

CEN 4010 Principles of Software Engineering

Milestone 1: Team Project Proposal and Description

Florida Atlantic University

Dr. Shihong Huang

shihong@fau.edu

Instruction:

This is a team project. Each team submits one document to Canvas

This document is the first milestone in your term-long project. You will propose a term project that will be developed and deployed over the course. Your team decides on the specific theme and features of your project, or you can work on the project that I have proposed for you (see Project Description on Canvas). If you choose your own project, please have instructor's approval before proceeding.

Your term project proposal and description should include at least the following sections, based on which you can add more sections when you see fit.

1 Executive Summary – Viviana

A short description of the final project and its key advantages, novelty, and values, up to 1 page. Make it an executive summary -- think of answering the question of why you develop this project and target at what market sectors. Assign a product name to your project. This executive summary should be readable to a general audience who is not a computer science specialist. The executive summary is also used to advertise and promote your project.

2 Competitive analysis – Max, Viviana

Analyzing competitive products available today. Present competitors' features vs. your planned ones. First, create a table with key features of competitors vs. yours. Only at very high level, 5-6 entries max. After the table, you must summarize what are the planned advantages or competitive relationship to what is already available.

3 Data definition – [all]

This section serves as the "dictionary" of your document. It defines 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, actors, types of users etc.) specific for your 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 software components and database elements etc. In later milestones, you will add more implementation details for each item. You will later expand this section with more details.

4 Overview, scenarios and use cases – Max

This section describes the project overview (in much more details) and likelihood usage scenarios of your product from end users' perspectives. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how is the application used. Focus on WHAT users do, their skill level, not on HOW the system is implemented. You can expand use cases provided in high level document in future milestones.

5 Initial list of high-level functional requirements – Jason

This refers to the high-level functionality that you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the users in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement and use these numbers consistently from now on. For each functionality use 1-5 line description.

6 List of non-functional requirements – Monica and Vern

For example, performance, usability, accessibility, expected load, security requirements, storage, availability, fault tolerance etc. Number each. When possible, try to quantify these quality attributes.

7 High-level system architecture – [all]

Lists of main software products, tools, languages and systems to be used, list of core APIs available at this point, supported browsers etc.

You also have to decide on which frameworks you will use if any. These provide both user interface, as well as cross-platform and cross browser layout/css. All external code you plan to use must be listed along with their license.

8 Team

List student group names, name of Scrum master, product owner and initial roles for each member

9 Checklist

For each item below you must answer with only one of the following: DONE, 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). Reflect these items in your Jira project space:

- a) Team decided on basic means of communications
- b) Team found a time slot to meet outside of the class
- c) Front and back end team leads chosen
- d) Github master chosen
- e) Team ready and able to use the chosen back and front-end frameworks
- f) Skills of each team member defined and known to all

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

10 Tasks before submission

Teams must collaborate in creating M1 document by having working M1 document on their team GitHub repository (similar to managing code) so all team members 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's GitHub where milestones and other similar files can be kept.

11 Submission

Each team submits one single word document with all the above required sections to Canvas by the due date. Must have a title page to your document, including:

- a) Course Title and term: CEN 4010 Principles of Software Engineering, Semester and Year
- b) Document name: Milestone 1 Project Proposal and High-level description
- c) Your team name, and project name (you can use the name you chose for your team)
- d) Team number (I will assign you one)
- e) Names of students (team lead first) with names and e-mails
- f) Documentation Date
- g) History table (revisions dates) (Note: you will update this document based on instructors' feedback so this is important)

12 Grading criteria

Your document needs to be well-written, well-organized (formatted) and reads well. Grading is based on cohesiveness and completeness.

- | | |
|---|-----------|
| 1) Executive Summary | 10 points |
| 2) Competitive analysis | 10 points |
| 3) Data definition | 10 points |
| 4) Overview, scenarios and use cases | 10 points |
| 5) Initial list of high-level functional requirements | 10 points |
| 6) List of non-functional requirements | 10 points |
| 7) High-level system architecture | 10 points |
| 8) Team and check list | 10 points |
| 9) Working with GitHub | 10 points |
| 10) Deliverable | 10 points |