University of Utah
UNDERGRADUATE RESEARCH JOURNAL

THE OSI MODEL AND THE SPIRAL UP APPLICATION PROTOCOL IMPLICATIONS

Anastasia Gonzalez (Joseph Zachary)
School of Computing

SPIRAL UP PROJECT OVERVIEW

In today's era, a healthy lifestyle is a large part of most people's personal goals. These goals require constant healthy behavior through the development of healthy habits. However, such habits are notoriously difficult to develop. Health management software is a popular and convenient way to assist this process, especially through the use of mobile apps. Nevertheless, basic activity tracking software does not effectively motivate and support the individual health goals of all users, which results in unsatisfied users who consequently abandon these apps and their habit formation.

Spiral Up targets a user's unique goals and motivations to focus and fuel better health. Within the *Spiral Up* framework, users can create and customize habits using a flexible, templated system, empowering them to leverage the app to precisely and effectively target any habit goal. *Spiral Up* also provides a social media interface, habit achievement badges, and informational articles to drive motivation and engagement in all types of users.

Additionally, *Spiral Up* does more than simply track and display the user's activity—it affords benefits commensurate with the effort the user invests, strengthening their social bonds, improving their self-efficacy, exposing and reinforcing health insights, and enabling them to consistently find success in forming new habits on their way to better health.

Spiral Up provides a user-friendly app containing a graphical user interface (GUI) that displays and allows for changes based on the user's inputs. The app is supported via a back-end server that offers all social media, health insights, data tracking, badges, and habits to be calculated, stored, and transmitted to the appropriate users. When the server or client has information to relay to the other, they use an agreed upon communication protocol that contains only the necessary information for the recipient to read and extract. This communication protocol design will be researched, analyzed, and designed in this paper to determine how to transmit social media, health insights, habit progress, and user input between the server and client in a quick and efficient way that is maintainable for the future.

THE OSI MODEL AND APPLICATION PROTOCOL

In 1969, the first datagram computer transmission was sent across a computer network using the first router from Stanford University to University of California, Los Angeles (UCLA), spurring the birth of the Internet over the next few months [1]. As a result, "the late 1960s produced a need for compatible data communications networks in the 1970s" [2, p.1]. These data communication networks needed to support multiple computers and hardware structures, while allowing various file formats to be transferred across vast distances in seconds. By 1970, many network architectures were based on a layer system, but systems were not compatible with each other [2, p.1]. This meant that router A could transfer data to router B only when both used the same network architecture, resulting in companies unable to update old hardware and software in addition to not supporting a mix of network routers with different network formats. Subsequently, the International Organization for Standardization (ISO) was

formed in 1977 with the goal to “develop a model and define the protocols and interfaces required to support an open system” for the Internet [2, p.1].

Conducting a program to develop general standards of networking resulted in two different network designs to be merged under the “Open Systems Interconnection (OSI) Model” in 1984. This model forms a more robust and abstract network design. By abstracting the network design, the OSI is able to create seven separate layers within its system which in turn provides a way to connect multiple computers, formats, and hardware.

The seven layers, sometimes called the network stack, divides the work between two main groups. The top group links the applications with the network. The bottom group moves messages from one router to the next within the network. The distinct layers support the development of protocols such as IP, TCP, and HTTP. Specifically, IP and TCP only contain a single layer’s functions in the lower network stack; whereas, HTTP combines a few protocols under one and provides a foundation for custom protocols to be built within its body.

All applications have the responsibility to develop the syntax of the data and format it before sending it through the net [3, p.81]. Data formats like JSON help structure information in a clean, concise way. Syntactically, it makes representation of all data (e.g. objects, strings, null, etc.) simple. Since it handles the syntax issues, developers need only to format the data. In the long term, building protocols that give standardized replies without messy, unidentified information helps with the protocol maintainability and flexibility as design specifications change. Not only does this promote the use of strong software practices, but also ensures the scalability of said protocols. Due to the importance of building a standardized protocol for *Spiral Up*, this paper researches previous protocol design goals in order to promote maintainability, scalability, and longevity for the *Spiral Up* application protocol.