Drunk@Home

Retrospective/Reflection/Epilog

First of all, The Drunk@Home Team wants to express its great relief of the project, and it’s oncoming completion.

3 Key Lessons

If we could sum up 3 key lessons, they would have to be:

* Division of labor: This is most likely the most important lesson we learned from this project, and although it has nothing to do (in particular) with the course’s syllabus, the Team feels that this was the Key Component, which we had problem implementing properly.
* Testing: A key lesson, which whomever does not implement it in its proper time, shall not reap his success (well, at least not on time…).

Testing is most likely the KEY of success in a software project, although very time-consuming, and drains you of your life fluids in its process, it can and WILL save your ass when judgment day comes (showing your progress to the class/tester)

* Prioritizing: In each MS’s end, the team always had a feeling that we “Bit more then we could chew”, this reflects both on the team’s moral, and both on the project’s/MS’s completion, we feel that we could of done better, if prioritized correctly.

What went well/worse/horrible…

What went well: Implementing our MVP’s – Search By Name, and Search By Ingredients, and of course the Admin Page all three work precisely as we wanted them too, and most importantly – on time.

What went worse: Implementing our peripherals, such as the “Missing Ingredient” system, which rendered itself pretty much useless once we introduced two new fields, “Basic Ingredients” and “Trivial Ingredients” which shortened this feature’s usefulness to pretty much a nil.

What went horrible: Our grades.

What we’ll do differently on the next project

In the beginning of this document we wrote about “3 Key Lessons”, I believe that now, after we have been burnt by these lesson’s fires, we will take them into the hardest consideration possible.

Challenges/Risks

We believe that we took on quite a few challenges/Risks in the making of this project, the main ones would be:

1. Learning new languages: We learned Perl (pretty much from scratch) in order to implement the Website’s functionality. We learned HTML5 (also from scratch) and CSS (likewise) in order to implement our design vision correctly.
2. Working under stressed hours, with other courses in mind: this section is pretty much self-explanatory, it is currently 02:00 AM, and were awake and working on the project…
3. Handling EVERTHING under ourselves: many teams did not implement their own servers, nor did they develop their ENTIRE code from scratch, but rather used an existing solution (e.g hosting service, the collage’s resources, etc..), we decided that we are going to do everything ourselves, like: setting up our own dedicated server running Apache, MySQL, SAMBA, remote control (VNC), SSH with RSA encryption, Webmin (server management panel), and so on. All these were of course overlooked in the project’s grading…

Tools and other “helpers”

In the beginning of the project we tried to research different MVC’s and RAD’s, like Microsoft’s Visual studio, Adobe’s Dreamweaver for website design, Adobe’s Flash, various automatic Android app builders, different SQL manages and other well-known software that’s widely used on the market.

We decided not to use anything. Since all these products wanted us to work “their-way”, not allowing us the freedom we wanted to properly implement our vision.

The tools used in the actual making of this product were putty/SSH, vi/vim, and some good-old pen and paper. (well, a bit of PowerPoint was used to create some of the pictures on the website…)

You can say that we did not stand up to assignment given to us by Reuven (he insisted everyone would use RAD’s and MVC’s..)

But we personally believe that we made the right choice for US.

What’s planned for the Drunk@Home project

We still consider purchase of a FQDN for the website, and we’re still going to host it on our ADSL line, with option to move to a dedicated hosting.

The project is already Open Source. We currently have no plan of developing it further.

General Comments(on the course):

We feel as if the course entirely blew-away (disregarded) the whole web development theme, and concentrated on “Big Project” development, such as different working methodologies (such as SCRUM, which we implemented ourselves, and Extreme Programming, which we did the entire time.)

Web Development is about 50% of today’s software development…