Architecture of web applications based on Angular Framework: A Case Study

Jelica Cincovic, Sanja Delcev and Drazen Draskovic
University of Belgrade, School of Electrical Engineering, Belgrade, Serbia
{jelica.cincovic, sanja.delcev, drazen.draskovic}@etf.bg.ac.rs

Abstract - Currently, Java Server Faces, Spring and Hibernate are taught on the course of Programming Internet Applications at the University of Belgrade - School of Electrical Engineering. Further, the Faculty Information System (FIS) is also realized on these web technologies. The primary focus of this research is to present the architecture of modern web applications based on the Angular Framework in the integration with other web technologies and their comparison with the existing FIS implementation. We have made comparative analysis of the Angular implementation in the frontend of the application with a backend developed in: 1) NodeJS framework; 2) .NET technology; 3) programming language Python. Based on selected analysis categories, the obtained results of this study will help software engineers to make better choice of appropriate technology and architecture, depending on the requirements of the application that they are developing.

I. INTRODUCTION

Our motivation for this research based on the analysis of new JavaScript technologies, is to improve our university course Programming Web applications and to find alternative software architectures for the Faculty Information System (FIS) implementation [1], at the University of Belgrade, School of Electrical Engineering.

JavaScript has gained increasing popularity over the span of a few years, which is a remarkable feat in itself. The JavaScript language supports the Model View Controller software architecture that maintains a readable code and clearly separates parts of the program code. Angular as one of the most popular JavaScript’s frameworks was built for performing speedy and agile work in software teams. On the other hand, software engineers who daily work with web technologies are increasingly become full stack developers, and they are able to work on both the front-end and back-end portions of an application with equal quality.

The authors performed a comprehensive analysis of existing frameworks based on JavaScript [2], and they concluded that Angular is currently one of the most popular and most used frameworks and that a reformed university course, as well as a modified information system, can be based on this technology. Further research focuses on the choice of technology on the server side, which is the main subject of this research.

The research should answer the following questions:

- What can be used on the backend of modern web applications, if we use Angular in the frontend?
- What are the advantages and disadvantages of the analysis done with three different technologies combined with the Angular framework?
- Conclusion of the authors on their case study: the application of these technologies within their university course and the proposal of new web technologies for updating the Faculty Information System currently based on Java Server Faces technology.

II. METHODOLOGY

Angular (or Angular 2+) is a framework for creating advanced web applications on the client side. It is based on the TypeScript language, as the main tool for writing code, because of its types checking and better structuring. It was made by Google. Angular JS is the precursor of the current version, but at the same time it is so different that it is considered a completely different framework.

Angular structures the code based on mutually weak dependent components, each of which has its own HTML, CSS and controller written in TypeScript. In addition to the Angular components, it also enables the creation of services and directives as well as various other elements. At the moment of writing this paper, Angular is one of the most popular modern web application programming frameworks on the client side. The survey used version 7.2.7.

The methodology of this research included the following phases:

- Comparative analysis of the following frameworks: AngularJS / Angular 2+, Ember, Knockout, Backbone [2];
- Modeling three different types of web applications based on a similar software architecture: Angular + server-side technologies + database (MongoDB / MySQL / SQL Server);
- Application of one model within the university course Programming Web application instead of the current technology;
- The pros and cons of the analysis based on open literature, papers and realized web application models.
III. TYPES OF ANGULAR WEB APPLICATIONS

In this research, the Angular student portal with all existing use cases and functionalities was realized in three different variants of the modern full stack architecture: (1) Express and NodeJS (App A), (2) .NET (App B), (3) Python (App C).

A. Angular and NodeJS

Angular and NodeJS are most commonly grouped into the MEAN stack: MongoDB, Express, Angular, and NodeJS. MongoDB is a free database that belongs to a group of non-relational NoSQL databases. It is based on documents, which means that it keeps all its data in the form of a document, which in structure looks like JSON files. This implies that the database itself has data structures such as arrays or objects, and the structuring itself can be much simpler than with relational databases. Documents are not directly dependent, which means that even if they are of the same type, they can again have a different structure. Redundancy with this database may be desirable because of the structure it owns, and because of the increase in performance. As well as relational databases, this database also has indexing and search by queries. The study used version 4.0.5.

Express is a minimalist and flexible framework for NodeJS. It greatly facilitates the work with web services, their creation and routing. Express allows us to upload a web application in a very quick and easy way, receive requests from users and send answers in the form of HTML or JSON type. Serving static content is also very easy within Express.

NodeJS is a server-based JavaScript work environment that does not execute JavaScript code within a web browser but on a server. It represents the "JavaScript everywhere" paradigm. It is based on an event-driven architecture, capable of asynchronous communication and asynchronous code execution. Although JavaScript itself is always executed in one thread, NodeJS follows, but also improves by invoking asynchronous calls and using its core libraries and functionalities that can be executed in parallel. NodeJS never blocks the execution of its code but uses the event loop with which it can send or receive events asynchronously. NodeJS is run on a very fast and efficient engine called V8, which is written in C++. For this reason, NodeJS is very popular today in modern web application [3].

![Fig. 1. Architecture of the MEAN stack](image)

In MEAN-based applications, the overall implementation of the server side of the system should be separated into one folder, and client-side in another. The main file for the NodeJS application is server.js which should contain the object of the Express application itself, its configuration, list all the routes to which it corresponds, and finally the launch of the application itself.

The service layer of the application should also be separated into a separate folder and each of the services that it implemented should be in a separate file. Files should be shared so that it knows which file is responsible for which data or actions within the system. Finally, the service layer communicates with the data access layer which should be located in a separate DAO (Data Access Objects) folder. These files and their objects are the only ones that communicate with the database. In this way, we get linear communication in both directions between the layers, which leads to a safer and cleaner program code.

When the server receives a request, he first attempts to resolve which route is called. If he succeeds, he invites a specific service set up to respond to it, forwarding to him a user-requested object, as well as an object that represents an answer to be sent to the user. The service retrieves data from the database or writes them to the database and returns the user's response in the form of status or some data. If the server fails to resolve the route, it must statically service the Angular frontend of the application. All routes should start with the /api path to make the routes on the server side different from the routes on the client side, for example:

```javascript
const users = require('./api/users');
app.get('/api/users', users.fetch);
fetch = {request, response} -> {
User.find()
  .then(users => response.send(users))
}
```

Mongoose library is used to work with MongoDB, which provides a clear and easy interface. First, it is necessary to create a schema or template by which documents in the database will be created. In each scheme we have to indicate the name of the document field, then the type for each field and if we want one of the validators. Validators may require a field to be compulsory, unique, or to follow some regex template. Apart from owning the values themselves, the field can refer to another document through a unique primary key that the database generates for each object itself. We can add certain actions to constructed scheme, that could be executed over the document. Such methods are arbitrary, but we can also define some existing ones, such as pre () or toJSON (), and thus change their behavior.

From the obtained scheme, we make the model and this model is made to the public domain of the system for using the selected type of document. Above it you can use all CRUD operations through the method: deleteOne (), findOneAndUpdate (), find (), save () as well as many others. Each of these models returns Promise, which means it functions asynchronously and non-blocking so we can continue to execute the code, while the server running in the background communicates with the database.

B. Angular and .NET Core

Angular can also be paired with ASP.NET Core (Active Server Pages .NET Core) on the server side. ASP.NET Core is a free open source framework developed by Microsoft. ASP.NET Core is a redesign of the ASP.NET 4.x workspace with architectural changes.
that have led to ASP.NET Core being a modular workflow that is distributed through NuGet packages. Applications that use this worksheet can be developed on both the Windows operating system, Linux and Mac. ASP.NET Core provides easy integration with modern client operating frames [4]. This worksheet also provides the Model View Controller framework. With the .NET Core, EF (Entity Framework) is often used. It is the open-source workflow for object-relational mapping. It allows engineers to work with data by using the appropriate classes without worrying about the tables in which the data are stored. Also, it eliminates the need to write a larger part of the code, otherwise necessary for access to the database and the data about it itself. Figure 2 gives an application diagram whose business logic is implemented in .NET Core and the data layer in the EF.

![Diagram](image)

**Fig. 2. Architecture based on .NET Core, Entity Framework and MSSQL Server RDBS**

This architecture works best with the MSSQL Server Relational Database. MSSQL Server is the primary query language (Transact SQL), which means that besides the basic classic SQL queries, it allows for more complex things such as changing a program flow, etc.

The role of the server module is to communicate with the database and external services, obtain data from them, process data and forward them to the client module. For external services, REST (Representational State Transfer) communications are used. These services use resources of various formats, such as XML (Extensible Markup Language), JSON (JavaScript Object Notation), and many others. The communication between the client and the server is by default without state. Each communication state must be sent explicitly for each interaction. Clients who use REST web services communicate with the server using the HTTP (HyperText Transfer Protocol) protocol and use URI to identify the resource during this communication.

The basic elements of the server module in this architecture are a set of classes that represents controllers and a set of classes that represents application models. In controller classes, communication with the model (representing the database) and with REST services takes place. One controller class should be for each logical part of the application.

### C. Angular and Python

Python is the most popular language for image and text processing, communication with other Internet of Things systems, applications based on a large number of budget and mathematics. In full stack, Angular appears in the combination of the Python and Flask programming language, with the most common connection with a relational database, such as MySQL and PostgreSQL.

Python belongs to high-level programming languages. The standard module library supports a large number of standard formats and protocols. There are a number of standard modules that enable efficient work in many areas: modules for creating a graphical user interface, connection with a relational database, working on lexical analysis using regular expressions, and many others. This language is actively used in science, in the areas of software implementation and testing.

In web applications, Python is commonly used to implement complex systems with Internet services, such as cloud systems, web browsers, and other tools. Python can store data in memory, files, or databases. It has bindings for database systems including MySQL, PostgreSQL, Microsoft SQL Server and Maria DB.

Popular frameworks of this programming language for Web applications are Django and Pyramid, and microframeworks Flask and Bottle are also used. Flask is the REST API framework for the Python programming language. In relation to the MEAN stack architecture, Flask is similar to Express.js. Flask is a microframework since it does not provide direct functions such as form validation, database abstraction, authentication, etc. This microframework is used in LinkedIn and Pinterest applications.

To successfully connect the Angular frontend and Python backend part of the application, it is necessary to install classes from the flask, flask_restful, and flask_cors packages. The server receives sent requests within the Python script, which then, based on the forward path, recognizes which method the class should call, which is enabled thanks to the path forwarded from the Angular service. After that, the corresponding class from rest.py scripts is activated. An example of a method that relates to a database and returns the result of all exams of a student during the test period is given by the Listing 1.

### D. The most popular Angular libraries

Angular Material is a library that adds components and directives that accompany the material design that has been designed and popularized by Google for the needs of client web pages in web applications.

NgRx Store is a library that introduces reactive state control under the name Redux in Angular. It is used on the client side, for data and condition needs.

Ngx Translate is a library that in a very simple way enables the application to be translated into other languages, also on the client's side of the application.

Angular Gridster 2 is a library that introduces a simplified creation and use of Grid Layout with components that can be moved, resized, and dynamically added or deleted.
Angular NOTIFY is a library that allows small short notifications at the bottom of the screen.

Angular 2 Moment is the implementation of the popular JavaScript library Moment.js, which makes it easier to format and work with dates.

Fig. 3. Class schedule in the new Angular application

```
def readMyExams(json):
    connection = sqlite3.connect('db.db'.format(timeframe))
    c = connection.cursor()
    c.execute("""SELECT * FROM login""")
    connection.commit()
    for row in c:
        username = row[0]
        c.execute("""select DISTINCT R.acronym, R.name, R.date, R.period""
                  from exam R,
                  student S, enrollment U,
                  course K
                  WHERE (K.semester IN
                        {{SELECT MAX(yearOfStudy)
                          FROM enrollment)*2-1},
                        {{SELECT MAX(yearOfStudy)
                          FROM enrollment}*2})
                        AND
                        R.module = K.module AND
                        K.acronym = R.acronym
                        AND S.username = (?)
                        AND U.username = S.username"
                    )
        connection.commit()
    exams = c.fetchall()
    acronym = []
    name = []
    date = []
    period = []
    num = 0
    for x in exams:
        acronym.append(x[0])
        name.append(x[1])
        date.append(x[2])
        period.append(x[3])
        num = num + 1
    c.close()
    return {"acronym": acronym,
            "name": name,
            "date": date,
            "period": period,
            "num": num}
```

Listing 1. Example of Python script in the data layer

IV. ANALYSIS AND DISCUSSION

Table 1 gives the main characteristics of the analysis of three applications with different technologies in the server part, in which the customer part is made with Angular 7 technology.

The most important criteria for selecting backend technology are:

- business logic requirements,
- application performance requirements,
- implementation of the frontend part of the application,
- selection of a database, relational or non-relational, and the amount of data within the application,
- the need for web services, microservices or customization to some special architecture.

Due to consistency in the programming language, Angular works best with the Node.js server part of the application. Also, in combination Node.js and Angular, it opens the ability to be supported non-array NoSQL database, in our example MongoDB, or some relational database like MySQL or PostgreSQL. MySQL is the most popular relational database, easy to implement and use, but it is not scalable when there are many competing write and read operations. PostgreSQL is the best solution that works well with complex queries and competitiveness.

The research focused on the transformation of existing web applications for students e-Student and teachers e-Employees, which are the most important parts of the Faculty Information System [1]. Existing applications were implemented on public web technologies, or JSF (Java Server Faces) framework, in combination with Spring, Hibernate ORM and Primefaces library, and raised over a relational PostgreSQL database. Of the 3 pilot applications implemented, the application was executed on the MEAN stack, and slightly slower applications implemented in .NET Core and Python. Regarding communication with the database, in each pilot application, a database has been tested with the best way to communicate with the server part. In the field of web application testing, all technologies implemented on the backend side have had good support for the unit and integration testing tools.

At first glance, JSF and Angular in the frontend have a lot of similarities, they offer a good level of abstraction, two-way data binding and use components to build an application. Also, both technologies have the ability for the developer to create special components that he can use, with the exception that it rarely happens in the JSF, while in Angular this is common. The main difference is that Angular provides client-side components, while JSF provides server-side components. When using JSF technology, the whole business logic and connection with the database is done on the server side, as well as generating the user interface in the form of HTML, CSS and JavaScript. As far as Angular is concerned, the web browser has a much larger role, and for this reason the server side becomes less loaded, and easier and faster to run. Table 2 gives the key advantages and disadvantages of JSF and Angular technology.
Also, Angular is completely independent of server technology, so the entire server implementation can be changed without changing the GUI, as opposed to JSF technology.

Due to the good impression that MEAN stack has left in our research and other research, [5] the same technology is included in the Programming Internet Applications program from the school year 2018/2019, where it replaced JSP and Java Servlet technology. Also, students have been given the opportunity to choose for a practical project whether they want to implement it in JSF technology or in the MEAN stack. The average number of students’ points is given in Table 3. The results of the introduction of Angular and MEAN stack showed that despite the less literature and lessons learned, they did much better tasks with Angular technology and that a greater number of students based their project on a more modern MEAN stack, compared to an alternative stack based on JSF technology. This combination of technologies is also included in many other universities in the world [6].

Table 3: Analysis of the success on the subject Programming Internet Applications in the period October 2018 – February 2019

<table>
<thead>
<tr>
<th>Category</th>
<th>Task #1 JSF stack</th>
<th>Task #2 MEAN stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of points on colloquia (%)</td>
<td>56.67%</td>
<td>70.60%</td>
</tr>
<tr>
<td>Average number of points on final exam in January (%)</td>
<td>44.85%</td>
<td>62.97%</td>
</tr>
<tr>
<td>Number of students who have chosen these technologies for the project</td>
<td>11</td>
<td>34</td>
</tr>
</tbody>
</table>

V. CONCLUSION

Trends show that in server-side applications, Python is progressing and will soon take over the primacy of PHP technology. At the same time, Node.JS also has an increasing number of applications that have applied that technology on the server side.

The analysis has shown that MEAN stack today is the best combination for pairing the backend of the application with respect to the frontend realized in Angular technology, considering the performance and speed of the application, communication between the client and the server, and the support of other libraries.

REFERENCES


Conclusion of comparing the existing front-end technology in FIS, Java Server Faces and Angular, shows that JSF is better for applications running on one server or for small clusters, but its scalability is therefore small. Angular has a better response, easier and faster development and maintenance. To use Angular, it is sufficient to know TypeScript and to find out which libraries to use from a large number of available ones.
