CSC 648-848 Spring 2023 Milestone 2
Part I: More Detailed Requirements, Specs, Architecture, UI mock-ups and
Part II: vertical SW prototype

03-09-23

Announce: TBD
Due: Check instructor's e-mail (deadlines for M2 document Oart 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 – add*** more details for data description; 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 prototype (from DB to browser) to test the infrastructure, educate the team, resolve technical issues and also serve as basis of further development (must ruin on the deployment server).

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** (one per team) to be reviewed by Class CTO (off-line)

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

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 iun general be done by back end team).

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".

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 Spring 2023"
  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 his/her e-mail
–"Milestone 2"
–Date
–History table (as in M1 – two key items: date submitted for review, date revised after feedback)

Milestone 2 document body must have the following sections, in the order as below:

1.Exceutive summary – simply copy it from revised Milestone 1 . Try to 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.

Then expand as necessary data description and entities from M1. Add full definition/list of sub items, but remain at logical level only

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

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

3. Functional Requirements - prioritized

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

Then expand requirements as necessary or repeat 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.). *Be sure to cover unique features of your product*. OK to add new or delete previous functional requirements from Milestone 1, if you can justify it.

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 bot but will be frozen in Milestone 4 which will constitute your commitment (especially priorities of 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)

- Create storyboards (mockups depicting user flow of UI) for all major use cases from M1 (e.g. 4-7 major use cases). Place only one to two mockups per page 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 for particular use case have a short version of that use case so the reader knows what is being done. If you need to annotate storyboards to make it easier for reader to follow, use text in *italics*.
- The format for UI mockups is very flexible but we recommend hand drawings, which you can scan and include in final Milestone 2 document. 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 are legible. You can also use some tools for mockup design like Figma
- Before submitting, test your mockups as follows: walk through your mockups following each major use case and make sure your storyboards work well.
- Use data terms and names consistently with section 2 data naming and with use cases.

For your design please consult class slides on GUI/Usability, related products (e.g. from your completive study in Milestone 1) and pay special attention to home page (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

- *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). Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 1 above. No need to make ER diagrams, just list each table name and its elements (columns).
- *Media storage*: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on file vs. BLOBs must be made by the end of M2). 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.
- 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 given 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 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).

Tell us then how you plan to resolve each actual risk you have. 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 person in charge and the deadline. 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.

Submission of *Milestone 2 Document* for Review

Structure and section instructions for Milestone 2 document must be followed precisely, as outlined below. 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.

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 to Petkovic@sfsu.edu and Anthony 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.

- **e-mail subject line:** Must be "CSC648-848 Spring 2023 Milestone2 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 Spring 2023 Milestone 2 Team*N*.PDF" (*N* is your team number). File format is PDF.

Instructor's Feedback, and Freezing the Milestone 2 *Document* for Final Project Delivery

After delivery of the Milestone 2 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 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 revision summary in history table. (This is similar to Milestone 1 review process).

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

***** All submission instructions must be followed precisely or negative points will be given when final documents are being reviewed and graded (Milestone 5), Negative points will also be given for any missing section of M2 document or for not following major feedback ******

In final grading, using Milestone 5 document, instructor will check Milestone 2 to make sure it reflects updates required for Milestone 1 such as in executive summary, data and functional requirements sections.

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.

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)

Suggestion: Start VP work as soon as you can and delegate it to back end tram.

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.

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 to test vertical prototype), then get a response form 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 (e.g. for renting site search of houses/apartment, for shopping search for products, for restaurants search for restaurants etc.)

- UI for *the VP test home page* is a throw-away test one, can be a simple one containing
  - Title as Class, semester, year, team number
  - One free text entry field (this text is then used in SQL %LIKE search – check architecture slides)
  - One pull down for 2-3 search parameters (like main search category like product types; housing type, restaurant food type etc.) – this is exercised using simple SQL filter – see architecture slides from the class
- *The VP result page* needs to display search results including images and later their thumbnails as well as maps (if applicable) in *any reasonable layout* (the goal here is to make sure you can access and display items and NOT the ultimate UI). Thumbnails need to be computed automatically using some open source thumbnail generation SW – you can do thumbnails later after you figure out the basic way to show original image (see architecture class slides)
- Develop VP test home page and VP Results page using 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 (5-7, with variety of data so search can be tested) The items in the DB shall be encoded with full schema as it is defined by now in M2 document. Initialize DB manually using WorkBench.
- Make sure that if user hits "search" or related button, and does not enter any parameters, you display full list of items from the DB – it is necessary so people who test it know what is in the DB
- No need to develop login and registration for VP – you will do it later, focus now on key VP functions as described above

VP has huge benefits. Besides "testing the pipes" early (this is critical), 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 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)

Some resources 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 at the end which give advice for team project

Organizing the DB for VP and beyond

If you have not done so yet 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

<u>Main options</u>

- 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 a similar as possible.

M2 Part II Vertical SW Prototype delivery/submission (different deadline from Part I M2 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 one from M2 document deadline). 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 Spring 2023 Milestone2 Vertical Prototype Team N" in the subject line (N is a team number (01, 02 etc.).

- **e-mail body** contains the form with your data, **as in Appendix I below**.

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

Instructors' feedback on vertical prototype 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 fill app.

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 bene incorporated or negative points will be given.

However, any issues like sever bugs that prevent M2 evaluation, or incorrect M2 submission process will be considered incomplete delivery and be noted under team grade rubric of SE Process.

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

Appendix I – form to be filled out and included in e-mail body for M2 Part II Vertical Prototype delivery

Milestone 2 Vertical Prototype Form
Spring 2023 Team Number <N>

The below form is used for submission of needed information for the current milestone. The below table is used to help access certain parts of your web application. Please make sure the information submitted is accurate and up to date to ensure grading of the milestone is completed in an efficient manner.

Item	Credentials
Website URL	
Website URL to search page	
SSH URL	
SSH Username	
SSH Password/Key	
Database URL	
Database Username	
Database Password	
Link to GitHub page that performs the Search	

Addition information, anything to help with executing or grading of the current milestone
Please explain what is working for your prototype and what can be tested. Also, what search terms to use. Please make sure to explain how to use certain pieces of the log in information. Especially if its more than a simple SSH command. If you server requires a Key please make sure to include it with the submission of the document, regardless if it was submitted with prior milestones. If you are worried about sending keys though emails , you can use links to google drive or drobox etc. Once key has been retrieved you can delete the link.