**CSC 648-848 SW Engineering**

**Fall 2024**

**Class Team Project initial high level requirements**

**Project to be developed:**

***Tutoring WWW site for and by SFSU students***

Prof. Dragutin Petkovic, Anthony John Souza,

San Francisco State University

07-16-24

**Introduction**

This document describes **only at a very high-level** the requirements for software engineering (SE) student team project i.e. a WWW-based application that all student teams will develop this semester. Each student team will design and develop their 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 modern Agile SW application development which will include many iterations with “user advocate” e.g. Prof. Petkovic. Hence, final designs might deviate a bit from these requirements.

We also require teams to use GenAI tools for tasks of their choosing. To motivate learning and experimentation, this usage will not be graded but must be documented. Specific instructions will be given in each milestone document. (We also posted some related resources in background reading section on Canvas (“Important skills for SW Engineers”) and will talk about this in the class).

Here is the environment we will be immersing ourselves in the class. Two entrepreneurs, SFSU students/alumni, decided to create a start-up comprising of a team of total of 5-7 students to develop a web-based service (which also renders well on mobile) that will allow buy and sell of items for *exclusive* use by SFSU students, staff and faculty (kind of e-Bay but just for SFSU). Teams will therefore model small startup and will receive guidance, user feedback and coaching from class instructor Prof. Petkovic who will play the role of CEO for the start-up assigned by Venture Capitalist (VC) firm considering financing student start-up. Anthony Souza shall be VC-assigned CTO e.g. technical lead advising student teams on technical questions.

Student tams (5-7 students) are chosen based on self-survey of student skills **(NOT used for grading so please be honest in your survey self-assessment)** in order to properly 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*. Team lead will be nominated by instructors (based on skills survey) and other leads will be chosen by the team. Each team will communicate with instructors mainly via their team lead. 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, Milestone 0 carries additional 10% - hence team related work is 60/100.

Selection and options for SW development and application deployment tools to be used are to be done in Milestone 0. All necessary SW and servers and github shall me available with no cost to students (but students must follow proper cloud management procedures so as to not incur unnecessary costs in using cloud services – guidance for this will be explained in the milestone documents).

After M0 the team project will be developed via five formal milestones (M1 – M5). In the last milestone, M5, teams will demo their project to instructors and class and submit their team project documentation for grading. All milestones will be formally presented in the class and will be in the form of explicit instructions which will be posted on Canvas together with the deadline for delivery.

**In general, all team members get the same grade for the final project (exceptions will be explained in the class, and general rubrics for grading are shown in Class Intro slides). Full and maximum focus and participation/attendance/communication to team project and attendance to all team meetings are required from each team member. Failure to observe this will result in reduced team project grade for that student. Team leads must report any lack of team members engagement to instructor asap so we can correct this on time.**

**Application to be developed by each team: *Tutoring WWW site for and by SFSU students***

**High level initial *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 with feedback from instructors, and then revise as necessary (with input from instructors) in the spirit of Agile SE process.*

Minimum functions of the envisioned WWW service must include:

* Browsing, searching, reviewing the tutor information
* Contacting the tutor
* Uploading of tutor information
* Dashboard for the tutors and students to show postings and messages received
* Functions for site administrator who will be required to approve each uploaded tutoring item info before it can go live, as well as to delete inappropriate items or users.
* Each team **must** develop, in addition to standard functions, some specific functions unique to SFSU in order to better compete with existing services

Developed WWW site will be exclusive to SFSU students: both tutors and student asking for tutoring must be SFSU registered students (e.g. have valid SFSU e-mail).

All functions delivered must be easy to use, free of bugs, and work on two bowsers and render well on mobile.

In addition, the code for the application has to be well organized and maintainable to allow this start-up to modify it for other customers like other CSU campuses.

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 that must be followed up:**

* 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 (e.g. use message window to enter message text which is stored in the database and accessed via dashboard of the recipient)
* To verify that the person who is registering is affiliated with SFSU, it will only be required to verify that e-mail of registering person has suffix 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 know this is not safe but we simply do it this way since we cannot get access to SFSU mailing list.
* No payment transactions nor any payment user interface (not even simulated) shall be developed, assumption is that payments are separately negotiated and executed upon delivery of service (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 (no native app to be developed)
4. Posting of tutor information and messaging to tutors shall be limited only to SFSU students
5. Critical data shall be stored in the database on the team’s deployment server.
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application shall follow established architecture patterns
11. Application code and its repository shall be easy to inspect and maintain
12. Google analytics shall be used
13. 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
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the class) for main data items
16. Media formats shall be standard as used in the market today
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools
18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Fall 2024. 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 department and are NOT subject to change by engineering team on their own – **all of them must be followed precisely and completely.** **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 # 18).**

**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 use 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).**

**Your data items like text, images and videos !!!have to abide by all rules regarding ethical digital content. Recall, this application becomes your portfolio for job search so make professionally looking demo with professional and realistic content (please to avoid jokes and funny stuff here!!!). We recommend that students create their own media set for the demo or use copyright/license free resources from the internet.**

Just enough images/data items 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).

**Personal or entity names, addresses etc.:** Since this project can be viewed by outside users we request that you come up with your own fictitious person or entity names and not real ones as to not confuse users and infringe on legal rights and privacy. For addresses (if necessary in your app) please 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.

**This application, related class design documents and project github will also serve as part of your portfolio for jobs each. We will cover job search and how to leverage your class for your portfolio in last class.**

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