Duale Hochschule Baden-Württemberg

Mannheim

PitchApp Documentation

by Team 1.

Faculty of Business Informatics - Sales and Consulting

20.11.2018-20.07.2019

Authors: Boglárka Lehoczki, Csaba Kegyes, Ethan Kelly and Stella Kamakari

Lecture Directors:

Web-Development Benedikt Sondermann

Project Management Dagmar Schulte

Contents

1 Introduction of PitchApp

by Boglarka Lehoczki

1.1 The Value PitchApp Provides for Businesses

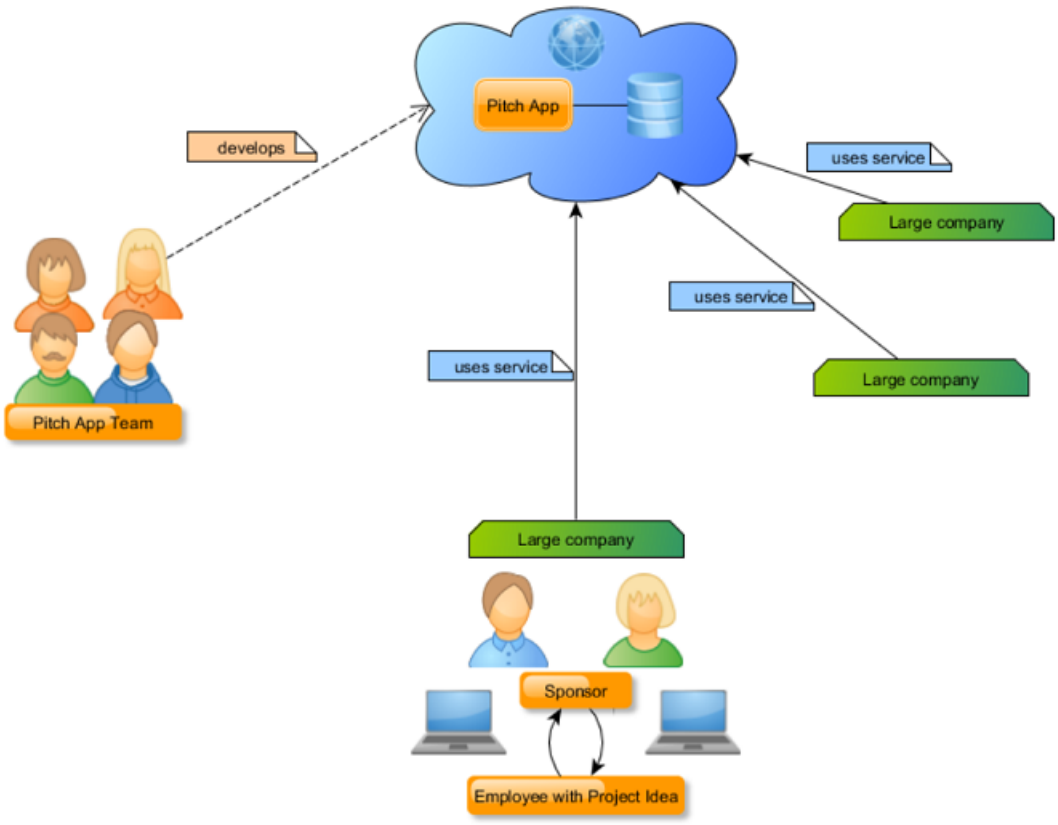
PitchApp is intended to be a platform to help employees find the required organizational support to turn their ideas from inception to reality. The goal of our web application is, thus, to facilitate the launching of new projects. By making project ideas of colleagues easier and faster visible to management, PitchApp encourages employees to contribute more actively to the success of the company at which they work. In this way, PitchApp helps to achieve higher degrees of intrapreneurship, which leads to business growth. Using our application will bring companies ahead of the game, in terms of innovation and employee engagement, as well as make big firms more competitive and flexible, thus more profitable. Hence, "fast innovators take leadership positions in their industries" (Stalk and Hout 1990).

1.2 Characteristics of PitchApp

PitchApp is a dynamic, single-page web application with database connection that we developed to enhance employee engagement and proactivity by connecting the employees' ideas to even the highest levels of executives. Managers with budgets and resources for projects (i.e. potential future sponsors) can browse between different project ideas, which are posted by the employees. Distinct types of ideas are sorted into groups like HR, Procurement, R&D etc., which facilitates searching among them. Then managers can offer their resources for the realization of a project idea, which they find valuable. Employees are also able to view the pitches posted by other colleagues in between their organization to avoid the sharing of redundant ideas. PitchApp is planned to be able to serve more large organizations at the same time and to be provided as a Software as a Service. PitchApp includes a user and session management system, which allows secure login and logout functionalities. It differentiates between public area, i.e. our landing page, and member area with two type of users, idea owners and idea sponsors.

Requirement	Status	Technology
Log in / Log out (differentiation between public section and member area)	done	Okta
User management	done	Okta
Session management	done	Okta
Application linked to a database	done	PostgreSQL
Dynamic content	done	React single-page web app
Not high complexity, but challenging/latest technologies	done	See above

The table above shows how PitchApp fulfills the given project requirements.



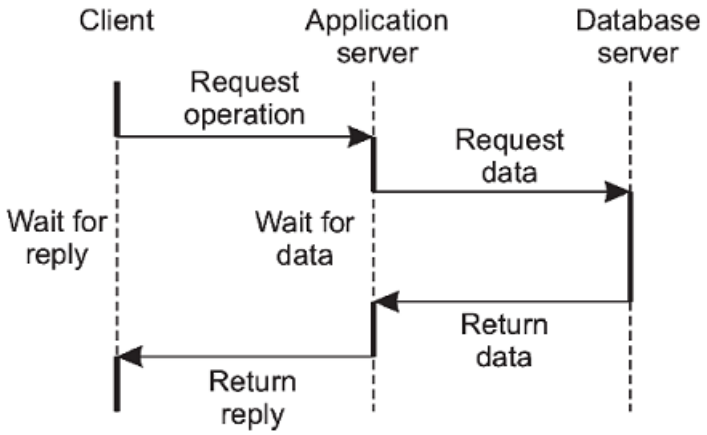
The figure above presents the general characteristics of PitchApp.

2 Architecture

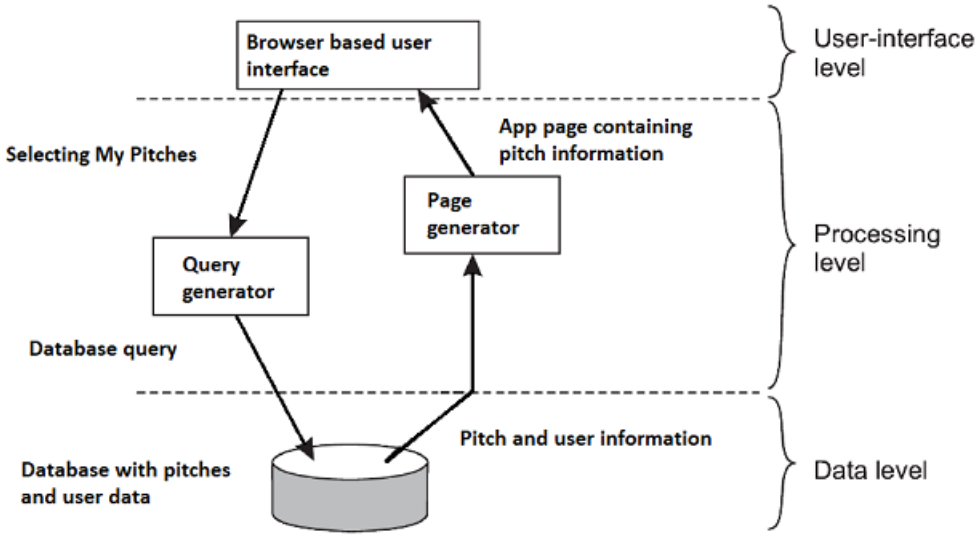
by Boglarka Lehoczki

2.1 Client-Server Architecture

PitchApp implements a classic client-server architecture. More specifically, PitchApp is a web application and has a 3-tier architecture. The three tiers are the user interface, the application server and the database server. In such an architecture, the user interface runs in a web browser like Google Chrome or Mozilla Firefox. The user interface communicates with the application server through HTTP requests and responses, as the application server also implements web server functionalities. The application server itself acts as a client of the database server (Tanenbaum and Steen 2017, p. 80). The interaction between these two servers can be based on different protocols or database connectivities, like JDBC for JAVA or ODBC for ABAP. In the case of PitchApp, this communication is solved by a Hasura GraphQL Engine, which auto-generates queries as part of the GraphQL schema from our Postgres schema model (Hasura 2019). The application server fetches the needed data from the database server, which returns it to the client in its reply. The process described above is shown by the following figure (Tanenbaum and Steen 2017, p. 80).



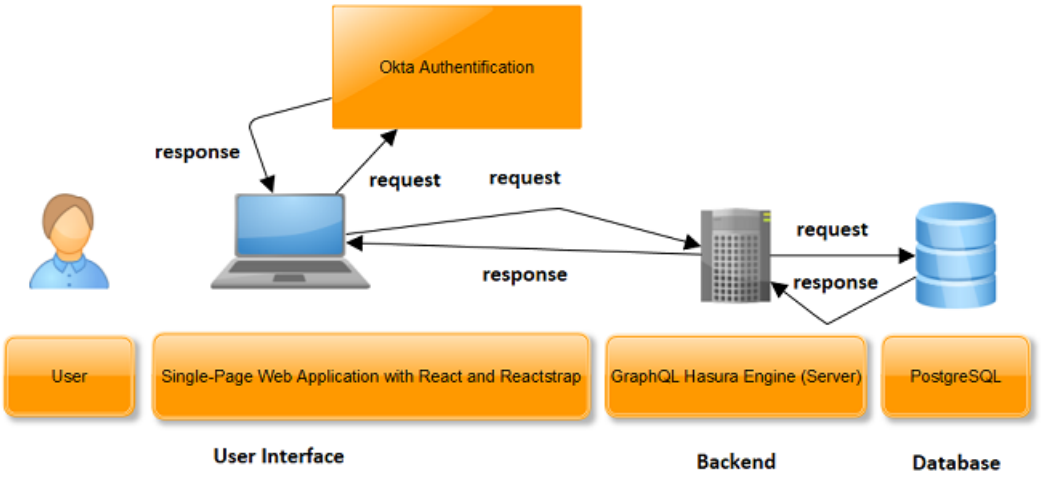
The following figure gives a generalized example on how the tiers interact with each other when a user wants to see only those pitches, which were created by him or herself. This figure is based on another one from the book Distributed Systems (Tanenbaum and Steen 2017, p. 61).



Developing a web application was a given requirement and it has several advantages. In comparison to a native application (with 2-tier architecture), the user do not have to install any additional application to its local machine, because the web application runs in a browser. From this also follows, that if in the future we e.g. change the user interface the user do not have to download updates onto his or her local machine. An other benefit of web applications is that they are easier to scale and the different tiers can be scaled separately based on the use case. From the viewpoint of PitchApp this is particularly important, as our application has to be able to serve a large number of users from our customer companies.

2.2 Technologies used for each Tier of PitchApp's Architecture

We selected state-of-the-art technologies to implement PitchApp. To develop a dynamic single-page web application, React was used. With Reactstrap, we were able to create a responsive and neat-looking user interface. Including an Okta modul to our web-application helped us to provide our users a secure authentication, user- and session management system. Our back-end is a Hasura GraphQL Engine which communicates easily with a PostgreSQL database. The following figure shows the architecture of PitchApp.



3 Technologies used for Implementation

3.1 React and Reactstrap

by Boglarka Lehoczki

React is a JavaScript library, which is used to create web-based graphical user interfaces (React 2019). As the basis of our web-application, a React App was created (Reactstrap 2019). The reasons, why we decided to implement PitchApp using React, are based on the general characteristics of React.

First, React is declarative. With this library, it is easy to develop interactive web-applications. PitchApp was developed to be an "idea pool", where ideas of users must be very straightforward to collect and display. We also needed an simple way to represent states of Pitches, i.e. if a Match occurred to a Pitch. A Match means that an idea sponsor found that a posted idea (Pitch) is worth to be realized.

Second, React is component-based. This enabled us to use already implemented components of Reactstrap (Components 2019), combine them and - as a result - get a complex, yet neat looking UI design. Each component manages states internally, which facilitated the handling of different states of UI elements. A render() method is implemented in each of the React components. This method takes input data and returns what to display. Render() can access input data by this.props. Internal state of a component can be accessed similarly with this.state.

Finally, React is compatible with Node, which we used on the server side for our back-end implementation.

Our goal was to build a convenient UI, where employees share their ideas happily and managers can smoothly brows between these. For this reason, we decided to use the neat looking design of Reactstrap. We used -inter alia- the Reactstrap components Badge to indicate a Match, Buttons to enable user interaction by clicking to navigate inside PitchApp, Jumbotron for the landing page design, Media to include our logo and Modals to display additional information to the users.

3.2 Okta Authentication
by Ethan Kelly

3.3 GraphQL Hasura Engine as Server
by Csaba Kegyes

3.4 PostgreSQL Database
by Csaba Kegyes

3.5 Docker

4 Final Results

5 User Manual

6 Difficulties of Implementation

7 Future Outlook: Missing Components and Functionalities

Bibliography

Components, Reactstrap (2019). *Reactstrap Components List*. https://reactstrap.github.io/components/alerts/ (Accessed 29/06/19).

Hasura (2019). *Hasura GraphQL Engine Documentation*. https://docs.hasura.io/1.0/graphql/manual/index.html (Accessed 29/06/19).

React (2019). *React Documentation*. https://reactjs.org/ (Accessed 29/06/19).

Reactstrap (2019). *Reactstrap Documentation*. https://reactstrap.github.io/ (Accessed 29/06/19).

Stalk, George Jr. and Hout, Thomas M. (1990). "Competing Against Time". In: *Research-Technology Management*. Volume 33, pp. 19–24.

Tanenbaum, Andrew S. and Steen, M. van (2017). *Distributed Systems*. New York, USA: Pearson Education, Inc.