**CSC 648-848 2020 Milestone 2: a) Architecture, UI mock-ups & GUI design and b) vertical SW prototype**

**Announce: Oct 6**

**Due: Oct 20 midnight (Document), Oct 27 midnight (Vertical SW prototype)**

Objective

**Goals** of Milestone 2 are:

* ***SW architecture design - Milestone 2 document***: design high level UI mockups (e.g. mockups and storyboards) & GUI design; design high level system architecture with DB organization, and practice UML
* ***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 run on the deployment server).

Milestone 2 delivery hence consists of two parts:

* **Milestone 2 document** (one per team, submitted similarly like Milestone 1)
* **Vertical SW prototype** (one per team) reviewed (off-line) - URL and access to code submitted via e-mail to TA and the instructor.

**Content and Structure for Milestone 2 Document for Instructors’ Review**

The sections you must have in Milestone 2 document are as follows, in the order below::

**Title page** (see below), followed by

1. Data Definitions V2

This should be reasonably consistent with Milestone 1 but should be expanded as needed and refined as per instructors’ feedback. Major data items that comprise of sub-data items have to be defined in full (list all its sub-data items, and for images/video list formats, max size etc.). **You must use all the data definitions and names consistently in all documents and SW, including UI text, naming for main variables, classes and database elements etc.** Focus on data items unique and important to your implementation. Be sure to cover ALL items critical to your project and especially those providing a competitive advantage. At this stage data describing user privileges, registration info and main info (raw data, metadata, supporting data) have to be fully defined (as much as it is possible at this stage)

# 2. Functional Requirements V2

Expand functional requirements from Milestone 1 into Milestone 2, with more details as necessary. 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.).

Prioritize each requirement/spec with 1, 2, 3. (1-*must have*; 2 – *desired*; 3 – *opportunistic* as defined in the class). To develop these priorities on behalf of the user, and making your application complete from usability, marketing and business aspects. Base this also on your skills, resources and schedules. Instructors will check final priorities. The priorities you set later in Milestone 3 and 4 will constitute your commitment (especially priorities of 1).

In terms of presentation, for easier review, please categorize all requirements and prioritize them inside the category by listing **first by priority** e.g. Priority 1 requirements first, then Priority 2 etc.

For priority 1 requirements, you need to improve in showing the technical feasibility of functional requirements, upon the feedback by the instructors from Milestone 1.

3. UI Mockups and Storyboards (high level only)

* Create mockups/storyboards for all major functional requirements (e.g. Priority 1 requirements). Start with black and white wire diagrams focusing on basic layout and description of the functions in each main area. Create simple “storyboards” (sequence of mockups) organized by user stories and functional requirements. This helps test the navigation and flow. For each storyboard repeat a short version of related functional requirement so the reader knows what is being done.
* The format for UI mockups is very flexible. Do not use graphics or colors yet (unless absolutely necessary), it draws attention from basic UI concepts (functions, behaviors, layouts, flow…).
* Design GUI design for major functional requirements. (For 5 different screens of UI mockup you generated from the above, do GUI design.)
* Review your mockups and GUI designs per UX principles which we addressed in the class. (refer to class slides of UX) and document it.

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

# 4. High level Architecture, Database Organization

* *DB organization*: Describe the main database schema/organization (high level), e.g. list main DB tables (e.g. their titles) and items in each DB table. Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 1 above.
* *Search/filter architecture and implementation*:
* *In regards to the functional requirements, please specify which operations should be processed on DB (e*.g. what DB terms will be searched, and displayed)
* Your own APIs (if any): Describe and define at high level any major APIs that you will create among modules.
* Describe how the below technical challenges will be addressed and architected.
  + How to search the exercise plan for a particular patient for a given duration (for a specific date or for a specific week),
  + How to search the progress (video, text) uploaded by a particular patient,
  + How to search all the activities of a PT (responding to questions, viewing videos, reading profile, writing notes) to a particular patient. Each activity should be displayed with time and duration (e.g.. Oct 2 10:00 ~ Oct 2 10:05, 5 mins)
* If you are using the 3rd party open source solution, please describe them in your architecture.
* If you have changed SW tools and frameworks or added any new one please describe it.

# 5. High Level UML Diagrams

Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet.

For Milestone 2 provide only:

*High-level UML class diagrams* for implementation classes of core functionality, i.e. functionality with provided interfaces. Focus on a main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your site. For UML, you could find many references including http://edn.embarcadero.com/article/31863.

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

Identify *only actual and specific* risks in your current work such as (list those that apply:

* *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 how do 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 the risk and explain (2-3 lines), list how you will address this issues’ (2-3 lines)

7. Project management

Describe in no more than half a page how you managed M2 tasks and how will manage future tasks and what tools you use.

**Submission of Milestone 2 *Document* for Review**

Formatting instructions for M2 document must be followed precisely, as outlined below. Submission must be done by the deadline specified; any extension has to be approved ahead of time.

In creating, editing and finalizing Milestone 2 document follow similar team process as outlined for Milestone 1 document

The whole student team submits one milestone document for Milestones 2, as follows (same as M1 submission): Team leads will send e-mail with a link (NOT the attached file) pointing directly to Milestone 2 Document to [hjsong@sfsu.edu](mailto:hjsong@sfsu.edu) and [TA](mailto:cgomez10@mail.sfsu.edu) with the subject line as specified below. This link MUST point directly to M2 file in the team group account on Github.

**e-mail subject line:** Must be “CSC648-848 -2020 Milestone2 Document Team N” in the subject line (N is a team number).

**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 2020 Milestone2 Team*N*.XXX** (*N* is your team number). File format could be MS word (preferred) or PDF.

## First page of Milestone 2 document must include

–“SW Engineering CSC648/848 2020 ”

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)

* **The rest of the document** must have numbered sections outlined above in “Content and structure for Milestone 2 document for review by institutors”. Each section must start on a new page

**Vertical SW Prototype**

In addition to the **Milestone 2 document,** the team will create a “**vertical SW prototype**” to test the infrastructure and chosen frameworks and to jumpstart the coding effort. The vertical prototype is the code that exercises full deployment stack from browser (with simple *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. 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 and thus to serve as a basic “scaffolding” for final product. It also serves as “teaching and training” tool to bring the rest of the team up to speed on SW, frameworks etc.

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 list data on *test home page* (simple home page used to test vertical prototype), then get a response form the DB and display the proper content. Every output should be retrieved from the DB.

* UI for *the test home page* can be a simple one containing below functions
  + #1 : list of patients
    - Assuming the test home page is logged in as the PT (PT A)
    - A list of patients dedicated to PT A
  + #2 : search exercise plan for a particular patient (PAT B)
    - Given a period of time entry field (the date entry initiates search inventory from SQL DB )
    - Show the 2-3 properties of the exercise plan which is assigned to PAT B (like video, description, # of repetition, etc.)
  + #3 : search the progress report from a particular patient (PAT B)
    - Show progress report in video or text which is uploaded by PAT B.
  + #4: search all the activities of a PT A for a PAB B.
    - Activity details should be displayed such as responding to questions, viewing videos, reading profile, writing notes
    - Each activity should be displayed with time and duration (e.g.. Oct 2 10:00 ~ Oct 2 10:05, 5 mins)
* The DB can have only a few items. The items in the DB shall be encoded with full schema as it is defined by now in M2 document

**Vertical prototype 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 make 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.

**Vertical SW Prototype delivery/submission**

Your team will submit vertical prototype similarly as M0, via e-mail to class TA and the instructor by the 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 2020 Milestone2 Vertical Prototype Team N” in the subject line (N is a team number).

**e-mail body contains:**

* some courtesy text explaining what the e-mail is about
* link to vertical prototype home test page so it can be ran and tested
* Link to code that executes above home page
* link to githiub there files for vertical prototype reside
* access and pointer to the DB used for vertical prototype
* any necessary permission keys etc.

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

* Functionality and correct display data results (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

We will not grade vertical prototype but you will **have to follow up feedback from instructors and revise accordingly after submission.**

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

# **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 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. Please enter the revision summary in history table. (This is similar to Milestone 1 review process).

After this revision freeze the Milestone 2 document and use it for final project document delivery. Store the document in Milestone folder on your team github repo.

Vertical SW prototype will be reviewed by class TA and you will get the feedback which you must analyze and incorporate as necessary.

**Note on project management for the tasks of Milestone 2 and beyond**

Milestone 2 is a good time to start front-end and back-end team operate more independently while also agreeing on common interfaces. This makes the team more efficient. It is critical to always assign tasks, and for each task know person in charge and deadline. For tracking the individual tasks you can use e-mail or slack or perhaps some tools like Trello <https://trello.com/>