CSC 648-848 Spring 2018 Milestone 2: More Detailed Requirements, Specs, Architecture, UI mock-ups and a vertical SW prototype

03/06/18

Announce: TBD
Due: Check instructor's e-mail and iLEARN

Objective

Goals of Milestone 2 are:

- Compete first phase of UI design (e.g. mockups with usability feedback from instructors) and DB design, and practice requirement prioritization and UML
- Develop first simple bare-bones prototype to test the infrastructure, educate the team, resolve technical issues and also serve as basis of further development.

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) to be demoed and reviewed in the class and if needed outside of the class

Milestone 2 has 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.

Milestone 2 differences DO NOT need to be edited in Milestone 1 doc which remains frozen. You should start with Milestone 2 only after you have incorporated instructors' feedback on Milestone 1. Milestone 2 document is a separate document from Milestone 1 document.

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 enter a search term on *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 *test result page*.

- UI for *the test home page* can be a simple one containing class and team identification and one entry field or pull down for search parameter *but also can be a first draft of your app home page with chosen CSS (try it!)*.
- *The test result page* needs to display search results including images and their thumbnails in a simple layout, as well as maps in *any reasonable layout* (the goal here is to make sure you can access and display items and NOT the ultimate UI).
- 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 (we recommend to use of WorkBench or some tools to set up the DB).

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

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 sue 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.

Content and Structure for Milestone 2 Document for Instructors' Review

Use as much space as you need, but the expected length is about 20-25 pages.

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

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 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,**

including UI text, naming for main variables, classes and database elements etc.
Focus on data items unique and important to your application and avoid explaining obvious things like Internet, Browser, Cloud, etc. 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.). Be sure to cover ALL and especially unique features of your product. OK to add new or delete previous functional requirements from Milestone 1, if you can justify it.

Prioritize each requirement/spec with 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, 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), so be very careful.

In terms of presentation, for easier review, the best is to group all requirements first by priority i.e. list Priority 1 requirements first, then Priority 2 etc. and within each priority section you should group them by actors (users, admin)

3. UI Mockups and Storyboards (high level only)

- Create mockups/storyboards for all major use cases (e.g. 4-6 major use cases, each storyboard appr. 1-3 screens). Have ONE mockup per page so we can easily read it comment
- Start with black and white wire diagrams focusing on basic layout and description of the functions in each main area of the GUI. Create simple “storyboards” (sequence of mockups) organized by use cases. This helps test the navigation and flow. The format for UI mockups is very flexible but we strongly recommend hand drawings, which you can scan and include in final Milestone 2 document. Do not use graphics or colors yet (unless absolutely necessary), it draws attention from basic UI concepts (functions, behaviors, layouts, flow…).
- Briefly describe behavior of each function using text (user action, output, navigation, error conditions)

- "Test" the above mockups, keeping ease of use and your use cases in mind. Walk through your mockups as if they are "live" with someone playing the role of user clicking on buttons as per sue cases.
- Use data terms and names consistently with Data Dictionary and use cases.
- Make sure that the actual display of mockups in the hard copy Milestone 2 document is easy to read

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

4. High level Architecture, Database Organization

- *High level Architecture* of the code must use MVC framework and be OO and consistent with UML class diagram below. Outline in no more than onea page how will you structure the code/functions to be MVC compliant (no need to be "purist" use common sense and good modular design – tell us at very high level what will go into M, into V and into C components of MVC or provide one chart for this)
- *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 (check instructors' suggestions). Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions in Section 1 above.
- *Media storage*: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision you must make by the end of M2). Describe any other special data format requirements like for video/audio/GPS etc.
- *Search/filter architecture and implementation*: what will be the alg/SW for search; what DB terms will be searched, how it will be coded and organized in the DB (check instructors' suggestions in the class. OK to use SQL and %like). Similarly, say what DB items will be filtered/sorted if applicable
- Your own APIs (if any): Describe and define at high level any major APIs that you will create
- 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. Any new SW or framework you will be using has to be approved by CTO in writing by this time.

5. Content for vertical prototype (text, data, images…)

Best to make your own images/data or get free to use images from internet.

Text and data associated with media can be arbitrary but make sure they look good for the demo..

If you use addresses make sure they are of public places and not private homes to avoid issue of privacy.

Don't forget to create image thumbnails (use some library) so that displays of large number of items/images in your app is fast. Use open source tools for this.

Be sure to use only appropriate content and text for general public according to SFSU polices.

6. High Level UML Diagrams

Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet. One good one is http://edn.embarcadero.com/article/31863

For Milestone 2 provide only:

a*) 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.

b) *UML Component and deployment diagrams*

Use data terms and names **consistently** with Data Definition Section 1 above.

7 .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 risks? The key is to resolve risks as soon as possible. (Note that we will provide you with basic set of images). Categorizing risk as above helps a lot in managing them. Be brief: identify the risk and explain (2-3 lines), list how will you address this issues' (2-3 lines)

Vertical SW Prototype delivery/submission

Your team will submit vertical prototype similarly as M0, via e-mail to class CTO Anthony cc Petkovic. 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 Section M Spring 2018 Milestone2 Vertical Prototype Team N" in the subject line (where M is section 01 or 02, N is a team number (01, 02 etc.).

e-mail body contains:

- link to vertical prototype home test page so it can be ran and tested
- link to githiub there files for vertical prototype reside

Vertical prototype shall be evaluated (but not graded) 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

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.

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 Petkovic@sfsu.edu and Anthony 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 Section M Spring 2018 Milestone2 Document Team N" in the subject line (where M is section 01 or 02, 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 Section M Spring 2018 Milestone2 Team*N*.PDF** (where M is section 01 or 02, *N* is your team number)

- **First page of Milestone 2 document must include**
  –"SW Engineering CSC648/848 Section M Spring 2018 "
  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
  Say if the team is Global or Local
  –Names of students (team lead first) -
  Say if the team is *local* or global
  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

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 any of: e-mail, markings on your document or in class during team meetings. 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.

Note: Instructors will comment form the standpoint of CEO, VP of Marketing (who translates customer and marketing requirements) and CTO (Architecture etc.). You may choose not to agree with the comments. This is OK as long as you justify them and are prepared to live with that and deliver. In some cases, instructors may insist on some features or decisions.

Vertical SW prototype will be reviewed off-line 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. Consult the class slides on project management. For tracking the individual tasks you can use e-mail or slack or perhaps some tools like Trello