CIS*3750 System Analysis & Design in Application

Lab Deliverable 5: Final Project Document.

Due Date: Friday, April 3rd, 2020 by midnight

Weighting: 10%

The following assignment must be completed as a group and submitted within your group's RedMine Wiki. Assume that your Wiki is a living document that your client will be using to check up on your progress and keep you accountable for your work. As such, the document should be **professional, organized, easy to follow, visually appealing, consistent and void of spelling or grammatical errors**. Failure to follow these standards or to provide all of the necessary deliverables (described within this document) will result in a deduction to your grade.

Keep in mind that a 70%-80% will be given to groups who **meet** the expectations as documented here. For higher grades, groups should strive to go above and beyond the deliverable requirements. In-depth analyses and reflections, extra subheadings, multimedia (pictures and videos), typographical emphasis and bullet points are all fantastic ways of making your Wiki stand out.

Learning Outcomes

- Draft well written requirements that are thorough, specific and measurable.
- Categorize requirements based on their importance to the overall system development
- Estimate the time required to complete specific requirements
- Identify requirements vs instructions and know the difference between the two.

Report Details

Each of the following sections are to be submitted as separate articles within your groups Wiki page (Except for section A, which will be included as a separate .csv file). These articles will be used throughout the course, and will provide a foundation for continued software development, so be sure that you write them clearly and with as little ambiguity as possible.

Part A - Final Document 100%

This document will be an integrative summary report on all of the deliverable documents on your wiki, plus the work you will do for the remainder of the semester. The integrative quality is critical, this document should be a flowing summary of the work that you have done. There must be even transitions from one topic to another and the content should be informative and interesting.

General Recommendations:

- Wrap up your development efforts and try to park them at a reasonable point. If you don't think that you can finish features in time to write about them, that is a good place to stop.
- Don't just rehash your previous deliverables, I am looking for refinements on your past work and insight into the problems you faced.
- If your group has corrected problems from past checkpoints, outline these changes in the document, I may give additional credit if you lost marks before.
- Document substance is much more important than length
- Aim to make the doc CLEAR and EASY to read.
- Did you produce any other documents that like a manual? attach it as an appendix.
- Find a way to get all of your group members involved in the writing, but one person should be in charge of editing the final document. This will help keep the tone and formatting consistent.

The below section headings must be included in your final document. The sub-headings are a non-exhaustive set of topics that you should address within each section; you don't have to cover all of the sub-headings and you are welcome to choose some of your own.

Section A: Intro/Overview

- Title Page, Table of Contents
- Client/Team Details
- Executive Summary
  - A one-page pitch statement, like the one presented in the CBase talk. This should be written in plain language, so that almost anyone could read it and have an idea of what your system does and why it would be helpful for your client. Should explain the problem, define target users and state project goals.
- Change Log
  - Draw my attention to any previous deliverables that you have improved on and where they are included in this document. Make your case for additional credit.
- User Groups, Personas, and Use Cases
  - What were the major findings from your Design Thinking Session and User Stories? Did the prototype and implementation you created reflect this initial work? Why? Include the User Stories you reference where necessary to make your point
- Requirements
  - Summarize your most important requirements, the full requirements table you submitted can be included as an appendix section at the end of this doc.

Section B: Prototyping

- Paper Prototype
  - Describe the process you used for creating your prototypes:
  - Include some pictures of your prototype
  - How did you decide on the user interface? What interface patterns did you use and why?
- Usability Tests
  - Describe the usability tests that you performed
  - Did your testers do things you didn't expect? Do you feel that your design decisions for the UI were correct?
  - What were the major findings from your paper prototyping sessions? How did you address the feedback that you were given?

Section C: Implementation

- Remote Work Strategies
  - How did you adapt to a remote work environment? What tools or resources did you use?
  - What impact did this have on your project?
  - What did you learn about this experience and how would it affect future projects you're involved with?
- Development Environment
  - Include a description of your environment and how you build/deploy your code.
  - Was the environment you chose easy to work with?
  - Did you use any 3rd party tools or libraries? Were you happy with them?

  - **Provide instructions for retrieving your code (i.e. zip uploaded to Gitlab)**
  - Provide class, ERD diagrams and any additional documentation.
- Technical Design
  - Describe the structure of your implementation (major classes, interfaces, etc.)
  - How did you model your data?
  - How did you resolve technical challenges?
- Time Estimation and Schedule
  - If you did scheduling, was your schedule accurate?
  - What features were you able to implement, which were mock-ups?
  - What major work remains to make your system 'complete'
- Testing
  - Summarize your overall testing strategy.
  - Did you uncover any issues in testing? what were they? Were you able to fix them?

Section D: Appendices (optional)

- Anything that is verbose and long (API docs, extra diagrams, etc.)
- Requirements Table
- Full Schedule
- Test Plan/Cases