

Standalone Web Server Part 1

Who:

- Satchel Spencer
- Brian Ensor
- Robert Ladd
- Dean Moser

Description:

A standalone wifi access point based on the esp8266 system on a chip. The access point will route to a server which will serve small webapps for communication and file sharing between users of the access point.

Vision Statement:

Provide a localized networking tool independent of the web.

Motivation:

This project provides an interesting opportunity to apply our knowledge of web based technologies in the context of embedded systems. Local networks avoid many of the inherent insecurities in the web, and can fulfill many of the same needs on a smaller scale.

Risks:

- The esp8266 has limited memory and we may need to integrate another microcontroller in order to achieve our desired feature set.
- Hardware latency may prove to be a limitation on the responsiveness of the client side application.

VCS:

- Github: <https://github.com/satchelspencer/13/>

User Requirements:

ID	Description	Sizing
U-0	The server will present itself as a wifi access point to potential users.	13
U-1	Users will presented with a login page in order to access their portal	8
U-2	Once Logged in, Users will be presented with a dashboard showing their content on the server	8
U-3	Users may see who else is online.	5
U-4	If other users are online, they may send messages to each other	13
U-5	Users can post text/file content that will be viewable by other users	13

Non - Functional Requirements:

ID	Description	Sizing
F-0	The system will initialize itself as a wifi access point with a SSID	13
F-1	The system will be available as a webserver at a specific IP address	8
F-2	The server will serve the static content of the webapp	13
F-3	The server will store login information/client data for each user	5
F-4	Requests for specific user data will be responded to appropriately	5
F-5	The system will be built using a microcontroller, communicating with an esp8266 for connectivity	13

Functional Requirements:

ID	Description	Sizing
F-0	Users will see a list of other users on the server, click to see their info, and start a chat.	8
F-1	Users can see all of their stored data/files and upload new ones through post	8

	requests.	
F-2	Messages sent to other users will be routed through the server, and confirmed on delivery.	13
F-3	Users can see a history of their activity.	5
F-4	Users can see a list of available apps with which to interact.	13
F-5	Apps will be stored simply as a static webpage with html/javascript that can interact with their user data	5
F-6	Apps will follow a URL scheme according to their names	3

Methodology:

- Agile Kanban

Project Tracking Software:

- meistertask: <https://www.meistertask.com/projects/4309fz0oxh/join>

