-If using a relational database, then manage no fewer than 4 and no more than 7 database entities and their relationships.

(You can use a non-relational database (Ex: MongoDB) to qualify as a stretch goal)

- Data access w/ JdbcTemplate or JPA.

(If you use JPA, that will qualify as a stretch goal).

- Spring Dependency Injection

- Spring Boot

- UI is HTML, CSS, and JavaScript.

UI may be rendered either client-side (@RestController w/ JavaScript) or server-side (via a templating engine like Thymeleaf).

- Authentication/Authorization

- At least one stretch goal!

(Research and successfully implement a complex technology we haven't covered in class.

Past examples include: Robust map integration, application (static content) caching, D3 (or any rich client-side graphic library), React/Angular/Vue, non-relational data stores).

- Robust testing for all backend components.

-These guidelines represent a minimum starting point. If you have a compelling case for alternative technologies: non-relational data stores, mobile development, hardware hacking, etc, then present it.

(Keep in mind that the more you stray from the curriculum, the more likely you will be on your own for research).

-We reserve the right to veto any and all deviations from these guidelines.