FYP Interim Report

# Blockchain-based Ebook Transaction System for Transparent Book Sales

By
CHAN Hoi Chun

Advised by
Prof. Qian Chenxiong

FYP24045
COMP4801 Final Year Project 2024-2025
Department of Computer Science
The University of Hong Kong

Date of submission: January 26, 2025

# Abstract

The lack of transparent transactions in current ebook transaction system provides a convenient environment for fraud from ebook platforms that inflict loss of income on the authors. Blockchain technology and InterPlanetary File System (IPFS) are introduced to the proposed transaction system in this project to provide necessary transactions and records to authors so that they can monitor and verify the flow of funds to prevent fraud or missing income. Both ebook transaction and subscription services with an individual-pay-per-page model are adopted in the system which matches the current ebook sales model. The system also supports refunds and ebook price changes. The whole system including blockchain, backend server, database, and mobile app will be implemented and tested to show the idea is implementable with Ethereum blockchain and IPFS. Finally, the characteristics of the system will be concluded in the interim report.

# Content

# 1. Introduction

## 1.1. Background

Ebooks provide an alternative path for reading with the advantages of convenience, searchability, sale prices, and environmental friendliness (Staiger, 2012). The ebook market is experiencing rapid growth. According to Errera (2023), the number of eBook sales in the United States rose from 69 million to 191 million between 2010 and 2020. Moreover, it is estimated that there will be more than 1.2 billion ebook users in 2027 globally. The process of ebook transactions mainly involves three parties. Authors and publishers, own their books and publish them on online ebook platforms. Platforms, such as Amazon Kindle Direct Publishing(KDP), Google Play Books, and Apple Books, provide software support for selling, purchasing, and reading ebooks. Readers purchase and read ebooks on ebook platforms.

There are differences between typical ebook transactions and subscription services. In typical ebook transactions, readers purchase the desired ebooks and pay the selling price for each ebook, then receive the right to read the ebooks permanently. Besides, subscription services are proposed for ebook transactions (Polanka, 2013). Readers can subscribe to the services monthly, and then freely access a large number of specified ebooks within the period of subscription. For example, Kindle Unlimited in KDP is one of the ebook subscription schemes that provides more than 4 million ebooks and allows subscribers to access 20 ebooks simultaneously (Chen, 2023). Huang et al. (2022) mentioned that subscription services produce a stable and long-term income for ebook platforms because they cultivate a group of subscribers who renew their subscriptions. Readers who have reading habits may also prefer to subscribe to the service because they can read huge amounts of ebooks at a relatively low price. However, the subscription services may not be friendly to authors. Authors may gain a lower income compared with typical ebook transactions (Larson, 2022). However, the distribution scheme to allocate the income to different may not be fair due to the different types and contents of ebooks (Shannon, 2015).

## 1.2. Motivation

The general typical ebook transaction process is illustrated in Figure 1. After the author publishes an ebook on the platform and readers buy the ebook, the royalties are remitted to the author. The statistics about the book such as the sales quantity can be obtained from the

platform. However, it is difficult to verify the royalties because of the lack of transparency in the transaction and verifiable statistics. Authors are unable to access the transaction records between readers and the platform, thereby lacking a reliable and trustable method to ascertain the existence of any missing royalties or fraud from the platform. The transaction process without any sufficient supervision from third parties provides an opportunity for fraud led by the platform so that the platform gains extra illegal profit. As a result, authors would experience an unfriendly transaction model with the risk of royalties loss (Nizamuddin et al., 2019).

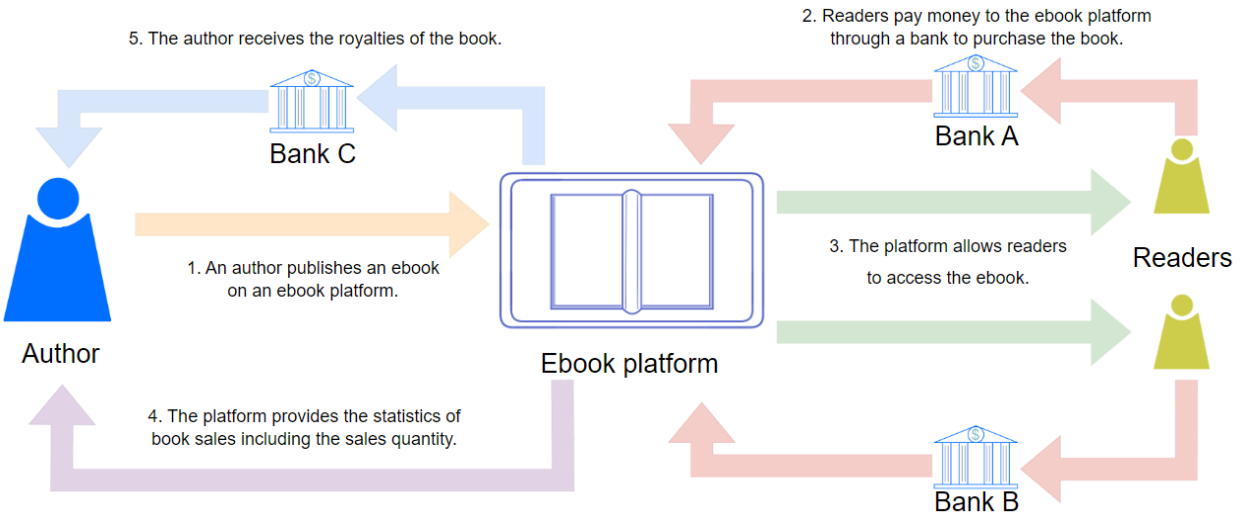


**Figure 1.** The general typical ebook transaction process

## 1.3. Proposed Solution

Blockchain technology is integrated into the project to provide authors with a reliable and trustworthy verification method for validating their profits. The lack of accessibility of transaction details in ebook trading is the main problem in royalty validation. Blockchain is a decentralized digital ledger technology that records transactions across multiple computers in a way that ensures the security, transparency, and integrity of the data. Transactions can be recorded on the blockchain directly and be admitted simultaneously. Compared with the platform to publicize the records of ebook transactions actively for royalty validation, completing the transactions on blockchain is more reliable. If the payment is completed on the blockchain using cryptocurrency, the record of the payment is transparent and immutable, so authors do not need to worry about the platform hiding or modifying the payment record and causing unreliable validation. The payment of refunds can also be finished on blockchain so that authors can validate the final

sales of ebooks. Blockchain smart contracts can be used to support multipurpose transactions for the proposed system. The idea of blockchain is not only for simple transactions that record the sender, receiver, and payment, but also smart contracts. For example, when readers use a smart contract to purchase an ebook, they can specify which ebook they buy in the smart contract. It helps authors to validate their profits.

## 1.4. Objectives and Deliverables

In this project, a transparent ebook transaction system will be constructed with the integration of blockchain technology. The system supports both typical transactions and subscriptions to fit the requirements of the current ebook sale model. The system also supports the refund function and allows authors to adjust the prices of ebooks. The whole system will be implemented including smart contracts, a mobile application, and a backend server. The reliability of the system will be analyzed, and the limitation on illegal behaviors from different parties will be illustrated.

The deliverables involved smart contracts, mobile application development, and backend development. Smart contracts for the blockchain application platform Ethereum will be implemented in this project. An Andriod mobile application will be developed to provide the interface for readers and authors selling, buying, and reading ebooks. A backend server of an ebook platform will be developed to support the services for readers and authors, for example, the server needs to store the data about users and ebooks.

The remaining sections of this interim report are outlined as follows. In Section 2, the related work is mentioned to provide a review of current research relevant to the project topic. Section 3 shows the details of the system and implementation. Lastly, the conclusion is mentioned in section 4.

# 2. Literature Review

Chi et al. (2020) designed a blockchain-based ebook transaction system for self-published ebook trading. The secure and reliable system is designed for self-published trading, in other words, the trade only involves readers and authors. As the system does not involve any third parties, authors can take all the profits and not need to validate their profits. On the other hand, piracy activities are not easy to be forbidden as no third parties can control the trading. Moreover, a repository is needed to store the encrypted ebook content and a service application is required to support the system services. Authors may need to pay the extra cost for maintaining the services.

Nizamuddin et al. (2019) also designed a blockchain-based framework for ebook transaction systems to guarantee authors receive expected royalties using smart contracts. The framework allows refunds if the purchase fails. They completed the implementation of the smart contracts using Remix IDE with the programming language Solidity. It was tested and analyzed against recognized attacks and vulnerabilities.

There may be a lack of research related to subscription ebook systems with blockchain. It is noted that both two systems are designed for typical ebook transactions but not subscription ebook transactions.

Documents can be stored in the decentralized storage system, InterPlanetary File System (IPFS) instead of the blockchain because it is costly to store data in the blockchain. The file in the IPFS is also immutable as the file is stored in the decentralized network and can be located by the hash of the file as the content identifier (CID). It provides a faster and cheaper method to store and access files and still contains the properties of immutable and public access (Sarode & Bhalla, 2022).

# 3. Methodology

## 3.1. System Overview

The high-level system overview will be illustrated in the section. The system mainly involves the communication between the mobile app, the backend server of the platform, IPFS, and Ethereum blockchain. The interactions between each entity are shown in Figure 2. Readers and Authors can complete most of the system services, such as selling, buying, subscribing, and reading ebooks. The platform provides the statistics related to them through the app. However, if they want to access the data stored on the IPFS and the blockchain network for profit validation, such as the transaction records, they should access the data without using the app to avoid the ebook platform modifying the data and defrauding the readers and authors. The data access can be finished through trustworthy third-party application platforms so that they can conveniently get the reliable data without holding the entire blockchain and the files. Infura which is a blockchain infrastructure service can be used for interacting with Ethereum and IPFS networks. In this project, the app and the platform server use Infura API to interact with the networks, so the server and the app do not need to store the heavy data. When readers want to purchase or subscribe to ebooks, they are required to sign to create payment contracts. The app will help them deploy the creation contents and signature to the Ethereum network, and also notify the platform at the same time. The server will keep listening to the events on the blockchain. It will be captured once the purchase or subscription contracts are created. The server also needs to share reading data on IPFS in order to support the validation of subscription services. The details of subscription services will be mentioned later.

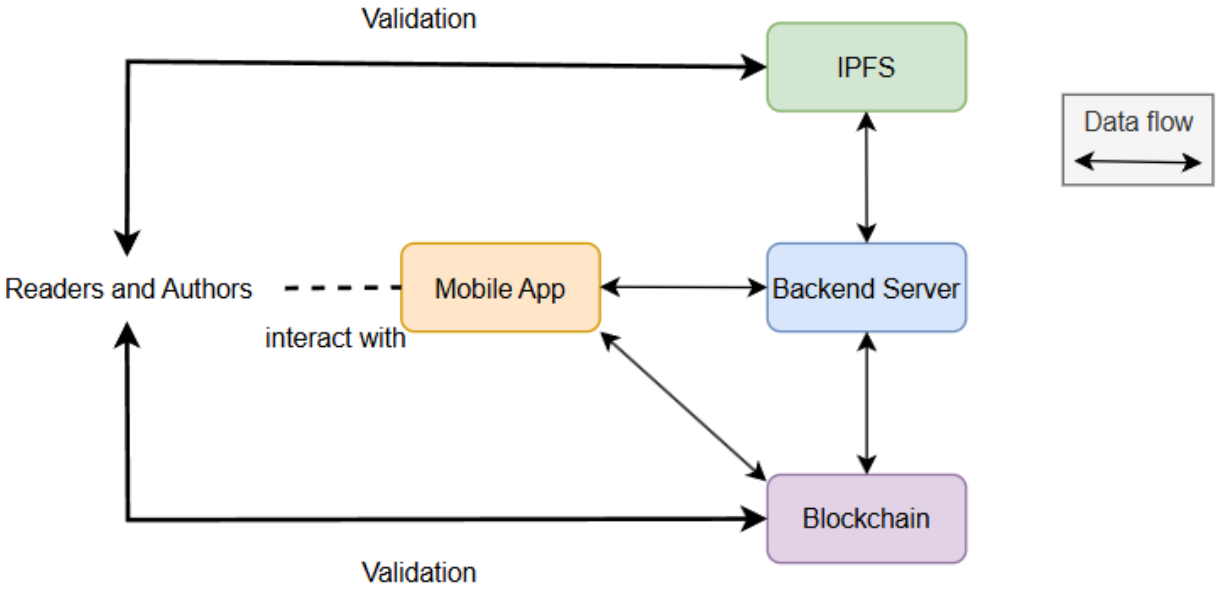


**Figure 2.** Data flows of the system

### 3.2. Process of Purchasing Ebook

The key steps of completing an ebook purchase and allowing readers to read the ebook are shown (Figure 3) and described in the following. Different parties must complete the required steps to prepare for ebook transactions. Platform deploys the framework of purchase contracts and the factory contract which allows readers to create purchase contracts with the framework by calling its function. Authors and readers should understand and agree on the framework and factory contract before it is put into use. The invariable content of the transaction can be deployed and confirmed before any purchases are created, such as the distribution ratio of the income between authors and the platform, so that readers do not need to verify the entire contract for each purchase. After that, readers only need to validate a few variables to create the contract for each time they want to purchase ebooks, such as the payment, ebook ID, and the address of the author account.

After the platform deploys the framework and the contract on blockchain network and is admitted by authors and readers. They can be used for creating feather transactions. When authors publish ebooks on the platform, they need to submit a copy of the ebooks and also their Ethereum account address. Once a reader wants to buy an ebook and clicks the purchase button, the app will generate the transaction for the reader to sign. More specifically, a

transaction is signed for calling a function of the factory contract with few arguments to create a purchase contract as well as depositing the cryptocurrency into the contract from the reader account as a payment. After the reader verifies the transaction and signs it with the private key of the reader, the app will help the reader deploy the transaction to the network and notify the platform server. They need to wait for the transaction added to the blockchain. The platform will keep track of the transaction. Once the platform knows the valid purchase contract has been created successfully, the platform will notify the reader through the app and allow the reader to read the ebook.

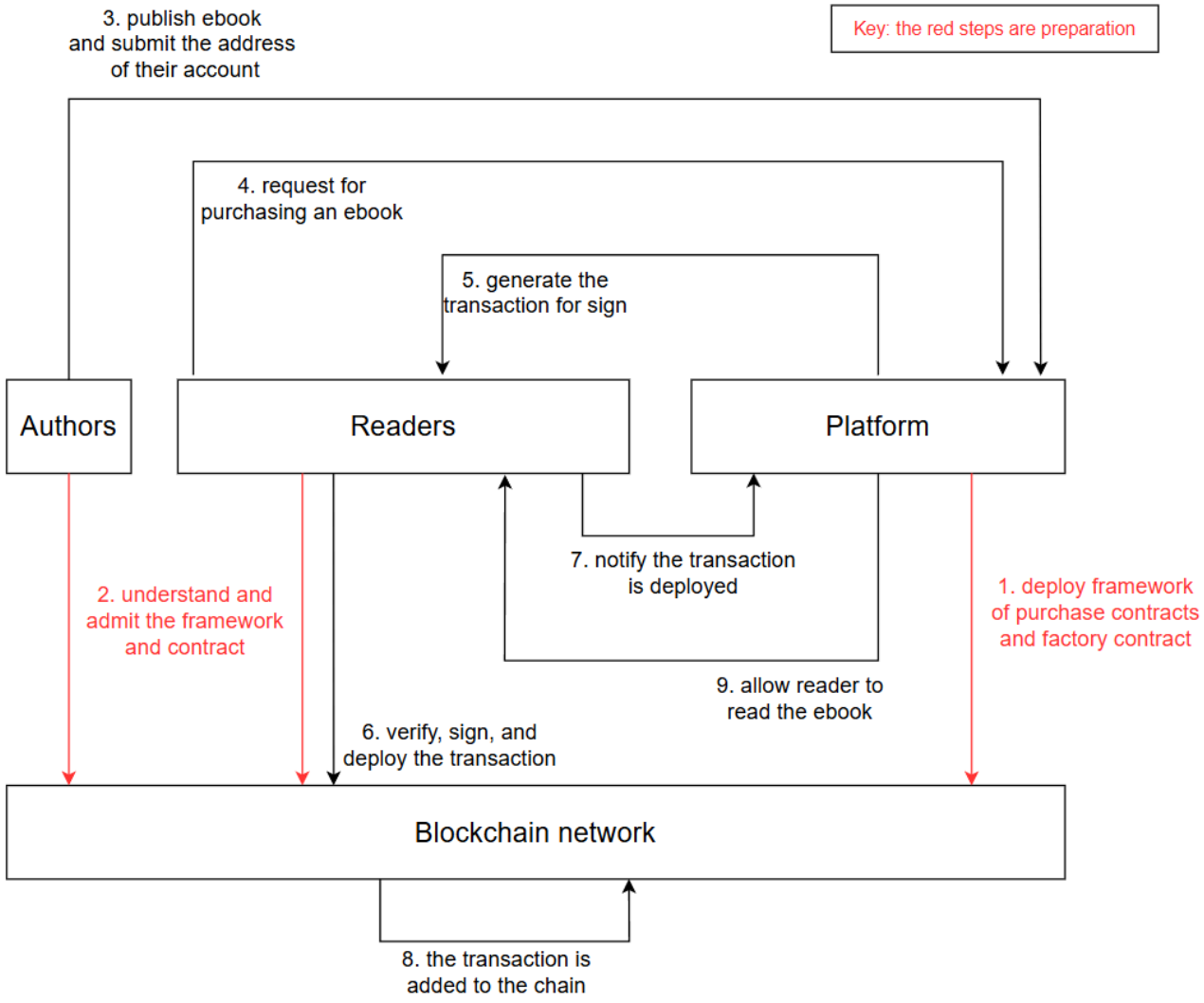


**Figure 3.** The main process for purchasing ebooks

### 3.3. Smart Contract Design for Ebook Purchase

As the code of the smart contract is immutable, we can design a purchase contract that limits the platform interaction with the payment stored on the contract. It guarantees that authors can receive the correct amount of royalty if purchase contracts are created correctly.

Figure 4 shows the timeline of a contract starting from the contract that was just created. When the contract is created, a refund is allowed. The platform can call a **refund()** function of the contract to return all funds to the reader account. Since the contract cannot control the cryptocurrency after transfer to another account, we do not want the platform to distribute the cryptocurrency immediately and the reader has no chance to apply for a refund. Therefore, in the implementation, readers get 30 buffer days for them to apply for refunds through the app. After that, the platform is allowed to distribute the funds to the author and its account with the fixed distribution ratio, but the platform is still allowed to perform a refund in case it needs more days to examine the application. However, it is not enough to ensure that the author receives the royalty. In real applications including the Ethereum network, if someone wants to create or modify something on the blockchain, they need to pay a fee to the miner. It is called gas fee in Ethereum. To avoid the funds being locked in the contract because the platform refuses to distribute the funds due to the high gas fee, a function can be called to penalize the platform. The **penalty()** function can be called by the author after 120 days of the creation, and all the funds will be transferred to the author account. At the same time, the platform cannot perform a refund and a distribution.

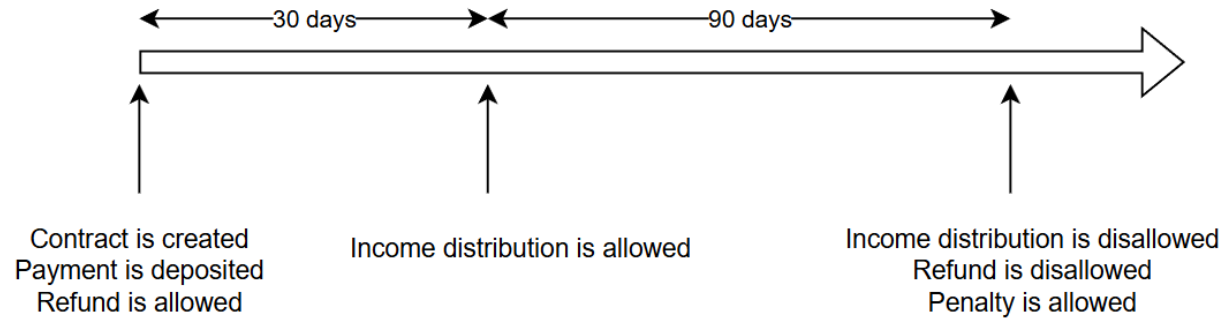


**Figure 4**. The timeline of the purchase contract limit to the platform

Figure 5 is the state diagram of purchase contracts. Each state also represents the state of the funds. The funds are in contracts if the contracts are “Created”; the funds are in the accounts of

readers if the contracts are “Refunded”; the funds are in the platform and author accounts if the contracts are “Distributed”; the funds are in the author accounts if the contracts are in “Penalized” state.

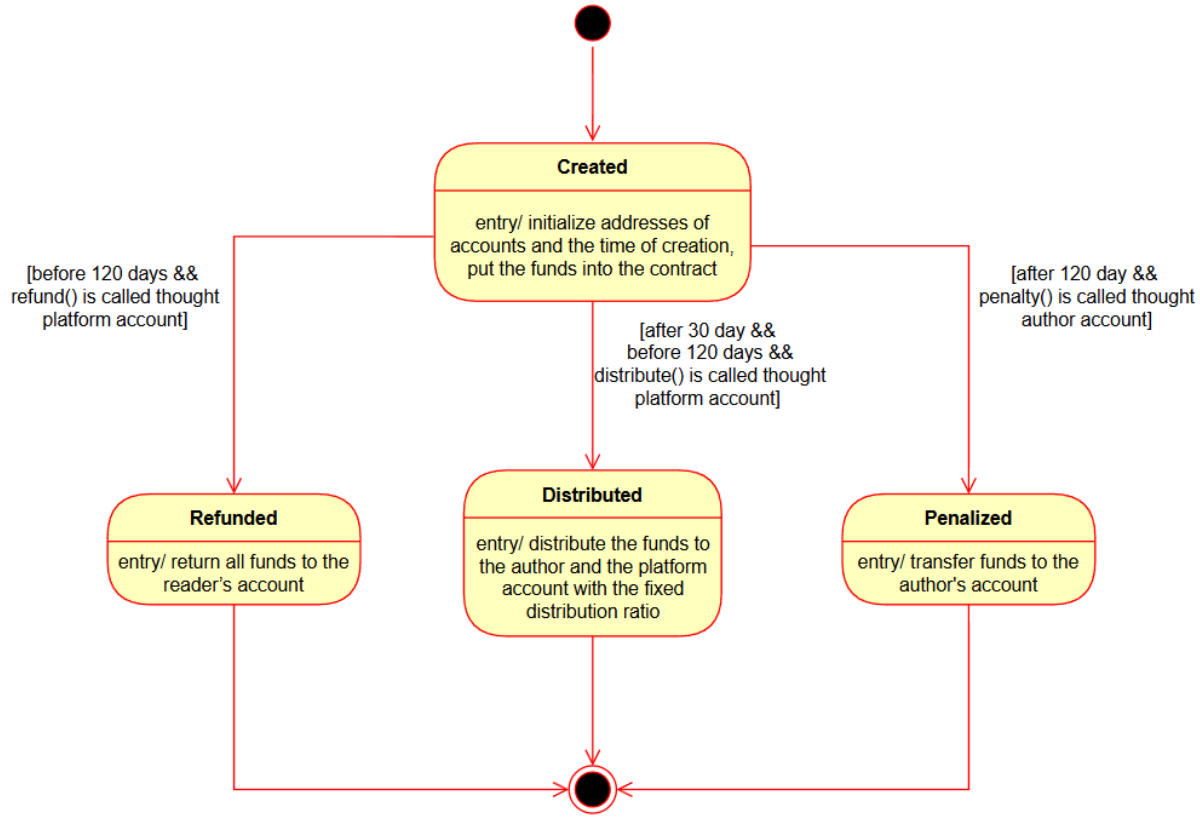


**Figure 5**. The state diagram of purchase contracts

Before the factory contract creates a purchase contract, the factory will check the deadline for the creation which should be given when deploying the transaction. Due to the mechanism of blockchain, after readers deploy the transaction to the network, they do not the time the transaction will be added to the blockchain. As the execution time is unpredictable, the contract may be created after days and weeks. It may cause a poor purchase experience for readers. To prevent the contract created after a long time and wasting gas fees for creating unwanted purchases, readers can set a deadline for the creation. The factory contract will not create an expired contract. The idea of the creation deadline also makes dynamic prices available. Authors can schedule new prices for their ebooks. The app will not allow readers to deploy an invalid transaction by setting the old price with the wrong deadline.

### 3.4. Royalty Validation in Ebook Purchase

The purchase contracts ensure authors receive the correct amount of royalties if the contracts are created correctly. In validation, authors may keep track of the contracts if the platform distributes the profits on time. If not, authors can take all the funds by calling the penalty function. There are some arguments readers need to verify before signing the transaction, including payment, book ID, creation deadline, and the address of author account. However, readers are not able to validate the address of author account because they cannot determine which is the correct address. Those arguments are generated by the platform and are shown to the readers through the app. Therefore, the platform can create and provide a fake address to readers, then the platform can scam all the funds in general. To avoid scamming, authors are required to keep monitoring the app to ensure that it always provides the correct author address for readers to sign.

### 3.5. Individual-pay-per-page Model

To support royalty validation in subscription services, it is important to know how the income is distributed. As we mentioned, readers can freely access a large number of specified ebooks when they subscribe monthly. One of the possible distribution models is pay-per-page model. For example, Kindle Unlimited in KDP applies the pay-per-page model as the distribution strategy (Shannon, 2015). For each month, the revenue and the number of pages read will be summed up respectively. After the platform takes a portion of the total revenue, the income belonging to a single page is counted, which is equal to the remaining revenue divided by the total number of pages read.

In the project, individual-pay-per-page (IPPP) model is used instead of pay-per-page model because it provides stronger security to avoid the platform scam the authors with the system design. The reason will be mentioned later. Instead of summing up the revenue before calculating the value of one page, IPPP counts the value of pages for each subscriber. The values can be different because they depend on how many pages subscribers read. If a reader subscribes to the service a month using 600 wei. Assuming the platform takes half of the revenue, the remaining revenue (300 wei) will be assigned to the authors. If the reader reads 200 pages in the preroid including 100 pages of book X, the author of book X can gain 150 wei. Therefore, if authors want to validate their royalties, they need to know the payment and the number of pages read for each ebook for all transactions.

### 3.6. Smart contracts and IPFS for Ebook Subscription

The design of contracts for subscription services is relatively simple. Since the reading behavior of the subscribers is unpredictable, it is unable to design a subscription contract that limits the platform distributing the funds with a fixed ratio and provides a fair penalty function for readers. Therefore, the subscription contract transfers the funds directly to the platform once the contract is created successfully.

Instead of deploying the reading records on blockchain for validation, they are uploaded to IPFS. Each reading record contains a hash of the record ID using SHA-256 which is the user ID concatenated a nonce. The nonce is a random number generated by the platform. It will be sent to the subscriber after the record is uploaded. Only the subscriber knows which record belongs to the subscriber. The user ID is also required in hashing to avoid the platform reusing the record for different subscribers. The record IDs are encrypted before the public on IPFS to prevent others know what books they read. The records also contain the payment, the ebook they read, the number of pages they read for each book, and the author account addresses.

### 3.7. Royalty Validation in Ebook Subscription

Authors need to validate the reading records and the payment for each transaction. However, only the platform and the readers know the real reading records. Authors cannot validate the reading records by themselves even if the records are deployed to blockchain or IPFS. Therefore, the participation of subscribers is necessary to achieve reliable royalty validation. Authors validate their royalties by assuming the reading records and the payment are correct. Readers examine their reading records. To provide the correct author account addresses for readers to check, the authors need to ensure the app shows the correct addresses with their ebooks.

The individual-pay-per-page (IPPP) model is used instead of pay-per-page model because the platform can earn illegal profits by creating fake ebooks, subscription contracts, and reading records. The platform can create large amounts of subscription contracts and claim that they read a lot of pages of the fake ebooks in the reading records. It is possible to produce profits for the platform if the average of pages readers read is relatively low. Readers can do a similar scam in IPPP by publishing an ebook with garbage content and adding the book to the ebook subscription scheme. They can get a “discount” by reading their books. Therefore, platform is suggested to set up a threshold for ebooks to join the scheme in terms of sales in purchasing.

# 4. Project schedule and completed works

An overview of the project schedule is shown in Figure 6. The grids in the table represent roughly 7-8 days each month. The initiation including tools exploration and system design is finished before October. The prototyping of mobile applications and smart contract implementation are also accomplished. We are working on the implementation part of the server and the app. It is planned to be completed before the middle of February. The testing of the system will be done after the implementation.

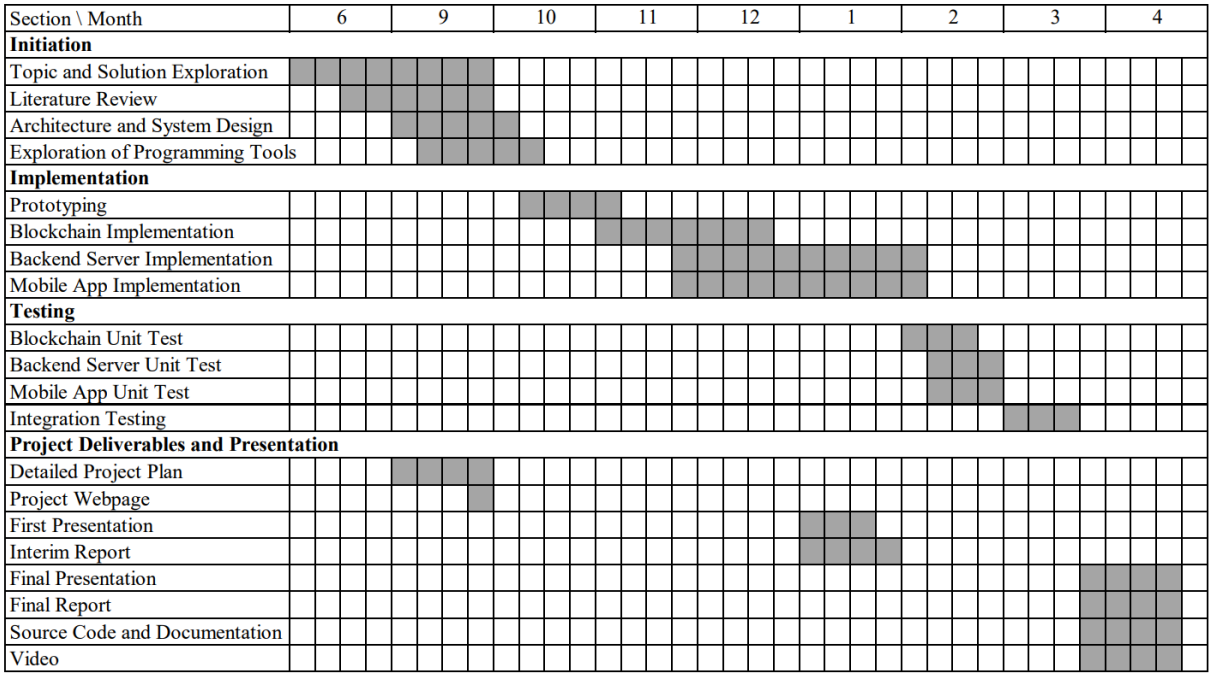


**Figure 6**. The schedule of the project

# 5. Conclusion

The development of a transparent ebook transaction system leveraging blockchain technology and the InterPlanetary File System (IPFS) is outlined in this report. The proposed system addresses the issues of fraud and income loss faced by authors in the current ebook market by providing a reliable mechanism for transaction validation and royalty monitoring. The project successfully developed a system that allows authors to validate their profits in ebook purchasing. The design of purchase contracts also further ensures authors gain their expected royalties. It also supports the main functions of current ebook trading such as refunds and price changes of ebooks. In addition, the project attempted to adopt ebook subscription services using individual-pay-per-page (IPPP) model. The proposed design of the system allows authors to validate their royalties with the participation of the subscribers. Refunds and price changes are also available. However, readers may lack the motivation and responsibility to assist with the royalty validation.

# 6. References

Chen, C. (2023, November 21). Answers to all your questions about the Kindle Unlimited reading subscription. *Amazon News*. https://www.aboutamazon.com/news/devices/what-is-kindle-unlimited

Chi, J., Lee, J., Kim, N., Choi, J., & Park, S. (2020). Secure and reliable blockchain-based eBook transaction system for self-published eBook trading. *PLoS ONE*, *15*(2), e0228418. https://doi.org/10.1371/journal.pone.0228418

Errera, R. (2023, May 31). Printed Books vs eBooks Statistics, Trends and Facts [2024]. *Toner Buzz*. Retrieved September 18, 2024, from https://www.tonerbuzz.com/blog/paper-books-vs-ebooks-statistics/#:~:text=E%2Dbooks%20experienced%20a%205.0,owned%20e%2Dreaders%20in%202014.

Huang, D., Liu, L., Deng, Y., & Chen, C. (2022). A Digital Media Subscription Management System Combined with Blockchain and Proxy Re-Encryption Mechanisms. *Symmetry*, *14*(10), 2167. https://doi.org/10.3390/sym14102167

Larson, C. (2022). Streaming books: confluencers, Kindle Unlimited and the platform imaginary. *Communication Culture and Critique*, *15*(4), 520–530. https://doi.org/10.1093/ccc/tcac032

Nizamuddin, N., Hasan, H., Salah, K., & Iqbal, R. (2019). Blockchain-Based Framework for protecting author royalty of Digital Assets. *Arabian Journal for Science and Engineering*, *44*(4), 3849–3866. https://doi.org/10.1007/s13369-018-03715-4

Polanka, S. (2013). Ebook Access: Business Models for Subscription Services. *Online Searcher*, *37*(2), 65–67. https://corescholar.libraries.wright.edu/cgi/viewcontent.cgi?article=1127&context=ul_pub

Sarode, R. P., & Bhalla, S. (2022). Supporting digital rights and copyright in online eBook stores using blockchain. In *Frontiers in artificial intelligence and applications*. https://doi.org/10.3233/faia220247

Shannon, S. (2015, June 25). Amazon's "pay-per-page" plan could alter writing as well as royalties. *The Guardian*. https://www.theguardian.com/books/booksblog/2015/jun/25/amazons-pay-per-page-plan-writing-royalties-ebook-kindle

Staiger, J. (2012). How e-books are used. *Reference & User Services Quarterly*, *51*(4), 355–365. https://doi.org/10.5860/rusq.51n4.355