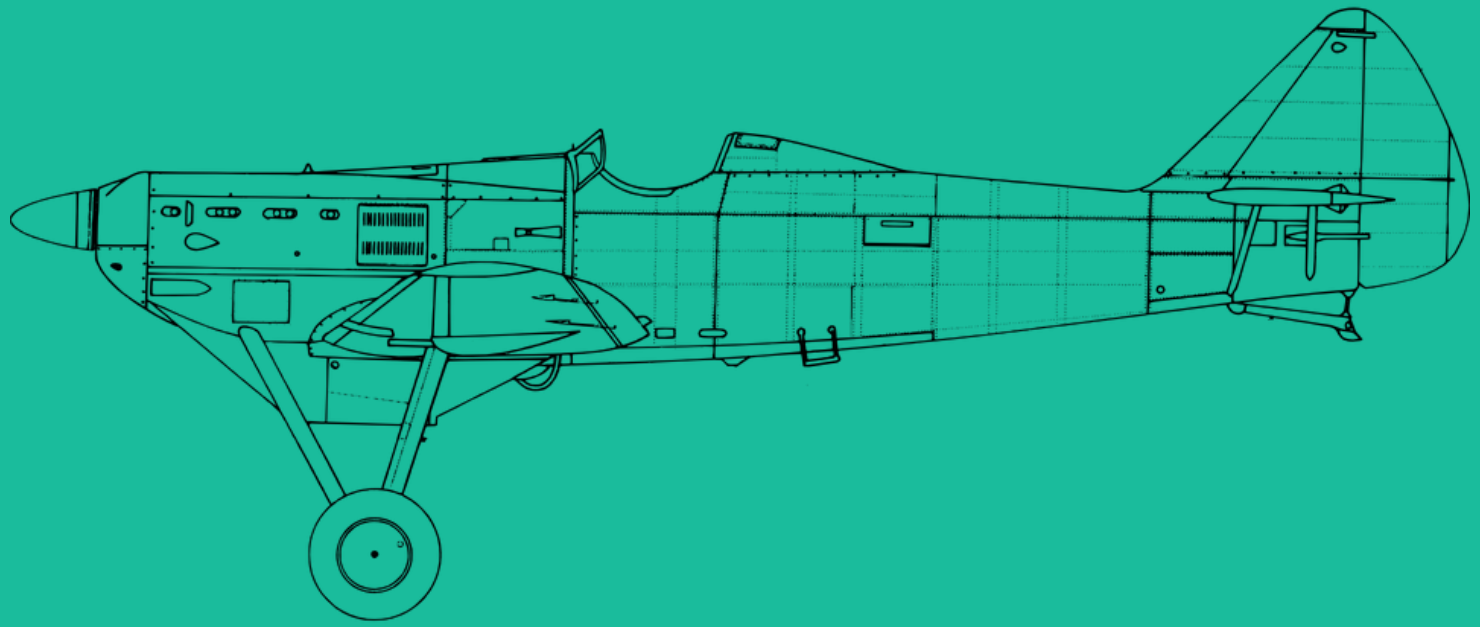

Project Definition

Bilkent University | CS443 | 2020, Spring | Dr. Orçun Dayıbaş

Definition

- **URL Shortener Application**
  - Design and implement a system to take user-provided URLs and transform them to a shortened URLs.
    - Ex: https://bitly.com/ , https://tinyurl.com/ , https://demo.polr.me/
  - Users will use the system to share their links with other parties (No need to be an authenticated user).
  - Shortened URLs will be used to access to original ones.
    - System redirects the incoming request to long (original) URLs.
  - Short links has to be short enough to be easily shared at mobile platforms (copy+paste, etc.).

Definition

- **Functional Requirements**
  - The system should generate unique and short links.
  - The system should redirect the request to original URL when a short link of that URL is accessed.
  - The system should have an option that the user can pick a custom short link for their URL.
  - The system should expire short links according to user-provided rules (specific exp. time, duration, etc.).
  - The system should provide “link analytics” for the user.
  - The system should provide “system-wise analytics” for the system administrator.

Definition

- **Non-functional Requirements**
  - The system should be highly available.
  - Redirection should happen with minimal latency.
  - Short links should not be easy to predict.
  - Link management UX should be designed for mobile.
  - The system should have REST endpoints as an API.
  - API should be designed to be used by other parties.

Constraints

- **Cloud-nativeness**
  - Design and implement your system according to “5 essentials” (see Chapter 1 for details) of cloud:

Constraints

- **Mobile Application & API**
  - The system will include an mobile application (B2C) and external facing API (B2B).
  - Analytics part is optional in the mobile application.
- **Monitoring & Testing**
  - Testing your system is a part of the project (important one).
  - Try to produce comparable results (e.g. user/core, user/disk).
  - At minimum, you have to validate your SLAs.
    - Remember “SLA → SLO → SLI”

Constraints

- **Tech stack**
  - You are free to use any service/framework/component for mobile & web application development.
  - Backend services (API) must be implemented in Java.

Do not use any external solution for the core business logic (e.g. do not use source code of any other link shortener project).

Deliverables & Demo

- **Design Report**
  - Includes system design (components, connections, etc.).
  - Includes all details of the system like which models will be used, what external services/APIs will be used, which metrics will be monitored, etc.
  - Includes SLAs for the system (you are required to convert functional/non-functional requirements into structural SLAs).
  - Includes tech. stack decisions (please elaborate these decisions) and trade-off analysis, assumptions (e.g. new URL reg.s are on 1M/day, redirects are on 1B/day).
    - If you assume anything, be sure you are doing this explicitly (read as “include this in the design report”)

Deliverables & Demo

- **Design Report**
  - Includes capacity estimations (followings but not limited to)
    - Bandwidth, Storage, Memory, CPU, etc.
  - Includes cost estimations (followings but not limited to)
    - Total cost of operation, cost of single user (you may assume it is an on-prem. solution to calculate these).

> You can use ballpark figures to calculate estimations. We will use these estimations just to compare with final results (but it is important to freeze estimations prior to implementation).

  - Includes testing strategies (performance test, etc.)
    - At least, you should test your system according to SLAs.

Deliverables & Demo

- **Final Report**
  - Evaluation of the process (decision updates, tasks of each team members, blocker events, solutions, etc.)
  - Use design report as a base and compare your final outcomes (Estimations vs. final findings, etc.)
  - API documentation
  - Details of codebase

- **Presentation & Demo**
  - Presentation (derived from “Final Report”)
  - Mobile UX, Feature tour, analytics dashboard, etc.
  - Test & Monitoring

Scheduling & Coordination

- **Hands-on implementation tutorials**
  - Docker, Kubernetes, etc… Starts with March 11th.
- **Deadlines, exam dates, team formations, etc.**
  - Expect an announcement from TA
- **Follow the course web page**
  - http://odayibas.github.io/CS443
- **Have fun...**