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?
   1. Our software group wants to take the newly acquired SeeFood AI and create a cloud-based Android experience that identifies food in user-supplied images. The application will store a gallery of all past image uploads from all users, and display the AI’s result in a user-friendly format. The focus of our application will be on the users first and growing our user base in order to collect data to create a more robust AI in the future, with the potential to analyze the nutritional content of food.
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.
   1. Correctness
   2. Usability
   3. Reliability
   4. Robustness
   5. Performance
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?
   1. Implemented as an Android application (No later than Android API 21 to maintain decent backwards compatibility)
   2. User should be able to select an image saved to a device for analysis by the AI (via native android gallery application, support PNG, JPEG, GIF, and BMP formats)
   3. User should be able to take an image on a phone camera for analysis by the AI (Launch Camera app)
   4. User should be able to submit multiple images at once
   5. User should be able to browse all past uploads made by other users (probably on a separate screen? So we’re not downloading images just by launching the app)
      1. Past uploads should be labeled food or not food
      2. Past uploads in database on VM
   6. Image classification and gallery should include the strength/confidence of a food/non-food description
      1. Confidence should be conveyed in a graphical, non-technical way
   7. Android application with a user interface that enables a user to choose an image and determine whether the image is of food
   8. The image uploaded can be taken real time or uploaded from their gallery
   9. Store images for later access in a compact format
   10. Should process images in under 3 seconds
   11. The AI system itself should be hosted on an Amazon EC2 virtual machine
   12. Software should support all image types
   13. Limit images pulled from the EC2 instance in the android application in order to promote performance and minimize perceived delays.
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.

We will use an android application that interfaces with an Amazon EC2 instance. The Amazon EC2 instance will host an HTTP web server api, backed by a filesystem of images and metadata about the AI’s response. The web server will interface directly with the AI system and parse its responses into a user-friendly format and pipe this back to the requesting android application. The android application will upload images to the server via HTTP POST requests and will receive a JSON response containing information about the outcome and certainty of the outcome, and request a gallery listing of previous image uploads via HTTP GET Requests.

The Amazon EC2 instance will run an Apache web server to handle HTTP GET and POST request routing, along with mod\_wsgi to route said requests to a python application. The server will have the seefood-ai embedded, along with the required tensorflow, pillow, and flask libraries. A python flask application will expose a REST API with functions to handle these requests by interfacing with the tensorflow AI and underlying libraries as necessary. The following REST API functions will be exposed to clients (i.e the android application):

* /gallery [GET] – Return a listing of the gallery of previous image uploads, as well as whether or not they contain food. Can be optionally limited to a certain number of uploads.
* /analyze [POST] – Upload and analyze a collection of images (storing them in the image gallery as well), returning a list of results for each image run through the AI.

The android application will run on Android API Level 21 (to maintain decent backwards compatibility across devices) and interface with the Amazon EC2 instance via HTTP requests. The application will be allow the user to select an existing image on the device via Android’s built-in gallery application to be uploaded and analyzed, or allow the user to take a picture from their camera to be uploaded and analyzed. The application will be capable of displaying a gallery of previous uploads to the See-food AI from any user by pulling them down from the EC2 instance on the fly. It will display one image at a time by requesting it from the server to minimize perceived delays. On this gallery and after uploading an image, the application will display results from the See-food AI’s analysis of whether or not the image contains food, as well as the certainty of the AI, displayed as a percentage.

1. 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.
   1. Useful for anorexic people familiarizing themselves with what food is again.
   2. Useful dieting information (see food diet)
   3. Useful for foreign food
   4. Competing with friends
   5. Get an A in CEG-4100
   6. Making a database for chefs
   7. Competing/ familiarizing the public with AI
2. 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.