

Project Assignment 1: Software Requirements Specification

Out September 14th; Due September 28th

Congratulations! Your software development firm’s acquisition of SeeFood was successful and the CEO is excited to explore it’s newly acquired AI system. In trying to finally break the barrier for “smart eating”, where a user can take a picture of food and instantly analyze its nutritional talent, your team has been tasked with implementing the core technology of analyzing a photo to determine if food is present in it.

The first step in developing a software system is establishing a requirements specification. The specification describes the goals of your project, analyzes its most important software qualities, and describes its functional requirements and major use-cases.

Description

An initial meeting with the CEO established the following minimum set of functional requirements for a proof of concept, due December 2017:

- The software should be implemented as a desktop application (windows, mac, or linux), as an iOS application, as an Android application, or as a web application that runs in any modern web browser.
- The software should make it easy for a user to select an image, either saved to a device or taken in real-time by a phone camera or webcam, and then submit it for analysis by the AI.
- The AI should provide a response in a nominal amount of time.
- The software should allow a user to submit multiple images at once.
- The software should allow the user to browse **all** past uploads made by other users, in a gallery, that includes a label indicating if the image is or is not food.
- The software should have a pleasing UI for image upload, image classification, and past image browsing.
- The UI for image classification **and** the gallery should include some notion of the “strength” or “confidence” of a food / not-food decision, presented in a graphical, non-technical way to the user.

Consider further the following implicit environmental and external requirements:

- The product should be easy to use, particularly for those who are not expert computer users.
- **Robustness** is a key software quality: presenting buggy clients to your CEO is not a good strategy for job promotion.

- The AI system itself should be hosted in the cloud, on an Amazon EC2 virtual machine. It may also be desirable, but it is not required, to maintain a database of the images uploaded and their class labels on the EC2 VM. Note that a database **does not** need to be something SQL-like: saving uploaded images to directories, with metadata about their classification, is reasonable.
- The software should support all common kinds of image formats, and the user should not be expected to alter the resolution or color properties of an image.
- For the image gallery to be pleasing all images should be of similar resolution.

Beyond these additional requirements, you are free to define the specific features and look and feel of your clients however you like.

Deliverable 1: Software Requirements Specification (SRS)

You will write a software requirements specification document. The SRS document codifies the specific overarching purpose, goals, requirements, and conceptual design of the system. Include a section that covers each item below:

1. A statement of goals for the product. What *is* this software about, in a few sentences? What is the vision for the product you are creating?
2. The essential (top 5) software qualities the software should exhibit. Explain why these software qualities are so important to your product, and why the remaining qualities are less important. Think realistically: of course, they are all important, but you need to think about the type of software you are writing and the qualities you will strive to achieve during your development period. Further consider that the submission is a proof of concept to your company CEO.
3. A set of functional requirements for your system. Describe what the functional requirements are and why they are included. Assign it a priority level with justification. There should be 10-15 essential functions with priority level between 1 and 5. There should be a mix of priority 1-2, 3-4, and 5 requirements, with slightly more 3-4 priorities than 1-2 or 5.

Before presenting the requirements, explain in your own words what a priority 1-2, 3-4, and 5 item means. Is priority 2 twice as important as priority 1? Is priority 5 orders of magnitude more important than a priority 4 functional requirement?

4. A high-level description of the product, including discussion of the conceptual organization of the system. What categories of functions exist, and do you have a picture of system components or modules? If an informal diagram or figure help conveys a high-level description, include them. It is **okay** for this description to change over time.
5. Provide a set of 4 different use-cases for the system. For each use-case mention the actors involved and list the steps each actor must take in order to complete the use-case.

6. A UI prototype. Provide diagrams containing rough sketches of your product's user interface. These diagrams should depict the major UI used to complete the use cases you are submitting. You should submit no less than two (2) UI diagrams. The diagrams can be hand-drawn, drawn by computer, or can come from screenshots of an actual programmed prototype if you like. If a window leads to a dialog box, drop-down box, etc., perhaps this should be included as a sub-diagram.

Your diagrams do not need to be works of art to get full credit, but they should be legible and reflect some forethought about what options will need to be shown and how the user will interact with the software.

Deliverable 2: Technology Sanity Check

Also due is evidence that you have successfully deployed SeeFood AI to an Amazon EC2 web server instance. Simply provide screenshots of the SeeFood AI running on EC2 and evaluating whether some image contains food. Include a screenshot of the image and a screenshot that offers evidence the analysis is being done on an EC2 instance.

Deliverable 3: Group and Customer Meeting Artifacts

Finally, submit records showing that you had at least one group meeting to discuss the project and the SRS document. Records may include the minutes of a meeting, or e-mail correspondence. Limit your records to two pages.

Summary

Submit the following as a single hard-copy document **as a single pdf**:

1. **The SRS document.** Page limit: 8-12 pages, inclusive of diagrams and formatting. While no particular format is specified, it will be obvious to Derek and Ning if you opt for large fonts or do formatting tricks to artificially inflate page counts. This will detract from your grade.
2. **The group and customer discussion artifact.** Note that these records do not count towards the 8-12 page limit of the SRS document.
3. **Evidence of the technology sanity check.** Annotating your screenshots or providing brief captions or explanations may help.