Milestone 1 Checklist

- ❑ Demo video and who-did-what document (Section 1)
    - ❑ Mentor must have seen demo (live or video)
- ❑ Tagged code in VCS (Section 2)
- ❑ Up-to-date plan and design artifacts (Section 3)
    - ❑ Mentor must confirm requirements (and possibly others)
- ❑ Up-to-date feedback collection, plans to handle, and outcomes (Section 4)
    - ❑ Mentor must sign off on
- ❑ Individual assignment outcomes (Section 5)
- ❑ Instructions to the instructors, submitted to eCourseware dropbox (Section 6)
- ❑ Teammate evaluations (Section 7)

Milestone 1 Instructions

The main goal of this milestone is to implement the most important features of the system (although polishing and minor bug fixes may be needed) and mitigate all the key risks. To that end, the team must submit the following things for this milestone.

1. Demo Video

Your team must create a demo video of your software. This video is mainly to assist the course instructors in grading your progress on the project. The video must also have an accompanying text document that lists who built each of the demoed features. The demo video and document must meet the following grading criteria:

- **Criterion**: Demonstrate the progress that the team has made so far.
    - **All the new features.** Include all the latest features in the demo. Don't leave any out. A big point of this exercise is to demonstrate all the wonderful progress that the team has made. Note that this criterion does not mean that you should skip re-demoing old features. It just means you shouldn't skip the new ones.
    - **What's new?** Clearly state which features are new as they are being demoed. You need not explicitly state which features are old, unless you think there may be confusion.
    - **Backend too.** Although UI features are a high priority, you may also demonstrate that backend functionality is working, even if it's not yet connected to the frontend. The key thing is to prove that the code runs and works! Along those lines, you may demo automated tests.
- **Criterion**: Make clear who contributed what to the project.
    - **Who made it?** In the text document, give an entry for each new/updated feature demoed. Each entry must include the name of the feature, the time offset in the video where the feature was demoed, and the U of M username(s) of who built the feature. If multiple people contributed to a feature, say who did what. Don't forget to credit anyone. Be sure not to make any factual errors.
    - **New or old?** If it's not clear, make it clear what part of the work is new.
- **Criterion**: Display the team's work in the best possible light.
    - **Emphasize user goals or stories.** Any demonstration of UI must be framed as fulfilling a high-level user goal and/or take place in the context of a cohesive story. For example, the presenter may describe one or more characters (with names, like Alice and/or Bob) and relate a story about the character using the software. The demo should be well thought out, and not leave the audience with the impression that the presenter is making it up as he/she goes along. Use realistic names for things and not made-up placeholders, like "foo" and "slafjsd".
    - **UI first.** Since the UI is generally most interesting, you should lead with that.

    - **General audience.** Don't forget that not everyone is as familiar with your project as you are. To be on the safe side, explain it as if you are talking to someone who has never seen it before.

    - **No special effects or fancy editing.** The video should clearly show a user (or users) interacting with your web app. Don't add special effects or sound effects, which distract or detract from the authenticity of the interaction.

- **Criterion**: Length and format constraints.

    - **Time limit.** The video must be no more than 10 minutes long.

    - **Fill the Time.** Your video should be at least 7 minutes; otherwise, you're probably doing it wrong.

    - **Video format.** The video must be in a format playable in VLC (http://www.videolan.org/vlc/), which accepts most common formats. You may also upload your video to YouTube, and submit it that way.

See the *Above and Beyond Points* document for A&B points associated with the creation of the video.

Your mentor must confirm that they have seen your work for the iteration demoed, and have provided you with any feedback they may have prior to the end of the iteration. The demo may be delivered live and interactively or via the video, as per the mentor's preference.

2. Tagged Code in Repo

The complete version of the code captured in the video demo must be tagged in your version control repository.

3. Up-to-Date Plan and Design Artifacts

The following planning and design artifacts must be kept up to date (and thus be up to date when the milestone is due):

- Requirements specifications and their statuses.
- Software architecture and designs. These artifacts are useful for comprehending your project's code, and thus, they should be kept up to date.
- Risks and their statuses. As the project proceeds, your team should be mitigating risks.
- Any of the other plans (schedule, QA, configuration management, collaboration) if they have changed.

You need not keep the interface sketches up to date. These sketches are mainly useful for developing the interface design, but once the interface is implemented, the sketches are no longer particularly useful. However, your team should archive your interface sketches for later reference by the instructor.

Your mentor must confirm that the requirements specifications and their statuses are up to date. They must also have been given the opportunity to review the other artifacts if they so wish.

4. Up-to-Date Feedback

Feedback collection and management must be part of each iteration and milestone. See the *Feedback Collection and Management* document for detailed instructions.

Your mentor must confirm that you have collected all the feedback that he/she has given your team, and that he/she agrees with your plan for handling it and/or outcome.

5. Individual Assignments

Full task planning and outcome reporting must be part of each iteration and milestone. See the *Individual Assignment Specification* document for detailed instructions.

6. Instructions to the Instructors

Since the way to access the various milestone artifacts may vary from team to team, each team must submit a document containing instructions to the instructors regarding how to access each artifact. This document must be submitted to an eCourseware dropbox by the milestone deadline.

7. Teammate Evaluations

At the end of each iteration, each team member must provide an evaluation of each other team member. Instructions and forms for performing these teammate evaluations will be communicated by email near the end of the iteration.

Milestone 1 Mentor Sign-Off Form

Mentors: Please indicate your approval of the following items—but ONLY if you agree 100% with the statement for the item.

If you have ANY disagreement, do not give your approval. Instead, provide the team with feedback, and have them resolve whatever issue is preventing your approval.

- ❑ I have reviewed a demo of the software, and I have provided the team any feedback I had.
- ❑ I have reviewed the requirements specifications, and they are up to date and consistent with my wishes.
- ❑ I have reviewed any other plan or design artifacts that I wish to see, and I approve of them.
- ❑ I have reviewed the team's feedback management system, and ...
    - ❑ All the feedback I have given them is accounted for.
    - ❑ I agree with all the team's plans for handling the feedback.
    - ❑ I confirm that all outcomes are accurate to the best of my knowledge.