# CSC 648/848 Milestone 1: User stories & High-Level Requirements

**Objective:**

Based on the brief high-level project description, the objective of Milestone 1 is to develop: a) key personas and user stories; b) high-level functional & non-functional requirements, and c) list high-level frameworks and tools to be used (generally the same as in M0). Note that M1 description includes requirements and specs to get early feedback. Future designs can deviate from Milestone 1 in the spirit of iterative SW design and development.

For user stories and requirements, it is encouraged to use your own ideas, interview with people around you and find your own value proposition. Please consult class material on User stories and Requirements.

The whole student team submits **one** milestone document for each Milestone 1 – 5. For submission, put it on your github (under Milestone folder) and send a mail to the instructor. Expected size of this document is about 6-10 pages.

**Content and structure for Milestone 1 document for review:**

In the document for Milestone 1 you must cover all the following subsections in exact order as below in one file.

1. Executive Summary: Short description of the final product/application and its key advantages, novelty, value (up to 1 page). Make it as an executive summary – think of answering the question of why a VC (venture capital) will fund this project. We suggest you assign a name to your project for easier reference and good "marketing" (code name). This summary should be readable to a general manger/executive that is not a CS specialist and is used to explain to advertise/promote your project. Typical outline is: one paragraph on the motivation and importance of the application you are developing, followed by a paragraph on what your application will be doing and how it helps the users (no jargon) and optionally what is unique in your product. At the end say in one paragraph something about your team (e.g. about your student startup team…).

2. Personas and User stories: Summarize several (3~5) key *personas* (categories of users) for your application – their general behaviors, interests, skills, pain points. In a user story you say how each persona will use your app (at high level). Please number your user story as their ID, organize user stories into a similar activity, and put your

priority for each user story. Simple text format is OK. Focus on WHAT users do, not on HOW is the SW implemented.

3.Data Definitions - define main terms, data structures and "items" or "*entities*" *at high or logical (not implementation) level* (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements used in your app, types of users and their privileges etc.) specific for this application and not on general, well-known terms. These terms and their names *must be used consistently* in all documents, user interface, also in naming software components and database elements in your code implementation. In later milestones you will add more implementation details for each item. You will later expand this section with more details.

4.Initial list of functional requirements – see class materials. This refers to high level functions you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep personas in mind. Develop these functions to be consistent with user stories. Number each requirement with *unique numeric value* and use these numbers consistently from then on. Each requirement has a requirement ID, title, 1-3 line of description, owner/initiator (optional), priority, user story to be referenced.

5.List of non-functional requirements (performance, storage space, usability, security, storage, availability, fault tolerance (with the expected number of users)) Number each requirement. Please refer to these non-functional requirements in your design and develop from here. Example are listed as below :

1) (compatibility) Application can be used on several mobile browsers including chrome and Mozilla Firefox.
2) (development requirement) Data shall be stored in the team's chosen database technology on the team's deployment server (e.g. Google cloud or Amazon AWS)
3) (usability) Application shall be easy to use and intuitive by taking users' voice input.
4) (development requirement) The code in the master branch of team's github repo should be well maintained and test, and guarantee working at any time leveraged by CI (continuous integration).

6.Competitive analysis: Find 3-4 competitive features against existing solutions which are available in the market. Present competitors' features vs. your planned ones. First, create a table with key features of competitors vs. yours planed, at only very high level, 5-6 entries max. After the table, you must summarize in one paragraph what are the advantages of your planned product to what is already available.

7.High-level system requirements Briefly provide itemized list of all main SW components such as frameworks, tools and systems to be used, supported browsers and deployment platform (SW and server) to be used. This list is to be the list of approved tools and systems from M0. Any other external (open source) code/API/tool must be listed.

8.Team: list student names, name of the roles for each member. *If you form the study group, please list them too with their key milestones. (If you present detailed study plan, you will earn extra points)*.

9.Checklist: for each below item, you must answer with only one of the following: **DONE**; or **ON TRACK** (meaning it will be done on time, and no issues perceived); or **ISSUE** (you have some problems, and then define what is the problem with 1-3 lines)

- Team found a time slot to meet outside of the class
- Scrum Master shares meeting minutes with everyone after each meeting.
- Github master chosen

- Everyone sets up their local development environment from the team's git repo.

- Team decided and agreed together on using the listed SW tools and deployment server

- Team ready and able to use the chosen back/front-end frameworks.
    - For each technology (front/back-end/DB/cloud) , team decides who will lead the study of each technology and what will be the specific goal of the study within one month from the M1 announcement.
        - Ex : implement DB scheme for main data items by next 2 weeks.
    - If you list a detailed study plan for this, earn extra point!

- Team lead ensured that all team members read the final M1 and agree/understand it before submission

**Background reading:**

- Class material on requirements and specs
- Relevant existing applications and products.
- Info about allowed frameworks
- M0 document and documentation on SW tools and frameworks you plan to use
- Git, Github tutorials along with a team tutorial session on Git/Github

**Submission for Milestone 1 document for review – you must follow the instructions below PRECISELY:**

Teams must collaborate in creating M1 document by having working M1 document on their team Github repository so all team members and instructors can access it. Added advantage of doing it this way is that it builds teamwork and communication. We recommend having a folder for project documentation on team github where milestones and other similar files can be kept.

**M1 document format and structure**

- **Title page** MUST include
  –"SW Engineering CSC648/848"
  Project/application title and name (you can use the code name you chose for your application)
  –Section number, Team number
  –Names of students (team lead first) and his/her role
  –"Milestone 1"
  –Date
  –History table (revisions) (Note: you will update this document based on instructors' feedback so this is important)

- **The rest of the document** has to contain sections as described above under "Content and structure for Milestone 1 document for review"

Team leads and M1 editors: make sure document is well formatted, reads well, is complete, and looks professional. This will be part of your portfolio and will influence the grade. Make sure all team members read final version and give comments before submission.

**Instructor's feedback and creating final Milestone document for Final Project delivery**

In the course of developing Milestone 1 you can ask instructor questions via e-mail and during team session in the class. Upon submission of Milestone 1 you will get feedback from instructors by any of the following: e-mail and in class during team meetings. This feedback must be analyzed and taken into account by your team in order to revise your

Milestone 1 and this must be used subsequently for the rest of the project. Please enter the revision summary in history table.

Instructors will comment from the standpoint of CEO. You may choose not to agree with the comments. This is OK as long as you justify this and are prepared to live with that design and deliver it. In some cases, instructors may insist on some features or decisions.

Upon getting instructors' feedback on your questions and submitted document, you need to revise your first draft, and use it as a basis for developing Milestone 2. This revised document Milestone 1 will be submitted as part of final project delivery in Milestone 5.

Future Milestone 2 functions and actual SW app may differ from what you proposed in Milestone 1, that is normal and in fact expected to happen in the spirit of iterative SW development. In that case it is encouraged to further modify M1 document.

**Evaluation and grading**

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 and in a way as specified above.**