

# CSC 648-848 Fall 2024 Milestone 2

## Part I: More Detailed Requirements, Specs, Architecture, UI mock-ups

## Part II: vertical SW prototype

10-05-24

**Announce: TBD**

**Due: Check instructor's e-mail (deadlines for M2 document Part I are different from deadlines for Vertical Prototype Part II)**

### Objective

Goals of two part Milestone 2 are:

- **Part I: more detailed requirements and design with UI mockups – specifically:**
  - More details for data description;
  - Refresh/update and prioritize requirements;
  - Design high level UI mockups and storyboards;
  - Design high level DB organization (logical level).
- **Part II: Vertical SW Prototype:** Develop first simple bare-bones SW prototype (from DB to browser) to test the infrastructure, educate the team (start coding!), resolve technical issues and also serve as basis of further development (must run on the deployment server).

Use of genAI tools is mandatory (for tasks of your choice, not graded, like in Milestone 1)

**Milestone 2 delivery** hence consists of two parts with separate deadlines:

- **For Part I - Milestone 2 (M2) document** (one per team, submitted similarly like Milestone 1)
- **For Part II - Vertical SW prototype (VP)** to be reviewed by Class CTO, submitted on-line with its URL and additional information

Milestone 2 Part I document e.g. designs have to be reasonably consistent with Milestone 1 and instructors' feedback *but it can also differ* from Milestone 1 based on what you discover and develop in your design process in the spirit of iterative SE process and

based on the feedback you get. (Modern Agile SW development calls for continuous iterations until final commitment)

**Milestone 2 differences from Milestone 1 DO NOT need to be edited in Milestone 1 doc which remains frozen and fixed after your revision as per instructors' feedback on Milestone 1.**

**You should start with Milestone 2 Part I only after you have incorporated instructors' feedback on Milestone 1. You can start Part II of M2 any time your back end team is ready (Part II should in general be done by back end team in parallel with other work on Milestone 1).**

Milestone 2 document is a separate document from Milestone 1 document, see submission instructions below.

Store all milestone documents in your team's github folder "Milestones".

**Team leads: please make sure all team members read and follow M2 document before submission for comments, and after submission and feedback for follow up in their work.**

**Note that up to 3 negative points will be given if instructor determines that quality of submission indicated the team has not done proper study and preparation for the delivery. To help, we list MUST readings for Milestone 2.**

In slides "Personae, Use Cases and Requirements" on Canvas

Use cases:

Functional requirements:

Data dictionary:

Priorities:

UX mockups and storyboards:

In slides "SW Architecture Patterns and Search Arch."

Check section on "Some Specific related to team project" - covers how to organize search, media data etc.

In "User Centered Design, Usability, GUI Design and testing" slides with more details on mockups and storyboards with a lot of examples

In slides "Ideas for UX design for team project: examples of some good home pages"

Check examples of good home pages to inform your home page designs

For use of genAI tools check class slides on GenAI for SE

## **Milestone 2 Part I instructions**

Use as much space as you need, but the expected length is about 20-25 pages. You must submit all sections.

Please do follow all instructions below

Milestone 2 Part I document is one PDF file (named as per submission instructions below) with *title* page and *body* as below.

### **Title page**

**Title page of Milestone 2 document must include**

- “SW Engineering CSC648/848 Fall 2024”  
Project/application title and name (you can use the name you chose for your application)
- Team number and name – make it clearly displayed for easy reference
- Names of students (team lead first) -  
Name of team lead and their e-mail
- “Milestone 2 Part I”
- Date
- History table (as in M1 – two key items: date submitted for review, date revised after feedback)

Milestone 2 Part I document body must have the following sections, in the order and titles as below:

**1.Executive summary** – simply copy it from revised Milestone 1. Please come up with product name by now. Make sure all required revisions requested in Milestone 1 for executive summary feedback are incorporated.

### **2. List of main data items and entities (expand as necessary)**

First, make sure all required revisions requested in Milestone 1 feedback are incorporated in Milestone 1 (M1) document. Then expand as necessary data description and entities from M1. Add full definition/list of sub items, but remain at logical level only (e.g. list item name and description and values)

Recall, focus here is on data not functions. Focus on proper and intuitive naming AND make sure you list all data sub-items you know at this time (e.g. for reg. record and main item/listing). Say if data sub item is mandatory or optional (e.g. video)

Be sure to cover data items referring to your specific SFSU functionality. Check class slides for guidance.

In general main entities reported here like reg. users, main data e.g. sales items, categories, user reg. record, messages drive the design of the DB

**The naming you define here MUST be used consistently in the rest of documentation as well as in code, DB tables etc.**

### **3. Functional Requirements - prioritized**

First, make sure all required revisions requested in Milestone 1 feedback are incorporated in M1 document.

Then edit/modify functional requirements from revised Milestone 1 into Milestone 2, with more details only if necessary. These should still be only high level requirements. Keep the same reference numbers with respect to Milestone 1 (i.e. if high level requirement was number 3 in Milestone 1, then Milestone 2 more detailed requirements of requirement 3 are 3.1, 3.2 etc.). OK to add new or delete previous functional requirements from Milestone 1, if you can justify it. You must address all your specific SFSU functionalities in requirements to ensure they will be implemented.

Then prioritize each requirement with Priority 1, 2, 3. (1-*must have*; 2 – *desired*; 3 – *opportunistic* as defined in the class). To develop these priorities think of the user, use cases, making your application complete from usability, marketing and business aspects AND your ability to design, implement and test them. Base this also on your skills, resources and schedules. Instructors will check final priorities. The priorities you set now may change a bit but will be frozen in Milestone 4 which will constitute your commitment (especially priorities of 1). At least one SFSU specific requirement must be Priority 1.

In terms of presentation, for easier review, please group all requirements first by priority i.e. list all Priority 1 requirements first, then Priority 2 etc. and then within each priority section you should group the requirements by types of users (for example first by unregistered users, then by registered users, and then by admin etc.). See below:

- Priority 1
  - Unreg user
  - Reg user
  - Admin
  - Any other user (restaurant owner, driver)
- Priority 2
  - Unreg user
  - Reg user

- Admin
- Other users
- Priority 3
  - Unreg user
  - Reg user
  - Admin
  - Other users

**4. UI Storyboards for each main use case (low-fidelity B&W wire diagrams only).**  
**NOTE: your design is only for browser UX not mobile**

- Create storyboards (set of mockups depicting user flow of UI screens) for all major use cases from M1 (e.g. 3-6 major use cases).
- NO need for mockups for admin UI since you will use WorkBench for that. In the document just say that in 1-2 lines of text.
- In terms of presentation of storyboards please place only one to two mockups per page (and so on until you complete the storyboard) so we can easily read it and comment. Mockups shall be black and white wire diagrams focusing on basic layout and description of the functions in each main area of the GUI. This helps focus on testing the navigation and flow, they key in this phase.
- At the beginning of each storyboard list related use case so the reader knows the context. If you need to annotate mockups/storyboards to make it easier for reader to follow, use annotation text in *italics*.
- Must address your SFSU specific functions in storyboards/mockups
- Need also to show login and registration mockups
- As said in non-functional req. no UX nor any payment functionality shall be done
- The format for UI mockups is very flexible but we are OK with hand drawings, which you can scan and include in final Milestone 2 document. You can also use some tools for mockup design like Figma. Do not use colors yet (unless absolutely necessary), it draws attention from basic UI concepts (functions, behaviors, layouts, flow...). Make sure your scans and handwriting in final PDF document are legible.
- Test your storyboards as follows: walk through your storyboards following each major use case and make sure your storyboards work well. This test should be done by those NOT involved in UX like back end team, use their feedback to revise before submitting.
- Use data terms and names consistently with section 2 data naming

For your design please consult class slides on UX/GUI/Usability, related products (e.g. from your comparative study in Milestone 1) and pay special attention to home page design (we posted some examples of good home pages).

We recommend that the front-end team be assigned to this task.

**5. High level Architecture, Database Organization summary only (NOTE: you must follow MVC design pattern) - start this as soon as work on Milestone 2 commences**

- *DB organization*: Describe the main database schema/organization (high level), e.g. only list main DB tables (e.g. their titles) and items (columns) in each DB table (check instructors' suggestions and class slides on architecture). This should follow data definitions described in Section 2. Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 2 above. No need to make ER diagrams, just list each table name and its elements (columns). Please consult architecture class slides with CTO advice on this
- *Media storage*: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on whether to implement file vs. BLOBs must be made by the end of M2. Check Architecture slides Part II). Describe any other special data format requirements like for video/audio/GPS etc.
- *Search/filter architecture and implementation*: explain what will be the alg/SW for search; how will you organize search items for the user; what DB terms will be searched, how it will be coded and organized in the DB (check instructors' suggestions in the class – architecture slides). Preferred option is to use SQL and *%like* and if you plan to use that simply say so (as in architecture class slides). If you propose to use anything else (e.g. other SW/API) mention it and it will have to be OKed by instructors. (This is described in arch slides Part II). Say which data items will be searched using *%like*.
- Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.)
- If you have changed SW tools and frameworks or added any new one please describe it and say that it is new. Any new SW or framework you will be using has to be approved by CTO in writing by this time.

This section should be assigned to back end team who can complete it while working on Milestone 2 Part II Vertical Prototype and getting some experience what works. Feel free to review and leverage template prototypes mentioned in the class and in Vertical Prototype section below

**6. Identify actual key risks for your project at this time**

Identify *only actual and specific* risks in your current work that you are aware of such as:

- *skills* risks (do you have the right skills),
- *schedule* risks (can you make it given what you committed and the resources),
- *technical* risks (any technical unknowns to solve),
- *teamwork* risks (any issues related to teamwork);
- *legal/content* risks (can you obtain content/SW you need legally with proper licensing, copyright).

For each identified risk tell us then how you plan to resolve it. The key is to resolve risks as soon as possible. Categorizing risk as above helps a lot in managing them. Be brief: identify only the actual risks you think you have at this point and explain (2-3 lines), then list how you will address these (2-3 lines)

## **7. Project management**

In this section please describe in no more than half a page how you managed and plan to manage M2 and future tasks and what tools you will use.

Milestone 2 is the time you will have to manage number of tasks. It is critical to always assign all tasks, and for each task know the person in charge and the deadline. Then follow up (often with checkpoint in mid point before delivery).

Also, Milestone 2 is a good time to make sure front-end and back-end team operate more independently while also agreeing on common interfaces. This makes the team more efficient.

It is a good idea to start using some simple tools to manage your tasks such as Trello <https://trello.com/> or similar tools for task management which offer unified dashboard view of all tasks and status for easier management and tracking.

## **8. Use of genAI tools like ChatGPT and copilot (mandatory) –focus only on usage in Milestone 2**

Please describe the following, in max 2 pages

- What genAI tool and version you used
- List tasks for which you used genAI tools and for each rate how useful it was, use LOW, MEDIUM, HIGH
- For each task above explain briefly how you used the tool and what benefit it offered.
- Provide examples of key examples and prompts
- Comment on anything else you found useful

This section is NOT used for grading but for learning and class discussion. You must follow class genAI policies and academic honor policies re: plagiarism. Be sure to always check and verify genAI output, YOU are in charge and responsible.

Note: genAI can be used for variety of tasks: improved writing, requirements (but likely note those SFSU specific), DB design, UX design, as well as prototyping code in M2. Part II. Try it for all tasks and get some experience, we do not grade it (but must document it). Check posted slides on ChatGPT and SE second part.

**9. Team Lead Checklist:** for each item below team lead must answer with only one of the following: **DONE/OK**; or **ON TRACK** (meaning it will be done on time, and no issues perceived); or **ISSUE** (you have some problems, and then define what is the problem with 1-3 lines)

- So far all team members are fully engaged and attending team sessions when required
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
- Team reviewed suggested resources before drafting Milestone 2
- Team lead checked Milestone 2 document for quality, completeness, formatting and compliance with instructions before the submission
- Team lead ensured that all team members read the final Milestone 2 document and agree/understand it before submission
- Team shared and discussed experience with genAI tools among themselves

### **Submission of Milestone 2 Part I Document for Review**

Submission must be done by the deadline specified; any extension has to be approved at minimum 24 h ahead of time.

In creating, editing and finalizing Milestone 2 document follow similar team process as outlined for Milestone 1 document. You can use any tool of your choice for creation and managing the M2 document (e.g. google docs, Figma) but you are required to put the final submission version of M2 document as one PDF file in github “milestone” folder. (Make sure to save also DOC version so you can edit it later). This way you will have code and documentation in one place – good for easier access and as your job search portfolio. You must follow the structure of Milestone 2 Part I document as outlined above, with no missing sections

**Submission:** Each student team submits one milestone document for Milestones 2, as follows (similar as M1 submission):

- Team leads will send e-mail with a link (NOT the attached file) pointing directly to Milestone 2 Document posted on github Milestone folder to [Petkovic@sfsu.edu](mailto:Petkovic@sfsu.edu) and Anthony [ajsouza@sfsu.edu](mailto:ajsouza@sfsu.edu) with the subject line as specified below. This link



MUST point directly to M2 file in the team group directory “Milestones” on github and not to the folder.

- **e-mail subject line:** Must be “CSC648-848 Fall 2024 Milestone 2 Document Team N” in the subject line (N is a team number (01, 02 etc.).
- **e-mail body** contains direct link to Milestone 2 document in team github . File name of the M2 document to which the link is pointing to MUST be: “CSC648-848 Fall 2024 Milestone 2 TeamN.PDF” (N is your team number). File format is PDF.

**Please follow all instructions and requirements in this document!**

### **Instructor’s Feedback, and Freezing the Milestone 2 Part I Document for Final Project Delivery**

As Milestone 1, Milestone 2 Part I is not graded upon submission, except in some exceptions, see below. Revised M2 after instructors’ comments will be graded as Milestone 1, when final Milestone 5 folder with all the documents is submitted at the end of the class.

After delivery of the Milestone 2 Part I document, you will get feedback from the instructors (D. Petkovic) by e-mail, similar as for Milestone 1. This feedback must be used to revise your Milestone 2 Part I and used subsequently for the rest of the project. If you have any questions or need clarification send e-mail to Prof. Petkovic or discuss in class teamwork session. Please enter the date of revision summary in history table. (This is similar to Milestone 1 review process).

After this revision you will freeze the Milestone 2 Part I document, place it in github “Milestone” folder and use it for final project document delivery. This frozen document does not need to be changed even in future designs change.

Team leads: please CC yourself (and the rest of the team) with submission e-mail – that is a good practice so you know what and when has been submitted AND it ensures your e-mail is not stuck somewhere (in which case you will not get CC e-mail...). **Excuses based on “my e-mail was sent on time but was stuck on the server” will NOT be accepted.**

In order to encourage iterations and dealing with feedback, we will not grade the first submitted M2 Part I version. We will only grade M2 document after it has been modified wrt. instructors’ feedback (see above) and when it is submitted with final project at the end of the class in “Milestone 5 folder” – it will be part of team’s *SE Process* grade.

**Exceptions to this “non-grading” method are below, and will result in recording of negative points which will be applied to final team *SE Process* grade:**

- M2 document Part I improperly submitted (e.g. not following ALL required submission rules). It will also be returned for re-submission
- M2 document Part I does not have all required sections
- M2 document Part I submitted late in case delay was not asked for by e-mail from team lead to [petkovic@sfsu.edu](mailto:petkovic@sfsu.edu) at least 24 h before the deadline
- M2 document Part I is of such poor quality (beyond simple design and language issues) indicating that team has not consulted class slides and milestone instructions

## **Milestone 2 Part II: Vertical SW Prototype (VP)**

In addition to the **Milestone 2 document**, (**and working in parallel**) the team will create an initial “**vertical SW prototype**” (VP) to test the infrastructure and chosen frameworks and to jumpstart the coding effort. **Time to start coding!!!!!!!!!! Start this work immediatly after completing Milestone 1.**

The purpose of vertical prototype is:

- To early and quickly test basic SW components and deployment infrastructure and frameworks as well as the key architecture patterns
- To serve as a start of the final app development e.g. to be a basic “scaffolding” for final delivery.
- It also serves as “teaching and training” tool to bring the rest of the team up to speed on SW, frameworks etc. (To help this document VP well and present it to the rest of the team)
- Helps you learn to use genAI tools for implementation and coding

**Start VP work as soon as you can and delegate it to back-end team, do not wait for M2 document part I is done.**

The vertical prototype is the form of a code that exercises full deployment stack on the server, from browser (with simple VP *test home page*), via middleware, to DB and back, using only your chosen and approved frameworks and SW components. It has to be deployed from team account on your chosen deployment server, **the same way the final product will be deployed.** In github we recommend that you have a branch dedicated for this.

**Note: now you must install DB, check Arch class slides, part with recommendations for team project, at the end of the slides. See also more on this below.**

We recommend that back-end team be assigned the task of constructing this vertical prototype, with front end team helping with front WWW page.

Vertical prototype shall allow one to enter a search term on a *VP test home page* (simple home page used only to test vertical prototype), then get a response from the DB and render it back on the browser in a simple *VP result page*. Search function refers here to search which is to be used in your final application home page.

- Overall architecture of VP must follow MVC design pattern – check class slides on SW architectures Part I and also our VP SW tutorials, see below.
- **UI for the *VP test home page* is a throw-away test one, and these below are the requirements and design instructions you must follow:**
  - Title including Class, semester, year, team number
  - Search UI shall consist of two main components below, modeled after Amazon search:
    - One pull down menu referring to CATEGORIES. Have say 3 entries here like electronics, books, furniture, and default entry. This component of search is exercised using simple SQL filter – see architecture slides from the class (Part II)
    - ANDed with above is one free TEXT ENTRY field (this text is then used in SQL %LIKE search on text string obtained by concatenating item title and description a values from the DB – check architecture slides Part II)
  - Search arguments (selection of menu and typed text) must remain persistent after search is executed
  - In upper left corner, below search UI and above listed results, show number of items found (check how Amazon is doing this)
  - If user does not enter any search parameters search results must show ALL items in the database (this is good for tester to know what is the content of the database).
- *The VP result page* needs to display search results e.g. list of items including title, price, description and image in *any reasonable layout*. Best it to find the way to group all components belonging to one search item using e.g. some frame or horizontal line
- Develop VP test home page and VP Results page using only the chosen front end framework, do not hard code. This way you can practice and test the interface between front end and back end with the frameworks you intend to use for full development
- The DB can have only a few items/rows (say 2-3 for each category). The items in the DB shall be encoded with full schema as it is defined by now in M2 document. Initialize and populate the DB manually using WorkBench, no need to develop item POST at this time.
- No need to develop login and registration for VP – you will do it later, focus now on key VP functions as described above.

VP offers huge benefits. Besides “testing the pipes”, early VP serves also to help the rest of the team get “on the same page” in terms of SW development. Back end team should also document vertical prototype code well and use it to educate the rest of the team on how to develop the rest of the product. Front end team can use the test home page to establish rules for CSS and UI development. Back end and front end teams should also agree on common way to connect UI with back end and document it for all.

You must use only selected and approved tools and frameworks for vertical prototype and deploy it on your chosen deployment server.

Be sure to test VP well before submission and make sure URL is accessible to instructors (try it on some other and not your browser)

We also strongly recommend that team agree on coding style and stick to it, AND use class/object/variable/DB table names consistent with data dictionary in M1 and M2 (see coding style slides posted on iLearn)

### **Our tutorials related to VP SW architecture and development**

- Tutorial with nodejs, developed by our former student and TA Nicholas Stepanov
  - <https://medium.com/@nicholasstepanov/search-your-server-side-mysql-database-from-node-js-website-400cd68049fa>
- Tutorial with flask, developed by our SE instructor Jose Ortiz
  - <https://medium.com/@joseortizcosta/search-utility-with-flask-and-mysql-60bb8ee83dad>
- Tutorial with PHP, by Jose Ortiz
  - <https://medium.com/@joseortizcosta/search-utility-with-php-and-mysql-as-backend-server-technologies-d3dac5128d8>
- How to use and leverage this:
  - Study code to learn
  - Customize for your app; deeply and test then put on master branch
  - Document well, establish good APIs
  - Use as templates/architecture to guide each team member

Please also check class slides on Architecture especially slides in Part II which give advice for the team project.

We strongly encourage you to use genAI tools for VP but check class slides on genAI. It is strongly advisable that YOU design the overall architecture (follow MVC design pattern) and the genAI for smaller components. You can also use genAI to analyze and evaluate your code. Be sure to always test and inspect genAI code, ultimately YOU are in charge and responsible

### **Organizing the DB for VP and beyond**

Obviously, now is the time to install and set up DB, see below

It is NOT a requirement to have your database and application on different servers → make it simple and put them both on one server

#### Main options

- Main production DB and local DB for each member
  - Each member deploys and maintains their own localhost database (e.g. on their laptops) and the team has one main production DB for your application.
  - However, keeping these databases consistent is going to be a challenging task
- Main/production DB and Test/development DB on deployment server, team accesses them (preferred and simpler)
  - Run 2 databases (under the same DBMS software) on the remote server: production DB and a Test/development DB.
  - Both should be maintained by the same team member.
  - The test and production database should be as similar as possible.

### **Milestone 2 Part II Vertical SW Prototype delivery and submission (different deadline from Milestone 2 Part I document)**

Your team will submit vertical prototype (e.g. let instructors know your VP is ready for review) via e-mail to class CTO Anthony cc Petkovic by its own deadline (different deadline from M2 Part I document deadline). Instructions for e-mail submission are in Appendix I

**The submission format and process** must be followed precisely, as always. Submission must be done by the deadline specified; any extension has to be approved 24 h ahead of time.

- **e-mail subject line:** Must be “CSC648-848 Fall 2024 Milestone2 Vertical Prototype Team N” in the subject line (N is a team number (01, 02 etc.).
- **e-mail body** - see Appendix I below.

### **Instructors’ feedback on vertical prototype (M2 Part II) and your responsibilities**

Vertical SW prototype will be reviewed off-line by class CTO and you will get the feedback which you must analyze and incorporate as necessary. Then you can use this code as a basis for developing the full app.

Vertical prototype shall be evaluated (but not graded) by class CTO based on:

- Functionality and correct search and results display (be sure to test before sending it)
- Code organization and architecture
- Proper use of frameworks
- Correct deployment on a chosen team server for final delivery
- Proper information required as per Appendix I

**Please test your VP for bugs and ensure instructors can access it.**

**Once you get CTO evaluations and suggestions you must follow up on all of them and/or ask questions.**

We will not grade vertical prototype but you must follow up feedback from instructors and revise accordingly after submission. Final grading of the SW code and architecture will be done upon final submission. At that time we will check that feedback given in M2 and later has been incorporated or negative points will be given.

**Exceptions to this “non-grading” method are below, and will result in recording of negative points which will be applied to final team *SE Process* grade:**

- Submission process not followed
- Provided URL does not work (e.g. instructors cannot access the app)
- Application has severe bugs that prevent evaluation (e.g. search does not work, results do not show...)

Don't forget, after revising and testing VP it should be well commented and presented to the rest of the team so they know how to integrate and code their own works, team leads please ensure that is done. Tag it and properly save it in github.

## Appendix I – Instructions for e-mail for M2 Part II Vertical Prototype delivery

When Milestone 2 Part II of the Vertical Prototype is done (and well tested by you), please email class CEO and class CTO:

- **e-mail subject line:** Must be “CSC648-848 Fall 2024 Milestone2 Vertical Prototype Team N” in the subject line (N is a team number (01, 02 etc.).
- **e-mail body:** For e-mail body please use the following template:

Hello [who the email is addressed to],

[Email greeting goes here]

[Link to webapp’s search page]

[Link to GitHub Repo of the file that contains search code on backend]

We [have | have not] populated our credentials folder with most current and correct information. (Any teams who submit M2-VP-Part 2 and your team’s server cannot be accessed will not be graded and the email will be returned and negative points might be issued.)

**Use of genAI tools:** [as for Milestone 2 Part I document please explain for what tasks you used it in this VP work and then how helpful it was, in a similar format to what you did for M2 Part I report and for M1. Max 2 pages]

Best,  
Team NN