Report

Faculty of Science and Engineering

Computing Science

In-Company Internship at Belsimpel

Author:
Evrikli Xhelo

Supervisors:
Rein Smedinga
Marc Holterman

July 2020

This report finalizes the in-company internship that I conducted at Belsimpel throughout ten weeks, from April 13th to June 22nd. The report discusses in summary the company where the internship took place, the main project that I was assigned, the key organization matters of the company, a brief conclusion of my experience and an appendix of the weekly tasks that were completed during the internship.

Company

Belsimpel The company currently operates in seven stores that are placed throughout the Netherlands and there are about 450 colleagues that work in different departments of the company. Given my experience and the work-preparation documents that I studied, I find it that Belsimpel thrives because of its main values: (1) *commitment to the customer's requests* (2) *positive outlook* on present situations; accepting them as they are and always aiming for improvement (3) practical, smart and confident *employees* (4) *reliability*.

Company history The history of Belsimpel dates back to 2006, when two students, Jeroen Doorenbos and Jeroen Elkhuizen, decided to take up a challenge. They saw an opportunity by noticing the gap in the market for telephone subscriptions and established *Metjelaptoponline.nl*, which in 2008 came to be known as Belsimpel. While conducting their studies, the team of founders stored a warehouse in the shared area of the student house where they used to live and converted the kitchen into customer service. Later on, an office building in Grote Markt (Groningen) became available and they seized the opportunity to use them as their own. Ever since then, the company would grow every year, improving the market that they serve while gaining the trust of their customers and winning yearly awards at the same time.

Market The market that the company operates in is the telecom market. Belsimpel offers customers a wide range of services, from phone subscriptions to mobile phones, mobile phones with phone subscriptions as well as a multitude of electronic accessories, including headsets, cables and tablets.

Assignment

In this section, the main project that I was assigned throughout my internship is discussed. A detailed overview of the context, followed by the main aspects and concepts that were a crucial part of developing the project, is presented.

Becoming acquainted with the new system(s)

From the beginning of my internship, I was assigned as part of the Operations-2 development team. This team is responsible for developing and implementing functionalities that are concerned with the warehousing system at Belsimpel. There are two main systems that the Operations-2 team is responsible for: *Alice* and *bo2*[1]. Both systems are used for crucial warehousing tasks starting from looking up purchase orders, to the stock in the warehouse, to packaging and shipment details as well as specific product details. Throughout my internship, my main project was concerned with the Alice system.

During the first week of my internship, I was assigned several minor tasks, purposely selected to help me get acquainted with Alice and how it was developed; Alice is one of the more recent systems in Belsimpel, with new updates and subsystems on a weekly and monthly basis, respectively. In the course of a week, I was introduced to the concept of '*searchers*' - a very important part of Alice, how the MVC pattern is implemented using `AR` Models, Controllers and templates

[1] back office 2

for the View, how to write unit tests and how to use the database. Later on in the internship, these concepts would become essential to my main project. Another important aspect that I was introduced to was the translations. Every Alice page [2] is available in Dutch and English, and it is important that when developers work on a certain page, they also provide the proper translations for it.

In the third week of my internship, I was introduced to my main project, a greater task that would acquire the implementation of a new system for Alice. The context for this project was the increased necessity to understand the productivity of the colleagues that were performing *order-picks* and *packs* in the warehouse. Belsimpel, like many other companies, had to adhere to new measures due to the Covid-19 pandemic. As such, day-to-day tasks became more difficult to follow. The purpose of the system that I would develop was to help colleagues track their productivity during the day and discern whether there were any aspects which could require improvement.

In the following sections, I will describe in more detail the project itself, which is called *'Productivity per Workflow'*, and some of the main concepts that were associated with it, starting with order-picks and packs.

Order-picks and packs

Order-picks and packs are two important concepts concerned with the 'Productivity per Workflow' project. An order that is placed by a customer via the Belsimpel website may consist of one or more products. *Order-pick is the action of 'converting' an online order into a 'physical' one by picking all of the products it consists of from their appropriate locations in the warehouse.* While doing so, the action is also recorded in the database as order-pick. The action of order-pick is executed by any Belsimpel employee who has access to the Alice system.

There is a specific system in Alice (let us call it **x**) that is dedicated specifically to order- (and product-) picks. Essentially, a product-pick refers to two things simultaneously: (1) physically picking a product from a shelf in the warehouse [3] and (2) using system **x** to mark the product as picked. In short, **x** is used to record the information about the product-pick actions in the database. Finally, an order-pick is completed (and, thus, recorded in the database) when all the products that comprise a specific order have been picked.

When an order-pick has been finalized, the next step that needs to be executed is order-pack, which refers to the action of preparing an order ready for shipment. There is also a dedicated system in Alice, let us call it **y**, specific to rendering this action. Similar to **x**, **y** is used to record a colleague's action of packing orders as an order-pack action in the database.

The storage of the aforementioned actions will be used extensively in the development of the 'Productivity per Workflow' system.

Action measurements

In Alice, we refer to actions like order- (product-) pick and order-pack, as action measurements. These actions are stored in the database and they can be used to retrieve other relevant information such as which colleague performed the action, what time and date was it executed and more. Consequently, we can think of an action measurement as an *object* that can give us information about the type of action (whether it is order-pick or product-pick for example), the executant -

[2] A page in Alice could refer to an endpoint/user-interface/sub-system, essentially, whatever the user can see and interacts with in Alice.

[3] We are excluding other details of the physical actions that the colleague has to perform, such as scanning the IMEI or EAN code (both are types of bar-codes that are used to uniquely identify products).

the colleague that performed the action - as well as the time and date of the execution. This type of object is essential to the system that I implemented.

Reports

As mentioned previously, Alice is comprised of several different types of systems which offer a multitude of functionalities, a *report* being one of these systems. Reports are used to present information. They are crucial because they depict a clear and precise overview of certain concepts that are required by the working departments. The 'Productivity per Workflow' project that I developed is also a report and will be referred to as such in the forthcoming sections.

From a developer's perspective, a report is merely a system that implements the MVC pattern. In the next section, we introduce the project and present more details about the report that I implemented.

Productivity per Workflow

Description The 'Productivity per Workflow' is a report that indicates how productive our colleagues have been during one day. The productivity is showcased by merely counting the *instances* of the action measurements that have been finalized by colleagues on a certain day. The workflow refers to the *type* of action measurement objects that have been finalized. More precisely, we have two different workflows: *picking* orders and *packing* orders. The action measurements that are associated with the former are: (1) product pick (2) order pick and (3) order skip. The action measurement that is associated with the latter is order pack.

In Figure 1a we present a mock-up that the product owner prepared for the 'Productivity per Workflow' report. Each workflow is merely a table where the headers are separated in time-slots of one hour, from 7:00 to 23:59. The left-side headers state the type of action measurement.

The **'Totals'** refers to the cumulative sum of the number of instances in each time-slot. Thus, for example, if 23 products were picked from 7:00 to 8:00 and 45 others were picked from 8:00 to 9:00, then the 'Totals' for the former time-slot would be 23, whereas for the latter, 68. In conclusion, this report indicates precisely how many action measurements are finalized on one particular day, from 7:00 to 23:59, every hour.

Planning The Operations-2 development team works with the Agile methodology. This means that the product owner presents the main task to the developers and an initial meeting is scheduled, where sprints and versions are discussed and set in collaboration together. Every version is an MVP [4] that consists of certain functionalities. When a version is released [5], the developers start working on the next iteration while also applying the feedback that is gathered from the former version(s).

On my first project meeting with my product owner, an overview of the task was presented and the initial versions were discussed. Throughout my internship, three versions of the 'Productivity per Workflow' report were released. I will provide a brief description for each one of them.

V1: Static productivity workflows for all the colleagues.

This was the very first version of the report that I developed. The goal was to depict the workflows as presented in the mock-up (1a) on the day that the report page was loaded by the user in Alice. That is, if the 'Productivity per Workflow' page was loaded on May 1st, 2020, then the

[4] minimum viable product

[5] When a merge request is approved and merged into the development branch, it will be released and placed in production.

report would depict the action measurements results that were recorded on 01/05/2020. Figure 1b depicts the results of releasing V1.

V2: Static productivity workflows for every colleague by name.

This version was a direct extension of V1. In addition to calculating the overall action measurements that were performed on the day that the page is loaded, we would now be showing how productive each one of the colleagues had been, that is, how many action measurements were performed by every specific colleague. Therefore, each one of the initial workflows would be extended by a new section called *'Per Colleague'*. This section includes as many tables (of the same structure as the one in Figure 1b) as there are colleagues who performed action measurements.

V3: Make the report dynamic by adding a date-picker.

In this version, the report page is dynamic. The users are able to see the same information as in V2, however, they are now able to choose any date they prefer, as opposed to only being able to see information for the day that the report page is loaded. In Figures 1c, 1a and 1b we depict the final result of V3, as was released in production.

Future work

The 'Productivity per Workflow' report page is quite relevant to the warehousing team. Therefore, there are more features that can be developed in the future, one of them being a time-picker, where the users can select a preference of the time-slots for which they would like to see data. At the same time, it should also be possible that all one-hour-ed time-slots are present in the workflows as opposed to only having the time-frame between 7:00 and 23:59. Another possibility is the addition of more action measurement objects and, thus, more workflows as well.

Organization

In this section, a summary of the key matters and organization principles of the company are discussed, based on my experience as an intern as well as the official documents of the company.

Teams The Belsimpel company consists of several departments that collaborate together, in order to help all matters run smoothly. The main departments are the team of developers, the team of product owners and the HR department (human resources). The team of developers is responsible for implementing new features and maintaining the codebase. Given the size of the company and its key principle to build everything 'in-house', there are several development teams, each of which working on specific matters, separate from other teams. For example, there is a development team that works specifically on developing the HR system, another one on developing the company's application and more. I was assigned as a member of the Operations-2 team, which works specifically with matters of the warehousing system.

Developers work closely with the team of product owners. The latter is responsible for creating *tickets* which describe new features, functionalities and/or systems that need to be implemented. Developers and product owners work together to establish sprints and versions for each of the tickets. It is usually the case that one ticket is assigned to one developer, but some tickets could have overlapping concerns, in which case, the corresponding developers and product owners collaborate together and discuss potential solutions. Furthermore, it is also possible that one developer is working on more than one ticket at the same time. There are no hard deadlines assigned to tickets, however, if it is the case that one feature is very important for a particular reason, then the developer responsible for it dedicates most of the time to its implementation.

There are no 'enforced' processes in the company, however, the usual flow of communicating issues between the teams is twofold: the team of colleagues that works first-hand with the company's systems (both physical and software) communicates their suggestions (or requests) to the team of product owners. The product owners, in turn, address the suggestions to the development teams by creating tickets that resonate the main concerns that are put forward.

Communication Throughout my internship, I discovered that communication is very important to the company on a small *and* wide scale. It is important that we, as employees, communicate any concerns or ideas appropriately. It will always be the case that someone will be able to guide us with a solution or suggestion for a next step. At the same time, I discovered how important communication is within the team that I was working with as well. The Operations-2 team would have daily meetings to discuss our goals and tasks. Furthermore, being an inexperienced software developer, the possibility of communicating with any one of my teammates helped me get up-to-speed early on in my internship.

Under normal circumstances, all the development teams have a *General* meeting every Monday, where the progress of every team is discussed. This has been an essential part of understanding the latest features that the company is offering to the customers and to its employees as well. Furthermore, every team also has a weekly meeting, called *Team* meeting, with specific details on what is currently being developed and what will be developed in the near future.

My function in the company and future improvement My internship was completed fully remotely. Due to the Covid-19 pandemic, throughout all the ten weeks, it was not advised that colleagues and teammates meet up together and, therefore, my experience is greatly based on this special situation. I found the daily meetings with the Operations-2 team (day-start and day-end calls) useful and helpful; it helped keep the morale up and we were also familiar with everyone's day-to-day tasks. On Mondays the meeting was more official, in that one of the team members would keep minutes. This was a substitute of the *Team* meetings that would occur before the pandemic.

No replacement for the *General* meeting was established and we were informed about the progress of other development teams via PowerPoint slides. This would be the only matter which I believe could be improved for the future. Given that the teams will continue working remotely for the time being, it could be a good idea to possibly replace the General meeting with an online 'meeting', where the teams also get to explain what they have been working on during the previous week. This matter was also brought up among the employees; the idea was put forward and considered as a good suggestion, therefore, it could be the case that it is established in the near future.

Conclusion

By the end of my internship, I developed a system which increases the efficiency of the warehousing team, by offering insight into the productivity of our colleagues. The 'Productivity per Workflow' report will also be used by product owners to analyze and, if necessary, improve their current metrics.

The opportunity to implement such a system from ground zero has allowed me to enforce the software engineering principles that I have been taught throughout the four years of my university studies. I was able to practice first-hand what it means to work as a software engineer in a well-established company, with senior software engineers. In doing so, I gained valuable lessons and practical skills.

Having had no prior experience in the software industry, I am quite satisfied with the outcome of my work. Every week I was able to learn at least one new software concept and be able to put it to practice almost immediately, which helped reinforce my lessons. Below, I will present a short summary of the key software concepts that I studied and practiced while working on my project.

Lessons In order to implement the 'Productivity per Workflow' report, I learned to write code in the PHP programming language. One of the principles of writing code is to test it. Therefore, I learned how write unit tests in PHP as well as how to run them properly, making sure that their execution was successful.

The next most important software concept I learned was to work with a relational database and perform SQL queries. In addition to this, SQL queries had to be integrated in the codebase as well, and as a result, I learned how to work with Active Record (AR) Models. However, there was no need for new AR Models to be implemented and I did not work in developing one.

Thirdly, I practiced implementing the Model-View-Controller [6] principle for the development of V1 of the report. I already possessed the knowledge of this principle, however, I learned how to properly apply it to the system at hand so that it would present an outlook consistent with the other systems in Alice. In order to implement the MVC principle, I learned how to work with *template* files, which were used to implement the View component of the principle.

Two important software engineering principles that I practiced in my workplace were *refactoring* and the *reviewing* of the merge requests. Refactoring my own code sometimes was trickier than I had initially expected it, however, it proved useful in the long run, especially when implementing new versions of the project. Having my teammates review my code always helped me gain new insight and make possible improvements to the code and its efficiency. Nonetheless, a very important lesson I learned from my colleagues was that when implementing a new feature, regardless of its magnitude, it is always crucial to think in terms of "*Do we want quick and correct results, but hard-to-maintain code?*" vs "*Do we want clean and maintainable code, but slower results* (in terms of time, not code efficiency)?" The answer to this question is to be determined by the software engineer and there are various stakes that need to be considered.

The final software engineering concepts that I learned were *JQuery* and *Ajax* from Javascript and *Handlebars* files. These software tools were used to implement a dynamic version of the report, which would allow some minor interaction with the user.

Final words Due to my main background on research, and my lack of experience in the software industry, before the start of the internship, I underestimated the value that practical work adds to learning a new concept, no matter how intricate that concept may be. I learned that when it

[6] MVC

comes to building software, trial-and-error is the best approach to problem-solving and learning. Additionally, I came to the realization that although it is possible to over-engineer a class, or a task or solution, it is almost never possible to '*over-test*' one's code.

After the end of my internship, I made the decision to continue working as a software developer in the Operations-2 team, where my main task for the time being will be the development of new features for the 'Productivity per Workflow' report. I thoroughly enjoy using software tools for problem-solving and would like to continue gaining more experience in the software industry.

Appendix

A brief summary of the weekly activities and responsibilities that I participated in during the ten weeks of my internship.

WEEK 1:

- Was introduced to the *Operations-2* team.
- Finished setting up my workspace, including creating a username and password for the workplace PC and establishing the VPN connection.
- Was introduced to the local and production development in Alice and bo2 as well as the logging in credentials.
- Was introduced to the Belsimpel database and the logging in credentials.
- Completed my first tasks: minor changes in one template file for the front-end followed by the necessary updates on translations for the back-end. As a result, I worked with `.tpl` files for the first time.
- Completed my first merge request.

[April 13th - April 17th]

WEEK 2:

- Was introduced to my second ticket: add the multi-selection option for the `PurchaseOrderSearcher`.
- As preparation, I was introduced to the concept of searchers, the `ActiveRecord` Model, `PHP` classes and *unit tests*.
- Started learning and working with PHP.
- Consequently, I created my first `.class.php` files and wrote my first unit tests.
- Completed my second merge request.

[April 20th - April 24th]

WEEK 3:

- Was introduced to my main project, the 'Productivity per Workflow' report.
- Spent time studying and understanding the concept of *action measurements*, which ones would be fetched from the database and what other retrievable information could be relevant for the project.
- Had meetings with my supervisor and product owner to decide on a V1 for the project.
- By the end of the week, V1 was established and I started working on developing it.

[April 27th - May 1st]

WEEK 4:

- Most of the week was spent studying and writing code for developing the API and the Controller for the 'Productivity per Workflow' report.

[May 4th - May 8th]

WEEK 5:

- Was advised to not use an API for the time being since V1 is a static version; View and Controller would be enough.
- Developed a searcher for action measurement objects ('ActionMeasurementSearcher'), wrote unit tests for the searcher, finished coding the Controller (the back-end logic of the report) and the View.
- By the end of the week the merge request for V1 was finalized and ready for review.
- Had a mid-evaluation meeting with my supervisor.

[May 11th - May 15th]

WEEK 6:

- Wrote my first migration for adding a new user right - the one that is necessary for accessing the 'Productivity per Workflow' report.
- Added the appropriate translations for V1.
- Started working on V2 - extending the workflows to depict information about the colleagues who perform the action measurements.

[May 18th - May 22nd]

WEEK 7:

- Finished developing V2 of the 'Productivity per Workflow' report.
- Released V2 of the project by the end of the week.

[May 25th - May 29th]

WEEK 8:

- Was informed about V3 of the 'Productivity per Workflow' report - making the page dynamic by adding a date-picker.
- Received directions from my supervisor to focus on `JQuery`, `Ajax` and `Handlebars`.
- Spent most of the week studying `Handlebars`.

[June 1st - June 5th]

WEEK 9:

- Spent half of the week studying `JQuery` and `Ajax` and started working on developing V3.

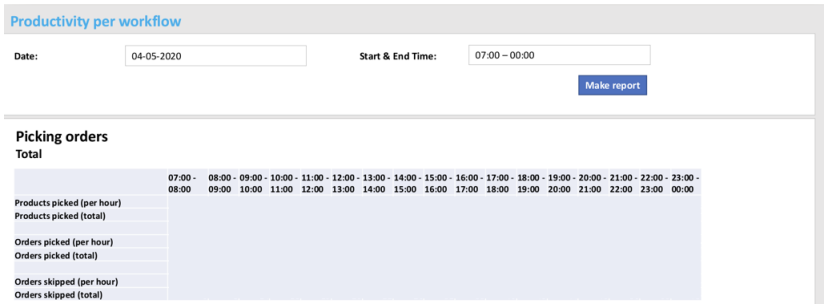
[June 8th - June 12th]

WEEK 10:

- Finished developing V3 by implementing the date-picker and showcasing the results for the 'Productivity per Workflow' report accordingly.
- By the end of the week, the merge request for V3 was finalized and the code was in production.
- Finished the week with an evaluation meeting with my supervisor and product owner.

[June 15th - June 19th]

Figures

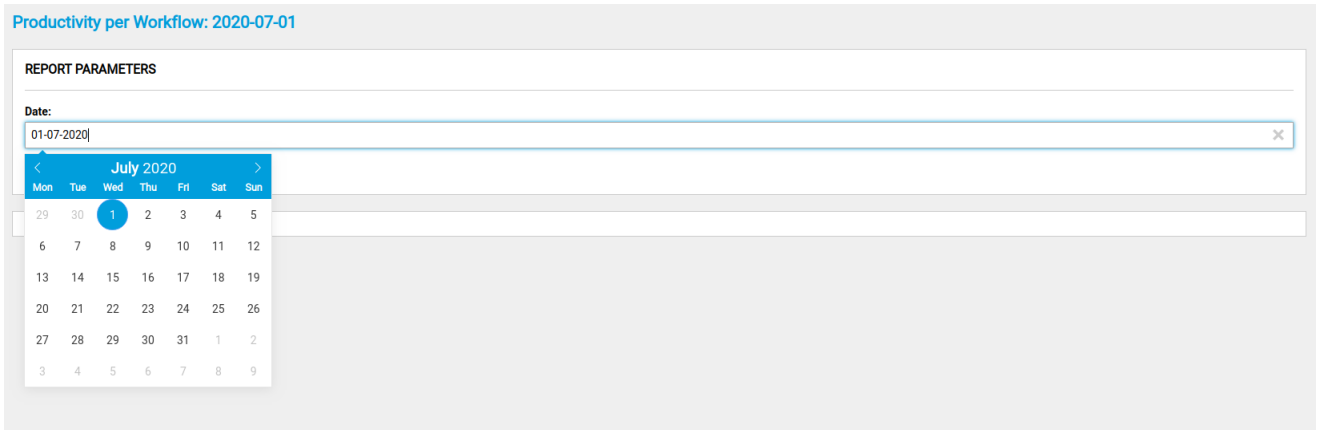


(a) This snippet is part of the mock-up that was prepared from the product owner of the project. For the sake of clarity, in this figure we only depict the workflow of *picking orders*. The workflow of packing orders is identical to this one, with the only difference being the names of the action measurements. For privacy concerns, we have excluded the values that were placed in the table cells.

PICKING ORDERS

	07:00-08:00	08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00	22:00-23:00	23:00-00:00
Products picked	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
Orders picked	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
Orders skipped	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b) This figure depicts how V1 of the 'Productivity per Workflow' report looks in production (we are only showing the workflow for picking orders). Note the three types of actions that have been presented, as well as the cumulative sum for each value in the time-slot in *Totals*.



(c) This snippet depicts how the report looks like after the release of V3. The date-picker allows the user to select any date and the action measurements results will follow.

PICKING ORDERS

	07:00-08:00	08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00	22:00-23:00	23:00-00:00
Products picked	0	0	0	3	1	0	0	0	0	0	0	0	1	0	0	0	0
Total	0	0	0	3	4	4	4	4	4	4	4	4	5	5	5	5	5
Orders picked	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
Total	0	0	0	1	2	2	2	2	2	2	2	2	3	3	3	3	3
Orders skipped	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PER COLLEAGUE

EVI XHELO

	07:00-08:00	08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00	22:00-23:00	23:00-00:00
Products picked	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Orders picked	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Orders skipped	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) This is a snippet of how the report looked like after the release of V2. Each workflow is followed by a 'Per Colleague' section.

PACKING ORDERS

	07:00-08:00	08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00	22:00-23:00	23:00-00:00
Orders packed	0	0	0	2	1	0	0	0	0	0	0	0	1	0	0	0	0
Total	0	0	0	2	3	3	3	3	3	3	3	3	4	4	4	4	4

PER COLLEAGUE

EVI XHELO

	07:00-08:00	08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00	22:00-23:00	23:00-00:00
Orders packed	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Total	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

(b) This is a snippet of V3 of the report, where we present only the packing workflow.