1 Introduction

Many restaurants offer their menu in the internet. These menus are often in PDF format, but also often as normal webpages. Aggregators, such as e. g. `pizza.de` or `lieferheld.de`, to name some of them, require the participating restaurants to provide their menu in a machine processable form. So the restaurants must provide their data, additional to their own website, also to other services.

The goal of this practical task is to develop a web application, that queries different publically viewable menus and presents them in a common user interface. The advantage for the user, amongst other things, to not having to search through all the menus for a special dish. He is presented with a menu aggregated of all the single menus of the respective restaurants and also will see which restaurants has the most inexpensive price. The exact task description follows in the next section.

2 Task Description

2.1 Vorbemerkungen

The following practical task may be worked on in groups of two members. If there is more than one participant all parts (programming, term paper) should be marked with who was responsible for them.

The submission consists at the end of a pdf file of the term paper and an archive (tar.gz, tar.bz2, tar.xz, zip) that contains the sources and a small manual on how to use them.

2.2 Programming

Develop a web application, that fulfills the following functionalities:

1. *Frontend:*
   (a) At least of three real restaurants (or other dishes providers), of which at least one has at least seven different dishes, the menu should be available. Aggregators (like e. g. in section 1 are prohibited.
   (b) All available dishes that are in the database can be listed.
   (c) It should be possible to search for dishes and incredients.
   (d) There should be detail information for dishes:
      i. Incredients per restaurant

ii. price per restaurant

(e) For all restaurants there should be data like address, telefone number, etc. available.

(f) For the decision if two dishes of different restaurants are equal, a simple case insensitive string comparison is sufficient.

2. *Backend:* The server side strucutre of the project should consist of two parts:

   (a) The *processing part* queries the data from the respective menus and saves the results into a database. It should be made so, that the websites of the restaurants are not overloaded and maximum once per day are queried for the complete menu!

   (b) The *REST-based interface* queries the database for the necessary data and provides them to the client. The interface may also initialize the reload of the data of the menus. For this project there is no need to create a proactive service, that always queries the menu data.

2.3 Term Paper

Further a term paper is to be written, that satisfies the following conditions:

1. Amount of about 10 pages (content!) for each project participant
2. An overview of the utilized technologies is give and why they were used. They will be also classified in the context of the lecture.
3. The project is presented.
4. In the *Appendix* (i. e. this is not part of the 10 pages) an API documention is inserted. It contains the following:

   (a) list of the functions of the REST interface

   (b) for every function the list of the parameters with type and meaning

   (c) for every function the list of the return values with type and meaning

   (d) Documentating by only listing of examples is *not* sufficient.

2.4 Additional Task

PDF-Menus: Not all menus are available in normal text form. Many restaurants provide just a menu as PDF. There are tools to inspect PDF files. Implement the extraction of the content of the PDF menus for at least one restaurant!

Categories: Often menus contain a categorization of the dishes, e. g. startes, main courses and dessert. Enable your project to take over these categories.

3 Examination

The examination consists of a 10 minute presentation, which follows the same criteria as the content of the term paper. The presentation does not contain the API documentation though. It is to show the project and the utilized technologies as well as its place in the context of the lecture.

After that some questions—concerning the project, but also out of the lecture—are given. Finally there will be a short consultation and you will be informed about your mark.

4 Dates

Handout of task description:	starting 28.05.2018 0:00 MESZ (UTC+2)
Submission of project:	until 02.07.2018 0:00 MESZ (UTC+2) via e-mail: • alexander.adam@informatik.tu-chemnitz.de or • daniel.richter@informatik.tu-chemnitz.de
Oral exam & talk:	in the examination period, lists for registration will be laid out with Ms. Hofmann in room 1/336e