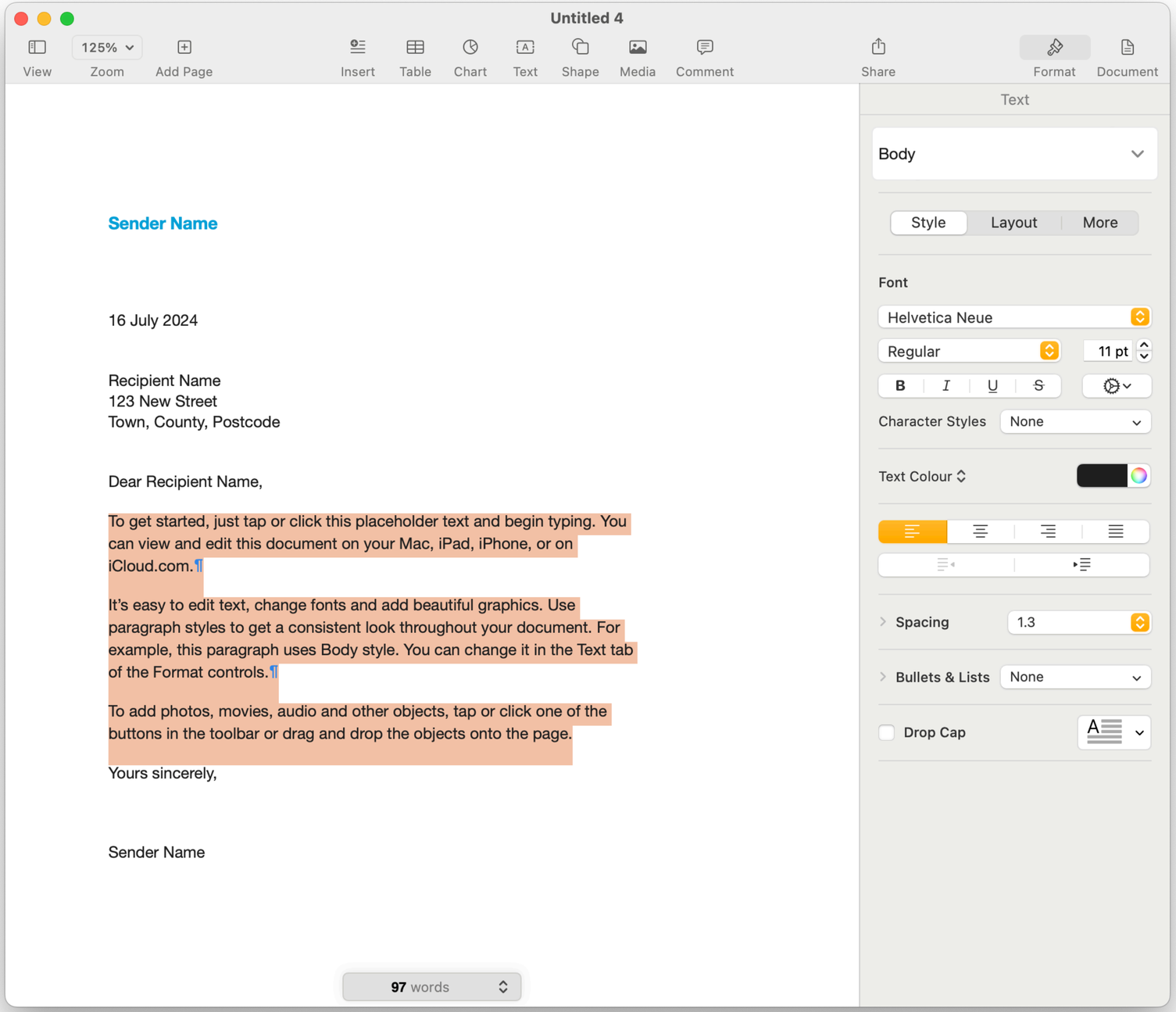
COMP350 - Software Design

Learning by building software

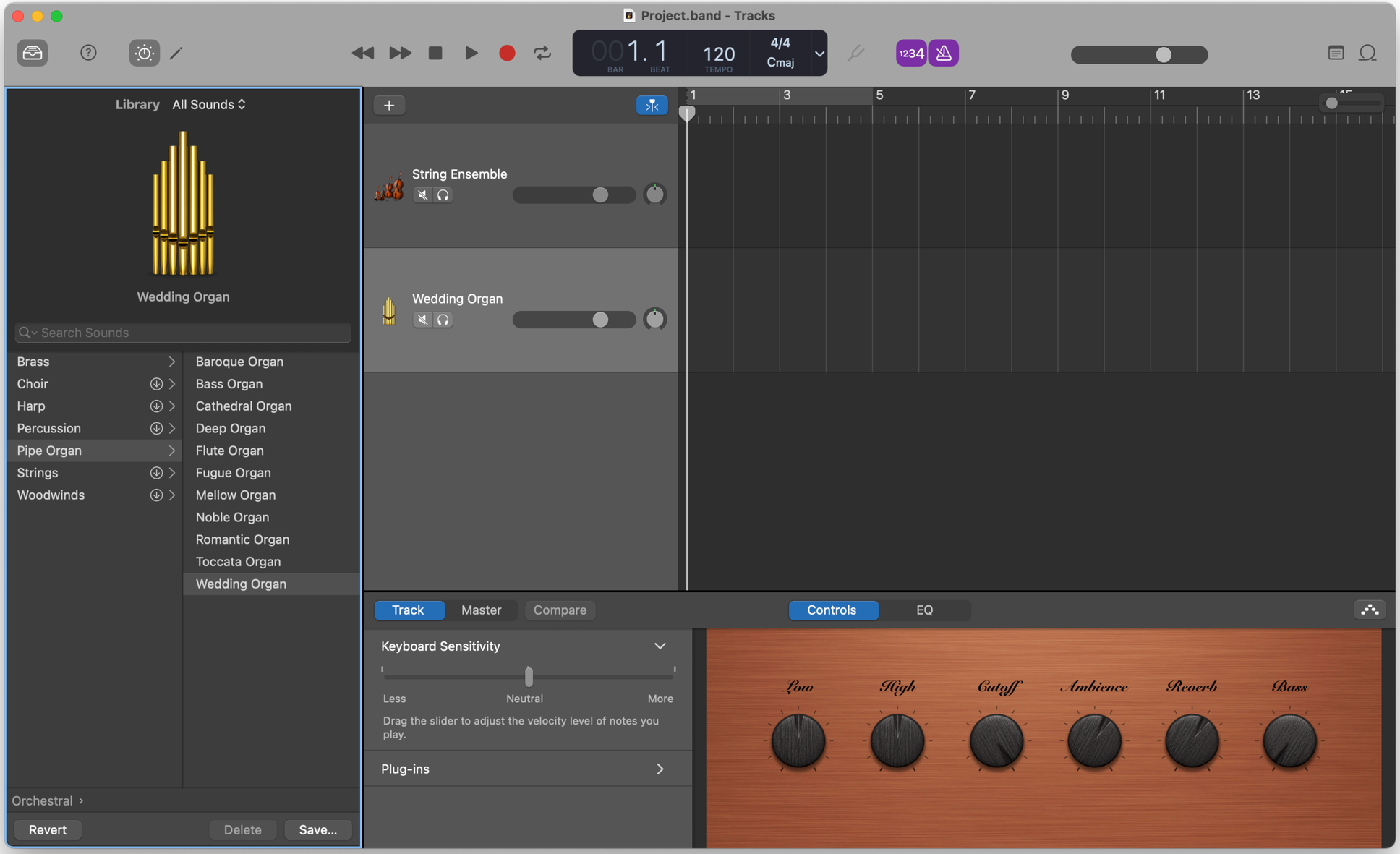
I notice …

I wonder …



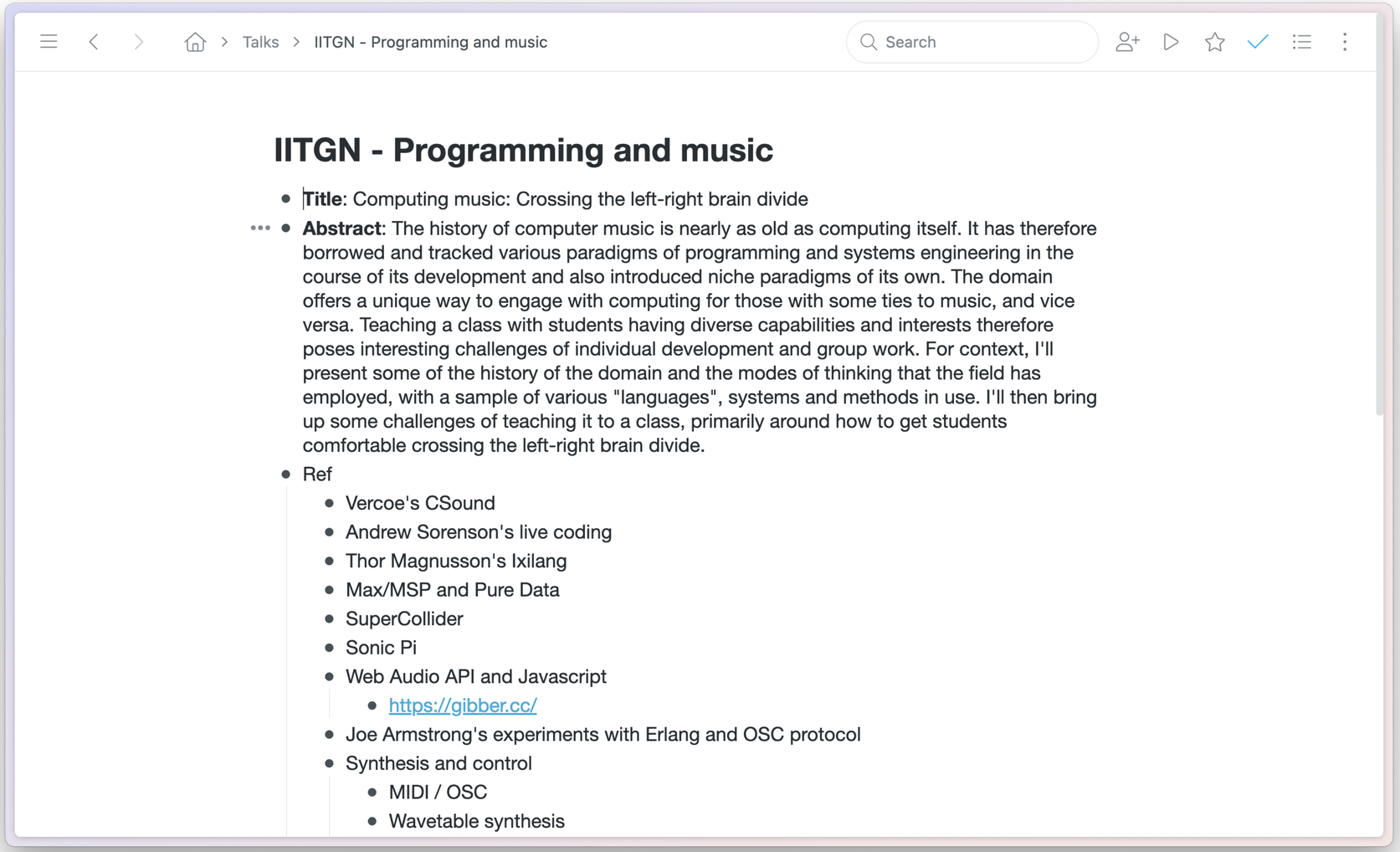
I notice …

I wonder …



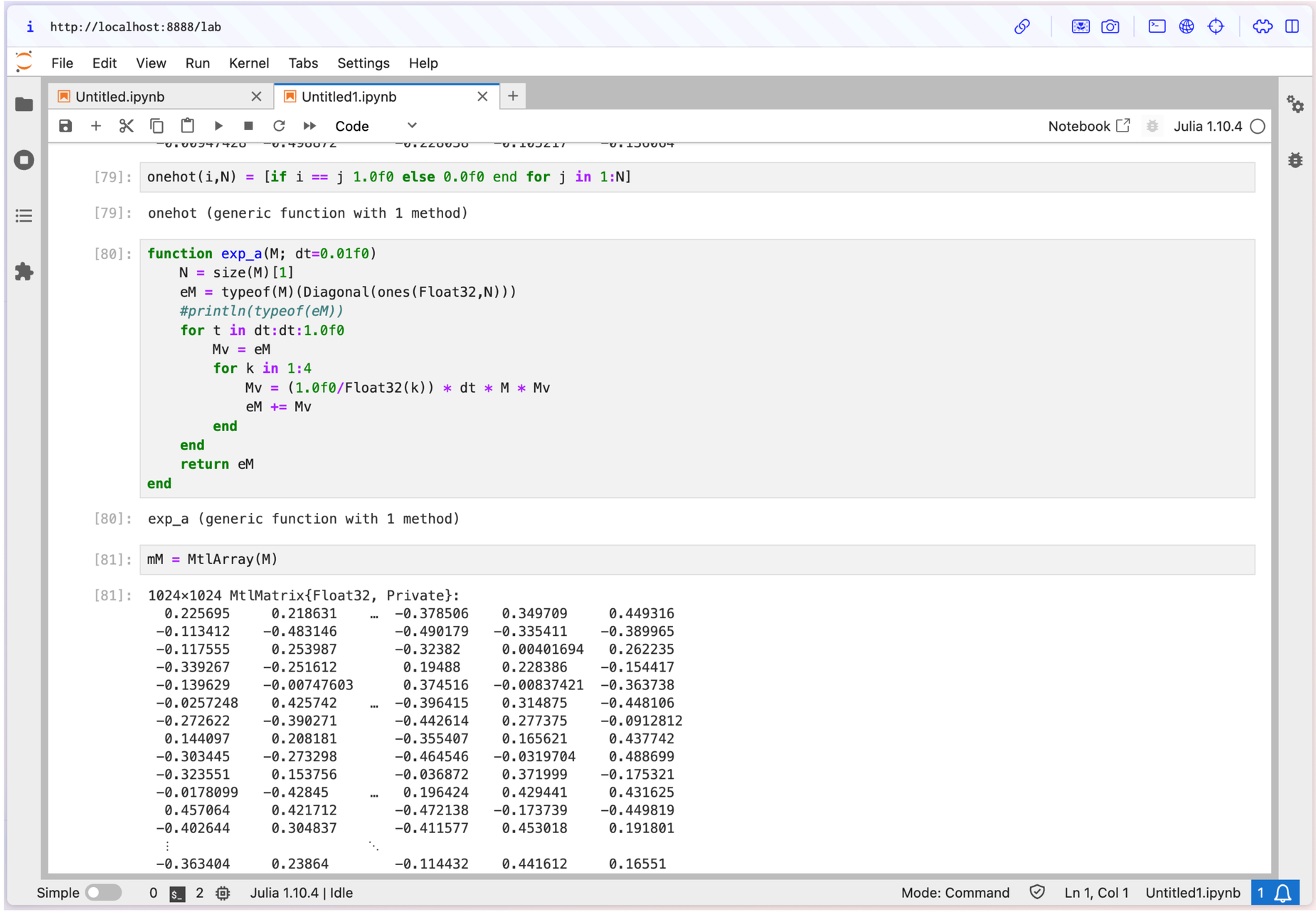
I notice …

I wonder …



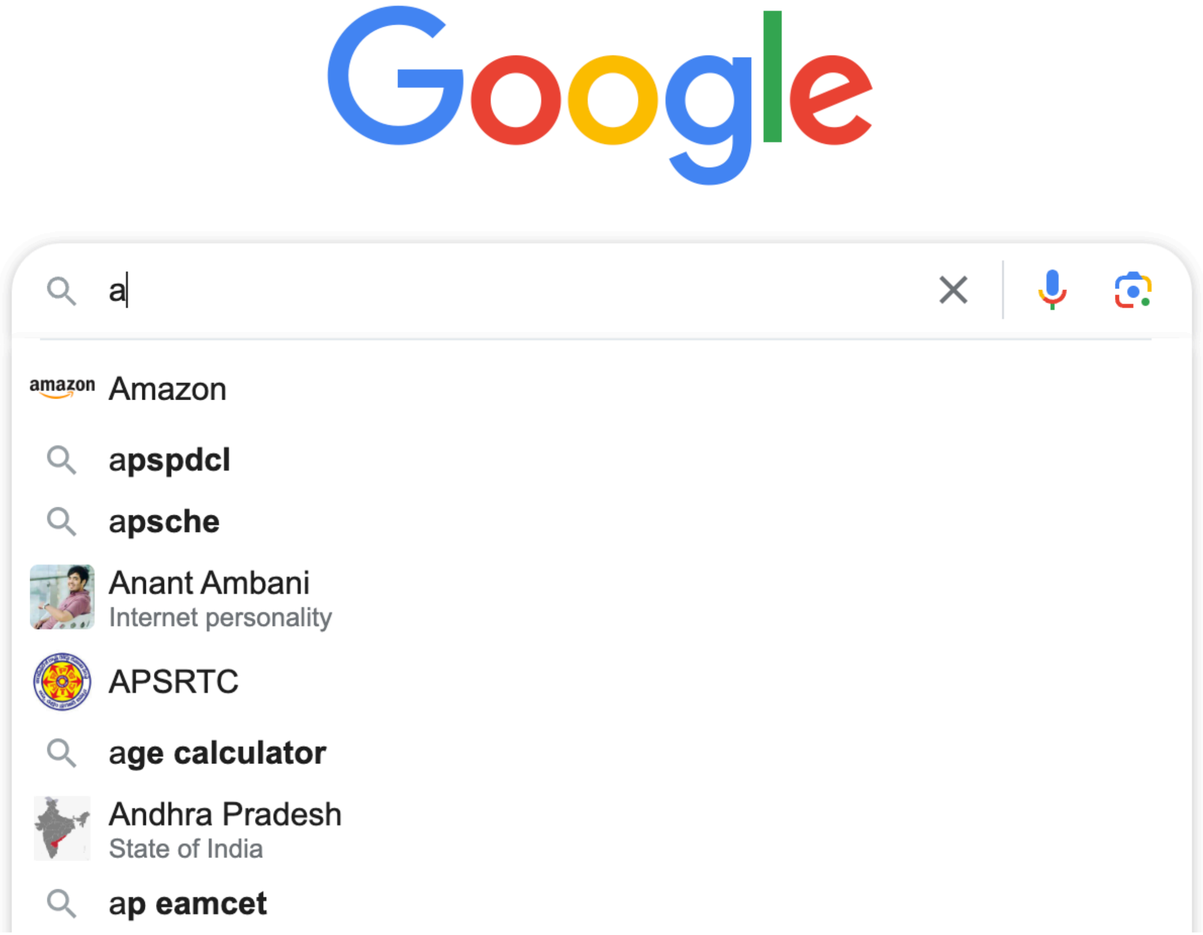
I notice …

I wonder …



I notice …

I wonder …



What does it take?

- Ideas
- Grounding and validation
- Planning and detailing of behaviour
- **Architecture**: Systems and components involved
- **Design**: Appearance and interactions
- **Code Design**: functions / modules / packages, their **interfaces** and **contracts**
- Coding, testing, debugging

What does it take? - Ethics

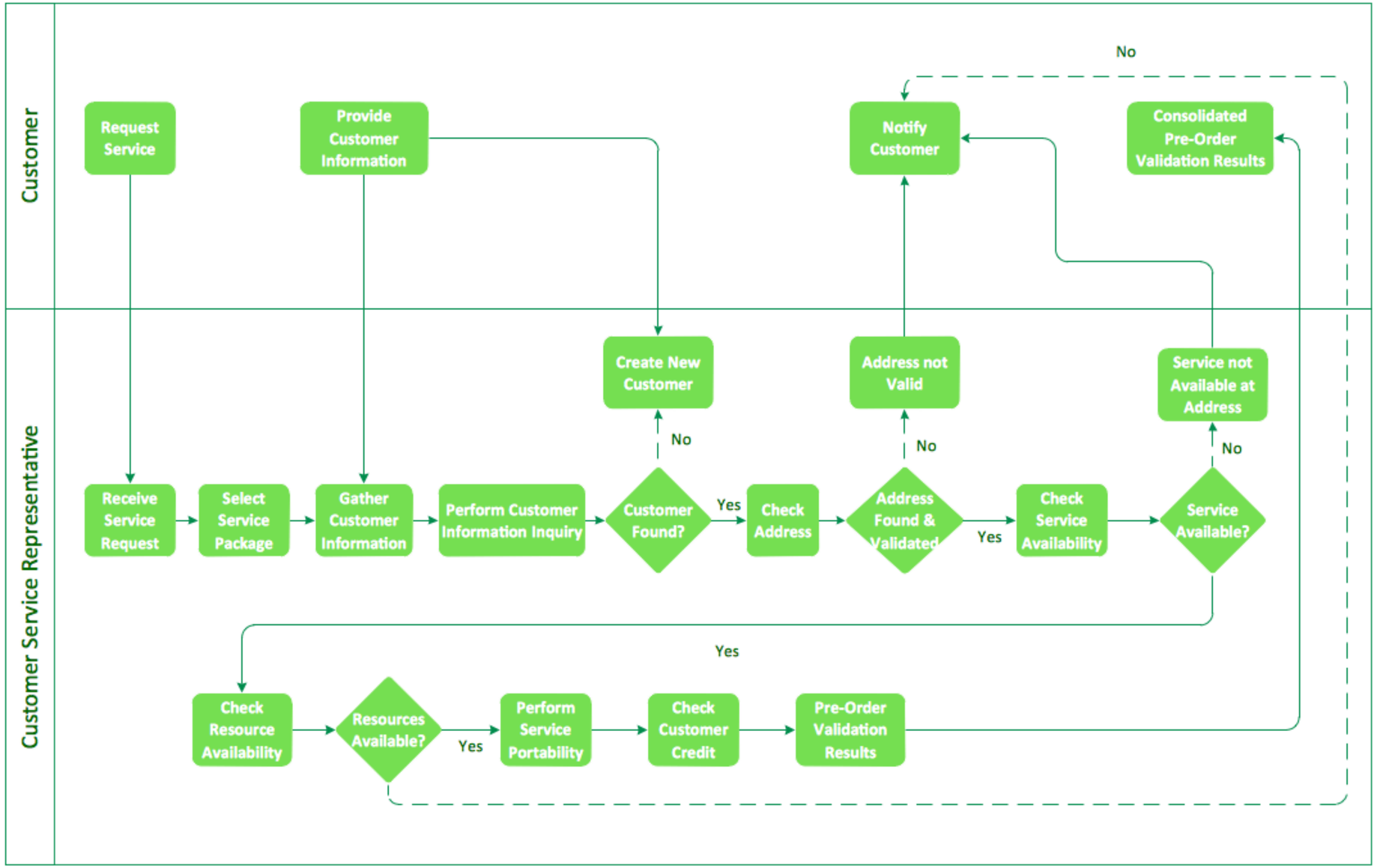
- Accessibility considerations
  - Readability, Colour blindness, Blindness, Physical disabilities, …
- Algorithmic biases and responsible design
- Correctness and consequences
- Team affordances and accommodations

Some questions to keep in mind

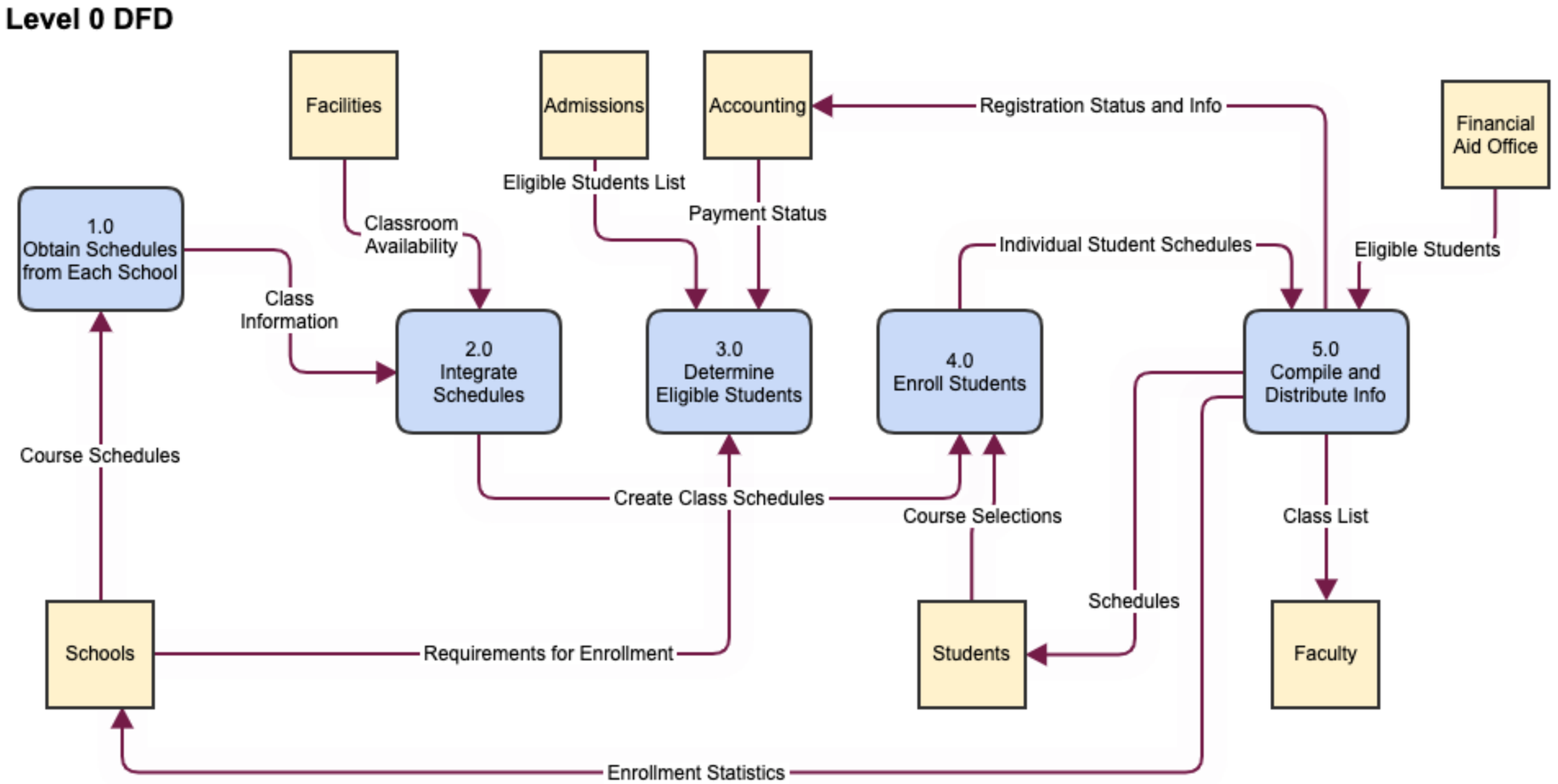
- **Who** is going to use your software?
- **What** are they doing with it?
- **How** are they going to go about doing it?
- **How** do we plan to find out and improve it?

Artifacts

Process flow diagram

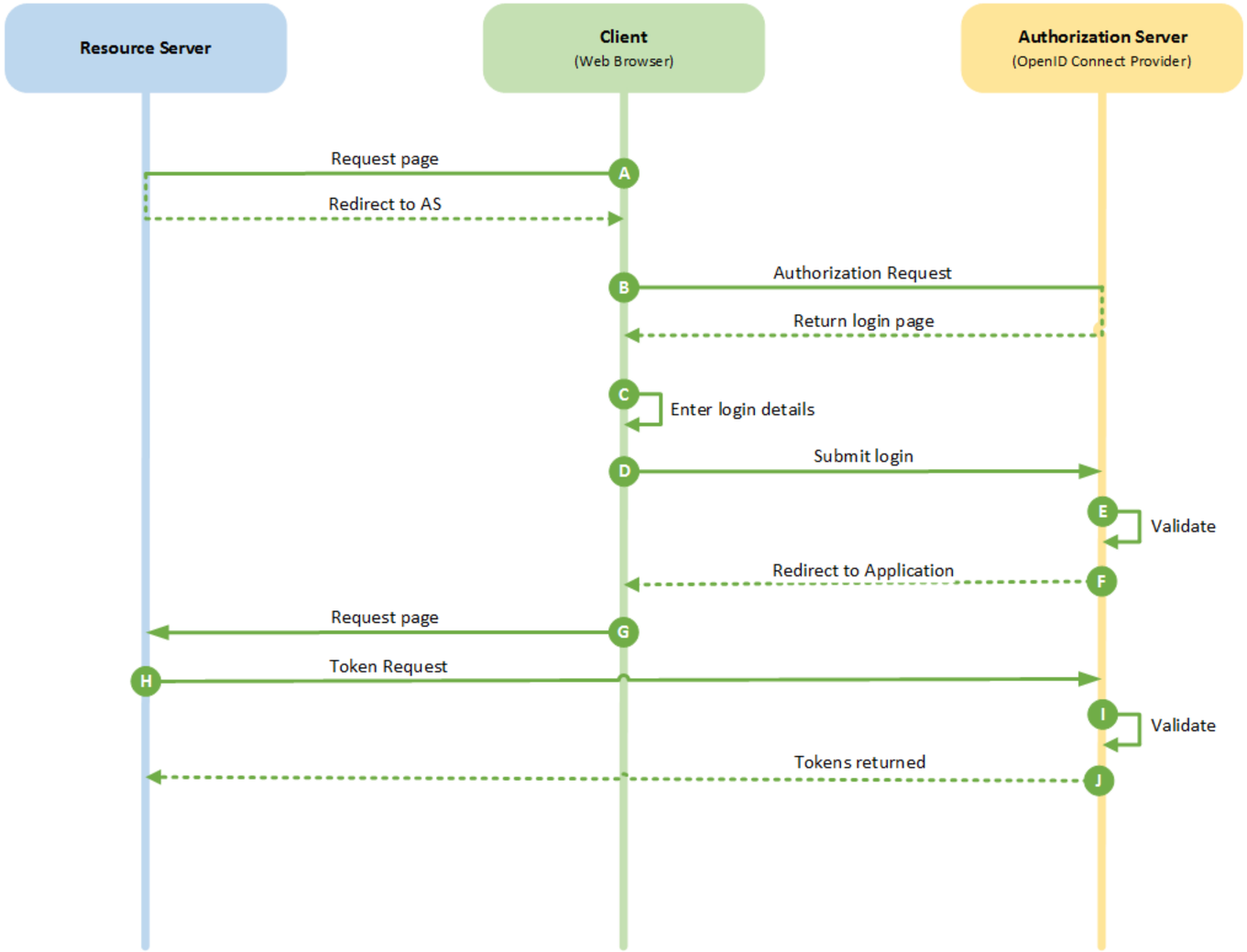


Data flow diagrams

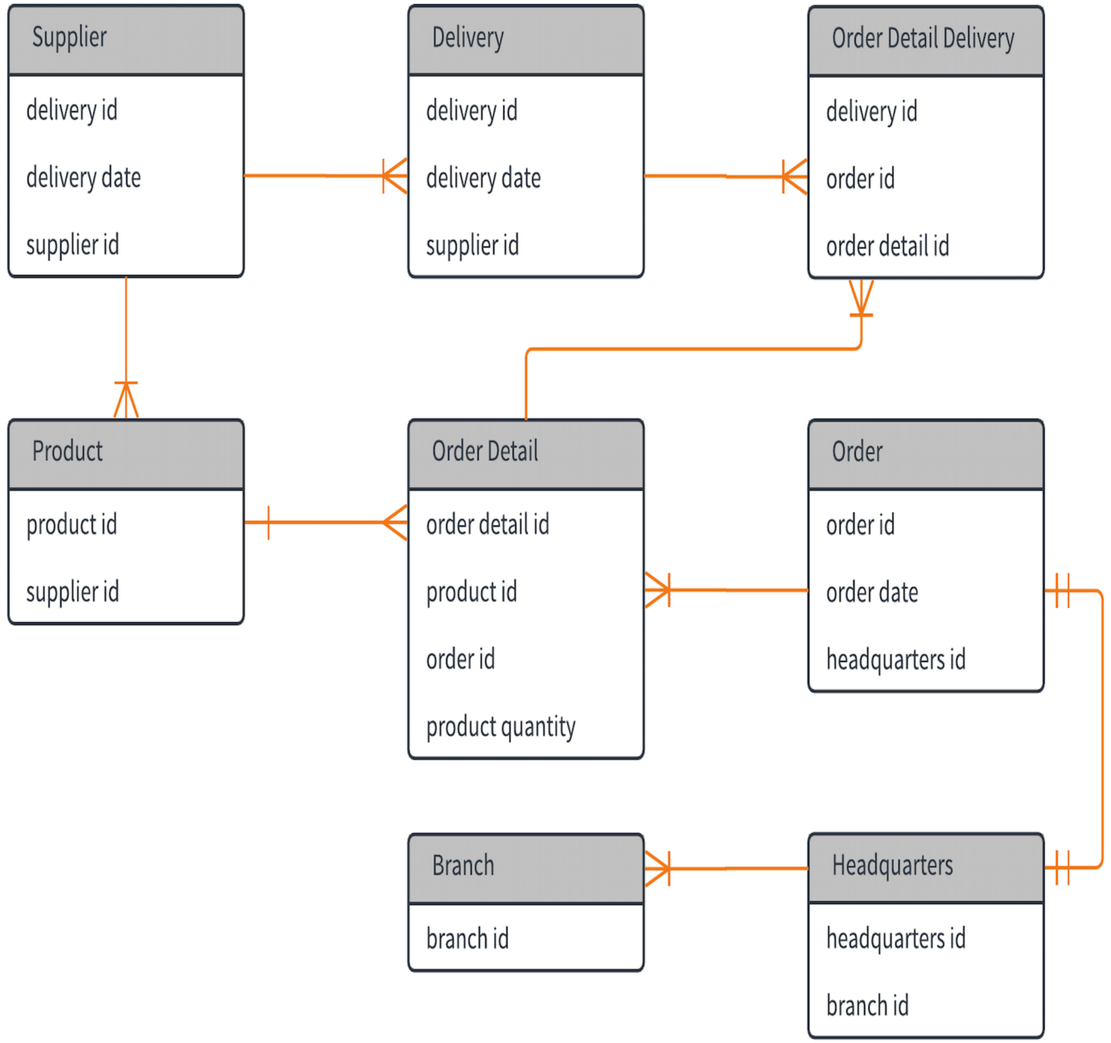


Sequence diagram

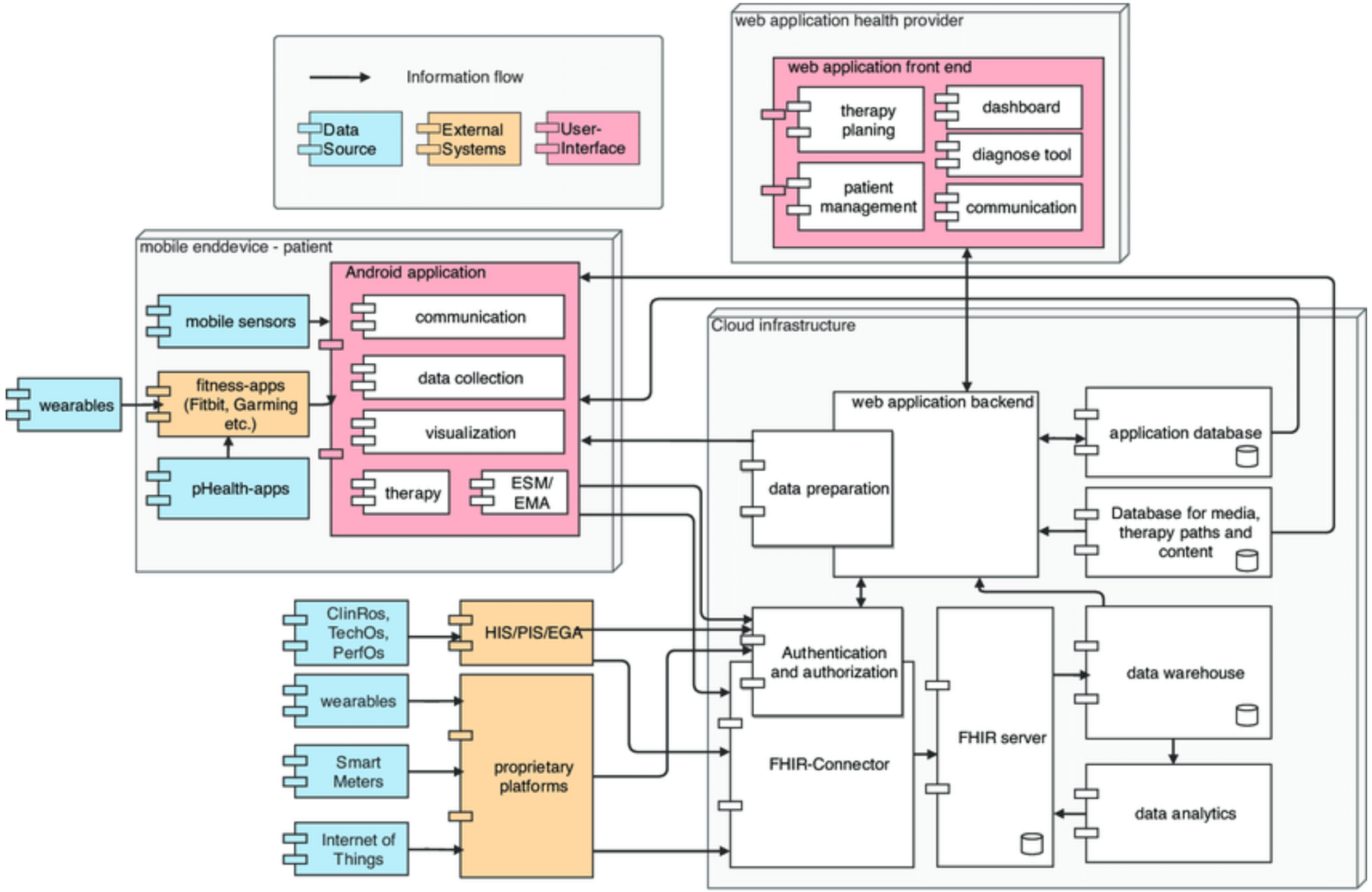
OpenID protocol



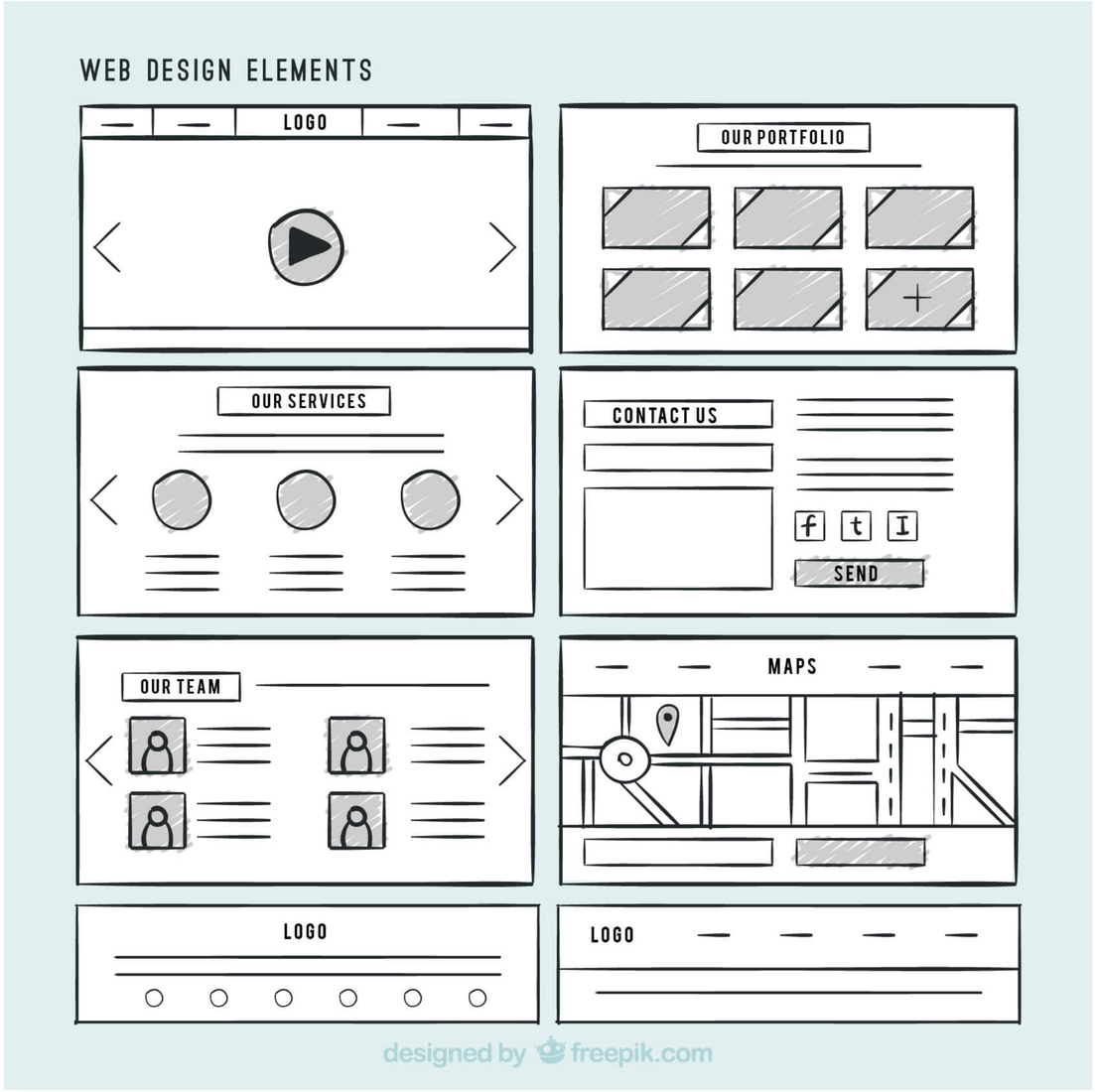
Entity-Relationship diagram



Component architecture



UI mockups


WEB DESIGN ELEMENTS
LOGO
OUR PORTFOLIO
OUR SERVICES
CONTACT US
SEND
OUR TEAM
MAPS
LOGO
LOGO
designed by freepik.com

Pseudo code

Algorithm 1 My algorithm

```
1: procedure MyProcedure
2:     stringlen ← length of string
3:     i ← patlen
4: top:
5:     if i > stringlen then return false
6:     j ← patlen
7: loop:
8:     if string(i) = path(j) then
9:         j ← j − 1.
10:        i ← i − 1.
11:        goto loop.
12:        close;
13:    i ← i + max(delta_1(string(i)), delta_2(j)).
14:    goto top.
```

Course modus

- Inverted classroom
- Some instruction - tools, artifacts of design, systems
- Building software and critical review
- Work in teams
- Communication will be via Canvas announcements
- Use the discussion board for class interactions
- Think “collaboration” instead of “competition”.