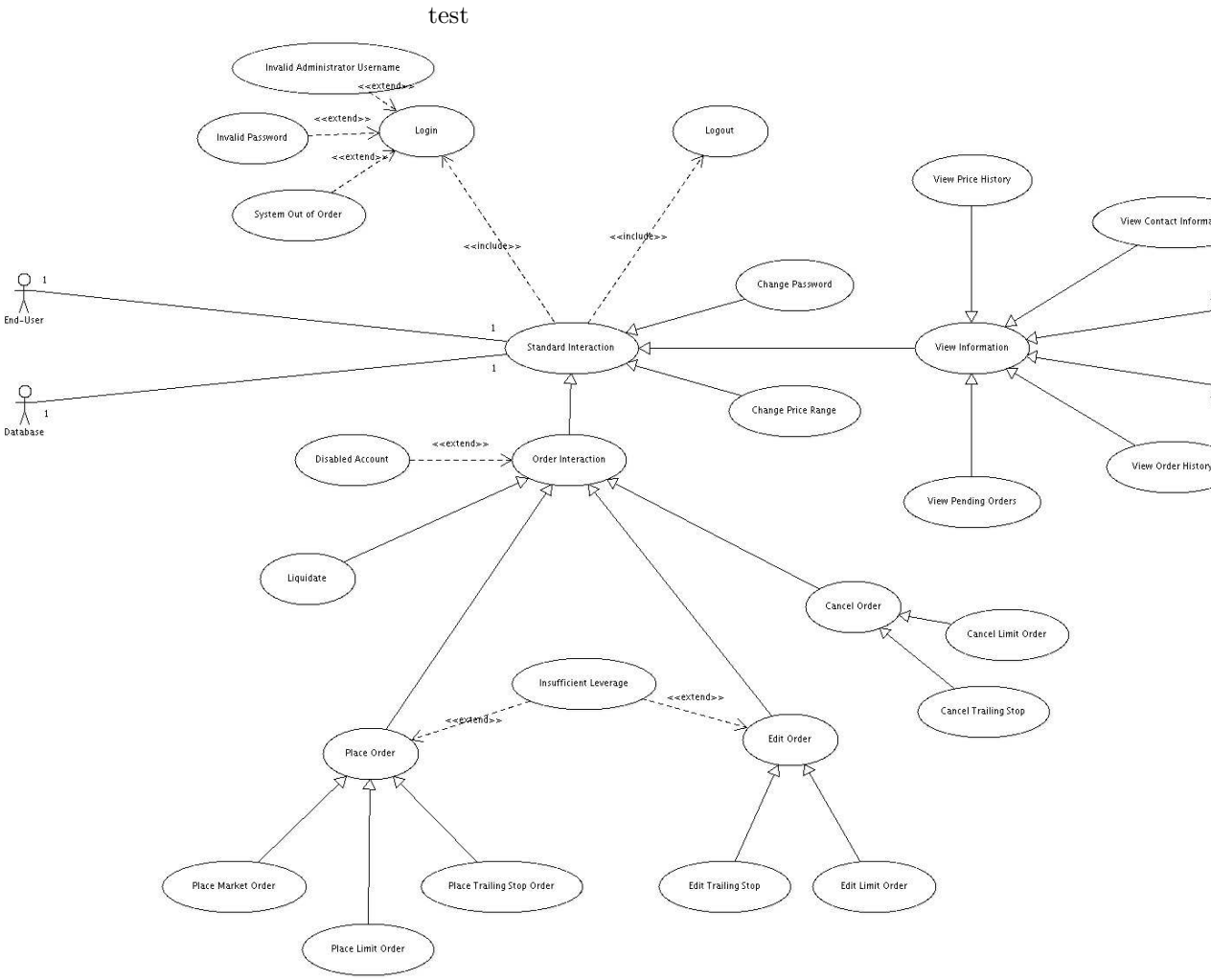

Figure 2: End-User Use-case Diagram

3.3 Use-case Listing

Following is a list of use cases for the **end-user**:

- **Login:** The end-user successfully logs into the ACE server.
- **Invalid Username:** When the end-user attempts to log into the ACE server but provides a username that is not in the ACE database, the ACE client will present to the end-user an error message that the username is invalid.
- **Invalid Password:** When the end-user attempts to log into the ACE server but provides the wrong password, the ACE client will present to the end-user an error message stating that the password is invalid.
- **System out of order:** When the end-user attempts to log into the ACE server but the ACE server is not available at the moment for whatever reason (for example the ACE server crashed or the end-user is not connected to the Internet), the ACE client presents the error message that the ACE server is out of order.
- **Logout:** The end-user logs out of the ACE server.
- **Change Password:** The end-user changes his login password.
- **Liquidate:** The end-user decides to liquidate all his currencies, and the ACE client asks the end-user if he is sure before proceeding with his demand.
- **Change Price Range:** The end-user successfully changes his preference for the price range of all his market orders. This means that when a market order is placed, the current market rate, with a price range above and below this current rate, is the rate at which the end-user will buy or sell.
- **Place Market Order:** The end-user places a market order, specifying the currency pair, amount, and whether to buy or sell.
- **Place Limit Order:** The end-user places a limit order, specifying whether it is a limit order or a stop loss, the currency pair, amount, the limit, and whether to buy or sell.
- **Place Trailing Stop:** The end-user places a trailing stop order, specifying the currency pair, amount, the number of trailing points, and whether to buy or sell.
- **Edit Limit Order:** While viewing all of his pending orders, the end-user chooses to edit one of his limit orders in order to change one of its attributes.
- **Edit Trailing Stop:** While viewing all of his pending orders, the end-user chooses to edit one of his trailing stop orders in order to change one of its attributes.
- **Cancel Limit Order:** While viewing all of his pending orders, the end-user chooses to cancel one of his limit orders.

- **Cancel Trailing Stop:** While viewing all of his pending orders, the end-user chooses to cancel one of his trailing stop orders.
- **Disabled Account:** If an end-user's account was disabled by the administrator and that end-user attempts to liquidate or place/modify/cancel an order, then the ACE client will present the error message that the account being used was disabled and the administrator should be contacted. The end-user is of course allowed to log into the ACE server, even though his account is disabled, so he may view his billing information.
- **View Pending Orders:** The end-user chooses to view all his pending orders and the ACE client presents the user with a table with this information. It is from this table which the end-user may edit or cancel any pending orders. The end-user may choose to view only those orders involving a particular currency, or for all currencies.
- **View Order History:** The end-user chooses to view his order history, and the ACE client presents this information in a table.
- **View Currency Balances:** The end-user chooses to view his currency balances, and the ACE client presents a table containing all currencies of ACE and the amount of each currency the end-user has.
- **View Billing Information:** The ACE client presents the end-user's billing information.
- **View Price History:** The ACE client presents in the form of a chart the history of a currency pair rate. The end-user may choose which currency pair to display as well as the time period.
- **View Contact Information:** The ACE client presents the end-user with his phone number and email. Should the information be incorrect, the end-user should call the administrator to fix it.

Following is a list of use cases for the **administrator user**:

- **Login:** The administrator successfully logs into the ACE server.
- **Invalid Administrator Username:** When the administrator attempts to log into the ACE server but provides a username that is not in the ACE database, the ACE client will present to the administrator with an error message that the username is invalid.
- **Invalid Password:** When the administrator attempts to log into the ACE server but provides the wrong password, the ACE client will present to the administrator an error message stating that the password is invalid.
- **System out of order:** When the administrator attempts to log into the ACE server but the ACE server is not available at the moment for whatever reason (for example the ACE server crashed or the end-user is not connected to the Internet), the ACE client presents the error message that the ACE server is out of order.

- **Logout:** The administrator logs out of the ACE server.
- **Create Account:** The administrator successfully creates a new end-user account, after filling out a form including a end-username and temporary password.
- **Username Already in Use:** When the administrator attempts to create a new end-user account but the specified end-username is already taken, the ACE client will present that the end-username is already in use.
- **Add Currency:** The administrator successfully adds a new currency to the ACE database.
- **Currency Already Exists:** When the administrator attempts to add a new currency, but the specified currency name is already in use, the ACE client will present the message that the currency already exists.
- **Modify Currency:** The administrator changes the name of a currency.
- **Remove Currency:** The administrator chooses a currency from a list and that currency is removed from the active ACE market. All pending orders involving the selected currency will be canceled. To preserve the integrity of the database, the currency will not be removed from it.
- **Change Default Transaction Fee:** The administrator changes the transaction fee applied by default to all newly created end-user accounts.
- **Change Default Interest Rate:** The administrator changes the interest rate applied by default to all newly created end-user accounts.
- **Change Default Leverage Ratio:** The administrator changes the leverage ratio applied by default to all newly created end-user accounts.
- **Acquire End-User Account:** The administrator specifies an end-user username, and the ACE client will display a menu filled with the various ways that the administrator may interact with that end-user account. This is synonymous to a login.
- **Invalid End-User username:** If the administrator attempts to acquire an end-user account but specifies an end-username that does not exist, the ACE Client will present a message stating that the end-user username is invalid.
- **Release End-User Account:** After the administrator has acquired an end-user account, the administrator can then choose to release this end-use account. By doing so, the ACE client presents a menu asking for an end-user name. This replaces the menu showing ways that the administrator can interact with an end-user account. This is synonymous to a logout.
- **Delete Account:** The administrator deletes an end-user account. This would imply that all pending orders involving this end-user account will be canceled, all of that user's currencies will be liquidated, no one may login using that account, and administrators may no longer acquire that user account. However, this end-user account will not be deleted from the database in order to preserve the database's integrity.

- **Disable Account:** The administrator disables an end-user account. The point is to disable a enabled end-user account, but if the account is disabled to begin with, there is no need to present an error message.
- **Enable Account:** The administrator enables an end-user account. The point is to enable a disabled end-user account, but if the account is enabled to begin with, there is no need to present an error message.
- **Deposit Funds:** The administrator successfully deposits funds into an end-user's account.
- **Withdraw Funds:** The administrator successfully withdraws funds from an end-user's account.
- **Customize Leverage Ratio:** The administrator changes an end-user's leverage ratio.
- **Customize Transaction Fee:** The administrator changes an end-user's transaction fee.
- **Customize Interest Rate:** The administrator changes an end-user's interest rate.
- **Change Contact Information:** The administrator successfully modifies an end-user's contact information.
- **View Account Summary:** The ACE client presents to the administrator the end-user's account information.
- **View Billing Information:** The ACE client presents to the administrator the end-user's billing information.
- **View Contact Information:** The ACE client presents to the administrator the end-user's contact information.

3.4 Use-case Description

Name:	**Place Market Order**
Purpose:	Demonstrate how an end-user can place a new market order.
Actors:	End-User, ACE Database
Entry condition:	The end-user is logged in.
Exit condition:	A market order is successfully placed.
Flow of events:	1. The end-user selects a currency pair. 2. The ACE client presents the latest rate for the selected currency pair. 3. The end-user specifies the amount. 4. The end-user chooses to buy or sell. 5. The market order is validated. 6. The market order is stored in the ACE database and acknowledged.
Exceptions:	• Market order could not be validated because the end-user attempted to buy/sell more than he is able to. • The end-user account is disabled.

Name:	**Place Limit Order**
Purpose:	Demonstrate how an end-user can place a new limit order.
Actors:	End-User, ACE Database
Entry condition:	The end-user is logged in.
Exit condition:	A limit order is successfully placed.
Flow of events:	1. The end-user selects a currency pair. 2. The ACE client presents the latest rate for the selected currency pair. 3. The end-user specifies the amount. 4. The end-user chooses between a limit stop and a stop loss order. 5. The end-user specifies the limit at which the order is triggered. 6. The end-user chooses to buy or sell. 7. The limit order is validated. 8. The limit order is stored in the ACE database and acknowledged.
Exceptions:	• Limit order could not be validated because the end-user attempted to buy/sell more than he is able to. • Limit order could not be validated because the end-user specified an impossible limit (explain). • The end-user account is disabled.

Name:	**Place Trailing Stop**
Purpose:	Demonstrate how to place a new trailing stop order.
Actors:	End-User, ACE Database
Entry condition:	The end-user is logged in.
Exit condition:	A new limit order is successfully placed.
Flow of events:	1. The end-user selects a currency pair. 2. The ACE client presents the latest rate for the selected currency pair. 3. The end-user specifies the amount. 4. The end-user specifies the number of trailing points. 5. The end-user chooses to buy or sell 6. The trailing stop order is validated. 7. The trailing stop order is stored in the ACE database and acknowledged.
Exceptions:	• Trailing stop order could not be validated because the end-user attempted to buy/sell more than he is able to. • The end-user account is disabled.

Name:	**Login (End-User)**
Purpose:	Demonstrate how an end-user logs in to the ACE server.
Actors:	End-User, ACE Database
Entry condition:	The end-user is running an ACE client.
Exit condition:	The end-user successfully logged in.
Flow of events:	1. The end-user selects the login tab. 2. The ACE client presents the login form. 3. The end-user enters a username and password. 4. The username and login pair is validated with the ACE database. 5. The ACE client presents the client with a message that he is currently logged into the ACE system.
Exceptions:	• The username does not exist in the ACE database. • The password does not match that username. • The ACE system is currently out of service.

Name:	**Create Account**
Purpose:	Demonstrate how an administrator can create a new end-user account.
Actors:	Administrator, ACE Database
Entry condition:	The administrator is logged in.
Exit condition:	The end-user account was created successfully.
Flow of events:	1. The administrator clicks on "create account". 2. The ACE client presents the administrator with a "create account form". 3. The administrator types a username, a temporary password, a contact phone number and email address. 4. The administrator sends the request to the ACE server. 5. The ACE server checks the database to see whether the username is available. 6. The ACE server adds a new account to the database. 7. The ACE server acknowledges the creation of account.
Exceptions:	• That username is already in use.

Name:	**Disable Account**
Purpose:	Demonstrate how an administrator can disable an account.
Actors:	Administrator, ACE Database
Entry condition:	The administrator is logged in.
Exit condition:	The account is disabled.
Flow of events:	1. The administrator clicks on "disable account" 2. The ACE client presents the administrator with a "disable account form". 3. The administrator types the username of the account to be disabled. 4. The administrator sends the request to the ACE server. 5. The ACE server asks the database to disable the account. 6. The ACE server acknowledges that the account was disabled.
Exceptions:	NONE

Name:	**View Billing Account**
Purpose:	Demonstrate how an end-user can view his billing account.
Actors:	End-user, ACE Database
Entry condition:	The end-user is logged in.
Exit condition:	The billing account information is sent to the ACE client.
Flow of events:	1. The end-user clicks on the "Billing Tab". 2. The ACE client sends the request to the ACE server. 3. The ACE server queries the database for the billing information about the end-user. 4. The ACE server sends the billing information to the end-user.
Exceptions:	NONE

Name:	**Display Chart**
Purpose:	Demonstrate how a chart is displayed to the end-user.
Actors:	End-user, ACE Database
Entry condition:	The end-user is logged in.
Exit condition:	A market price history chart.
Flow of events:	1. The end-user clicks on the "Chart Tab". 2. The end-user chooses the currency pair. 3. The end-user chooses a time scale. 4. The ACE client requests a market price history to the ACE server. 5. The ACE server queries the database for the market price history. 6. The ACE server sends the market price data back to the ACE client. 7. The ACE client draws a chart on the GUI using the market price data.
Quality requirements:	The chart should include one curve for ask prices and another for bid prices.
Exceptions:	NONE

Name:	**Customize Transaction Fee**
Purpose:	Demonstrate how an administrator can customize the transaction fee of one specific user.
Actors:	Administrator, ACE Database
Entry condition:	The administrator is logged-in.
Exit condition:	The user account is updated with a customized transaction fee.
Flow of events:	1. The administrator clicks on "Customize Fee". 2. The ACE client presents the administrator with a "customize fee form". 3. The administrator types a username and a transaction fee. 4. The administrator sends a request to the ACE server. 5. The ACE server modifies the user-specific customized fee entry in the database. 6. The ACE server acknowledges the modified user-specific transaction fee.
Exceptions:	NONE

Name:	**Liquidate All Open Positions**
Purpose:	Demonstrate how an end-user can liquidate all his open positions.
Actors:	End-user, ACE Database
Entry condition:	The end-user is logged-in. The ACE client already fetched the open position data.
Exit condition:	All open positions were liquidated.
Flow of events:	1. The end-user clicks on the "Position Tab". 2. The list of all open position is displayed along with related information. 3. The end-user clicks on "Liquidate". 4. The ACE client sends a request to the ACE server. 5. The ACE server queries the database for the user's balance for every of his trading currencies. 6. The ACE server sets up a market order for every non-null currency balance. 7. The ACE server updates the database with the adjusted the billing account. 8. The ACE server acknowledges the liquidate operation.
Exceptions:	• The account is disabled.

4 Static Model

4.1 Class Diagrams

4.2 Class Listing

Name:	**ACEServer**
Role:	Decides which action to perform based on user's input. It's a class containing a model of logic of the ACE. (Main driver class)
Attributes:	ArrayList<User> (collection of users), DBConnection, MessageServer
Backward traceability:	Requirements Document (3)
Related use cases:	All use-cases

Name:	**ACEClient**
Role:	Manages all the interaction between the user and the interface, as well as the communication with the server.
Attributes:	MessageServer
Backward traceability:	Requirements Document (3)
Related use cases:	All use-cases

Name:	**DBConnection**
Role:	Only implemented in ACEServer class. Maintains an active connection to the DB server, allowing for multiple read/write/update queries to be sent at the same time, asynchronously. It can easily connect to any type of relational database. When a command is issued, after it is checked for validity by the logic of the application, the security token of the sender is mapped to an internal map: end-users all map to an internal DB user that only has certain, clearly defined rights on certain tables. This ensures a second layer of security. Administrator users are mapped to an "Operator" user that has both read/write writes to the DB (also, strictly and precisely defined within the DB itself) but cannot modify its structure.
Attributes:	DBConnectionString, SQLCommandInbox, SQLCommandOutbox, SecurityToken.
Backward traceability:	
Related use cases:	All relating to ACEServer.

Name:	**UserAccount**
Role:	Represents a specific user's account. It is used by the System class.
Attributes:	String userName, int userBalance, int billingAmount, arrayList<Order> (collection of orders), String billingHistory,
Backward traceability:	Requirements Document (3.1.2-3.1.3)
Related use cases:	Create Account , Login, Logout , View Pending Orders, Disable Account

Name:	**Currency**
Role:	Represents a specific currency that a user can choose to sell or buy.
Attributes:	double buyPrice, double sellPrice, String currencyName
Backward traceability:	Requirements Document (3.1.2.2)
Related use cases:	View Currency Balances

Name:	**Order**
Role:	Represents a specific order that a user can make.
Attributes:	String orderType, currency pair, int buy/sell price, int amount to be bought/sold, int timeSpan, double PriceRange
Backward traceability:	Requirements Document (3.1.2.2)
Related use cases:	Place Limit/Market Order, Cancel Limit/Market Order

Name:	**Billing**
Role:	Represents a specific user's Billing information. It is used by the UserAccount class.
Attributes:	String userName, int billingAmount, arrayList<Order> (collection of orders), String billingHistory, String transactionHistory
Backward traceability:	Requirements Document (3.2.1.7)
Related use cases:	View Billing Information

Name:	**Mail**
Role:	Represents an email that will be sent to the end-user.
Attributes:	textObject
Backward traceability:	Requirements Document (3.3.1.3)
Related use cases:	All containing billing actions performed by ACEServer

Name:	**OrderPanel**
Role:	It is a class that gives a visual presentation of Order options and actions for user.
Attributes:	buttons, fields
Backward traceability:	Requirements Document (3.2)
Related use cases:	Place Limit Order, Place Market Order, Cancel Limit Order, Place Market Order

Name:	**MarketPanel**
Role:	It is a class that gives a visual presentation of currency price
Attributes:	tabs, fields, tables
Backward traceability:	Requirements Document (3.2)
Related use cases:	View Price History, Display Chart

Name:	**AccountPanel**
Role:	It is a class that visually presents a summary of the margin accounts, open positions and pending orders, and transaction logs.
Attributes:	tabs, fields, tables
Backward traceability:	Requirements Document (3.2)
Related use cases:	Create Account, Login, Logout

Name:	**LoginPanel**
Role:	allows end-user to login/logout to/from the System
Attributes:	tabs, buttons
Backward traceability:	Requirements Document (3.1.1)
Related use cases:	Login/Logout

Name:	**VisualRepresentationPanel**
Role:	Allows the user to interact with the ACE system. Serves as a container for other GUI classes.
Attributes:	tabs, buttons, panels.
Backward traceability:	Requirements Document (3.1.1)
Related use cases:	All use-cases

Name:	**MessageServer**
Role:	Performs all the message-parsing and encrypted message communication between the server and client. It manages connectivity status, encrypted channel and keys, and the set of possible messages to be sent/received. It is the same on both server and client. It has a "mailbox" (buffer) for storing multiple messages that need to be sent or that have been received. Note: any functions that are performed by a Client (regardless whether the Client is the one that we have and built or a custom application created by an end-user) will go through this component.
Attributes:	ConnectionStatus, PrivateKey, PublicKey, Inbox, Outbox, MessageSet.
Backward traceability:	Not in Requirement Doc
Related use cases:	All use-cases

5 Traceability Matrix

5.1 Requirements Document

This subsection lists all the functional requirements taken from the Requirements document, and enumerates all related use-cases and system classes.

ID	Name	Use-cases	Classes
3.1.1.1	login credential	Login Invalid Username Invalid Password System Out of Order Invalid Administrator Username	LoginPanel
3.1.1.2	logout request	Logout	LoginPanel
3.1.1.3	change password	Change Password	LoginPanel UserAccount
3.1.2.1	change price range request	Change Price Range	UserAccount
3.1.2.2	instant buy/sell request	Place Market Order	Order
3.1.2.3	limit order request	Place Limit Order Edit Limit Order Cancel Limit Order	Order
3.1.2.4	trailing stop	Place Trailing Stop Edit Trailing Stop Cancel Trailing Stop	Order
3.1.2.5	view billing account	View Billing Information	Billing
3.1.2.6	query pending orders	View Pending Orders	UserAccount Order
3.1.2.7	query orders history	View Order History	UserAccount Order
3.1.2.8	check currency balances	View Currency Balances	Currency
3.1.2.9	price history	View Price History	MarketPanel
3.1.2.10	liquidate	Liquidate	Order

3.1.3.1	create account	Create Account Username Already In Use	AccountPanel UserAccount ACEServer
3.1.3.2	enable/disable account	Disable Account Enable Account	AccountPanel UserAccount ACEServer
3.1.3.3	delete account	Delete Account	AccountPanel UserAccount ACEServer
3.1.3.4	change user contact information	Change Contact Information	AccountPanel UserAccount ACEServer
3.1.3.5	deposit funds request	Deposit Funds	AccountPanel UserAccount ACEServer
3.1.3.6	withdraw funds request	Withdraw Funds	AccountPanel UserAccount ACEServer
3.1.3.7	customize end-user leverage ratio	Customize Leverage Ratio	AccountPanel UserAccount ACEServer
3.1.3.8	customize end-user interest rate	Customize Interest Rate	AccountPanel UserAccount ACEServer
3.1.3.9	change default leverage ratio	Change Default Leverage Ratio	AccountPanel UserAccount ACEServer
3.1.3.10	change default interest rate	Change Default Interest Rate	ACEServer
3.1.3.11	add/modify/remove currency	Add Currency Currency Already Exists Modify Currency Remove Currency	ACEServer Currency
3.2.2.1	view account status	View Account Summary	
3.2.2.2	view contact information	View Contact Information	
3.2.2.3	view billling information	View Billling Information (Admin)	

5.2 Additional Requirements

The following table lists the forgotten requirements that were not listed in the Requirements document.

Name	**Use-cases**	**Classes**
Change Default Fee	Change Default Transaction Fee	ACEClient ACEServer
Customize Fee	Customize End-user Transaction Fee	ACEClient ACEServer
Acquire End-User	Acquire End-User Account	ACEClient ACEServer
Release End-User	Release End-User Account	ACEClient ACEServer
ACE Protocol API	All containing ACE Client	ACEClient ACEServer

6 Task Assignment

- Documentation Formatting (Gabriel, Everyone(Final checks))
- Source Code Standardization (Kirill, Michael)
- System
  - Integration (Alex)
  - Client/Server Network Interface (Alex, Kirill)
  - ACE Transactions Engine (George, Alex)
  - End-User GUI (Gabriel (Major decisions), Everyone(Minor Decisions))
  - Administrator GUI (Gabriel, Kirill)
  - UserAccount (Alex, Gabriel)
  - MarginAccount (Alex, Kirill, Gabriel)
  - BillingAccount (Alex, Kirill)
  - CurrencyAccount (Alex, Kirill)
  - Currency (Kirill, Alex)
  - Order (Michael, George)
  - Billing (Michael)
  - Mail (Michael, Kirill)
- Database
  - Database Design (George)
  - Database Interface (George)
  - Database Management (George)
- Testing & QA (Gabriel, Alex, Everyone (Minor decisions))

test

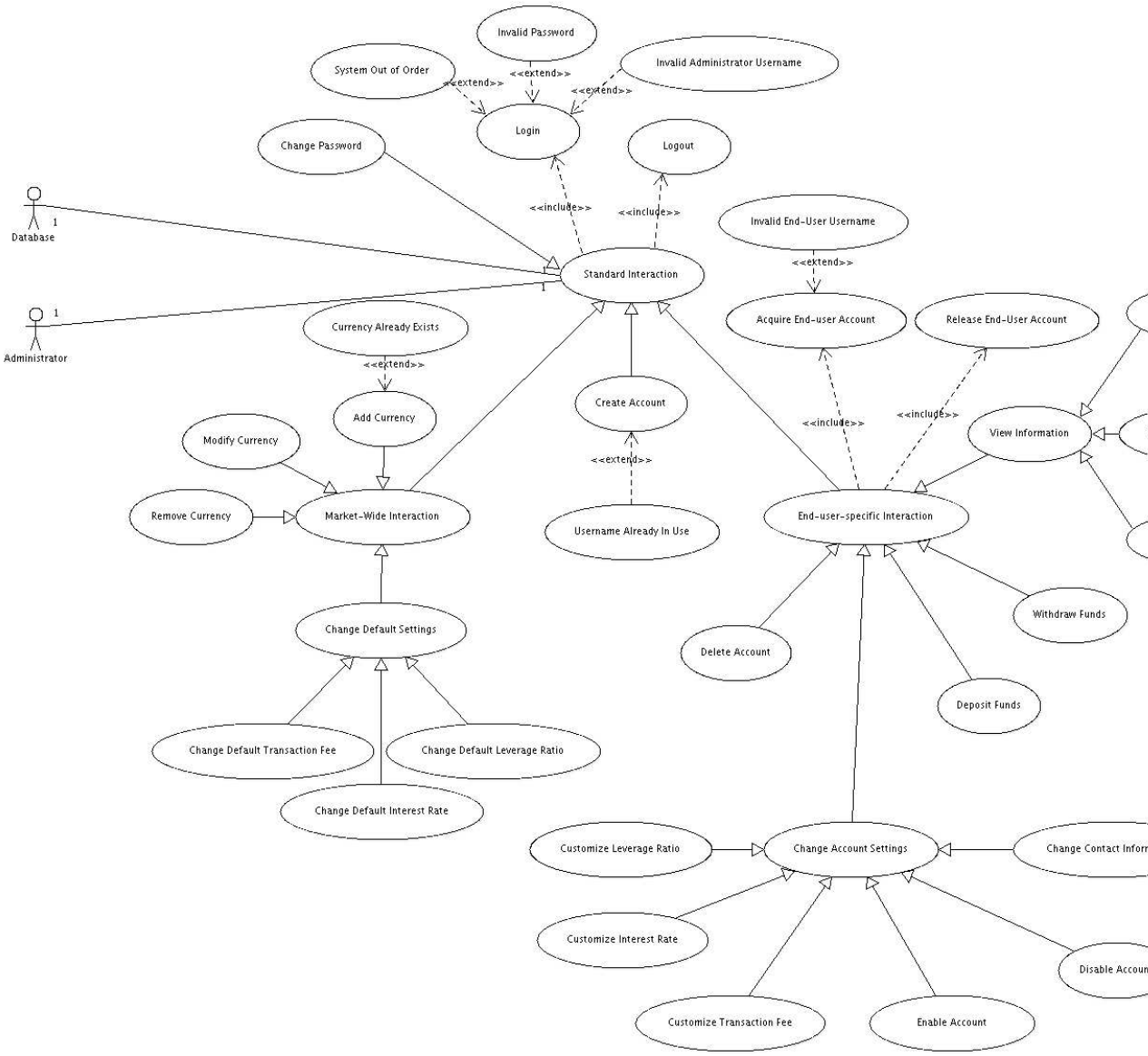

Figure 3: Administrator Use-case Diagram

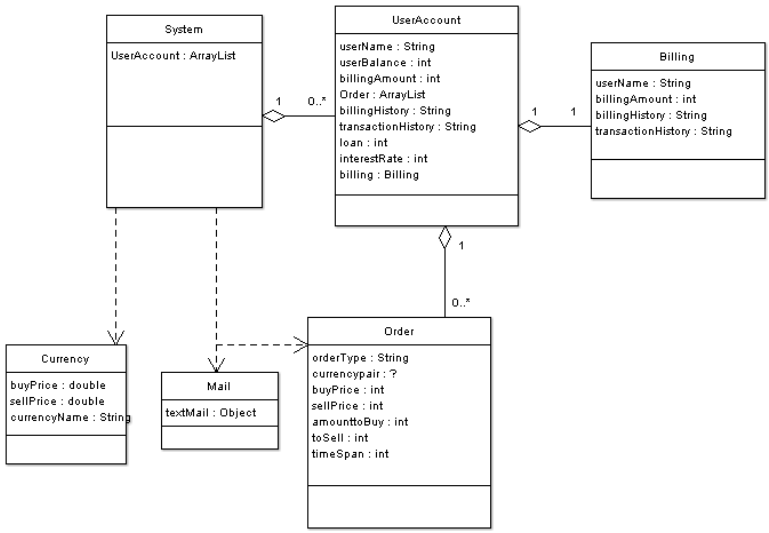

Figure 4: System Classes

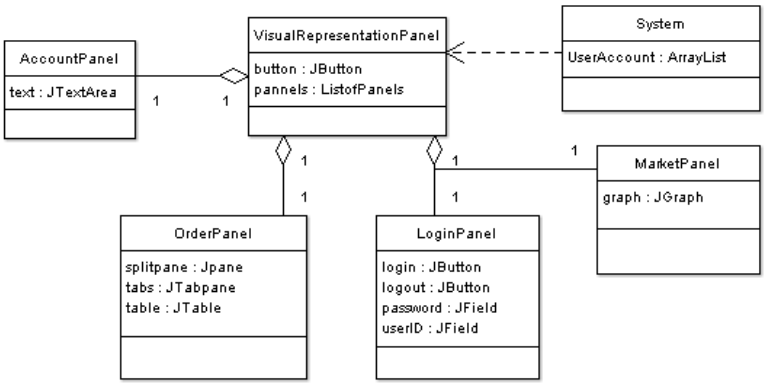

Figure 5: GUI classes