Software 6

4/16/18

Milestone 4 Beta Launch and Reviews

Principles of Software Engineering, Spring 2018

CEN 4010

Team Info

Team name: Software 6

Project: Access Control Device

Team Number: Milestone Group 6

Team Members:

Timothy Duncan (Front/Back-end Developer) – tdunca11@fau.edu

Bentialy Saint Julien (Front-end Developer) -

Jonathan Giger (Project Owner) -

Jonathan Parreira (Back-end Developer) -

Mihail Sandor (Scrum Master) -msandor2014@fau.edu

2.2 Product summary

For example, how would you market your product. Write it for wider audience: for example, executives, marketing and customers. Maximum ¾ page.

1) Name of the product

2) Explicit list of ALL major committed functions. These are your FINAL Priority 1 functions for which you will be graded and that your team shall actually deliver and test for. You should use plain English as if you would talk to a customer, and not the language used for formal requirements specification. This is your FINAL functional commitment. In other words, failure to deliver on some of Priority 1 functions will result in reduced grade. Please write it in the list format (each item max 1-3 lines) so it is easy to check.

3) Describe unique features in your product (if any)

4) URL to your product accessible to instructors, on deployment server The list of final functions will be checked on your final delivery for functionality and correct operation as you specified them. Failure to deliver complete list of these committed functions may result in reduced grade.

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.

2.3 Usability test plan – maximum 2 pages

Select ONE major function (NOT login or registration) to be tested for usability. We recommend search or upload/post.

Write a usability test plan for this selected function. Please consult class material on developing usability test plan and questionnaire. This test plan is to contain:

1) Test objectives: 0.5 page

2) Test plan: System setup, starting point, task to be accomplished, who is the intended user, completion criteria, URL of the system to be tested. 3/4 page

3) Questionnaire form: 3 Lickert scale questions, in a form easy to be used by reviewer (check class slides). 3/4 page

Your test plan must be formatted to be easy to read and use by usability testers, including the questionnaire.

You can also ask your friends or team members to do the usability test.

2.4 QA test plan – maximum 2.5 pages

For the same function you chose for the usability test, write AND execute a QA test plan (check class slides)

a) Create formal QA test plan (consult QA class material). Basically, it has to contain: 1) Test objectives: max 0.5 pages

2) Hardware and software setup: max 0.5 page

3) Feature to be tested: max 0.5 page 4) Actual test cases: 3 test cases and results of testing them on your system: 1 page

You must provide test plan and test summary in the format (e.g. form) allowing easy reading and analysis by management e.g. in a table format like presented in the lecture.

Suggested format for QA Table columns are: test #; test title; test description; test input; expected correct output; test results (PASS or FAIL for each tested browser) 5) Perform the testing as per plan above and record the results in a form above.

6) Apply the above test on 2 browsers of different type and record it in the above table

2.5 Code Review

By now you should have chosen a coding style. In the report state what coding style you chose.

Chose the code (substantial portion of it) related to the feature you used for QA and usability test. You need to submit an example of the code (or part of it – 2 pages or so MAX) for its function to be peer reviewed, and document this as follows:

1) One team member should submit code to other team member(s) for peer review. 2) Peer review should be performed by other group member(s) (1 review is OK).

3) Peer review is to be done by e-mail and comments are to be included in the code 4) Submit listing containing the peer review and commented code and communication related to this in your Milestone 4 document

Important: It is critical that code reviews are friendly and helpful, intended to help and education, and not to criticize. It is strongly suggested that you use peer review in the development of the whole system.

2.6 Self-check on best practices for security – ½ page

1) List major assets you are protecting

2) Confirm that you encrypt password in the DB

3) Confirm Input data validation (list what is being validated and what code you used) – we request that you validate search bar input;

2.7 Self-check: Adherence to original Non-functional specs

Copy all original non-functional specs as in high level application document published at the very beginning of the class and then for each say DONE if it is done (which is expected and required); ON TRACK if it is in the process of being done and you are sure it will be completed on time; or ISSUE meaning you have some problems and then explain it.

Note: you must adhere to all original non-functional specs as published in the original high-level specification document. Failure to do so may cause reduced grade.

3 Submission

Team lead submit Milestone 4 document to Canvas by due date 4 Grading criteria Your document needs to be well-written, well-organized (formatted) and reads well.

Grading is based on cohesiveness and completeness.

1) Title page 10 points

2) Product summary 10 points

3) Usability test pan 20 points

4) QA test plan 20 points

5) Code review 20 points

6) Best practice for security 10 points

7) Non-functional requirements 10 points Total: 100 points