Software Engineering (SE) Course CSC 648-848
Spring 2023
Class Team Project

WWW site for restaurant search, order and pickup for SFSU students, staff and faculty

Prof. Dragutin Petkovic, Anthony John Souza,
San Francisco State University

01/27/23

Introduction

This document describes only at a **very high-level** the requirements for software engineering (SE) team project i.e. a WWW-based application that all student teams will develop this semester. Each student team will design and develop its own version of the same web application which is described in this report. This brief high level description of team project serves only to initiate the project and is on purpose somewhat vague (as most often in real life) in order to engage students in the full cycle of SW application development starting from scratch and involving both design and delivery of a real SW application closely simulating the process used in industry today.

Selection and options for SW development and application deployment tools to be used are to be done in Milestone 0.

Final project organization polices and rules of participation, as well as grading rubrics are described in the class slides "about final team project", posted on Canvas. The final team project carries 50% of the class grade. In general, all team members get the same grade for the final project (exceptions will be explained in the class). **Full and maximum focus and participation/attendance to team project and attendance to all team meetings <u>are required</u> from each team member**

Student tams (6-7 students) are chosen based on sel-survey of student skills (NOT used for grading) in order to properly and mix the skills and experience for each team. Each team will have formal *team lead, front and back end leads, github lead and team members*. Each team will communicate with instructors mainly via their team lead.

Class instructors will serve also as CEOs (Petkovic), CTO (Souza) and customer advocates to give feedback on UI (Petkovic).

Application to be developed by each team: WWW site for restaurant search, order and pickup for SFSU students, staff and faculty

Functional specifications e.g. what functions and services are envisioned –

NOTE: *this is initial, vague and incomplete list just to get you started, it can change if you think it would lead to a better product. You will be developing more specific list in your Milestones 1 and 2 and during the class decision, and then revise as necessary (including as per instructor feedback) in the spirit of Agile and User Cantered SE process:*

Here is the story that motivates your team project. Two entrepreneurs, SFSU students (YOU) , decided to create a start-up comprising of a team of about 6-7 SFSU students (your class team) to develop a web-based service for *exclusive* use by SFSU students, staff and faculty to search for local restaurants, order meals, and then either pick them up or request delivery. We assume that students, faculty and staff will be physically located at SFSU or in close dorms where they want to get the food. Take into account reasonable COVID related policies for safe food pickup and delivery.

Minimal functions envisioned include: a) searching and review of nearby restaurants; b) ordering food (physical delivery or pickup at the restaurant); c) ability of restaurants to register for the service and post their advertisement and menu; d) ability of restaurants to manage orders such as delivery instructions for the driver or list of items for pickup. The site will also have administrator who will be required to approve each restaurant which applied to be part of the service before it can go live, as well as to delete inappropriate items or users. Other functions may be added by student teams.

This start-up wants to ultimately win over the stiff competition that already exists in the marketplace. Therefore student teams are required to think of some unique functions to make their site stand out from the competition (hint: how to make it uniquely useful for SFSU students, staff and faculty). In addition, the code for the application has to be well organised and maintainable to allow this start-up to modify it for other customers. This WWW app must be very easy to use and must be created in such a way as to facilitate its usage (e.g. be media rich and provide basic mapping interfaces).

For this class, the application will *focus on desktop/laptop browsers, and NOT on native app development for mobile browsers*. Application will however have to be developed using *responsive UI* implementation so that all or critical subset of its functions renders well on mobile devices.

Important constraints:

- Due to site security issues and limitation of cloud servers where the application will be developed and deployed, we will not allow use of any e-mail clients hence messaging between sellers and buyers will be done via in-site messaging

- To verify that the person who is registering is affiliated with SFSU, it will only be required to verify that e-mail suffix has string ”sfsu.edu” at the end – no PW activation via email nor access to SFSU database to verify real e-mails will be required nor allowed (we simply cannot get this access)

- No payment transactions nor any payment user interface (not even simulated) shall be developed, assumption is that payments are done upon delivery (this is in order not to confuse people who might enter real personal data)

High-level non-functional specifications (how the app is delivered and other constraints) that MUST be adhered to

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. All or selected application functions shall render well on mobile devices
4. Data shall be stored in the database on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected
7. The language used shall be English (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application shall follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used
12. No e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2023. For Demonstration Only"* at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

These non-functional specs are in real life usually provided by clients, CEO, business and legal departments and are NOT subject to change by engineering team on their own. Hence copy them in your milestone documents (and optionally add details as necessary). **You are not allowed to remove any of these non-functional requirements by yourself and must abide by them exactly as they are written (including the text for # 17).**

Data and Content

The website has to be media rich in order to facilitate business and adoption of this service. However, since the projects will be posted for outside viewers, students must ensure that they have legal rights to use all posted content (including logos and background imagery). Students are encouraged to make up their own content.

In providing the digital content for the project and demos, you must adhere to responsible sue of IT policies adopted by CSU https://calstate.policystat.com/policy/10593951/latest/ (this is also why we have the role of admin to approve all content before it is live). We recommend that students create their own media set for the demo or use copyright/licence free resources from the internet

Enough images should be provided for the effective demo such that you can effectively demo search and filtering functions (e.g. 5 or so items per search category). This means total of about 30 items is enough for successful demo.

Restaurant names, addresses, menus: Since this will be viewed by outside users we request that you come up with your own fictitious restaurant names and not real restaurants as to not confuse users and infringe on legal rights of real restaurants. For restaurant addresses pick up some general address (like county parks, government buildings) so they can be displayed on the map and again not to infringe on privacy of other people or businesses. For restaurant menu items, names of items and prices make something reasonable as to make demo attractive and realistic.

This application, related class design documents and project github will also serve as part of your portfolio for jobs each.

Have fun, work well with your team, ask lots of questions, and produce a great product!