

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 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. Instructor 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 group all requirements **first by priority i.e. list Priority** e.g. 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 use cases. Have only one to two mockups per page so the class instructor can easily read it and 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. For each storyboard repeat a short version of related use case so the reader knows what is being done
- The format for UI mockups is very flexible, but I 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...).
- “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 database architecture and organization

I will teach you everything you need to know about how to design the database for your application. Take into consideration that my students from my DBMS class learn this stuff in approximately two months. You'll need to learn all the basics to get you started in one lecture.

- DB organization: Create a high level database model based on 5 to 10 business rules that pertain to the same group of priorities (functional requirements).
 1. Define clearly your 5 to 10 business rules (must be extracted from section 2)
 2. Describe your entities, their attributes, relationship, and domains at the high level.
 3. Create an entity relationship diagram (ERD) based on those business rules.
 4. Create a database model based on your ERD
 5. In one sentence define which DBMS you will choose to create the database and why
- ERD and Database models can be created by handwriting or any other tool that is more convenient to you. But, I recommend for ERD draw.io and MySQLWorkbench for the database model. If you are planning to use a DBMS other than MySQL in your product, you still can create the data model in MySQLWorkbench, and export the model to your DMBS.
- Media storage: Decide if images and video/audio will be kept in file systems or in DB BLOBs (decision on file vs. BLOBs must be made 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; how will you organize search items fro the user; 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).

5. High Level APIs and Main Algorithms

- Your own APIs: 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 (instructor) in writing by this time.

6. High Level UML Diagrams

I will teach you how to create good UML diagrams in class, but if you wish to explore more about this topic by yourself, here is a good UML resource

<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 highlevel classes only (one or at most two levels deep). This must reflect an OO approach to implementing your site.

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

7. High Level Application Network and Deployment Diagrams

At the high level, create application network and deployment diagrams.

- Application Networks Diagram: this diagram will describe the logical and physical of your system. It must show protocols used for each service in your system, network configuration, security, firewalls, proxies.....
- Deployment Diagram: this diagram will show the architecture of the system as deployment (distribution) of software artifacts, or modules in your system, to deployment targets

Those diagrams are based on how your application interact with some other internal and external components at run-time. For instance, if you are using external libraries in your application, you also need to research about their network and deployment architecture, and therefore, show them in your network and deployment diagrams.

8. 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 me how do you plan to resolve each actual risk you have. 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 you will address this issues' (2-3 lines)

9. 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. We must start using Trello or similar tools for task management which offer unified dashboard view of all tasks and status.