Distributed Computing

Assignment 1: Services-oriented architecture

Master of Computer Science 2018-2019
Miguel Camelo, Pedro Isolani, Steven Latré

Introduction

Social media sites are virtual communities that bring people together to interact and share ideas and interest. Facebook, Twitter, Instagram, Pinterest, and other social networks have been created to target a specific type of public, topics, media content, or a mixture of them, which provide them a unique signature in the way people interact to others and may even impact their real life.

In their initial development states, these social networks were small, and their business logic was simple to maintain following a monolithic architecture. However, once the applications became large, they started suffering from problems such as overloaded web containers, had challenges to perform updates, suffered from limited scalability, and even the development team was forced to have a long-term commitment to a given technology. Therefore, creating, developing, and deploying social network web applications requires the use of flexible design patterns and service delivery architectures such as micro-services.

Project scope

This project aims to build a web site for a new social network that promotes pro-social and sustainable behavior. However, how to design a new social network that influences and stimulates a behavior change? The theories on persuasion and behavior change (e.g., social influence, heuristics, nudging, behavioral economics, etc.) will provide the framework you will need to deal with technology acceptance and adoption and the practice of evidence-based technology design.

In this project, you will work in a multidisciplinary team together with students from the Persuasive Technology class[1], which will be actively involved in designing a habit-forming persuasive technology based on the theories and models discussed in the theoretical lectures during their lessons.

Project specification

The primary goal for this project is therefore to create a mockup social network that allows an administrator and regular users to:

- Register and log in.
- Visualize user's interactions (e.g., posts, messages, comments, etc.)

[1] https://www.uantwerpen.be/popup/opleidingsonderdeel.aspx?catalognr=2100FSWPUT&taal=en&aj=2018)

- Manage user’s relationships (manage and control interactions among friends, followers, and etc.).
- Visualize news feed to allows users to see the content uploaded by others and following the public/private policy and relationships.
- Comment on user’s content.
- Simple one-click valorization of content (e.g., like, recommend, etc.).
- Remove own content.
- Tag users in their own content.
- Follow and unfollow others’ profile & content.
- Receive notifications on the screen. Some examples of actions that may trigger a notification are: receiving new messages, when new content is uploaded by friends, when a component of the application is down, or social-network-depended to trigger user actions.

In addition, a management dashboard should be provided where administrators can:

- Visualize information about the number of users registered, number of posts in the last day, week and month.
- Manage user accounts (e.g., create, block, and remove users).

In app-applications:

- Messaging system: Allow friends (or similar) to chat off-line (e.g., direct messaging) and real-time (chat).
- Personalized advertisement system: Provides advertisements that are targeting specific preferences of the user based on, e.g., words in a post that can trigger a certain ad.
- Cyberbullying analyzer: when a user of the platform writes a post/publication, the system should detect cyberbullying based on content analysis of written text and provide real-time feedback on how offensive social network comments can be perceived. This is similar to how the strength of a password is reported on various sites: as reactions become more and more offensive, subtle cues should be sent to the users. For example, when a user is writing a ‘mean word,’ something happens (like a pop-up: ‘are you sure you want to post this? It can be hurtful) before the user publish it.

Applications

There are one main application and a minimum set of distributed services that need to be considered in this project:

- Main app: The social network application with views for
  - The user (actual site)
  - The administrator (management and dashboard functionality)
- In-app applications:
  - Login and registration system.
  - Messaging system.
  - Advertisement system.
  - Cyberbullying analyzer.

The in-app applications have also extra requirements:

- Login and registration system: Presentation layer that can be accessed by admins to visualize information regarding registered users.
- Messaging system: Presentation layer to allow a stand-alone use of the chat (similar to the messenger on Facebook and Hangouts from google).
- Advertisement system: Presentation layer that allows uploading content (keywords and media content) to the advertisement system.
- Cyberbullying analyzer: Presentation layer that allows uploading content (e.g., keywords/dictionaries) to the cyberbullying analyzer.
- They can be deployed in distributed environments (multiple computers) and be resilient in case of failures. Specifically, if there is a failure (network or application level), then the system should be able to be restored automatically. In the meantime, the user must be notified about it, e.g., if the user is trying to log in and the service is down, then you may show a message saying, “the login system is down, we are working to bring it back as soon as possible.”

Project Assignment

Your team will create a prototype for our new social network following a micro-service-based architecture. You will build this in Flask and the technologies seen in the lab sessions. This prototype must contain the social network and the in-add applications.

The project will be split up in several assignments, starting with creating a Service-Oriented Architecture and ending with a demo of your team prototype by the end of this semester.

Assignment 1: Service-Oriented Architecture

Make a proposal where you and your team describe a solution in a Service-Oriented Architecture, which is compatible with a micro-service-based implementation. Keep in mind that the system is distributed and has to scale to a large number of users. Identify and describe all the services that will compose your applications and organize them into layers (Presentation, Logic, and Storage), if needed. For each service within these applications, describe their purpose and functionality. Also, describe how the applications communicate with each other. Make sure that you define your proposal in a technology neutral way and think about the granularity at which you define the different services.

Deliverable

A document with a high-level service composition diagram for both applications
and a detailed service profile containing the functionality for each service.
Deadline: March 10th, 2019, 23:59 PM via Blackboard.
Feedback: March 15th, 2019, during lab session time.