**CSC 648-848 Fall 2017 Milestone 2: More Detailed Requirements, Specs, Architecture, UI mock-ups and a vertical SW prototype**

**10/07/17**

**Announce: TBD**

**Due: Check instructor’s e-mail and iLEARN**

Objective

Milestone 2 consists of two parts:

* **Milestone 2 document** (submitted similarly like Milestone 1)
* **Vertical SW prototype** 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 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 one.**

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, via middleware, to DB and back, including your chosen framework. It has to be deployed from team account, same way the final product will be deployed. It shall allow one to enter a search term in the browser, then get a response form the DB and render it back on the browser. UI for this can be simple one field entry and DB can have only a few items. The items in her DB shall be encoded with full schema as it is defined by now (we recommend use of WorkBench to set up the MySQL DB). 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.

**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 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 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 documents 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 ½ a 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 and items in each DB table (check instructors’ suggestions)
* *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
* Your own APIs: Describe and define at high level any major APIs that you will create
* Describe any significant non-trivial algorithm or process (like rating, ranking, automatic prioritizing of items etc.)

Any external code or framework you will be using has to be approved by CTO in writing by this time.

Use data terms and names (class diagrams, table names, variables) consistent with your Data Dictionary.

# 5. 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>

At minimum provide:

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 Glossary/Data Dictionary.

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 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**

Your team must demo the following during the SCRUM meeting with instructors in the last hour of the class (exact schedule to be announced). Vertical prototype might also be reviewed outside of class.

The goal is to create a SW basis for your final project. In terms of functions it takes a keyword/string from simple UI, then displays the DB item on the page. For those who decide to use BLOBs for images this prototype serves to test this in terms of development and performance.

Guidance and requirements:

* The vertical SW prototype must use the core infrastructure and frameworks your team has specified in Milestone 2 document, only with frameworks chosen and approved.
* It must run from your team’s account on AWS team account. The code must be checked in team private repo on github
* Architecture of the code must use MVC framework and be OO and consistent with UML class diagram– this will be checked
* Using MySQL Workbench, create about 5 database items that have the structure of your key DB item which includes all its main items (text, images…). (This way you will learn WorkBench which will be app of choice for admin function!). Follow instructions and suggestions from the class
* Create simple test page with one field where you can type the key of the above data item. (This can be very simple and partially hardcoded and can be B&W page with team name and one entry field)
* Implement search algorithm and filtering you plan to have for final product
* Show the response by displaying values of that item in some form on the browser using CSS from the framework you have chosen. The UI is not the key here, showing a functioning data infrastructure is much more important. For images make sure you create thumbnail of large image for displaying small images ion results lists
* Document the vertical prototype code so it can provide guidance and learning for the rest of the team how to expand it.

Content for the demos (text, data, images…)

Make your own images/data or get free to use images from internet. Be sure to use only appropriate content and text for general public according to SFSU polices. Text and data can be arbitrary.

Don’t forget to create image thumbnails (use some library) so that displays of large number of items/images in your app is fast.

During

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

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

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](mailto: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 Fall 2017 Milestone2 TeamN” in the subject line (where M is section 01 or 02, N is a team number (01, 02 etc.).

* **File name of the M2 document to which the link is pointing to MUST be**: CSC648-848 Section M Fall 2017 Milestone2 Team*N*.PDF (where M is section 01 or 02, *N* is your team number)

## First page must include

–“SW Engineering CSC648/848 Section M Fall 2017 ”

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

For each student the university affiliation

–“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 in class and potentially off-line and you will get the feedback which you must analyze and incorporate as necessary.

# **Evaluation and Grading**

We will grade each milestone only when it is submitted with final project at the end of the class (as Milestone 5 docs), after it has been modified incorporating instructors’ feedback. Instructors’ feedback and intermediate versions are not graded in order to encourage iterative improvements to the documents. The final project submission has total of 5 milestones with equal weight.

Milestones improperly submitted will first be returned, and if problems persist 10% penalty will be applied to the grading of that milestone.

Milestones have to be submitted on time. In case of justifiable reasons to delay, permission has to be obtained by e-mailing to [Petkovic@sfsu.edu](mailto:Petkovic@sfsu.edu) prior to the deadline. Late submissions with no permission incur 10% penalty on the grading of that milestone.