

Chương 9: Khởi công xây dựng app đấu giá online - Auction Sniper

Để bắt đầu từ đầu:

Chúng tôi khởi công xây dựng một ứng dụng tự động đặt giá (bid) trong các cuộc đấu giá (auction).

Chúng tôi sẽ thảo luận về việc nó cần hoạt động như thế nào và nên có các thành phần chính nào. Chúng tôi sắp xếp một kế hoạch thô sơ về các bước tăng dần mà theo đó chúng tôi sẽ phát triển dần ứng dụng.

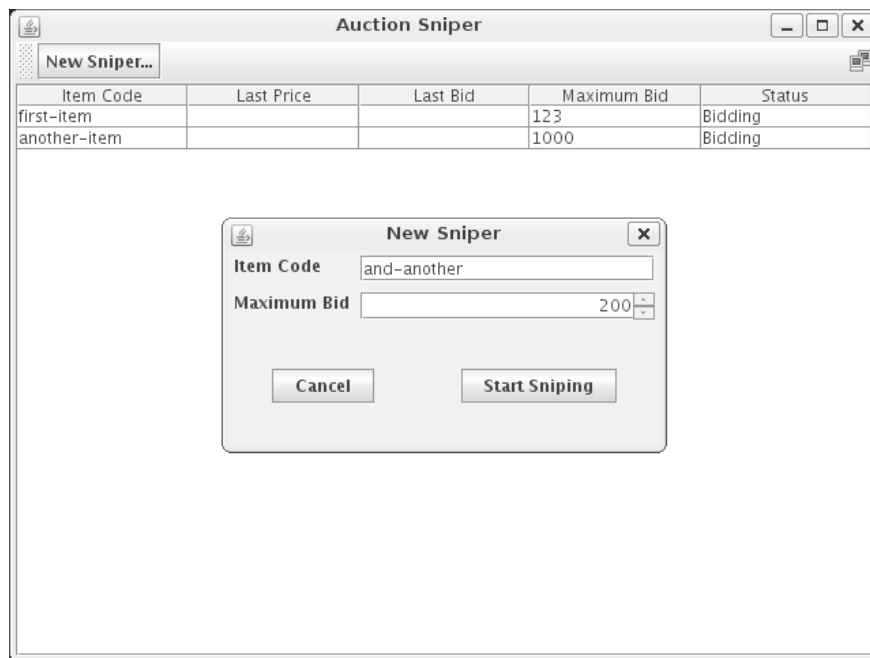
Chúng tôi là một đội phát triển đang làm việc cho Markup and Gouge, một công ty mua đồ cổ từ thị trường chuyên nghiệp để bán cho khách hàng "với giá tốt nhất có thể". Markup and Gouge hiện đang thực hiện khá nhiều hoạt động mua bán trực tuyến, chủ yếu từ hãng Southabee's - một nhà đấu giá rất muốn phát triển online. Vấn đề là những nhân viên mua đang tốn nhiều thời gian kiểm tra trạng thái của cuộc đấu giá một cách thủ công để quyết định xem có đặt giá hay không, họ thậm chí bỏ lỡ một vài đồ vật đáng quan tâm chỉ vì họ không thể phản ứng đủ nhanh. Sau nhiều thảo luận, bên quản lí quyết định khởi công Auction Sniper, một ứng dụng theo dõi các cuộc đấu giá online và tự động đặt giá hơi cao hơn một chút mỗi khi giá thay đổi, cho đến khi nó đạt đến một ngưỡng dừng (stop-price) hoặc khi giá bán cuối cùng đã được xác định (auction closes). Nhân viên mua rất muốn có ứng dụng này và một số đồng ý giúp chúng tôi xác định rõ cần phải xây dựng cái gì.

Chúng tôi bắt đầu bằng việc thảo luận sơ về các ý tưởng của hãng với nhóm nhân viên mua, và thấy rằng, để tránh hiểu nhầm, chúng tôi cần thống nhất về một số điểm căn bản:

- *Item* là cái gì đó có thể định danh và mua
- *Bidder* là một người hoặc một tổ chức có muốn mua một item
- *Bid* là một tuyên bố rằng một bidder sẽ trả một giá cho trước để mua một item
- *Current price* - giá hiện hành - là bid cao nhất hiện tại cho item
- *Stop-price* - ngưỡng dừng - là giá cao nhất mà một bidder sẵn sàng trả để mua một item
- *Auction* - cuộc đấu giá - là một tiến trình quản lý các bid cho một item.
- *Auction house* - nhà đấu giá - là một tổ chức chuyên đứng ra tổ chức các cuộc đấu giá.

Các cuộc thảo luận tạo ra một danh sách dài các requirement, chẳng hạn như có khả năng đặt bid cho các nhóm item có liên quan với nhau. Không thể có chuyện hoàn thành được mọi thứ trong một khoảng thời gian đủ ngắn để có thể có ích, do đó chúng tôi bàn một lượt qua các lựa chọn và những nhân viên mua đành phải đồng ý rằng trước nhất họ muốn có một ứng dụng cơ bản chạy được cái đã. Một khi có nó rồi, ta có thể làm cho nó nhiều chức năng hơn. Hóa ra là trong hệ thống online, có một cuộc đấu giá cho mỗi item, do đó chúng tôi quyết định dùng định danh của mỗi item để chỉ đến auction của nó. Thực tế cho thấy, ứng dụng Sniper không phải lo quản lý những item đã được bán, vì các hệ thống khác sẽ xử lý việc thanh toán và giao hàng. Chúng tôi quyết

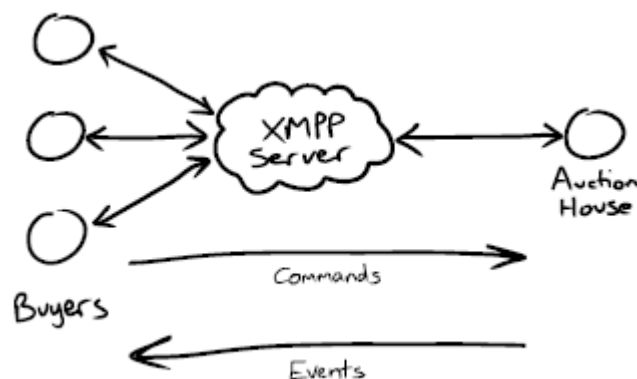
định xây dựng Auction Sniper là một ứng dụng Java Swing. Nó sẽ chạy trên desktop và cho phép người dùng cùng lúc bid nhiều item. Với mỗi item đang nhắm bắn, nó cũng sẽ hiển thị : id, stop-price, current price, và trạng thái của item đó. Nhân viên mua sẽ có thể dùng UI để thêm các item cần nhắm bắn, và các giá trị được hiển thị sẽ thay đổi theo các event đến từ nhà đấu giá. Nhân viên mua vẫn còn đang làm việc với các thành viên của đội usability, nhưng chúng tôi đã thống nhất được một phiên bản thô sơ trông đại loại như trong Hình 9.1.



Hình 9.1. Giao diện người dùng đầu tiên

Rõ ràng là nó xấu và chưa hoàn chỉnh, nhưng nó đã đủ gần đúng để cho chúng tôi bắt tay vào việc.

Trong khi các cuộc thảo luận trên còn đang diễn ra, chúng tôi nói với các kỹ thuật viên tại Southabee's, những người hỗ trợ chúng tôi về các dịch vụ trực tuyến của họ. Họ gửi chúng tôi một tài liệu mô tả giao thức đặt bid trong các cuộc đấu giá, giao thức này sử dụng XMPP (Jabber) cho tầng communication bên dưới. Hình 9.2 mô tả cách nó xử lý nhiều bidder cùng gửi bid qua XMPP cho nhà đấu giá, Sniper của chúng tôi là một trong những bidder đó. Trong khi cuộc đấu giá đang diễn ra, Southabee's sẽ gửi các event đến cho tất cả những bidder đang kết nối để thông báo khi có bid của ai đó vừa nâng current price và khi cuộc đấu giá kết thúc.



Hình 9.2. Hệ thống đấu giá trực tuyến của Southabee's

XMPP: the eXtensible Messaging and Presence Protocol

XMPP là một giao thức truyền các phần tử XML qua mạng. Nó vốn được thiết kế cho hệ thống chat Jabber và có tên Jabber, sau đổi tên thành XMPP khi đăng kí với IETF để được duyệt làm một chuẩn Internet. Vì là một framework tổng quát cho việc trao đổi các phần tử XML qua mạng, nó có thể được dùng cho nhiều ứng dụng đa dạng mà cần trao đổi dữ liệu có cấu trúc trong điều kiện gần như thời gian thực.

XMPP có một kiến trúc phân tán client/server. Không có server trung tâm, ngược lại với các dịch vụ chat khác như AOL Instant Messenger hay MSM Messenger. Ai cũng có thể chạy một server XMPP chứa dữ liệu về các user và cho họ liên lạc với nhau và với các user đặt tại các server XMPP khác trong mạng.

Một user có thể log vào một server XMPP đồng thời từ nhiều thiết bị hay client, thuật ngữ XMPP gọi đó là các resource. Một user gán cho mỗi resource một độ ưu tiên (priority). Trừ khi được đánh địa chỉ tới một resource cụ thể, các message cho một user sẽ được gửi đến resource có độ ưu tiên cao nhất đang được đăng nhập. Mỗi user trong mạng có một ID Jabber phân biệt (thường viết tắt là JID) giống như một địa chỉ email. Một JID chứa một username và một địa chỉ DNS của server mà user đó định cư, tách nhau bởi một dấu @ (chẳng hạn username@example.com), và tùy theo lựa chọn có thể có thêm đuôi là một tên resource sau một dấu chéo (ví dụ username@example.com/office).

Liên lạc với một cuộc đấu giá

Giao thức đấu giá - The Auction Protocol

Giao thức cho các message giữa một bidder và một nhà đấu giá rất đơn giản. Bidder gửi một command, là một trong các loại sau:

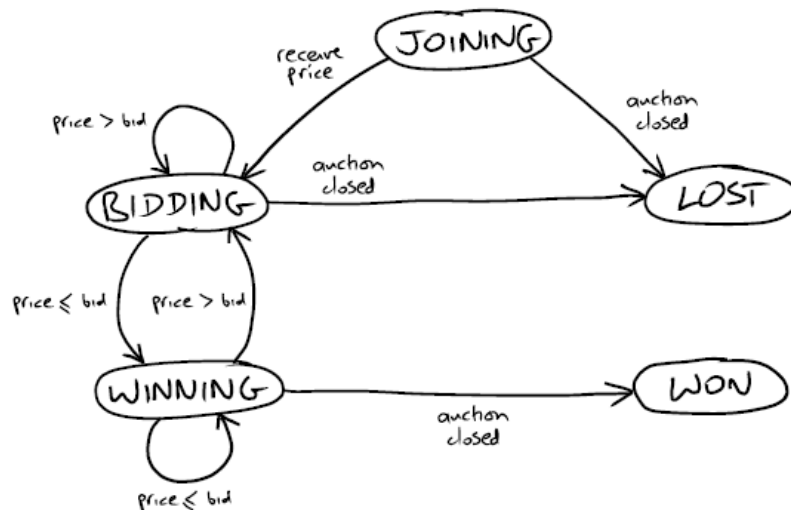
Join: một bidder gia nhập một cuộc đấu giá. Sender của XMPP message xác định bidder, và tên của chat session xác định item.

Bid: một bidder gửi một giá bid tới cuộc đấu giá.

Các cuộc đấu giá (auction) gửi các event, thuộc một trong các loại:

Price: Một cuộc đấu giá thông báo về current price mới. Event này cũng chứa cả mức tăng tối thiểu mà bid tiếp theo cần đạt được, và tên của bidder vừa đặt giá mới đó. Cuộc đấu giá sẽ gửi event này cho một bidder vừa mới gia nhập, và gửi cho tất cả bidder khi có bid mới nhận được.

Close: Một cuộc đấu giá tuyên bố kết thúc. Người đặt giá cuối cùng là người thắng cuộc.



Hình 9.3: sơ đồ trạng thái biểu diễn behavior của một bidder.

Chúng tôi dành thời gian đọc tài liệu và nói chuyện với những người hỗ trợ tại Southabee's On-line, và vẽ được một sơ đồ trạng thái cho thấy một Sniper có thể chuyển trạng thái như thế nào. Đại thể, một Sniper gia nhập một cuộc đấu giá, rồi xảy ra một số vòng đặt giá (bidding), cho đến khi đấu giá kết thúc, tại thời điểm đó Sniper đã hoặc thua hoặc thắng. Xem Hình 9.3. Để đơn giản, chúng tôi đã tạm bỏ qua stop-price; nó sẽ quay lại trong chương 18.

XMPP Message

Southabee's On-Line còn gửi cho chúng tôi chi tiết về các định dạng mà họ sử dụng bên trong các message XMPP. Chúng khá đơn giản, chỉ có một vài cái tên và giá trị, và được serialized thành một dòng gồm các cặp key/value. Mỗi dòng bắt đầu với một số hiệu version của chính giao thức đang sử dụng. Các message trông thế này:

SOLVersion: 1.1; Command: JOIN;

SOLVersion: 1.1; Event: PRICE; CurrentPrice: 192; Increment: 7; Bidder: Someone else;

SOLVersion: 1.1; Command: BID; Price: 199;

SOLVersion: 1.1; Event: CLOSE;

Southabee's On-Line dùng các login name để định danh các item để bán, do đó, để bid cho một item có định danh 12793, một client sẽ mở chat tại server của Southabee's với "user" auction-12793. Server có thể dùng id của người gọi để xác định xem ai đang bid, giả thiết rằng các account đã được thiết lập từ trước.

Đến đích an toàn

Ngay cả một ứng dụng nhỏ như thế này cũng quá lớn để có thể viết một lèo là xong, cho nên chúng tôi cần vạch ra đại khái những bước mà chúng tôi có thể sẽ thực hiện để đạt đến đích. Một kĩ thuật quan trọng đối với phát triển tăng dần là học cách cắt lát chức năng, sao cho mỗi lần có thể xây dựng một ít một. Mỗi lát cần đủ cụ thể và có ý nghĩa để đội phát triển có thể xác định thế nào là hoàn thành, và đủ nhỏ để chỉ tập chung vào một concept và có thể được hoàn thành sau thời gian ngắn. Việc chia công việc ra thành những mẫu nhỏ mà mỗi mẫu đều có ý nghĩa mạch lạc giúp chúng ta quản lý rủi ro trong quá trình phát triển. Chúng tôi thường xuyên nhận được feedback cụ thể về tiến độ đang đạt được, nhờ đó ta có thể điều chỉnh kế hoạch khi đội phát triển phát kiến thêm về domain và công nghệ.

Nhiệm vụ ta cần làm ngay là xây dựng một se-ri các bước phát triển tăng dần cho ứng dụng Sniper. Bước đầu tiên nhất định phải là feature nhỏ nhất mà ta có thể xây dựng, “bộ xương di động” mà ta đã mô tả tại “Đầu tiên, test một bộ xương di động” (tr.32). Ở đây, khung xương sẽ kết nối một đường tối thiểu qua Swing, XMPP, và ứng dụng của ta. Nó chỉ đủ để cho thấy rằng ta có thể lắp các thành phần đó lại với nhau. Mỗi bước tiếp theo bổ sung thêm một mảnh vào ứng dụng sẵn có, và xây dựng tiếp trên cơ sở những gì đã đạt được. Sau một cuộc thảo luận, chúng tôi đã nghĩ ra chuỗi các feature cần xây dựng như sau:

Single item: join, lose without bidding – một item; tham gia và thua mà không đặt bid lần nào
Đây là case khởi đầu, khi ta lắp ghép hạ tầng cơ sở; đó là chủ đề của chương 10.

Single item: join, bid, and lose – một item, tham gia, bit, và thua.
Thêm hành động bid vào chuỗi kết nối cơ bản.

Single item: join, bid, and win – một item, tham gia, bid, và thắng.
Xác nhận người đặt bid thắng cuộc.

Show price details - Hiển thị thông tin chi tiết về giá.
Bắt đầu thêm chi tiết cho giao diện người dùng.

Multiple items – nhiều item.
Hỗ trợ đặt bid cho nhiều item trong cùng một ứng dụng.

Add items through the user interface – Thêm item bằng cách sử dụng UI.
Cài đặt nhập dữ liệu qua UI.

Stop bidding at the stop price – Dừng đặt bid khi đã chạm ngưỡng dừng.
Thêm một chút thông minh vào thuật toán Sniper.

Ở danh sách đó, những nhân viên mua đã sắp thứ tự ưu tiên cho giao diện người dùng lên trên ngưỡng dừng, một phần là vì họ muốn đảm bảo rằng họ sẽ cảm thấy thoải mái với ứng dụng, phần khác vì sẽ khó có thể dễ dàng thêm nhiều item, mỗi item với một ngưỡng dừng riêng, mà không có một UI tử tế.

Một khi danh sách này chạy ổn, ta có thể bắt tay vào những tình huống phức tạp hơn, chẳng hạn như bid lại nếu hỏng, hoặc dùng các chiến lược bidding khác. Ở thời điểm này, cài đặt những feature liệt kê ở trên đã đủ để chúng tôi bận rộn rồi.

To Do

single item - join, lose without bidding

single item - join, bid & lose

single item - join, bid & win

single item - show price details

multiple items

add new items through the GUI

stop bidding at stop price

Hình 9.4. Kế hoạch ban đầu.

Chúng tôi không biết rằng mình sẽ tiến hành đúng theo thứ tự trên hay không. Nhưng chúng tôi tin rằng mình cần qua tất cả các bước đó, và chúng tôi có thể điều chỉnh trong quá trình làm.

Đây không phải chuyện thực.

Đến đây, bạn đọc có thể đang phản đối về tất cả những nhiệm vụ thực tế mà chúng tôi đã bỏ quá. Chúng tôi cũng thấy như vậy. Chúng tôi đã đi tắt ở quy trình và thiết kế để trong phạm vi một cuốn sách có thể mang lại cho bạn đọc cảm nhận về một project thực. Cụ thể:

- Kiến trúc này không thực tế, XMPP không đáng tin cậy hay an toàn, nên nó không phù hợp với các giao dịch. Việc đảm bảo các tiêu chí đó nằm ngoài phạm vi của chúng ta. Có nghĩa là, các kĩ thuật nền tảng mà chúng tôi mô tả vẫn đúng cho bất kì kiến trúc nào ở bên dưới.
- Đây không phải lập kế hoạch kiểu Agile. Chúng tôi đã lướt qua quá trình lập kế hoạch dự án để xây dựng một todo list. Trong dự án thật, chúng tôi thường phải xem xét toàn bộ kế hoạch giao sản phẩm trước đã. Bạn đọc có thể tham khảo các cuốn sách như [Shore07] và [Cohn05] để học thêm về Agile Planning.
- Thiết kế Usability như ở đây không giống thực tế: Việc thiết kế user experience tốt cần nghiên cứu nhiều khía cạnh. Dự án này đủ đơn giản để chúng tôi có thể phác thảo một hình dung về những gì muốn đạt được và sẽ làm.