**APPLICATION API DESIGN AND IMPLEMENTATION WITH SPRING FRAMEWORK**

Tahir Turgut

27035

26/07/2021 – 20/09/2021

Commencis Teknoloji A.Ş.

Aslı Gönülşen

01.10.2021

Faculty of Engineering and Natural Sciences

Sabancı University

Computer Science and Engineering

**ABSTRACT**

During my compulsory internship project, I have work with Commencis Technology. Commencis is a company that offers help to the brands in digital strategy, product design, development, quality assurance, cloud transformation, and growth marketing platform. They have profession on creating products, analysing big data, digital transformation process of companies that they are working with.

In my internship project, it is explained to me that nowadays most of the companies transform their associated workspace to the digital somehow. However, it is also emphasized that companies like Commencis are also in need of internal technologies to enhance inside interaction of the employees. After some research and discussion, we have decided that an internal technology to be used in company would be a great example of digital transformation. It is also discussed that, this project can be carried out with Spring Framework, which is one of the most used technologies with Java, and database connection can be handled via JPA and Hibernate technologies, which are again among the most popular technologies. Also, database system can be designed and implemented via MySQL, we thought.

To indicate, I have learned a lot, thanks to my supervisor. Furthermore, working with the latest and most popular technologies had a great impact to my sector knowledge. On the other hand, working remote can reduce the efficiency of learning, obviously.

I can recommend that, even though internship is conducted online, that time can be used to enhance your professional network and socialize a little bit, as well.

**Table of Contents**

**1.** Introduction ........................................................................................................................... 4

**2.** Company Information ........................................................................................................... 5

**3.** Project Background ............................................................................................................... 6

3.1. Department information ......................................................................................... 6

3.2. Status of the Project or the Problem at the beginning ............................................ 6

3.3. Motivation and/or Problem Definition ....................................................................6

3.4. Related Literature ................................................................................................... 7

**4.** Internship Project .................................................................................................................. 9

4.1. Project Objective .................................................................................................... 9

4.2. My Responsibilities .............................................................................................. 10

4.3. Methodology / Tools ............................................................................................ 10

4.4. Expected Outcomes and Deliverables .................................................................. 10

4.5. Details ................................................................................................................... 11

4.6. Results .................................................................................................................. 17

**5.** Internship Experience ......................................................................................................... 18

5.1. Learning ............................................................................................................... 18

5.2. Relation to undergraduate education .................................................................... 18

5.3. Difficulties ............................................................................................................ 19

**6.** Conclusions ......................................................................................................................... 20

**7.** Recommendations ............................................................................................................... 21

**8.** References ........................................................................................................................... 22

**1. INTRODUCTION**

Nowadays, digital transformation processes are crucial for the firms. Moreover, internal technologies have become more important, in order to increase the productivity within the company and communication network between the co-workers of the firm. That is why, by the purpose of solving an internal -to a closed community, more specifically for the employees of the company- problem is as important as creating a digital solution to another company.

For the purpose, which is mentioned above, Commencis have created an application project to be used inside the company. That application is designed to provide an environment to employees to sell their second-hand belongings to their colleagues. It is planned to serve in 3 platforms: web; mobile -more specifically, iOS and android. Also, profiles of the users are derived from their internal mail addresses, names etc. Therefore, there is no need to register while using this application, only authentication with Commencis credentials is enough.

In this report, information about Commencis, background knowledge about the technologies that is used for this project, details of project and its implementation, what is learnt and experienced during this internship and finally conclusions and recommendations will be presented, respectively. In final page, references that is referred in that report can be found.

**2. COMPANY INFORMATION**

|  |  |
| --- | --- |
| Full Title | Commencis Teknoloji Anonim Şirketi |
| Address | Park Plaza, Reşitpaşa Mahallesi Büyükdere Cad. No: 223 B, 34467 Sarıyer/İstanbul |
| Website | https://www.commencis.com/ |
| Contacts | 0 534 406 98 34 – Aslı Gönülşen |
| Number of Employees | 200+ |
| Industry | Software Technologies |

Commencis has 21 years of sector experience, starting from *pozitron* which is founded in 2000. Then, another company, called *Monitise*, has purchased positron in 2014. Finally, *Monitise* was purchased by former CEO of *pozitron* in 2018 and its name has been changed to Commencis. Company has also another associated company in Berlin, called *finbyte*.

It has 5 offices: two in Istanbul, ITU Teknokent and Park Plaza, Maslak; one in İzmir IYTE Teknokent, one in London and one in Berlin.

Commencis is a technology company that helps brands via building digital transformation paths and products. Company also serves in big data, analytics, and cloud products. They have also a big data analytics software, called *Dataroid* as a sellable product and semi-independent department. In general, Commencis works withs brands in the following areas: Banking, Consumer goods and services, Travel, Insurance, Retail.

Commencis have very well-known customers such as Türkiye İş Bankası, Mapfre, Pegasus Airlines, Denizbank and more like them.

**3. PROJECT BACKGROUND**

**3.1. Department Information**

During my internship, I have worked under the engineering department. This department generally is in contact with all other departments since all other departments use my departments’ endpoint designs. Moreover, the tasks related to database, creating endpoints and open API configurations are handled in that department.

I have closely worked with my mentor, Saltuk Erkan, during my internship. He also works in Engineering department as Software Engineer. Mail: [saltuk.erkan@commencis.com](mailto:saltuk.erkan@commencis.com)

Also, our supervisor, Aslı Gönülşen, can be reached via mail: [asli.gonulsen@commencis.com](mailto:asli.gonulsen@commencis.com)

**3.2. Status of the Project or the Problem at the beginning**

First of all, a kick-off has been planned to start project and assign the related tasks with the related workers. In that kick-off, a workflow is also presented via figma. Priorities has been set, and starting point are emphasized accordingly. To exemplify, in my specific case database design and creation was prioritized among the others, since it has to serve as a base for the endpoints afterwards. Also, mock server creation tasks are assigned, in order to not block other departments with endpoint implementation.

To conclude, before project has started, those tasks are discussed, prioritized, and distributed among the project assignees. Those plans are shared with the others when the project kick offs -a meeting which lasts 2 to 3 hours.

**3.3. Motivation and/or Problem Definition**

After the pandemic, some of the employees relocated -to their hometown or away from Istanbul etc. Because of that relocation traffic, somehow the need of selling their belongings is also emerged. Therefore, the tool that we implemented was a necessity to provide them an environment to sell the surplus, somehow.

To solve the above problem, an application is designed as a solution. The target audience are company employees, and it is expected that they might want to reach that application in anywhere, anytime. Because of that, the tool to be created, has to be reachable in any platform. That is why, the application is planned to be served in three platforms: web, iOS, and android.

**3.4. Related Literature**

First of all, the application would require a database, basically. That database should be well designed and planned before project implementation starts and it should be finalized as much as it can be. It is planned and visualized through ER diagram.

Secondly, request and response models of endpoints are discussed. From the template, which is shared via *Zeplin*, team have decided which endpoints should be created and which parameters has to be taken to handle those requests. After that, according to designs, we have checked whether each page of the application is covered or not with a particular endpoint. Finally, those models are created via Swagger Open API.

Entity-relationship diagram (ER diagram or ERD in short) is a diagram that shows the relationship of entities or entity sets in the database design. It is used to simplify and explain the logical background of the structure which is used -or to be used- in database. In general, three basic components are used in ERD: relationships, entities or entity sets and attributes. Different shapes are used to distinctly represent those components. For instance, circles or ellipses are used to represent attributes, whereas rectangles stand for entity or entity sets.

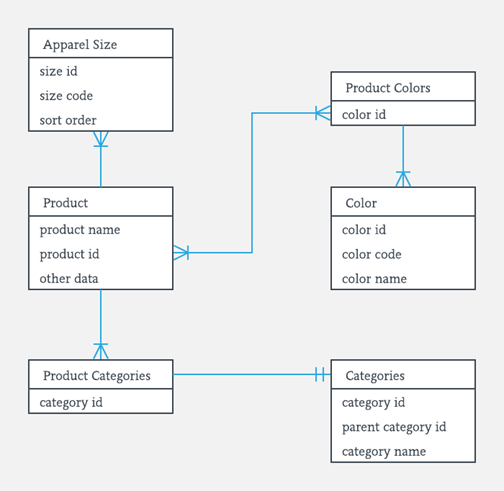


Figure 1: An example of ER Diagram

Swagger is a set of API developer tool, which is created by SmartBear, that specifies a formal standard methodology for REST API’s. For this project, Swagger Codegen is being used. It is a tool for generating models and their implementations as they are specified in a “.*yaml*” file. As it can be observed in the Figure 2, the *yaml* configurations on the server can be served to users, with a good-looking UI, that they can easily understand request and response model of specific endpoint that is created via configuration.

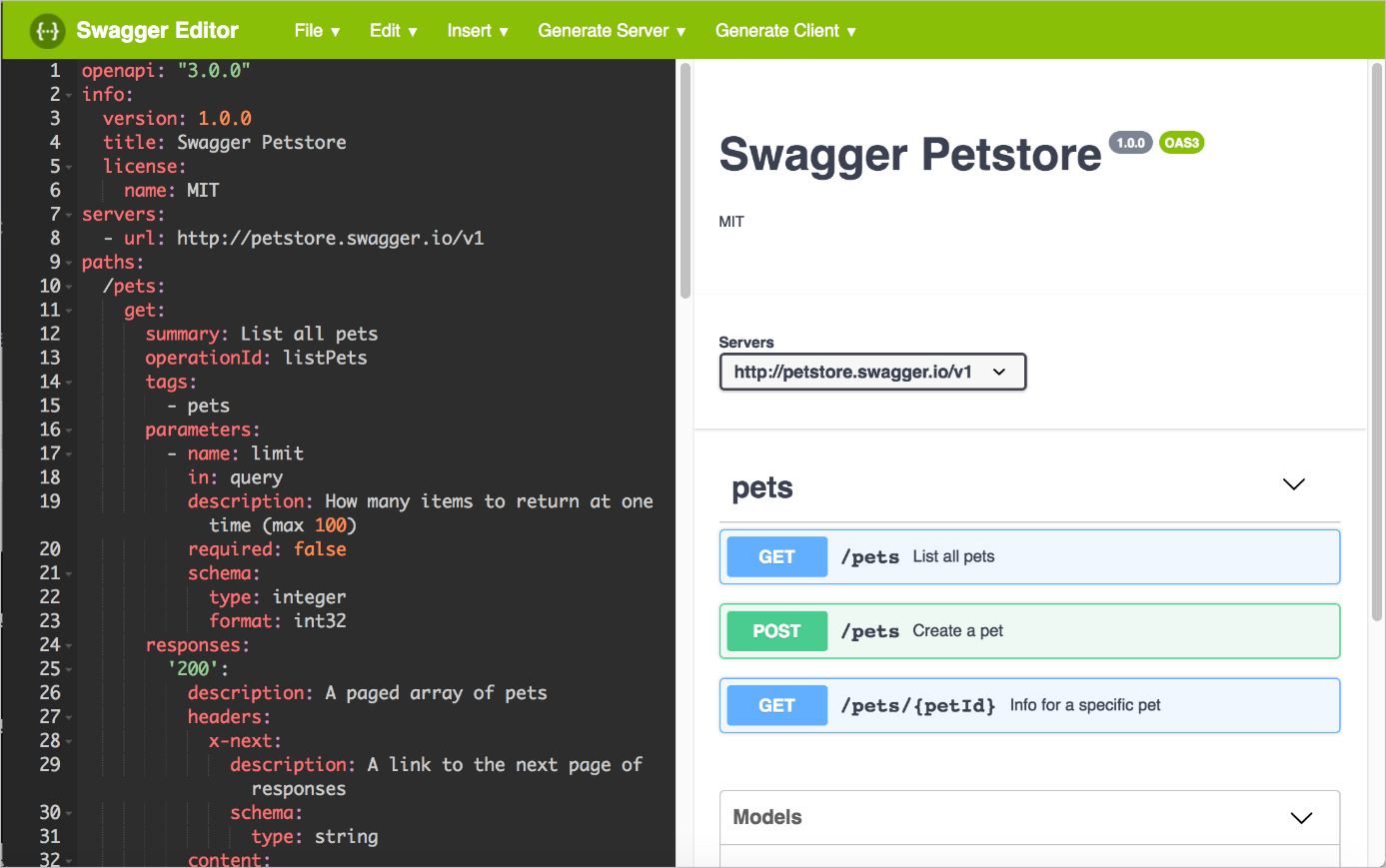


Figure 2: An example of yaml file and its server-side representation

**4. INTERNSHIP PROJECT**

**4.1. Project Objective**

The project aims to develop a software for an application which will be used for providing an environment for employees to sell their second-hand belongings. This application will be developed in 3 platforms, as mentioned above. It will be open to a closed-community (Commencis workers), however, after a while if it succeeds, it might become available for other companies in the same manner like in the Commencis.

It is designed to solve a problem which is mentioned above -when people relocate, they might need to sell some of their belongings-, it answers that problem directly by providing environment to sell those belongings. Its scope is also defined closed since register and login system are integrated with Commencis mails.

**4.2. My Responsibilities**

In this project, I have worked with the backend team. I have designed the database and initialize that by using JPA. After that, I oversaw handling the changes in database. I have also designed some endpoints; it will be mentioned in the following sections. Furthermore, I was in contact with the other departments to explain what is provided to the front side, and what is expected for the requests from client. In addition to those, I was responsible of logging the users’ interaction with the database. For instance, when a user adds a product, it is processed to database and that is logged, in order to make possible future programmers that will deal with that application can detect the bug cause.

Also, there were other departments that are handling the client side of the project for example iOS, Android, and Web Frontend team, they were getting help from a designer from company.

**4.3. Methodology / Tools**

The project was designed in Java programming language. Spring boot/framework technology is used to handle endpoints. Also, designed endpoints are configured via an open API tool called Swagger. Furthermore, database connection and query requests are handled via JPA and Hibernate. For the database side, MySQL is used as workbench, whereas for database design I have used an online drawing tool called draw.io.

**4.4. Expected Outcomes and Deliverables**

As a result of that project, an application is expected to be produced as a deliverable, in 3 platforms. In that application, users will be able to post their items that they are willing to sell, edit their profiles -how their profile looks- and most importantly, show other items which is open sale. There will be a feed which contains items that satisfies the conditions of filters or orders, and a detail page for each of the items. In addition to those, a profile page and a page that contains that specific user’s items on sale will be implemented, too. At the end of the internship duration, a demo will be presented to team leads.

**4.5. Details**

In this section, preparation process for the project will be served, firstly. Then, project kick-off and database design/implementation part will be explained. Furthermore, project’s open API configurations will be presented. Before resulting the section, project’s endpoint design and implementation will be highlighted. And finally, result of the project and the demo will be mentioned.

For preparing the project, before the kick-off in other words, I have worked with Spring framework, JPA and Hibernate, just to get familiar with the technologies that I will be using during project. I have also used Swagger Codegen, as it is mentioned above it will be used in the project, as well. In order to getting used to those technologies, 3 endpoints are designed and created and already deployed services are used to handle requests. First endpoint was for login service, which is already deployed again, the only thing to experience was getting username and password from client and call the appropriate service to handle rest of the process. Then in second endpoint, user’s information is retrieved. To do that, user credentials that is created by login service is used. Finally, from the user credentials, their bank accounts are retrieved via another appropriate and already deployed service.

As another preparation, after the project is designed by designer team, the requirements have been calculated and necessary work division are decided, accordingly. For instance, by checking the feed page that is designed, backend team has decided that an endpoint should be created to return all the items from the database. After that, entity requirements have been shaped, which item specification does client side needs and how should those specifications be stored in tables are decided. The details of database design will be presented below paragraphs.

When all the requirements are met, teams are set, we have settled a meeting for project kick-off. In this kick-off *Jira* integrations, *Jenkins* authorizations and other necessary arrangements have processed. I, as representative of my part, have presented the endpoint designs, planned configurations for the models and finally entity models that will be used in database. After the feedback process with other departments, database design took its final form. First tasks are assigned to related team members. I have taken, firstly, the database design and implementation task as an urgent task since it is the main framework of the project’s backend.

For the database design, I have worked with a drawing tool called draw.io, to be used to get feedback and develop further, before implementing it. Unfortunately, internship duration is over, and I cannot reach those files anymore. However, database was consisted of 2 main entities: Item, User. Whereas there are side entities to help extract exact properties of those 2 main ones, such as Photo, User Activity Log and Category -which will be implemented later as an additional feature-. Item entity has everything related to a post in feed, whereas User entity has related to a post’s owner or specifications of users as it can be seen in the figure 3.

Photo entity has three basic parts: encodedPhoto, encoded byte array version of the photo; isCoverPhoto, to emphasize whether photo is the photo to be displayed in feed; and as a foreign key itemId, to relate photo with the specified item.

User Activity Log entity has three basic parts, as well: User Id as a foreign key, to underline which user has done that activity; Item Id as another foreign key, to specify which item is under effect of that activity; and finally, Activity Type enumeration, there are pre-defined activities such as displaying, adding to favourites -out of scope of this project-, buying the item -again online payment out of scope.

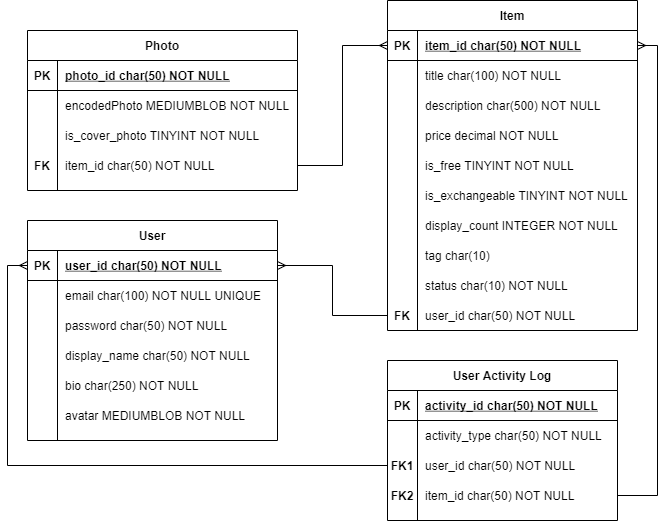


Figure 3: Database design of the project

Those entities that are specified above has been implemented via JPA. As it can be seen in Figure 3, annotations that is provided with JPA has been used. That example belongs to User Activity Log, but other entities are created in nearly same manner. JPA is able to convert those properties into column specification for a table. As it can be seen, Data, Entity and Table annotations are used to specify that model will be used to create a table. In below of the figure, Column annotation is used to again specify that stands for a column in the table. Also, column’s name can be set via a parameter. Finally, ManyToOne and JoinColumn annotations are used to create relationship between tables. As it is mentioned above, User and Item data types are other entities, by stating referenced column name as parameter, JPA direct the join operation into specified column in that parameter, “id” in that case. Many to one relationship is most sensible type, since There might be more than one activity per user in that table, whereas per one activity there cannot be more than one user, same goes for item.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 4: User Activity Log implementation with JPA

Beside of those, I was responsible of open API configurations of the project, as well. As it is mentioned before, Swagger Codegen is used for this purpose. In the Figure 4, an example configuration can be seen. Other configuration has slight differences in request parameter part, which will be explained at the end of this paragraph. This path is designed for profile editing page. The bio and avatar changes -if that exists-, has been retrieved from client side and if it is necessary -not in this case, actually- it is casted to another type that fits the database models and entities at the first place. In that case, the request has been retrieved as the type of ProfileEditRequest which is a structure that stores bio -as string- and avatar -as byte array. Then, in the service implementation, they have been processed into database -user’s related information is being updated.

“/profile/edit” part specifies the path that endpoint is called by client, in one line down, “post:” specifies the request method to be created, it might be *get* or *delete*, as well. On the other hand, “tags” part defines which group will that endpoint belong to. Security part was out of my scope, a particular token associated with specific user has been created by authorization package. Whenever a request has been called by client, that token is being checked whether it is valid -belongs to a real client or not. However, to check whether user is authenticated or not, that security part specifies the parameter to be checked for validation. “requestBody” part defines the expected parameter from the client, in that case it is an actual body that contains more than one parameter, it might be different for another endpoint designs. “required” part emphasize that request body cannot be null, it has to be retrieved, otherwise an error will be returned (400 bad request error response). Finally, in “responses” part, error and success responses are defined. Whenever service returns 200, its response body will be empty -because it is not specified. However, when the returned http code from service is 400, “ErrorResponse” schema will be returned as highlighted in content and schema parts, respectively. “application/json” parts stand for that response/request bodies will be in json type.

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 5: An example endpoint yaml configuration

In addition to that example, there might be path or query parameters rather than request bodies. In figure 5, a path variable can be seen. They are attached to path that request has been sent. For instance, in “/item/12345” path, “12345” has been retrieved as a parameter which is called item id in that specific example.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 6: An example path parameter design

Finally, when there is a response body in success case -a response that client side should retrieve- it has to be casted from an entity to appropriate from -ItemDetailResponse- in the figure 6. That process is handled via a converter file that uses Spring Framework. As it can be seen in figure 7, what needs to be casted is specified one by one such that Item’s title has to be casted to ItemDetailResponse’s title etc. At the end, by using the related entity model, the data has been transformed into response model of specific endpoint.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 7: Response model of item detail endpoint

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 8: Converter of Item to ItemDetailResponse (appropriate body for a specific endpoint)

**4.6. Results**

During my internship duration, project has not been released as a product. After the demo, they have said that they will adopt it after a few changes and optimization, but the time has not specified exactly.

I have configured all the endpoints that essentially needed for the project and database was running properly. As far as I have informed, endpoint implementations are over, as well. Actually, on the last week, I worked to implement logging system for most of the services, for the purpose of easing the future programmers’ job when they have to find any bug in the implementations. Even though it is out of the scope of the project, logging for database interactions has been implemented for future programmers that will work on this project, as mentioned before.

**5. INTERNSHIP EXPERIENCE**

**5.1. Learning**

To begin with, I have learned how to use Springboot to create REST API’s and how to connect those APIs to servers or database via JPA and Hibernate. In addition to that, I have learned creating configuration files with Swagger Codegen, which can be used to create models to be used in program.

Also, I have experienced enough -for an undergraduate student, of course- in backend engineering. Moreover, I can say that backend team in software engineering can be suitable for me. I will be thinking on focusing that particular area in software engineering, maybe. However, it made me understand that I am comfortable with software engineering, and I want to try other departments such as iOS developing, Android developing, Web Frontend Developing, and maybe Full Stack Developing etc.

**5.2. Relation to undergraduate education**

First of all, I have learned that my academic background was enough to adopt any kind of new technology in the industry. I can manage to gain sufficient experience in Spring Framework within 1-1.5 weeks, even sufficient that can be used to create a project from scratch.

Furthermore, I have taken the Software Engineering course, which has objectives such as design patterns, agile working methodologies. Also, we are obligated to create a project in the course that an e-commerce application which has web and mobile parts. In the course itself, knowing agile methodologies saved enormous time, while adopting the company in first weeks, since they arrange daily meetings and add me to their 2 week sprints. Moreover, I have created APIs in the project of that course, and I was responsible of backend with my 2 other friends. Even though we have worked with Javascript and node.js in that project, it gave me the insight of how to design an endpoint and implement it via using pre-deployed libraries. That is why I have not struggled with any of those parts, I just had to learn a new technology like I have learned node.js, Spring Framework.

Maybe in CS201 or CS204, clean coding skills can be checked in homework, because students generally develop bad coding habits such as creating unmeaningful variable names, duplication of same strings again and again. However, it can be prevented at the beginning, one of the introductory courses of CS major.

**5.3. Difficulties**

Firstly, I was unexperienced in Git usage. Whenever I create something to push the server, I had hard times. To overcome that, I have arranged a meeting with my mentor and asked him to make one commit with me to make me learn how to use it afterwards. It actually worked and I repeated those steps for whole project.

Secondly, I have had also issues in communication because of the COVID situation. It was really hard to arrange a meeting and communicate via Microsoft Teams, sometimes internet connection is lost -it is extra slow because I have to use VPN to connect remotely to company’s server- and the point is missed by me. It was impossible to overcome, unfortunately. However, my mentor somehow created a strong communication channel via chat since it is more stable than meeting. Whenever I had a question, I asked that on Teams chat, and got answer within minutes.

Thirdly, writing clean codes was tough to adopt. Since I have developed bad habits in my homeworks or projects, it was hard to get what is asked by requesting clean codes in the project. I thought that my code is understandable in general, however it is not proper for other programmers to understand my way of coding. To overcome, I had watched videos about clean coding which is assigned to me via Udemy by my company. It actually worked, since I have seen many bad coding habit examples in that course -most of them were familiar to me.

**6. CONCLUSIONS**

During my internship process in Commencis, I have developed an application which is designed to serve company’s employees and have the purpose of providing an environment to them to sell their second-hand belongings. This application will be run in both mobile and web. This project has been designed to be created with using Spring Framework -web service implementation- and JPA and Hibernate -database connection with services. That is why it can be said that internship project aims to prepare interns to trending technologies in industry which can be exemplified with above technologies. In this application, users can search for specific features such as open to exchange or a keyword in the feed. In addition to that, each user is able to upload an item to feed and edit those items afterwards if this is necessary. Moreover, items in feed can be ordered by their popularity -display count, or upload/edit date etc.

After the internship project is over, I have observed that agile methodology is being used in industry a lot, and I have found chance to experience those methods in a full-time workspace. Also, I have gained confidence about learning new technologies, since neither Java nor Spring Framework was familiar to me before that internship. However, I have managed to adopt those technologies in a short period, thanks to background that I have learned in school.

**7. RECOMMENDATIONS**

It is important to take course the related course and internship beforehand. For instance, it will be hard to conduct a software engineering internship without successfully passing CS308, or CS310 and mobile development can be counted in that manner, as well. In addition to that, students should not be afraid to learn new technologies, because the background that is taken via those related courses, I mentioned above, is sufficient to adopt any new technology, since their base logic is totally the same.

However, it might go wrong, as well. Sometimes, students were not aware of the review processes and think that every update that they do will show its effect immediately. Unfortunately, this is not the situation; other colleagues should check your code whether it can be improved further -properly working code is not sufficient for approval, obviously. Also, they should not expect immediate result of the project, too. It was the case for me, to be honest. Because of the experiences that I had in my homework or projects, it seems like whenever you finished to coding part -actually there should be also a design part, as well, a perceptible result is expected. Nonetheless, processes must be reviewed and checked again and again in the industry, be aware of it.

**8. REFERENCES**

Commencis. About Commencis. Commencis. Retrieved September 30, 2020, from <https://www.commencis.com/company/>

Spring.io.  Introduction to Spring Framework. Spring.io. Retrieved September 30, 2020, from <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/overview.html>

Peterson, R. (2021). ER Diagram: Entity Relationship Diagram Model | DBMS Example. Guru99. Retrieved September 30, 2020, from<https://www.guru99.com/er-diagram-tutorial-dbms.html>

Swagger. About Swagger. Swagger. Retrieved September 30, 2020, from <https://swagger.io/about/>