Final report

Micro service 3 (MS3)

Team 301

Mersi Stafa
01276603

Overview:

- ***Discussion of deviation from your original plan (laid in SUPD)***

Everything was implemented as planned and also the missing parts were added in the application. This final implementation offers the completed and whole assignment given in the beginning of the semester.
The only problem that we had during the implementation was while trying to connect our micro services. When the client of my micro service tried to connect with the available version of the first micro service, a cross-origin problem was encountered. The request that was sent by the client was seen as forbidden and that caused the issue. At such a case, the only solution was using a Cross-Origin Resource Sharing filter so that the client of my micro service could actually make requests to the first micro service. This was easily done by changing the "cors.allowOrigin" field to a "*" in the web.xml file for web content on the first micro service. This could be seen as a deviation from the plan we had on the first micro service.
The plan that was offered in supd is already completed and there is no deviation from my side on that.

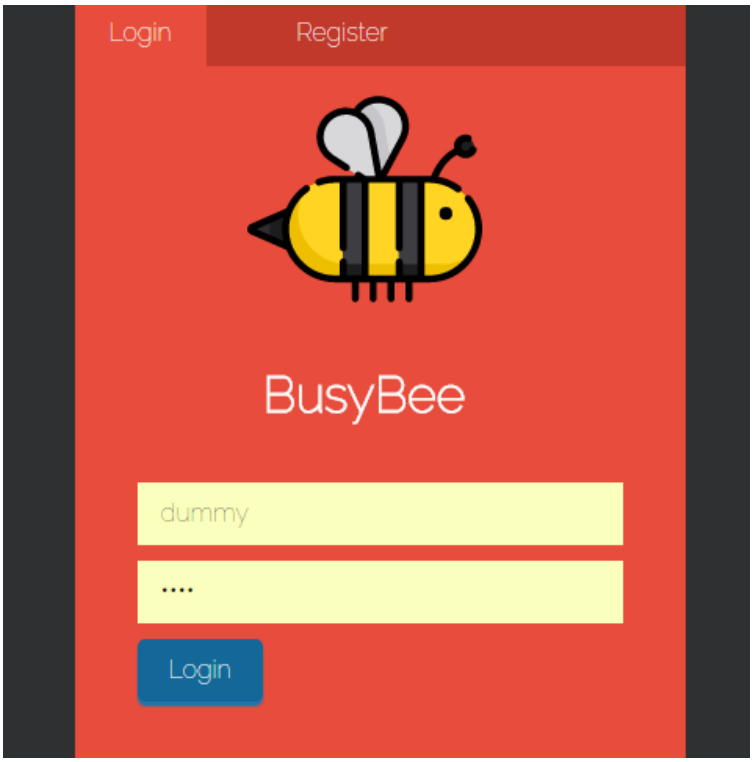
- ***Description of the final service's interface***

The technologies, as before planned and that are being used in this micro service are:

ReactJS is a an open-source library from JavaScript used to build user interfaces. It processes only the user interface. ReactJs allows you to a create an application that uses data and can change frequently without the need of a page reload.[3]
Redux is a predictable state container used for JavaScript applications. It provides live code editing and also helps you write consistent applications that run in different environments. It is highly suggested to be used with React.[4]
Webpack is an open-source JavaScript module bundler. Modules with dependencies are taken and static assets representing those modules are generated. Webpack is used in order to control the packages of the NodeJs server offered in the second micro service. Webpack allows hot redeployment and works best with NPM.[1]
Npm is a package manager used for JavaScript. It makes it easier to reuse the written code and also adapt it to other or new applications as well as incorporation of code.[2]

Description of user interface:

Login Component:
Login component is the first component of the application. The Login component contains a login form which has two fields: the login one and the password one. There exists also validation on submission. If the username or password are not matching , an error popup will be shown on the screen.

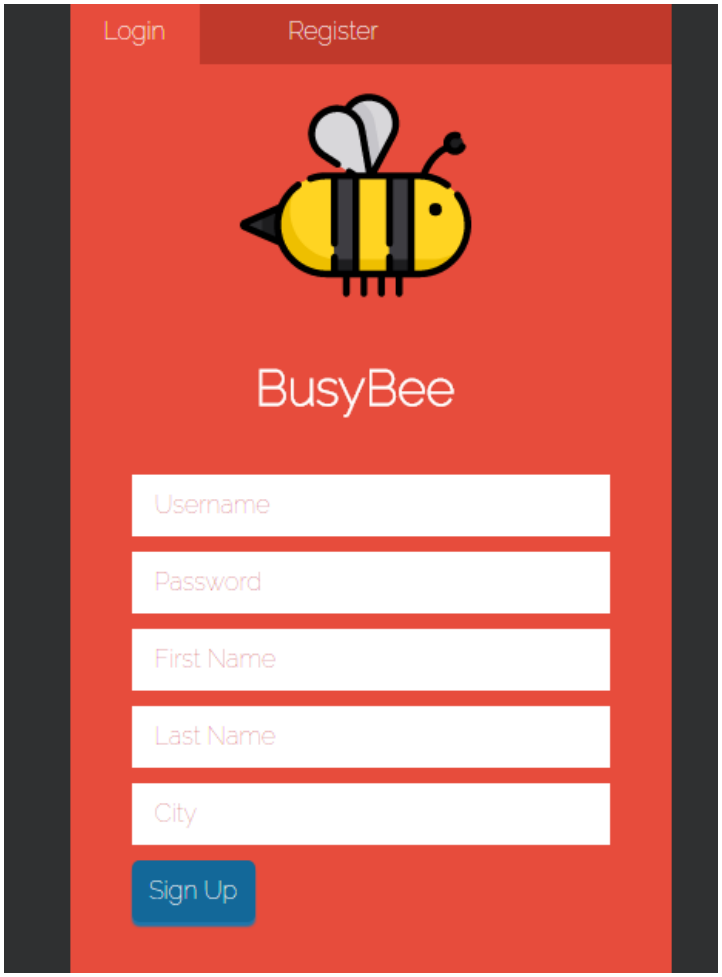


After a successful login, you will be redirected to the TODO list page.

Registration Component:
If you click on Register Tab you will get an extended version of the login form. This means that the user is new and he needs to fill out this form in order to continue in the next step. This form contains:

- Username
- Password
- First Name
- Last Name
- City

If the user already exists, an error popup message will appear on the screen.

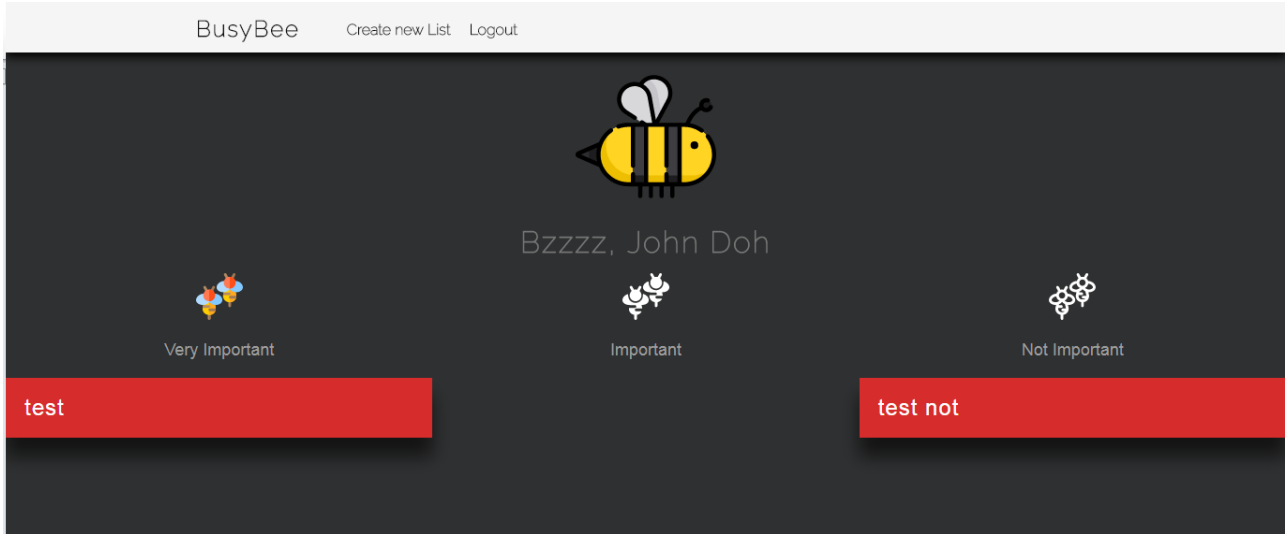


After a successful registration, you will be redirected to the TODO list page.

TODO Lists Overview Component:
The TODO List component is a dashboard with existing TODO lists in it. There are three already given categories of the TODO list which emphasize the importance of this list:

- Very important
- Important
- Not Important

Each category has a unique icon which differentiates them as below shown.



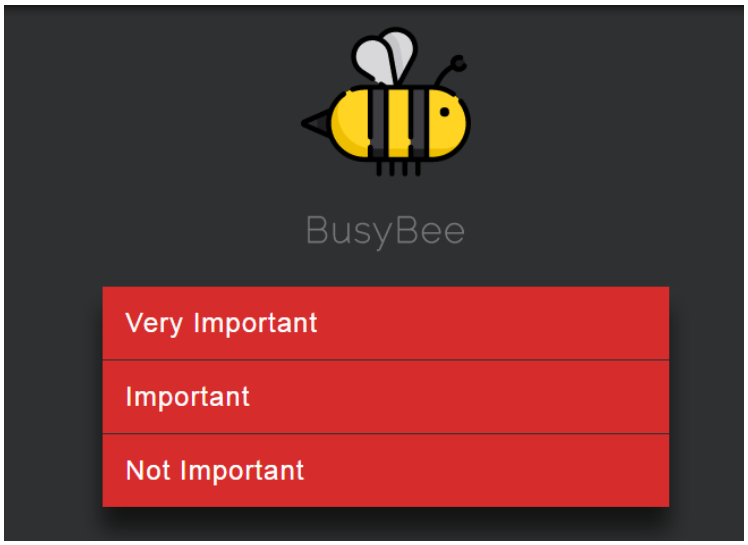
Single TODO List component
A Single TODO list component contains multiple subcomponents:

- TODO List name, which can be also edited. Near the todo list name, there are two icons which allow editing a todo lists name or removing the whole TODO list. When removing a list, you will be directed to the todo list overview.
- Tasks Creation Subcomponent.
  This component allows to create a new items that are part of the list the user is at the moment using.
- Tasks list subcomponent.
  It shows the added items or tasks of the list the user is momently in. You can edit the task name, mark the task as done or undone and also delete the task by using the given buttons on the side.

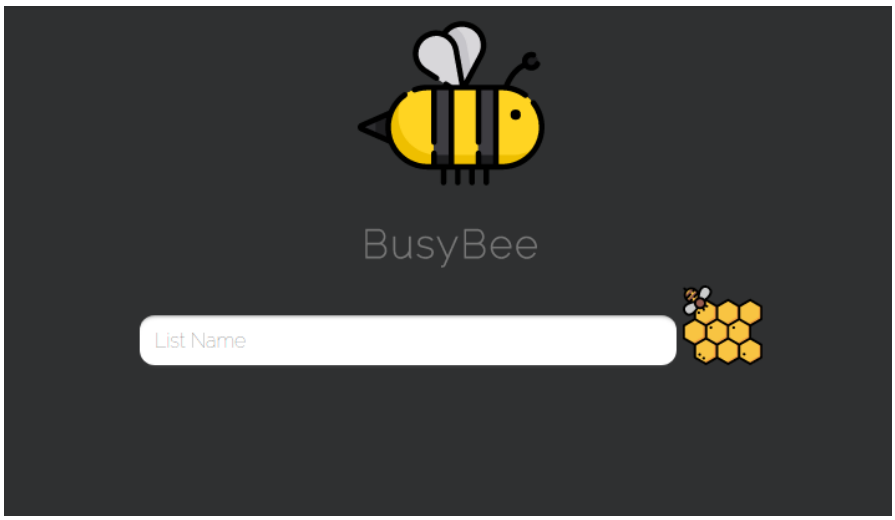
Create TODO List Component:
There are few steps which have to be done to create a new TODO list:

- Choose a category (also described above)
- Give a name for the TODO LIST
- Add tasks or items in the list if it is needed

Choosing a category:
In this step, you need to decide to which category your list is going to belong by highlighting its importance.



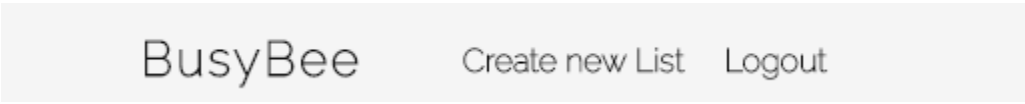
Finding a name:
The second step is to give a name for a list. The name can be written in the box and it can be added by clicking the button on the side.



After these both steps, the last step is to add items to the list by simply writing their names.

Navigation Component:
The Navigation bar contains three components which are:

- The name of the application
- Create new List
- Logout button



- ***Quick start tutorial***

The tutorial is intended for this micro service only, to be used alone (not as a whole with the others).

There are a few pre requirements that need to be checked before the actual installation:

1. Npm version 5.5.1 or higher.
2. BigIPEdge connected on the laptop or to be under the University of Vienna network.
3. OpenVPN GUI in order to make the application server accessible via VPN to others.

After these requirements are done, the installation may continue by completing the following steps:

1. Clone MS3 git project or simply download it.
2. Go to ms3 folder
3. Type “npm install” in the terminal and wait until all packages and other dependencies needed are installed and stored to the node_modules folder.
4. Type “npm run dev” in order to quickly start development server.

Please note that the application will be available on port 3000 on you localhost.

Usage:
Go to: http://localhost:3000
Using “dummy” data without the need of the micro services MS1 and MS2.

Login:
Username: dummy
Password: empty

After that, you will be redirected to the dummy TODO list view.
There are two TODO LISTS predefined:

- Test
- Test not

You can click on the TODO LIST and you will find out some dummy tasks. You can manipulate this list and also the tasks in it, namely:

- Create a new task
- Mark the task done or undone
- Remove task
- Rename task
- Rename TODO List
- Remove List (the list will not be really removed because it is dummy data we are actually using)

In order to make application distributed so that everyone can use it, you need to use the University of Vienna Network or a VPN offered by BigIPEdge as well as OpenVPN. After that, the UI is going to be reachable on http://10.103.101.13:3000/ if the machine is also reachable.

- ***Sources:***
    1. https://en.wikipedia.org/wiki/Webpack
    2. https://docs.npmjs.com/getting-started/what-is-npm
    3. https://en.wikipedia.org/wiki/React_(JavaScript_library)
    4. https://redux.js.org/