Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

PONDD: an online platform for printing-on-demand and delivery of daily press, magazines and books

Nikolaos Konstantinou[1], Evi Papaioannou[2], Christos Kaklamanis[2]

University of Patras, Greece
University of Patras and CTI "Diophantus", Greece
Evi Papaioannou, papaioan@ceid.upatras.gr

Abstract. Motivated by difficulties experienced in the distribution of printed press, mainly daily press, i.e., newspapers, but also sometimes of periodical press, i.e., magazines, or even books, we designed and developed PONDD, an online platform for printing-on-demand and delivery of daily press, magazines and books. PONND can serve as an innovative service which can directly bring together readers and publishers via printing points, like for example kiosks or mini markets. Readers can select their desired newspaper, magazine or book, define their desired features for the print-out (e.g., single/duplex, colored/black&white) and choose an available nearby printing point for serving their requests, i.e., for picking-up or having delivered to them their selected newspaper, magazine or book. PONDD is easy-to-use and responsive offering a comprehensive set of well-defined functionalities. In our current era, information is a critical resource and the internet should be exploited not only as a modern information distribution channel but also as a backbone for supporting the continuation of free, undisturbed and uninterrupted distribution of traditional printed press.

Keywords. Online platform, press, printing-on-demand, free, undisturbed and uninterrupted distribution of printed press, environmental sustainability, Angular, Typescript, Java, Spring.

1. Introduction

The traditional printing industry, particularly for daily press such as newspapers, periodical press like magazines, and books, faces several significant challenges. The primary issues stem from the inefficiencies in the distribution and printing processes [1], [2]. Overproduction is a major problem, with publishers often printing large quantities of materials to anticipate demand. This approach leads to substantial waste when unsold copies are discarded. Additionally, maintaining large inventories of printed materials incurs high storage costs and requires significant logistical coordination, further increasing operational expenses.

Distribution is the most crucial issue in the traditional printing industry [3]. The time required to transport printed materials from central warehouses to various retail points or directly to customers can be considerably high [4]. This issue is worsened for more remote places such as islands, villages, and other remote areas, where the distribution process is significantly slower. The logistical challenges involved in reaching these remote locations include limited transportation options, longer travel distances, and infrequent delivery schedules. As a result, readers in these areas often experience delays in receiving their daily press, which can negatively impact customer satisfaction and the perceived relevance of the content. This issue is especially critical for daily press and news, which are expected to be delivered in the very early hours before the day starts. Ensuring that newspapers are available for

readers in the morning requires overnight printing and early morning distribution, which is both inconvenient and costly for printing points and the daily transfer logistics. The requirement for early morning deliveries places additional strain on logistical operations, leading to increased operational costs and potential delays that can undermine the relevance of the news content. Further implications can arise in cases where distribution is performed by a single agent/company thus creating room and conditions for potential symptoms of monopolistic situations negatively impacting readers and publishers as well as freedom of information and freedom of press [5], [6], [7], [8]. Moreover, the environmental impact of traditional printing and distribution practices cannot be overlooked. The logistics involved in transporting printed materials contribute to a significant carbon footprint, and the waste generated from unsold copies also worsens the environmental impact the printing industry has.

From a socio-culture perspective [9], [10], [11], [12], it is important for readers to maintain the traditional option of getting the news via real, printed newspapers. Older people who are not familiarized with digital devices and electronic services find it too hard to read the news through scrolling on a tablet while deeply appreciate their rather ceremonial daily habit of going to a kiosk and purchasing a printed newspaper to spend their day on. Also, younger people, despite being familiarized and even highly skilled in digital technology and services, still wish to have the option of slowing down their life pace via reading a classical, printed newspaper, magazine or book. In our current internet-based era, it seems that digital press is promoted as a single option continuously limiting the room for printed press. But, limiting choices down to singletons does not support freedom, independence and free will that individuals must enjoy in healthy and well-functioning societies.

To address these problems, we designed and developed PONDD, an online platform for printing-on-demand and delivery of daily press, magazines, and books. PONDD connects readers and publishers directly through local printing points, such as kiosks or mini-markets redefining, rationalizing and normalizing the role of intermediates on the distribution of press. Readers can select their desired newspaper, magazine, or book, customize the print-out (e.g., single/duplex, colored/black & white), and choose a nearby printing point for pick-up or delivery. PONDD eliminates overproduction by allowing on-demand printing. Printing points only print the number of copies that are ordered, greatly reducing waste and saving resources. The platform reduces the need for large inventories and extensive storage facilities, cutting storage costs and simplifying logistics. Local printing points print materials as orders are placed, ensuring efficient use of resources. This model not only saves money for publishers but also reduces the complexity and cost of distribution. PONND eliminates the most crucial problem in the printing industry, distribution. By decentralizing the printing process, publishers can now meet early morning delivery demands without the heavy logistics of traditional methods. This makes it easier for printing points and ensures newspapers are up to date and on time. Moreover, reducing the need for long-distance transport to remote places improves delivery times and reliability for readers in islands, villages, and rural areas. This means that even in the most remote locations, readers can have the latest news and magazines delivered without significant delays.

Additionally, significant benefits for small kiosks and mini-markets are gained with the use of the platform. These small businesses can become printing points without the risk of buying large quantities of newspapers and magazines upfront. Since printing is done on-demand, these businesses do not need to worry about unsold inventory. This model allows them to offer a wide range of publications, attract more customers, and earn additional income with minimal risk. By partnering with PONDD, small kiosks and mini-markets can increase their profitability and provide a valuable service to their communities.

To the best of our knowledge, no similar application for printing-on-demand and delivery of press (i.e., daily or weekly newspapers, magazines) currently exists, at least at a national level (i.e., the relevant market in Greece). While there are applications that offer print-on-demand services for books, none provides a decentralized solution for both daily press and books with the specific benefits and problem-solving approach that PONDD offers.

PONDD application has been developed via the exploitation of a robust set of open-source technologies. For the frontend, it utilizes Angular and Angular Material 14, alongside Node.js and Typescript, to create a responsive and dynamic user interface. Key libraries like Chart.js and Google Maps are integrated to provide interactive data visualization and location-based services. The backend is powered by Java 17 with Spring and Spring Security 3.2 frameworks to ensure secure and efficient processing, while AWS is used for file uploads. The application's data is managed with PostgreSQL, ensuring reliable and scalable database management.

The rest of the paper is structured as follows. In Section 2, we address in detail the design and implementation of the system. In Section 3, we present a use scenario to demonstrate the functionalities, including the interactions between publishers, users, and printing points. We conclude in Section 4 by discussing the results, the positive impacts on all involved parties as well as on the environment and even society and culture and outlining potential future enhancements.

2. System implementation

PONDD, short for Printing ON-Demand and Delivery of daily press and books, revolutionizes the delivery and distribution of print press such as newspapers, magazines, and books. It connects readers directly with local printing points via a digital platform, eliminating the need for extensive inventory and reducing waste. The system is designed to cater to three main user roles: Customers, Publishers, and Printing points, each with tailored access and functionalities within the platform to suit their specific needs. After successful sign-in from PONDD main page (https://pondd-frontend-f7866f45f380.herokuapp.com/home) users according to their role are redirected to a specific environment tailored for their needs. Users can interact with the PONDD system through a dedicated portal (https://pondd-frontend-f7866f45f380.herokuapp.com/shop), where they can choose publications, order, customize printing options, and select nearby printing points for pickup or delivery. Publishers access their management interface (https://pondd-frontend-f7866f45f380.herokuapp.com/publisher) to upload new content, manage publications, and track orders and distribution analytics. Printing points use their specialized dashboard (https://pondd-frontend-f7866f45f380.herokuapp.com/print-point) to handle print requests, manage operational settings, and update order status.

2.1. User interface and functionalities

In this section, we present the user interface and the functionalities of the PONDD application. We provide detailed descriptions and screenshots to illustrate the available features and the user experience for different roles within the system. The PONDD application offers distinct interfaces and functionalities tailored to specific user roles, ensuring a seamless and efficient experience for all users. In particular, we address interface and functionalities offered to authenticated and non-authenticated users, that is functionalities available to registered users and visitors who browse the application without logging in, respectively. Then, we address interface and functionalities offered to "Publishers" who upload and manage their publications. Finally, we address interface and functionalities offered to Printing points that handle print requests and manage deliveries. We include screenshots of the relevant interfaces and describe the key functionalities available to users in each category/role.

2.1.1. Authenticated / non authenticated users. The PONDD home screen, available to both authenticated and non-authenticated users, is shown in Figure 1. Users have the option to filter items based on category, change the current language of the system, and choose whether to sign up (i.e., register) or sign in (i.e., log in). Additionally, there is a functionality to view a list of current printing points, where details of each printing point can be seen and a PDF containing these details can be downloaded. Users can also display a map, filter it based on their postal code, and view all nearby printing points. Furthermore, there is an accessibility menu available, as well as options at the footer for additional information about the application.

Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

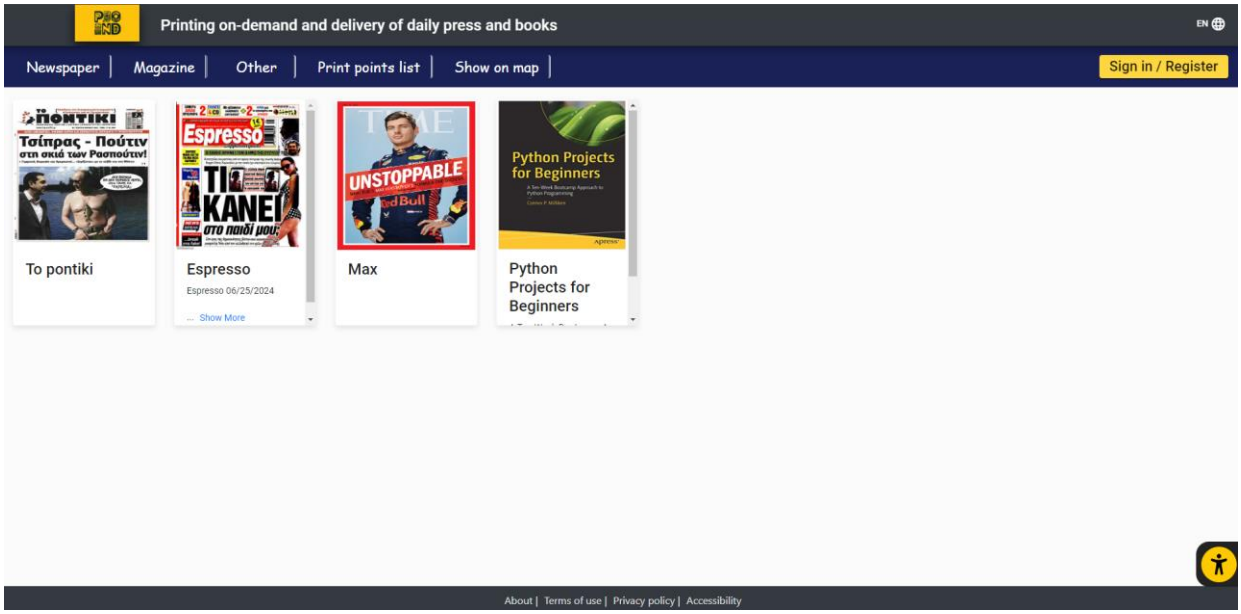

Figure 1. Home page

Upon selecting an item, the system redirects to the item details page. As shown in Figure 2, this page provides full details for the selected item, including the availability period, full description, author, number of pages, and cost. The preview image is displayed in full size, making it more visible. The option to 'Add to cart' is not available to non-authenticated users.

Figure 2. Item details

After selecting “Sign in / Register” from the top right corner at the homepage (depicted in Figure 1), the system prompts the user to select an action from the access modal, as shown in Figure 3.

Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

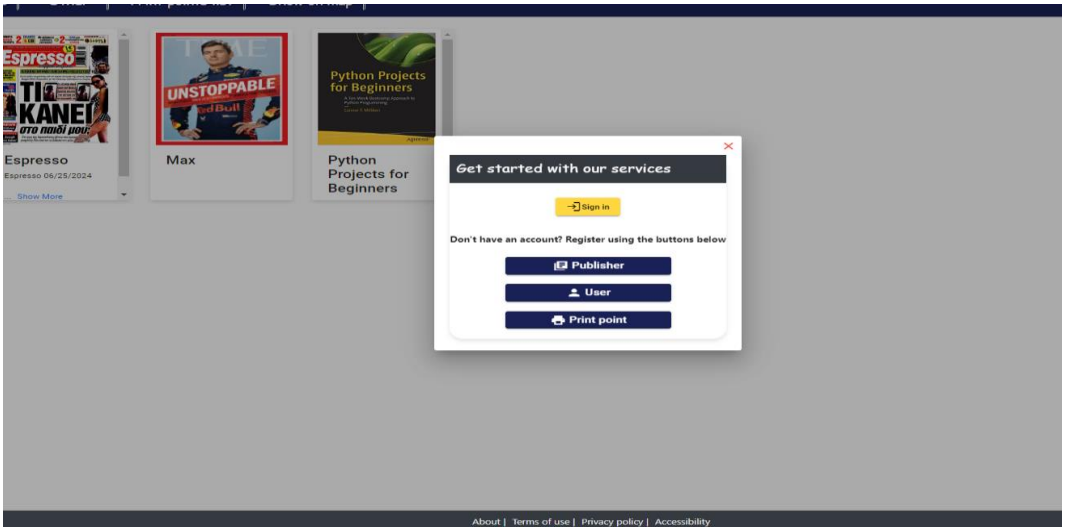

Figure 3. Sign in / Sign up modal

After signing in as a 'User', the home page shown in Figure 1 appears while the "Add to cart" option (depicted in Figure 2) is now enabled. While logged in, users can see two widgets at the top bar: the cart widget, which displays the items added to the cart, and the account widget, which shows the username of the authenticated user. This setup allows users to easily verify their authentication status. Additionally, the widget provides a dropdown menu with options to "Logout", navigate to “Account details" where they can edit details of their account apart from username and registered email, and "Orders" where placed orders are available. These widgets are depicted in Figure 4, where cart details are shown so that users can see the items that are added to their cart before finishing the order. To proceed with their order, users must select a printing point for each item to be printed. There are also options regarding printing preferences like for example binding method, paper type, print quality which may alter the print cost of each item. The system calculates the printing cost based on the selected printing preferences, item cost and pages and informs the users for the resulting total cost. The default delivery method of each item can be also changed. If "Deliver to address" is chosen, further location details must be filled in during the next steps before the completion of each order.

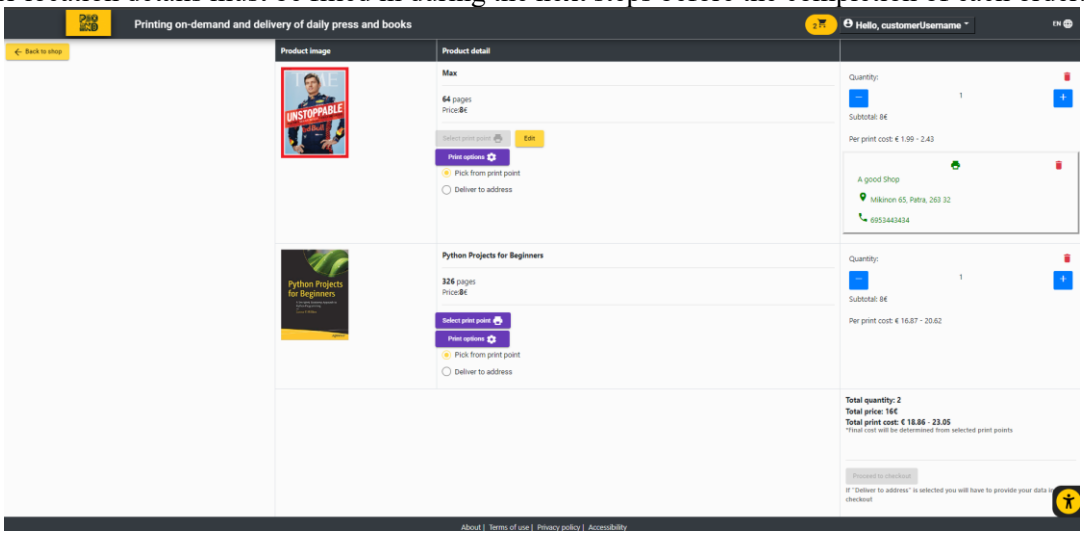

Figure 4. Cart details

After proceeding with the order in Figure 4, the user must select either the payment option "Card" where card details are requested, or "Pay with cash" to complete the order. As shown in Figure 5, after

Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

a successful order placement, the system displays an overview with the most important information regarding the order just placed. The order ID is prominently shown, and detailed information about the selected printing options and the selected printing point for each item can be viewed by expanding the available dropdown menus. There is also the option to download a PDF with the order details.

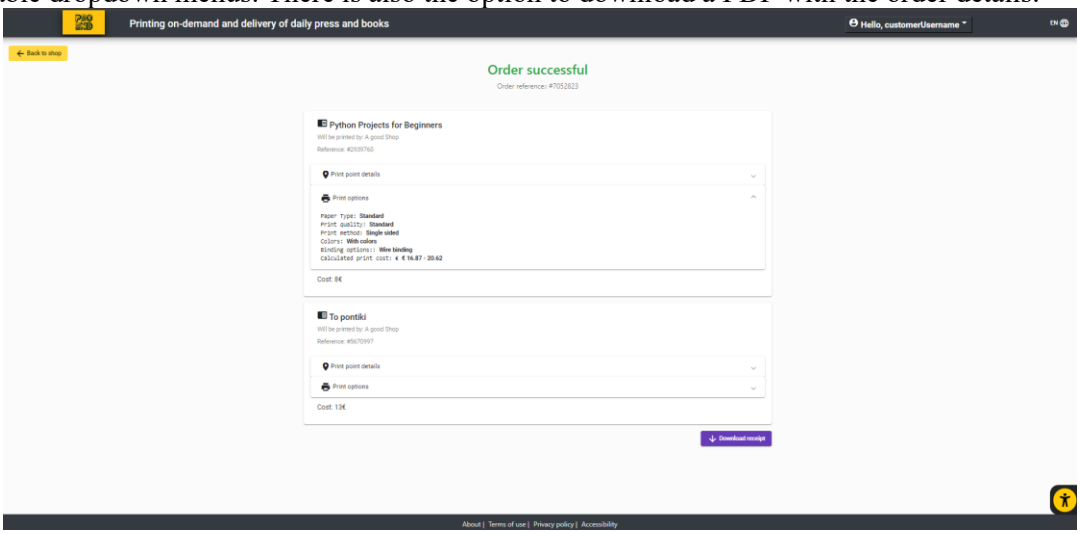

Figure 5. Order Successful

Lastly, the authenticated user can be redirected to the "control panel" through the dropdown menu in account widget, choosing either the "Orders" or "Account" option. As shown in Figure 6, the control panel provides options for the user to edit their account details and change their password. Users can view current and past orders and filter them based on the order reference number or the date the order was submitted. Detailed information about the order is displayed, including the order number, date ordered, total items, and total cost. Each item within the order is listed with its reference number, title, price, and status (ready or pending), and users can view the selected printing point and print options for each item. Users can download a summary of the order as a PDF or select and download more detailed information for an order item as a PDF. The control panel follows a clear and easy to use design, featuring filters that help users find their desired orders quickly and provide informative navigation options. Users can easily manage their accounts and orders through a clear side menu that displays all available options.

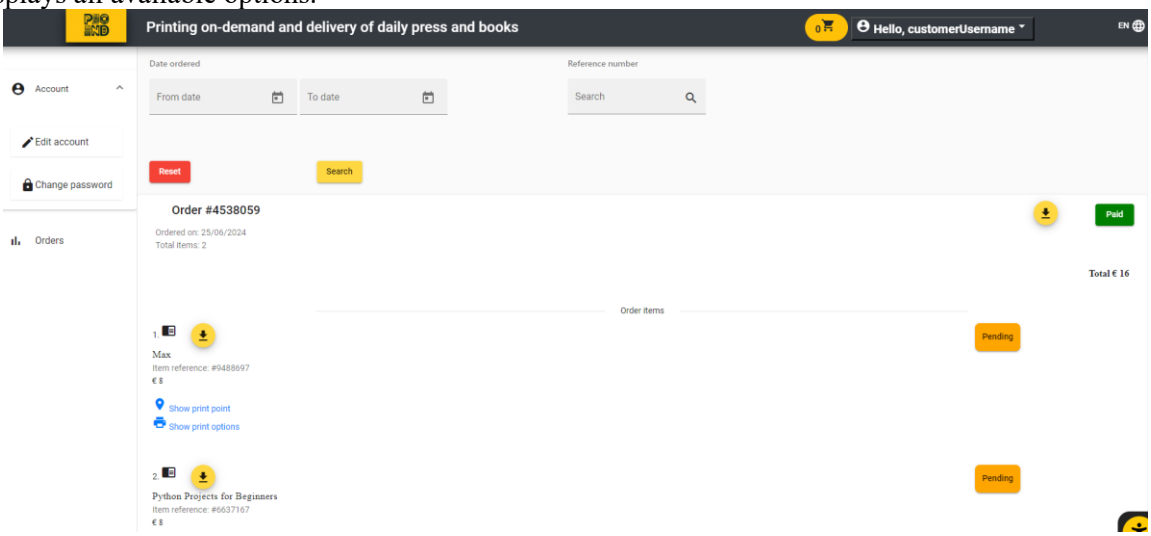

Figure 6. Customer control panel

Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

2.1.2. Publisher. When a publisher logs into the system, the main page, shown in Figure 7, features an interface similar to the "control panel" used by customers. The publisher can navigate through various options via the side navigation bar. Users in the category "Publisher" can edit their account details, change passwords, view current uploads, upload new items, and examine analytics and history related to their uploads through visualizations. Current uploads are categorized into three sections based on their status: active, upcoming, and deleted. The status of an upload is determined by the availability period set for the item. Additionally, within each section, there are filters available to sort uploads by category (newspaper, magazine, other) and by upload date, enhancing the publisher's ability to manage and view content efficiently. When publishers upload an item, they must enter several mandatory fields: a preview image, the content of the item as a PDF file, the main title, number of pages, price, category, and item availability. Additionally, publishers can set optional fields that may be more suited to the type of the upload, such as author, short description, and full description. These fields allow for an extensive description of the item, making it easier for customers to understand and choose items to print on-demand. When publishers upload an item, they select an availability period that spans from the current day to future dates. If a future date is selected, the item is set as "upcoming" and publishers can make changes to this item until the start date arrives. Once the start date is reached, the item is marked as "active" and becomes available for ordering. If the availability period ends, the item is marked as "deleted" and is no longer available for ordering. However, publishers have the option to restore it by selecting a new availability period.

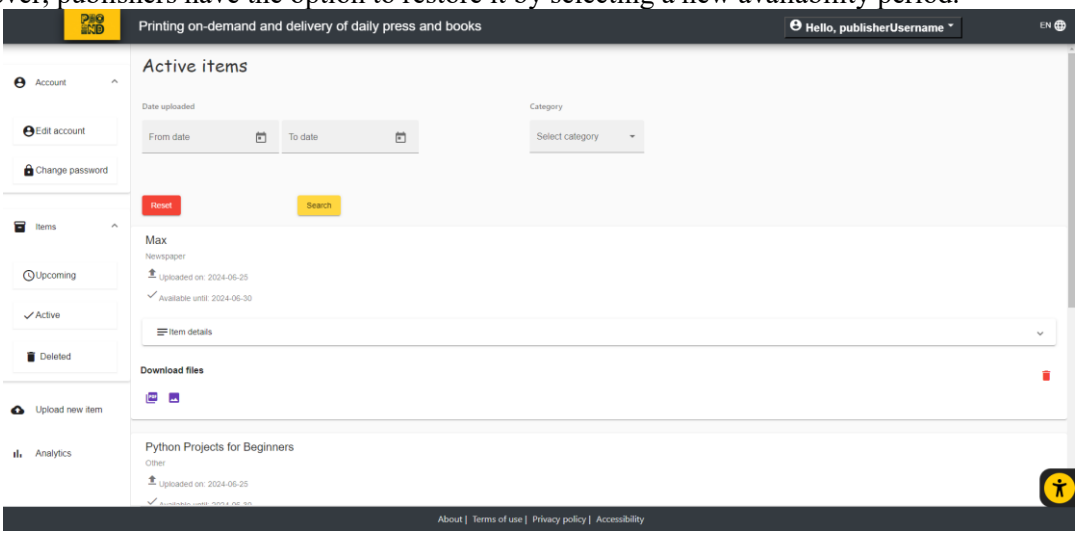

Figure 7. Publisher active items

Figure 8 illustrates the interface of the publishers' analytics page. This interface displays a donut graph that provides insights into the distribution of uploaded items across different categories: newspapers, magazines, and others. These categories are represented as percentages based on the total number of items uploaded. Additionally, two bar charts visualize data concerning the orders per date for the uploaded items and uploads a publisher has made per date. A dropdown menu is also featured, enabling publishers to select one of the uploaded items to view specific data related to that particular item.

Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

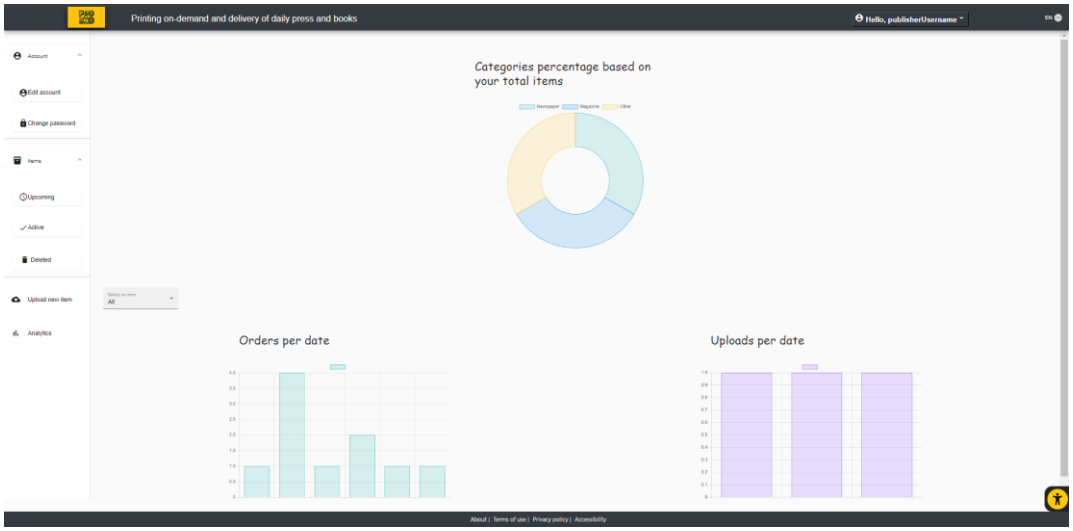

Figure 8. Publisher analytics

2.1.3. Printing Point. Printing point owners are offered an interface similar to that of the publishers but designed to their specific needs. As shown in Figure 9, there is a left side-menu containing options to edit account details and printing point details. The "Edit printing point details" functionality allows owners to change the name, email, and phone number of the printing point.

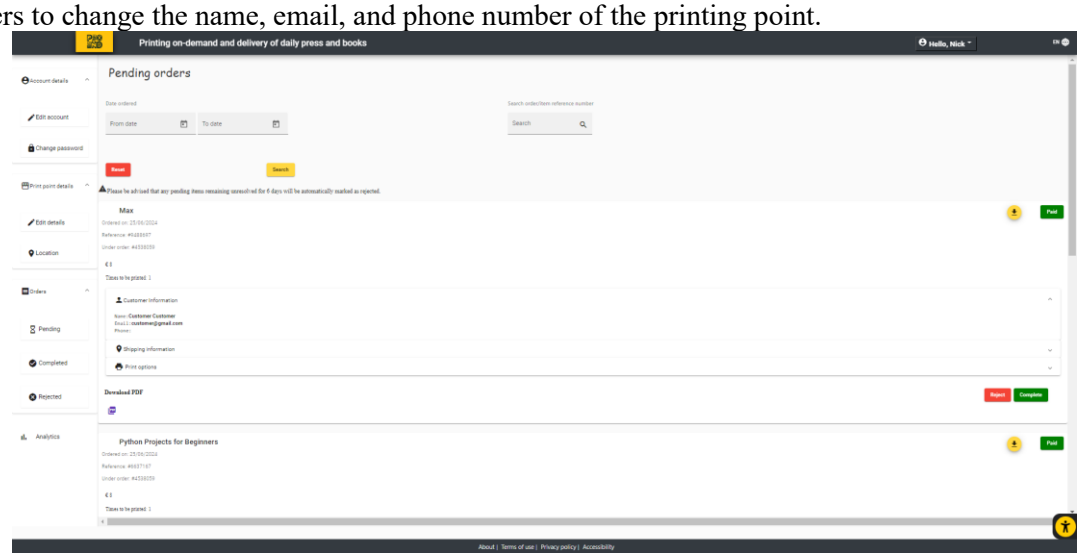

Figure 9. Printing point pending order

There is also the option to modify the current location of the printing point through a modal, where the owner must select a specific position on a map. PONDD provides customers with detailed information, including the exact location of the printing point on the map (an example is depicted in Figure 10). Orders are structured in dropdowns based on their status: pending, completed, and rejected. Within each category, the functionality to filter orders by order date or by order reference number is provided, enhancing viewing convenience and facilitating easier navigation. Depending on the type of the order status selected, the owners of printing points can reject or accept submitted orders and view details crucial for processing orders, such as printing options, shipping information, and customer details. There is also the functionality to download this information locally as a PDF file. Upon completing an order, the system prompts the owner to fill in the printing cost. After filing printing cost, the owner previews the content before sending it for printing and completing the order. If the owner chooses to reject the order, a reason for rejection must be provided, and the customer is

Technium Vol. 24, pp.11-25 (2024)
ISSN: 2668-778X
www.techniumscience.com

informed of this reason. The printing point has six days to update a "pending order"; if this period elapses, the order is automatically marked as rejected.

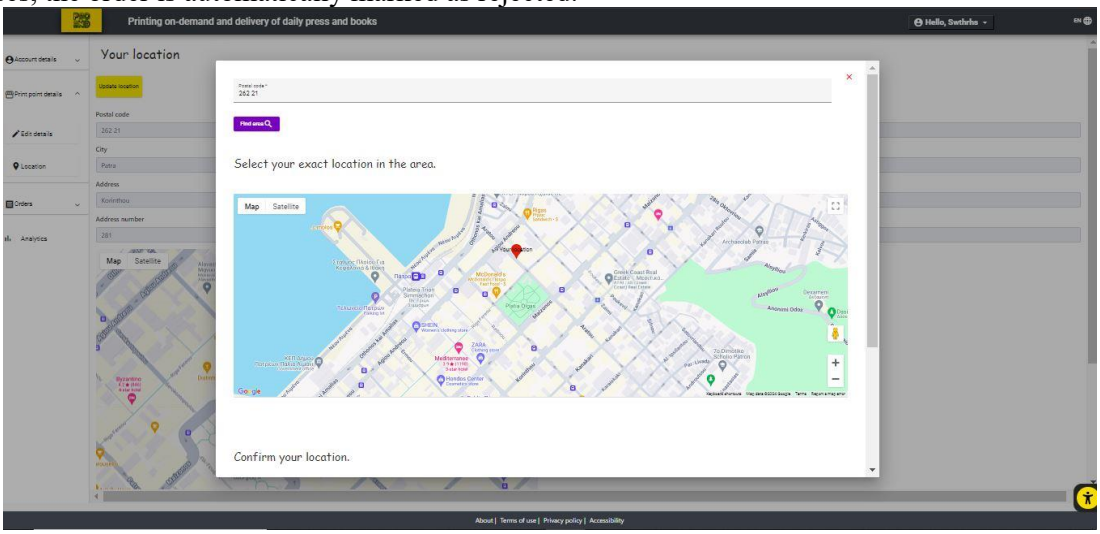

Figure 10. Edit printing point location

Similar to the interface for analytics available to “Publishers”, there are printing point analytics offering the option to view orders per month visualized via a bar chart and the percentage of item categories based on total orders visualized via a donut chart, as shown in Figure 11.

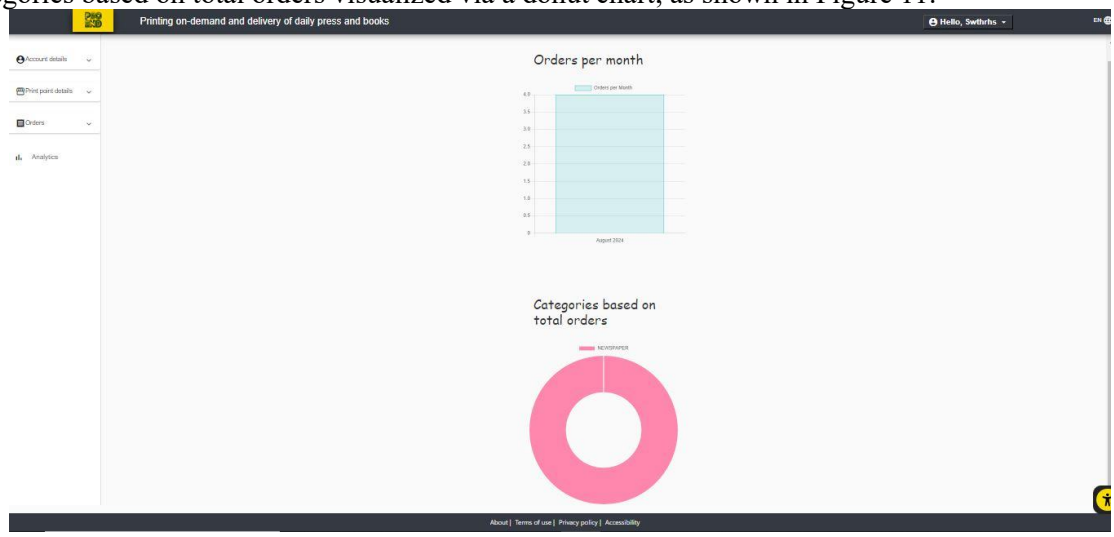

Figure 11. Printing point analytics

3. System implementation

In this section, we provide an overview of the technologies and software used for the development of the PONDD application, organized into three main parts. The first part outlines the high-level system architecture. Moving on, the next sections are split into two parts: one focusing on the technologies and software used in the frontend and the other for the backend, each explaining the choice of technologies and software as well as their contribution to a robust and efficient system.

3.1. System Architecture

The PONDD application is designed as a single-page application (SPA) [13], an architectural choice that enhances user experience by dynamically updating the webpage with new data from the web server, instead of the traditional method of reloading entire new pages. In an SPA, all necessary

HTML, JavaScript, and CSS code is either retrieved with a single page load or the appropriate resources are dynamically loaded and added to the page as necessary, usually in response to user actions. This results in a more fluid and faster application experience, as it minimizes disruptions due to page reloading. Angular [14] and Angular Material [15] are utilized to construct this SPA framework, providing a seamless and responsive interface that adapts to different devices and screen sizes. The backend architecture of the PONDD application employs Java [16] and the Spring Framework [17] to establish a robust and scalable infrastructure suitable for enterprise-level applications. Java provides the platform independence necessary for flexible deployment, while Spring Framework offers comprehensive support for modular architecture through dependency injection and aspect-oriented programming. Security measures are enforced using Spring Security [18], which integrates JSON Web Tokens (JWT) [19] to facilitate secure and stateless authentication across RESTful services. Data management is handled by integrating Hibernate ORM [20] with a PostgreSQL [21] database, ensuring efficient transaction processing and scalability. For external storage, the system utilizes AWS [22], optimizing resource allocation by offloading static content management. This architectural framework ensures that the backend can efficiently support current demands and easy scale on future growth. Figure 12 illustrates the high-level architecture of the PONDD application.

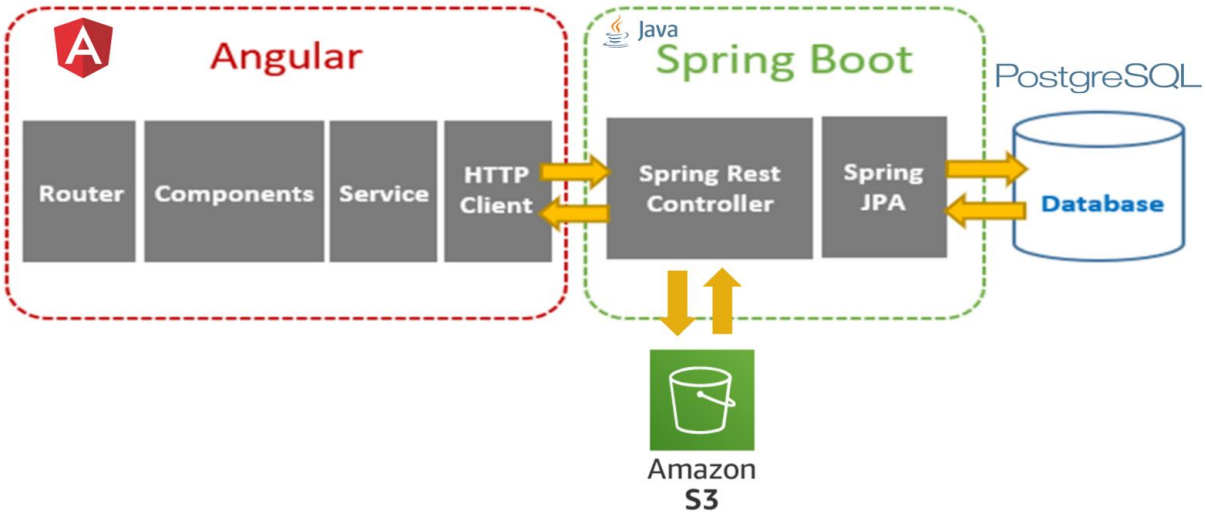

Figure 12. High-level architecture diagram

When a user requests a protected resource, a common scenario within this architecture, the behavior of the system is as follows. Upon successful authentication, the system generates a JSON Web Token (JWT) for the user. This token is crafted based on the user's role, the current time, and a secure private key that the system uses to ensure the token's integrity. The JWT serves as a credential for the user's session, encapsulating their identity and authorization level in a secure manner. When a request for a protected resource is made, the system first checks the provided JWT to verify if the user is authenticated, if the token has expired, and whether the user's role permits access to the requested resource. If the JWT is valid and the user is authorized, the system proceeds to send the data in JSON format, as communication occurs in a RESTful manner. If the JWT validation fails, the system methodically handles the error based on the specific reason for failure and informs the user accordingly. This mechanism is implemented across all parts of the application where the front end requires access to protected server resources, ensuring that only users with the appropriate roles can access certain data. For instance, pending orders from a specific printing point are only accessible to the respective printing point, reinforcing the secure and role-based access control throughout the app.

3.2. Frontend

The main goal in designing the frontend part was to make it very friendly and easy to use. Therefore, it was essential to design a very simple and clear user interface (UI) without fancy animations that might cause frustration to users. The focus was on creating an intuitive and easy-to-navigate experience, ensuring that all users, regardless of their tech-experience, could efficiently use the application. For this reason, we chose to use Angular and Angular Material, as they provide robust frameworks and components that are designed for simplicity and ease of use, ensuring a consistent and accessible user experience.

Angular [14] is a comprehensive web framework maintained by a dedicated team at Google, designed to empower developers to build fast, reliable applications. It provides a robust suite of tools, APIs, and libraries that simplify and streamline the development workflow, making it easier to create scalable applications. We chose Angular for its ability to handle complex tasks efficiently, its strong community support, and its capability to ensure our application can grow with both the size of our team and the complexity of our codebase. Agular's powerful features, such as two-way data binding, dependency injection, and a modular architecture, make it an ideal choice for developing a robust and maintainable application. Angular is the core framework that powers the entire frontend. Every aspect of the user interface, from routing to page rendering, is managed through Angular. The framework's two-way data binding facilitates seamless synchronization between the model and the view, ensuring that the user interface reflects real-time data changes without additional coding overhead.

Angular Material [15] offers high-quality, internationalized, and accessible components that are well-tested to ensure performance and reliability. Its straightforward APIs provide consistent cross-platform behavior, and it includes tools to help developers build custom components with common interaction patterns. Angular Material plays a crucial role in shaping the look and feel of PONDD frontend. We leverage its comprehensive library of components to build interactive and user-friendly forms, including sign-up forms, upload forms, and other input fields. The date pickers, toolbars, sidebars, and icons within the application are also crafted using Angular Material. Customizable within the bounds of the Material Design specification, Angular Material integrates seamlessly with Angular, making it easy to start from scratch or integrate into existing applications. This combination ensures a consistent, user-friendly, responsive and accessible interface for our application.

TypeScript [23], developed and maintained by Microsoft, enhances JavaScript with features suited for large-scale application development. As a superset of JavaScript, it introduces static typing and other advanced capabilities. Code written in this language is compiled into standard JavaScript, ensuring compatibility with any environment that supports JavaScript, whether in the browser or on the server-side. TypeScript is the foundational language used throughout the entire frontend development process. By offering features like type annotations, interfaces, and advanced tooling support, we can improve code quality, readability, and maintainability that are key factors for a stable and bug free application.

Several important libraries were used to enhance the functionality and usability of PONDD. The ngx-translate/core [24] library supports internationalization(i18n) and localization, making it easier to manage translations and serve users who speak different languages. In our application, this library is used to provide a seamless option for users to switch between two available languages—English and Greek—ensuring accessibility for a broader audience. The auth0/angular-jwt [25] library provides JSON Web Token (JWT) support for Angular, facilitating secure authentication and authorization processes within the application. In PONDD, this library is extensively used in a shared service across the entire application to extract and decode information from JWT tokens sent from the backend. This ensures that every request made by the frontend to the backend is secure and that user roles and permissions are correctly validated. The ng2-charts library integrates Chart.js [26] with Angular, allowing for easy creation and management of interactive charts, enhancing data visualization within the application. It is specifically utilized for the "Publisher" and "Printing point" analytics, where it helps visualize data and metrics through bar and donut graphs, providing users with clear insights. The angular/google-maps library available from angular material offers Angular

components for using Google Maps, enabling seamless integration and usage of maps within the application for location-based features. This library is used extensively throughout PONDD wherever map visualization is required. For example, it is featured prominently on the main page, where it displays the locations of printing points on a map and during the sign-up process for printing points, where users can select and verify their location. These libraries collectively help in building a powerful, user-friendly, and secure frontend for the PONDD application.

3.3. Backend

Given that our system is expected to store a significant amount of sensitive data, the primary objective in implementing the backend is to ensure security and data integrity. To achieve this, the backend is constructed using Java [16] and the Spring Framework [17]. Java was selected for its platform independence, performance, and comprehensive ecosystem that support enterprise-level applications. The Spring Framework enhances Java by providing a powerful and flexible framework for building web applications, offering extensive support for dependency injection, aspect-oriented programming, and transaction management. Additionally, Spring Security is integrated to maintain high standards of authentication and authorization, protecting user data and ensuring that only authorized users can access specific functionalities. To achieve this, we have implemented a JSON Web Token (JWT) based system for authorization, leveraging Spring Security.

JWT [19] provides a flexible and stateless way to verify user identities and secure API endpoints, making it an ideal choice for modern web applications. In our implementation, incoming requests are intercepted to extract and validate the JWT from the Authorization header. If the token is valid, the user's details are retrieved, and the user is authenticated, allowing access to protected resources. The backend configuration is designed to support secure user authentication, with session management set to stateless, and necessary security measures like disabling CSRF and enabling CORS. This JWT-based approach ensures that sensitive data is protected, allowing only authorized users to access the system's resources. By integrating JWT with Spring Security, we have created a robust, scalable solution with high standards of security and data integrity for the PONDD application.

Spring [17], a comprehensive framework for building enterprise-level Java applications, was chosen for the backend development of the PONDD application due to its robust and flexible architecture. Spring offers essential features such as dependency injection, aspect-oriented programming, and seamless integration with various technologies. These features enable the creation of modular applications consisting of loosely coupled components, facilitating easier maintenance and scalability. Additionally, Spring provides built-in support for common tasks such as data binding, type conversion, validation, exception handling, and internationalization, streamlining the development process and enhancing code maintainability.

Spring Security [18], a core component of the Spring ecosystem, was integrated to address our critical need for secure authentication and authorization. It offers out-of-the-box support for protecting applications against common security threats and provides flexible and customizable security configurations. Spring Security is used to safeguard controllers from unauthorized access by implementing built-in methods to enforce business and logic rules, validate data, and ensure that only users with the appropriate roles can perform specific operations. Additionally, we configured security settings to intercept and verify every request before it reaches PONDD controllers, providing a robust layer of protection that checks for proper authentication and authorization at the application's entry point. By leveraging Spring Security, we ensured that sensitive data is safeguarded and that access to resources is restricted to authorized users only.

Hibernate [20] is an open-source Object-Relational Persistence and Query service for any Java Application. Hibernate was utilized for its powerful and flexible Object-Relational Mapping (ORM) capabilities, which seamlessly integrate with Spring. This combination facilitated efficient data management and interaction with our database. In the PONDD application, Hibernate was used to map Java objects to database tables, allowing for smooth and efficient CRUD (Create, Read, Update,

Delete) operations. For example, Hibernate was employed to manage entities such as “Users”, “Orders” and “Printing points”, automatically handling the conversion between the application's data models and the underlying database schema. The use of Hibernate and Spring not only facilitated rapid and efficient development but also provided a scalable and secure foundation, crucial for the integrity and reliability of the PONDD application.

PostgreSQL [21] is a powerful, open-source object-relational database management system known for its robustness, scalability, and support for advanced data types. We used PostgreSQL for the PONDD application due to its seamless integration with Java and the Spring Framework. PostgreSQL's compatibility with Hibernate, the ORM tool we use with Spring, facilitates smooth and efficient data management. This integration simplifies the development process by allowing us to leverage Spring Data JPA, which provides repository support, transaction management, and query derivation mechanisms. Throughout PONDD, PostgreSQL serves as the primary database and the backbone for storing and managing essential data, such as user accounts, orders, and printing point details. By leveraging PostgreSQL, we ensured that the PONDD application has a reliable, scalable, and efficient database solution that aligns perfectly with our Java and Spring-based technology stack.

In the PONDD application, we utilize AWS S3 buckets [27] for storing PDF files, preview images, and other uploaded content. Initially, we faced several challenges with storing these files directly in the database. Storing large files in a relational database can lead to performance issues, such as increased load times and database bloat, which in turn can degrade overall system performance and complicate backup and recovery processes. Additionally, managing file storage within the database can be complex and inefficient, as it requires handling file retrieval, storage, and scaling within the same system designed primarily for structured data. To overcome these challenges, we opted for AWS S3, a scalable and reliable storage service. S3 provides many advantages, including high availability, effortless integration with Spring applications, and ease of access. By offloading file storage to S3, we streamlined our database, ensuring it remains optimized for transactional operations and data integrity. However, storing files in the cloud introduces potential concerns regarding data protection and privacy. To address these concerns, we implemented several security measures. First, we utilized AWS's built-in encryption mechanisms [28] to ensure that all data stored in S3 buckets is encrypted both at rest and in transit. This encryption safeguards the data against unauthorized access. Additionally, we configured strict access control policies using AWS Identity and Access Management (IAM) [27] to restrict access to the S3 buckets, ensuring that only authorized services and users can interact with the stored files. Finally, we employed logging and monitoring tools to track access and modifications to the S3 resources, allowing us to detect and respond to any potential security incidents promptly. Through these measures, we ensured that the use of AWS S3 in the PONDD application aligns with best practices for data protection, eliminating the risks associated with cloud-based storage.

4. Concluding remarks and future plans

We designed and developed an innovative online service – PONDD - for printing-on-demand and delivery of daily press, magazines, and books that directly connects readers and publishers via printing points. The PONDD platform revolutionizes print media distribution by addressing traditional challenges such as overproduction and inefficiency through on-demand printing technology. By directly connecting readers with local printing points, PONDD ensures that newspapers, magazines, and books are printed only when needed, significantly reducing waste and logistical complexities. The platform not only enhances distribution efficiency but also supports timely content delivery, especially in remote areas, maintaining the viability of print media in the digital era. It also offers a sustainable business model for small kiosks and mini-markets, converting them into essential nodes of the distribution network without the financial risk of unsold stock. Furthermore, PONDD eliminates potential consequences usually arising in monopoly situations regarding the distribution of daily press thus offering the opportunity to publishers to directly and freely serve their readers. Last by not least, even though PONDD is a digital service, it actually supports the well-established tradition maintained

for generations of people reading printed newspapers. In our current internet-based high-speed era where people (are supposed to) get informed by quickly scrolling over headlines on a smartphone or tablet, a printing on-demand service for daily press like PONDD can play an important socio-cultural role contributing to the preservation of the traditional way people getting informed via reading a real newspaper (not a digital one), an option still preferred by a not negligible number of individuals. Looking forward, PONDD may expand geographically, integrate a broader range of print formats, print preferences, and introduce enhanced user customization. Future developments could include data analytics to anticipate user preferences and AI to optimize distribution, thus exemplifying how digital innovation can rejuvenate traditional industries.

References

[1] "The Environmental Impact of Printing and how to Reduce It » Latest Printing News." Accessed: Jul. 16, 2024. [Online]. Available: https://www.latestprintingnews.com/news/the-environmental-impact-of-printing-and-how-to-reduce-it/

[2] "How Environmentally Friendly is Digital Printing? 4 Reasons - VSL Print - NYC Printing Company." Accessed: Jul. 16, 2024. [Online]. Available: https://www.vslprint.com/printing-nyc/digital-printing/how-environmentally-friendly-is-digital-printing-4-reasons/

[3] "Digital vs. Traditional Printing: Exploring the Advantages and Drawbacks of Contemporary Printing Methods. - SEPS." Accessed: Jul. 16, 2024. [Online]. Available: https://sepsgraphics.com/digital-vs-traditional-printing-exploring-the-advantages-and-drawbacks-of-contemporary-printing-methods/

[4] "Print Workflow: How To Manage And Optimize." Accessed: Jul. 16, 2024. [Online]. Available: https://www.gelato.com/in/connect/blog/print-workflow

[5] Layton J. and Wilson Bob, "How Newspapers Work | HowStuffWorks.", 1 April 2000. HowStuffWorks.com https://people.howstuffworks.com/newspaper.htm 24 August 2024

[6] A. Schiffrin. (Ed.) In the Service of Power: Media Capture and the Threat To Democracy. National Endowment for Democracy, 2017

[7] Hellenic Competition Commission (HCC). Market Investigation in the Press Distribution Sector. First HCC report (March 23, 2020), Second HCC report (7 October 2021)

[8] G. Marion The "free Press", Portrait of a monopoly. New Century Publishers: New York, 1964. The Project Gutenberg eBook: [eBook #70107], February 22, 2023. https://www.gutenberg.org/files/70107/70107-h/70107-h.htm

[9] A. A. Fuentes. Cuiñas and P. A. Vailati. "The Smell of Printed Books: A Qualitative Comparison between Printed and Digital Formats," American Journal of Qualitative Research, vol. 6, no. 1, pp. 214–225, 2022.

[10] J. J. Stiegler-Balfour, Z. S. Roberts, A. S. LaChance, A. M. Sahouria, and E. D. Newborough, "Is reading under print and digital conditions really equivalent? Differences in reading and recall of expository text for higher and lower ability comprehenders," International Journal of Human Computer Studies, vol. 176, 103036, 2023

[11] N. Bresó-Grancha, M. J. Jorques-Infante, and C. Moret-Tatay, "Reading digital- versus print-easy texts: a study with university students who prefer digital sources,"

Psicologia: Reflexao e Critica, vol. 35, article number: 10, 2022.

[12] S. Cecily, "Role of agents and subagents in the distribution of newspapers: A functional analysis," International Journal of Advanced Academic Studies, vol. 4(1), pp. 181–184, 2022.

[13] "SPA (Single-page application) - MDN Web Docs Glossary: Definitions of Web-related terms | MDN." Accessed: Jul. 23, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/SPA

[14] "Home • Angular." Accessed: Jul. 16, 2024. [Online]. Available: https://angular.dev/

[15] "Components | Angular Material." Accessed: Jul. 16, 2024. [Online]. Available: https://material.angular.io/components/categories

[16] "What is Java?" Accessed: Aug. 28, 2024. [Online]. Available: https://www.java.com/en/download/help/whatis_java.html

[17] "Spring Boot." Accessed: Jul. 16, 2024. [Online]. Available: https://spring.io/projects/spring-boot

[18] "Spring Security." Accessed: Jul. 16, 2024. [Online]. Available: https://spring.io/projects/spring-security

[19] "JSON Web Token Introduction - jwt.io." Accessed: Jul. 24, 2024. [Online]. Available: https://jwt.io/introduction

[20] "Your relational data. Objectively. - Hibernate ORM." Accessed: Jul. 23, 2024. [Online]. Available: https://hibernate.org/orm/

[21] "PostgreSQL: About." Accessed: Jul. 16, 2024. [Online]. Available: https://www.postgresql.org/about/

[22] "What is Amazon S3? - Amazon Simple Storage Service." Accessed: Jul. 16, 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

[23] "TypeScript: The starting point for learning TypeScript." Accessed: Jul. 23, 2024. [Online]. Available: https://www.typescriptlang.org/docs/

[24] "GitHub - ngx-translate/core: The internationalization (i18n) library for Angular." Accessed: Jul. 23, 2024. [Online]. Available: https://github.com/ngx-translate/core

[25] "GitHub - auth0/angular2-jwt: Helper library for handling JWTs in Angular apps." Accessed: Jul. 23, 2024. [Online]. Available: https://github.com/auth0/angular2-jwt

[26] "Chart.js · GitHub." Accessed: Jul. 23, 2024. [Online]. Available: https://github.com/chartjs

[27] "Identity and Access Management for Amazon S3 - Amazon Simple Storage Service." Accessed: Aug. 26, 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-iam.html

[28] "Protecting data with encryption - Amazon Simple Storage Service." Accessed: Aug. 26, 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingEncryption.html