

Project 4 – QT Assignment  
March 23, 2011  
Chris Nance  
Alex Clemmerr

## Introduction

Welcome, dear grader to the worst assignment horror story of all time. Sit down and prepare to be dazzled.

Chris and I initially started working a Twitter called qTwitter. The original idea was to add two buttons, one labeled “Make a Friend” and the other labeled “Make an Enemy”. The former finds a random person with more than 50 followers and gives them a compliment, while the other compares the spelling of many Twitter users and find tweets where a very commonly-spelled word is misspelled, and issues them an insult.

**Problem 1:** As you may already know, Twitter disallowed a lot of apps this last week. Ours stopped working on Thursday, *after* we had finished a lot of the work. Twitter uses OAuth for client authentication, which basically makes it so that applications do not have to store your password themselves – instead the apps are given client tokens, and the users have to approve the app to act on their behalf. But the client never stores a password.

The Twitter API changes literally caused this to stop working for all clients that the OAuth library; other clients for which this is true include (a better-designed client called) qTwit. I wrestled with the OAuth library for awhile, but after 12+ hours of looking through it, I decided that it probably was not going to be fixed in time.

Jim is aware of this, and he told us to mention it in the report. So here it is. We considered simulating a Twitter stream, but attempts at this were really a lot weaker than the actual app performed. Simulating via plain old POST and GET requests was very laborious, and neither of us had time to implement them directly.

So, we decided to switch ideas entirely on Sunday. Or actually, I should say that *I*, Alex Clemmer, decided to switch ideas on Sunday. This leads us to:

**Problem 2:** Chris has been MIA since Thursday’s class. I *really* don’t want to get anyone into trouble here, and I especially don’t want to have an effect on anyone’s grade, but the simple fact of the matter is that our new project sucks, and it sucks because I had to do it *alone*, and *in a day and a half*.

Yes, you read that right. Chris has not responded to any of my attempts to contact him since *Thursday*. Normally he’s a pretty reliable guy, and my best guess is that he got into some sort of accident (???), but the simple facts are the simple facts. I have been on my own with this, and I can’t tell you how frustrating it is to be closing in on the deadline with literally dead silence from his end.

The timeline breaks down like this:

On March 19<sup>th</sup>, 4 days before deadline, I sent him an email explaining that I’m digging around in OAuth to find out where the problem was, but that it is largely fruitless. We’ve planned it so that he’s supposed to have a backup plan in case this doesn’t pan out.

The 20<sup>th</sup>, 3 days before deadline, I send him another email, explaining that I definitely know where the problem is, but that I’m definitely not sure how to fix it. I suggest that we move onto one of the replacements. I try to give him some space because I’m not sure how to handle it.

The 21<sup>st</sup>, 2 days before deadline, I tell him that I'm getting very nervous. I suggest that the need to move onto another project is imminent. No response. I send him 4 more emails suggesting solutions. I independently start to freak out and go about building a solution on my own.

The 22<sup>nd</sup>, 1 day before deadline, I send him 4 emails explaining my progress on the project. No response on any of them. I am starting to get really, really scared. I ask if he's getting these emails. No response.

The main source of frustration for me here is that I am very certain that this project would have been done if only one of these things had happened. If Twitter never changed, our project would have been much closer to done anyway, and as long as I could have gotten Chris to push his work into the git repo before he went offline, I could have done the rest. Likewise, if Twitter had failed but he had been there to help, the project would again have been done. Having both of them happen at once has been catastrophic.

Another huge frustration for me is that I feel that I definitely could have gotten the project done if I'd gotten started over the weekend, and I would have gotten started on the weekend if I'd known that he was going to drop out. I am not even close to being the fastest coder, and I really rely on talking to people about problems and having time to absorb the code. Without those things I am completely crippled, and I personally feel that I did not have enough time to do everything by myself on a completely new project.

## **In conclusion**

Overall, I'm not going to make excuses for this all not working out. I understand I am not the perfect partner, and that I did screw some things up here. But I can't express to you how upsetting this whole scenario has been. And even more frustrating is that there does not appear to be a good solution to the question at hand. I shouldn't be exempt from class rules, but I also don't feel I should get a grade like a C or a D, because there isn't a whole lot I feel I could have done that I didn't do. But again, not having done (in my opinion) C work, it makes it difficult to ask for such a grade. The fact that this is a dilemma at all makes me bitter both towards group work and towards school in general, and in a way that is difficult to even communicate.

## **Planning, project 1**

As said before, our specific plan was to add two buttons to the bottom of the qTwitter pane. One reads "make a friend" and the other reads "make an enemy". Clicking each brings up a dialog box that has text which the respective insult or compliment in it; we select them at random.

In the background, we would need to find users to both compliment and insult. This needs to happen while the user is doing other things. So there needs to be a saved base of Tweets that we have access to. Not a lot, just enough.

## Implementation, project 1

All of the GUI stuff was implemented as of Twitter's OAuth break. That part is easy. Also implemented was posting the Tweet: it was really just a method call. What needed to be implemented was to find the users who needed to be insulted. I contend that this is not as hard as it sounds, but not having access to the API anymore, I wouldn't know.

## Planning, project 2

As said previously, in the last day and a half I've been feverishly working on another project entirely—zNotes. I've been implementing a spell checker over the plaintext notes. The button is on the very left end of the control panel when you open it. When you click this spell checker button, it highlights words that are wrong. I'd hoped to have a "replace" button, but this will probably not pan out in time. Users will have to edit the text manually themselves.

## Implementation, project 2

I'm really pleased with the progress I've made in the short amount of time I've had, but it is not done. The button that brings up the dialog box, the dictionary load, and the finding of the words all work in principle, but transitioning between words using the "next" button has not been shown to work, and I've not tested the app at all. PLEASE NOTE: this may change, as I'm writing this report before the deadline to give me the ability to work until the deadline.

## Summary

Overall, I learned a lot during this project. I started out building the QT project using only vim. *Huge* mistake—you want to use Qt Creator. It's loads faster. As in, fast enough that I felt like I was cheating. I also have a healthy appreciation for how much worse callbacks are compared to the Qt signal/slot relationship. I particularly like the fact that slots and signals are typesafe by default.

The other thing I really understand now is how crucially important commenting is. None of the projects we found were built with good comments, and it made them stupendously horrible to work with. Digging through the code to find declarations and tracing through method patterns is exhausting. Strangely, having done it *twice* now, I can say that this astonishingly is not better the second time around.

This all, of course, is to say nothing of the things I talked about before.