“SwapStreet” – A Local Community App

Introduction

You are employed by a company called WeRware Pty Ltd. The team who developed a Web application called “SwapStreet” has decided to leave the Company and establish their own startup. This, however, has presented the project manager, Kate Stefanov, with a problem: They left the app half-finished. The release date is looming, only 12 weeks left, and your team has been selected to quickly fix up and improve “SwapStreet” to get it ready for its launch.

Definitions:
Favour: any activity on SwapStreet performed by one member for another member. It is either exchanged for another favour directly or it goes towards future favour exchange(s).
Service: a type of favour, that requires a specialised skill and/or certificate e.g. plumbing.

What is SwapStreet?
SwapStreet is a Web application that allows for people who live locally to swap favours. Members offer to do favours in exchange for other favours within a certain proximity to each other.

The app is aimed for people to get to know their neighbours through those interactions. Thus, not only do they help each other, but also a happier local community is established.

“SwapStreet” is similar to NextDoor that way, but is focused only on exchanging favours without money playing any part in that exchange.

There is no anonymity on SwapStreet; each member has to reveal their identity to all other members and the app has to be able to enforce that and be able to establish a member’s identity.

How does SwapStreet work?
Say, I am good at painting walls. I just moved into a place where the lawn needs to be cut. I neither own a lawnmower, nor do I know how to cut grass.

I become a member of the site. There I can see everyone who is also a member of SwapStreet.
I select all members who have advertised the skill of mowing a lawn and advertised that service. They may need some painting done, in which case, this is a direct exchange.

If not, I could still contact them and they could provide the service to me and that favour could be swapped for another favour or favours they might need now or in the future.

The team had decided to go for a design where each new member is given, say, 20 SwapStreet coins to trade for favours at the beginning. This is their “Welcome to the community gift”.

If the new member choses to use up the coins right away, they could, but they will have to earn more for future favours they might need done.

Favours examples:
Drive someone to the supermarket or to the doctor
Sew a button, undone hem, other more involved sewing tasks
Clean the house at end of lease or on a regular basis
Babysit
Companion for an elderly person

Weed the garden
Paint internal/external walls
Any other favours that might require certificate and the favour provider is willing to do via SwapStreet e.g. electrical or plumbing work (called service).

What are you required to do?
Unfortunately, the team that left was not keen on documentation, so there are no system specification documents that you could use.

Your first task will be to examine the code the left. You are required to write a short report to your manager. The report should be about two pages long, up to 800 words and should contain:

1. A package diagram describing the code base provided
2. An overview on what you have discovered when examining the code – strengths, weaknesses (200-300 words)
3. Describe
    - ➢ One front-end feature that you want to improve; justify why, propose how
    - ➢ One back-end feature that you want to improve; justify why, propose how
    - ➢ One new feature that you believe will improve the app; justify why, propose how it will be implemented (high level)

Some examples of existing features to improve:
- authentication, database, member profiles, comments (endorsements)

Some examples of new features to consider:
- swap tools / kitchen appliances
- casual use of a garden area or a workshop shed
- proof of identity for new members using e.g. of 100 pt system, biometrics
- accept as a member by recommendation from three existing members.

Scope
The team left in a hurry, so no there is no documentation as mentioned before, but you might also discover that the code base requires refactoring/restructuring in view of best software practices.

You might also decide to base the product on other technologies, although getting away from React, on which the current project is based, might require a complete rewrite.

Consider carefully how much you can accomplish before the launch and devise a plan (series of sprints) that will ensure that you have a good quality Web application before the due date.

Disclaimer:
Please note that the code that you write in Comp4050 becomes part of the code base and might be used in future projects.

27 July 2021
Comp4050 S2
Dr Kate Stefanov