

# Tech Brief Questions

## Basics:

1. What is the main purpose of this website?
2. How “big” is this project?
  - a. Approximate number of screens/pages in the website/application
  - b. What are the basic functions of the site? i.e.Registration, Game, Ecommerce, Search, catalog, photo upload, discussion boards, voting, etc
3. What data will be stored/accessed locally? Will the website/application require a database?
4. How will the site be updated?
  - a. Will it require a CMS?
  - b. What content is dynamic vs static?
  - c. How frequently is the data updated (hourly, daily, weekly, monthly, periodically)?
  - d. Where will the data come from? (client or administration back-end of any kind)
  - e. Who is responsible for updating the data?
5. What are the reporting needs?
  - a. Basic site traffic?
  - b. Will it require content-specific reporting? (i.e. how many new users signed up between specific dates, etc.)
6. Will there be a mobile/WAP component?
7. What operating systems, browsers and browser versions are we targeting? Do we care about IE6 or other older browsers?
8. Are there additional considerations such as printer-friendly versions, BOBBY/ADA-compliance, age verification or other that we need to be aware of?

## Domain Name & Hosting:

1. Where will this application/website be developed? If it will be developed on a different machine than its final hosting situation, we must provide time for file transfers, additional QA once the site is moved, etc.
2. Where will this application/website be hosted?
  - a. If we are hosting, will it require a dedicated host, or virtual hosting environment?
  - b. If client is hosting, have we confirmed that their server supports the technology we intend to use?
3. Will it require its own domain name?
  - a. If yes, will the client provide, or will we?

- i. If yes, we must allow time for propagation
  - b. If no, at what url will this live? (sub-domain of noise, sub-domain of primary client domain, etc)
    - i. If sub-domain of client, we must provide time to work with the client's IT dept to configure DNS settings
- 4. Will we need to manage any e-mail accounts for the project? If so, what are the account names, and are they pop3 or forwarding accounts? Who will be checking them?
- 5. Will this project be migrated to the final hosting location from an existing location? If so, a migration plan (including e-mail address, DNS, content, database, etc migration should be created.)
- 6. Is this project ongoing, or for a limited time? If for a limited time:
  - a. What ongoing service or support will be required? Webmaster support email? Unsubscribe administration?
  - b. What is the plan for the domain name and/or website once the project is over? Do we keep hosting the project? Do we let the domain name expire, or simply redirect it to the main client site?
  - c. Should the site be added to our monitoring services via *Site 24x7* (checking for downtime, response time, etc.) Who should be on the list of alert contacts if there is a problem detected? There are two different kinds of monitoring services we have available:
    - i. Simple response time ping – this service attempts to contact the server every x minutes. If the server doesn't reply within y milliseconds, it is classified as a failure and alerts are sent.
    - ii. Complex application monitoring – this service allows us to create an action template (must be created using a Windows machine) that reproduces the process of logging in to an application and performing specific actions. This could be used for Facebook applications or other applications that require a login.
- 7. If the project is hosted on a server we are managing, server setup should include:
  - a. Realistic bandwidth estimate. We should confirm with the hosting company we select that if we surpass anticipated bandwidth, there will be no interruption in services. (Extra charges will apply, but service should NOT be suspended for bandwidth issues.)

- b. Maximum throughput configuration available. There is generally an insignificant additional monthly charge for this.
  - c. If the project relies heavily on MySQL, we should consider making adjustments to the my.conf file to tune the database configuration for optimal settings. (I.e. if the database queries are largely SELECTs, optimize for SELECTs, etc.)
  - d. Fine-tune the apache httpd.conf file to allow for maximum concurrent connections
8. Backup and Restore – if we are hosting the project on a server we manage, dedicated or virtual, some level of backup MUST be implemented for every account. For large-scale, high profile clients running a dedicated server, NAS backup is recommended. For smaller projects and/or projects that are virtually hosted, a nightly automated backup will likely be adequate. Backup should include files, user data, and applicable databases. If the backup has been done properly, restoration is fairly quick and painless.

In the event of catastrophic system failure (such as a hard drive failure), the backup should be ported to an alternate machine as soon as possible to restore services. Once the affected machine is running, DNS should be altered to point to the replacement machine as soon as possible, and DNS records at the registrar level should be changed if the replacement machine is going to then serve as the live machine.

Alternately, a cloud-computing solution may be desirable, depending on project requirements and scope. If a cloud computing solution is chosen, this decision must be made early on in the process so that the development can be done with this setup in mind.

## Facebook Application & Verification:

Facebook application development is a complex process and will be covered in more detail in another document, however every Facebook application we develop should conform and pass Facebook Verification. It should be assumed that any Facebook application we develop will be submitted for Verification, and should be planned and executed with the goal of passing Verification.

Besides cultivating user trust, there are a few additional (significant) perks to being Facebook verified:

### **Badging**

1. A verified badge graphic displayed on your application's About Page indicating that it has been verified by Facebook (see attachment).
2. A special icon displayed next to your application in listings within the Application Directory.

### **Distribution**

1. Increased allocations for requests, notifications, and emails.
2. Increased visibility for News Feed stories.

The evaluation process is still new, so it may end up taking longer than they are currently estimating, but according to Facebook, the process should take "several" weeks. If an application does not pass validation, we would be notified as to what we need to do to fix it. If an application does not pass due to Facebook policy violation, they will take appropriate action against us (such as suspending the application's notifications/newsfeeds, etc), we would be unable to submit the application for verification for 3 months, and our application fee would not be refunded.

We would not need to postpone launches if the verification process takes longer - we can always launch according to schedule, and when the verification goes through, we can begin to take advantage of the additional perks.

For more information on the Verification process, see:

<http://developers.facebook.com/verification.php?tab=checklist>

<http://developers.facebook.com/verification.php?tab=about>

### **Flash/Video:**

1. Will there be substantial flash or video aspects to this project? If yes, we should consider hosting arrangements and ensure we have the bandwidth available and ensure the throughput is set up to handle maximum traffic.
2. If the primary site or application content is Flash-based, we must budget time to create an SEO-optimized version of the page in plain text. Copy will need to be obtained from the client, and MUST accurately reflect the true content of the Flash. Attempts to keyword bomb or deceive search engines will result in being blocked from the search engine altogether.

## Email:

1. Will the application or website be communicating with users via e-mail or text messages? If so, we may wish to consider outsourcing email/text messaging for ease of management and to circumvent potential SPAM blacklisting issues. This decision needs to be made on a case-by-case basis

## Integration:

1. Will the application or website be required to interface (push or pull) with a third party system or service? If yes, documentation on the integration should be obtained at the beginning of the development planning so any potential obstacles can be addressed.

## User and Content Filtering:

In projects where users will be contributing or creating content using the tools we develop, is there a need to build in support for:

1. Banning or blocking specific users by username or IP address?
2. Moderate content before it is displayed publicly or at least the ability for an administrator to delete or unapprove content to remove it.
  - a. If yes, by what method will offensive material be flagged as offensive? Does the application require a "report this content" function, and if so, what needs to happen when content is flagged? Who is notified, and what actions do they take? Does the system automatically unapprove content that is flagged many times?
3. Scrubbing or filtering bad words. If yes, what action should be taken if an offensive word is found? (i.e. is the offensive word replaced with \*\*\*\*, is it rejected by the system, etc.)

## Pre-Existing & New Content:

1. Does this project require us to import existing client data or content into the application? If so, in what format will the client be providing this data? (Excel, csv, Word, etc.) Adequate time must be allowed to scrub the data and create custom import scripts, and then to QA the imported data.

2. If content being provided runs on a timeline (such as the Daily 200, where a new question is presented every day), do we need to set up an alert system that notifies an administrator if content is about to or has already run out?
3. Will there be ongoing new content that would be appropriate for use in an RSS feed? (Events, articles, blog content, etc.)

### User Support:

1. What options does the user have if they have experiencing a problem with the application or website? Is there a Help/Tech FAQ page, a contact page, or some other way that they can reach out to us for assistance? What e-mail address should be used for replies?
2. If we are going to be sending the user e-mails or text messages, we should outline the nature and frequency of this communication very clearly on a page available to the user.
3. A privacy policy should be built into the site or application whenever we are collecting any information from the user.
4. We should make it very easy for a user to globally opt out of our communications. Where applicable, on the opt-out form, we should ask the user why they chose to opt out so we can better tailor future communications.
5. Any email or text communication should require a double-opt-in confirmation.

### Statistics & Reporting:

1. What aggregate statistical reporting needs to be done on this website/application? (Omniture, Google Analytics, Refresh Analytics for FB apps)
2. Do we plan on implementing click heatmap technology and/or multiple page versions for conversion rate evaluations? (May not be available on Facebook applications.)

### Marketing and Promotion:

1. What media is driving traffic to this site? Online/Offline
2. What is the schedule for that media?
3. What is the total exposure?
4. Are there any specific activities that would drive spikes in traffic? Superbowl Ad, etc.

