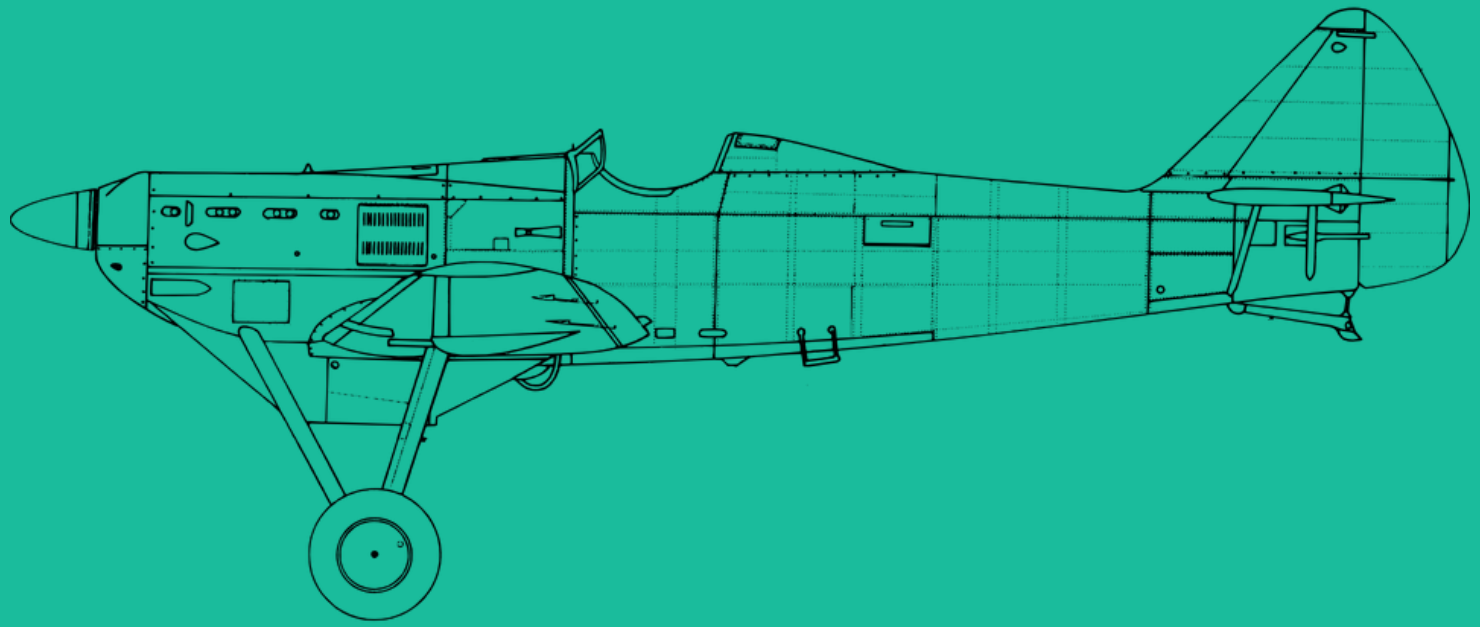


Project Definition

Bilkent University | CS443 | 2021, 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).
    - For instance, you are a unauthenticated/3rd-party user of goo.gl when you click on http://goo.gl/l6MS (it redirect you to long link → you are a 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 provide “system-wide monitoring” 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**
  - Try to apply 12-factor App principles.
  - Design and implement your system according to **“5 essentials of cloud”** (see Chapter 1 for details):

Constraints

- **Mobile Application & API**
  - The system will include an mobile application (B2C) and external facing API (B2B).
- **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 its architecture, what external services/APIs will be used, which metrics will be monitored (indicators to be measured), 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, etc.
  - Test & Monitoring

Scheduling & Coordination

- **Hands-on labs**
  - Docker, Kubernetes, etc.
- **Deadlines, exam dates, team formations, etc.**
  - Expect an announcement from TA
- **Follow the course web page**
  - http://odayibas.github.io/CS443
  - Use “watch” function of github and/or follow commits
- **Have fun...**

Q/A

Questions & Answers

- **Q&A-1**
  - We plan to only show the “system-wide monitoring” from a web application to the sys.adm., is it fine?
  - Yes, you can do this (as per that design, sys. admin will use web app only). UX design decision like that will be included in your design report.
    - You are expected to design your own solution that satisfies the requirements (functional, non-functional, cloud-nativeness, ease of use, etc.).

Questions & Answers

- **Q&A-2**
  - How will the B2B and B2C endpoints differ? Aren't they supposed to provide the same functionality, i.e, creating a short link and redirecting users to the long link?
  - B2B users (hypothetical) will consume the API from their own applications. Therefore, we will have two type of consumers: our mobile application (B2C clients) and 3rd parties (B2B clients). A good solution is expected to be able to differentiate these consumers (nice to have).
    - Examples: being able to define different SLA's for B2C and B2B or being able throttle one or another in such a lack of resource cases, etc.
    - Introducing these kind of trade-off points and mechanisms (first in you design and then in your implementation) is nice to have.

Questions & Answers

- **Q&A-3**
  - Do we need to include database tables, columns?
  - No, you don't have to… However, you may include them if you think they have an significant impact on reaching goals of the project. The idea is quite simple;
    - Include your objectives and design points in the design report and
    - compare/discuss with your final implementation in the final report (pivoting from the originally designed solution is perfectly fine as long as it is done explicitly and the final version is justifiable)

Questions & Answers

- **Q&A-4**
  - What measures should be taken in terms of application security?
  - It is part of your solution (read as “that question is implicitly asked to you”). It's your responsibility to include "enough" measures to secure your solution. For instance -I think- it is not that hard to tell; that kind of open system has to handle DoS attacks to provide high availability. Please do not be biased with my example, it is your design/solution.

Questions & Answers

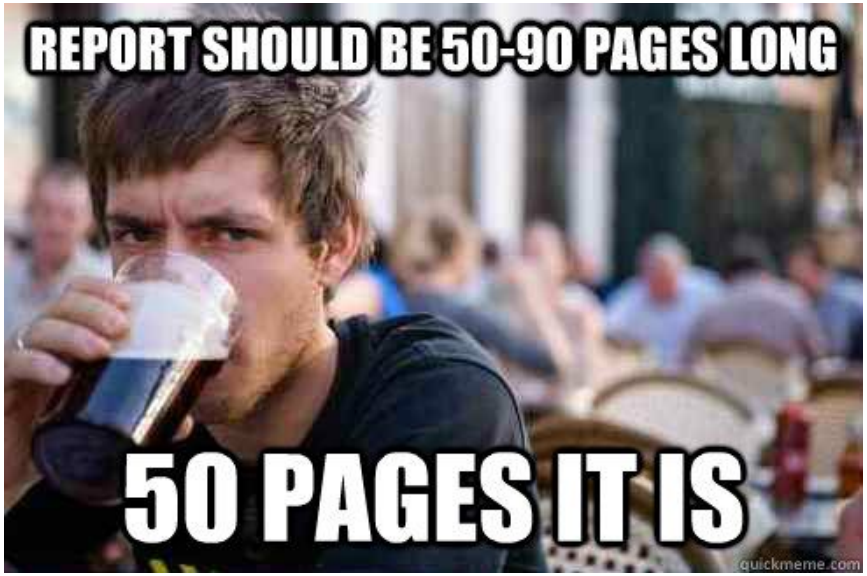
- **Q&A-5**
  - Are we expected to come up with new functionality that could affect our grades?
  - You need to be more into increasing/implementing/enhancing "5 essentials of cloud computing".
    - Frankly, all content and deadlines are aligned to give you enough resource (read as “man-hour”) to implement these non-functional requirements.
    - That's why the functional requirements are very light in this assignment.

Questions & Answers

- **Q&A-6**
  - As per the definition, testing is also part of the project workload. What platform/inputs&outputs will be used to test our system?
  - Yes, validation is important and you need to -at least but not limited to- cover your SLA promises (define SLAs and validation methods in the design report and evaluate your results in the final report).
    - Since load test will be your bare minimum, below platforms are very first tools to consider (not an exhaustive list, others are also fine)
      - [illegible]

Questions & Answers

- **Q&A-7**
  - Could you prepare a short report guideline to state how long should the different parts of the report be?
  - I can't see any value on that. Project description includes all details to decide the content of the report. Furthermore, the format is not part of evaluation (but it has to be readable). You can't get any extra point by generating redundant pages...



Questions & Answers

- **Q&A-8**
  - What do you mean by "testing strategies"?
  - It is not a business term. Literally you need a strategy to test your promises (like SLAs).
    - Ex: first you define an SLA for your solution's availability (basically that means you have "SLO & SLI". There will be no any penalty since this is a toy project but you may define things like a hypothetical refunds for the sake of completeness) and then you use that SLI for monitoring; that means you have a metric to measure/monitor/observe but still you need a strategy (e.g. creating different loads for specific time intervals and calculating moving window avr., percentiles for a representative/extrapolatable time frame like a week/day) to prove that you -as a solution provider- are keeping your promises.
    - At the end of the day, it is crucial to be able say that **"we are or will be confident that our solution provides X availability by using these testing methods on these indicators"** (elaborate it in your reports).

Questions & Answers

- **Q&A-9**
  - We couldn't find a standard for "capacity estimation", "cost estimation" is also not relevant since we don't pay anything for this project (free tier or on-prem). Do we need to use AWS/Azure plans to come up with these estimations?
  - We have discussed these extensively (during the physical class hours): What is max. capacity (*C*) -in terms of active users- that your solution can handle with your current resources (using free tier of any cloud provider, university server quotas, home-made server or dev. laptop, etc.)? What is expected cost for *C*x1000 or *C*x1M or ….? Show the scalability of your solution in terms of resources (hardware costs, subscription fees, etc. remember CAPEX/OPEX)
    - Be sure to define your resources (e.g. server specs) explicitly in your reports

Questions & Answers

- **Q&A-10**
  - We don’t understand the assumption part. Should we write what we assume about the usage limit of the system like how many user will use it or what we assume about the implementation part?
  - Explicitly write it down if you would take a design decision by assuming anything (Ex: We assume max. 1M/day new URL registrations, otherwise it is not possible to satisfy req.s with that implementation, which would, due to X component, fails. Thus, replacing X with Y would help for such cases like 10M/day or more).

Questions & Answers

- **Q&A-11**
  - In the system design diagram do we need to show external services in detail such as prometheus? If yes how so?”
  - Yes, you do. Be sure to include types of services (on-prem / hosted by yourself, cloud services, etc.) while enlisting them. C4 or UML can be used as a notation for design diagram.
    - https://c4model.com/
    - http://www.agilemodeling.com/artifacts/deploymentDiagram.htm
    - Give a proper legend if you choose to use non-standard notation or any other notation other than C4 and UML diagrams.

Questions & Answers

- **Q&A-12**
  - What kind of metrics do we need to show to the system administrator? Is showing metrics for business KPIs enough? Or do we need to show system-wide metrics? Asking this to confirm if just using an application database is enough to achieve this without having the need to use tools such as "Prometheus".
  - It’s part of your solution. Assume you are a sys.adm., would you like to monitor business KPIs or infra... or the both (if so, separate indicators or compound values like ratios)... Think about it, we have covered these details in the class (see "performance of cloud computing" in chapter 2).