SW Engineering CSC648/848 Summer 2018”

PhotoStock

Team 03

David Hobby [dhobby@mail.sfsu.edu](mailto:dhobby@mail.sfsu.edu)

Sandesh Fill-in

Cavit Altindag

Ekta Fill-in

Arnold Fill-in

Michael Fill-in

“Milestone 1”

Date

(revisions)

**David Hobby**

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 we should fund this project. We suggest you assign a name to your project for easier reference and good “marketing”. This summary should be readable to a general manger/executive that is not a CS specialist and is used to explain and also 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 (high level only, no jargon) and optionally what is unique and special in your design. At the end say in one paragraph something about your team (e.g. about your student startup team…).

**Ekta**

1. **Personas and Use Cases:** Summarize key personas (categories of users) for your application – their characteristics, goals, skills, pain points etc. Half a page per persona – see class notes. Then provide 4-5 main use cases (one paragraphs for each use case) - see class notes on more detailed format for requirements. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how the application is used. Focus on WHAT users do, their skill level, not on HOW is the SW implemented. NOTE: avoid specific on HOW functions will be done and text resembling user manual: this is supposed to guide the design of the future product and is NOT a description of how the product will work (you don’t know that yet) – see class slides for details.

**Sandesh**

1. **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 know terms. These terms and their names *must be used consistently* from then on in all documents, user interface, in naming SE components and database elements etc. In cases where you attach behavior and privileges to data items (e.g. user types) that also drives the design of the SW. In later milestones you will add more implementation details for each item. You will later expand this section with more details.

**Cavit**

1. **Initial list of functional requirements:** – see class notes. 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 the user in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement with *unique numeric value* and use these numbers consistently from then on. For each functional requirement use 1-3 line description. At this stage no need to prioritize the requirements.

**Arnold**

1. **List of non-functional requirements:** (performance, expected load, security requirements, storage, availability, fault tolerance…) Number each. Note that mandatory high level non-functional specs are given in high level document, so for Milestone we recommend you simply copy them from high level document from iLearn. Please observe and adhere to these non-functional requirements in your design and development from now on – you are not allow to change them unless you get permission.

**Michael**

1. **Competitive analysis:** Find 3-4 competitive products. Present competitors’ features vs. your planned ones. First, create a table with key features of competitors vs. yours planed, only very high level, 5-6 entries max (as shown in the class). After the table, you must summarize in one paragraph what are the planned advantages or competitive relationship of your planned product to what is already available. In the table clearly mark your product, e.g. shade its column/data.

1. **High-level system architecture:** Briefly provide itemized list of all main SW components such as frameworks, APIs, 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 code/API/tool must be approved by instructors and you have to justify it.
   * Server Host, Instance size ( CPU and RAM )
     + AWS, 1 vCPU, 1 GiB RAM
   * Operating System and Version Number
     + Linux version 4.4.0-1060-aws (buildd@lgw01-amd64-036) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.9) ) #69-Ubuntu SMP Sun May 20 13:42:07 UTC 2018
   * Database and Version Number
     + mySQL/5.7
   * Webserver and Version Number
     + Apache/2.4.18 (Ubuntu)
   * Server-Side language and Version Number
     + Express.JS
   * Also list any technologies or packages you will need that you think is important.
     + Enzyme, Jest.
2. **Team:** list student names, name of the team leader, names of front and back team lead and initial roles for each member.

**Team Lead:**

**David Hobby**

**Front-End Lead: QA: Back-End Lead:**

**Ekta Fill-in Michael Fill-in Sandesh Fill-n**

**Engineers:**

**Front-End: Back-End:**

**Arnold Fill-in Cavit Altindag**

1. **Checklist:** for each item below 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 **DONE**.
* Github master chosen **DONE**.
* Team decided and agreed together on using the listed SW tools and deployment server **DONE**.
* Team ready and able to use the chosen back and front end frameworks and those who need to learn and working on it **ON TRACK.**
* Team lead ensured that all team members read the final M1 and agree/understand it before submission **ON TRACK**