Automated Invoice Manager for the E-Commerce HP Department

Alexandre Galvany Pardo

Resum— L'objectiu d'aquest treball és eliminar les ineficiències de l'actual sistema de gestió manual de factures del departament de comerç electrònic d'HP automatitzant i centralitzant tota la captura i gestió de factures en una única implementació end-to-end. Per desenvolupar el projecte s'ha utilitzat la metodologia Scrum of One junt amb el Software de Jira. L'eina d'extracció de dades de correu electrònic desenvolupada permet que les factures rebudes per correu electrònic i descarregades en una carpeta compartida siguin processades per un analitzador que extreu totes les dades rellevants requerides pels gestors de projectes. Un cop extretes les dades i emmagatzemades a la base de dades, els responsables del projecte poden visualitzar, editar o eliminar les dades de les diferents factures a través d'una pàgina web. Aquest lloc web, desenvolupat des de zero, ofereix la possibilitat de visualitzar les factures registrades així com afegir nous registres de factures manualment, exportar les dades en diferents formats, crear previsions pressupostàries pròpies mitjançant la definició de PO's[1] o fins i tot fer un seguiment de com evoluciona el pressupost real respecte a la previsió. Aquesta cadena fa que el procés de gestió de factures sigui més eficient i menys feixuc.

Paraules clau—Automatització, Centralització, E-Commerce, End-to-End, Factures, HP, Parser.

Abstract— The objective of this work is to eliminate the inefficiencies of the current manual invoice management system of the e-commerce department of HP by automating and centralizing all invoice capture and management in a single end-to-end implementation. To develop the project, the Scrum of One methodology has been used together with the Jira Software. The e-mail data extraction tool developed allows invoices received by e-mail and downloaded into a shared folder to be processed by a parser that extracts all relevant data required by project managers. Once the data is extracted and stored in the database, the project managers can view, edit or delete the data of the different invoices via a website. This website, developed from scratch, offers the possibility to view the registered invoices as well as manually add new invoice registers, export the data in different formats, create own budget forecasts by defining PO's or even track how the actual budget evolves compared to the forecast. This chain makes the invoice management process more efficient and less cumbersome.

Index Terms—Automation, Centralizing, E-Commerce, End-to-End, Invoices, Parser.

1 Introduction - Work Context

Currently, the e-commerce department of HP faces several challenges in managing invoices related to its projects. The existing system is based on a manual process of receiving invoices by email from outsourced service providers and extracting the necessary data for approval and registration. This cumbersome and error-prone process has proven to be inefficient and impractical.

All project managers engaged in extracting invoice information from emails spend 2 to 4 minutes each. This time could be better used to perform a more thorough analysis of the invoices instead of extracting the data manually, a task that can be easily automated.

In addition, the way of working with the data is based on an excel shared between the different project managers. Although excel is a tool with many advantages, it is still quite limited to perform a proper analysis of the invoices.

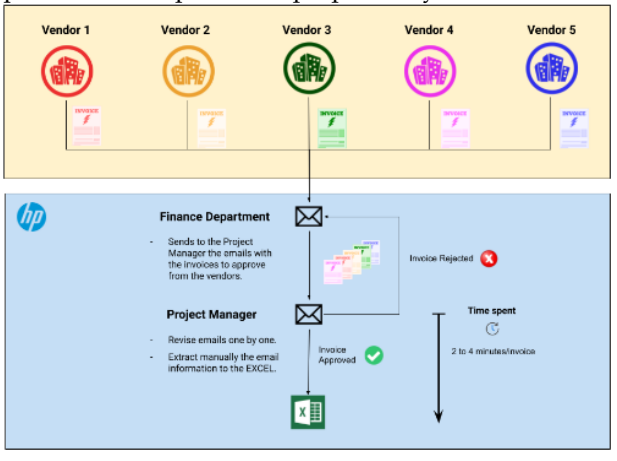

Diagram 1. *Actual Problem of the Invoice Tracking System*

1.1 Solution Proposed

To overcome these difficulties, a solution was proposed to automate the invoice management system. The core idea is to develop an end-to-end application specifically for the e-commerce department of HP. This application will address

- *E-mail de contacte: alexgalvany14@gmail.com*
- *Menció realitzada: Enginyeria del Software*
- *Treball tutoritzat per: Joan Protasio Ramírez (Departament de Ciències de la Computació)*
- *Curs 2022/23*

[1] Purchase Order.

several important aspects, such as extracting data from invoices received via email using an email parser.
This solution is designed to improve the overall efficiency of the process of managing and recording their invoices. Project managers no longer will spend time and effort manually entering invoice data into an Excel spreadsheet. Instead, the new system automatically forwards the emails to a shared folder, where the email parser extracts the data and makes it available to project managers for approval.

One of the most important aspects of the solution is the creation of a web interface that allows project managers to easily and intuitively view, filter, approve and manage invoices, as well as manually add new registers as needed.

The solution will also include a system to export invoices in different formats such as PDF or Excel to improve interoperability with other systems.

Another highlight is the ability for project managers to create budgets and compare them to actual costs through the same web interface. This enables better planning and monitoring of projects and helps to avoid variances and make more informed decisions.

To ensure the scalability and adaptability of the system, the application is developed with an architecture that allows the incorporation of future functionalities and improvements. This ensures that the application can evolve with the changes and needs of HP's e-commerce department without requiring drastic changes or migrations to other solutions.

In summary, the proposal to automate the invoice management system for HP's e-commerce project managers aims to improve efficiency, reduce errors and provide a completer and more user-friendly platform for managing and capturing invoices. By using an email parser, a relational database, an intuitive web interface and features such as budget forecasting and comparison with actual costs, the overall process should be significantly improved.

1.2 Objectives

To comply with the proposed solution, 7 main objectives have been created, all with the same level of prioritization.

- Develop an end-to-end application to improve the management of HP e-commerce invoices.
- Implement an email parser to extract invoice data from emails.
- Design a database to store invoice data and manage it.
- Create a website to visualize, filter and manage the invoices.
- Implement a system to export invoices in different formats.
- Allow project managers to create forecast budgets and compare them with the actual cost.
- Ensure the scalability of the application for possible improvements and future functionalities.

1.3 What's to come

In the following points of the article, you will find the state of the art of the project and similar, the methods used to successfully carry out the project, the architecture of the solution, a summary of the development of the work carried out to achieve all the previously mentioned objectives, the results obtained, and the conclusions drawn.

2 State of the Art

A research was conducted for possible frameworks already developed for the automatic management of invoices and email extracting data to determine if there are already solutions on the market that could meet this need. Here the existing solutions found:

- **Excel**: Excel is a tool that can be used perfectly for invoice management. There are also many templates to carry out this invoice management **[1]**. Even so, one of the reasons why this project is carried out is because project managers want to add functionality to their liking and although Excel is very versatile with macros it is also quite limiting in certain extra features that can only be developed from scratch.

- **Email Parser [2]**: this framework could solve the part of data extraction directly from the emails, but it has obvious problem as it does not provide a sophisticate interface to analyse the data other than Excel.

- **Invoice Management Software SAP[2] [3]:** this option allows the user to have a better interface to process and analyse the data, but it does not integrate the system to read and extract the information of the invoices from the emails.

Although these frameworks can work if they are combined, for example the Email Parser with the Excel or the Invoice Management Software of SAP, there is a major drawback that cannot be solved: **adapting to the inflexible privacy policy that HP works with**.

The company requires that all sensitive data, including invoices or PO's, are stored on company servers and not on third-party servers.

Therefore, the best option is to develop an end-to-end application from scratch, customized to the company's nedds and inside the envoirment and servers of HP.

Even so, to develop the tool some fuctionalities from the the email reader "Email Parser", as well as for the web in the "Invoice Management Software of SAP" are

[2] System Applications and Products

incorporated.

3 METHODOLOGIES USED

3.1 Scrum of One

The methodology used is Scrum of One **[4]** instead of Kanban **[5]** for the project management. Some of the reasons to choose Scrum are:

- **Focus on delivering results**: Scrum puts a strong emphasis on delivering results, which is critical in this project because there are multiple stakeholders who want to see the results of the progress.
- **Splitting work into sprints**: Scrum allows work to be split into 1-4 week sprints (in the case of Scrum of One), which helps to better manage the tasks being developed.
- **Previous experience**: worked with Scrum several times before and feel very comfortable with this methodology. Moreover, it is widely used in the company's teams, which makes it easier to collaborate and adapt to the company's way of work.
- **Flexibility in the face of change**: both Scrum and Kanban are agile methodologies and adapt well to projects with variable requirements.

In this sense, Scrum provides good flexibility to deal with changes that may occur during the project.

In summary, the decision to use Scrum of One is based on its focus on results, the division of work into sprints, previous experience, and flexibility to change, which are important factors for an effective project management.

3.2 Jira Software

The tool used for tracking the project is Jira Software **[6]**.

This tool is very well optimized for working with the Scrum methodology. It allows to create the different sprints, user stories and epics for the project **[7]**.

The previous usage of this tool is another reason why Jira is used as it proved to be highly effective. Furthermore, it is already used within the company to manage various projects.

3.3 GitHub

To have code control, a GitHub repository is used to keep track of the different developments in branches and thus have the code controlled and centralized.

Also, GitHub allows synchronization with the Jira Software, which allows the project manager himself to follow the progress of the development.

4 PLANIFICATION

To allow a good planning of the project, the requirements were divided according to the methodology of Scrum epics and user stories. This methodology allows to define the system requirements from the point of view of the different users that will use the application.
In this case, the users defined are the project managers, who have full permission to use the application, and the finance stakeholder, who only has permission to view the web.

Also, the grouping of these user stories is done using Epics, which are working blocks of related user stories.

Finally, to plan well the time needed to complete each task, a certain number of Story Points **[8]** were assigned. These Story Points are a type of work measurement that exists in Scrum to determine how many hours of work each functionality involves.
For this project, 1 SP[3] is equal to 3 working hours. (Table 1.)

Epics	User Stories	Forecasted Story Points
Email Parsing and Data Extraction	4	17 SP
Invoice Management	7	40 SP
Sorting and Filtering Invoices	3	15 SP
Expense Tracking and Reporting	7	20 SP
Purchase Orders Management	6	33 SP
Sorting and Filtering POs	3	15 SP
Web Access Permissions	2	6 SP
TOTALS	32 US	146 SP

Table 1 . Forecast in Story Point of the Epics

The total amount of work forecasted is **438 hours of work** as there are many of the user stories that are not a mandatory.

For the must have user stories, the total amount of time is **294 hours of work**.

4.1 Sprints

The amount of work was divided into different sprints lasting between 1 week and 4 weeks starting the 20th of February and ending the 18th of June (*Diagram 2*).

[3] Story Point.

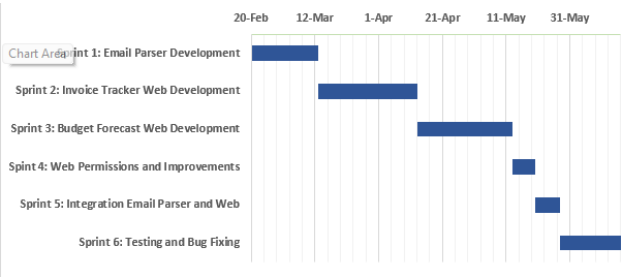

Diagram 2 . *Sprints Gannt Diagram*

These are the epics that have been worked on in each of the sprints:

Sprint	Epics
1. Email Parser Development	Email Parsing and Data Extraction
2. Invoice Tracker Web Development	Invoice Management
	Sorting and Filtering Invoices
3. Budget Forecast Web Development	Purchase Orders Management
	Sorting and Filtering POs
4. Web Permissions and Improvements	Invoice Management
	Purchase Orders Management
	Web Access Permissions
5. Integration Email Parser and Web	Email Parsing and Data Extraction
	Invoice Management
6. Testing and Bug Fixing	Email Parsing and Data Extraction
	Invoice Management
	Purchase Orders Management
	Web Access Permissions

Table 2 . *Sprints and Epics inside of them.*

5 SYSTEM ARCHITECTURE

The system architecture is based on four main blocks (*Diagram 3*):

1. **The Email Parser**: its function is to read the invoice data coming from the emails and store them in the database.
2. **Relational Database**: responsible for storing all data related to invoices and purchase orders (POs).
3. **Web Back End**: responsible for performing the processing of invoice and POs data, as well as managing the API that will connect to the front end.
4. **Web Front End**: provides an intuitive interface to the user, displaying all invoices and POs, along with the required functionalities.

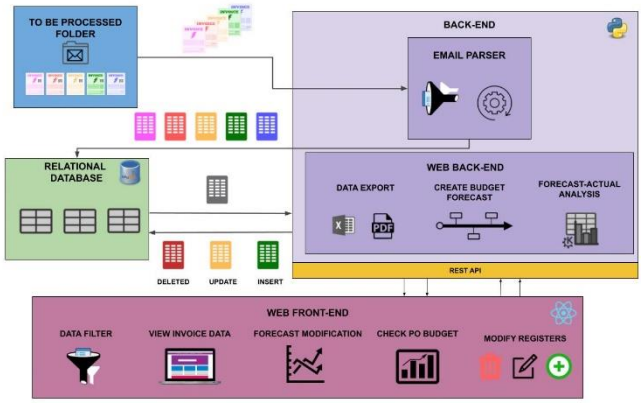

Diagram 3 . *General Architecture of the Automated Invoice Tracker System*

6 EMAIL PARSER

6.1 Data Extracion from Emails

To extract the information from the e-mails containing the invoices, the system developed works when it detects that a new file of type ".msg" has been added to the shared folder, the whole data extraction system starts to analyze and extract the necessary data from the e-mail body.

In this case, the data extracted from the email is:

- Invoice Number.
- PO Number.
- Gross Amount.
- Received Date.
- Invoice Date.

To extract this data from the body of the message, search patterns are used in the email to find the specified information, since emails always follow the same template (*Fgure 1*).

Invoice number	Gross Amount(Doc cur)
Invoice date (MM/DD/YYYY)	Due date (MM/DD/YYYY)
Vendor name	SAP Document number
Co code	MM Doc
Approval Type	

Item No	Purchase Order No	GL Account No	Cost Center	Quantity	Currency	Unit Price	Amount	Requester

Figure 1. *Invoice Email Template*

Once the data is correctly extracted from the email, it is uploaded to the database, along with any data required to store an invoice, such as the PO information or the quarters of the extracted dates.

The approval status that is stored in the database depends on the gross amount. If the gross amount is less or equal than $10,000, the approval status is "Approved" and " Auto Approved" is set to "Yes". If the gross amount is higher, the approval status is "Pending" and "Auto Approved" is set to "No".

Once the data is saved in the database, the e-mail is placed in another folder that contains all emails that have been

previously processed according to the fiscal year to which they belong (*Diagram 4*).

If a problem occurs during invoice extraction, the data is not saved in the database, the e-mail is sent to a folder containing the emails that could not be processed correctly, and the user is warned by a pop-up message that the e-mail could not be processed correctly and must be processed manually.

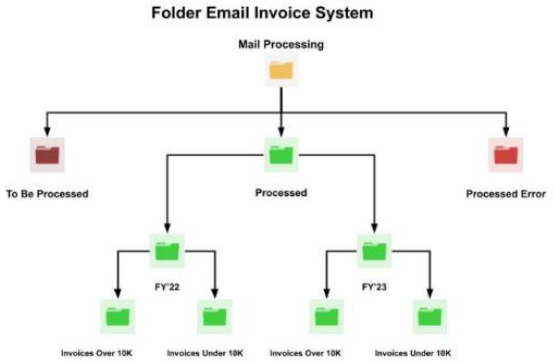

Diagram 4. *Folder Email Parser Invoice System*

6.2 Email Parser Execution

For the email parser to run, an executable file (.exe) is used. Any project manager who wants to use the email parser must run this file. When the file is run for the first time, the project manager is prompted to select the shared folder to which the emails will be extracted for processing by the program.

Once the folder is selected, the program will work correctly and automatically process any email added to that folder. In the following usages of the email parser, it is not necessary to select the shared folder again when starting the program.

To create this .exe file the **pyinstaller [9]** library was used and specified the libraries of the mysql-connector **[10]** as the pyinstaller library by default does not add these libraries in the executable file and there is an error when executing.

7 Relational Database

To store all invoices, a relational database with predefined tables is needed. Within the company, I was offered two frameworks to use:

- PostgreSQL **[11]**.
- MySQL **[12]**.

The main difference between the two is where the database logic is developed. While in the case of MySQL it is developed in the back end itself, in the case of PostgreSQL it is developed within the database itself.

The choice is **MySQL**, as its better to use the database only for INSERT, UPDATE and DELETE and have all the logic in the back end part, either in the parser. Also, to do all the logic in the database itself with PostgreSQL will require a larger learning curve.

These are the tables used for the invoice tracker:

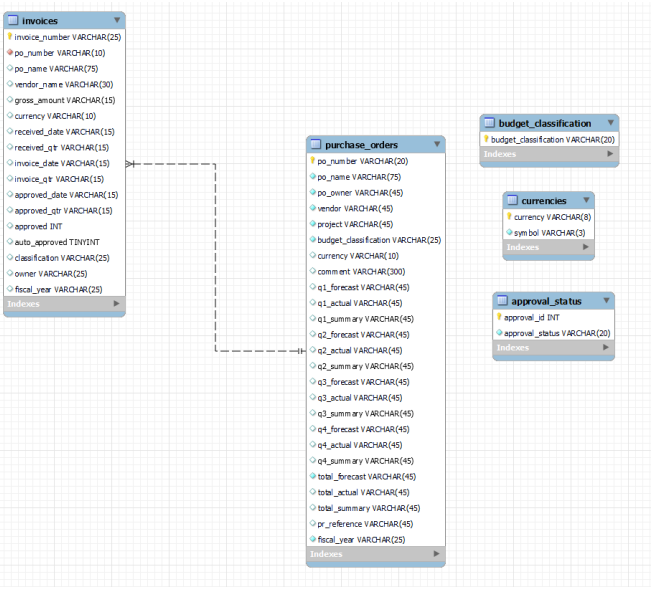

Diagram 5 . *MySQL Database Architecture*

8 Web Back End

The back end of the web has been developed with **Python**.

8.1 Rest API

To develop the back end of the invoice tracker web a REST API **[13]** is implemented.

This architecture is used as it has so many benefits, here some of them :

- **Scalability and Flexibility**: separating the front end and back end into distinct components allows them to be developed, deployed, and scaled independently.
- **Technology Independence**: by using a REST API, the front end and back end can be built using different technologies and programming languages.
- **Code Reusability**: the separation between the front end and back end encourages the development of reusable code.
- **Improved Maintainability**: when the front end and back end are decoupled, it becomes easier to maintain and update each component separately.

All the necessary information regarding the available API calls and their respective endpoints is described in detail in **Appendix A1**, which contains comprehensive tables providing a clear overview of the API's functionality and usage.

8.2 Main Back End Funcionalities

Database Connection:

- The back end is responsible for establishing a connection with the MySQL database to execute all queries.

Purchase Order Budget Calculations:

- Calculates the budgets of the Purchase Orders (POs) whenever there is a modification in the

invoice tracker or when the PO itself is edited.

Addition of Invoices and Purchase Orders:

- Handles the preparation of queries to add all attribute values of the POs and invoices in the database.

Update of Invoices and Purchase Orders:

- Handles the preparation of queries to update all attribute values of the POs and invoices in the database.

Delete of Invoices and Purchase Orders:

- Handles the preparation of queries to delete the POs and invoices in the database.

Data Processing:

- Performs data formatting and processing tasks for invoices and POs, ensuring that the data is properly structured and ready for further use.

8.3 Functional and Unit Tests

To test that the back end work as expected some unit and functional tests were passed so that the API's and all the other functions like the budget calculators work correctly.

When the functional tests of the API's were conducted, it was found that 90% of them were already working perfectly and doing what they were supposed to do.
However, it was noticed that one function related to updating the budgets of the purchase orders once an invoice was added was not working correctly.
Now the problem is fixed, and the back end passes all 32 functional tests.

Unit tests were performed to different functions that use the APIS for their operation and the results have been positive since they have passed the 118 tests.

To see the results of the tests you can consult the report "**backend-testing-report.html**" that you can find in the dossier.

9 WEB FRONT END

The front end of the project is divided into 3 views and has been developed with **ReactJS** and using the PrimeReact library **[14]**:

- **Select Web Permissions View**
- **Invoice Tarcker View**
- **Budget Forecast View**

9.1 Select Web Permissions View

To be able to select what permissions the user must access the web, a landing page is created where the user has to choose between two options:

- Edit Mode.
- View Mode.

Once these options are selected, the Invoice Tracker view opens with the permissions selected. (*Image A2.1*)

Nevertheless, within the views themselves, the users can change the permissions if they wish.

9.2 Invoice Tracker View

Invoice Table

In this view, the user can see all invoices registered in a table where mostly of those that were processed with the email parser tool and then stored in the database. This table differs in color according to the approval status of the invoices (*Image A2.2*):

- Green: invoices that have already been approved.
- Yellow: invoices that are pending approval.
- Red: invoices that have been rejected.

Table Filters

The user can also filter the table according to the different attributes of the invoices so that only the desired invoices are shown, as well as perform a global search through a global search browser.

Invoice Modification Functionalities

If the user has editing permissions, they can do some actions to keep up to date the invoice tracker registers such as:

- Add a new invoice register manually (*Image A2.3*).
- Edit an existing invoice register (*Image A2.4*).
- Delete an existing invoice register (*Image A2.5*).

Exporting Data

The user also has the option to export the data of the invoice table in different formats:

- Export data in PDF (*Image A2.6*).
- Export data in Excel (*Image A2.7*).

9.3 Budget Forecast View

Every invoice that can be found in the Invoice Tracker View is associated with a purchase order that is added in this one.

Purchase Orders Table

The main functionality of this view is the collection and budgeting of the budget, which is basically assigned to each purchase order. To do it all the purchase orders registered in the database are displayed (*Image A2.8*).

Budget Quarter Tracker

This view also allows the user to track the evolution of purchase orders throughout the year, so they can see if the original budget forecast was correct or if adjustments need to be made (*Image A2.9*).

Purchase Orders Invoices

It is also possible to see within each PO what specific invoices are associated with each one (*Image A2.10*).

Purchase Orders Modification Functionalities

In this view, only the users with the appropriate permissions can do modification action to the POs such as:

- Add a new PO register (*Image A2.11*).
- Edit some fields of a PO register (*Image A2.12*).
- Delete only those PO that don't have any invoice attached (*Image A2.13*).

Exporting Data
The user also has the option to export the data of the purchase orders table in different formats:

- Export data in PDF (*Image A2.14*).
- Export data in Excel (*Image A2.15*).

9.4 Beta Testing

To test the web, it was used what is called **beta testing** which is an important method for collecting feedback from users and identifying potential problems or improvements before the official release. This method consists of asking the user to perform an action, a user case, without any hints and see if they can do it and if they find a bug.

The users that perform these beta tests have different roles, like people who have never used the web and are not the end users, and project managers who are the main stakeholders and end users of the project.

The results of this beta testing are in the pdf in the dossier called "**Beta Testing of Invoice Tracker Web.pdf**".

10 Results

Once we have seen the development of the project lets see what the results have been related to the completion of the user stories of the product backlog and the benefits that this project has had for the project managers in their daily work with invoices.

10.1 Metrics of the Project User Stories

In the table below (*Table 3*) **25/32 user stories** that were defined in the Product Backlog have been completed.

Priority	User Stories	Done	Done Forecasted Story Points	To Do
Must Have	21	21	98 SP	0
Important	4	3	9 SP	1
Nice to Have	7	1	4 SP	6
Totals	32	25/32	111 SP	7/32

Table 3. *User Story Resume by Priority and Status*

From those 32 user stories all the ones that had a Must Have priorization (21/32) were successfully completed.

Only 7/32 User Stories are still to be completed:

- 1 user story with the Important priorization.
- 6 user stories with the Nice to Have priorization.

The following table (*Table 4*) shows the difference between the forecasted and actual time for each of the Stroy Points completed user stories epics.

Epics	Forecasted Story Points	Actual Story Points	Time Resume
Email Parsing and Data Extraction	13 SP	19 SP	- 6 SP
Invoice Management	30 SP	34 SP	- 4 SP
Sorting and Filtering Invoices	9 SP	8 SP	+ 1 SP
Expense Tracking and Reporting	16 SP	11.5 SP	+ 4.5 SP
Purchase Orders Management	28 SP	26 SP	+ 2 SP
Sorting and Filtering POs	9 SP	4 SP	+ 5 SP
Web Access Permissions	6 SP	2 SP	+ 4 SP
Totals	111 SP	104.5 SP	+ 6.5 SP

Table 4. *User Stories Time Resume Forecasted vs Actual SP*

As we can see, although the Email Parsing and Data Extraction and Invoice Managament epics have exceeded the time estimated at the beginning, with the rest of the epics the expected time was reduced, giving as a difference that the work has been done **6.5 SP** (19.5 hours) earlier than expected, which represents a positive deviation of approximately **6%**.

10.2 Benefits of the Solution

To evaluate the benefits of the project on the Project Managers' tasks, it is essential to get their feedback on the improvement in the quality of their work related to invoices once the tool has been implemented.

These are the evaluations of the 2 Project Managers who are going to use the tool:

Claudia Herz - (Financial Project Manager):
"
Time savings:

- Manual invoice register (typing all details into an Excel): 3mins x invoice.
    - Example: In 2 quarters I had 130 invoices = 260 invoices x year = 780 mins = 13 hours. That does not sound much at first, but if we automate more and more processes this adds up.
    - Tool is scalable and could be used by other departments.
- Quick and easy to use filter and search functionality.
- Budget overview in few clicks including easily understandable charts (manual work to do this in Excel)

- Fast & easy data export

Quality improvement:

- Automated solution ensures better quality through e.g., field validation, automatic mapping between PO#/ PO Name to invoice. Manual entry is more prone to errors.
- View only access ensures data is not manipulated by unauthorized personnel.
- UI design and features like budget overview allow for better & faster overview, thus issues can be spotted easier.

"

Juan Manuel Melendez - (Project Manager Team Manager):

"

This platform will be a great help for our team in terms of funding planning and controlling.

You have done a great job. And I would like to highlight the following points as key success factors:

- You have been adapting and improving the user experience of the tool step by step. Adapting it according to the feedback you got from the subject matter experts.
- You focused first on the MUST-have features you got from the subject matter experts.
- You designed and developed the platform in a module way. Always considering that this tool will eventually need to be extended by other developers in the future.
- You have faced several technical challenges when developing the parser. However, you were able to find different technical alternatives to overcome them.

Once again, congratulations for the result and thank you very much in advance from our entire team.

"

11 CONCLUSIONS

After completing this project, it is time to draw up the conclusions. This project, which began in early February, has involved facing several challenges, both technical and managerial, with the aim of making it useful and functional for the university and the company (HP).

Technical Challanges

As for the technical challenges, it is worth noting that this was the first time programming a website with **React JS**.

Despite the inexperience in this language, I decided to develop the system with this programming language as it is currently one of the most used by large companies and in the world of work. Admittedly, at the beginning, it was difficult to understand the way of thinking about the components and how they communicate with each other, as it was quite different from similar languages such as HTML or JavaScript.

However, once the initial stage of learning how ReactJS works, web development became more interesting and dynamic, allowing to work more confidently and quickly.

It is also worth noting that this is the first time a **REST API** was used to communicate with the back end. At first it was difficult to fully understand the API concept, but in the end, it was appreciated its usefulness in decoupling the back end and the front end into two totally independent modules. This advantage is very interesting, as it allows to design a very modular project architecture and facilitates the modification of one of them, if necessary, without affecting the other one directly.

One important technical challenge is building an **End-to-End** system that includes the database, front end, back end, and data collector (in this case, the Email Parser). This hands-on experience has helped gain a real understanding of how projects like this work and allowed me to identify my preferred development area, which, in this case, is the front end.

Management Challanges

Focusing now on the management challenges faced during the project, it is worth mentioning that being an end-to-end project has been a great challenge itself. In addition to constantly changing the programming language and methodology, it was difficult to move from the role of programmer to project manager. This has been, without a doubt, the biggest challenge faced, as sometimes it was difficult to leave programming aside to perform project management tasks.

However, this situation has allowed me to learn a lot and to understand the two perspectives that are often found in a project: that of the Project Manager and that of the programmer.

Real Client Challange

Finally, it should be noted that working on this final degree project for the company HP has posed several challenges.

First, the level of personal demand has increased considerably, since it was considered that it wasn't only a project for the university, but it was a project that had to satisfy and approve an external client who had contracted the services to develop a solution for their needs. This high level of internal demand has motivated me to do my best to ensure that the result was as perfect as possible.

Moreover, during the development of the project, there have been changes in the requirements from the stakeholders. This situation has forced me to adapt quickly to the changes, otherwise the project would not progress. To achieve this adaptation, it should be noted that the **Scrum methodology** has been of great help, since it has provided me with the necessary guidelines to adapt to the changes with the minimum possible impact on the project schedule.

12 Future improvements

This application has many possibilities for improvement for the future, which is why it is always aimed at development, so if anyone wants to add new functionalities or any other type of improvements, these do not have the complications that a very robust system .
Here are some possible improvements that could be applied to the system:

- Access to the web through HPID Authentication (Way HP Employee login every HP Service).
- Implement an Outlook macro to automatically download emails to the shared folder.
- Enabling and disabling active and inactive invoices or purchase orders in the table displayed in the web.
- A log register for the records of invoices and POs, indicating who changed or added them and when.
- Implementation of a function to add and remove attributes of invoices or POs in the registration table for invoices or POs.
- Perform more thorough testing of the front end of the website.

13 Acknowledgements

First, I would like to express my sincere gratitude to my corporate TFG tutor, Juan Manuel Melendez, for his invaluable help and support at every stage of this project. His guidance and knowledge have been fundamental to my growth as a computer engineer and to develop this project.

I would also like to thank my colleague Claudia Herz for her constant cooperation and willingness to help me whenever I needed it. Her dedication and commitment were a great support during the development of this thesis.

I would also like to mention the unconditional support of my family and partner during these 4 years of studying and trainning as a computer engineer. Their constant encouragement and understanding have been fundamental pillars that enabled me to achieve this success.

14 Bibliography

[1] "Excel Template for Invoices," Someka. [Online]. Available: https://www.someka.net/products/invoice-tracker-excel-template/
[2] "Email Parser," Email Parser. [Online]. Available: https://www.emailparser.com/
[3] "SAP Invoice Management SW," SAP. [Online]. Available: https://www.sap.com/products/financial-management/invoice-management.html
[4] "Scrum of One," Kodeco. [Online]. Available: https://www.kodeco.com/585-scrum-of-one-how-to-bring-scrum-into-your-one-person-operation
[5] "What is Kanban," Kanbanize. [Online]. Available: https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban
[6] "Jira Sotware Benefits," educba. [Online]. Available: https://www.educba.com/jira-project-management-tool/
[7] "Stories, Epics and Initiatives," Atlassian. [Online]. Available: https://www.atlassian.com/agile/project-management/epics-stories-themes
[8] "Story Points and Estimation," Atlassian. [Online]. Available: https://www.atlassian.com/agile/project-management/estimation
[9] "Pyinstaller library," Pyinstaller. [Online]. Available: https://pyinstaller.org/en/stable/
[10] "MySQL Connector library," MySQL. [Online]. Available: https://dev.mysql.com/doc/connector-python/en/
[11] "PostgreSQL," PostgreSQL. [Online]. Available: https://www.postgresql.org/
[12] "MySQL," MySQL. [Online]. Available: https://www.mysql.com/
[13] "What is a Rest API," IONOS. [Online]. Available: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-openapi/
[14] "PrimeReact Library," PrimeReact. [Online]. Available: https://primereact.org/

APENDIX

A1. API CONTRACT

Description: API for managing invoices and purchase orders.

Endpoint	Method	Description
/api/invoices	GET	Get all invoices.
/api/purchaseorders	GET	Get all purchase orders.
/api/addinvoice	POST	Add a new invoice register.
/api/updateinvoice	POST	Update an existing invoice register.
/api/deleteinvoices/{invoice_number}	DELETE	Delete an invoice register by invoice number.
/api/addpurchaseorder	POST	Add a new purchase order register.
/api/updatepurchaseorder	POST	Update an existing purchase order register.
/api/deletepurchaseorder/{po_number}	DELETE	Delete a purchase order-register by po number.
/api/budgetclassifications	GET	Get all the budget classification options.
/api/currencies	GET	Get all the currencies options.
/api/approvalstatus	GET	Get all the approval status options.

Data Schemas:

Invoice Received from Front End:

Attribute	Type
invoice_number	String
po_number	String
invoice_date	String
received_date	String
approved	Integer
approved_date	String
gross_amount	Float
auto_approved	String

Invoice Returned to the Front End:

Attribute	Type
invoice_number	String
po_number	String
po_name	String
vendor_name	String
gross_amount	String
currecy	String
received_date	String
received_qtr	String
invoice_date	String
invoice_qtr	String
approved_date	String
approved_qtr	String
approved	Integer (1,2,3)
auto_approved	Integer (0,1)
classification	String
owner	String
fiscal_year	String

Purchase Order Received from Front End:

Attribute	Type
po_number	String
po_name	String
po_owner	String
vendor	String
project	String
budget_classification	String
currency	String
comment	String
q1_forecast	Float
q2_forecast	Float
q3_forecast	Float
q4_forecast	Float
pr_reference	String
fiscal_year	String

Purchase Order Returned to the Front End:

Attribute	Type
po_number	String
po_name	String
po_owner	String
vendor	String
project	String
budget_classification	String
currency	String
comment	String
q1_forecast	String
q1_actual	String
q1_summary	String
q2_forecast	String
q2_actual	String
q2_summary	String
q3_forecast	String
q3_actual	String
q3_summary	String
q4_forecast	String
q4_actual	String
q4_summary	String
total_forecast	String
total_actual	String
total_summary	String
pr_reference	String
fiscal_year	String

A2. WEB FRONT-END IMAGES

1. Web Permissions Selection View

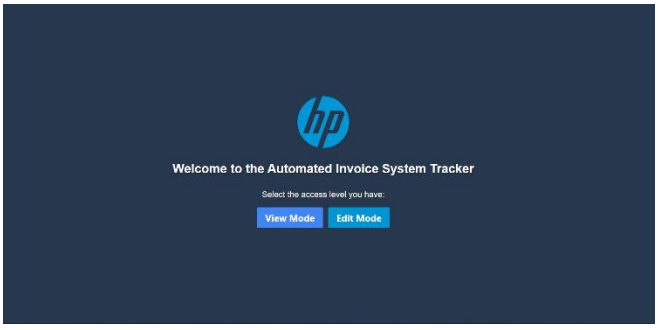

Image A2.1

2. Table of Invoices in the Invoice Tracker View

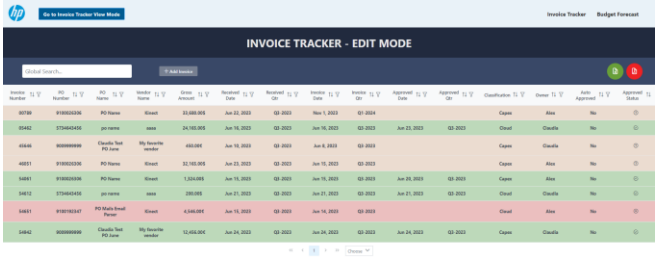

Image A2.2

3. Add new invoice register form

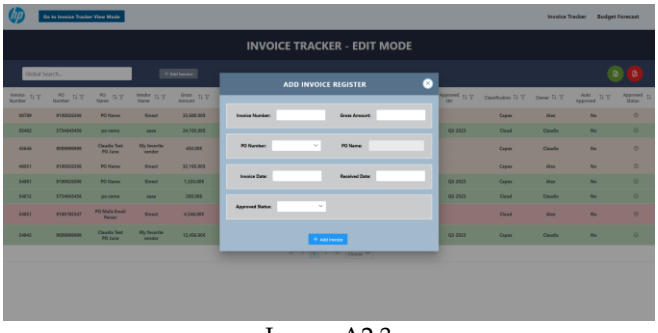

Image A2.3

4. Edit an existing invoice register

Image A2.4

5. Delete an existing invoice register

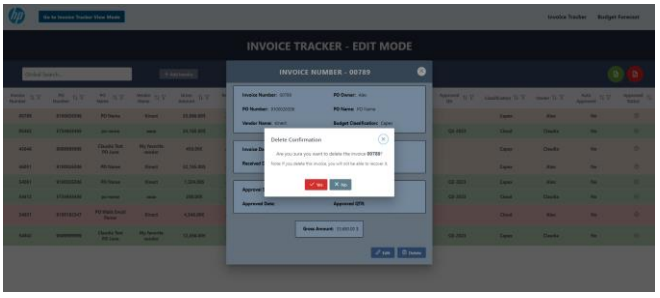

Image A2.5

6. Exporting the invoices in PDF

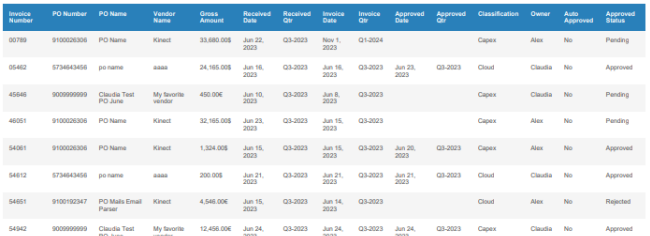

Image A2.6

7. Exporting the invoices in EXCEL

Image A2.7

8. Table of Purchase Orders in the Budget Forecast View

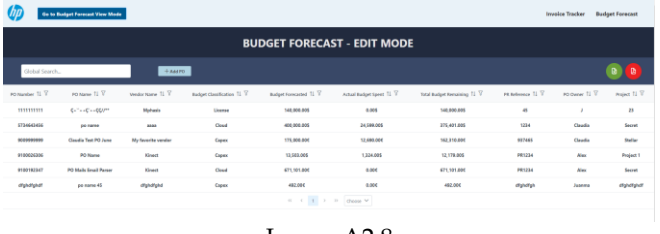

Image A2.8

9. Budget Quarter Tracker Charts

Image A2.9

10. Invoices Attached to a Purchase Order

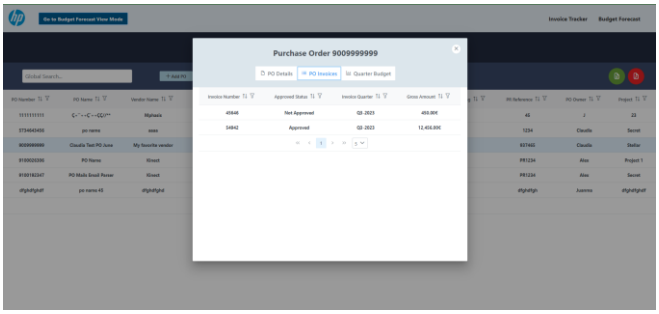

Image A2.10

11. Add a new purchase order register

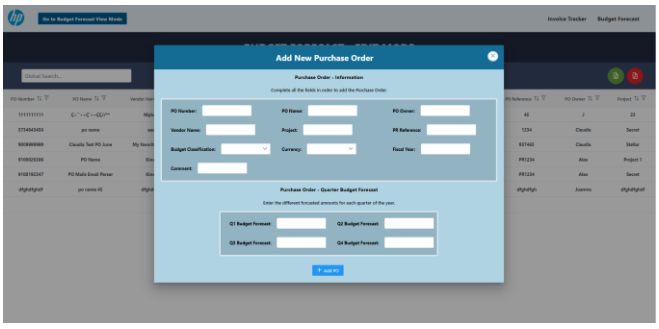

Image A2.11

12. Edit an existing purchase order register

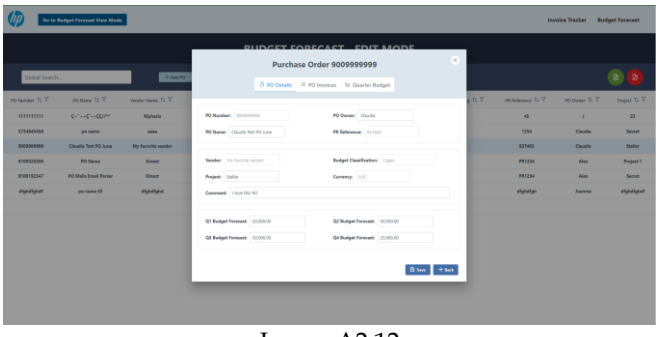

Image A2.12

13. Delete an existing purchase order register

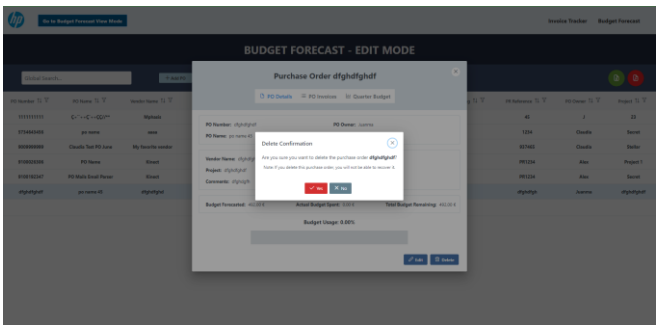

Image A2.13

14. Exporting the purchase orders in PDF

PO Number	PO Name	Vendor Name	Budget Classification	Budget Forecasted	Actual Budget Spent	Total Budget Remaining	PR Reference	PO Owner	Project
1111111111	Ç+¨++Ç¨++ÇÇ// **	Mphasis	License	140,000.00$	0.00$	140,000.00$	45	J	23
5734643456	po name	aaaa	Cloud	400,000.00$	24,599.00$	375,401.00$	1234	Claudia	Secret
9009999999	Claudia Test PO June	My favorite vendor	Capex	175,000.00€	12,690.00€	162,310.00€	937465	Claudia	Stellar
9100026306	PO Name	Kinect	Capex	13,503.00$	1,324.00$	12,179.00$	PR1234	Alex	Project 1
9100192347	PO Mails Email Parser	Kinect	Cloud	671,101.00€	0.00€	671,101.00€	PR1234	Alex	Secret
dfghdfghdf	po name 45	dfghdfghd	Capex	492.00€	0.00€	492.00€	dfghdfgh	Juanma	dfghdfghdf

Image A2.14

15. Exporting the purchase orders in EXCEL

	A	B	C	D	E	F	G	H	I	J	K
1	PO Number	PO Name	Vendor Name	Budget Classification	Budget Forecasted	Actual Budget Spent	Total Budget Remaining	Currency	PR Reference	PO Owner	Project
2	1111111111	Ç+¨++Ç¨++ÇÇ// **	Mphasis	License	140000.00	0.00	140000.00	USD	45	J	23
3	5734643456	po name	aaaa	Cloud	400000.00	24599.00	375401.00	USD	1234	Claudia	Secret
4	9009999999	Claudia Test PO June	My favorite vendor	Capex	175000.00	12690.00	162310.00	EUR	937465	Claudia	Stellar
5	9100026306	PO Name	Kinect	Capex	13503.00	1324.00	12179.00	USD	PR1234	Alex	Project 1
6	9100192347	PO Mails Email Parser	Kinect	Cloud	671101.00	0.00	671101.00	EUR	PR1234	Alex	Secret
7	dfghdfghdf	po name 45	dfghdfghd	Capex	492.00	0.00	492.00	EUR	dfghdfgh	Juanma	dfghdfghdf
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											

Image A2.15