Project proposal Comp 260

The three artifacts I propose are as follows: A one-to-one direct messaging app. A streaming service app and an app focused on a new method of calling emergency services. Out of these ideas I have chosen the one-to-one direction messaging app as the artifact I am going to create. To summarise this app, it will be a real time messaging app between two clients. Created with a basic HTML (Potentially using handlebars for templating if there are many pages) and CSS frontend with express.js and the socket.io JavaScript library, which will allow real time communication between clients, necessary to create an application of this sort.

A breakdown of the application’s main features and requirements includes a front-end page that includes a user interface where messages can be written through an input field, sent via a server provided by socket.io and received by the end user. Through an output field users would be able to view who they are logged as, which users are active, active user ids, when their message was sent, which user it was sent too and the time of the message delivery. These features are achievable within the scope as one front-end page, a basic user interface and two register clients to send and receive are required for this app to work with the back-end functionality taking up most of the time frame. With additional features such an SQLite database that allows users to view the sent messages in a database format for persistent storage, a system management system to have multiple communications occurring at once and a user registration system to create more than two users.

This artifact can be considered a distributed system because of the socket.io library’s realtime element. With the artifact featuring two clients it therefore achieves the element of communication across different networks with sending messages across the network being a basic process of a distributed system. Socket.io allows this to be in real time by having users operate off differently assigned ports to run the application off the same system. Within web applications this sort of application is considered ‘live chat’ service, a modern and increasingly popular [1] version of web page communication between site owners and the client.

The service this artifact provides is a solution to individuals or small groups needing some form of communication from their webpages. An example of this is small static sites such as portfolio or blog pages. These are usually created by a small team or individuals, and these sites often use a contact form to sends emails. Although this method is easy to achieve with the use of plugins such as ‘formsubmit.io’ it can create issues. If the email provided is personal and not specifically for that site, which can often be the case for individuals, it can create a lot of unwanted spam. Methods to solve this problem would be to use a reCAPTCHA, although these can be cracked [2]. Therefore, a ‘live chat’ style application becomes a solution. By including messaging system in the site, in place of a contact form it not only prevents this issue it allows direct and more frequent communication between the admin and the client, as opposed to being forwarded to a customer support email. Which then requires an additional non-automate response for the client to reply. As a result, removing these steps with ‘live chat’ provides a better user experience [3]. There are security concerns having messages directly accessible however, but this can be countered using encryption methods such as hashing.

Implementing a ‘live chat’ (one to one messaging) system gratifies user satisfaction levels to a greater extent than a static contact form [4] translating into positive attitudes towards the sites systems. This is attributed to live chat being a more personal service, as the client engages directly with a real person unlike a static form that is often forwarded to an automated system, although on larger sites chatbots often replace people. Therefore, by targeting small team sizes that have a lower chance of opting to chatbots it maintains this personal aspect. As a result ‘live chat’ generates more site traffic, conversion rates, and satisfaction levels when compared to static forms [5]. Small sites specifically rely on initial traffic to develop utilising ‘live chat’ speeds this process.

This artifact was chosen over the other two proposals due their foreseen constraints. The streaming service proposal will require lots of licensing to use actual thumbnails and videos outside of using stock. This would prevent most of the sites features from working correctly. The emergency services app was designed to work more efficiently than phone calls. However, this created far more issues than the problem it was trying to solve. Not using phone signal would mean the application would need WI-FI signal instead and a strong connection is often harder to receive than cellular, especially in remote areas. GPS location would also have to be utilised to create the improved efficiency, which has the potential to create security issues. The maps used in the GPS would also have to be divided into named quadrants that users could quote provided clarity to the emergency service about their location and therefore improved efficiency. However, providing this data even for a small local area would be far out of the projects scope.

Bibliography

[1] Cleveland, B., 2015. *The Stats Behind Chat: It's Popular and Growing*. [online] Icmi.com. Available at: <https://www.icmi.com/resources/2015/the-stats-behind-chat-its-popular-and-growing> [Accessed 9 February 2022].

[2]Tschacher, N., 2021. *Breaking the Google Audio reCAPTCHA with Google's own Speech to Text API*. [online] incolumitas.com. Available at: <https://incolumitas.com/2021/01/02/breaking-audio-recaptcha-with-googles-own-speech-to-text-api/> [Accessed 9 February 2022].

[3] Ruby. 2020. *The Benefits of Live Chat on Website User Experience - Ruby Blog*. [online] Available at: <https://www.ruby.com/blog/live-chat-user-experience/> [Accessed 14 February 2022].

[4] Elmorshidy, A., 2013. *Applying The Technology Acceptance And Service Quality Models To Live Customer Support Chat For E-Commerce Websites*.

[5]Clark, D., 2019. *Why Live Chat Is Important for Your Business?*. [online] Digital Doughtnut. Available at: <https://www.digitaldoughnut.com/articles/2019/september/why-live-chat-is-important-for-your-business> [Accessed 9 February 2022].