CIS*3750 System Analysis & Design in Application

Lab Deliverable 3: Requirements and Use Cases

Due Date: 1:00 PM Friday February 14th, 2020
Weighting: 10%

The following assignment must be completed as a group and submitted within your group's Gitlab 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**. **Also, all submitted files that are posted to the wiki should contain a header or cover page with your groups section/number and your group name (if you choose one).** 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

- Introduce your team members, roles and your 'company'
- Describe your client, as well as their needs and challenges
- Perform a usability and cosmetic review of digital signage on campus
- Define Personas representing fictional user groups of your system
- Build User Epics and decompose User Stories related to your Personas

Report Details

Each of the following sections are to be submitted as separate articles within your groups Wiki page. **So after submission, your Gitlab wiki will contain a title page linking at least 5 articles.** 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.

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 posted 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.

A. **Specification Conformance** **[1 Mark]**
   Did your group conform to the guidelines set out in the initial paragraph and details? Also: Does your wiki have a title page? Are the written parts of your wiki in sentence and paragraph form? Do your article titles match the checkpoint headings?
B. **Definitions & Concepts** **[3 Marks]**
   What domain specific terms or concepts should be defined for your project to help the reader? This should also include terms that apply to the process of using your agenda, for example, what is content moderation? Any terms you want to use in your requirements should be defined, for example, what is a service, event, article, etc?
C. **Requirements** **[6 Marks]**
   Based on your project goal, the description of the your clients needs, the available literature and information gathered during classes and labs, develop a set of requirements for your project. That is, what are the things that your system will do? Some terms or abbreviations may need to be defined for clarity, remember this document might be read by individuals with drastically different backgrounds and disciplines. Classify your requirements based on the MuSCoW classification system and select which actor (not persona) each requirement applies to, *these actors should be defined in your definition section (for example; Admin, Moderator, and Student)*. Use category labels ‘Must’, ‘Should’, ‘Could’ or ‘Wont’ and do not include the apostrophes. Each requirement should also be ranked and all dependencies should be identified, as discussed in our class slides. The document should also list time estimates (in days) for each functional requirement. The smallest allowable estimate is 0.5 days, and only integer multiples of 0.5 are allowable. If a given requirement depends on more than one requirement, only list the requirement with the highest priority level, e.g. if 10 depends on 3 and 7, only list 7 (where 7 has a higher or equal priority than 3).

   Your team should aim to complete at least 80 requirements. This is far less daunting than it may first seem! If you are running out of ideas, revisit your user epics and user stories and incrementally decompose them into smaller and smaller chunks, no chunk is too small when it comes to requirements. Your requirements list should be complete, consistent and measurable and should include both functional and non-functional requirements. Be sure to remove blank spaces in the table as well as the end of the file and do not colour or otherwise format the file.

   Your requirements MUST be submitted as a .csv, based on the provided template, created a wiki article named after this section and upload the .csv to that article. Do not

change or modify the template in any way. This file must contain a header row including (in this order) the labels; ID, Category, User, Requirement, Dependencies, Priority, TimeEstimate. Each Requirement should have a unique ID.

D. Use Cases **Optional**

This part of the checkpoint is optional, marks will not be deducted if your wiki does not contain this article or if it is empty/incomplete. If you have made good progress on your requirements, you may find this task useful for your upcoming design and specification tasks.

Build 5 Fully-Dressed Use Cases, offering at least 3 alternate scenarios for each. Each requirement should include a summary, including actors, pre-conditions, core path, alternate path, and post-conditions as shown in the Slides for Lab 3. Try to pick high priority requirements for this exercise, or requirements that couldn't be further be broken down without becoming instructions, such as "Admins must be able to grant access to the content moderation system".