Patrick Austin

CS 425 Individual Assignment 1  
9/10/2017

**1. Possible Project Idea**

Example product, explanation of innovation and usefulness

One software project that I would like to develop in a small group would be an experimental social networking app for phone and PC platforms. Unlike many prominent competitors in the social networking space which seek to connect people who already know each other via family, school, or other pre-existing social group, this app would emphasize encountering new people and coming into contact with new viewpoints. This approach is partly inspired by forum systems of the 1990s and early 2000s, in which users self-selected their communities based on personal interests rather than being herded by friend-suggestion algorithms, which often work to connect people who already know each other or share existing social circles. This focus on avoiding the properties of existing social networks that promote ‘cliques’ or ‘echo-chambers’ would represent an innovative approach that may appeal and be of use to individuals not enamored of current social network ecosystems.

Target users, functions, capabilities

Thus, the target users for this application would be those who do not use existing social networks much or at all, as well as those who use them but are experiencing fatigue or dissatisfaction. In terms of features, this app would aim to try to modernize the forum experience for the social network age, possibly by providing de rigeur social network features such as profile pages, individual and group chat, and algorithms that suggest new friends and groups. These could be joined with more classical features of forums, such as persistent discussion topics, and perhaps even a system akin to the upvote/downvote used on the popular modern quasi-forum Reddit so that users can curate content to promote quality discussion.

A high degree of emphasis would be placed on algorithm design, in order to differentiate this social network app from competitors. Where Facebook and Twitter study your existing friends and groups and point you towards friends and groups that are similar, this app would deliberately seek to connect you with people and groups that may share some of your interests but differ on others, as well as those who may exist outside of your comfort zone in terms of geographic location, age, educational background, and the like.

Idea origins

As mentioned, this idea originates in largely antiquated forum systems of the earlier days of the internet. As someone who grew up in that milieu, I have long been struck by the lasting friendships I have witnessed and experienced that grew out of these systems. They connected far-flung people in surprising ways that cut across divides of class, race, and geography, and were, I think, the first window of many in my generation to the wider world. In contrast, popular social networks in the modern day have little appeal to me personally; they often strike me as a place to go to have one’s own biases confirmed, to preach to the choir, and to reinforce communities that already exist rather than to promote the creation of new ones.

Comparison with existing products

Obviously social networking is a robust multi-billion dollar industry and a small team with little capital would have to pick its battles carefully. To compete with Facebook or Twitter feature-for-feature would be impossible, although from the perspective of an aspiring software developer I’m sure the attempt alone would be illustrative. Were this idea to truly compete it would face an initial gap of features and users. The challenge facing any new social network, I suppose, is how to begin; these are systems driven by inertia, where the product gains value the more widely it is used. What, then, is a brand new social network to do? Still, the basic concept serves to differentiate this social network from competitors, and I think even the narrow goal of exploring new paradigms for algorithms that connect users could be innovative, fruitful, and realistic to boot.

**2. Software engineering and your professional career plans**

a. Web developer

A web developer is a software engineer who works on the front or back end of a web-based project. Front-end development focuses on a user’s experience with the app via web browser via HTML, JavaScript, or the like. Back-end development involves the creation of code and systems for the server side of the web app, which users interact with only indirectly. The back-end role can involve creating a robust API of commands which can then be called on a variety of platforms, such as mobile apps. From an educational perspective, this job emphasizes computer science knowledge and especially programming. Salaries range from $ 60-135k. I expect that software engineering knowledge gathered in this course will be directly applicable to this career path, seeing as web development is a vibrant and high-demand area of software development. Learning what it’s like to work in, say, an agile environment would be of great use to an aspiring web developer.

b. Product manager

A product manager is the one who coordinates or runs the development of a software project. This job involves managing a team of software developers, and can also involve coordinating with other areas in the organization such as marketing and art departments, as well as adherence to time and resource budgets, and perhaps an ability to pitch the project persuasively to investors. This job involves broad knowledge and cross-disciplinary experience, so suggested educational areas include not just computer science but also management and business. Salaries range from roughly $ 100-136K. Intuitively, familiarity with software engineering should help one to coordinate, guide, assist, and understand the needs of a team of software engineers in order to enable the project’s success. For example, a product manager has an important voice in choosing the software development style best for the project.

c. Database architect

A database architect, or database specialist, is responsible for designing and maintaining an architecture for data the organization gathers. This includes designing data structures and metadata, designing appropriate procedures for access, use, and modification of the data, and even analyzing data via data science in order to extract important lessons and correlations from highly complex multidimensional data sets. This is a senior position for those with ample past experience in database development, analysis, and engineering. Salaries fall in a range from roughly $ 111 - 135K. Even if this is not directly a software engineering position, it still involves close coordination with software developers, both so that useful data is being securely and accurately gathered from the software and so that past lessons from data can be incorporated into future software design. Understanding best practices of software development would help a database architect to interact with software engineers so that the best possible data can be gathered and utilized.

**3. Software quality**

Purpose of the application

One software application I am familiar with from everyday use is Foobar2000. Foobar2000 is a free audio player for Windows, and has been my tool of choice for playing music on a desktop platform for almost a decade. On the basis of my longstanding use of and appreciation for this app, I would call it a piece of good quality, high functioning software.

Examples of core functions and features of quality, explanations of quality

One core feature of Foobar2000 is its support for a wide variety of audio file formats. Even the heavyweight iTunes and its maker Apple have long refused to support lossless .flac audio files; Foobar2000 handles these and a multitude of other common and uncommon file formats with comparative ease. That a free app succeeds in this respect where a titan like iTunes continues to fail after more than fifteen years of development is surely a sign of quality on the part of Foobar2000.

Another core feature of Foobar2000 is its powerful customizable user interface. Users can customize and reskin Foobar2000 to an extremely high degree, opting to display, omit, or rearrange a wide variety of information about audio files that are relevant or irrelevant to the particular user. Foobar2000 also provides an SDK for third-parties that enables even more extreme modification of the interface. Some of these are completely unrecognizable when compared against the stock Foobar2000 UI. This feature enables users to achieve the type of audio player experience that they desire, whether minimalistic and unintrusive or encyclopedic and visually striking. This ability to appeal to a wide variety of users with one application is a sign of quality, I think.

A third core feature is the lightweight nature of Foobar2000. Whereas iTunes has become (in this humble user’s opinion) bloated over the years in terms of features, file size, and performance, Foobar has maintained a very light footprint on all fronts by focusing on its core area of speciality: music. Foobar does not support video, ebooks, an online storefront, device synchronization and backup, photos, and etc. that have accumulated in iTunes; it is thusly able to offer the best specialized experience that it can in its area of focus. I think this is another mark of quality for the software.

Areas for improvement and suggestions

One feature of Foobar2000 that could use work is its interaction with multimedia devices. Foobar2000 is unabashedly focused on the PC platform, and thus can come across as a bit of a relic in the age of the cell phone as music player. In order to sync music to an iPhone, for example, one cannot avoid dealing with proprietary Apple software like iCloud and iTunes even if one prefers Foobar2000 otherwise. If Foobar2000 could sync a music library to iPhone or Android directly, I would be able to cut out the Apple middleman and uninstall iTunes entirely; I would be thrilled. This would probably be a highly difficult and complicated feature to implement, however, so I do not fault the creator of Foobar2000 for its absence. Moreover, its addition would to some extent contradict the otherwise minimalistic ethos of the app.

Another feature I would like to see in Foobar2000 are better out-of-the-box skins for the user interface. In general the UI presets included with base Foobar2000 tend very much towards the minimalistic; those seeking the more beautiful bespoke UIs can be drawn into a vortex of search for third party add ons that make use of the UI modification SDK. It would be nice for new users if some more impressive, striking options came included. This would help Foobar2000 have a stronger appeal to new users and users who want a fast, easy setup experience. Since these add ons already exist and are free, their inclusion would not on its face seem too difficult a task. I think this is a practical area for Foobar2000 to improve and expand its audience.