17-723: Designing Large-scale Software Systems

Interface Design Exercise

Carnegie Mellon University

This Lecture

- Response to your Feedback
- Overview of Milestone 3
- Interface Design for the Project
- Outlook towards Milestone 4

Carnegie Mellon University

Your Feedback!

Instructors are "responsive and patient to answer questions in class or slack." ☺

"The instructors is enthusiasm and encouragement and make me feel happy to join the class." ☺

Exit tickets provide effective practice for students to summarize the lecture, and an opportunity for students to resolve confusions. ☺

Examples in class provide students with concrete applications of design principles. ☺

Encouraging students to participate with **chocolate** rewards is effective. ☺

In-class discussions help students more deeply think about the course material. ☺

Carnegie Mellon University

Your Feedback!

Providing students with **code samples or templates** for the project will allow them to **focus on the design elements** and less on the front-end development.

➢ Thanks for the feedback! We will do this next year!

➢ Feel free to ask for support on Slack

➢ **Pair Programming** (two developers coding at one laptop, one typing the other one talking) can help you get up to speed faster with a new framework

Carnegie Mellon University

Your Feedback!

“Some of what we are taught is too abstract and high-level. Can give more concrete tactics”

The focus on principles is useful but you may want to provide a few concrete **guidelines or best practices** to students for them to take into professional contexts.

➢ We will try to include more concrete “**design recipes**” in future lectures

➢ Keep in mind that design cannot always be reduced to step-by-step instructions (otherwise ChatGPT / GitHub Copilot could do it). Experience with many examples will teach you more than following concrete steps that someone has laid out for you

➢ Most concrete guidelines typically apply to specific domains or scopes; principles are more widely applicable and long-lasting!

Your Feedback!

The **project** (develop a web application) may **not feel aligned** to the topic of the course (**systems design**) for some students who do not have domain knowledge required for the project

➢ We will make the **connection** between the project milestones & course topics more clear

➢ In the project you will experience **making design decisions** in large-scale systems and **experience the consequences of your decisions** (this is why we let you implement it rather than just draw it)

➢ More on this today

Carnegie Mellon University

Recall Interface Specifications

Syntactic View	Describe document format, the actions that can be performed, their parameters, and outputs.
Semantic View	Describe the purpose / meaning of the resource / action: • **Side-effects**: Changes to the state of a resource or environment • **Usage restrictions**: Who can perform this action? • **Error Handling**: What errors can occur and why? • **Examples**: Examples of outputs for a given input

Carnegie Mellon University

OpenAPI GET Specification Example

```yaml
paths:
 /users/{userId}:
   get:
     summary: Returns a user by ID.
     parameters:
       - name: userId
         in: path
         required: true
         description: Parameter description in CommonMark or HTML.
         schema:
           type : integer
           format: int64
           minimum: 1
     responses:
       '200':
         description: OK
```

Lets you generate server stub code for many languages and **generate HTML documentation**

See https://swagger.io/docs/specification/basic-structure/

Carnegie Mellon University

OpenAPI POST Specification Example

```
paths:
  /users:
   post:
     summary: Creates a user.
     requestBody:
       required: true
       content:
         application/json:
           schema:
             type: object
             properties:
               username:
                 type: string
     responses:
       '201':
         description: Created
```

Lets you generate server stub code for many languages and **generate HTML documentation**

See https://swagger.io/docs/specification/basic-structure/

Carnegie Mellon University

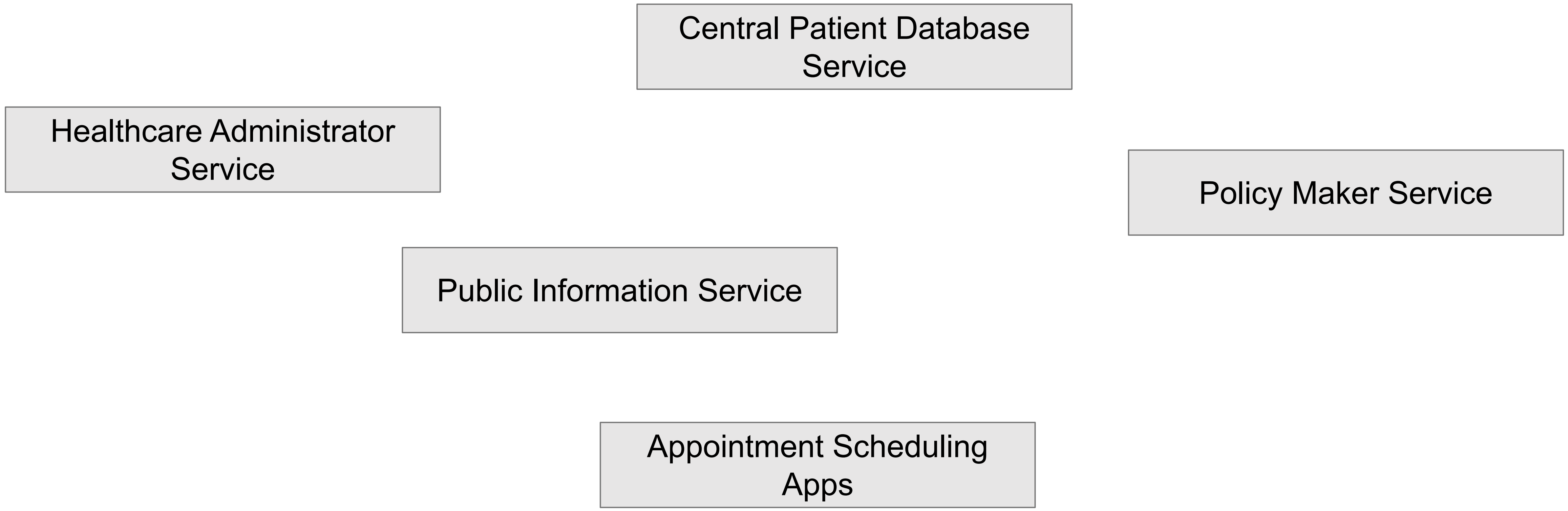
Example Interface Specification (GDS from Interoperability Lecture)

Add-on: **`(price, name, description, id)`**

- **`price(int):`** The price in cents (excl. tax) additionally charged when this add-on is selected
- **`name(str):`** The name of the add-on as shown to the user (in UTF-8)
- **`description(str)`**: A short description shown to the user in order to decide if they want to purchase the add-on (in UTF-8)
- **`id(str):`** Unique identifier of this add-on starting with the flight number (in ASCII)

Carnegie Mellon University

Responsibility Assignment


Central Patient Database Service
Healthcare Administrator Service
Policy Maker Service
Public Information Service
Appointment Scheduling Apps

Carnegie Mellon University

What Data Needs to be Exchanged?

Test Reporting

Quarantine Recommendations

API Design Exercise

Design a first draft of your team's APIs (syntax) in the Shared Document.

Teams for the Test Reporting & Quarantine Recommendations:

-> Create OpenAPI specification

Then meet with other teams to give each other feedback!

API Design Exercise: Semantics

Meet in your Project Groups to **update the syntax** of the your API and **add descriptions of the semantics**.

Semantic View

Describe the purpose / meaning of the resource / action:

- **Side-effects**: Changes to the state of a resource or environment
- **Usage restrictions**: Who can perform this action?
- **Error Handling**: What errors can occur and why?
- **Examples**: Examples of outputs for a given input

Carnegie Mellon University

Project Tasks

- Design your Service's API **by the end of Thursday Mar 14** and document it in the shared Google Docs Document
- Comment on the APIs of other Teams to ensure consistency