CSE 2000 Software Project - Project Plan

Group 11A

Lucian Negru
Radu Constantinescu
Kendra Sartori
Ahmed Ibrahim
Manar Al-Robayi

TA: Oskar Lorek
Coach: Thomas Overklift
Client: Unetiq BV

April 2022

1 Introduction

Businesses have a growing problem with their online shops as globalisation advances and an increasing amount of orders appear worldwide [1]. This problem is in the form of unnecessary manual labour from the employees in charge of processing and shipping the order, and is especially noticeable in the field of manufacturing. Clients have to search through a large catalogue of similarly priced and small products, often ordered in vast quantities. This can be very time-consuming for the client if they manually have to select the products, or for the shop employees, if they are given a large shopping list.

The team at Unetiq BV [1] has devised a solution, namely a software which can allow the client to simply attach a list of desired products and an algorithm will process them and add them automatically to the basket. The Automated Order Processing System will be very flexible by allowing ordered and un-ordered data alike, as well as support for Excel and PDF files. This algorithm will be in the form of an application and be implemented for the Furning [2] online shop first, it will also be easily integrated into other online shops in the future. The Automated Order Processing System should be operational by June of 2022.

In this project plan, we aim to tackle different aspects of the project, such as the problem, the requirements, the development process, etc. This plan will help guide us through the various stages of implementation and keep us on track.

---

[1]https://www.unetiq.com/
[2]https://furning.com

2 Problem Analysis

In this section, we will be describing what the project is about, what research we have to make, how the project may be used, and what our goals are as a team.

2.1 Problem Statement

The problem we are assigned to solve consists of these four parts:

1. **How to structure the data from Excel files**
   We are required by the client to make an Excel file processing algorithm for the orders of the system. We are given a dataset of an online shop as well as samples of orders from those shops in the form of excel documents. The challenge with this data is that it is unstructured and may have missing information. We have to decide what type of database to use to store the dataset and query from.

2. **How to create an algorithm that processes data from PDF files**
   The system should also support orders in the form of PDF documents. The PDF documents are very unstructured and may have completely different formats from one another. Therefore, we have to decide on a solution that can adapt to any of the formats they may have and accurately query from the database.

3. **Choose what type of interface will be in the front end of the online shop**
   The client has also requested a working front end. The decision we have to make is between a browser extension or a react application that can be integrated into the online shops. We have to document ourselves with the benefits and drawbacks of both approaches

4. **Satisfying the Stakeholders**
   As mentioned in the requirement engineering section, the project has multiple stakeholders. One of our biggest priorities is that we satisfy all of them.

2.2 Project Goals

The Automated Order Processing System is a project already started, in part, by Unetiq. Our goal is to implement new functionalities to the project in such a way that they can be integrated into the existing system and be easily expanded upon. Since we start the functionalities from scratch, we aim to be efficient with our time and take great consideration in our choices as they will affect the continuation of our's as well as Unetiq's work. Our project goals are as follows:

1. **Develop the System which processes orders**
   Processing of orders from Excel documents must be implemented and functional by the end of the course. Processing from PDF documents is the next step of the project and is not expected of us to finish it by the end of the course, however, what we do should be well documented so that developers can work on it in the future.

2. **Scalable implementation**
   Everything we implement should be well documented and all research made must be readily available for future developers to continue to expand the project. We aim for our features to be easily implemented in the project as a whole by the end of the course.

2.3 Market Survey

In this section, we evaluate if this can be a viable and useful product by determining the factors and context in which it can be successfully used. We can ask ourselves questions such as: Is there a demand for such a product in the market? Would this product have a meaningful impact? Is there proof that such a product would benefit the targeted stakeholders? This section will be backed up by empirical research. To accurately retrieve data on this issue, we can look at the companies that implement a similar product and have conducted and published market research on this.

One of them is Conexiom [3] which states that almost half of purchased orders are received as emailed documents and must be manually entered into a system, often this being done by an employee. In a study that Conexiom has conducted [4], they say that annually in the US, 17 trillion dollars are spent in manufacturing and distribution, out of which 50% of it is manually processed. In this study, they also point out additional interesting factors which say that per day a CSR spends on average 2-3 hours on re-entering mistyped orders and that another side effect of the manual transaction is that it is unable to scale when the business is growing.

From another perspective, if we look at another competitor, Palette Software [5], and look at a customer case study they have done on another sector, namely transportation, we can immediately see the huge impact that their product had from the customer's perspective. Emily Grantham, AP Supervisor at Landstar System inc., says "Before we deployed PaletteInvoice, our invoicing system was labor-intensive and time-consuming for staff. Automated processing has significantly increased our overall efficiency, shortened payment cycles, and helped to improve the manageability of transactions" [6]. This means that customers do see immediate benefits from automation and that such products will have a positive impact on any organisation.

Moreover, we can see that automation in any part of a process can only provide benefits overall, most of them can be seen when talking about capital, by reducing costs, and time and increasing customer satisfaction by having a more efficient processing system. According to a 2016 Accenture report [7], 52% of the surveyed companies had changed their providers in that year due to poor customer satisfaction. Another big company, Capgemini, tells us that 63% of the organisations managed to decrease this issue of customer satisfaction by integrating intelligent automation [2]. This means that automation could be the first step of implementation for any organisation that wishes to tackle this issue.

---

[3] https://conexiom.com

[4] https://conexiom.com/wp-content/uploads/2021/01/$CXM_{Infographic_{StartingtheBusinessCaseforAutomation-3.pdf}}$

[5] https://www.palettesoftware.com

[6] https://www.palettesoftware.com/wp-content/uploads/2021/01/Palette-Customer-Case-$Landstar_{N}Y.pdf$

[7] https://www.accenture.com/us-en

2.4   Research on Existing Solution

The Automated Order Processing System is an eCommerce Automation product. To be able to compare our project to existing products in this category we first have to define what the category is and then look at existing solutions.

2.4.1   eCommerce Automation

eCommerce Automation refers to "the software that helps your online store convert most or all of the manual, repetitive tasks into self-fulfilling, automated tasks." [8] This means that employees can shift their focus to other areas of the business and consumers can rely on a more efficient method of shopping.

Most automation can be described by a Trigger, Condition, Action workflow. Triggers are events initiated by the client directly, in the case of our system it would be the client uploading a file to the online shop interface. Then various conditions can be checked to determine the following action. In our case, the conditions would be the type of processing that should be used depending on the file type, whether the items on the list of orders are available, etc. Lastly, there is the Action stage where the automation process determines what should happen and makes it happen. In the case of our System, that would be the desired items being added to the basket.

eCommerce is one of the world's fastest-growing sectors, shown in Appendix 8.1, with a consensus that it helps businesses thrive and improves customer experience [3]. Therefore, we find it important to compare those factors between our project and existing solutions.

2.4.2   Conexiom

Similarities: The conexiom software [9] offers processing of orders through the use of machine learning to save time. It receives different file types of unstructured data and runs them through a processing algorithm to label, structure, and identify the content. Then it forwards the extracted data to the system for various tasks.

Shortfalls: The software doesn't allow for files to be sent through the user interface, only via emails. Our system aims to improve that by allowing customers to add the files to the online shop directly through the interface and keep things simpler.

2.4.3   Tipalti Approve

Similarities: Tipalti Approve [10] is a platform that allows for PO (purchase order) generation by having users upload invoices in multiple formats and levels of structure, then processing the information and display it back to the user. The returned data can then be ordered or further analysed.

Shortfalls: The user has to upload the invoices to the app, which then sends an order. Our implementation will streamline this by allowing the users to upload directly to any website using the system. Therefore, the platform has to support the different online shops and introduces a middle-man in the ordering process.

---

[8]https://www.bigcommerce.com/ecommerce-answers/what-is-ecommerce-automation/
[9]https://conexiom.com/sales/
[10]https://www.approve.com/po-management-lp/

2.4.4 Palette Automated PO Software

Similarities: The Automated Po Software [11] is software that can be implemented into a business either stand-alone or in an ERP (Enterprise Resource Planning) system. The system receives the orders either through emails or scans and matches them to the recipient store. If a match is not met then it is forwarded through to a manual pipeline where an employee will handle it. The matches are then forwarded to the ERP where the data can be used for various actions, including ordering.

Shortfalls: The POs received have to be rather structured and the software does not make much use of machine learning for more complex and unstructured files. The processing consists of parsing rather than adapting, forcing the users to conform to a certain structure of invoice and compromising on user experience.

2.4.5 Research Summary

In conclusion, there exists products that implement similar functionality to the one proposed by our client. There are even products, such as Palette Automated PO Software, which contains most of the requirements. However, we have not found any to completely satisfy all of the requirements, this leads us to conclude that implementation of the Automated Order Processing System is necessary.

2.5 Use Case Analysis

For this analysis we decided to look over the use cases of the product for our client, this will allow us to give a technical which takes advantage of the company's background. Two types of users can take advantage of the product, the online shop customer and the purchase order reviewer.

The online shop customer

- Customer wants to order a large list of products

- Customer wants to order an unordered list of products

- Customer wants to efficiently manage and order the set of wished products.

- Customer wants to order a list of products with (possibly) wrong data

- Customer wants to order a list of products with (possibly) empty data

- Customer wants to receive an ordered list from the unordered list of products

- Customer wants to upload the invoice of orders directly to the online shop

- Customer wants to upload multiple PDF formats to the online shop

The online shop employee

- The employee wants to receive an ordered list from the customer

- The employee wants to review only a subset of orders, the rest are automatic

---

[11]https://www.palettesoftware.com/resources/purchase-order-automation/toggle-id-5-closed

3 Requirements Engineering

To fully understand the goal of the project and the wishes of our client we will use 'Requirement Engineering'. Requirement Engineering is the process used to recognise and express the software specifications that allow us to solve application problems [4]. It makes it possible to translate the client's needs and identify incomplete or inaccurate specifications that potential users may have. The following subsections discuss the various methods used to determine these specifications.

3.1 Stakeholders

To identify all the requirements for our project, we first need to know who the stakeholders are. Stakeholders are the parties that are affected by a development project [5]. This project involves three different parties, each with different requirements for the application. The following subsections discuss each stakeholder and their goals.

3.1.1 Unetiq

Unetiq are our main stakeholders and provide us with the necessary information to complete the project. Unetiq is a company that develops and customises AI software for various other companies to accommodate the automation of manual processes [6]. Their goal is to create a web application that allows customers of online shops to upload their orders through unstructured Excel and PDF files. The orders should be processed in such a way that no human interaction is needed to complete an order. The application should scan the uploaded file and extract a structured order that can then be processed further.

3.1.2 Customers

There are two types of customers that we have identified as stakeholders. The first ones are the customers of Unetiq, they will be discussed in the next subsection. The second type of customers is the customers of online shops. The goal of these customers is to be able to place an order by uploading an Excel/PDF file, instead of going through online shopping and adding each product one by one into a virtual basket.

3.1.3 Online Shops

Online shops are the customers of Unetiq. In this report, we use the term 'online shops' as a term for identifying online shops that want to automate the process of handling orders through Excel and PDF files. The goal of this stakeholder is to allow their customers to upload Excel/PDF files to place their orders. This can be realised by integrating an option to upload Excel/PDF orders on their online shop.

3.2   Requirements Elicitation

There are several types and techniques for elicitation that allow developers to identify all requirements. The two types of elicitation techniques are:

- Direct Approach

- Indirect Approach

Indirect approaches are used to obtain information that could be challenging to extract and articulate. Direct approaches focus on understanding the problem that the client faces and the way they think this problem can be solved [7].

To collect the requirements for the project we have used a direct approach, namely the 'interview' elicitation technique, by meeting with Unetiq. These meetings have also provided us with a clear understanding of the requirements of the other stakeholders.

The interview was unstructured. The questions were formulated on spot during the interview to allow as much flexibility as possible.

The answers to these questions gave us a clear and adequate understanding of the requirements that had to be formed. The requirements were then split into two groups; functional and non-functional requirements. Functional requirements focus on the ability of the system to perform certain tasks, while non-functional requirements focus on the behavioural aspect of the system [8].

The list of all requirements can be found in Appendix 8.2.

3.3   MoSCoW Method

The requirements in Appendix 8.2 are divided into four categories. The categorisation is done using the MoSCoW technique. This categorisation allows the team to work sustainably and efficiently.

One of the key factors of MoSCoW categorisation is that it defines the priorities for each requirement [9]. The requirements are divided into four sections:

- **Must Haves:** The implementation of these requirements gives a minimal viable product on top of which all other requirements are built.

- **Should Haves:** This category includes all requirements that are not vital for a system to work, but are of such importance that they need to be implemented.

- **Could Haves:** These are features that offer improvements to the product, but are not necessary for the deliverable.

- **Won't Have:** These are requirements that are seen as infeasible to implement within the development team.

4 Feasibility Study

This study has been conducted after we had several meetings with our client, in which we discussed exactly the way and setting in which the final product will be used,to properly understand and assess the practicality of the proposed project. The final analysis has been done after considering multiple factors some of which are technical, operational, and time-related.

In addition to those factors, we also looked at other factors that will guarantee this project can be of use to multiple clients, and it can serve as a solution to the recurring problem that a lot of companies have as specified in the **Market Survey**

4.1 Technical Feasibility

In terms of technical feasibility, our team looked at the currently existing technology, and if it suffices our needs. Since this project will be built from scratch, we consider that, after thorough research, several options exist in which we can combine currently existing technologies to achieve the final product. Moreover, we consider that the team has the necessary skills in terms of technical knowledge to accomplish this project. In addition to that, the existence of products that achieve similar functionality to our project ensures that there would not be any technical barriers present.

4.2 Operational Feasibility

Regarding operational feasibility, this project matches the business objectives and goals of the company, since their team is also working on it, however with slight differences in terms of requirements. In addition to this, the integration within the organisation's other projects will be of ease since the company is already working on a similar product. Also, the company has provided us with samples of order databases, both in Excel and PDF formats, that we can make use of to test and develop the product.

4.3 Time Feasibility

Time is another constraint that this project is faced with. Given that the project is part of the course provided by the university, we and our stakeholders are aware of the limited time frame in which this product has to be delivered. With this situation, we considered that a prioritisation technique was of utmost importance. Therefore, we decided to use the MoSCoW method to set our targets straight. We consider that, despite this constraint, the team will be able to accomplish this task and in the end at least deliver the minimum viable product. All of this has been established in the 'Should have' section of the MoSCoW analysis.

4.4 Other Factors

Other factors that could be taken into consideration could be economic and legal feasibility. For the first one, we consider that given the setting in which the project is done, it falls outside of our scope. It can be easily argued that the impact of such a project would be a positive one on the company, although there is no way of telling the extent of such. Regarding the legal feasibility, the project can be done under legal requirements if all laws and regulations are met.

4.5 Feasibility Summary

In conclusion, as long as all the aforementioned factors are met, this project is feasible. As a team it is our responsibility to make sure we take everything into consideration and respect all feasibility factors in order to deliver a viable product in time.

5 Risk Analysis

It is of utmost importance to assess the risk associated with a project in terms of the different criteria that could influence the development process. Although many aspects might be considered irrelevant when looking at the bigger picture, including them in our risk analysis can ensure that we take into account issues in an anticipated manner; as to prevent a failure to deliver a highly qualitative project.

In the following subsections, we will describe the different types of problems that we might encounter throughout the following weeks and their relevance. Furthermore, we will try to find solutions that are feasible and suitable for all group members.

5.1 Schedule and Communication Capabilities

After making a thorough team analysis, including Belbin's team role management theory [12], we have gained insight into how each individual in our team might affect the way we work as a group. We have also taken into consideration the balance of roles within the group to better distribute the work accordingly.

The team members' prior experience consists of other projects which are part of the curriculum imposed by our study program. However, the Software Project we are currently part of is far more permissive in terms of time allocation and resource management, as it is the first project in which the students have a connection to a real-world client. This might pose a challenge, whereas the impression of freedom could easily affect keeping a proper schedule and making sure we are on track with our project.

On the other hand, the way that people interact has changed significantly in the past couple of years. In addition to the commodity everyone has from working at home, online meetings feel like a much more effective way to discuss further developments. This is further supported by the fact that group members live in different cities, which makes it harder to meet in person. This can however also affect productivity and morale, leaving people feeling rather lonely than part of a team.

We aim to minimize the risk of running into these problems by having frequent meetings—at least three times a week while keeping in touch daily—to make sure we are all on track with the work that needs to be done. Furthermore, we plan on meeting at least once a week to work together in person; as to create the general feeling of belonging to a team.

5.2 Possible Growth in Requirements

Requirement elicitation is the foundation on top of which our project is built. The requirements dictate all other software engineering processes which can also influence productivity, quality, and risk. Requirements engineering constitutes the foundation of the whole software development process [10]. For that reason, we must have a clear overview of what we have to do and what the different stakeholders expect from us. However, some of the requirements are probably less obvious and there is always a chance that we miss some of them, which might negatively impact our process and cause unnecessary delays.

---

[12]https://www.belbin.com/about/belbin-team-roles

In this regard, we believe it is important to have a good relationship with our client and constantly keep him updated regarding the development of the product, through the weekly meetings we have set. Furthermore, we have decided to follow the guidelines described by the agile development framework and structure the process into sprints that last one week each.

5.3   Lack of Experience with React

Our client has expressed his preference for us to make use of React for building a user interface for our final deliverable. However, none of the team members has experience with this JavaScript library, which could hinder and delay the process by a significant amount of time, equivalent to the time required for us to get acquainted with it and start making use of it.

We wish to overcome this lack of expertise by setting a clear deadline by which each team member needs to try getting acquainted with React. Additionally, we hope that a great collaboration between the members will make so that mutual help is offered between us to ensure we are all up for the challenge that creating this component poses.

5.4   Lack of Knowledge About GDPR and Other Legal Aspects

Given that we are developing a tool that will be, hopefully, widely used by different categories of stakeholders, it is our job to ensure that it is easy to use and does not have any negative impact on these stakeholders. In other words, it is our job to protect the users of our tool from possible harm that it might do, especially in terms of privacy and security. In this regard, we need to make sure that it is compliant with the guidelines described by the GDPR directive as well as other laws and regulations that we might not yet be aware of.

We hereby wish to carry out further research on the possible dangers that a user could be faced with and get more informed on these matters. Furthermore, we will ask our client for further information about things that we need to take into consideration regarding the legal aspects of our product.

6 Development Methodology

In this section, we will discuss the architecture and technologies that we decided to use to deliver the final product. Note that the chosen technologies can change throughout the life cycle of the project if the team decides that certain paths are preferable instead of others.

6.1 Front-End

In terms of the front-end, we decided to use React since it is one of the most popular frameworks and this can be useful for further implementations on this project. On top of this, using React was one of the non-functional requirements that the client provided us.

6.2 Back-End

For the back-end we agreed on using python as the main programming language and connecting it with a framework which will either be Flask or Django, which will be decided upon during the implementation stage. The choice of choosing python is one connected to the non-functional requirements that we were provided by the client.

6.3 Database

In terms of databases, we decided to use a server-less option, because of the small scale of the product and also the additional benefits that it provides; namely reduced cost, good resilience, scale potential, and little to no server management. The database we chose was DynamoDB, a non-relational database that is optimised for intensive read workloads provided by Amazon.

6.4 Processing

The processing will be done via an algorithm that will be created during this life cycle. The first iteration of the algorithm will focus on finding proper heuristics to correctly identify the unstructured data labels. If this solution reaches a good accuracy of 70% we will focus on further optimizations. In the case it does not, the team will explore other processing options, using natural language processing models. Because of the time constraint provided by the project, we will most likely make use of pre-trained models, like the ones from ModelZoo [13], that have a lot of documentation and examples that we can make use of.

6.5 Testing

For testing, we will make sure that each piece of functionality has to be tested thoroughly before it can be reused. If we assess this constraint at the beginning, we will make sure that through the development cycle bugs or errors can be caught more easily.

In addition to this, to test the overall functionality of the application, we will make use of the dummy orders that the company provided us; which already have the potentially correct labelled data. We will make use of this to reach the 70% accuracy that we agreed with the client upon.

---

[13]https://modelzoo.co/

6.6 Low Fidelity Prototype

A low fidelity prototype is helpful in the early stages of development, offering a visualisation of alternative design solutions, which provokes innovation and improvement, alongside offering users a more comfortable way of making suggestions [11].

To have an overview of what our implementation will look like, we created a low fidelity prototype. This prototype will serve as a boilerplate on which we can brainstorm with our client to achieve their desired implementation. The prototype can be found in Appendix 8.3.

7 General Planning

In this section, we will be discussing the plan for some general aspects of the project. This will cover the implementation methodology, communication and scheduling.

7.1 Scrum Methodology

In this project the Scrum methodology is used to track the process of the project. It is an agile methodology which ensures that the quality of the product is high. This quality is ensured as testing is an important part of an agile methodology. Another reason for this is that an agile methodology is both iterative and incremental. Being interactive means that in each iteration the work will be improved using the feedback from the client. It is incremental as the parts of the projects that are finished are provided to the client during the project [12]. Scrum is based on sprints which are small periods of time that are used to reflect on the work and to receive feedback. Reflecting is useful to improve certain parts of the project and the feedback will after receiving it be processed [13].

The team decided together with the client that the sprints would be weekly from Monday to Sunday. On Monday the progress of the previous sprint will be discussed and on Tuesday the next sprint will be discussed. This makes it possible to reflect on the progress of the previous week and to create a sprint for the next week.

7.2 Communication

There are different forms of communication within the project. These forms of communication are communication within the team, communication with the client, communication with the teaching assistant and communication with the coach.

7.2.1 Communication Within the Team

The team makes use of WhatsApp, Discord and Zoom for online communication. The team meets at the Delft University of Technology or the company in Amsterdam for physical communication. The purpose of this is to work together on the project and to discuss the progress of the project.

7.2.2 Communication with the Client

The team communicates with the client through Zoom meetings and Slack. The team meets with the client once every two weeks at the company in Amsterdam for physical communication. The purpose of this communication is to discuss the requirements, questions about the project and the progress of the project.

7.2.3 Communication with the teaching assistant

The team communicates with the teaching assistant through Mattermost and weekly meetings at the Delft University of Technology. The purpose of this is to discuss the progress of the project.

7.2.4 Communication with the Coach

The team communicates with the coach through weekly meetings at the Delft University of Technology. The purpose of this is to discuss the requirements, questions about the project and the progress of the project.

7.3 Schedule

We have created a broad schedule for each week, including all the meetings we have with the client, coach and TA. This schedule can be found in Appendix 8.4.

References

[1] G. Partners, "Benefits and Challenges of Globalization," Feb. 2020. [Online]. Available: https://www.globalization-partners.com/blog/benefits-and-challenges-of-globalization

[2] Capgemini. (2018, Jul.) Consumers are embracing AI and will reward organizations that offer more human-like AI experiences. [Online]. Available: https://www.capgemini.com/us-en/news/consumers-are-embracing-ai-and-will-reward-organizations-that-offer-more-human-like-ai-experiences/

[3] A. Maier. (2020, May) Expert insights into Customer Experience. [Online]. Available: https://codecoda.com/en/blog/entry/expert-insights-into-customer-experience

[4] Z. Jin, "Chapter 3 - importance of interactive environment," in *Environment Modeling-Based Requirements Engineering for Software Intensive Systems*, Z. Jin, Ed. Oxford: Morgan Kaufmann, 2018, pp. 29–39. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128019542000030

[5] Concepta. (2018, Aug.) How to Define Stakeholders for Your Software Development Project. [Online]. Available: https://www.conceptatech.com/blog/how-to-define-stakeholders-for-your-software-development-project

[6] Unetiq. (2022) Unetiq - Your Agency for AI and Data Science Services. [Online]. Available: https://www.unetiq.com

[7] S. Khan, A. B. Dulloo, and M. Verma, *International Journal of Information and Computation Technology*. International Research Publications House, 2014.

[8] M. Glinz, "On non-functional requirements," in *15th IEEE International Requirements Engineering Conference (RE 2007)*, 2007, pp. 21–26.

[9] Komodo. (2022) How to Use MoSCoW Prioritisation to Help Deliver Better Software Products. [Online]. Available: https://www.komododigital.co.uk/insights/how-to-use-the-moscow-prioritisation-technique-to-execute-software-projects

[10] N. T. More, B. S. Sapre, and P. M. Chawan, "An insight into the importance of requirements engineering," *International Journal of Computer and Communication Technology*, vol. 8, no. 1, pp. 29–31, Jan. 2017.

[11] A. Affairs, "Prototyping | usability.gov," 2022. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/prototyping.html

[12] M. Cohn. (2014, Nov.) Agile needs to be both iterative and incremental. [Online]. Available: https://www.mountaingoatsoftware.com/blog/agile-needs-to-be-both-iterative-and-incrementalauthor

[13] (2022) What Is Scrum Methodology? Scrum Project Management. [Online]. Available: https://www.digite.com/agile/scrum-methodology

8 Appendix

8.1 Graph displaying the growth of the eCommerce market (note: * signifies prediction)
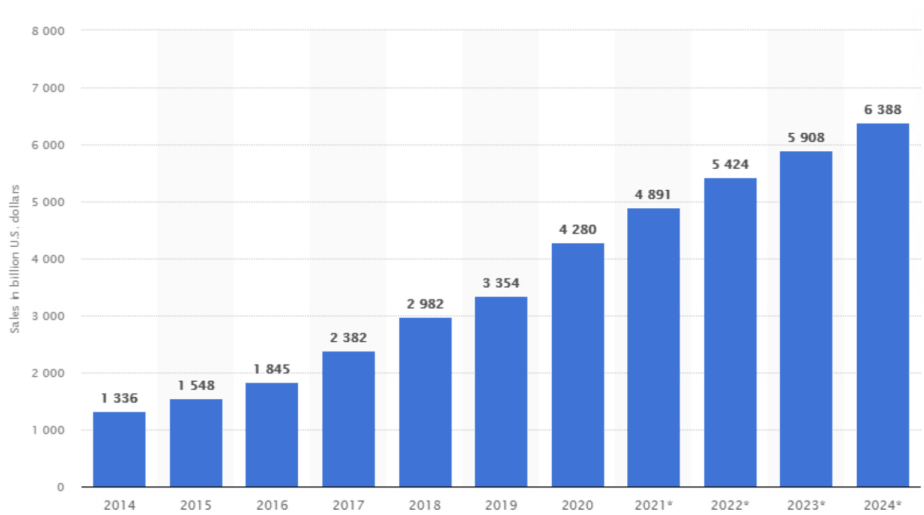
Source: statista.com

8.2 Requirements List

8.2.1 Non-Functional Requirements:

- The system will be scalable to the extent that it can also process orders of other file-types.

- The system will make use of a server-less database.

- The system will process at least 70% of the excel orders in a correct way.

- The system will run on a web application with React Framework.

- The system will have a Python back-end.

8.2.2 Must Haves:

- As a customer I want to be able to search for a product on a search bar.

- As a customer I want to get a list of products that match my search keys.

- As a customer I want to be able to upload an excel file with an order.

- As a customer I want to get an 'automatically created structured order' after uploading my order in an Excel file.

8.2.3 Should Haves:

- As a customer I want to be able to download a structured CSV file with the automatically created order from the Excel file that I have uploaded.

- As a customer I want the system to notify me if my order contains an unavailable product.

- As a customer I want the system to notify me if my order contains a product that does not exist.

- As a customer I want the system to notify me in case my order can not be processed.

- As a customer I want to be notified if an actual price differs from my expected price.

8.2.4 Could Haves:

- As a customer I want to be able to manually add products to the automatically created order.

- As a customer I want to be able to change an automatically created order.

- As a customer I want to be able to delete items in an automatically created order.

- As a customer I want to be able to let the system know whether my order was processed correctly

- As a customer I want to be able to upload a PDF file with an order.

- As a customer I want to be able to upload multiple files at the same time.
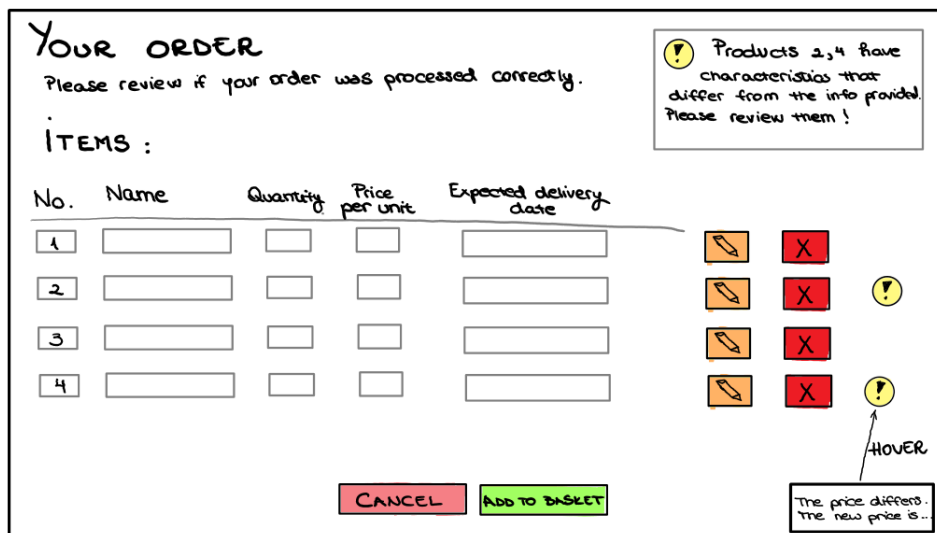
8.2.5 Won't Haves:

There are no requirements that we will certainly not do.

8.3 Low-Fidelity Prototype

8.3.1 Main Search Interface
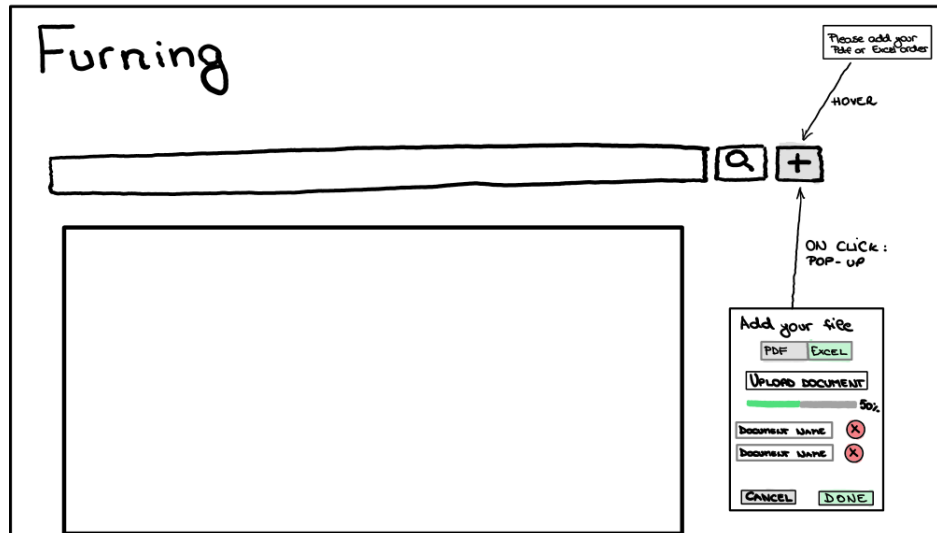
8.3.2 Returned List Interface

8.4 Schedule

Week	Meetings				Assignments				Project progress
	TEAM	UNETIQ	COACH	TA					
1	3	2	0		1	Assignment 1			Meeting with UNETIQ
2	3	2	1		1	Assignment 2&3			Setting up requirements
3	3	2	1		1	Assignment 4&5			Dividing tasks and start coding
4	3	2	1		1	Assignment 6 & Peer review 1			Work on coding
5	3	2	1		1	Assignment 7&8			Work on coding
6	3	2	1		1	X			Work on coding
7	3	2	1		1	Assignment 9			Finishing the code
8	3	2	1		1	Assignment 10			Work towards end of the project
9	3	2	1		1	Assignment 11 & Peer review 2	Finish the project