

FTS Web Architecture and Helmet Computer Software Integration Estimate

This is meant to be an estimate of time and expenses for the website and its integration into the overall FTS system. Any items with a grey background are being left out of this estimate for now but will be added as needed later.

1. Setup new Linode hosting account, link it to www.eyehatch.com and setup Linux OS on Linode share there. This involves coordinating the registrar (GoDaddy.com) and Linode so that GoDaddy points their DNS records to the Linode DNS servers. It's more trouble than it should be as I learned when I set up my own website. **[2-3 hours]**
2. Installation of "LAMP" web Software on Linode server **[1-2 hours]**
 - 2.1. Apache Web Server
 - 2.2. PHP language installed and activated.
 - 2.3. WordPress
 - 2.4. mySQL
3. WordPress blog: the initial homepage. Total time estimate **[12-24 hours (depends on amount of graphics customizations)]**
 - a) Customizations: create more homepage-like user experience while allowing for us to easily add or delete content quickly. Custom artwork, logos, effects, etc.. can be created and added to WP blog. I can create some artwork and do some photography but I'll warn up front that those things can be time consuming.
 - b) Maintenance: will be a back-and-forth process after initial site creation with ill-defined time requirements. (ongoing process – billed as needed by hourly rate)
4. mySQL database: designed and diagrammed as Entity Relationship Diagram – ERD. Database contains both website users and players as well as information about teams, leagues, etc.. **[6-12 hours]**
 - a) Data structures needed (minimum)
 - Realtime game and testing data on players or test subjects (location, speed, acceleration, mass, height, etc...) Helmet Computers make remote database calls to sync their positions to the database. Realtime user view from web server is then constructed using that data.
 - Website user data. Users have the usual user attributes plus user privilege level: Guest(free), Fan, Fan Premium, Admin.
 - Team, league, and other forms of recreational organization data.
 - Product database and e-commerce site to sell product. Leaving this out for now as we have no product yet. E-commerce also needs special attention to security and extensive testing of the whole process including how to handle refunds, RMAs, etc...
5. Firewall software. This is initially just the standard firewall software that comes with any version of Linux. Essentially I'll limit access to our server to just the mySQL port (3306), HTTP port (80), and SSH port (22). I'm thinking that eventually we might have a virtual private network (VPN) for players and the database server and then have the webserver as the only machine that is on both the VPN and public internet. Initially

there won't be a VPN. If we need one later I'll have to set that up and it will be a separate job. **[1-2 hours]**

6. Web-based GUI to replace current Python GUI fully integrated with player data coming in from the field. This also functions as a way for both fans, testers, and us to see the locations of players in realtime. I'm thinking we might even use a 3D view of the field using a thing called threejs. You can see an example of where I'm using this to show off some of my 3D printing designs on my homepage here <http://gregorypropf.com/designs/>. If you click and drag with the mouse you can move the viewpoint and the mousewheel zooms in and out. The people who wrote the graphics library have many demos at their site <http://threejs.org/>.

This is a little different from what WordPress was designed to handle and will involve setting up a web application framework. I'm leaning towards using the Luminus framework at the moment. This uses the Clojure programming language. I like this language environment because it provides a way to write both the server and client side code in the same language. Client side code compiles to JavaScript and runs in the user's browser. We'll be using jQuery AJAX calls to allow for an animated realtime view of the field in the browser. This will also be useful for us during the testing period. **[12-36 hours (as above with #3, it depends on how complex the graphics are)]**

Time Estimates (in hours)							
Item #	Min Time	Uncertainty Factor	Max Time	Comments	Hourly Rate	Low Est.	High Est.
1	2	1.5	3		60	120	180
2	1	2	2		60	60	120
3	12	2	24	see notes above about graphics work	60	720	1440
4	6	2	12		60	360	720
5	1	2	2		60	60	120
6	12	3	36	see notes above about graphics work	60	720	2160
Total						2040	4740