CIT/P Portfolio Subproject 3 Requirements

Complex IT Systems

November 21, 2024

This document describes the requirements to the third portfolio subproject of Complex IT Systems. (Please refer to the note *CITP Introduction to Project Portfolio* for a general introduction.)

The goal of portfolio subproject 3 is to design and implement a frontend that provides a graphical user interface to the Movie application. This frontend must run in users' browsers and it must communicate with the backend that you have implemented as part of portfolio subproject 2.

You should structure the frontend so that it consists of decoupled parts organized into well-defined parts or layers with separate responsibilities. For example, the frontend may be internally organized into a Presentation Layer responsible for how data is presented to the user, a Business Logic Layer containing any logic independent of the presentation layer and data access, and a Data Access Layer implementing the communication with the backend web service.

Besides communicating with your own backend, the frontend must fetch pictures of persons from The Movie Database (TMDB), by making requests similar to those of Assignment 6. More details follow in Section D in this document.

Section H at the end of this document describes how, when, and what to hand in.

A User interface design

You must design and document the user interface and the information architecture of the frontend.

You should design the user interface such that it complies with Jacob Nielsen's 10 Usability Heuristics for User Interface Design. (See `https://www.nngroup.com/articles/ten-usability-heuristics/`.)

A.1. Design the intended site structure of the frontend. Document this design with a site structure diagram that shows the different "pages" or "views" that the frontend provides and how the user can transition and navigate between them. (See `https://www.bluehost.com/resources/website-structuring/`.)

A.2. Design the intended user interfaces of the individual "pages" and components of the frontend. Document them using wireframe models. (See `https://en.m.`

`wikipedia.org/wiki/Website_wireframe`.)

A.3. Design and document the routes that can be used to navigate the application.

B Presentation Layer

The actual graphical user interface must be implemented in JavaScript using the React library and Bootstrap. It should implement the design from Section A.

B.1. Document React components that you have developed and their roles; document the component hierarchy.

B.2. Describe React patterns, principles, and techniques that you have used (e.g., prop drilling, lifted state, useState, useEffect, useContext, and any other hooks applied, DOM router, and controlled components).

B.3. Describe the purpose of third-party JavaScript, React, or Bootstrap libraries or components that your application uses.

B.4. Your solution must implement the user interface with (React) Bootstrap. Describe how (and if) Bootstrap helps you implement the user-interface design from Section A.

C Business Logic Layer

You may chose to represent data in the frontend differently from the representation in the backend. Such an internal representation may, for example, reduce coupling to the backend and may ease combining data from different sources or servers. (See Section D.)

C.1. Describe the internal representation of data as objects and describe any classes or functions that you have implemented to help maintain these representations.

C.2. Describe other helper functions for manipulating data in the frontend.

D Data Access Layer

A data access layer may provide an interface to the backend: Instead of connecting to the web service offered by the backend directly from the presentation layer, you may implement a *data service* that defines an internal interface (for example, a set of functions or one or more classes) that the business logic layer can use to communicate (indirectly) with the backend.

This architecture prevents future changes in the backend from propagating into the business logic layer and the presentation layer of the frontend.

The data service should communicate asynchronously with the backend.

D.1. Show where and how data access is implemented and document the functionality it provides.

D.2. The frontend must fetch pictures of persons from The Movie Database (TMDB), following the approach outlined below. (These pictures should, of course, be used where appropriate by the Presentation Layer).

The attribute `nconst` in table `name_basics` uniquely identifies a person. We can use this IMDB-specific id to find the corresponding person in TMDB, by issuing a request to TMDB with parameter `external_source=imdb_id`, as follows (where *N* is this IMDB-specific id and *K* is your TMDB API key).

`https://api.themoviedb.org/3/find/`*N*`?external_source=imdb_id&api_key=`*K*

The result of this request is a JSON-encoded object *r* whose `person_results` field contains an array of *person objects* (in the format described in Assignment 6). Therefore, we can get the TMDB-specific id of the (first) person of this response by the JavaScript expression *r*`.person_results[0].id`. (For example, Steven Spielberg has `nconst` `nm0000229` in IMDB and id 488 in TMDB.) Given this TMDB-specific id, you can get profiles (and therefore `file_paths` and URLs) of pictures of a person, as described in Task 7 in Assignment 6.

E Functional requirements

The frontend must provide a user interface to the Movie application. The user interface must be designed such that it appears consistent and coherent.

E.1. Your solution should be a single-page application. (If you divert from this design, then explain where and why.)

E.2. The frontend must provide a navigation bar to transition between different types of content. It may also provide links to transition between content where you find it relevant.

E.3. Content with listed information must be provided with pagination when possible.

E.4. Features supported by your framework model should also be available from the frontend, such as (see section C in the note *CITP Portfolio Subproject 1 Requirements*) registration of users and bookmarking of titles and people.

E.5. The functional requirements listed under section D in the note *CITP Portfolio Subproject 1 Requirements* should be supported by the frontend. In particular, the frontend should support searching for movies and people, and rating titles. Optionally, your frontend may display word clouds.

F Non-functional requirements

F.1. You must implement the frontend using React, JavaScript, HTML, CSS, and (React) Bootstrap.

F.2. You should aim for the implementation being modular, scalable, maintainable, testable, fault tolerant, and extendable. In particular, you should split the solution into separate (JavaScript and JSX) files with well-defined interfaces, and export only relevant components, functions, variables, constants, and classes from these files.

G Individual reflections

In addition to the implementation and project work done in your group, the final report must also include so-called *individual reflections.*

G.1. Each group member must write two pages that include his or her own reflections discussing one or more concepts from the course (taken from any of the four sections) and relate these to the group's product design or product implementation. If you select concepts not addressed by your design and implementation, you should explain how they might be relevant.

H The final project report

You are supposed to work in groups and submit your solution to portfolio project 3 in a report (in PDF format) through `eksamen.ruc.dk`.

The final report must include a front page with (a) a title, (b) the group number, (c) the full names of all group members, and (d) a URL to your source repository. Make sure that your remote database at `cit.ruc.dk` is *updated and includes all changes* you've made during the project.

Your report should consist of around 5–10 pages excluding appendices and excluding individual reflections. (Spend the number of pages that you find reasonable. Don't stretch your writings solely to reach the maximum.) You are welcome to include summary and reflections on the three portfolio subprojects as a whole, but these are not required. The submission deadline for the report as well as the portfolio 3 product is December 18, 2024 at 10:00 am.