Big picture stuff about creating a full education system

Goals: I would like to design the rest of the education system. This is high level design. No coding, no specifications of anything. I want this to be a discussion only. I want to discuss the big decisions that will allow us to build something will be fast to create initially and will also allow us to have maximum flexibility. Getting something up and running quickly and building it incrementally from there is crucial to success.

The system will be used in the following ways:

1. By independent students who buy a subscription to help them practice for the SATs.
2. By tutors for the same.
3. By teachers for the same.
4. By school systems for the same.
5. Possibly integrated into advanced, even AI-oriented teaching systems.

Functionality:

1. We will assume that the python exists to generate all the types of problems we want. We are not considering the process that creates that python.
2. We will use the python to generateproblems of all types.
3. We will run the python to generate many different instantiated problems of each type.
4. We will store the jsons in a database that will allow us to filter jsons on the basis of fields within the json.
5. We will create an AI program to test instantiated problems.
6. We will create a viewing screen to look at all the features specified in the jsons associated with the instantiated problems.
7. We will test the problems using an AI and/or manual viewing.
8. We will create a database that holds problem types in a tutor/teacher-specified hierarchy.
9. We will use the problem type db to support decisions about which problem type to serve to a student next.
10. There will be a rule base that a tutor/teacher/school system can customize to support decisions about how to serve problems in various situations.
11. There are several use-cases of how the system is used (and thus how problems are served to the student):
    1. Problems may be served into a test according to rules.
    2. Problems may be served to help support certain learning themes assigned by a teacher.
    3. Problems may be served because a student couldn’t solve the specific problem.
    4. Problems may be served that are similar to problems the student couldn’t solve.
12. We will create a customer database. This can be in a different DB than that chosen to store problem jsons, etc. For instance, this might use a SQL database. This will support:
    1. Creating a new customer
    2. Creating a new customer via an automated onboarding process.
    3. Tracking payments and other business-related issues associated with serving customers.
    4. Managing privacy, security, etc.
13. We will create a student database. This may be distinct from the customer database.
14. The student database will be associated with:
    1. Tracking each test and each problem the student has been given.
    2. Tracking info about the student’s performance both at a high level and at the level of each problem.
    3. Associating the student with any given teacher/school system/etc.
15. We will create a teacher/school system/tutor database to
    1. support problem hierarchies,
    2. give permission to them to look at student work.
16. I would like to choose hosts for the databases, problem servers, etc. that are
    1. easy to start with (preferably free).
    2. Safe – using dockerized containers for instance to ensure consistent performance.
    3. Scalable over time.
    4. Support best practices.