**What is SRS**: This will detail your product features and expectations.

Minimum requirements:

1) Title

2) Version [Version 0 would be what you submit on Oct 10]

3) Personnel of the project and their roles (names and emails)

4) List of abbreviations and notations, naming conventions and definitions

**1-6, all: Archivist. This person handles the administrative details, and also proof-reading and grammar, making the final document up to spec, rewriting all sections.  
  
8-9, 10: Requirements Writer. This person works on the functional requirements, basically the entirety of the proposal that is functional. Will need to communicate with all other members. The 'hard' part of SRS. Most important will be the description of features, which will be shared responsibility with Co-requirements Writer.  
  
7, 10-12: Non-functional Requirements Writer. This person works on user requirements. The 'soft' part of SRS.  
  
13-16: Project Coordinator. This person will basically do all the workflow and planning based on what Requirements Writer writes.  
  
17-18, (contributes to 8-9, 10): Co-requirements Writer. Complements the requirements writer by helping with half the stuff, focusing on technological implementation. Also does the Gantt chart.**

5) Table of contents and contributions (Who contributed what part, in a table format)

6) The Purpose of the Project

**SRS (9-11 pages):**

7) The Client, and other Stakeholders

8) Project Constraints (high level choices you have made or that are mandated by your

supervisor) and relevant facts. Examples are app versus web-based or limiting your

solution to certain scenarios. Please consult your supervisor if you have one.

9) Functional Requirements (formal list with priority ranking (P0-P3)). Include details about

what data is needed for building each function. This is the most important section and

needs to clearly state what you will build. Detail both backend and frontend features.

Please consult your supervisor if you have one.

**In descending order of priority, under each category:**

---User Account---

Legacy Code bug-fixing

As a user, the first thing you do is upload a file, your own model, private or public.

Sort by model types, download the results fast

Password Restriction

---Hosting---

PARALLELIZED for simultaneous load.

CRUD functionality and authorization for admin access to change database features, change something from private to public. Delete a row, cull database, etc. Can be an ADMIN PAGE.

BACKEND - TIMER TO TIMEOUT SUBMISSIONS, PREVENT DDOS.

--Features--

MATLAB / PYTHON INTEGRATION

COMPARE 2 Models, add a compare page.

SHARE ALGORITHM, include documentation details.

COMMENTING, USER INTERACTION UPGRADES

10) Data and Metrics (This is particularly important for ML/AI projects, but metric could be

important for any project with quantitatively measurable outcomes):

a. For each feature, explain what data, if any, you will use to train/build.

b. Links to dataset or a clear plan to obtain or simulate data

c. One or more performance metrics (accuracy, precision, recall, area under ROC

curve, area under precision-recall curve). Explain why you think the metric is

relevant. Also set expectations (Goal for the metric).

11) Non-functional requirements (refer to the template for a full list).

a. Look and Feel Requirements

b. Usability and Humanity Requirements

c. Performance and speed requirements

d. Security and Privacy

e. Legal

12) Risks and issues predicted.

**Project Development Plan (4-6 Pages):**

What is project development plan: Briefly, this is supposed to describe: “Who does what, when, with what tool” and your “what” should cover the requirements from the SRS, along with your workflow and communication.

13) Team Meeting and Communication Plan (including how you share documents and work on them together). Use of program management tools is optional but strongly encouraged.

14) Team Member Roles

a. Who is responsible for each of the functional and non-functional requirements (multiple people might work on the same component)?

b. Who is the coordinator/program manager?

15) Workflow Plan

a. How will you be using GitLab or GitHub, including branches, pull request, issue management?

b. Using agile methods is encouraged. You can use scrum and sprint planning in Jira. State if you will do this. We will not monitor your sprints.

c. Where do you store your data (especially if you are doing machine learning)?

d. Where do you run compute heavy tasks like training models.

e. What tool/method is used to achieve each of the requirements and achieve the performance metrics that were proposed in your SRS?

16) Proof of Concept Demonstration Plan

a. What will you demonstrate during your proof-of-concept demonstration to convince yourself that you will be able to overcome this risk? For example, one approach would be to have a mock website for a service, with limited backend. There must be code. You cannot just plan to show slides. Seek advice from your TA/instructor if you are unsure.

17) Technology

a. Specific programming language (front end and backend), coding environment. State if will use unit testing framework, why or why not. If your project is primarily software development (as opposed to research), we expect you to follow software engineering best practices including unit testing.

b. ML libraries (if relevant)

c. Will you use GPU? Any other relevant technology aspects.

18) Project Scheduling: Include a Gantt chart