****

**کارفرما:**

**شرکت مديريت فناوري بورس تهران**

**مستندات پروژه شبيه ساز بورس**

****

**پيمانکار:**

**شرکت سرزمين ايده‌هاي فن‌آوري اطلاعات**

**فهرست:**

گزارش پيکر بندي ............................................................................................................................................. 3

گزارش طراحي ................................................................................................................................................... 5

گزارش پياده سازي ............................................................................................................................................ 7

راهنماي کاربري ................................................................................................................................................ 10

راهنماي نصب ................................................................................................................................................... 15

**گزارش پيکربندي**

نرم‌افزار شبيه ساز معاملات بورس با هدف شبيه‌سازي عمليات هسته اصلي نرم‌افزار معاملات بورس و با رويکرد رسيدن به بالاترين حد سرعت ممکن براي پردازش سفارش‌ها و انجام معاملات در بازه‌ي زماني مشخص طراحي و پياده‌سازي شده. با توجه به آزمايش‌هاي صورت گرفته، اين نرم‌افزار براي کارکردن به مدت ۱ ساعت و پردازش حداکثر صدهزار سفارش در ثانيه در اين مدت زمان نياز به سروري با حداقل ۱۶ گيگابايت رم و دو پردازنده‌ي چهار هسته‌اي دارد، و از آنجايي که ثبت تمام اتفاقاتي (ايجاد سفارش‌ها و انجام معاملات) که در سيستم مي‌افتد يک نياز اساسي محسوب مي‌شود، سرعت و کش بيشتر ديسک‌هاي سخت يک مزيت مهم تلقي مي‌شود. باقي مشخصات و امکانات سخت‌افزاري تاثير چنداني در عملکرد سيستم نخواهد داشت. از لحاظ سخت‌افزار تنظيم خاصي علاوه بر تنظيمات معمول (مثل raid و …) نياز نيست.

اين نرم‌افزار بر روي همه‌ي نسخه‌هاي سيستم‌عامل لينوکس قابل نصب و راه‌اندازي مي‌باشد. براي اجراي اين نرم‌افزار به Oracle JDK نسخه‌ي ۷ نياز است و براي عمليات کامپايل و نصب نيز از ابزار maven مي‌توان بهره گرفت و هيچ نرم‌افزار ميان‌افزار جانبي ديگري نياز نيست.

نرم‌افزار نصب شده در شرکت مديريت فناوري سازمان بورس روي سيستم عامل لينوکس نسخه‌ي Ubuntu 12.04 نصب شده و JDK نيز در مسير

/opt/jdk1.7.0\_03

نصب شده است. نرم‌افزار شبيه ساز معاملات بورس براي رابط کاربري تحت وب خود از وب سرور tomcat نسخه‌ي ۷ به شکل embedded استفاده مي‌کند و نيازي به نصب و پيکر‌بندي وب سرور جداگانه ندارد.

کتابخانه‌هايي که در اين نرم‌افزار بکار رفته‌اند به شرح زير مي‌باشند:

log4j - 1.2.9

commons-lang - 2.6

commons-cli - 1.2

commons-io - 2.1

httpcomponents - 4.1.3

springframework - 3.1.1.RELEASE

commons-discovery - 0.5

tomcat-catalina - 7.0.16

tomcat-embed-core - 7.0.16

tomcat-jasper - 7.0.16

ecj - 3.7.2

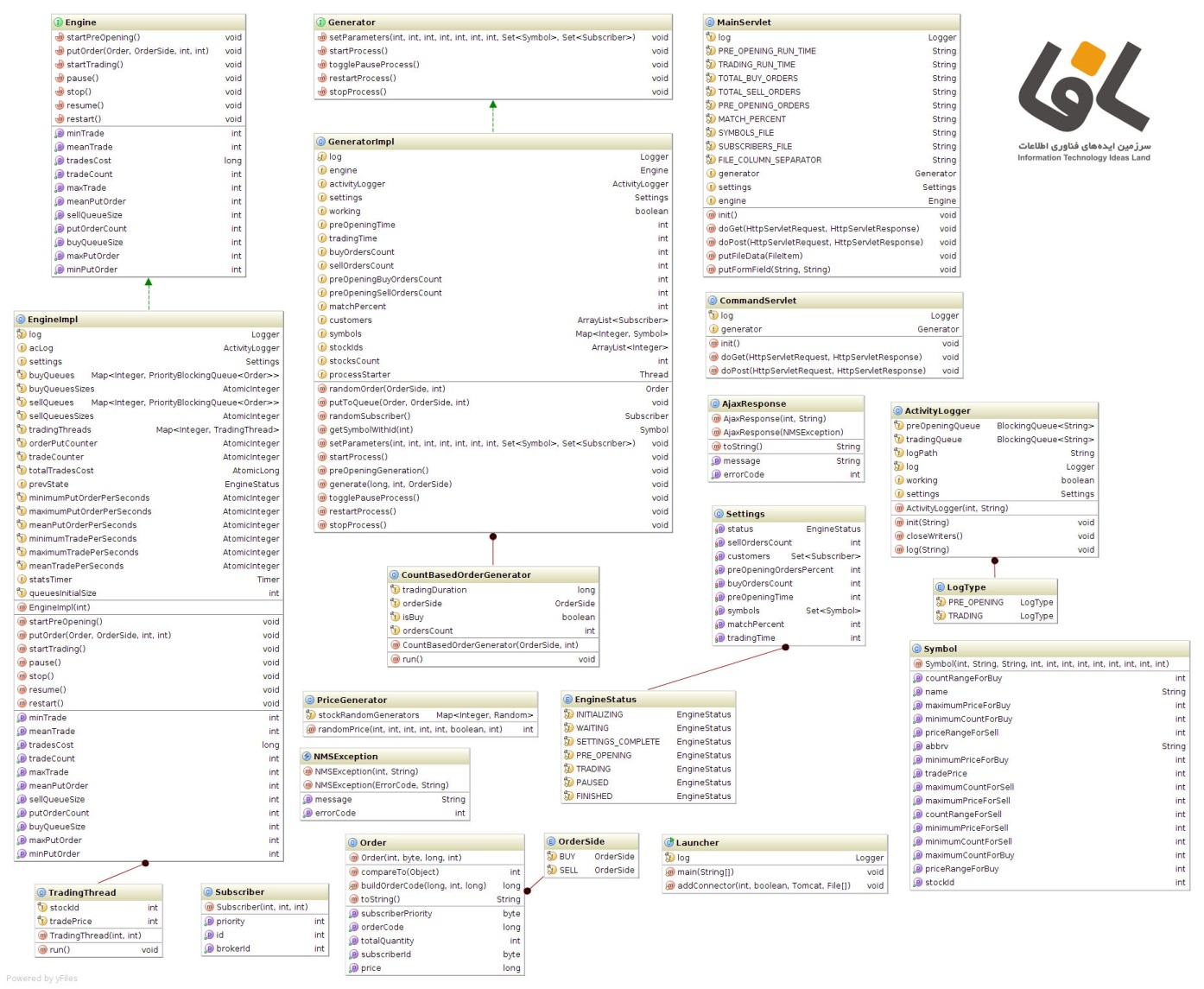
jstl - 1.2

google.guava - 12.0

commons-collections - 3.2.1

**گزارش طراحي**

در طراحي و پياده سازي نرم‌افزار شبيه‌ساز معاملات بورس از متدولوژي اسکرام ([http://fa.wikipedia.org/wiki/%D8%A7%D8%B3%DA%A9%D8%B1%D8%A7%D9%85](http://fa.wikipedia.org/wiki/اسکرام)) استفاده شده و شماي کلي کلاس‌هاي اين نرم‌افزار به اين شکل است:



اين نرم‌افزار داراي ۶ ماژول اصلي مي‌باشد که با قرار گرفتن در package هاي متفاوت از هم مجزا شده‌اند:

1. Web Interface: اين ماژول يک ماژول وب است و حاوي servlet‌ها، فايل‌هاي jsp و ديگر موجوديت‌هاي مربوط به رابط کاربري تحت وب مي‌باشد.
2. Entity: موجوديت‌هايي که اين نرم‌افزار با آنها کار مي‌کند در اين ماژول قرار گرفته‌اند. شامل ۴ کلاس Order، Settings ، Symbol و Subscriber مي‌باشد.
3. Common: شامل يک کلاس به نام NMSException است که براي مديريت خطاهاي داخلي سيستم بين ماژول‌هاي مختلف به کار مي‌رود.
4. Generator: که وظيفه‌ي توليد سفارش‌ها را بر عهده دارد. اين ماژول از يک اينترفيس به نام Generator و دو کلاس به نام‌هاي GeneratorImpl که پياده‌سازي اينترفيس Generator را در بر گرفته و PriceGenerator که توليد قيمت براي سفارش‌ها را با توجه به تنظيم‌هاي صورت گرفته براي رسيدن به درصد انجام معاملات مورد نظر را بر عهده دارد، تشکيل شده است. از کلاس GeneratorImpl يک شي به نام generator ساخته شده و در spring context قرار مي‌گيرد. ماژول‌ web interface با در اختيار گرفتن اين شي دستورات لازم را جهت ثبت تنظيمات و ارسال فرامين با فراخواني متد‌هاي آن انجام مي‌دهد.
5. Engine: که هسته‌ي اصلي سيستم است از يک اينترفيس به نام Engine‌ و کلاس پياده‌سازي آن به نام EngineImpl تشکيل شده است. از اين ماژول نيز يک شي به نام engine که از کلاس EngineImpl ساخته شده در spring context موجود است.
6. Log: داراي يک کلاس به نام ActivityLogger‌ مي‌باشد و وظيفه‌ي ثبت event ها (شامل ايجاد سفارش و انجام معاملات) را بر عهده دارد. از اين کلاس نيز يک شي در spring context موجود است که توسط engine به کار گرفته مي‌شود.

**گزارش پياده‌سازي**

پياده‌سازي نرم‌افزار شبيه‌ساز معاملات بورس پس از بحث و بررسي با کارشناسان شرکت مديريت فن‌آوري بورس تهران و توافق بر سر الگوريتم‌هاي توليد سفارش‌ها، نهوه‌ي صف بندي آن‌ها و انجام معاملات شروع شد. نحوه‌ي ماژول بندي اين نرم‌افزار و وظايف هر کدام از ماژول‌ها در مستند طراحي به تفضيل شرح داده شده است. در اين مستند مراحلي که توسط هر کدام از اين ماژول‌ها و توابع موجود در آنها براي نيل به هدف شبيه‌سازي صورت مي‌گيرد را شرح مي‌دهيم.

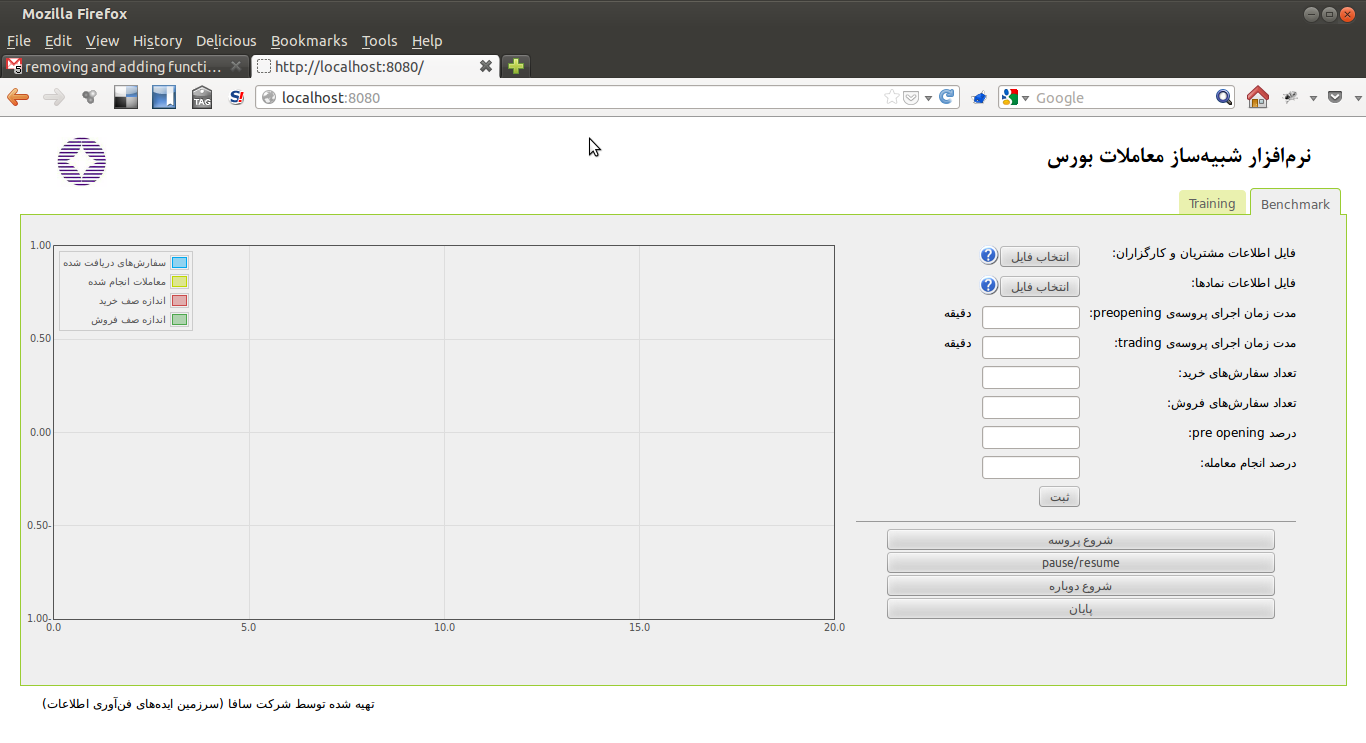
MainServlet در ماژول web، نقطه‌ي ارتباط با کاربر و Servlet اصلي نرم‌افزار است که با فراخواني آن به صفحه‌ي اصلي نرم‌افزار هدايت مي‌شويم. با ثبت تنظيم‌ها و معرفي فايل‌هاي داده‌ي مورد نياز اين Servlet تنظيمات مورد نياز براي کار نرم‌افزار را روي شي‌اي که از کلاس Settings ساخته شده و در spring context موجود است انجام مي‌دهد. همينطور اين Servlet وظيفه‌ي دريافت اطلاعات مربوط به گزارش‌ها و نموداري که در صفحه‌ي اصلي کشيده مي‌شود را از engine بر عهده دارد. Servlet ديگري که در اين ماژول موجود است CommandServlet‌ مي‌باشد که براي دريافت فرامين و انتقال آنها به Generator از آن بهره مي‌گيريم. اين Servlet چهار فرمان شروع پروسه (start)، پايان پروسه (stop)، وقفه/ادامه (pause) و شروع مجدد (restart) را به Generator منتقل مي‌کند و نتايج اين فرمان‌ها را براي نمايش باز مي‌گرداند. صفحه‌ي اصلي رابط کاربري نرم‌افزار به نام index.jsp در مسير main/webapp از زيرشاخه‌ي src پروژه موجود است. به منظور ساده‌سازي و جلوگيري از load شدن‌هاي چند باره‌ي اين صفحه، همه‌ي فعل و انفعالات اين صفحه با Servlet ها توسط Javascript و با استفاده از تکنولوژي Ajax صورت مي‌پذيرد. بنابراين در حين کار با رابط کاربري اين نرم‌افزار ما صرفا با همين صفحه روبرو خواهيم بود. همينطور timer هاي موجود در اين صفحه متناوبا وضعيت engine را چک مي‌کنند و در هر لحظه بنا به وضعيت سيستم عمليات مختلف مورد نياز (کشيدن نمودار، نمايش گزارش و ...) را انجام مي‌دهند.

کلاس GeneratorImpl در ماژول Generator که کلاس اصلي اين ماژول محسوب مي‌شود داراي يک Thread‌ اصلي براي انجام مراحل مختلف پروسه مي‌باشد و در هر مرحله بنا به نياز Thread هاي ديگري را ايجاد کرده و وظايف مختلف را به آنها محول مي‌نمايد. نحوه‌ي کار Thread اصلي در اين ماژول که با فراخواني متد startProcess ساخته مي‌شود به اين شکل است که ابتدا Logger سيستم و سپس engine را از شروع پروسه با خبر مي‌سازد تا مراحل اوليه در هر کدام انجام پذيرد. سپس دو Thread ايجاد مي‌‌کند که وظيفه‌ي يکي توليد سفارش‌هاي خريد و ديگري توليد سفارش‌هاي فروش مي‌باشد، و اين Thread ها را اجرا مي‌کند. پس از آن به مدت تنظيم شده براي پروسه‌ي Pre-opening منتظر مي‌ماند. اگر در طول اين مدت فرمان پايان از رابط کاربري برسد Thread اصلي interrupt مي‌شود. Thread اصلي پس از هر مرحله وضعيت سيستم را که در شي Settings ثبت مي‌شود چک مي‌کند و بنا به وضعيت ثبت شده عکس‌العمل نشان مي‌دهد. پس از انتظار براي پايان يافتن پروسه‌ي Pre-opening ماژول Generator فرمان شروع پروسه‌ي Trading را به engine‌ ارسال مي‌کند و خود دوباره دو Thread ديگر براي توليد سفارش ها ايجاد کرده و آنها را اجرا مي‌نمايد و دوباره تا پايان يافتن مدت زمان Trading منتظر مي‌ماند. پس از پايان يافتن مدت زمان trading ماژول Generator فرمان پايان يافتن پروسه را به engine ارسال مي‌کند.

کلاس EngineImpl که پياده‌سازي engine‌ توسط آن صورت گرفته حاوي دو نوع از متد‌هاست. گروهي از اين متد‌ها براي گرفتن اطلاعات داخلي engine توسط رابط کاربري به کار مي‌روند که نام همه‌ي آنها با get‌ شروع مي‌شود. باقي متد‌ها براي ارتباط با Generator به کار مي‌روند و فرامين را از او دريافت مي‌کنند. Engine داراري دو Map به نام‌هاي buyQueues و sellQueues مي‌باشد که اين دو Map براي هر نماد صف‌هاي خريد و فروش را نگهداري مي‌کنند. همينطور map ديگري با نام tradingThreads وجود دارد که به ازاي هر نماد يک Thread براي مقايسه‌ي سفارش‌ها و انجام معاملات در آن نگهداري مي‌شود. همينطوري engine داراي شمارنده‌‌هاي مختلفي نيز هست که مقادير مختلف مورد نياز براي گزارش‌ها را در آنها نگهداري مي‌کند. Engine با استفاده از متد putOrder سفارش‌ها را دريافت مي‌کند و آنها را در صف مربوط به خود قرار مي‌دهد. اين صف‌ها از نوع sorted مي‌باشند که هر شي‌اي که در آنها قرار مي‌گيرد با توجه به متد compareTo که از کلاس آن override شده به صورت sort شده در صف قرار خواهد گرفت. متد compareTo از کلاس Order که حاوي اطلاعات سفارش است به شکلي پياده‌سازي شده که صف‌ها با الگوريتم Origin FIFO مرتب شوند. ترتيب sort در دو صف خريد و فروش بر خلاف يکديگر است. به اين معني که مثلا دو سفارش از مشترياني با priority يکسان در صف خريد به شکلي sort مي‌شوند که سفارش با قيمت بالاتر در ابتداي صف قرار گيرد ولي در صف فروش سفارش با قيمت پاينتر در ابتداي صف قرار خواهد گرفت. پياده‌سازي Thread هاي Trading نيز در همين کلاس انجام گرفته. هر Thread در صورتيکه سفارشي در صف‌ها موجود باشد سفارش ابتدايي هر دو صف خريد و فروش را برداشته و باهم مقايسه مي‌کند. در صورتيکه اين دو سفارش منجر به معامله شدند تعداد آنها را از هم کم مي‌کند و اگر از سفارشي تعدادي باقي ماند آنرا به صف باز مي‌گرداند. در صورتيکه با سفارش‌هاي ابتداي هر دو صف امکان انجام معامله وجود نداشته باشد، سفارش‌ها به صف بازگردانده مي‌شوند و Thread منتظر مي‌ماند تا سفارش جديدي به صف وارد شود. متد PutOrder وارد شدن سفارش جديد را به اطلاع Thread مي‌رساند. با هر سفارش جديدي که وارد صف شود اين پروسه تکرار مي‌شود.

**راهنماي کاربري**

رابط کاربري نرم‌افزار شبيه ساز معاملات بورس روي آدرس http://localhost:8080 قابل مشاهده است. این نرم افزار در 2 مد آموزشی و benchmark قابلیت کارکرد دارد.



در مد benchmark براي شروع پروسه ابتدا بايد فايل‌هاي حاوي داده‌هاي مشتريان و نماد‌ها بارگزاري شوند. اين فايل‌ها با فرمت csv بايد ايجاد شده باشند. نمونه‌ي اين فايل‌ها در پوشه‌ي sample-data از مسير اصلي پروژه موجود است. همچنين با کليک بر روي علامت سئوال کنار دکمه‌ي انتخاب فايل توضيحات مورد نياز در مورد شکل فايل‌ها و معني هر ستون در هر کدام از فايل‌ها قابل مشاهده است. شرح این فایل ها به این شرح است:

فایل اطلاعات مشتریان و کارگزاران:

فایل اطلاعات مشتریان و کارگزاران فایلی به فرمت csv است که دارای سه ستون می باشد.

این سه ستون که با استفاده از کاما "," به عنوان جداکننده از هم تفکیک شده اند حاوی اطلاعات زیر می باشند:

مشخصه ی مشتری (عددی)

مشخصه ی کارگزار (عددی)

اولویت مشتری (عددی)

فایل اطلاعات نماد ها:

فایل اطلاعات نماد ها فایلی به فرمت csv است که دارای دوازده ستون می باشد.

این دوازده ستون که با استفاده از کاما "," به عنوان جداکننده از هم تفکیک شده اند حاوی اطلاعات زیر می باشند:

مشخصه نماد (عددی)

نام نماد (متنی)

علامت اختصاری نماد (متنی)

حداقل قیمت برای سفارش های خرید (عددی)

حداکثر قیمت برای سفارش های خرید (عددی)

حداقل قیمت برای سفارش های فروش (عددی)

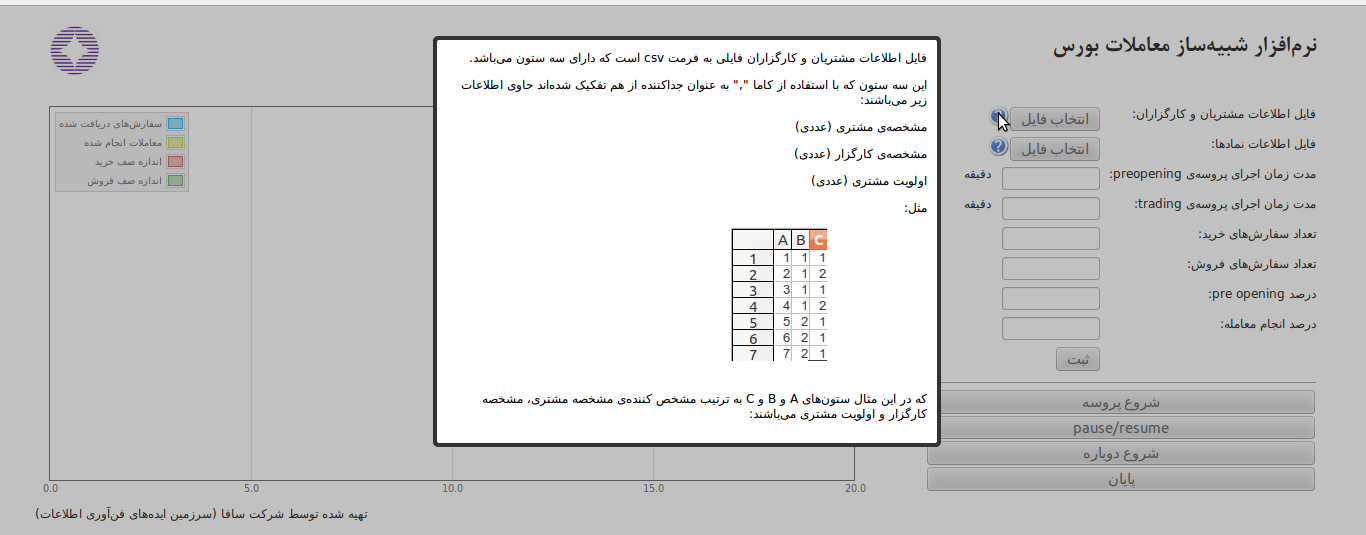
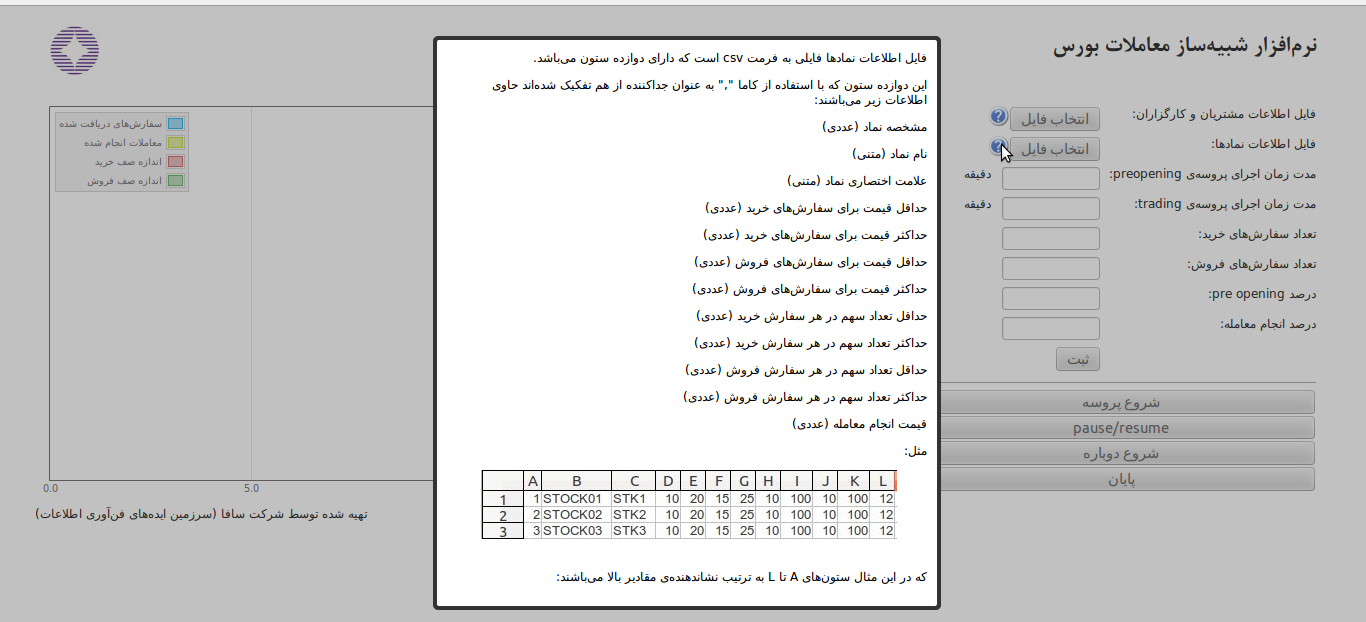
حداکثر قیمت برای سفارش های فروش (عددی)

حداقل تعداد سهم در هر سفارش خرید (عددی)

حداکثر تعداد سهم در هر سفارش خرید (عددی)

حداقل تعداد سهم در هر سفارش فروش (عددی)

حداکثر تعداد سهم در هر سفارش فروش (عددی)

قیمت انجام معامله (عددی)

پس از بارگزاري فايل‌ها و وارد کردن تنظيم‌هاي مورد نياز با کليک بر روي دکمه‌ي ثبت تنظيمات را براي اجراي پروسه ثبت می‌کنيم.

در موارد زير پيغام خطايي مشاهده خواهد شد:

زمان اجراي پروسه pre-opening به علاوه زمان اجراي پروسه‌ي trading‌ بيشتر از يک ساعت باشد.

تعداد سفارش‌هاي توليد شده در زمان تنظيم شده براي پروسه‌ها به throughput بيشتر از صد‌ هزار سفارش در ثانيه منجر شود.

درصد pre-opening بيشتر از ۵۰ باشد.

درصد انجام معامله کمتر از ۳۰ باشد.

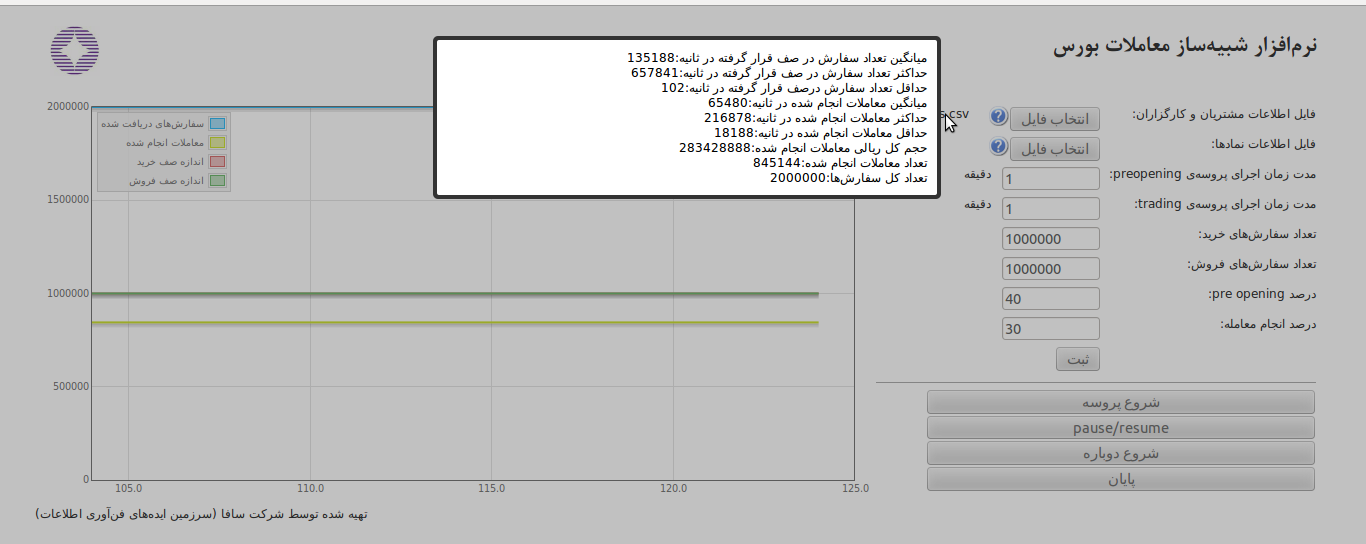
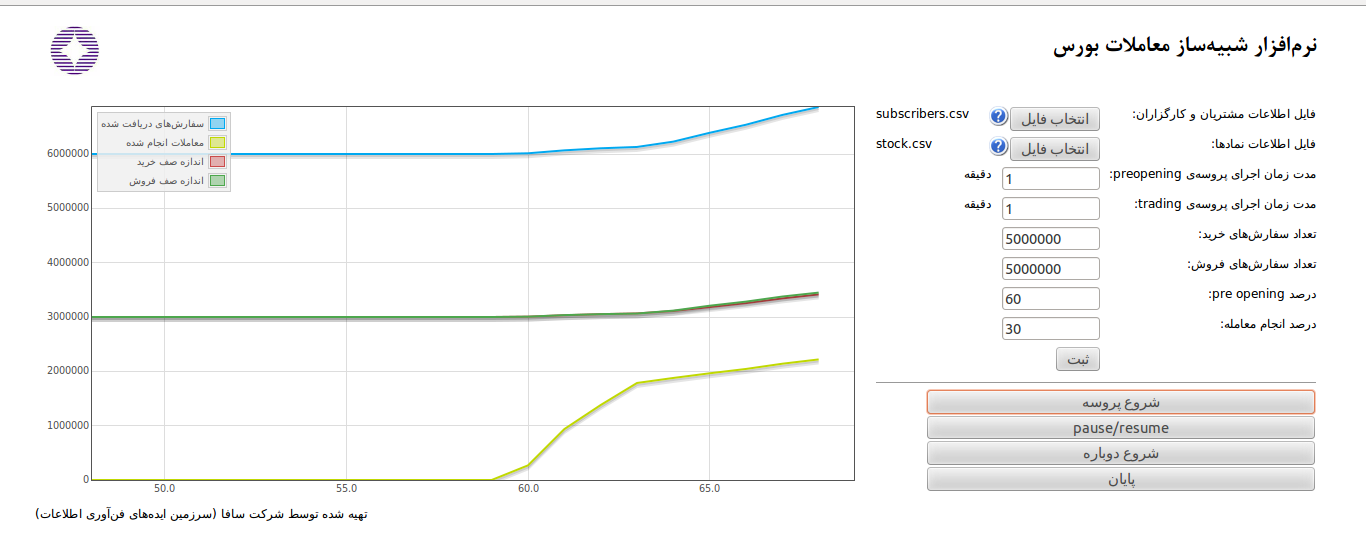
براي شروع اجراي پروسه بر روي دکمه‌ي "شروع پروسه" کليک می‌کنيم. با شروع پروسه و توليد سفارش‌ها و قرار گرفتن آنها در صف نمودار سمت چپ مقادير زير را نمايش خواهد داد:

تعداد کل سفارش‌هاي توليد شده

تعداد معاملات انجام گرفته

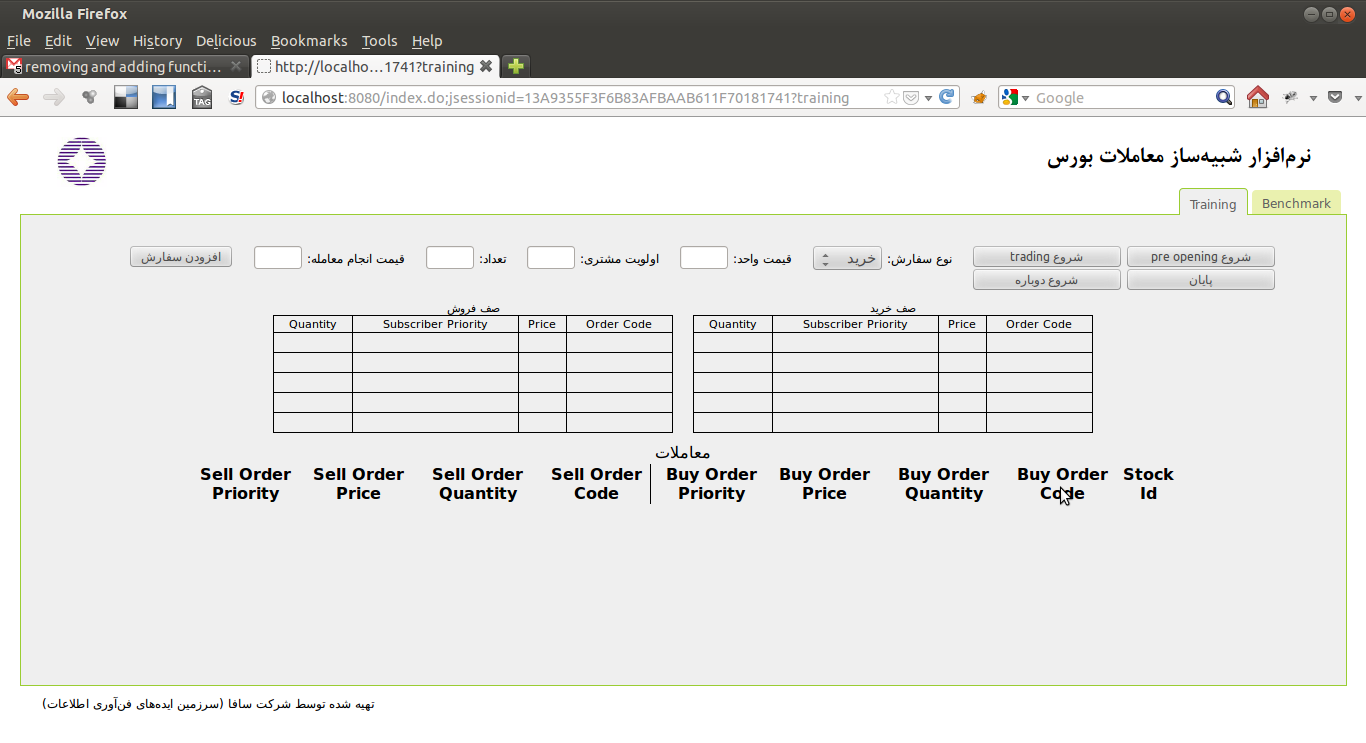
تعداد سفارش‌هاي موجود در صف خريد

تعداد سفارش‌هاي موجود در صف فروش

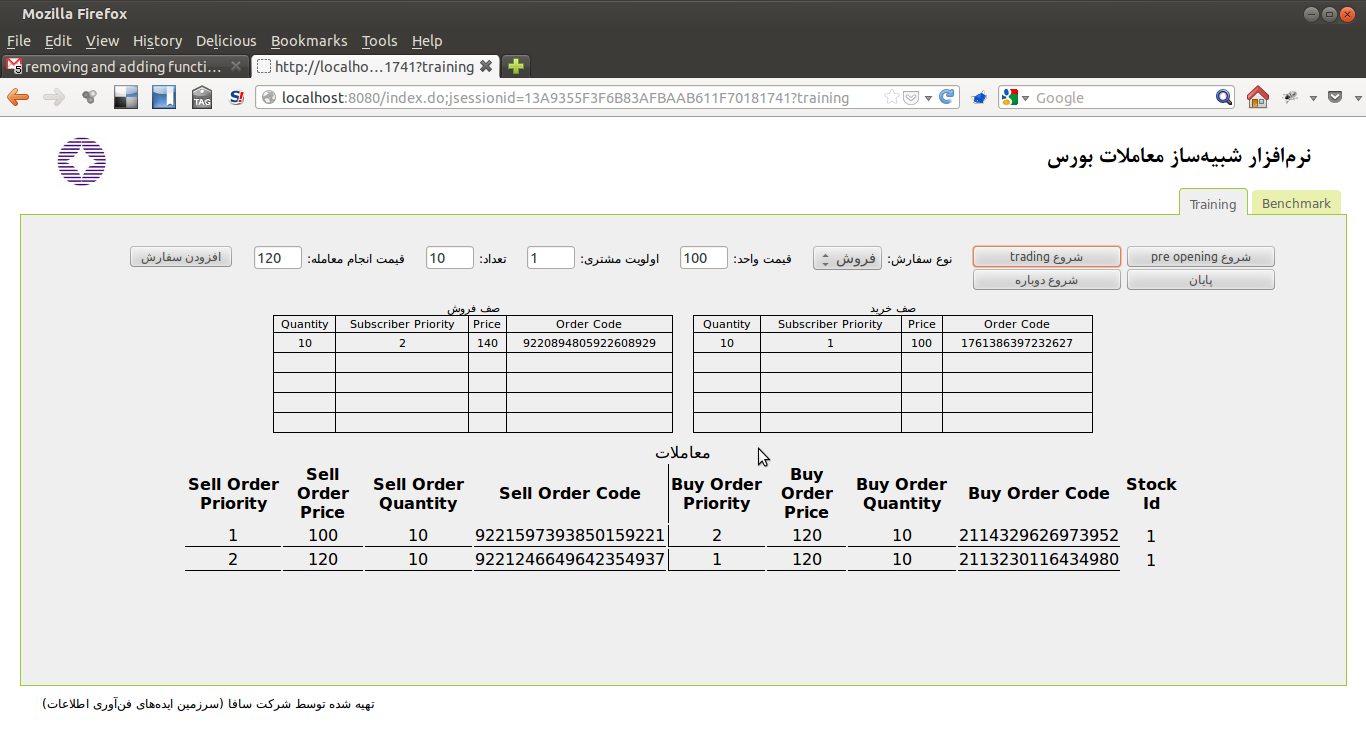
در هر لحظه از اجراي پروسه مي‌توان با کليک بر روي دکمه‌ي پايان به کار نرم‌افزار خاتمه داد. با پايان دادن به پروسه يا اتمام زمان و پايان يافتن پروسه گزارش نهايي از اجراي پروسه نمايش داده خواهد شد که امکان دانلود اطلاعات نمايش داده شده در يک فايل csv با کليک بر روي لينک مربوطه در گزارش امکان‌پذير است:

پس از پايان پروسه با کليک بر روي شروع دوباره مي‌توان دوباره نسبت به ثبت تنظيمات جديد و اجراي پروسه‌ي ديگر اقدام نمود.

در مد training ایجاد سفارش ها به عهده ی خود کاربر است. در این مد نیز قبل از ایجاد سفارش ابتدا باید پروسه ی pre-opening را شروع کنیم.



پس از شروع پروسه ی pre-opening می توان سفارش ها را به صف اضافه نمود. با شروع پروسه ی trading معاملات نیز انجام می گیرد و در جدولی زیر جداول صف ها مشخصات معاملات انجام شده نشان داده می شود. با ایجاد و اضافه نمودن هر سفارش order code مربوط به سفارش ایجاد شده نمایش داده می شود. سفارش هایی که باقی مانده ی یک معامله ی انجام گرفته هستند با order code قرمز رنگ در جدول نشان داده می شوند.



**پيغام‌هاي خطا:**

بعد از هر فرمان پيغام بازگشتی از آن نمايش داده مي‌شود و در صورت عدم موفقيت پيغام تفضيلي آن (strack trace) در لاگ فايل system.log که مسير آن در فايل log4j.properties مشخص شده نوشته مي‌شود. اين پيغام‌ها به اين شرح مي‌باشند:

0 , SUCCESS : عمليات با موفقيت به انجام رسيده.

FILE\_UPLOAD\_EXCEPTION , -2: اشکال آپلود فايل

INVALID\_SYMBOLS\_FILE , -3: اشکال در فايل اطلاعات نماد‌ها

INVALID\_SUBSCRIBERS\_FILE , -4: اشکال در فايل اطلاعات مشتريان

5- , SETTINGS\_ERROR: اشکال در يکي از مقادير settings

INTERNAL\_SERVER\_ERROR , -6: خطاي دروني سيستم

**راهنماي نصب نرم‌افزار شبيه ساز معاملات بورس**

**پيش‌نياز‌هاي نصب:**

براي نصب و اجراي اين نرم‌افزار به Oracle JDK نسخه‌ي ۷ و ابزارMaven نسخه‌ي ۲ نياز داريم.

Oracle JDK با استفاده از اين آدرس قابل دانلود است:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

و براي دريافت و نصب Maven‌ نيز از اين آدرس مي‌توان استفاده نمود:

<http://www.apache.org/dyn/closer.cgi/maven/binaries/apache-maven-2.2.1-bin.tar.gz>

**تنظيمات:**

قبل اقدام به کامپايل مي‌توان تنظيمات موجود در فايل bnmstest.properties را تغيير داد. اين فايل در اين مسير قرار دارد و تنظيمات قابل تغيير شامل موارد زير مي‌شود:

/path/to/bnmstest/ + src/main/resources/bnmstest.properties

* tomcat-http-port: پورتي که webserver رابط کاربري نرم‌افزار روي آن منتظر request خواهد بود. (پيش‌فرض: 8080)
* tomcat-ajp-port: پورت داخلي رابط کاربري (پيش‌فرض: 8009)
* queues\_initial\_size: اندازه‌ي اوليه براي صف‌ها (وابسته به اندازه حافظه، 3.000.000 براي ۴ گيگ حافظه)
* activity\_logger\_queue\_size: اندازه‌ي اوليه براي صف logger (مقدار پيش‌فرض: 30.000.000)
* activity\_log\_path: مسير نوشتن logfile ها

فايل log4j.properties نيز در همان مسير بالا براي تنظيم لاگ‌هاي سيستمي نرم‌افزار مي‌باشد، که مي‌توان مقاديري مثل مسير لاگ فايل و سطح لاگ‌ها را تعيين نمود.

**کامپايل:**

با تغيير مسير به پوشه‌ي اصلي پروژه (در اين مستند همه جا با /path/to/bnmstest نشان داده شده) و اجراي دستور mvn clean install مرحله‌ي کامپايل و ايجاد فايل‌هاي مورد نياز براي اجرا انجام خواهد شد.

cd /path/to/bnmstest

mvn clean install

* در اين مرحله ممکن است با پيغام خطاي maven مبني بر در دسترس نبودن کتابخانه‌هاي استفاده شده براي دانلود مواجه شويم. در اين صورت از ميان تمام کتابخانه‌هاي استفاده شده در اين پروژه که در پوشه‌ي lib قرار داده شده، کتابخانه‌اي که maven براي دريافت آن با مشکل برخورده را به روش دستي نصب مي‌کنيم. براي مثال:

mvn install:install-file -Dfile=gxp-0.2.4-beta.jar -DgroupId=com.google -DartifactId=gxp -Dversion=0.2.4-BETA -Dpackaging=jar

پس از اجراي موفقيت‌آميز دستور maven، پوشه‌يtarget در همان مسير پروژه ساخته خواهد شد که حاوي فايل bnmstest-1.0.war خواهد بود.

**اجرا:**

براي اجراي نرم‌افزار کافيست فايل run.sh که در پوشه‌ي اصلي پروژه وجود دارد را اجرا کنيم.

cd /path/to/bnmstest

./run.sh &

پس از اجراي نرم‌افزار رابط کاربري از طريق آدرس [http://localhost:8080](http://localhost:8080/) در دسترس خواهد بود.