Sheffield Trading Specifications

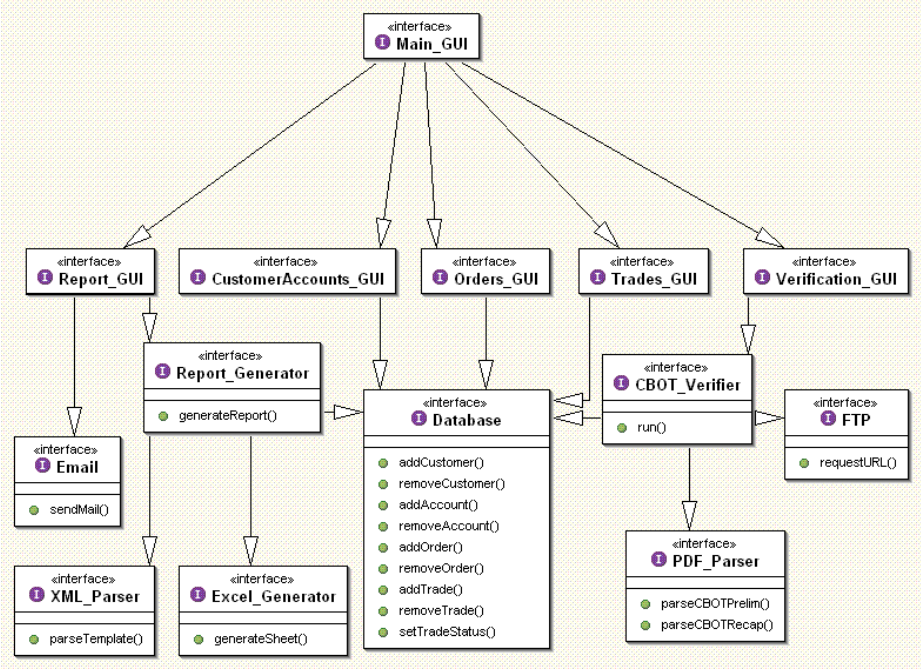
Sam Cates

9 February 2006

1 Description of Project

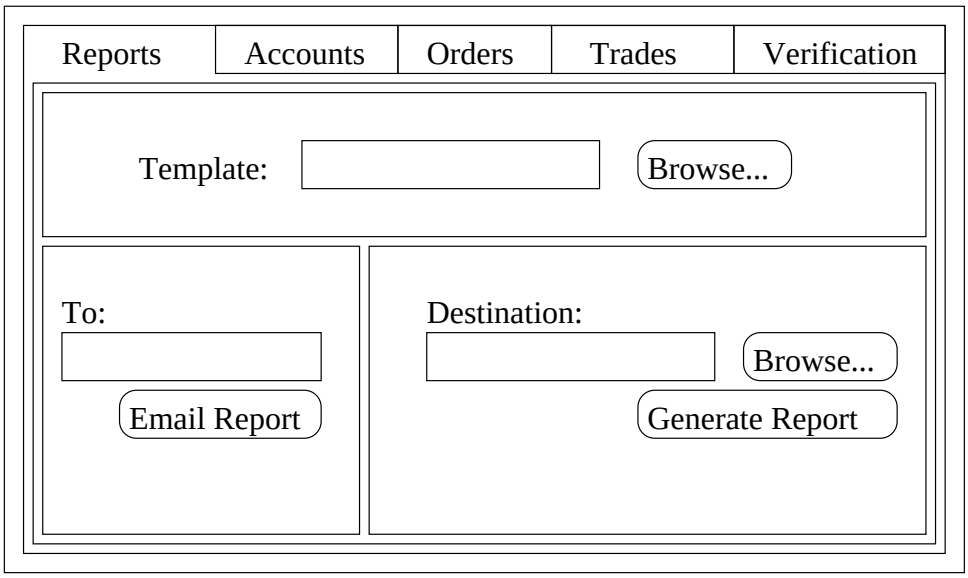
Bucky and Connie are futures brokers. They want software that will keep track of their daily trades faster, easier and more reliably than their current system (which is basically Connie doing it all by hand). They need to be able to enter orders into the system as they get them and then enter trades as they are filling the orders. They also need a way to keep track of a couple levels of verification that the trades were actually executed as they entered them. They also want the system to generate different kinds of reports (daily trade reports, monthly invoices, monthly revenue sheets, etc.).

2 System Model Diagram



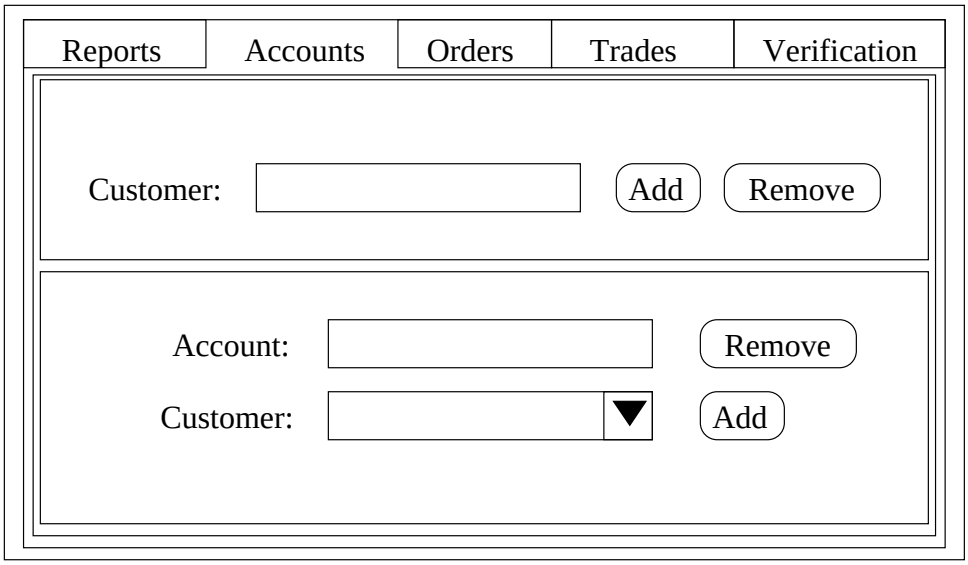
3 User Interface

3.1 Reports



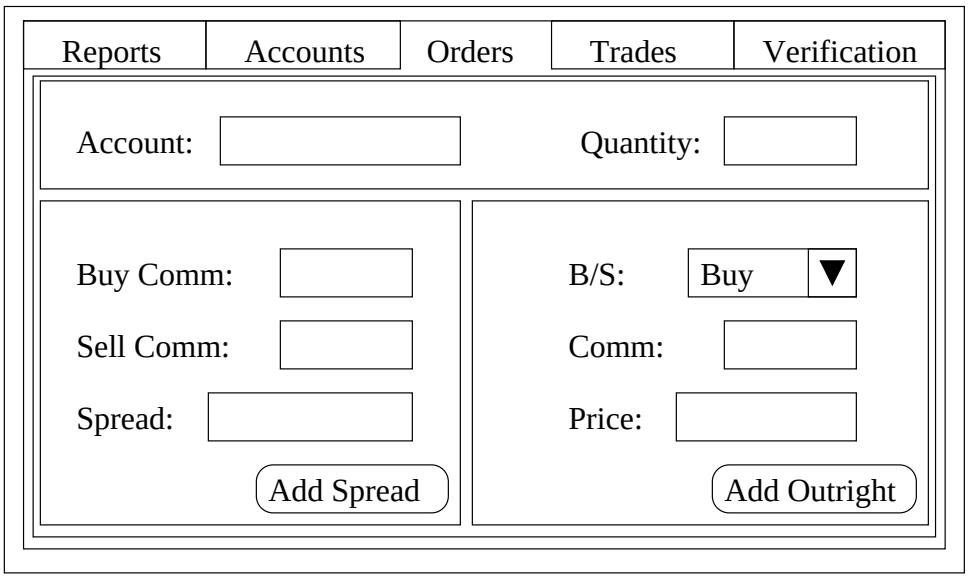
This pane is for generating and emailing reports.

3.2 Accounts



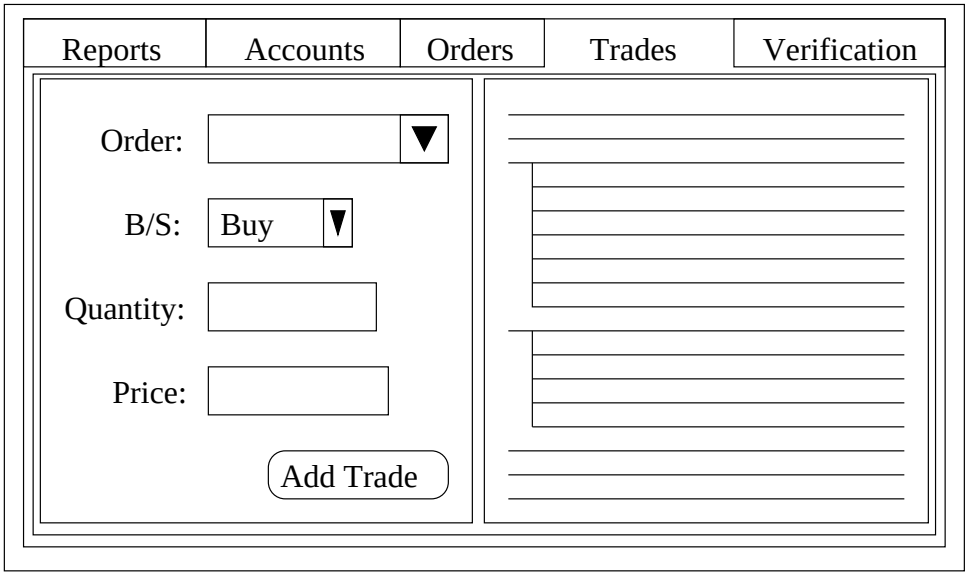
This pane is for managing accounts and customers. Every account will be associated with a customer and some customers will have multiple accounts.

3.3 Orders



This pane is for adding orders at the beginning of the day and throughout the day.

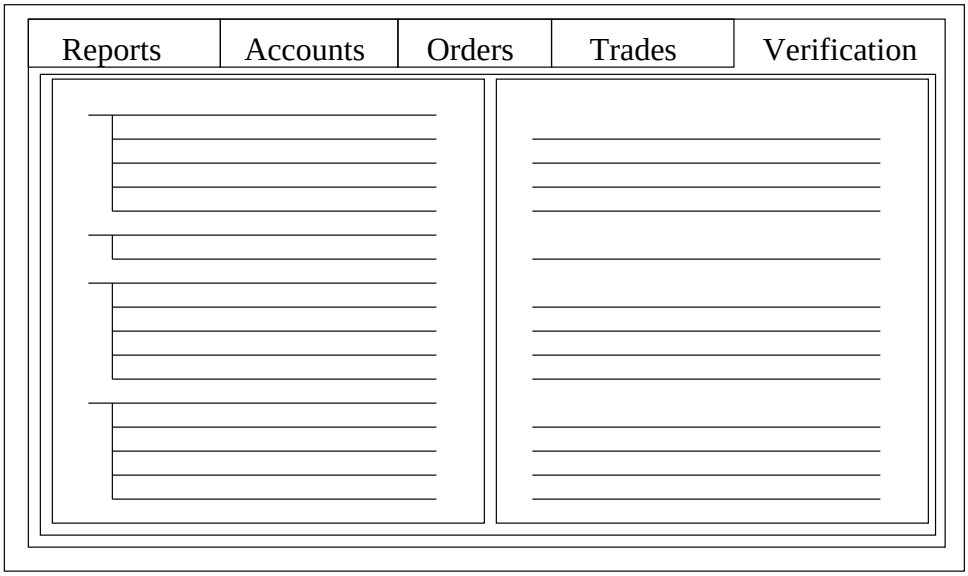
3.4 Trades



This pane is for adding trades throughout the day. The tree on the right shows a list of all the orders for the day and each order may be expanded to see all the trades that have been done for that order. Right clicking on an order gives

the option to delete it. Right clicking on a trade gives the option to delete it or edit it. Clicking edit brings that trade's data into the fields on the left.

3.5 Verification



Throughout the day the Chicago Board of Trade (CBOT) will publish preliminary trade reports via an ftp site. The CBOT Verifier will automatically grab those prelims and compare them to the data in our database. This window basically displays a diff of the data.

4 Non-Functional Requirements

4.1 Performance

- Reports should not take longer than 10 seconds to generate.
- All other interactions should not take longer than 1 second on average.
- The application should not take longer than 5 seconds to start up on average.

4.2 Reliability

- Running in a "stable" environment, the application should not crash more than once a year on average.
- Running in a "stable" environment, the application should not currupt or lose data more than once a year. If lost data is recovered quickly and automatically, then the loss does not count against the once-a-year quota.

- The application should be able to sustain several hours of heavy use five days a week.

4.3 Testing

We can simulate workdays by getting all the data for one day from Sheffield Trading LLC and input it as if it were happening in realtime. Things to pay attention to:

- Are reports generated quickly and accurately? Are the templates parsed correctly? Excel spreadtsheets generated correctly?
- Does the database respond quickly and store information accurately?
- Does the verifier automatically detect new prelims? Does it parse them corectly? Does it diff the database well?

We will also need to have Connie try out the software to see if she's comfortable with it. Maybe she can do some simulation days to get used to it and then give it some live trial runs to see how it performs under stress.

4.4 Ease of Use

The GUI needs to be built very carefully so that it will be as easy and intuitive as possible for Connie and Bucky (not the most computer savvy people on the planet).

4.5 Portability

This application needs to run on Windows, so programming it in Linux, while possible, may prove to be a pain.

4.6 Dependencies

Polling the FTP server and sending Emails both require use of the Internet which means the application is going to rely on Sheffield Trading's Internet connection. Furthermore, we're likely to implement both of those modules, as well as the XML Parser and the Excel Generator simply by using some library that either C++ provides or we find somewhere online. In which case, we'd be relying on some external library for that functionality.

4.7 Documentation

- There will be a help option in the menu bar of every window that leads to information about how to use the current window.
- There will be a README that goes with the software and explains the software from a more general perspective.

- The README will also include an outline that links to the same help pages as are displayed by the help menu options.

5 Requirements (by priority)

1.
   - keep track of all customers and their accounts
   - be able to enter trades into a database
   - group trades by account and order
   - generate mid-day reports (excel)
   - generate day-end reports (excel)
   - generate monthly invoices (excel)
   - generate month-end revenue sheet (excel)
   - generate year-end revenue sheet (excel)
2.
   - keep track of trade verification levels
   - email generated reports at click of mouse
   - generate arbitrary report (based on template [xml?])
3.
   - automatically parse and check CBOT trade prelims and recaps against database
   - automatically parse and check CBOT account statements against database
4.
   - monitor quote ticker and display relavent quotes/spreads based on current open orders
   - take an excel spreadsheet made with certain special tags and convert it to a report template
5.
   - color code and/or sort quotes based on nearness to order price

6 Risks

- If the GUI isn't done just right, Connie probably wont use it.
- Parsing PDF may be difficult.
- Generating formatted Excel spreadsheets may be difficult.
- Defining XML to describe report templates may not be trivial.
- Getting such a complicated application to be super-reliable will be a challenge.