**Please submit a single PDF that contains your team's Requirements Documents.**

Your requirements submission will include the following:

**1.  Contents**

* Requirements. Neatly organized with unique identifiers and a priority. The priorities also get listed by ID in a table (next bullet).
* Requirements Priority Table. Prioritize everything as "Must Have", "Need to Have", and "Nice to Have".
  + "Must Have" - *Cannot* solve the problem without it. Basically, there's no reason to do the project if we don't get the "Must Haves".
  + "Need to Have" - *Really need it*, but there is a possible workaround if not completed *or* it could be done in the very near future after the Must Haves.
  + "Nice to Have" - *Want it*, but can wait until all of the higher priority items are done. Without these, the system still works, problems are getting solved, and no one is hugely impacted.
* As many decision tables as needed to cover the conditions in the requirements and related actions
* As many use case diagrams as needed to cover the requirements, actors, etc. Include a written descriptive use case flow for each diagram that explains the flow of data. Write the description so that someone unfamiliar with use case diagrams can read the use case flow and follow it visually in the diagram. See the example submission below.

**2. How to Produce Your Diagrams**  
  
Draw your use case diagrams in Dia <http://dia-installer.de/>. You can use something different if you don't like Dia. Dia is free, cross-platform, easy to use, and produces good results. Do *not* draw your diagrams on paper and take pictures - your diagrams must look great, follow the correct vocabulary of use case symbols, and be a great way to discuss the use cases, flow through the system, and the actors' relationship to those use cases. Your team will not be happy with the assignment grade if the use case diagrams don't look good.

**3. Professional Presentation**  
  
Obviously, your requirements need to exhibit excellent grammar, spelling, formatting, etc. Make your English teacher proud here. Page numbers, headings, etc. All of that matters here. When a customer is paying for something and they get a document back that looks like you wrote it 5 minutes before the meeting, your reputation suffers and the customer is more likely to fire you.

**4. Submission Requirements**  
  
Compile everything into a single PDF. Add your PDF to your shared team repository in the Documentation/1\_Project\_Plan folder. Call your file "coffee-project-requirements.pdf". *Each* team member will submit your team repo link in the D2L assignment folder. As before, everyone on the team will receive the same grade. The only exception is if an individual forgets to do their individual submission in D2L.

---------------------------

**Example Submission**

[Here is an example submission from a prior semester's Pizza project.](/d2l/common/dialogs/quickLink/quickLink.d2l?ou=%7borgUnitId%7d&type=content&rcode=usgq-27887978) This document is not perfect (the use case diagram should have been split into two or more), but it was A-level work and demonstrates the polish, detail, and effort I expect for a top score. As before, do not copy this document! Do your own work.

---------------------------

**GRADING RUBRIC**

10% - Professional, well-organized presentation (there better not be any spelling issues or bad grammar!)  
      - Consider a cover page, table of contents, etc. Add page numbering, footers, etc. Do all of the things your English teacher taught you about writing professionally.  
      - Past students who have done great on this bit went to the University's writing center for help.

30% - Complete, thorough, detailed, and well organized requirements

10% - Requirements prioriziation is complete *and* makes sense. If everything is "Must Have", then nothing is priorized. Assign priorities with care. Include a table

25% - Use case diagrams are detailed, thorough, professional, follow the proper use case diagram vocabulary, and can be used when explaining the actors and overall data flow of the completed system. I am not telling you how many use case diagrams you need. That is up to you to decide what you think is enough. I do advise not making one giant diagram that covers it all nor should you make 100 diagrams that don't cover enough. Each diagram needs a descriptive use case flow that explains each piece of the diagram. Do not write that as a giant fat paragraph... <https://www.writingclearscience.com.au/bullet-point-lists-versus-paragraphs/>

25% - Decision tables are detailed, thorough, cover the various conditions, rules, etc. you expect the system to manage. I am not telling you how many decision tables you need. That is up to you to decide what you think is enough. I do advise making sure each table is logically related and doesn't combine a bunch of unrelated decisions into a giant table.