PRDv2: Your Living Requirements Document: A Shared Google Doc

• Authors, Team, Project Title
• Intro: problem, innovation, science, core technical advance (3+ pages)
  – Define project specifics, team goals/objectives, background, and assumptions
• System architecture overview
  – High level diagram (1 page)
  – User interaction and design (1+ pages) – ie detailed design
• Requirements (functional and non-functional)
  – User stories or use cases (links) → 20+ for PRDv2 prioritized w/acc. tests
  – Prototyping code, tests, metrics (10+ user stories): github commits/issues
• System models (1+ pages)
  – Contexts, interactions, structural, behavioral (UML)
  – Use cases, sequencing, event response, system state, classes/objects
• Appendices - Technologies employed
Your Project Design: PRDv2

• Architecture (hardware/software)
  – Evolve your overview picture from PRDv1 to provide significantly more detail and any updates or changes

• Detailed design
  – UML diagrams of primary data structures that comprise the system architecture connected via their associations (if any)
    • Ensure that each "class" is balanced in terms of cohesion & coupling
    • Annotate with pre/post conditions when appropriate
  – Sequence diagrams
    • synchronous and asynchronous for key interactions between classes
      – At least 3 different interactions
    • User interactions with the system
      – At least 3 different interactions
      – Can be a human user or a machine user (API) interaction
        » Event response, updated application state
      – If you have a user interface: Provide mockups for primary UIs
PRDv2 User Stories / Use Cases

• Revise spec to add detail to the functional specification to match your design
• Add user stories and break up the stories you have into finer grained stories
  – Provide UML, sequence diagrams, dataflow diagrams
  – Goal: a CS senior should be able to take your doc and implement the project
• For each fine-grained story, provide a description and acceptance test
  – Provide time estimates (1 person-hours) for each story implementation
    • Ensure you can finish the implementation in the time you have (this/next quarter)
  – Prioritize tasks to have a complete prototype by the end of this quarter
    • Focus on the externally facing interfaces, mock out what you cannot get to
  – Write unit tests to implement tasks for mandatory tasks
    • Document these tasks (autogen the documentation/usage)
  – Add trello/pivotal task links (titles must match) to PRDv2 for each story
• Prototype designed mandatory tasks; add github commit ID/link to PRDv2
  – Github must have unit tests, documentation (for anything without unit tests), and prototyping implementations for each story in Sprint
• If you have a user interface
  – Provide mockups that are tied to the functionality described in 1+ components