SMU
SINGAPORE MANAGEMENT UNIVERSITY
School of
Computing and
Information Systems

IS483 Final Report

Cloud-based Automation for Digitalising Funds Disbursement Processes

Team Cloudify

Track

Digitalisation & Cloud Solutions

Version Number: 2.0.0

Submission Date: 30th November 2024

Team Members All DCS Track	1. Daryl CHUA Wei Xuan (daryl.chua.2021@scis.smu.edu.sg) – **Product Owner** 2. Aaron KWAH Boon Hou (aaron.kwah.2021@scis.smu.edu.sg) – **Scrum Master / Team Lead** 3. Jada TAN (jada.tan.2021@scis.smu.edu.sg) – **Backend Engineer** 4. Ray QUEK Cheng Wee (ray.quek.2021@scis.smu.edu.sg) – **Database / Cloud Engineer** 5. Maurice HO Zi Kai (maurice.ho.2021@scis.smu.edu.sg) – **Frontend Engineer** 6. Xavier KOO Jing Xuan (xavier.koo.2021@scis.smu.edu.sg) – **DevOps Engineer**
Sponsor	1. Ashton Kam (ashtonkam@gmail.com) - **Internal Affairs & Operations Associate** 2. Dylan Chong (it@alcorp.com.sg) - **IT Administrator**
Faculty Supervisor	1. Prof LUI Hoe Keong (hoekeonglui@smu.edu.sg)

Executive Summary

Team Cloudify presents a comprehensive solution that addresses critical operational inefficiencies in legal disbursement processing at Anthony Law Corporation (ALC), a law firm specialising in real estate conveyancing. The existing manual workflow, which consumed approximately eight hours daily, posed significant challenges in data accuracy, processing speed, and resource utilisation. Through the implementation of an automated disbursement management system, the team has achieved substantial improvements in operational efficiency while maintaining stringent security and compliance requirements.

The solution's technical architecture combines modern frontend technologies with robust backend services, implementing three key innovations. First, an enterprise-grade on-premise deployment ensures data sovereignty and security compliance. Second, an advanced OCR document processing system successfully handles diverse form formats from five major banks with 99% accuracy. Third, a sophisticated backup and recovery mechanism enables system restoration within five minutes, ensuring business continuity.

Quantitative analysis demonstrates significant performance improvements, with processing time reduced from 47.5 minutes to under 15 minutes per case. The system achieves a customer satisfaction score of 85.7% and recovers 246.66 work hours monthly, equivalent to 1.5 full-time employees. These improvements directly enhance operational efficiency while reducing human error and processing delays.

This report details the system's architecture, implementation methodology, and achieved results, providing insights into both technical innovations and business impact. While acknowledging current limitations inherent in on-premise deployment, we also outline future enhancement opportunities, including advanced notification systems and potential hybrid cloud integration. The solution establishes a foundation for continued optimisation of legal disbursement processing, positioning ALC for sustained operational excellence.

Table of Contents

1. Project Overview

Property transactions involve complex legal disbursements that traditionally consume significant time and resources. This chapter explores how we tackled these challenges through automation, setting the stage for a transformation in legal disbursement processing.

1.1. Sponsor Background

Anthony Law Corporation (ALC) is a well-established law firm based in Singapore, known for its comprehensive range of legal services. The firm specialises in areas such as real estate conveyancing, civil and criminal litigations, and family law. ALC leverages on a **strategic partnership** with a leading **property agent company**, who channels all closed deals to ALC for legal services. This relationship ensures a steady stream of real estate transactions, positioning ALC as a trusted solicitor for all property matters.

1.2. Problem Statement

ALC faces significant **operational inefficiencies** in managing the disbursement of financial assets on behalf of clients (who buy houses) due to **overwhelming paperwork** and **manual data processing** evident in Appendix A: On-site Discovery. These inefficiencies lead to frequent **errors**, unwanted **delays**, and **costly form resubmissions**. The complexity is further compounded by **varying requirements** from **different banks' forms**, exacerbating delays and errors in form filling.

Key Pain Points

- Difficulty in Retrieving Information: Locating information from **innumerable physical folders** and scanned documents (Appendix B: Current Dataset and Data Model) is cumbersome and time-consuming
- Manual Data Entry: The manual entry of data into different banks' forms increases the likelihood of **errors** and **inconsistencies**, with an **error rate** of **10%**
- Process Delays: The time-consuming nature of manual data entry and information retrieval causes **significant delays**, averaging an additional **4 days per case**

- High Costs: Errors in manual data entry lead to costly form resubmissions and additional administrative overhead, resulting in a **15% increase** in **operational costs**

If left unaddressed, these problems will notably reduce client satisfaction, prolong the increase in operational costs, and hinder the firm’s growth.

1.3. Value Proposition

Our solution streamlines the Law Firm's funds disbursement process by automating data extraction and form-filling, addressing key pain points effectively. By digitising data, it **eliminates** the **need to sift through** physical folders and scanned documents. This automation also **reduces errors** inherent in manual data entry, **ensuring accuracy** and **consistency** in form-filling. As a result, our solution **minimises process delays**, **accelerates workflow**, and **decreases operational costs** associated with form resubmissions. These improvements significantly enhance efficiency and streamline workflow management within the firm. Furthermore, features on the cloud-based platform such as robust search, drag-and-drop uploads, and an activity log enhances usability and operational transparency. Product and process innovation is illustrated in Appendix C: Innovation Summary.

1.4. Solution Scope

Our project delivers a legal disbursement management system for ALC’s property transactions. It is scoped into three objective tiers *(Primary, Secondary, and Tertiary)* to ensure focused development and clear deliverables.

The project will exclude any External bank integration, Mobile application, Cloud deployment, Third-party application integrations, and Legacy system migration.

Priority	Module	Core Functions
Primary	Case Management	Transaction workflow, client data management
	Form Generation	Multi-bank form automation, template handling
	PDF Processing	Document data extraction, data validation
Secondary	Role-based Access Control (RBAC) IAM	Access control, user authentication
	Activity Logging	Audit trails, system event tracking
	Scheduled Backup	Data retention, recovery procedures
	API Protection	Endpoint security, request validation
Tertiary	File Upload	Drag & drop interface, multi-file handling
	Case Search	Dynamic filtering, result optimisation
	Performance	Response optimization, caching
	API Logging	Service monitoring, error tracking
	System Recovery	Error handling, data consistency

1.5. Process Models

1.5.1. As-Is Process Analysis

Referencing Appendix D: As-Is Process Model, the current disbursement workflow requires the Business Development Managers (BDM) and Legal Secretaries to retrieve key client information from a physical data storage. Each of the clients will have their own dedicated folder also known as a “Case” which contains the transaction details. New information will be scanned using their OCR System and the physical copy will then be stored back into the Case folder.

This manual workflow, while functional, is heavily dependent on physical documentation and manual data entry, making it susceptible to delays and human error as mentioned previously in Section 1.2 Problem Statement.

1.5.2. To-Be Process Design

Our solution is designed to transform the disbursement process through three key changes (Appendix E: To-Be Process Model).

How data is handled and stored

The first change will be how we handle and store the data. Instead of manually searching through physical documents, we make use of the scanned documents that get outputted by their OCR Machine and extract key information automatically before storing it systematically into our database. Thereby creating a digital, centralised repository of all client and case information, eliminating the time-consuming task of physically searching through folders.

Automation of Manual Form Filling Process

The second change is the automation of the form filling process that took up hours of manual effort. By making use of the structured data stored in the database, our system can now automatically populate all relevant fields in disbursement forms with just a single click. This automation simplifies the process by working seamlessly across different support banks.

Redefining Roles of BDMs and Legal Secretaries

The third change is how we redefine the roles of BDMs and Legal Secretaries. We will automate the routine data entry they used to do and change their role to a Validator. Instead of spending hours combing through physical documents and manually keying in data, they will now focus their expertise on verifying pre-populated forms and making any necessary adjustments before submission. Thereby directly addressing the key pain points that are identified early on.

2. Technical Design and Implementation

Building a system that could extract, store key disbursement transaction information and process it into ready-to-use disbursement forms while making sure it is secure and operable by non-technical end users posed unique technical challenges. This chapter dives into our architectural decisions and implementation approaches that made the solution achieve its goals.

2.1. System Architecture Overview

2.1.1. High-Level Architecture

Our solution implements a hybrid between a service-oriented architecture together with a modular monolith architecture. It is coupled with a centralised web interface designed to service multiple user roles (Admins, BDMs, Lawyers, and Secretaries).

We chose this particular structure to best suit our sponsor's technical team while maintaining the benefits of modularity and clear separation of concerns. We achieved this by separating the authentication service into its own microservice while Case Management Service maintains the monolithic architecture but with module separation inside (Appendix F: Solution Overview Model).

With that, each service is designed to handle specific aspects of the system while maintaining secure and efficient communication channels.

2.1.2. Technology Stack

The technology stack for this project was carefully selected to balance modern development practices with robust enterprise requirements and our use case applicability (Appendix G: Technology Stack).

Front-end Technologies

On the Front-end, we implemented a responsive user interface using React and NextJS framework. Next.js supports Client-Side Rendering (CSR) which helps to provide the end-user with fast, interactive user experiences because there is no need to reload the entire page for dynamic data updates. Furthermore, it supports Page-based Routing which helps to speed up the development process as new pages can be defined using file structure instead of traditional code configurations.

For the Styling and Components System, we used Tailwind CSS and ShadCN components and customised it to fit the sponsor’s requirements and maintain a consistent styling throughout the application.

Back-end Technologies

The backend services are strategically split between Node.js with Express handling authentication flows, and FastAPI managing case processing operations, chosen for their respective strengths in secure session management, high-performance data processing and suitable library support for our use cases.

Data Management Technologies

Data persistence is achieved through MongoDB, selected for its flexible document model that accommodates varying bank form structures and it is suited for our project since the sponsor has requirements might be subjected to change over the course of the project (Explained in detail in Section 2.3.1 Choice of Database), while Redis provides caching capabilities to optimise frequent data access patterns.

Our core document processing uses a suite of Python libraries including pymupdf, openpyxl, and python-docx, enabling comprehensive handling of various document formats.

Security Technologies

For security, we implemented JWT for stateless authentication, bcrypt for password hashing, and RSA for encryption. File storage utilizes a local file system, optimized for the law firm's on-premise deployment requirements.

Service Orchestration

The entire system is containerized using Docker Compose, facilitating consistent deployment and scaling, with services communicating via RESTful APIs. The benefits of containerisation and overall architecture is further explained in Section 4.1 Quality Attributes Analysis.

2.2. Core Technical Components and Features

2.2.1. Frontend System Features

The system's interface architecture focuses on four core areas that directly address the operational challenges faced by legal secretaries and administrators.

Case Dashboard

The Case Dashboard is the primary interface that most of the users will be interacting with. It provides immediate visibility into summarised details that help to identify a particular case such as CaseID, Property Address, Type of transaction, Status and a list of actions that can be taken (Appendix H: Case Dashboard UI). The interface uses intuitive color coding and status indicators to highlight the case progression (Appendix I: Detailed UI Components).

Search and Filtering

Search and Filtering capabilities are implemented through a responsive interface that supports both basic and advanced query parameters. The search component utilises debounced inputs and optimized query handling to provide real-time results while maintaining system performance. Users can filter cases based on multiple criteria including status, date ranges, and assigned personnel.

Form Processing

The Form Processing features a streamlined document upload mechanism with **drag-and-drop** functionality (Appendix J: Document Upload UI) . This interface integrates directly with the OCR processing pipeline, providing immediate feedback during document processing and allowing users to validate extracted information through an interactive verification interface.

Form Generation

The Form Generation interface dynamically adapts to different bank requirements, presenting only relevant fields and validation rules based on the selected banking institution (Appendix K: Form Generation Options and Appendix L: Predefined Templates). It is made even more intuitive by allowing the form generation to be done via a **One-Click system**, cutting down overall steps and time needed to generate the disbursement forms.

2.2.2. Backend Services Features

API Services

Our application integrates a robust API service designed and implemented using **FastAPI**, adhering to the principles of **REST API (Representational State Transfer Application Programming Interface)** conventions. This ensures that our system provides scalable, high-performing, and well-structured interfaces for client-server communication. By utilizing FastAPI, we leverage a modern framework that facilitates rapid development, type validation, and automatic documentation generation, making our APIs reliable and user-friendly.

We adhere strictly to **REST (Representational State Transfer)** conventions in our API design. REST is a widely accepted architectural style that provides a set of rules for building scalable and stateless web services. Our APIs follow these principles by utilizing standard HTTP methods such as GET, POST, PUT, and DELETE to define clear actions for resources (Appendix M: Sample API Documentation). Each endpoint represents a specific resource, identified by logical and descriptive URLs. For instance, “/cases/archived” retrieves a list of archived cases, aligning with RESTful conventions for resource naming.

In compliance with REST standards, responses are structured in JSON format, ensuring consistency and ease of integration for clients. HTTP status codes, such as “200 OK” for successful requests and “404 Not Found” for missing resources, are used to provide clear and intuitive feedback to clients. This consistency simplifies error handling and debugging for consumers of our API.

We leveraged FastAPI's capability to perform data validation at runtime, reducing potential errors and improving the robustness of our application. For example, our endpoints validate incoming data using Pydantic models, ensuring that only well-structured requests are processed. This eliminates common issues related to invalid inputs, enhancing both security and reliability.

Multi-bank Form Processing

Our Multi-bank Form Processing feature uses distinct endpoints for each bank to address the unique requirements of their forms. Each bank typically provides forms with varying structures, fields, and validation rules. By creating dedicated endpoints, we ensure that each bank's specific needs are handled independently and efficiently.

This design approach enhances flexibility and simplifies maintenance. Each endpoint is tailored to the data structure and validation logic of its respective bank, reducing the risk of errors and ensuring seamless processing. Changes or updates to one bank’s requirements can be implemented without impacting other endpoints, promoting modularity and minimizing complexity.

The separation also improves performance by eliminating unnecessary conditional checks that a unified endpoint would require. It ensures scalability, allowing us to onboard new banks or adjust existing endpoints without affecting the overall system. Additionally, this structure simplifies the integration process for API consumers by providing clear and predictable pathways for form submissions.

Balance towards Cash Price Calculation

The Automatic Calculation of Balance Cash Towards Purchase Price feature streamlines the process of determining the cash required for property purchases. Traditionally, users calculate this amount manually, which is both time-consuming and prone to errors. Our application eliminates this challenge by dynamically calculating the balance cash required based on user-provided inputs such as purchase price, deposit, loans, and CPF contributions.

When users input the relevant data, the application immediately computes the balance cash using the formula (Appendix N: Formulae Calculation & ToolTips):

Balance Cash = Purchase Price - Deposit - Housing Loans -
Term Loans - Bridging Loans -
CPF Lump Sum

This value is displayed in the designated field, ensuring real-time feedback for the user. To enhance transparency and usability, we have incorporated a tooltip feature. By hovering over the balance cash field, users can view the formula and a breakdown of the calculations, showing how the application derives the value. This provides clarity and builds user confidence in the computed results.

2.2.3. Document Storing Features

Our Document Management System simplifies file handling by enabling users to drag and drop documents directly into the application. The system ensures seamless organization, automatic tagging, and reliable backups to meet user needs and enhance document security.

When a user uploads a document, the application automatically generates a structured file name that includes relevant metadata, such as the case ID and timestamp. This file is then tagged and stored in a dedicated folder corresponding to the case ID within the local file storage system. This systematic approach ensures that all documents are easily traceable and organized (Appendix O: Supporting Document Storage).

Each uploaded document is automatically saved in a folder named after the associated case ID. This structured storage system reduces the risk of misplaced files and provides a clear linkage between documents and cases. Users can easily retrieve documents without the need for extensive manual searching.

To enhance reliability and prevent data loss, the system is integrated with OneDrive. Each document upload is simultaneously backed up in the respective case ID folder on OneDrive. This redundancy ensures that critical documents are accessible and secure, even in the event of local storage issues. The backup process is automated and follows the same case-specific folder structure, maintaining consistency across storage locations.

2.2.4. Custom Identity Access Management (IAM)

Since we are **constrained to on-premise system interactions** and **limited external network communications**, we had to implement a **custom Identity and Access Management (IAM)** solution. Our IAM comprises an authentication server built on Node.js and Express.js. We use Node.js's event-driven architecture to handle concurrent authentication requests efficiently while maintaining secure user sessions.

Node.js

The authentication server leverages on Node.js's built-in cryptographic support for implementing RSA encryption and JWT signing operations. This native integration minimises external dependencies while ensuring secure transmission of sensitive user credentials. The system's asynchronous design helps it manage multiple concurrent users, a common scenario in legal practice environments, without compromising performance or security standards.

Express.js

Furthermore, Express.js provides the framework for implementing RESTful APIs with structured route definitions for authentication operations. The framework's middleware architecture enables comprehensive request processing, implementing token validation, input sanitization, and error handling in a modular fashion. This approach not only enhances security through systematic request validation but also facilitates maintainable code structure (detailed in Section 2.5 Security Implementation).

Integration

The integration layer implements a public-private key infrastructure for credential encryption, ensuring secure transmission of login data. Upon successful authentication, the system issues JWTs for session management and role-based access control, providing granular authorization for different user roles within the legal practice (explained further in Section 2.5.4 Authentication Flow).

2.2.5. Document Processing Features

Our solution addresses two key workflows in the legal disbursement processing: automated case creation through OCR text extraction and a more streamlined disbursement form generation process. Both features are designed to support documentation requirements for six major banks, UOB, Maybank, Citibank, Bank of China, HSBC, and DBS.

OCR Text Extraction

We have an OCR text extraction component that uses PyMuPDF to access, process and extract the pre-scanned documents from our sponsor's OCR machine mentioned earlier in Section 1.5 Process Models. We added our own custom parsing logic such as the use of regular expressions and pattern matching so that we could detect and extract the fields for the case details accurately. This extraction implementation is specifically designed to identify and categorise fields such as client names, case references, and financial details, so that the creation of a new case will not require much manual data entry.

To mitigate OCR recognition uncertainties, our solution identifies multiple potential matches for each field, presenting users with validated options for selection, thereby ensuring data accuracy while maintaining processing efficiency.

Disbursement Form Generation

For disbursement form generation, our solution had to support both Excel-based forms and Word document format. We capitalised on openpyxl library for Excel sheets and python-docx for Word documents so that we can meet the functional requirements set.

Predefined templates for each bank are stored in the system, with the precise positions of each field mapped to specific case attributes. These templates ensure consistent formatting and compliance with each bank's unique requirements. During the form generation process, relevant case details, such as client information and various disbursement amounts, are dynamically retrieved from the database and populated into their designated fields within the templates. Thus, the generated forms adhere to the specific formatting and compliance requirements of each bank.

Upon form generation, users can review the documents for accuracy, print them, and submit them to the respective banks. This automated approach has transformed an error-prone manual process into a streamlined operation, significantly reducing processing time while maintaining document accuracy. Detailed results can be found in Section 4.1.1 Functional Suitability.

2.3. Data Management Features

2.3.1. Choice of Database

The selection of MongoDB as the database system was driven by the complex nature of legal disbursement data and specific operational requirements. The system's data model includes relationships between clients, properties, and financial details, making a document-based architecture particularly advantageous.

Unlike traditional relational databases that would require complex join operations across multiple tables, MongoDB's document model allows entire case records to be stored as cohesive units, significantly improving query performance and data retrieval efficiency.

MongoDB aligns naturally with the JSON-like structure of the case data. Each case, with all its nested details, can be stored as a single document, reducing the need for complex joins and ensuring efficient retrieval. MongoDB also offers features such as aggregation pipelines and indexing, which enhance querying capabilities.

Alternatives like DynamoDB are less suitable because of their limited support for deeply nested data and complex queries. Similarly, DocumentDB's one-month free trial imposes cost constraints, making it less viable for long-term use. MongoDB's combination of flexibility, scalability, and robust features makes it the best fit for this application.

2.3.2. Data Structure/Model Design

With reference to Appendix P: Database Model, the structure of the data models are modelled similarly to how the sponsor has their information stored physically. For example, A dedicated folder where they section out the paperwork that are related to finance, client information, communication, property etc, are mapped accordingly on MongoDB.

We used nested structures for logical grouping to group related data logically. For example, fields like clients, property, and finance_details are embedded within the Case document, ensuring that all information relevant to a single case is stored together. This design eliminates the need for multiple joins and allows for efficient read operations.

To maintain comprehensive audit trails and historical records, we use arrays for tracking changes and user actions. This design decision proves particularly valuable in legal contexts where transaction history and change tracking are crucial compliance requirements. The logs and clause_history arrays maintain temporal data within the parent document for easy association and retrieval later on.

Fields like “bdmid”, “lawyerid”, and “secretaryid” are stored as references to external collections, allowing for modularity and efficient cross-referencing. This balance between embedding and referencing ensures optimal performance for both read and write operations.

2.3.3. Backup System

We developed a comprehensive backup and recovery system to ensure data integrity and business continuity in the event of a system failure.

Backup Process

For the backup process, we leverage MongoDB’s native utilities combined with our custom automation script to ensure minimal data loss. Our custom automated cron backup jobs are ran based on a set schedule daily (depending on client requirements), where it captures the latest state of the primary and authentication databases and exports it in BSON format to preserve schema and data consistency. This automation minimises manual intervention and ensures regular backups (Appendix Q: Detailed Backup CRON Job Configuration).

To optimise storage and improve organisation, the files are compressed into timestamped .zip files where they are stored in a dedicated directory, providing a structured and accessible repository for all backups. The timestamping ensures clear versioning, preventing overwrites and enabling precise restoration of specific backup versions.

Disaster Recovery

To enhance disaster recovery capabilities, our solution extends beyond local storage by integrating with Sponsor’s OneDrive for secure offsite backup storage. The compressed backup files are securely uploaded and synced to their account. This offsite storage provides an additional layer of protection, safeguarding data against local system failures (recovery procedures detailed in Appendix R: Recovery Procedure).

2.4. Performance Engineering

2.4.1. Redis Caching

We implemented Redis caching and containerized resource management. Redis serves as the primary caching layer, storing frequently accessed data including user sessions, case details, and form templates in memory. This implementation significantly reduces database load during peak operations, with cache invalidation managed through a hybrid approach combining time-based expiration and event-driven updates (Appendix S: Caching Strategy).

2.4.2. Container Resource Allocation

We utilised Docker Container orchestration which allowed us to set consistent resource allocation to maintain system stability. We are able to set the optimal CPU and memory limits for each service container, preventing resource contention while ensuring reliable operation under varying workloads (Appendix T: Sample Resource Configuration).

2.5. Security Implementation

2.5.1. Identity and Access Management (IAM) Framework

The IAM framework provides secure access to the system by enabling precise access control through **Role-Based Access Control (RBAC)**. By categorising users as **Administrators** or **Users**, the system is able to enforce strict segregation of duties.

Administrator	User
Exclusive access to admin pages and API routes	Limited to specific functionalities, such as modifying their passwords
Privileged capabilities such as creating new users and modifying user details/passwords	Restricted from accessing admin-exclusive pages and routes

This design ensures that sensitive actions are performed only by authorised personnel, minimising the risk of unauthorised access or privilege escalation.

2.5.2. Data Protection

Our solution implements a three-part data protection strategy such as RSA Encryption, Bcrypt Hashing and Secure Session Management using JWT.

RSA Encryption

On the Transport Level, we use RSA encryption to secure transmission of credentials between client and server. This asymmetric encryption approach utilises public key cryptography, where credentials are encrypted with the authentication server's public key and can only be decrypted using the corresponding private key held securely on the server.

Bcrypt Hashing

For Password Security, we use Bcrypt hashing with dynamic salt generation. This will protect the stored credentials against both rainbow table and brute-force attacks through computationally intensive hashing algorithms. The addition of unique salts to each password hash provides an additional security layer, ensuring that even identical passwords produce different hash values in the database.

JSON Web Token (JWT)

For Session Management, we use JWT which securely encodes the user’s role and permission. These digitally signed tokens prevent unauthorised manipulation while maintaining stateless authentication. The JWT tokens are stored in session storage with appropriate security flags, protecting against cross-site scripting (XSS) attacks.

2.5.3. API Security

Our system has strict API security checks to prevent any unauthorised API access. We validate the user’s role at the authentication server level using the JWT that is provided by the user during each API call. Tokens are also validated each time at the start of each API request to prevent the misuse of tampered tokens. Thereby protecting against common attacks such as unauthorised access, token replay, and privilege escalation.

2.5.4. Authentication Flow

The full authentication process can be found in the Appendix U: Authentication Flow, but the general secure end-to-end flow is as follows:

Client-Side Encryption	● Users initiate login by encrypting their credentials with the server's public RSA key.
Authentication Server	● Decrypts the credentials and validates the user ● Issues signed JWT tokens with role-based claims
Session Management	● Tokens are stored in the client's session storage and used for subsequent requests
Validation on Every Request	● Tokens are validated on both the authentication and case management servers to ensure authenticity and integrity before processing client requests

3. Development Process and System Testing

This chapter will look into the methodological approaches we took to develop, manage and test our solution. It contains the framework used, comprehensive testing strategies, and risk management protocols implemented throughout the project lifecycle.

3.1. Development Methodology

3.1.1. Agile Framework Implementation

Our team follows the Agile Framework to ensure the efficient and flexible delivery of the project outcomes. We chose this approach because the application we are creating for ALC is new and they may not know what the complete feature list they want at the beginning of the project. This approach allows our team to be flexible and adapt the product through continuous feedback since it is crucial for a system such as this to have close alignment with legal processes and user needs.

Sprint Cycles

The development process was structured around one to two-week sprint cycles (Appendix V: Example of a Sprint), each comprising Sprint Planning, Daily Execution, Sprint Review and Retrospective (Appendix W: Sprint Activities). To mark an item as done, we came up with a clear definition of done (Appendix X: Definition of Done).

Tools used

We utilised JIRA as our issue tracker and created tickets to assist in the documentation and logging of user stories and manage our overall consistency with Burndown charts.

Documentation Standards

This framework was supported by systematic documentation including sprint planning artifacts, detailed meeting minutes, review outcomes, and retrospective (sample documentation in Appendix Y: Sprint Planning Artifact, Appendix Z: Meeting Minutes, Appendix AA: Sprint Review, and Appendix BB: Retrospective).

3.1.2. Stakeholder Management

Our team communicated with stakeholders (internal/external) using established guidelines (Appendix CC: Communication Guidelines) since the start of the project with a minor modification to how we conducted Sprint Reviews with the client.

External Stakeholders

For External Stakeholders, Product Owner served as the primary liaison with project sponsors, maintaining clear and consistent communication channels. External interactions were primarily conducted through Sprint Reviews for completed work demonstrations, User Acceptance Testing sessions, regular progress updates.

School Stakeholders

For School Stakeholders, Team Lead will lead the discussions with our project supervisor, Prof Lui Hoe Keong, via regular face-to-face check-in meetings after the end of each sprint where we will clarify any questions and keep him updated on that sprint's achievements and any potential risk (Appendix DD: Post Sprint Check-in). At the end of each check in, we will update the document and send it to Prof Lui over Teams. This helps everyone to always be on the same page before we start with the next sprint.

Internal Team Communication

For Internal Team Communication, we use a combination of Telegram, Stand-Up sessions, Face-to-Face sprint plannings and sprint retrospective meetings over Zoom or in-person to communicate issues and updates. Every Sprint meeting

3.1.3. Code Management

As the project team will be working closely on the project codebase, we adhered to the convention set in Appendix EE: Git Workflow & Commit Convention to ensure clear communication.

3.1.4. Knowledge Sharing

We utilised Confluence Wiki as our central knowledge repository, ensuring efficient information sharing and documentation (Appendix FF: Knowledge Sharing). This helped us to maintain team alignment and understand the sponsor's business context faster.

3.1.5. API Documentation

For each API, we ensured that the respective inputs and outputs are documented so that it facilitates knowledge transfer between developers and ease the handover process from the project team to the sponsor (Appendix M: Sample API Documentation).

3.2. Risk Management

The project encounters several risks that could affect its successful completion, including schedule risk, project management risk, compliance and security risk, and technical risk (Appendix GG: Project Risks Analysis). We have identified them early and came up with mitigation efforts to minimize the impacts of these risks. We used the Risk Assessment Matrix to categorise the different risks based on impact and priority.

3.3. Testing Strategy

3.3.1. Unit Testing

For unit testing, we used Vitest as the front-end testing framework to ensure comprehensive validation of the application components and PyTest as the back-end testing framework. Vitest provides an efficient environment for running unit tests, using modern Javascript features, and supporting TypeScript, which aligns well with our project's tech stack (Appendix G: Technology Stack). It provides instant feedback on test coverage and failures, enabling the team to maintain high-quality code with immediate insights. PyTest works similarly to Vitest but it is designed for carrying out Python API testing.

We focused on two key aspects of our testing strategy. Firstly, component-level validation ensures that individual components such as data tables, search functionality, and case forms perform as expected. Secondly, integration points verify interactions between components, such as the seamless display of data and correct handling of user actions like editing and searching.
An example of a Vitest coverage report showcasing detailed metrics, such as statements, branches, functions and lines tested, is provided in Appendix HH: Unit Test Report.

Vitest is ideal for ensuring an immediate feedback loop while developing because it seamlessly connects with modern build tools like Vite. Furthermore, we easily understand untested code paths due to the built-in functionality for coverage analysis, which enables us to focus our efforts on improving coverage in crucial areas.

3.3.2. Integration & End-to-End (E2E) Testing

For E2E testing, we configured Cypress as the primary end-to-end testing framework which could handle the comprehensive validation of critical user workflows. It could execute tests within the browser environment, providing accurate simulation of real-world user interactions and immediate feedback on test failures.

We had two main workflows 1. Legal secretaries processing disbursements and 2. Administrators managing system access. The test cases would simulate the user roles to ensure authentication flows and document handling processes maintain integrity across system updates. An example of an automated test report and the Cypress UI can be found in Appendix II: Cypress Automated Test Report & User Interface.

Cypress had integration with the browser's JavaScript engine so that it could simulate the dynamic interactions such as form validation. In addition, it has built-in debugging like time-travel debugging and automatic waiting mechanisms which help to speed up our team’s overall testing process. If there was any failure during the test case, we will be able to see screenshots at those critical workflow points and points of failure to resolve the issue more efficiently.

3.4. Deployment Pipeline

For Deployment, we use the Continuous Integration and Continuous Deployment (CI/CD) principles and implemented it using GitHub Actions to streamline the testing, building, and deployment processes.

The pipeline mirrors the branching strategy mentioned previously in Section 3.1.3 Code Management. It has 3 stages, and each stage has its own purpose to ensure a predictable and well-monitored path:

Feature Development	Developers work on feature, bug, or chore branches, which are merged into the dev branch after local testing.
Integration and Testing	The dev branch is merged into the staging branch, where automated tests run to validate code changes and identify potential issues.

Release to Production	After successful validation in the staging environment, the changes are merged into the main branch and deployed to production following final approval.

The pipeline is triggered on pull requests targeting the staging branch and can also be initiated manually through the *workflow_dispatch* event. Once triggered, a testing environment is set up and automated tests will run. The pipeline also builds and runs the application using Docker Compose, simulating the production environment. This step ensures that any potential issues related to containerization or deployment are identified early. An example of the GitHub actions workflow can be found in Appendix JJ: GitHub Actions Workflow.

4. Analysis & Results

The transformation of the disbursement process yielded quantitative and qualitative improvements in operational efficiency. This chapter presents a detailed analysis of our system's performance using key metrics and Analysis frameworks.

4.1. Quality Attributes Analysis

The quality attributes analysis evaluates the system's effectiveness across eight key software quality characteristics defined by ISO/IEC 25010, providing quantitative measurements of the system's functional and non-functional requirements.

4.1.1. Functional Suitability

In terms of Functional Suitability, our system achieves **100% coverage of sponsor requirements** mentioned in Section 1.4 Solution Scope with the feature implemented mentioned earlier in Section 2.2 Core Technical Components and Features while maintaining **99% accuracy in data extraction**. Most significantly, it reduces the disbursement process time **from 47.5 minutes to under 15 minutes per case**, representing a substantial efficiency improvement in daily operations.

4.1.2. Performance Efficiency

In terms of Performance Efficiency, we managed to achieve remarkable improvements with data retrieval speeds increased by **48 times with Redis Caching** compared to without. Our system is able to **consistently handle 30 concurrent users** without performance degradation which is effectively supporting the firm's peak operational demands (Appendix KK: Caching Time Improvement).

4.1.3. Compatibility

In terms of **Compatibility**, Our use of NextJS and React mentioned earlier in Section 2.1.2 Technology Stack allowed our solution to support multiple browsers (Chrome, Firefox, Safari, Edge) with **100% feature parity**. Our solution maintains a consistent performance across all browsers with **<5% variation in application response times** between initial page loads (Appendix LL: Browser Response Time Test). Our implementation of RESTful API further supports the seamless integration with existing and future systems.

4.1.4. Usability

In terms of **Usability**, our use of ShadCN component library, wireframing, consistent User Acceptance Test (UAT), and feedback iteration process allowed our solution to achieve promising results in unguided user tests where users are able to achieve a **100% task completion rate after 2 hours of training**. Furthermore, we achieved an **overall 4.375/5 user satisfaction score**. Further details on User Experience can be found in Section 4.3 User Experience Analysis.

4.1.5. Reliability

In terms of Reliability, our system has a consistent backup process and a streamlined recovery procedure that is mentioned earlier in Section 2.2.3 Backup System that is able to restore operations **within 3 minutes** of a complete system failure (Appendix NN: System Recovery Time Test).

4.1.6. Security

In terms of Security, with the implementation of the robust authentication system mentioned in Section 2.5 Security Implementations, our solution is able to provide a comprehensive access control and test has shown that it is able to differentiate the two distinct user roles, and **prevent any unauthorised access attempts** from going through.

4.1.7. Portability & Maintainability

In terms of Portability and Maintainability, our solution showed that it can **go from nothing to a fully deployed system** on different server configurations **within one hour** (accounting for varying network configuration times). Furthermore, our solution has a **100% API documentation**

coverage and comprehensive code testing (Appendix HH: Unit Test Report) . The modular architecture mentioned earlier in Section 2.1.1 High Level Architecture meant that it is able to handle independent component updates without cascading modifications, significantly reducing maintenance overhead.

4.2. Process Transformation Analysis

Quantitative measurements demonstrate successful transformation across key operational metrics: from 45 minutes to under 15 minutes per case reduction in processing time, 99% data accuracy, and elimination of having to sift through physical storage. These improvements validate the transformation strategy outlined in Section 1.5 Process Model.

Since our solution is still in the testing phase, we do not have the additional numbers to measure long-term impacts such as bank rejection rates. Initial staff feedback indicates positive workflow changes. Legal secretaries mentioned that the time saved from using our solution enabled them with additional time to do higher-value activities, suggesting successful alignment with the transformation objectives. Detailed results can be found in Section 5.1 UAT Process and Results.

4.3. User Experience Analysis

Applying the feedback structure and process mentioned in Section 5.1, we are able to achieve the following results from our end-users UX feedback sessions.

Quantitative results include:

- Customer satisfaction score (CSAT): 87.5%
- Application responsiveness: 100% ("Did the application load/respond quickly enough?")
- Task completion rate: 100% ("Were you able to successfully generate the required forms?")
- Overall satisfaction: 93.3%
- Time taken for task completion: Average of 7 min 54 seconds
- Error rate: Average of 5.66 errors per user interaction

On User Satisfaction, our solution achieved a **high Customer Satisfaction Score (CSAT) of 87.5%** and is supported by the overall **93.3% overall satisfaction rating** during testing. As for operational metrics, our solution was able to achieve a 100% for application responsiveness and task completion rate which supports our solutions's effectiveness in addressing end user's pain points and business problems mentioned earlier in Section 1.2 Problem Statement.

Our performance measures indicate an **Average Task Completion Time of 7 minutes and 54 seconds** which is a significant improvement over manual processes. However, the analysis also identified opportunities for improvement in the user interface, with an **average of 5.66 errors per user interaction** suggesting areas for potential refinement.

Qualitative feedback through open-ended questions highlighted two specific enhancements that can be made to the solution. For example:

1. **Activity Logging:** One user asked, "*For activity logging, is there a way to search?*" The root cause analysis revealed that while the activity logs were comprehensive, they required manual inspection. The feedback was prioritized as low-medium, and the action plan included a short-term solution for search and filter functionality, with a long-term goal of adding notifications for specific log events.
2. **Alerts and Email Notifications:** Another user inquired, "*Are there alert or email notifications? Because sometimes two secretaries help each other on the same case*". The system lacked a notification feature, and manual monitoring was prone to oversight. This was prioritized as medium-high, and the action plan included short-term email notifications and long-term customizable email templates

These enhancements have been added to the Section 7.2 Future Directions.

5. Implementation Journey

5.1. UAT 1 (Sprint 1-3)

5.1.1. Process

UAT 1 was conducted for Sprint Reviews 1, 2, and 3, where the development team demonstrated the functionality to the end-users and walked through the acceptance criteria for each user story. After each demonstration, the team asked the end-users if they observed the expected outcomes, to which the end-users simply responded with a yes or no, confirming their agreement. This process continued for each user story until all criteria were validated and accepted.

5.1.2. Results

The results of UAT 1 mainly reflected user acknowledgement rather than active interaction with the system. While the feedback was largely positive with all criteria being ticked off, this approach does not uncover potential usability issues or provide deeper insights into user experience, as it lacked real-world task execution by the users.

5.1.3. System Enhancements

UAT 1's reliance on team-led demonstrations with end-users confirmation did not provide meaningful real-world insights into system usability or user behaviour. To address this, we changed our approach to UAT 2, which focused on capturing authentic user interactions and feedback to better evaluate the system's intuitiveness and real-world challenges.

5.2. UAT 2 (Sprint 4,5)

5.2.1. Process

Our UAT 2 was conducted during Sprint Reviews 4 and 5 focused on validating system usability through unguided user interactions. This approach, where end users completed role-specific tasks independently, provided authentic insights into user behavior and system intuitiveness.

Metric	Type	Method
Application Responsiveness	Quantitative	User survey
Ease of Navigation	Quantitative	User survey
Task Completion Rate	Quantitative	Observed and Recorded
Overall Satisfaction	Quantitative	User survey
Improvement Suggestions	Qualitative	Open-ended survey
Additional Feedback	Qualitative	Open-ended survey
Time Taken	Quantitative	Observed and Recorded
Error Rate	Quantitative	Observed and Recorded

We collected Feedback including targeted improvement suggestions and additional sentiments via open-ended questions. The collected feedback then underwent a structured process where we identified underlying issues through Root Cause Analysis, prioritization of problems based on the impact and required effort levels before creating short-term or long-term user stories for the product backlog (Appendix NN: Feedback to Action Process).

5.2.2. Results

Quantitative

For each user scenario, we measured the average of their quantitative scores. Secretaries performed at an average of around 9 minutes and 15 seconds, corroborating our claims of process time reduction to around 11 minutes. Additionally, admins were able to complete both secretaries' and their scenarios at an average of 10 minutes and 12 seconds. Results also conclude that all end users felt the application was responsive and were able to complete their respective tasks successfully, with an overall customer satisfaction score of 87.5%. Full UAT results can be found in our (Appendix OO: UAT Results).

Qualitative

Open-ended questions allowed us to identify specific and targeted areas for improvements, such as OCR extraction from other relevant documents to be added to case details. Others included tangible feedback which we reflected upon and is illustrated in our feedback to action process shown in our User Experience Analysis.

5.2.3. System Enhancements

System enhancements resulting from UAT 2 focused on four key areas:

OCR Support Expansion

First, OCR functionality in Section 2.2.5 Document Processing Features was expanded to support forms from four additional banks, bringing the total to five supported institutions. This enhancement significantly broadened the system's capability to handle diverse case requirements.

OCR Support Expansion

Secondly, we included support for the Completion Account section within Disbursement Forms, addressing a specific client requirement identified during mid-term review.

Security Improvements

Thirdly, we added security improvements focused on strengthening authentication mechanisms, as detailed in Section 2.5.2 Authentication Flow, ensuring robust protection of sensitive client data.

Performance Improvements

Lastly, we improved the performance through Redis caching implementation, described in Section 2.4.1 Redis Caching, achieved significant response time improvements for frequently accessed data, particularly in high-demand workflows like case management and form generation.

5.3. Change Management

The change management strategy addressed several key transitions in system deployment and user adoption.

5.3.1. Cloud to On-premise Pivot

A significant pivot occurred early in the project when security considerations required our team to make a shift from cloud to on-premise deployment at the sponsor's request. This transition required substantial architectural adjustments, particularly in implementing in-house identity management and JWT-based authentication for secure client-server communication and general development/deployment strategies (Appendix PP: Cloud to On-Premise Architecture Change).

5.3.2. Functional Scope Changes

During the development phase of the project, we had to maintain flexibility in order to address the evolving functional requirements through an agile feedback loop. Functional refinements were systematically managed through Jira tasks and user stories, enabling continuous improvement without compromising the core project scope.

Our team was new to the legal domain and as a result, had limited domain knowledge in legal conveyancing. This was mitigated effectively by having regular and open communication channels via Telegram as well as structured sprint review sessions. This not only helped us increase the domain knowledge but also made it easier for us to prioritise the features the sponsor wanted to see the most and keep it aligned with the business needs.

5.3.3. User Training and Support

User adoption and training for our new solution was also another key transition issue we had to tackle. We had Post-UAT training sessions with the end users so that they could effectively use the new solution. In addition, we crafted the user guides for both the administrator and the end-user to handle common issues such as deployment, form upload and extraction use cases to complement our existing communication channel (Appendix QQ: User Guides).

To ensure long-term system sustainability, detailed technical documentation such as API Documentation, Recovery Procedures was developed for the sponsor's IT personnel, enabling independent system maintenance and troubleshooting.

6. Achievement and Limitations

6.1. X-Factors

6.1.1. Enterprise On-Premise Deployment

The biggest technical win for our team was the successful implementation of an enterprise-grade on premise deployment of our solution. Our team had to pivot our solution's deployment plans midway through the project to accommodate the stringent security and compliance requirements of the Sponsor. This deployment model required sophisticated architectural decisions beyond standard cloud implementations, particularly in establishing secure infrastructure within the law firm's environment. The technical implementation encompassed three critical components.

Limited to On-Premise Connectivity

First, our team could not use any available Cloud IAM solution provider like AWS Cognito and must develop our own custom security architecture mentioned in Section 2.2.4 Custom IAM which integrates Role-Based Access Control with JWT token validation to provide protection while maintaining system usability.

Unknown Physical Infrastructure

Second, the Physical Infrastructure Implementation meant that we had to develop our application in a way where we could deploy it regardless of platform/OS because we had no visibility to what compute resource that would be allocated to us and had to plan ahead. The physical infrastructure also meant that we had to be strategic with the versions we deployed since we could not easily update it without going on-site. We utilised Docker Orchestration and Resource Management to optimise hardware resources, ensuring consistent performance across all system components.

Need to Configure Network Connectivity

Third was the Network Configuration, our team had to manage the network settings to ensure other devices that were on the sponsor's office intranet could access and reach the server, that meant we had to set a static IP address and configure the firewall and Access Control List (ACL). Our team went beyond just making sure the application was suitable for on-premise deployment by developing it so that it could be deployed on any cloud compute provider through the use of containerisation.

The First Time for All

Overall, considering that it was the first time anyone in our team had the opportunity to develop and deploy the enterprise-level application on an enterprise on-premise infrastructure, it was in itself a major feat. Our solution exceeded the requirements expected from our client, and granted them full operational control over the system with future-proofing built in.

6.1.2. OCR Document Processing

The second biggest technical win was developing and successfully implementing the OCR Document Processing as it played an important role in automating the complex legal document workflow. Our system successfully addresses the challenge of diverse bank form formats through an adaptive approach mentioned in Section 2.2.5 Document Processing Features. It currently supports five major banks with 99% accuracy in data extraction. Together with the other modules in the Case Management Service, it is able to deliver significant value to the client's business.

The key technical innovation includes the ability for our system to handle varying forms structures without requiring extensive reconfiguration. By combining several Python libraries together, we managed to not only streamline the processing of current banking documents but also provide a flexible framework to incorporate the additional form types. The system's architecture, combining PyMuPDF for document analysis with custom validation logic, ensures error handling while maintaining high throughput rates.

Our implementation's versatility extends beyond its current project application in the legal disbursement field. It has the potential to be adapted and used in many document-intensive sectors like Healthcare, Logistics, and Retail. Our solution can assist those fields with automating record digitisation and data extraction workflows.

6.1.3. System Backup and Recovery

The third biggest technical win was the implementation of an efficient backup and recovery system given the constraints of limited network access and hardware resources. The challenge demanded an innovative approach to ensure business continuity without relying on traditional redundancy methods like failover systems or master-slave configurations.

We had to get creative to prevent complete loss of data in the event of complete system hardware failure while making it easy and intuitive so that non-technical people can restart and restore without much help, quickly and effectively.

The technical solution combines automated database exports with local document synchronisation through a scheduled CRON job implementation. The details of the implementation is mentioned in Section 2.3.3 Backup Systems. The integration with Company's OneDrive for secure offsite storage showcases our elegant solution to external storage limitations, enabling reliable backup without compromising security protocols.

Most notably, the recovery system achieves exceptional efficiency through a streamlined restoration script that enables non-technical users to restore system state within three minutes. The solution demonstrates how our team's technical innovation can address enterprise requirements even under significant infrastructure constraints.

6.2. Business Impact

The business value delivered through our solution is demonstrated through several key performance indicators (KPI). CSAT score as mentioned earlier in Section 4.3 User Experience Analysis achieved a **significant score of 85.7%**, validating that the technical implementation effectively addresses real user needs and operational requirements.

The system's impact on operational efficiency mentioned earlier in Section 4.1.1 Functional Suitability is particularly noteworthy, achieving a **78% reduction in disbursement processing time**. This transformation is most evident in the end-to-end process comparison: **traditional manual processing required approximately 47.5 minutes per case** (including case creation, editing, and disbursement form preparation), while the **automated system completes the same process in under 15 minutes** through OCR-enabled case creation and automated form generation.

This efficiency improvement translates to substantial resource savings, with the system recovering **246.66 work hours monthly** - equivalent to 1.5 full-time employees. The automation of case creation alone demonstrates a **5x speed improvement**, significantly reducing the initial processing time for property transactions and enabling staff to focus on higher-value activities.

6.3. Limitations

The implementation of the solution faces several technical constraints, primarily stemming from the requirement for on-premise deployment. While this deployment model successfully addresses data sovereignty requirements, it introduces specific technical limitations across multiple system aspects.

6.3.1. Infrastructure limitations

The on-premise architecture, while secure, lacks the inherent elasticity of cloud deployments, requiring manual resource allocation and configuration. Although OneDrive integration provides a practical solution for automated backups, the system faces potential recovery time limitations when handling large datasets compared to cloud-native disaster recovery solutions.

6.3.2. Operational monitoring

The absence of cloud-native monitoring tools necessitates custom implementations of performance tracking systems. While solutions like Prometheus and Grafana could provide comparable functionality, their implementation requires additional infrastructure and maintenance overhead, impacting the total cost of ownership.

6.3.3. System availability

The current architecture, without redundant hardware or failover capabilities, faces inherent downtime during software updates and hardware maintenance. This constraint affects system reliability and requires careful scheduling of maintenance activities to minimize operational impact.

7. Learning Points and Future Directions

7.1. Learning Points

Maintaining Team Spirit and Morale

One of the key learning points during the project was the value of a motivated and cohesive team. While technical goals and deliverables were essential, the team learned that maintaining a positive work environment directly influenced their productivity and collaboration. Challenges such as tight deadlines and evolving project requirements could have created stress and misalignment. However, prioritizing open communication, celebrating small wins, and fostering mutual respect among team members kept morale high. This lesson underscored that emotional resilience and teamwork are as important as technical expertise in achieving long-term project success.

Effective Stakeholder Management

Balancing the needs and expectations of various stakeholders was a critical learning experience for the team. The project involved interacting with sponsors, end-users, supervisors and evaluators each with their unique perspectives and requirements. The team had to actively listen to feedback, align priorities, and negotiate solutions that addressed all parties' concerns. For example, transitioning from a cloud-based system to an on-premise solution required compromise and recalibration of technical deliverables. This process demonstrated the importance of empathy, adaptability, and clear communication in maintaining stakeholder trust and ensuring the project's success.

Proactive Vision and Innovation

Adopting a proactive approach was pivotal in navigating uncertainties and driving innovation throughout the project. Instead of waiting for challenges to arise, the team anticipated potential obstacles and prepared solutions in advance. For example, designing the system architecture to accommodate future scalability ensured that the solution could evolve with the sponsor's needs. This forward-thinking mindset not only helped overcome immediate technical challenges but also positioned the system as a sustainable and adaptable tool for the sponsor. The team learned that innovation requires not only technical expertise but also the vision to shape solutions that anticipate future demands.

The Role of Testing in Development

Early in the development phase, the team faced issues such as data inconsistencies and functionality mismatches that could have compromised the system’s reliability. By implementing a rigorous testing strategy—including unit, integration, and end-to-end testing—the team ensured the system met both functional and non-functional requirements. This process not only improved the solution’s quality but also instilled a culture of accountability within the team. The lesson learned was that testing is not an afterthought but an integral part of the development lifecycle, ensuring the solution’s robustness and reliability.

Continuous Learning and Improvement

Software development is an iterative process, and this project reinforced the idea that there is always room for improvement. Each sprint brought new challenges and opportunities to refine the solution based on feedback and testing. For instance, integrating OCR capabilities for additional banks required the team to learn new technical tools and adapt the system’s architecture. This iterative approach fostered a growth mindset, where each challenge was seen as an opportunity to improve skills and deliver a better product. The team realized that continuous learning is not just about technical skills but also about adopting a mindset that values growth and adaptability.

Communication and Collaboration

Collaboration and effective communication proved to be essential in navigating the complexities of the project. Regular stand-ups, sprint retrospectives, and open feedback sessions ensured that the team remained aligned despite diverse roles and responsibilities. Effective communication is not limited to the team but also to stakeholders and sponsors, as it plays a crucial role in ensuring that everyone involved in the project is on the same page. By maintaining clear and consistent channels of communication, the team was able to address concerns, clarify requirements, and adapt to feedback in a timely manner.

7.2. Future Directions

Advanced Notification System

Implement a comprehensive alert and email notification system to streamline communication and task tracking among stakeholders.

Cloud Hybrid Deployment

While the current system is on-premise, exploring a hybrid model could enhance scalability and operational monitoring using cloud-native tools. A hybrid deployment would allow critical operations and sensitive data to remain on-premise, meeting security and compliance requirements, while leveraging cloud-based resources for non-sensitive tasks like performance monitoring, analytics, or backup management.

8. Conclusion

Our solution is a funds disbursement management system designed to streamline and automate the disbursement process for ALC's property transactions. By digitizing workflows, automating form generation, and integrating OCR capabilities, our system eliminates the reliance on manual data entry, reduces human error, and accelerates operational efficiency. This transformation reduces the time taken for disbursement processing from 47.5 minutes to under 15 minutes per case while maintaining a 99% accuracy rate. The system not only enhances productivity but also supports ALC in meeting stringent compliance and security requirements through an enterprise-grade, on-premise deployment. By addressing ALC's operational pain points, our solution delivers a robust, scalable, and efficient answer to their business challenges.

References

Dennis, P. (2006). *Getting the right things done: A leader's guide to planning and execution.* Lean Enterprise Institute.

ISO/IEC. (2011). *ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.* Retrieved from https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Appendices

Appendix A. On-site Discovery

Appendix B. Current Dataset and Data Model

Given that ALC currently relies on manual and paper-based processes, representing their current datasets and data models requires acknowledging the absence of digital data structures, but rather, the manual and physical nature of their information management.

Current Datasets

Physical Client Files	Physical folders containing all relevant client information, legal documents, correspondence, case details (inclusive of completed disbursement forms)
Digitised Client Files	Digitised format of the above, which may include less or more documents

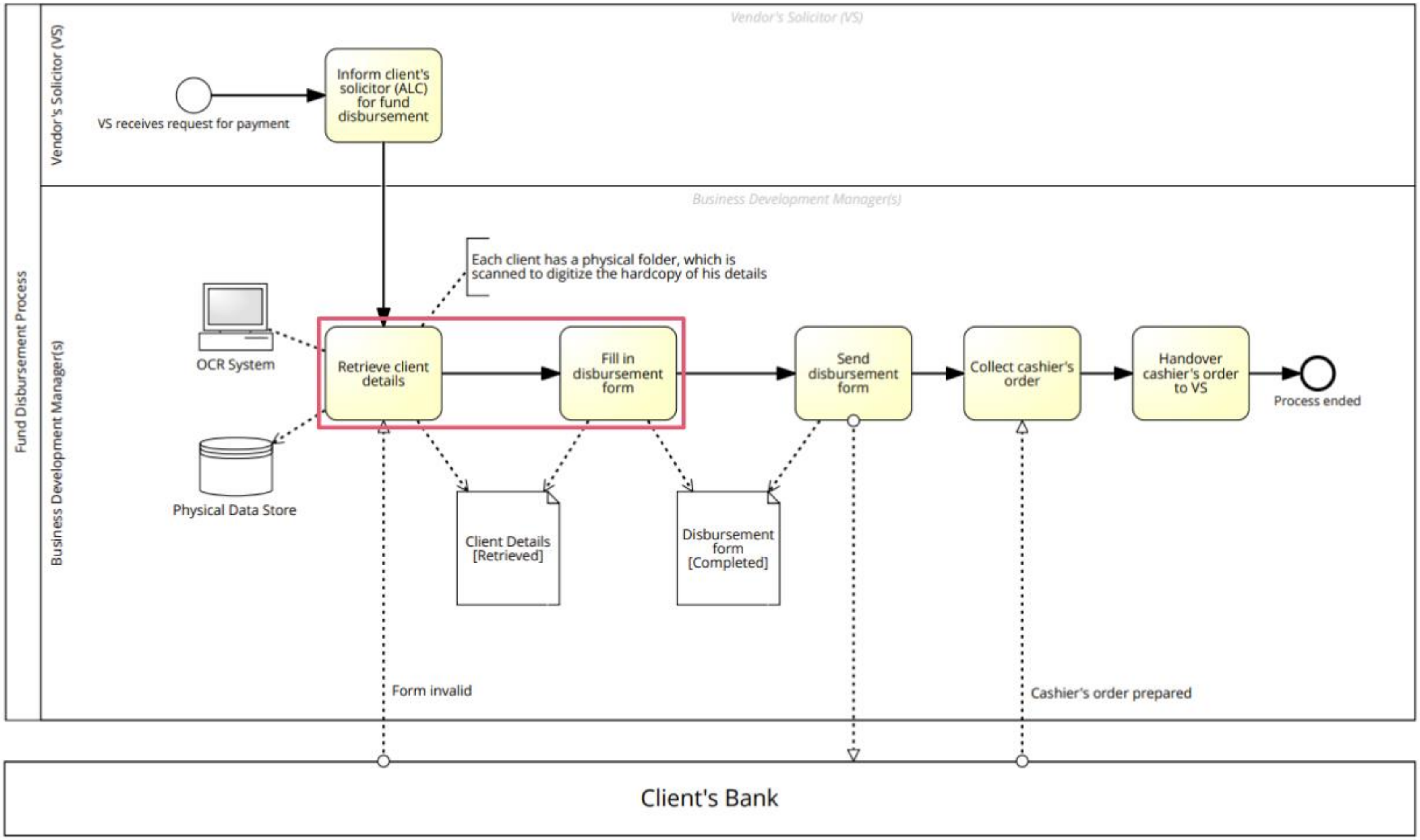
Current Data Models

Physical Filing System	Physical client files are sorted by property name, followed by property address
Digitised Client Files	Digital folder stored by the respective Secretary-In-Charge (SIC)
Manual Indexing System	Paper list of entries of files pertaining to property name and location of files. Staff reference the index to locate files
Disbursement Form Templates	Editable template PDF forms that are to be filled out electronically and sent back to the respective bank for processing

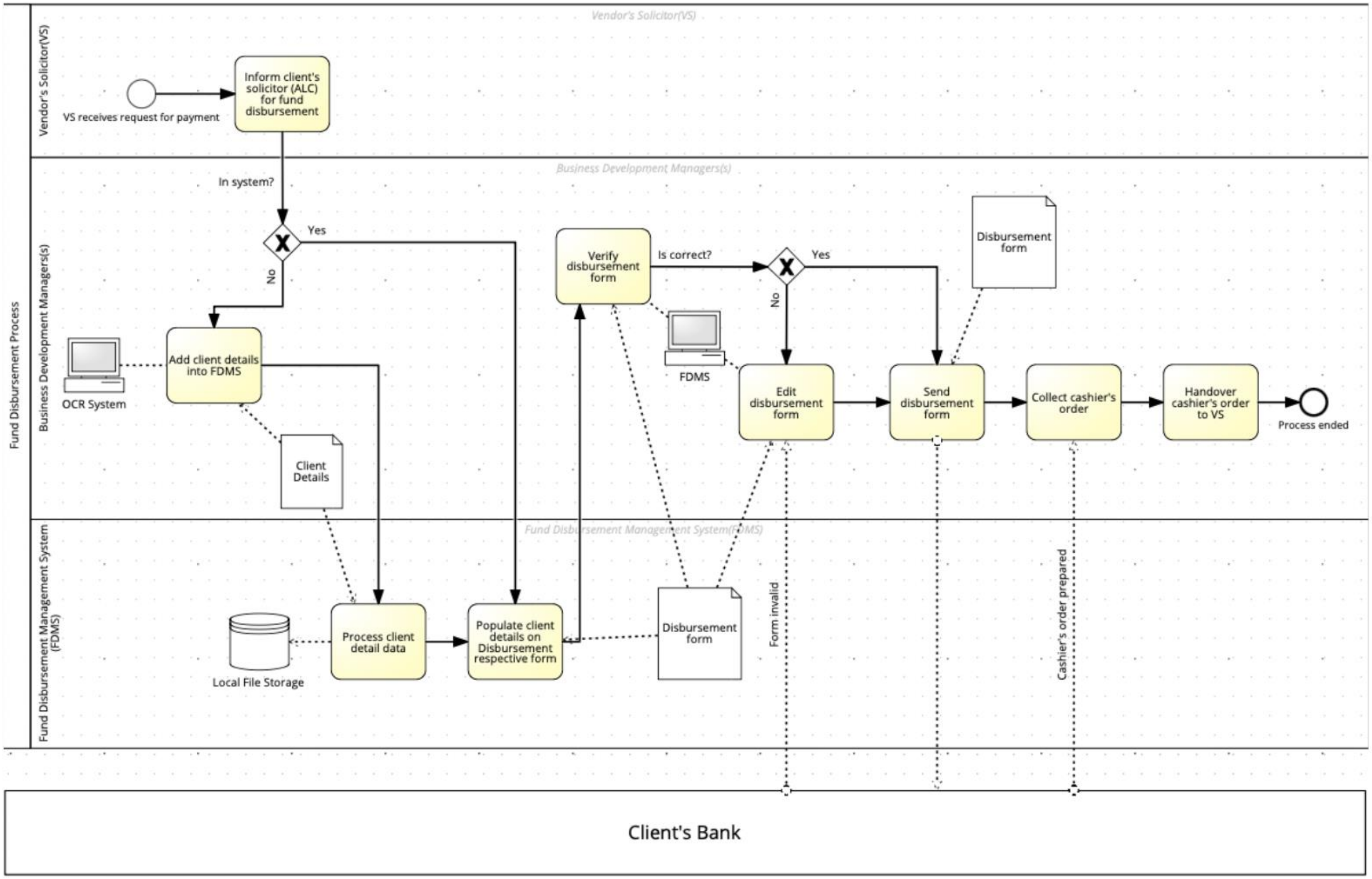
Appendix C. Innovation Summary

	Old	New
What? **Value proposition**	The law firm prides itself on its meticulous funds disbursement process, which ensures a high level of accuracy and personal oversight. The experienced team carefully reviews and processes each disbursement request, extracting data from physical folders and scanned documents to fill out necessary forms by hand. This hands-on approach allows them to catch potential errors and address them promptly, maintaining a high standard of financial integrity and accountability. By personally managing each step, the firm provides a customised and thorough service that builds trust and reliability with clients. Their commitment to detailed record-keeping and manual verification processes ensures that every transaction is handled with the utmost care and precision.	Our solution streamlines the Law Firm's funds disbursement process by automating data extraction and form-filling, addressing key pain points effectively. By digitising data, it **eliminates** the **need to sift through** physical folders and scanned documents. This automation also **reduces errors** inherent in manual data entry, **ensuring accuracy** and **consistency** in form-filling. As a result, our solution **minimises process delays**, **accelerates workflow**, and **decreases operational costs** associated with form resubmissions. These improvements significantly enhance efficiency and streamline workflow management within the firm. Furthermore, features on the cloud-based platform such as robust search, drag-and-drop uploads, and an activity log enhances usability and operational transparency.
Who? Customer Segment	Clients who engage ALC in purchasing or selling a property.	No Change
How? Business Process	AS-IS Workflow Model	TO-BE Workflow Model
Value? Revenue model	**Revenue Stream:** The law firm generates revenue through a percentage fee per case. This fee is applied to each disbursement transaction managed by the firm. **Cost Structure:** The primary cost is the labour involved in the manual processing of funds disbursement. This includes time spent on data extraction, form filling, verification, and record-keeping. **Overhead Costs:** Additional costs include office supplies, storage for physical folders, and potential overtime pay for extended processing times.	**Revenue Stream:** The revenue stream remains unchanged. The firm continues to generate revenue through a percentage fee per case. **Cost Structure:** The primary new cost is the maintenance fee for the funds disbursement software solution. This cost could be a lump sum, a monthly or annual fee, depending on the final pricing model.

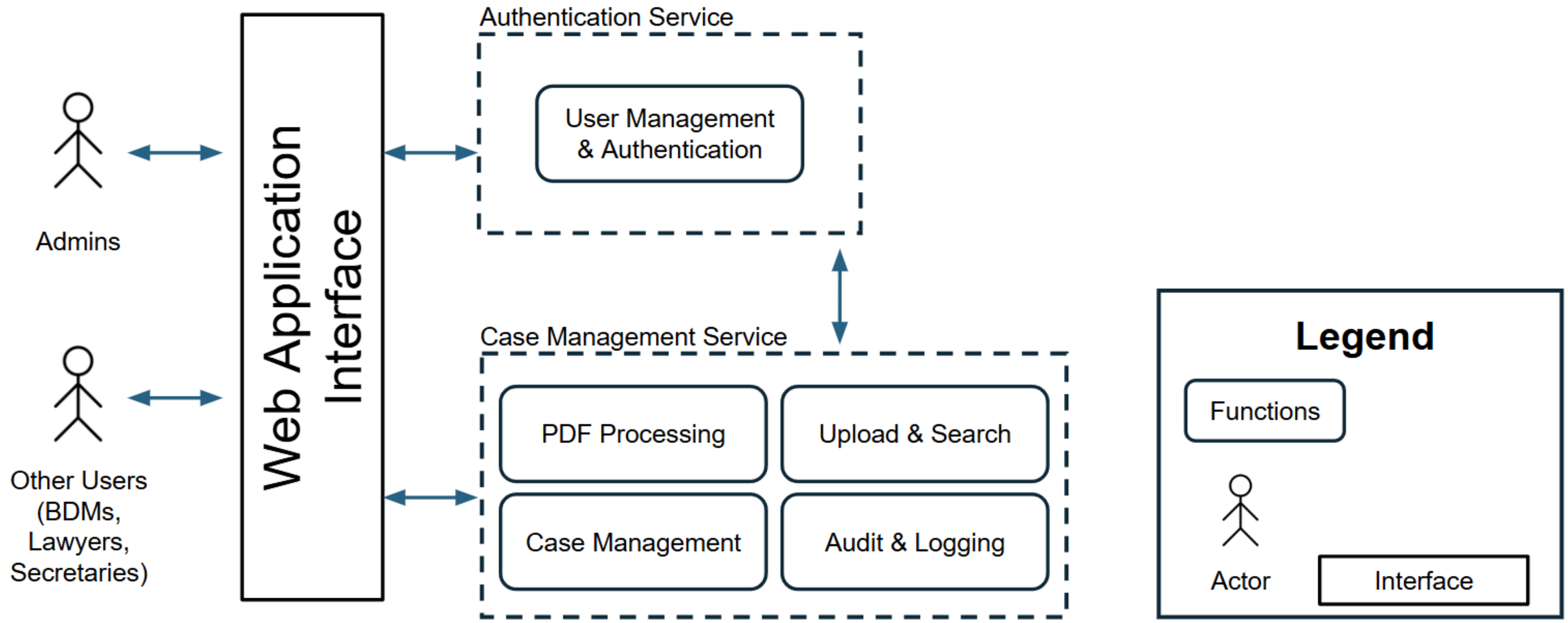
Appendix D. As-is Process Model


Fund Disbursement Process
Vendor's Solicitor (VS)
Vendor's Solicitor (VS)
VS receives request for payment
Inform client's solicitor (ALC) for fund disbursement
Business Development Manager(s)
Business Development Manager(s)
Each client has a physical folder, which is scanned to digitize the hardcopy of his details
OCR System
Physical Data Store
Retrieve client details
Fill in disbursement form
Send disbursement form
Collect cashier's order
Handover cashier's order to VS
Process ended
Client Details [Retrieved]
Disbursement form [Completed]
Form invalid
Cashier's order prepared
Client's Bank

Appendix E. To-Be Process Model


Fund Disbursement Process
Vendor's Solicitor(VS)
Vendor's Solicitor(VS)
VS receives request for payment
Inform client's solicitor (ALC) for fund disbursement
Business Development Managers(s)
Business Development Managers(s)
In system?
Yes
No
OCR System
Add client details into FDMS
Client Details
Verify disbursement form
Is correct?
Yes
No
FDMS
Edit disbursement form
Disbursement form
Send disbursement form
Collect cashier's order
Handover cashier's order to VS
Process ended
Fund Disbursement Management System (FDMS)
Fund Disbursement Management System(FDMS)
Local File Storage
Process client detail data
Populate client details on Disbursement respective form
Disbursement form
Form invalid
Cashier's order prepared
Client's Bank

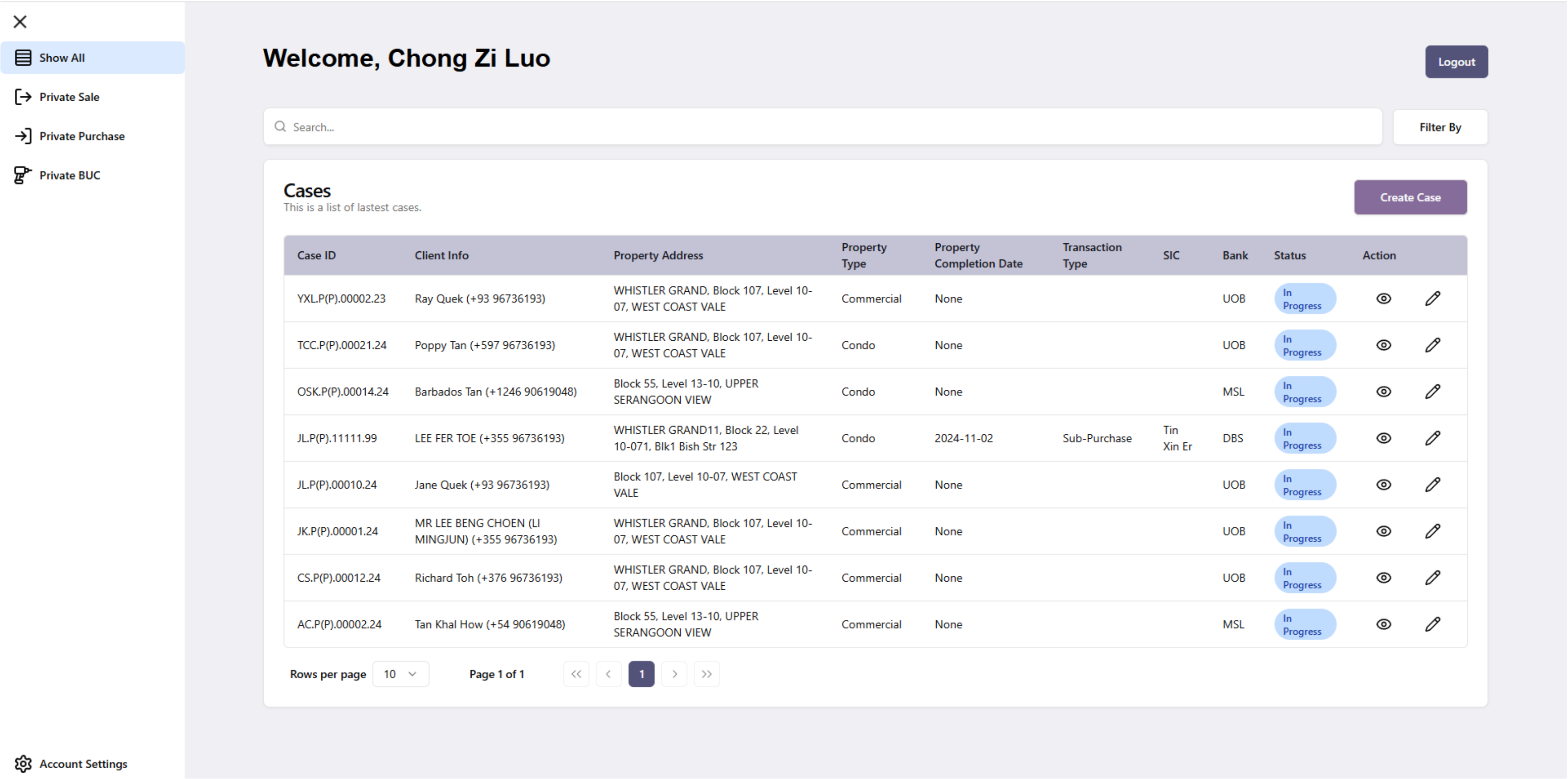
Appendix F. Solution Overview Model



Appendix G. Technology Stack

Layer	Technologies	Justification
Frontend	React, Typescript, Tailwind CSS, Next.js, ShadCN	Modern UI development, rapid styling capabilities
Backend	Node.js, Express, FastAPI	Robust API development, extensive ecosystem
Database	MongoDB	ACID compliance, Document-oriented storage, flexible schema
Caching	Redis	In-memory caching, session management
PDF Processing	pymupdf, openpyxl, python-docx	Document generation, text extraction
Authentication	JWT, bcrypt, RSA	Secure token-based auth, password hashing, Encryption
File Storage	Local File System	On-premise requirement compliance
Deployment	Docker Compose	Consistent Deployment and Scaling

Appendix H. Case Dashboard UI



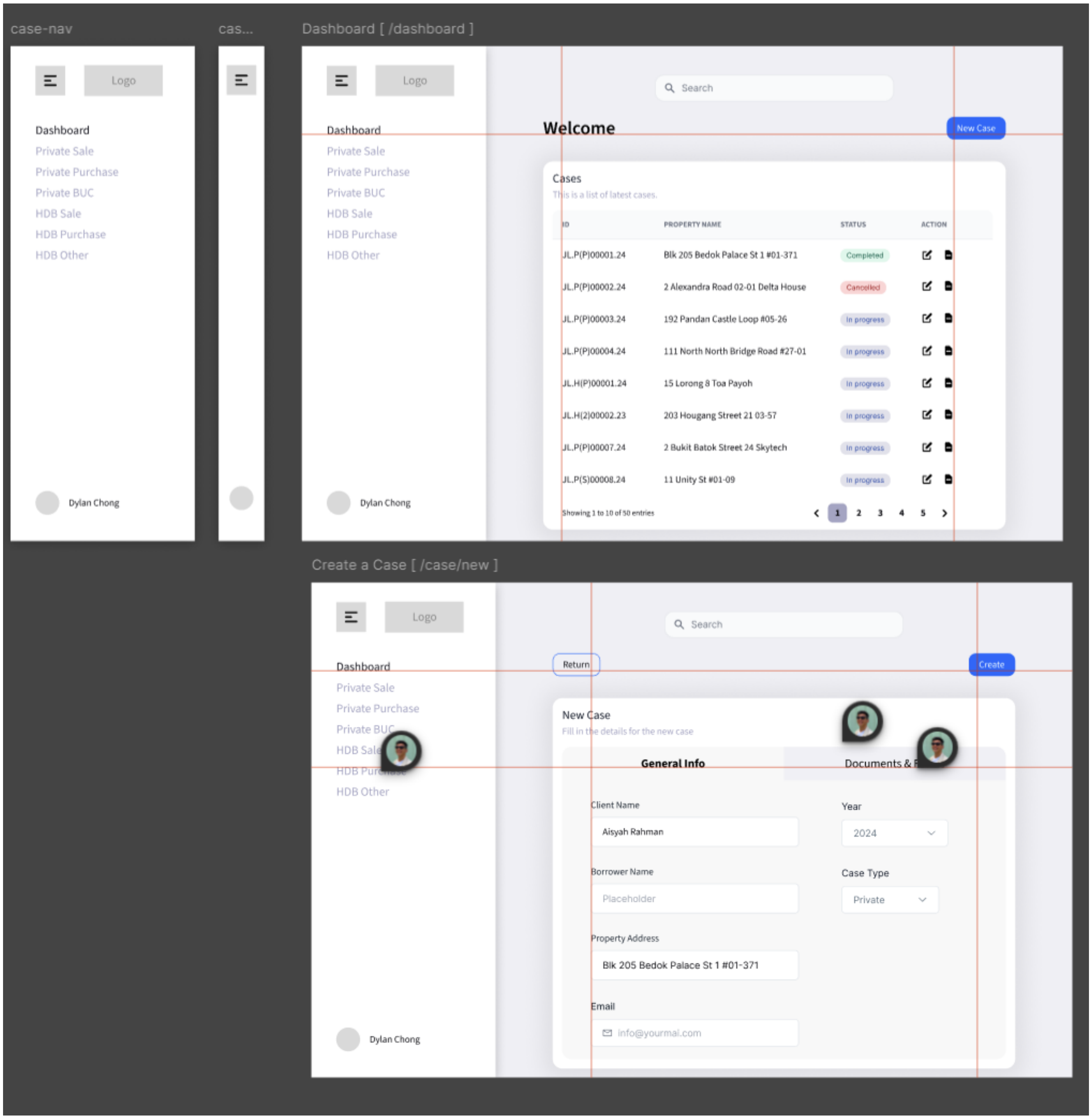
Welcome, Chong Zi Luo

Cases

This is a list of lastest cases.

Case ID	Client Info	Property Address	Property Type	Property Completion Date	Transaction Type	SIC	Bank	Status	Action
YXL.P(P).00002.23	Ray Quek (+93 96736193)	WHISTLER GRAND, Block 107, Level 10-07, WEST COAST VALE	Commercial	None			UOB	In Progress	👁 ✎
TCC.P(P).00021.24	Poppy Tan (+597 96736193)	WHISTLER GRAND, Block 107, Level 10-07, WEST COAST VALE	Condo	None			UOB	In Progress	👁 ✎
OSK.P(P).00014.24	Barbados Tan (+1246 90619048)	Block 55, Level 13-10, UPPER SERANGOON VIEW	Condo	None			MSL	In Progress	👁 ✎
JL.P(P).11111.99	LEE FER TOE (+355 96736193)	WHISTLER GRAND11, Block 22, Level 10-071, Blk1 Bish Str 123	Condo	2024-11-02	Sub-Purchase	Tin Xin Er	DBS	In Progress	👁 ✎
JL.P(P).00010.24	Jane Quek (+93 96736193)	Block 107, Level 10-07, WEST COAST VALE	Commercial	None			UOB	In Progress	👁 ✎
JK.P(P).00001.24	MR LEE BENG CHOEN (LI MINGJUN) (+355 96736193)	WHISTLER GRAND, Block 107, Level 10-07, WEST COAST VALE	Commercial	None			UOB	In Progress	👁 ✎
CS.P(P).00012.24	Richard Toh (+376 96736193)	WHISTLER GRAND, Block 107, Level 10-07, WEST COAST VALE	Commercial	None			UOB	In Progress	👁 ✎
AC.P(P).00002.24	Tan Khal How (+54 90619048)	Block 55, Level 13-10, UPPER SERANGOON VIEW	Commercial	None			MSL	In Progress	👁 ✎

Appendix I. Detailed UI Components



Welcome

Cases

This is a list of latest cases.

ID	PROPERTY NAME	STATUS	ACTION
JL.P(P)00001.24	Blk 205 Bedok Palace St 1 #01-371	Completed	
JL.P(P)00002.24	2 Alexandra Road 02-01 Delta House	Cancelled	
JL.P(P)00003.24	192 Pandan Castle Loop #05-26	In progress	
JL.P(P)00004.24	111 North North Bridge Road #27-01	In progress	
JL.H(P)00001.24	15 Lorong 8 Toa Payoh	In progress	
JL.H(2)00002.23	203 Hougang Street 21 03-57	In progress	
JL.P(P)00007.24	2 Bukit Batok Street 24 Skytech	In progress	
JL.P(S)00008.24	11 Unity St #01-09	In progress	

Showing 1 to 10 of 50 entries ‹ 1 2 3 4 5 ›

New Case

Fill in the details for the new case

General Info | Documents & F

Client Name: Aisyah Rahman

Year: 2024

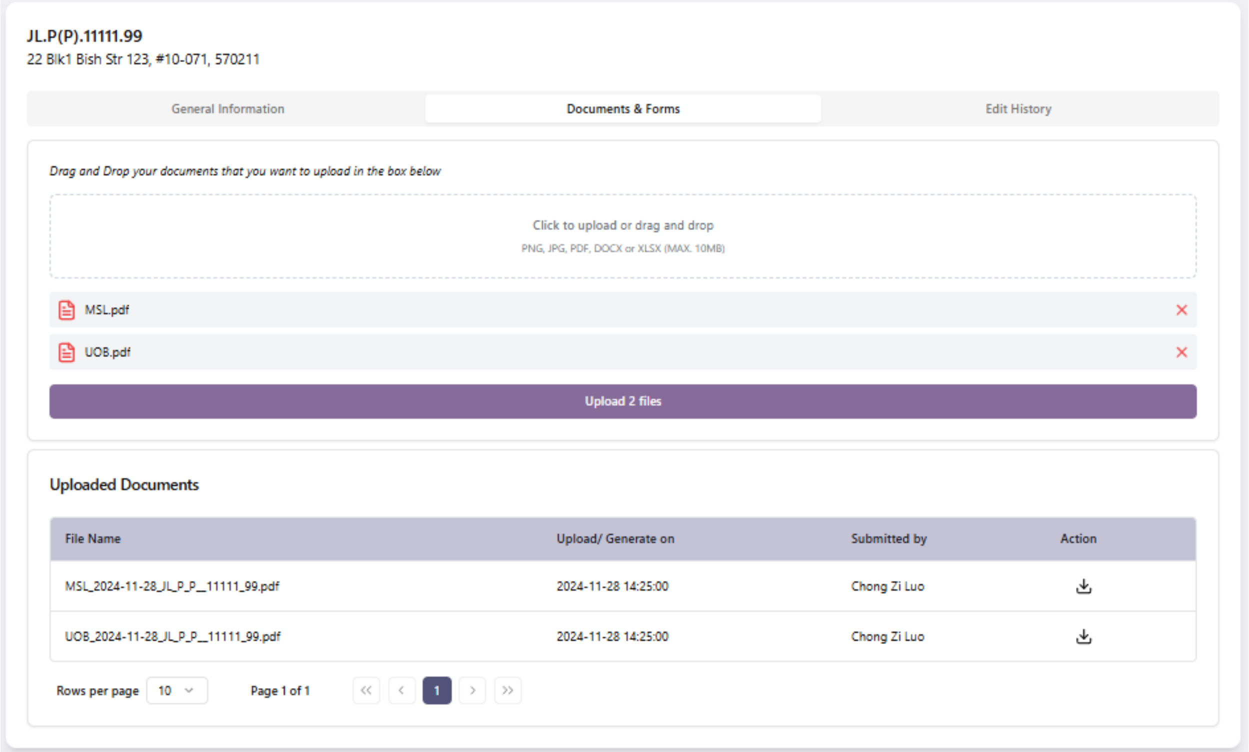
Borrower Name: Placeholder

Case Type: Private

Property Address: Blk 205 Bedok Palace St 1 #01-371

Email: info@yourmai.com

Appendix J. Document Upload UI



JL.P(P).11111.99

22 Blk1 Bish Str 123, #10-071, 570211

Drag and Drop your documents that you want to upload in the box below

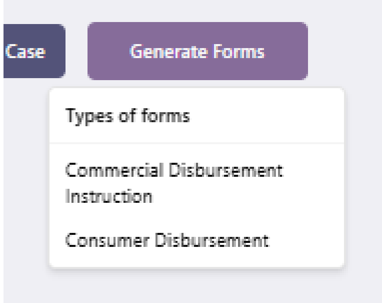
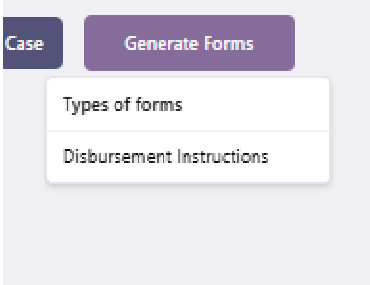
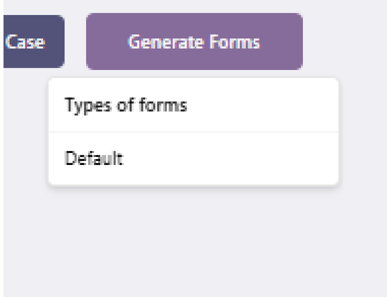
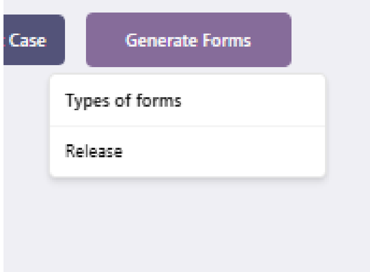
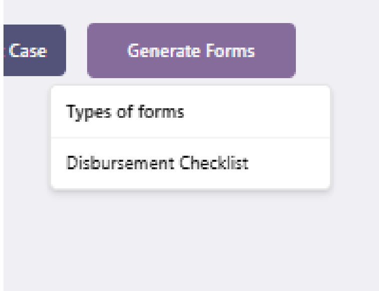
MSL.pdf ✕

UOB.pdf ✕

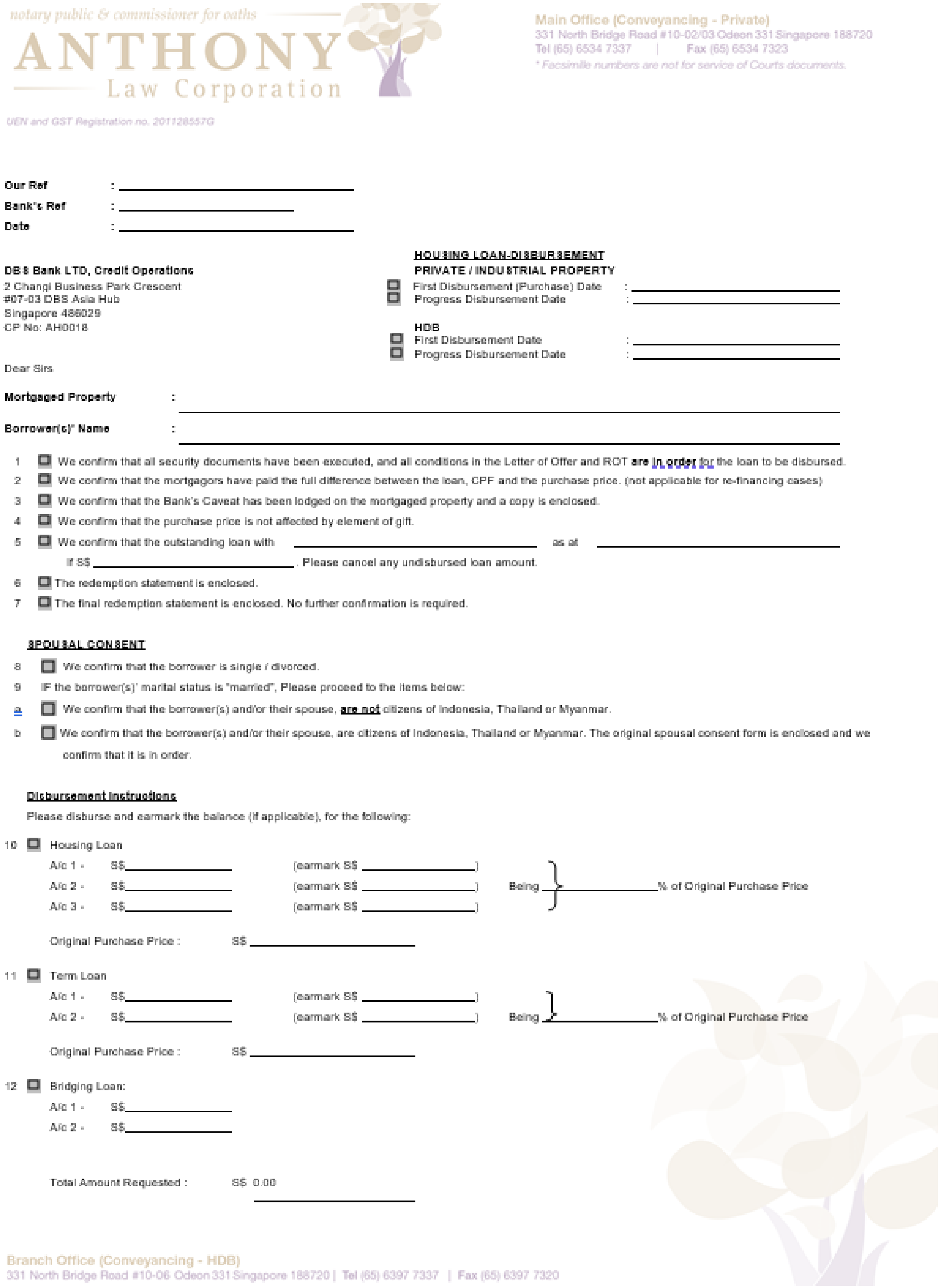
Uploaded Documents

File Name	Upload/ Generate on	Submitted by	Action
MSL_2024-11-28_JL_P_P__11111_99.pdf	2024-11-28 14:25:00	Chong Zi Luo	⤓
UOB_2024-11-28_JL_P_P__11111_99.pdf	2024-11-28 14:25:00	Chong Zi Luo	⤓

Appendix K. Form Generation Options

Maybank	 Case Generate Forms Types of forms Commercial Disbursement Instruction Consumer Disbursement	**DBS**	 Case Generate Forms Types of forms Disbursement Instructions
UOB		**Citibank**	
HSBC	 Case Generate Forms Types of forms Disbursement Checklist		

Appendix L. Example of a Predefined Bank Template



notary public & commissioner for oaths
ANTHONY
Law Corporation

UEN and GST Registration no. 201128557G

Main Office (Conveyancing - Private)
331 North Bridge Road #10-02/03 Odeon 331 Singapore 188720
Tel (65) 6534 7337 | Fax (65) 6534 7323
* Facsimile numbers are not for service of Courts documents.

Our Ref : ____________________
Bank's Ref : ________________
Date : ____________________

DBS Bank LTD, Credit Operations
2 Changi Business Park Crescent
#07-03 DBS Asia Hub
Singapore 486029
CP No: AH0018

HOUSING LOAN-DISBURSEMENT
PRIVATE / INDUSTRIAL PROPERTY
□ First Disbursement (Purchase) Date : ____________________
□ Progress Disbursement Date : ____________________

HDB
□ First Disbursement Date : ____________________
□ Progress Disbursement Date : ____________________

Dear Sirs

Mortgaged Property : ____

Borrower(s)' Name : ____

1 □ We confirm that all security documents have been executed, and all conditions in the Letter of Offer and ROT are in order for the loan to be disbursed.
2 □ We confirm that the mortgagors have paid the full difference between the loan, CPF and the purchase price. (not applicable for re-financing cases)
3 □ We confirm that the Bank's Caveat has been lodged on the mortgaged property and a copy is enclosed.
4 □ We confirm that the purchase price is not affected by element of gift.
5 □ We confirm that the outstanding loan with ____________________ as at ____________________
is S$ ________________ . Please cancel any undisbursed loan amount.
6 □ The redemption statement is enclosed.
7 □ The final redemption statement is enclosed. No further confirmation is required.

SPOUSAL CONSENT

8 □ We confirm that the borrower is single / divorced.
9 IF the borrower(s)' marital status is "married", Please proceed to the items below:
a □ We confirm that the borrower(s) and/or their spouse, **are not** citizens of Indonesia, Thailand or Myanmar.
b □ We confirm that the borrower(s) and/or their spouse, are citizens of Indonesia, Thailand or Myanmar. The original spousal consent form is enclosed and we confirm that it is in order.

Disbursement Instructions

Please disburse and earmark the balance (if applicable), for the following:

10 □ Housing Loan
A/c 1 - S$____________ (earmark S$ ____________)
A/c 2 - S$____________ (earmark S$ ____________) Being } ____________% of Original Purchase Price
A/c 3 - S$____________ (earmark S$ ____________)

Original Purchase Price : S$ ________________

11 □ Term Loan
A/c 1 - S$____________ (earmark S$ ____________)
A/c 2 - S$____________ (earmark S$ ____________) Being } ____________% of Original Purchase Price

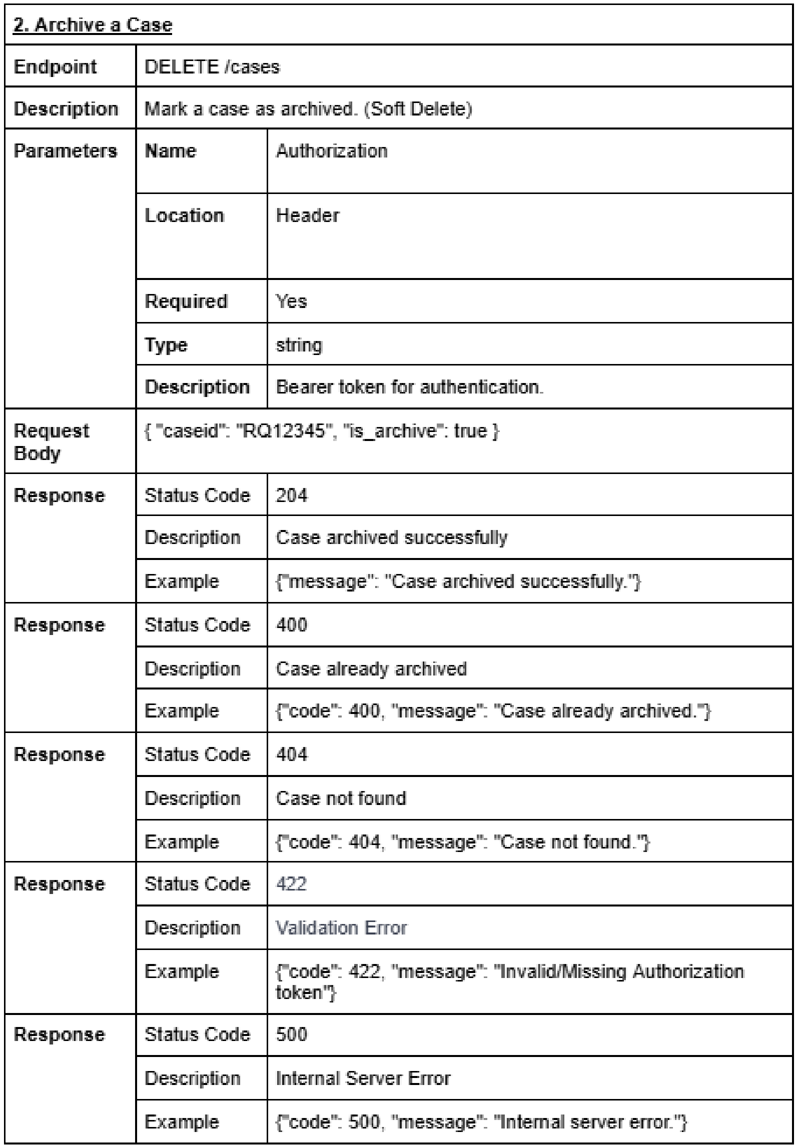
Original Purchase Price : S$ ________________

12 □ Bridging Loan:
A/c 1 - S$____________
A/c 2 - S$____________

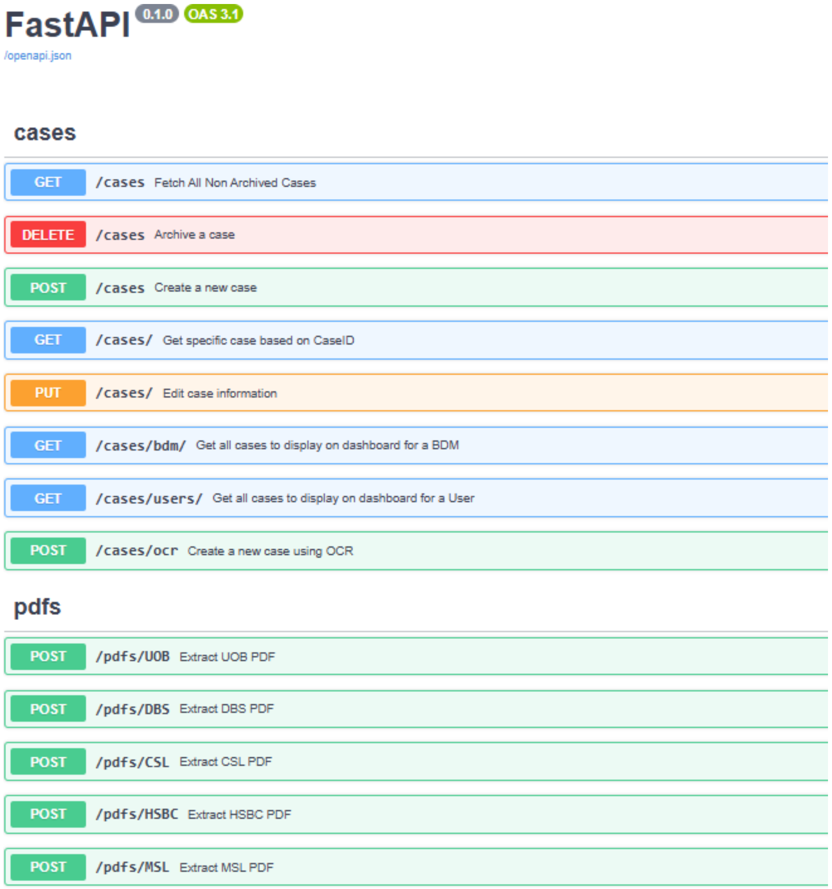
Total Amount Requested : S$ 0.00

Branch Office (Conveyancing - HDB)
331 North Bridge Road #10-06 Odeon 331 Singapore 188720 | Tel (65) 6397 7337 | Fax (65) 6397 7320

Appendix M. Sample API Documentation



2. Archive a Case		
Endpoint	DELETE /cases	
Description	Mark a case as archived. (Soft Delete)	
Parameters	**Name**	Authorization
	Location	Header
	Required	Yes
	Type	string
	Description	Bearer token for authentication.
Request Body	{ "caseid": "RQ12345", "is_archive": true }	
Response	Status Code	204
	Description	Case archived successfully
	Example	{"message": "Case archived successfully."}
Response	Status Code	400
	Description	Case already archived
	Example	{"code": 400, "message": "Case already archived."}
Response	Status Code	404
	Description	Case not found
	Example	{"code": 404, "message": "Case not found."}
Response	Status Code	422
	Description	Validation Error
	Example	{"code": 422, "message": "Invalid/Missing Authorization token"}
Response	Status Code	500
	Description	Internal Server Error
	Example	{"code": 500, "message": "Internal server error."}



FastAPI 0.1.0 OAS 3.1

/openapi.json

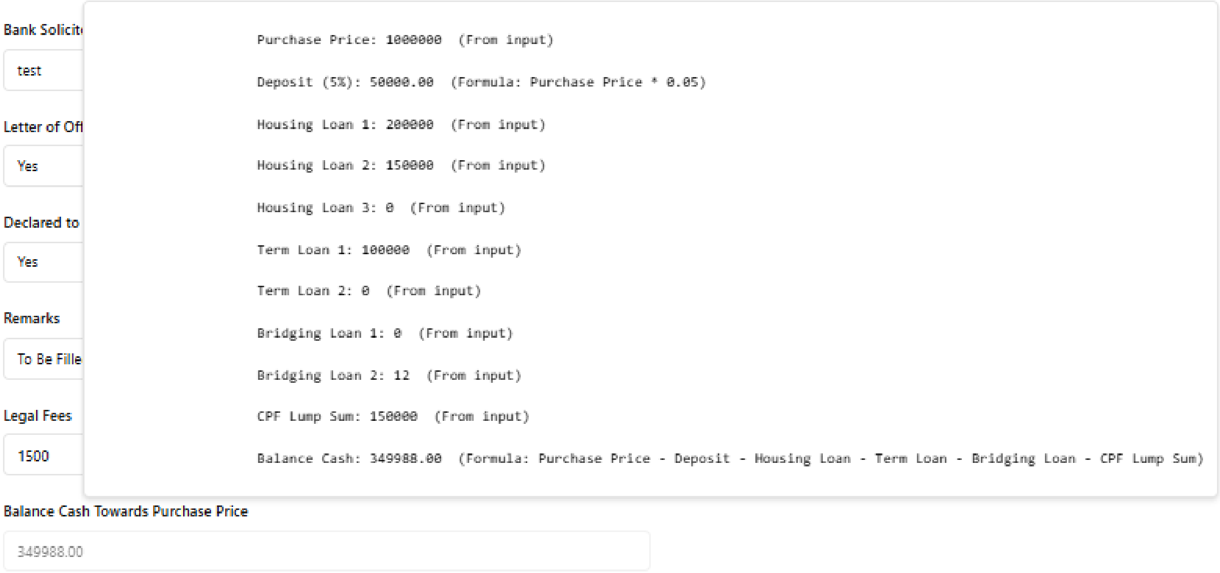
cases

GET	`/cases`	Fetch All Non Archived Cases
DELETE	`/cases`	Archive a case
POST	`/cases`	Create a new case
GET	`/cases/`	Get specific case based on CaseID
PUT	`/cases/`	Edit case information
GET	`/cases/bdm/`	Get all cases to display on dashboard for a BDM
GET	`/cases/users/`	Get all cases to display on dashboard for a User
POST	`/cases/ocr`	Create a new case using OCR

pdfs

POST	`/pdfs/UOB`	Extract UOB PDF
POST	`/pdfs/DBS`	Extract DBS PDF
POST	`/pdfs/CSL`	Extract CSL PDF
POST	`/pdfs/HSBC`	Extract HSBC PDF
POST	`/pdfs/MSL`	Extract MSL PDF

Appendix N. Formulae Calculation & Tooltips



Bank Solicit

test

Letter of Off

Yes

Declared to

Yes

Remarks

To Be Fille

Legal Fees

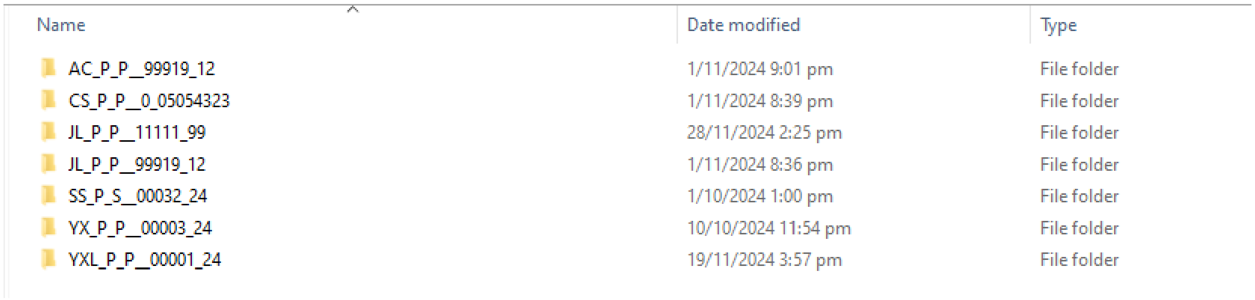
1500

Balance Cash Towards Purchase Price

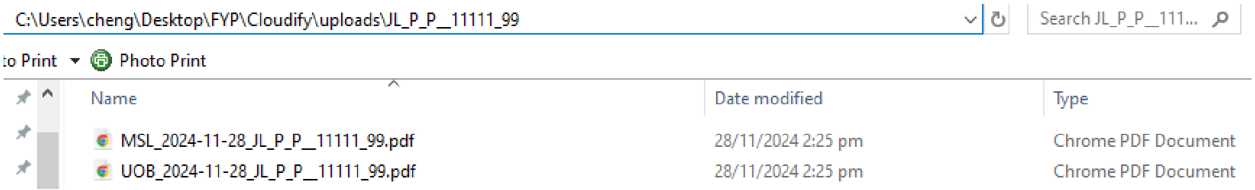
349988.00

```
Purchase Price: 1000000  (From input)

Deposit (5%): 50000.00  (Formula: Purchase Price * 0.05)

Housing Loan 1: 200000  (From input)

Housing Loan 2: 150000  (From input)

Housing Loan 3: 0  (From input)

Term Loan 1: 100000  (From input)

Term Loan 2: 0  (From input)

Bridging Loan 1: 0  (From input)

Bridging Loan 2: 12  (From input)

CPF Lump Sum: 150000  (From input)

Balance Cash: 349988.00  (Formula: Purchase Price - Deposit - Housing Loan - Term Loan - Bridging Loan - CPF Lump Sum)
```

Appendix O. Supporting Document Storage



Name	Date modified	Type
AC_P_P__99919_12	1/11/2024 9:01 pm	File folder
CS_P_P__0_05054323	1/11/2024 8:39 pm	File folder
JL_P_P__11111_99	28/11/2024 2:25 pm	File folder
JL_P_P__99919_12	1/11/2024 8:36 pm	File folder
SS_P_S__00032_24	1/10/2024 1:00 pm	File folder
YX_P_P__00003_24	10/10/2024 11:54 pm	File folder
YXL_P_P__00001_24	19/11/2024 3:57 pm	File folder

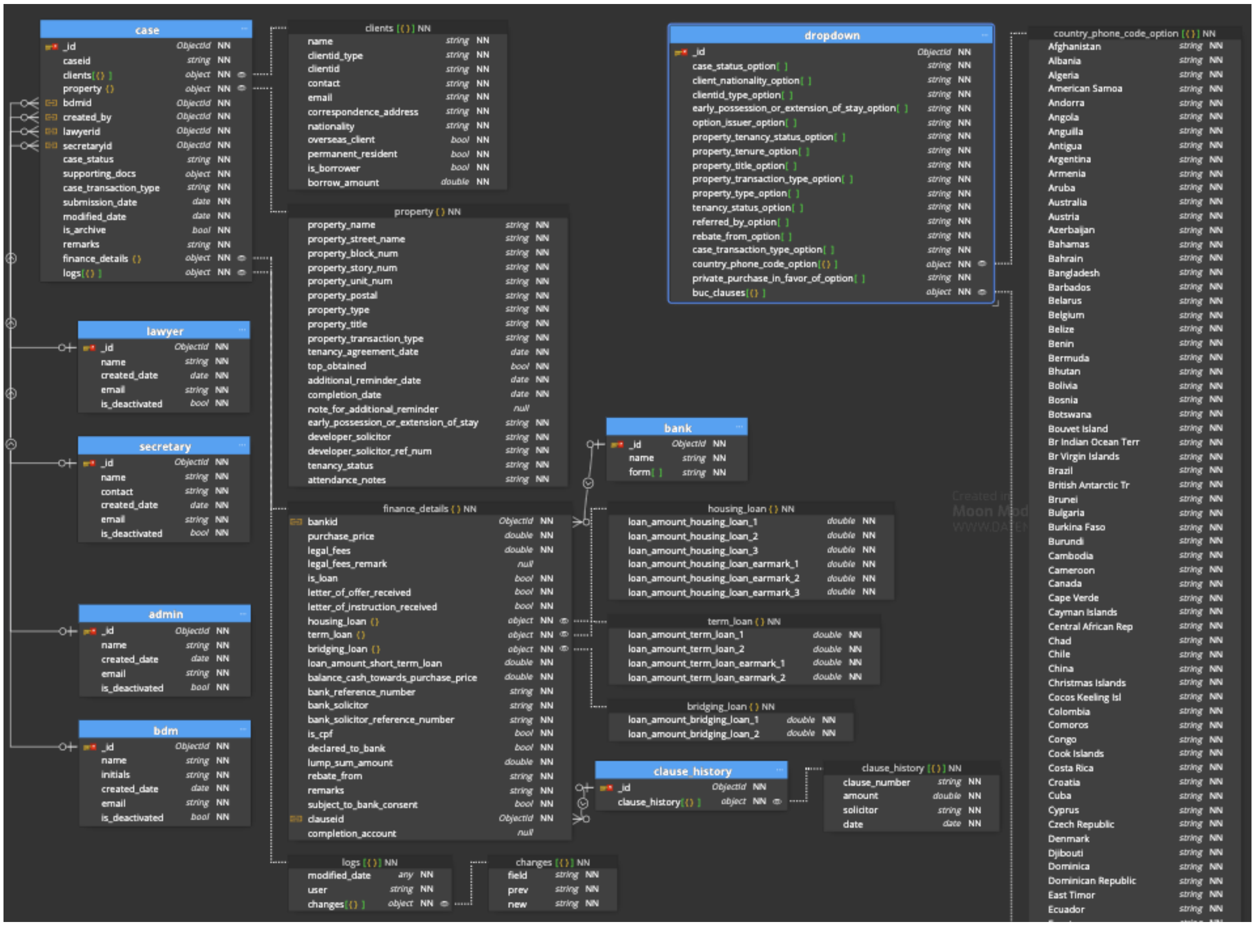


C:\Users\cheng\Desktop\FYP\Cloudify\uploads\JL_P_P__11111_99

Name	Date modified	Type
MSL_2024-11-28_JL_P_P__11111_99.pdf	28/11/2024 2:25 pm	Chrome PDF Document
UOB_2024-11-28_JL_P_P__11111_99.pdf	28/11/2024 2:25 pm	Chrome PDF Document

Appendix P. Database Model

Main Database



case

_id	ObjectId	NN
caseid	string	NN
clients[{}]	object	NN
property {}	object	NN
bdmid	ObjectId	NN
created_by	ObjectId	NN
lawyerid	ObjectId	NN
secretaryid	ObjectId	NN
case_status	string	NN
supporting_docs	object	NN
case_transaction_type	string	NN
submission_date	date	NN
modified_date	date	NN
is_archive	bool	NN
remarks	string	NN
finance_details {}	object	NN
logs[{}]	object	NN

clients [{}] NN

name	string	NN
clientid_type	string	NN
clientid	string	NN
contact	string	NN
email	string	NN
correspondence_address	string	NN
nationality	string	NN
overseas_client	bool	NN
permanent_resident	bool	NN
is_borrower	bool	NN
borrow_amount	double	NN

property {} NN

property_name	string	NN
property_street_name	string	NN
property_block_num	string	NN
property_story_num	string	NN
property_unit_num	string	NN
property_postal	string	NN
property_type	string	NN
property_title	string	NN
property_transaction_type	string	NN
tenancy_agreement_date	date	NN
top_obtained	bool	NN
additional_reminder_date	date	NN
completion_date	date	NN
note_for_additional_reminder	null	
early_possession_or_extension_of_stay	string	NN
developer_solicitor	string	NN
developer_solicitor_ref_num	string	NN
tenancy_status	string	NN
attendance_notes	string	NN

finance_details {} NN

bankid	ObjectId	NN
purchase_price	double	NN
legal_fees	double	NN
legal_fees_remark	null	
is_loan	bool	NN
letter_of_offer_received	bool	NN
letter_of_instruction_received	bool	NN
housing_loan {}	object	NN
term_loan {}	object	NN
bridging_loan {}	object	NN
loan_amount_short_term_loan	double	NN
balance_cash_towards_purchase_price	double	NN
bank_reference_number	string	NN
bank_solicitor	string	NN
bank_solicitor_reference_number	string	NN
is_cpf	bool	NN
declared_to_bank	bool	NN
lump_sum_amount	double	NN
rebate_from	string	NN
remarks	string	NN
subject_to_bank_consent	bool	NN
clauseid	ObjectId	NN
completion_account	null	

logs [{}] NN

modified_date	any	NN
user	string	NN
changes[{}]	object	NN

changes [{}] NN

field	string	NN
prev	string	NN
new	string	NN

lawyer

_id	ObjectId	NN
name	string	NN
created_date	date	NN
email	string	NN
is_deactivated	bool	NN

secretary

_id	ObjectId	NN
name	string	NN
contact	string	NN
created_date	date	NN
email	string	NN
is_deactivated	bool	NN

admin

_id	ObjectId	NN
name	string	NN
created_date	date	NN
email	string	NN
is_deactivated	bool	NN

bdm

_id	ObjectId	NN
name	string	NN
initials	string	NN
created_date	date	NN
email	string	NN
is_deactivated	bool	NN

dropdown

_id	ObjectId	NN
case_status_option[ ]	string	NN
client_nationality_option[ ]	string	NN
clientid_type_option[ ]	string	NN
early_possession_or_extension_of_stay_option[ ]	string	NN
option_issuer_option[ ]	string	NN
property_tenancy_status_option[ ]	string	NN
property_tenure_option[ ]	string	NN
property_title_option[ ]	string	NN
property_transaction_type_option[ ]	string	NN
property_type_option[ ]	string	NN
tenancy_status_option[ ]	string	NN
referred_by_option[ ]	string	NN
rebate_from_option[ ]	string	NN
case_transaction_type_option[ ]	string	NN
country_phone_code_option[{}]	object	NN
private_purchase_in_favor_of_option[ ]	string	NN
buc_clauses[{}]	object	NN

bank

_id	ObjectId	NN
name	string	NN
form[ ]	string	NN

housing_loan {} NN

loan_amount_housing_loan_1	double	NN
loan_amount_housing_loan_2	double	NN
loan_amount_housing_loan_3	double	NN
loan_amount_housing_loan_earmark_1	double	NN
loan_amount_housing_loan_earmark_2	double	NN
loan_amount_housing_loan_earmark_3	double	NN

term_loan {} NN

loan_amount_term_loan_1	double	NN
loan_amount_term_loan_2	double	NN
loan_amount_term_loan_earmark_1	double	NN
loan_amount_term_loan_earmark_2	double	NN

bridging_loan {} NN

loan_amount_bridging_loan_1	double	NN
loan_amount_bridging_loan_2	double	NN

clause_history

_id	ObjectId	NN
clause_history[{}]	object	NN

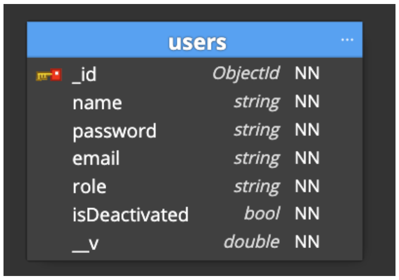
clause_history [{}] NN

clause_number	string	NN
amount	double	NN
solicitor	string	NN
date	date	NN

country_phone_code_option [{}] NN

Afghanistan	string	NN
Albania	string	NN
Algeria	string	NN
American Samoa	string	NN
Andorra	string	NN
Angola	string	NN
Anguilla	string	NN
Antigua	string	NN
Argentina	string	NN
Armenia	string	NN
Aruba	string	NN
Australia	string	NN
Austria	string	NN
Azerbaijan	string	NN
Bahamas	string	NN
Bahrain	string	NN
Bangladesh	string	NN
Barbados	string	NN
Belarus	string	NN
Belgium	string	NN
Belize	string	NN
Benin	string	NN
Bermuda	string	NN
Bhutan	string	NN
Bolivia	string	NN
Bosnia	string	NN
Botswana	string	NN
Bouvet Island	string	NN
Br Indian Ocean Terr	string	NN
Br Virgin Islands	string	NN
Brazil	string	NN
British Antarctic Tr	string	NN
Brunei	string	NN
Bulgaria	string	NN
Burkina Faso	string	NN
Burundi	string	NN
Cambodia	string	NN
Cameroon	string	NN
Canada	string	NN
Cape Verde	string	NN
Cayman Islands	string	NN
Central African Rep	string	NN
Chad	string	NN
Chile	string	NN
China	string	NN
Christmas Islands	string	NN
Cocos Keeling Isl	string	NN
Colombia	string	NN
Comoros	string	NN
Congo	string	NN
Cook Islands	string	NN
Costa Rica	string	NN
Croatia	string	NN
Cuba	string	NN
Cyprus	string	NN
Czech Republic	string	NN
Denmark	string	NN
Djibouti	string	NN
Dominica	string	NN
Dominican Republic	string	NN
East Timor	string	NN
Ecuador	string	NN

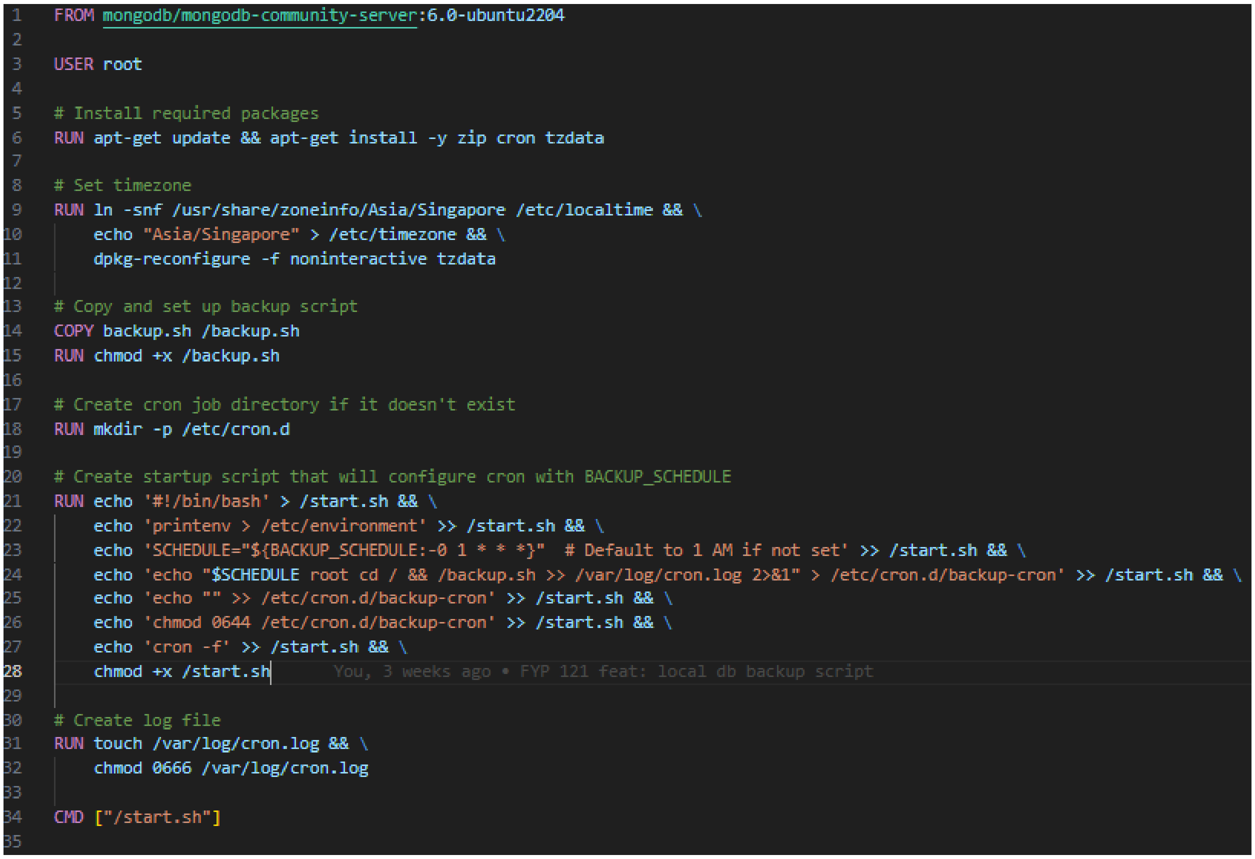
Authentication Database



users

_id	ObjectId	NN
name	string	NN
password	string	NN
email	string	NN
role	string	NN
isDeactivated	bool	NN
__v	double	NN

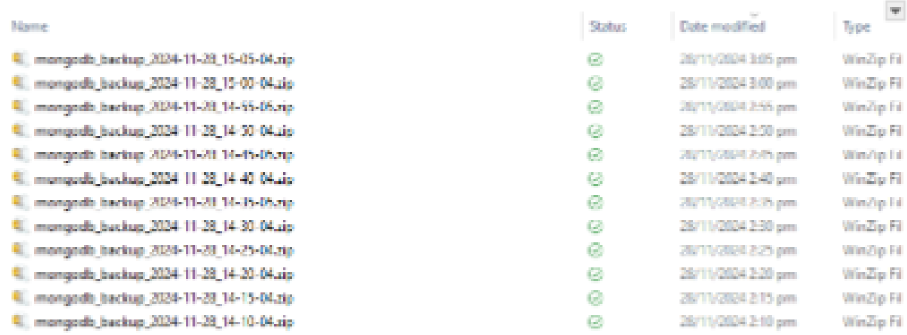
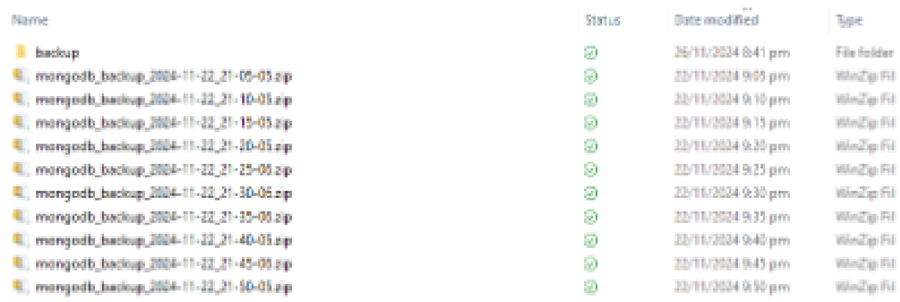
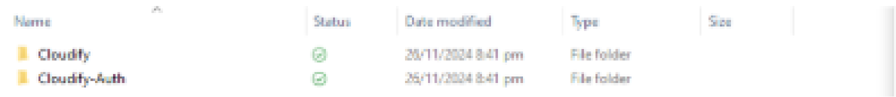
Appendix Q. Detailed Backup CRON Config

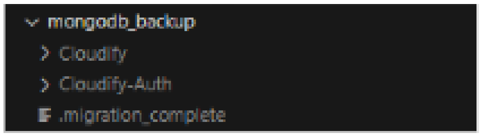


```
FROM mongodb/mongodb-community-server:6.0-ubuntu2204

USER root

# Install required packages
RUN apt-get update && apt-get install -y zip cron tzdata

# Set timezone
RUN ln -snf /usr/share/zoneinfo/Asia/Singapore /etc/localtime && \
    echo "Asia/Singapore" > /etc/timezone && \
    dpkg-reconfigure -f noninteractive tzdata

# Copy and set up backup script
COPY backup.sh /backup.sh
RUN chmod +x /backup.sh

# Create cron job directory if it doesn't exist
RUN mkdir -p /etc/cron.d

# Create startup script that will configure cron with BACKUP_SCHEDULE
RUN echo '#!/bin/bash' > /start.sh && \
    echo 'printenv > /etc/environment' >> /start.sh && \
    echo 'SCHEDULE="${BACKUP_SCHEDULE:-0 1 * * *}"  # Default to 1 AM if not set' >> /start.sh && \
    echo 'echo "$SCHEDULE root cd / && /backup.sh >> /var/log/cron.log 2>&1" > /etc/cron.d/backup-cron' >> /start.sh && \
    echo 'echo "" >> /etc/cron.d/backup-cron' >> /start.sh && \
    echo 'chmod 0644 /etc/cron.d/backup-cron' >> /start.sh && \
    echo 'cron -f' >> /start.sh && \
    chmod +x /start.sh       You, 3 weeks ago • FYP 121 feat: local db backup script

# Create log file
RUN touch /var/log/cron.log && \
    chmod 0666 /var/log/cron.log

CMD ["/start.sh"]
```

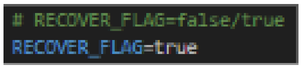
Appendix R. Recovery Procedure

Recovery User Guide

Steps	Description
1.	Open OneDrive localdb_backup folder
	 Name Status Date modified Type mongodb_backup_2024-11-28_15-05-04.zip ✓ 28/11/2024 3:05 pm WinZip Fil mongodb_backup_2024-11-28_15-00-04.zip ✓ 28/11/2024 3:00 pm WinZip Fil mongodb_backup_2024-11-28_14-55-05.zip ✓ 28/11/2024 2:55 pm WinZip Fil mongodb_backup_2024-11-28_14-50-04.zip ✓ 28/11/2024 2:50 pm WinZip Fil mongodb_backup_2024-11-28_14-45-05.zip ✓ 28/11/2024 2:45 pm WinZip Fil mongodb_backup_2024-11-28_14-40-04.zip ✓ 28/11/2024 2:40 pm WinZip Fil mongodb_backup_2024-11-28_14-35-05.zip ✓ 28/11/2024 2:35 pm WinZip Fil mongodb_backup_2024-11-28_14-30-04.zip ✓ 28/11/2024 2:30 pm WinZip Fil mongodb_backup_2024-11-28_14-25-04.zip ✓ 28/11/2024 2:25 pm WinZip Fil mongodb_backup_2024-11-28_14-20-04.zip ✓ 28/11/2024 2:20 pm WinZip Fil mongodb_backup_2024-11-28_14-15-04.zip ✓ 28/11/2024 2:15 pm WinZip Fil mongodb_backup_2024-11-28_14-10-04.zip ✓ 28/11/2024 2:10 pm WinZip Fil
2.	Unzip the latest backup file
	 Name Status Date modified Type backup ✓ 26/11/2024 8:41 pm File folder mongodb_backup_2024-11-22_21-05-05.zip ✓ 22/11/2024 9:05 pm WinZip Fil mongodb_backup_2024-11-22_21-10-05.zip ✓ 22/11/2024 9:10 pm WinZip Fil mongodb_backup_2024-11-22_21-15-05.zip ✓ 22/11/2024 9:15 pm WinZip Fil mongodb_backup_2024-11-22_21-20-05.zip ✓ 22/11/2024 9:20 pm WinZip Fil mongodb_backup_2024-11-22_21-25-06.zip ✓ 22/11/2024 9:25 pm WinZip Fil mongodb_backup_2024-11-22_21-30-06.zip ✓ 22/11/2024 9:30 pm WinZip Fil mongodb_backup_2024-11-22_21-35-06.zip ✓ 22/11/2024 9:35 pm WinZip Fil mongodb_backup_2024-11-22_21-40-05.zip ✓ 22/11/2024 9:40 pm WinZip Fil mongodb_backup_2024-11-22_21-45-06.zip ✓ 22/11/2024 9:45 pm WinZip Fil mongodb_backup_2024-11-22_21-50-05.zip ✓ 22/11/2024 9:50 pm WinZip Fil
3.	Copy both Cloudify and Cloudify-Auth folders from the unzip backup folder
	 Name Status Date modified Type Size Cloudify ✓ 26/11/2024 8:41 pm File folder Cloudify-Auth ✓ 26/11/2024 8:41 pm File folder
4.	Go to the repository and replace both existing Cloudify and Cloudify-Auth folders in the monfdb_backup folder. Note: Do not delete the .migration_complete file

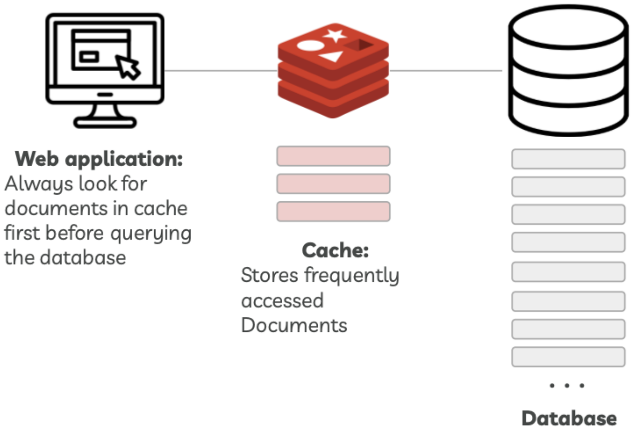
	
	`⌄ mongodb_backup`
	`  › Cloudify`
	`  › Cloudify-Auth`
	`  .migration_complete`

5.	Go to the .env file and make sure RECOVER_FLAG is true

	
	`# RECOVER_FLAG=false/true`
	`RECOVER_FLAG=true`

6.	Docker compose build and Docker compose up the application again

Appendix S. Caching Strategy


Web application:
Always look for documents in cache first before querying the database
Cache:
Stores frequently accessed Documents
Database

Purpose

- Enhance response time by caching frequently accessed admin data.
- Reduce database load by serving data from cache whenever possible.

Caching Details

- **Cache Key**: admin:all_admin
- **Cache Duration**: 30 minutes (1800 seconds)
- **Caching Mechanism**: Redis-based cache implemented via CacheService.

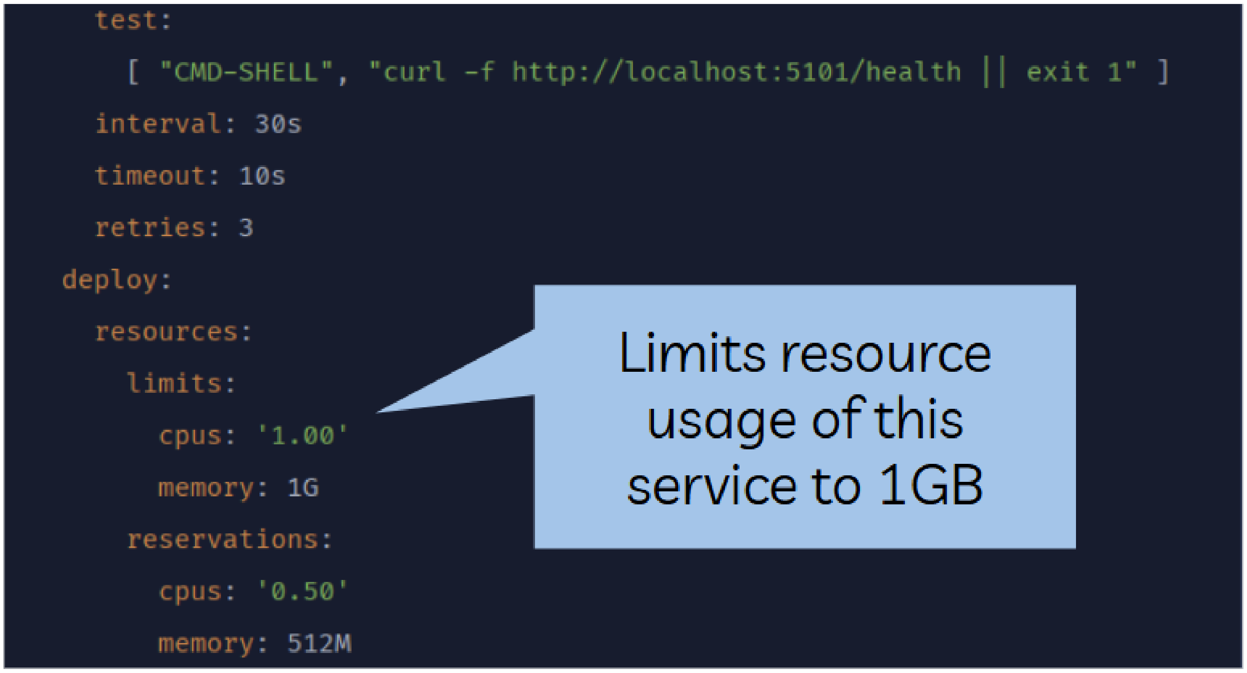
Implementation

1. **Fetch Data**:
    - Check cache for the required data (CacheService.get_json_cache).
    - On a cache miss, query the database, process the data, and store it in the cache.
2. **Automatic Data Formatting**:
    - Converts date fields between datetime and ISO format for consistency.
3. **Cache Management**:
    - **Invalidate Cache**:
        - Clear cache when admin-related actions are performed (e.g., create, update, deactivate).
    - **Refresh Cache**:
        - Explicitly fetch fresh data from the database and re-store in the cache.
    - **Check Cache Status**:
        - Retrieve time-to-live (TTL) and verify cache availability.

Scenarios

1. **Fetching Admin Data**:
    - If data exists in the cache, serve it immediately.
    - If the cache is empty or expired, fetch fresh data from the database and update the cache
2. **Admin Modifications**:
    - After creating, updating, or deactivating an admin account, invalidate the cache to prevent serving outdated data.

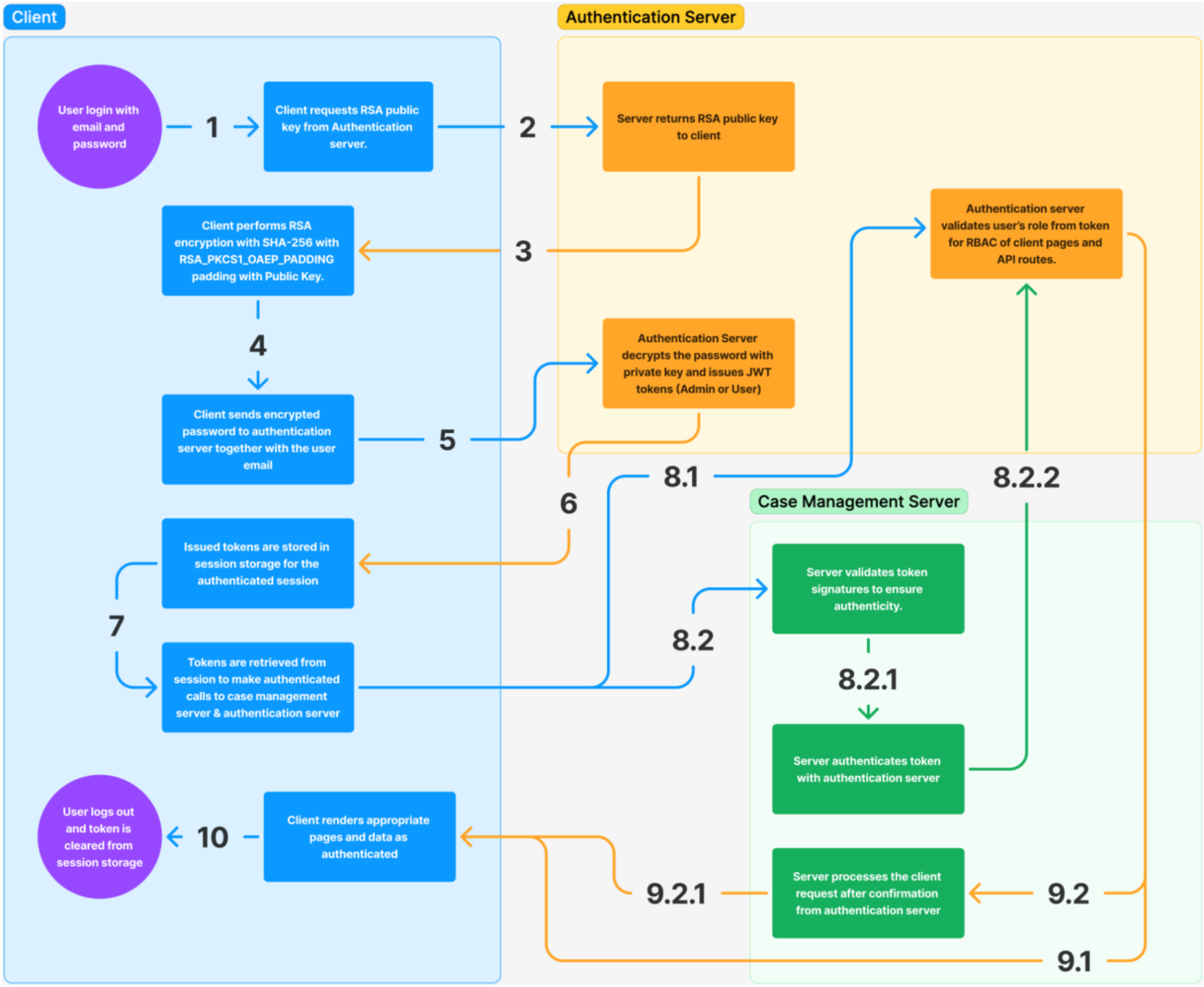
Appendix T. Sample Resource configuration



```
    test:
      [ "CMD-SHELL", "curl -f http://localhost:5101/health || exit 1" ]
    interval: 30s
    timeout: 10s
    retries: 3
  deploy:
    resources:
      limits:
        cpus: '1.00'
        memory: 1G
      reservations:
        cpus: '0.50'
        memory: 512M
```

Limits resource usage of this service to 1GB

Appendix U. Authentication Flow


Client
Authentication Server
Case Management Server
User login with email and password
1
Client requests RSA public key from Authentication server.
2
Server returns RSA public key to client
3
Client performs RSA encryption with SHA-256 with RSA_PKCS1_OAEP_PADDING padding with Public Key.
4
Client sends encrypted password to authentication server together with the user email
5
Authentication Server decrypts the password with private key and issues JWT tokens (Admin or User)
6
Issued tokens are stored in session storage for the authenticated session
7
Tokens are retrieved from session to make authenticated calls to case management server & authentication server
8.1
Authentication server validates user's role from token for RBAC of client pages and API routes.
8.2
Server validates token signatures to ensure authenticity.
8.2.1
Server authenticates token with authentication server
8.2.2
9.1
9.2
Server processes the client request after confirmation from authentication server
9.2.1
Client renders appropriate pages and data as authenticated
10
User logs out and token is cleared from session storage

Appendix V. Example of a Sprint

1 Week Sprint (10 Days)

Day 0	D+1	D+2	D+3	D+4	D+5	D+6	D+7	D+8 Intermission Day	D+9
Sprint Planning (F2F)		Stand-Up		Stand-Up		Stand-Up		Sprint Review + Retrospective	Rest Day

2 Week Sprint (17 Days)

Day 0	D+1	D+2	D+3	D+4	D+5	D+6	D+7	D+8	D+9
Sprint Planning (F2F)		Stand-Up		Stand-Up		Stand-Up		Stand-Up	
D+10	**D+11**	**D+12**	**D+13**	**D+14**	**D+15 Intermission Day**		**D+16**		
Stand-Up		Stand-Up		Stand-Up	Sprint Review + Retrospective		Rest Day		

Appendix W. Sprint Activities

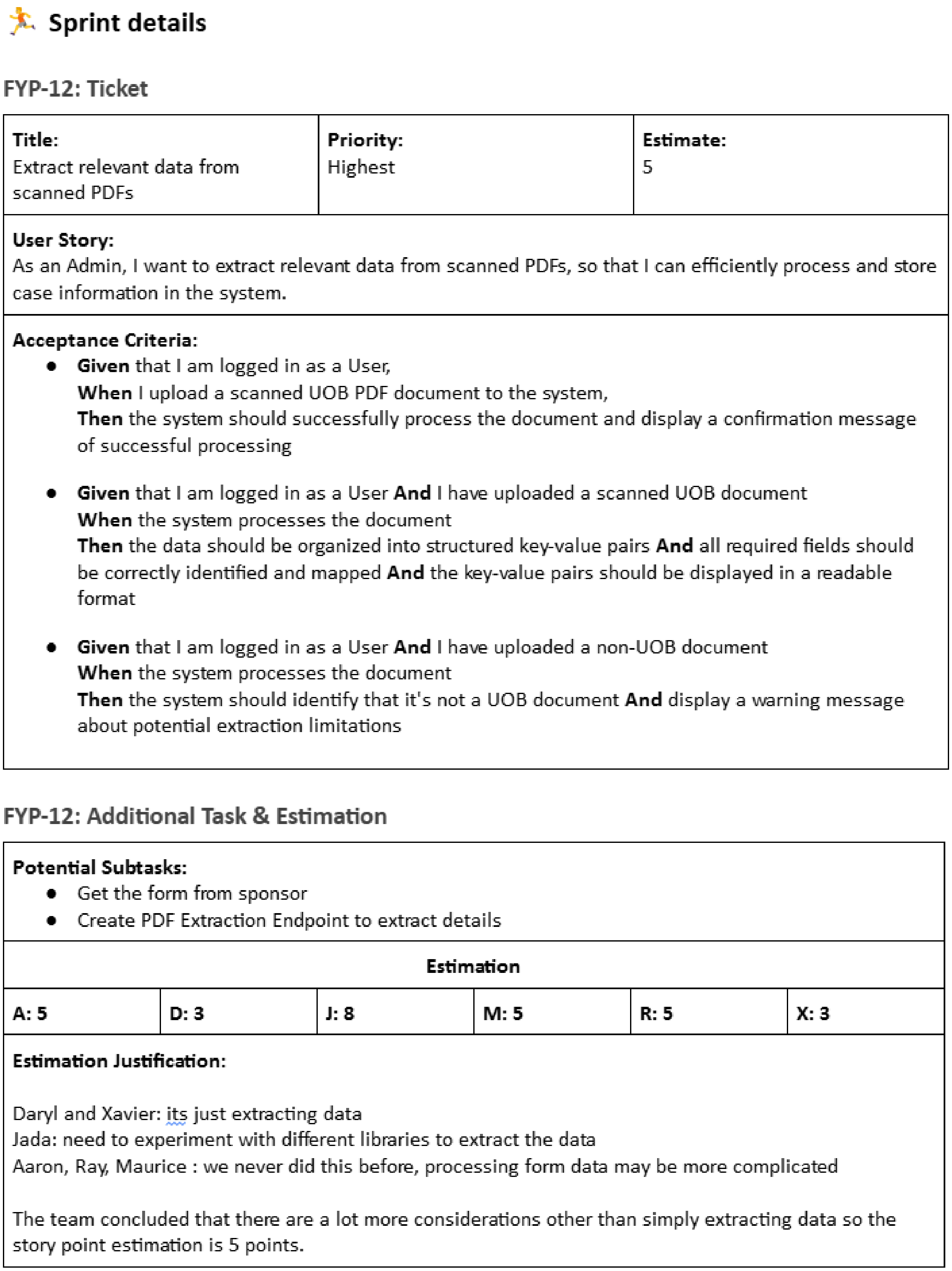
Activity	Description
Sprint Grooming & Planning	Face-to-face sessions focusing on backlog refinement and sprint scope definition. Key activities include story point estimation, task breakdown, sprint goal setting, and capacity planning. The Product Owner presents prioritized features while the team assesses feasibility and resource requirements
Standups	Every other day's 15-minute Zoom meetings where team members discuss three key points: previous days' accomplishments, current day's plans, and any blockers encountered.
Sprint Review	End of Sprint Face-to-Face demonstrations to stakeholders showcasing completed features and gathering feedback. The session includes live system demonstrations, discussion of completed user stories, and collection of stakeholder input for future improvements.
Sprint Retrospective	End-of-sprint Zoom/face-to-face meetings focused on process improvement. The team discusses what went well, what could be improved, and agrees on specific action items for the next sprint. Insights are documented in Confluence for future reference.
End-of-sprint Supervisor Check In	Dedicated session with project supervisor to review sprint outcomes, discuss technical challenges, receive guidance on implementation approaches, and align on academic deliverables. Technical decisions and recommendations are documented in Confluence and Check-in Slides.

Appendix X. Definition of Done

For any of our Tickets or Stories to be marked as done, they must fulfil the following criteria:

- All Acceptance Criteria Met
- Pass Code Linting
- No Build Failures
- Test Cases added into the Test Case Document
- Pass all implemented Unit Test

Appendix Y. Sample Sprint Planning Artifact



🏃 Sprint details

FYP-12: Ticket

Title: Extract relevant data from scanned PDFs	**Priority:** Highest	**Estimate:** 5

User Story:
As an Admin, I want to extract relevant data from scanned PDFs, so that I can efficiently process and store case information in the system.

Acceptance Criteria:

- **Given** that I am logged in as a User,
  When I upload a scanned UOB PDF document to the system,
  Then the system should successfully process the document and display a confirmation message of successful processing

- **Given** that I am logged in as a User **And** I have uploaded a scanned UOB document
  When the system processes the document
  Then the data should be organized into structured key-value pairs **And** all required fields should be correctly identified and mapped **And** the key-value pairs should be displayed in a readable format

- **Given** that I am logged in as a User **And** I have uploaded a non-UOB document
  When the system processes the document
  Then the system should identify that it's not a UOB document **And** display a warning message about potential extraction limitations

FYP-12: Additional Task & Estimation

Potential Subtasks:

- Get the form from sponsor
- Create PDF Extraction Endpoint to extract details

Estimation

A: 5	**D: 3**	**J: 8**	**M: 5**	**R: 5**	**X: 3**

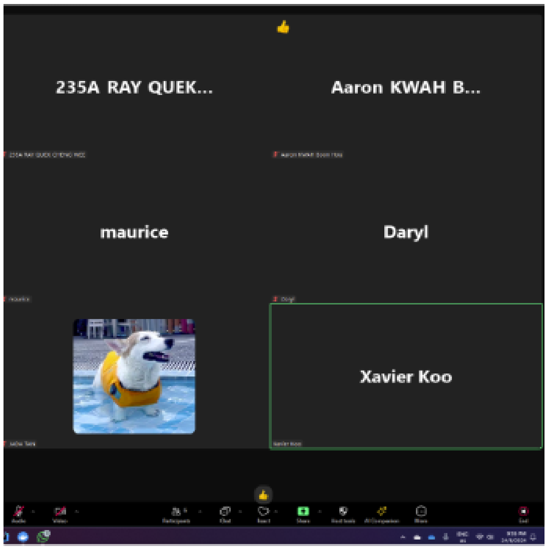
Estimation Justification:

Daryl and Xavier: its just extracting data
Jada: need to experiment with different libraries to extract the data
Aaron, Ray, Maurice : we never did this before, processing form data may be more complicated

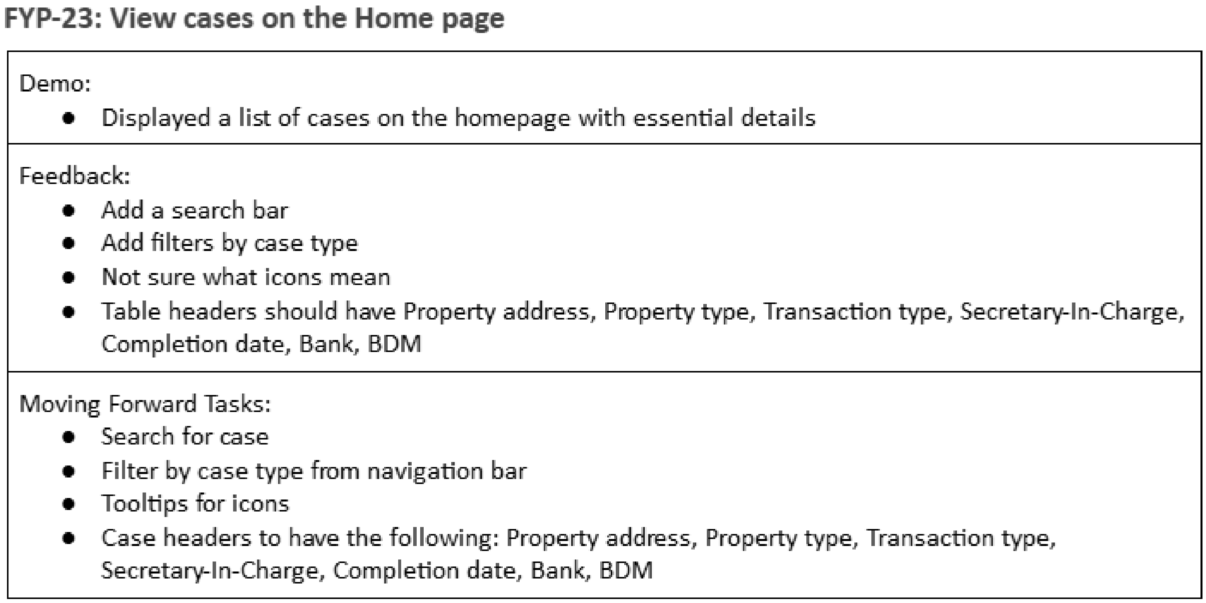
The team concluded that there are a lot more considerations other than simply extracting data so the story point estimation is 5 points.

Appendix Z. Sample Meeting Minutes

📅 24/08/2024

Name	Priorities 🧐	Progress 😊	Problems 😐
Aaron	● Review Daryl's Pull Request ● Look at Deployment Issue	● Created 3 API Endpoints for Retrieval by Role, Case Deletion, Case Creation ● Reviewed Ray and Maurice's PR	-
Daryl	● Return button for Individual Case	● Created PR for Displaying Individual Case Details	● Issue with resolving Next.js export and deployment
Jada	● Mapping key-value pairs for extraction	● Accessed PDFs using screen reader	-
Maurice	● Look into deployment issue	● Created UI Dashboard and integrating cases into table format	-
Ray	● Work on Case Deletion UI	● Created endpoint for fetching specific case detail	-
Xavier	● Heroku Deployment ● Finish Case Creation	● Created first version of app deployment ● Created layout for new case creation for data ingestion workflow	-
			

Appendix AA. Sample Sprint Review



FYP-23: View cases on the Home page

Demo:

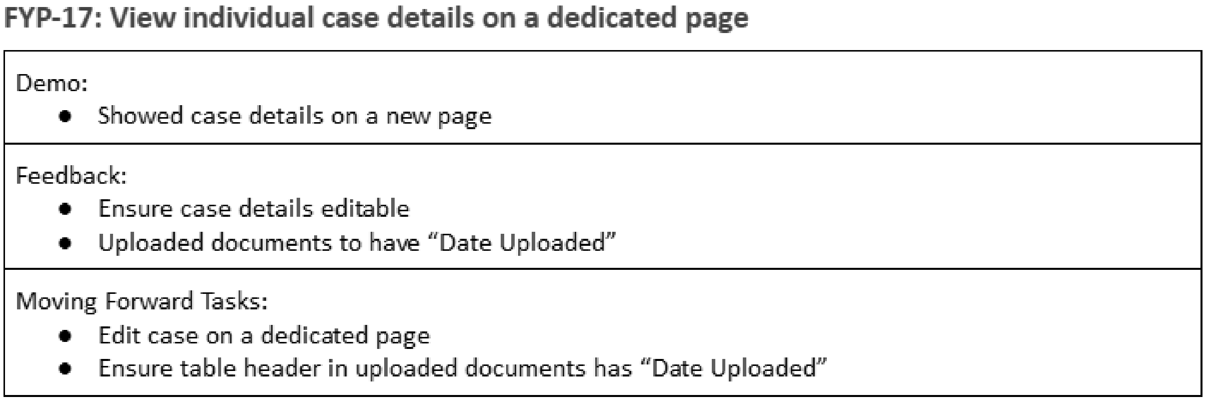
- Displayed a list of cases on the homepage with essential details

Feedback:

- Add a search bar
- Add filters by case type
- Not sure what icons mean
- Table headers should have Property address, Property type, Transaction type, Secretary-In-Charge, Completion date, Bank, BDM

Moving Forward Tasks:

- Search for case
- Filter by case type from navigation bar
- Tooltips for icons
- Case headers to have the following: Property address, Property type, Transaction type, Secretary-In-Charge, Completion date, Bank, BDM



FYP-17: View individual case details on a dedicated page

Demo:

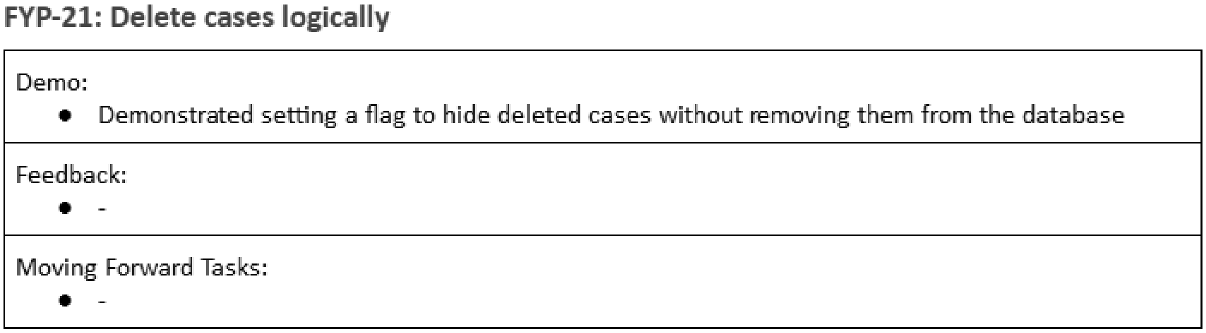
- Showed case details on a new page

Feedback:

- Ensure case details editable
- Uploaded documents to have "Date Uploaded"

Moving Forward Tasks:

- Edit case on a dedicated page
- Ensure table header in uploaded documents has "Date Uploaded"



FYP-21: Delete cases logically

Demo:

- Demonstrated setting a flag to hide deleted cases without removing them from the database

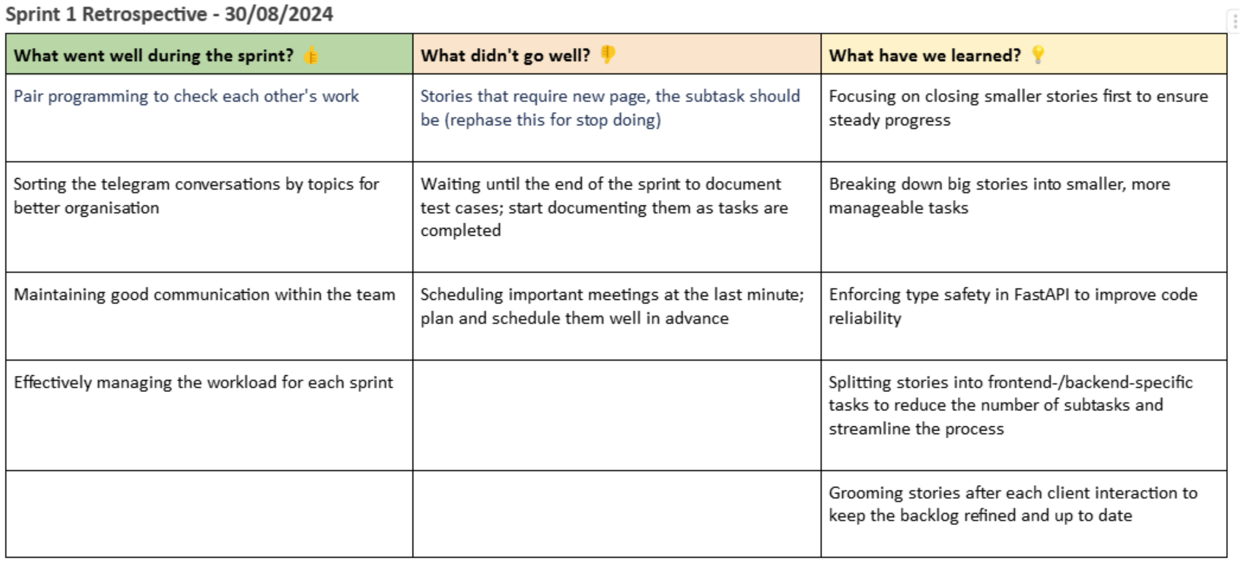
Feedback:

- -

Moving Forward Tasks:

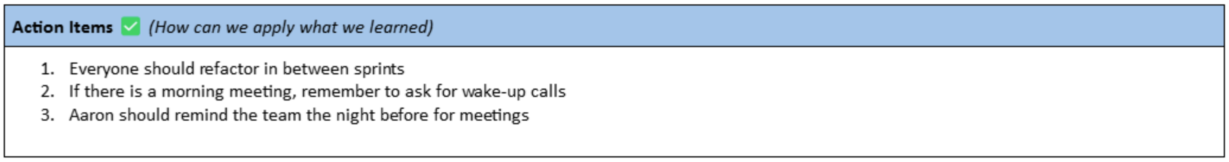
- -

Appendix BB. Sample Retrospective



Sprint 1 Retrospective - 30/08/2024

What went well during the sprint? 👍	What didn't go well? 👎	What have we learned? 💡
Pair programming to check each other's work	Stories that require new page, the subtask should be (rephase this for stop doing)	Focusing on closing smaller stories first to ensure steady progress
Sorting the telegram conversations by topics for better organisation	Waiting until the end of the sprint to document test cases; start documenting them as tasks are completed	Breaking down big stories into smaller, more manageable tasks
Maintaining good communication within the team	Scheduling important meetings at the last minute; plan and schedule them well in advance	Enforcing type safety in FastAPI to improve code reliability
Effectively managing the workload for each sprint		Splitting stories into frontend-/backend-specific tasks to reduce the number of subtasks and streamline the process
		Grooming stories after each client interaction to keep the backlog refined and up to date



Action Items ✅ *(How can we apply what we learned)*

1. Everyone should refactor in between sprints
2. If there is a morning meeting, remember to ask for wake-up calls
3. Aaron should remind the team the night before for meetings

Appendix CC. Communication Guidelines

Communication between Team & External Parties

1. Face-to-Face (F2F): Used for Sprint Review
   - **Sprint Review**: The team will go down to the Project Sponsor's office for the sprint review where the team will present the completed work to stakeholders/sponsors, gather feedback and assess whether the sprint goals were achieved.

2. The **Product Owner** will take the lead in communicating with the sponsor.

Communication between Team & Supervisor

1. Face-to-Face (F2F): Used for Regular check-ins
   - **Regular check-ins**, where the team will provide status updates on a regular basis (usually at the end of each sprint) to the Project Supervisor. These status updates may include Sprint Achievements, Key Risk, Timeline, Project Scope Updates, Client Feedback.

2. The **Team Lead** will take the lead in communicating with the Supervisor.

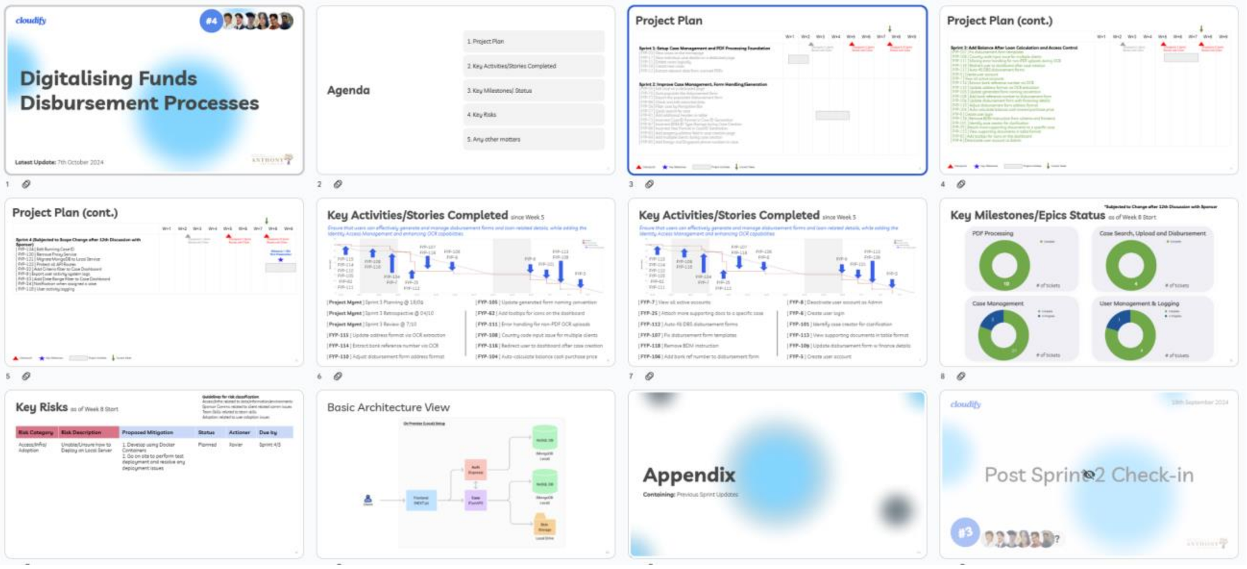
Communication within Team

1. Telegram: Used for Issue Communication
   - **Telegram** is used to communicate and address any issues that arise during the sprint. It provides a quick and asynchronous way to discuss problems, seek solutions, or request assistance.

2. Zoom: Used for Standup and Retrospective
   - **Standup**: Zoom is used for daily standup meetings where team members provide status updates, discuss work plans for the day, and identify any obstacles.
   - **Retrospective**: Zoom/F2F is also used for the sprint retrospective, a meeting where the team reflects on the past sprint, identifies what went well and what could be improved, and discusses action items for the next sprint.

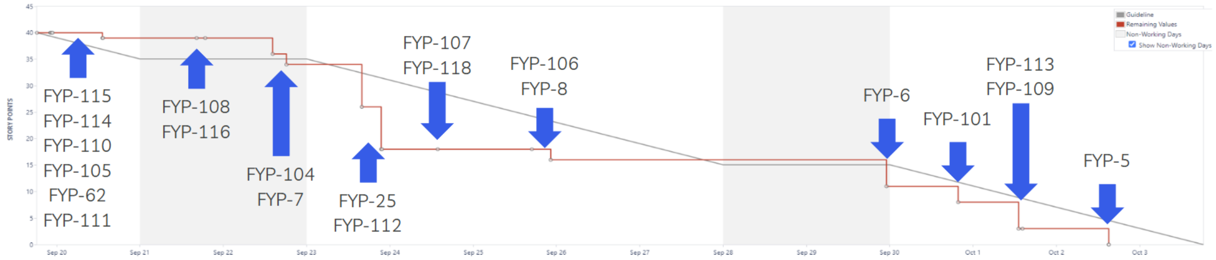
3. F2F: Used for Sprint Planning
   - **Sprint planning meetings**, where the team defines the scope of work for the upcoming sprint, are conducted face-to-face. This allows for in-depth discussions and collaborative decision-making.

Appendix DD. Post Sprint Check-in with Project Supervisor


Digitalising Funds Disbursement Processes
Latest Update: 7th October 2024
Agenda
1. Project Plan
2. Key Activities/Stories Completed
3. Key Milestones/ Status
4. Key Risks
5. Any other matters
Project Plan
Project Plan (cont.)
Project Plan (cont.)
Key Activities/Stories Completed since Week 5
Key Activities/Stories Completed since Week 5
Key Milestones/Epics Status as of Week 8 Start
Key Risks as of Week 8 Start
Basic Architecture View
Appendix
Post Sprint 2 Check-in
1
2
3
4
5
6
7
8

Key Activities/Stories Completed since Week 5

Ensure that users can effectively generate and manage disbursement forms and loan-related details, while adding the Identity Access Management and enhancing OCR capabilities.



[**Project Mgmt**] Sprint 3 Planning @ 18/09

[**Project Mgmt**] Sprint 3 Retrospective @ 04/10

[**Project Mgmt**] Sprint 3 Review @ 7/10

[**FYP-115**] Update address format via OCR extraction

[**FYP-114**] Extract bank reference number via OCR

[**FYP-110**] Adjust disbursement form address format

[**FYP-105**] Update generated form naming convention

[**FYP-62**] Add tooltips for icons on the dashboard

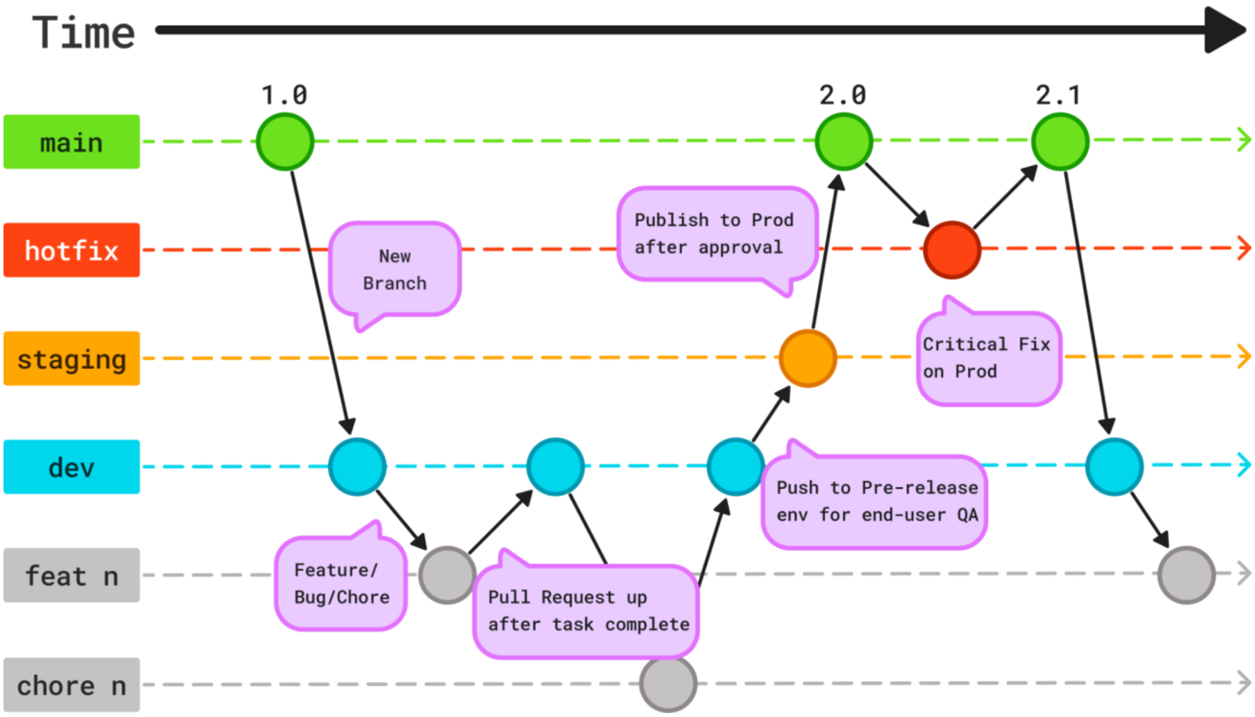
[**FYP-111**] Error handling for non-PDF OCR uploads

[**FYP-108**] Country code input issue for multiple clients

[**FYP-116**] Redirect user to dashboard after case creation

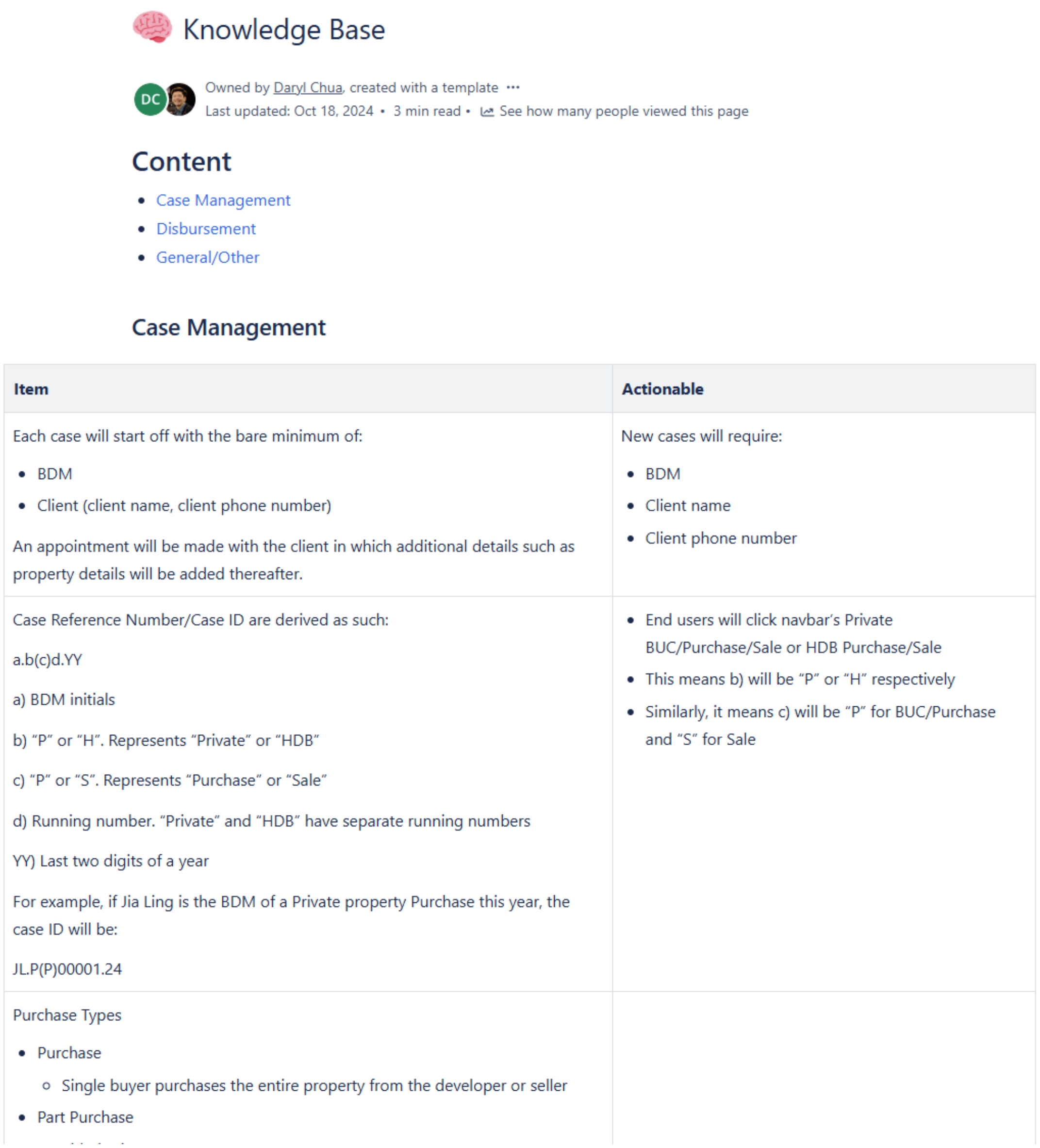
[**FYP-104**] Auto-calculate balance cash purchase price

Appendix EE. Git Workflow & Commit Convention


Time
1.0
2.0
2.1
main
hotfix
staging
dev
feat n
chore n
New Branch
Publish to Prod after approval
Critical Fix on Prod
Push to Pre-release env for end-user QA
Feature/ Bug/Chore
Pull Request up after task complete

Prefix	Description	Example
feat:	For introducing a new feature.	*feat: add user authentication module*
fix:	When making a bug fix.	*fix: resolve crash on login when user credentials are invalid*
style:	For features and updates related to styling.	*style: update button colours to match the new branding guidelines*
refactor:	When refactoring a specific part of the codebase.	*refactor: optimise database queries for better performance*
test:	Changes tied to testing.	*test: add unit tests for the payment processing module*
docs:	All things documentation.	*docs: update API documentation for the new authentication endpoints*
chore:	For regular code maintenance tasks.	*chore: update dependencies to the latest versions*

Appendix FF. Knowledge Sharing



🧠 Knowledge Base

DC Owned by Daryl Chua, created with a template ···
Last updated: Oct 18, 2024 • 3 min read • See how many people viewed this page

Content

Case Management

Item	Actionable
Each case will start off with the bare minimum of: • BDM • Client (client name, client phone number) An appointment will be made with the client in which additional details such as property details will be added thereafter.	New cases will require: • BDM • Client name • Client phone number
Case Reference Number/Case ID are derived as such: a.b(c)d.YY a) BDM initials b) "P" or "H". Represents "Private" or "HDB" c) "P" or "S". Represents "Purchase" or "Sale" d) Running number. "Private" and "HDB" have separate running numbers YY) Last two digits of a year For example, if Jia Ling is the BDM of a Private property Purchase this year, the case ID will be: JL.P(P)00001.24	• End users will click navbar's Private BUC/Purchase/Sale or HDB Purchase/Sale • This means b) will be "P" or "H" respectively • Similarly, it means c) will be "P" for BUC/Purchase and "S" for Sale
Purchase Types • Purchase ○ Single buyer purchases the entire property from the developer or seller • Part Purchase	

Appendix GG. Project Risk Analysis

Risk Assessment Matrix (RAM)

- **Red**: High Risk
- **Amber**: Moderate Risk
- **Green**: Low Risk

Likelihood \ Impact	High (3)	Medium (2)	Low (1)
High (3)	9	6	3
Medium (2)	6	4	2
Low (1)	3	2	1

Mitigation Outcome Labels

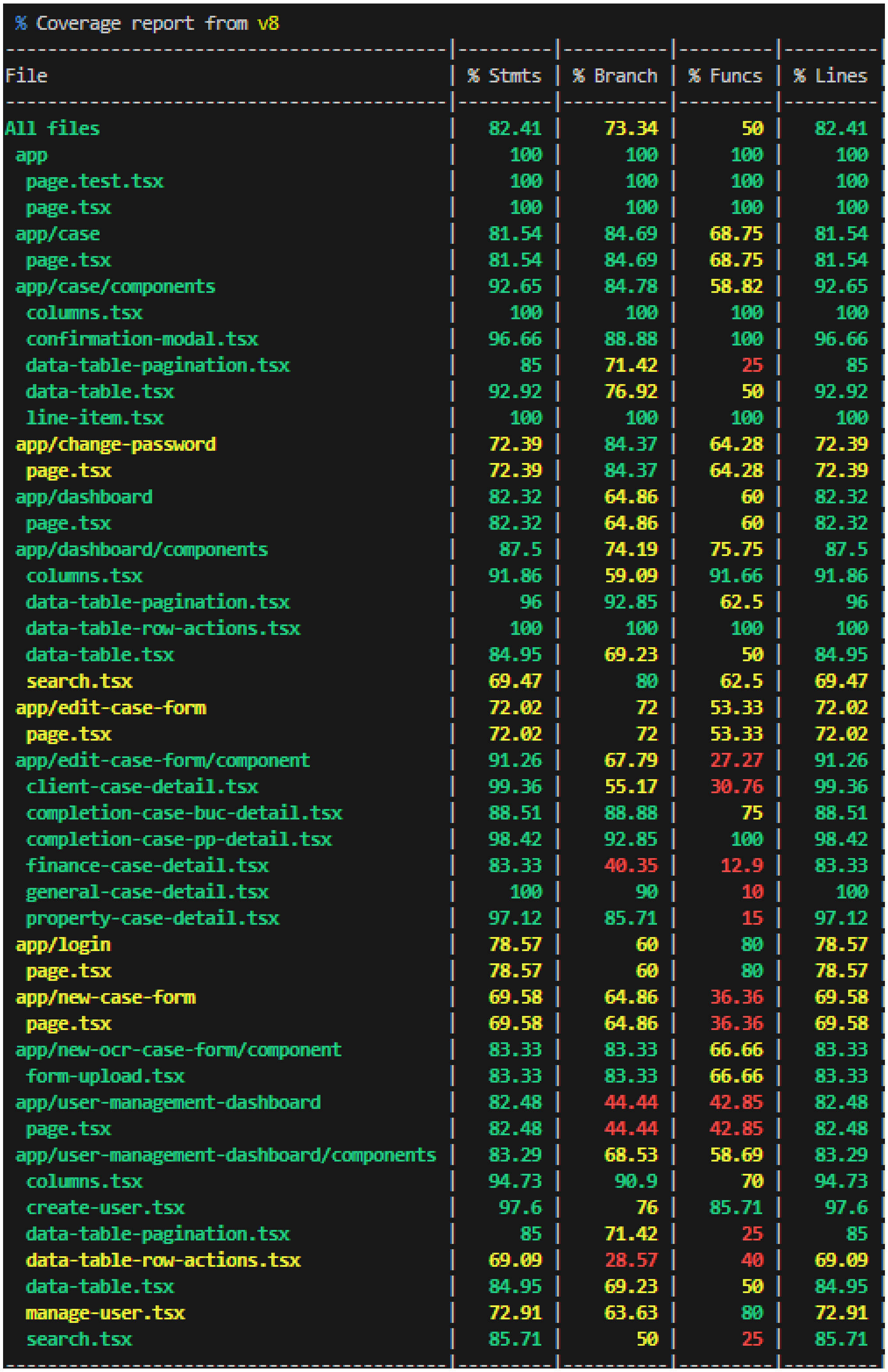
Highly Effective	Strategies that are consistently successful in mitigating risks effectively.
Partially Effective	Strategies with limited success, addressing some but not all risk factors.
Ineffective	Strategies that fail to mitigate risks or yield negative outcomes.
Non-Risk	Risks that are no longer relevant due to fundamental project changes.

Risk Mitigation

Risk Description	Consequence	RAM Level	Proposed Mitigation	Mitigation Outcome
Not able to deploy to Heroku/Cloudflare pages due to technology incompatibility.	Unable to use intended infrastructure, leading to delays in deployment and increased complexity.	Likelihood: H (3) Impact: H (3) Total: 9	AWS ECS + Fargate as alternative for deployment of FE and BE.	**Non-Risk** The risk was fully mitigated as the project shifted deployment strategy to on-premise, removing dependency on incompatible technologies.

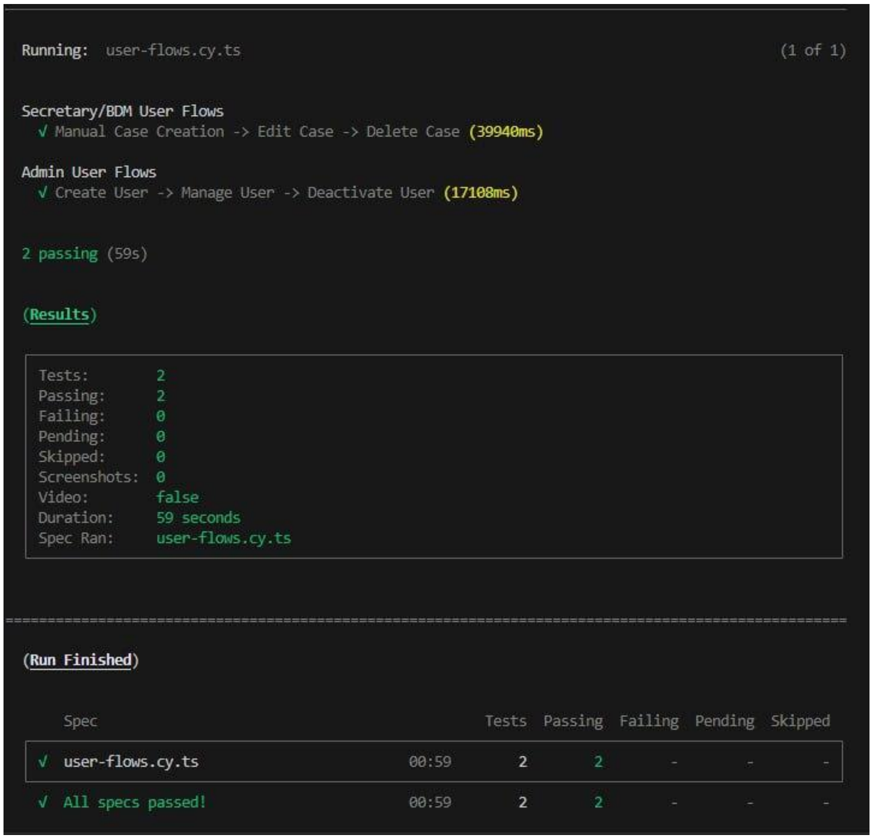
Alignment with sponsors' currently existing cases	Data inconsistency or misalignment with sponsor workflows, leading to incorrect or unusable case data.	Likelihood: M (2) Impact: M (2) Total: 4	Allow users to edit case ID's running number **[FYP-124]**	**Highly Effective** Allowing users to edit case IDs resolved data alignment issues, ensuring workflows meet the sponsor's requirements.
Intricacies of business may result in new undiscussed knowledge	Misunderstandings may cause process or requirement mismatches, affecting project functionality.	Likelihood: M (2) Impact: M (2) Total: 4	Clarify with sponsor as needed	**Partially Effective** Regular clarifications helped address some issues, but unforeseen changes caused occasional mismatches, limiting full effectiveness.
Unable to extract data and map fields accurately due to sponsor's OCR	Errors in data extraction and mapping, resulting in invalid or incomplete records.	Likelihood: M (2) Impact: M (2) Total: 4	Allow users to view and edit fields before saving. **[FYP-68]**	**Highly Effective** Allowing manual review and editing resolved inaccuracies, resulting in a robust solution for clean and reliable data.
Unable/Unsure how to Deploy on Local Server	Deployment delays and untested environment configurations, potentially causing issues	Likelihood: H (3) Impact: H (3) Total: 9	1. Develop using Docker Containers 2. Go on site to perform test deployment and resolve any deployment issues	**Highly Effective** The Docker-based deployment and on-site testing successfully resolved all deployment issues, enabling smooth operations.

Appendix HH. Unit Test Report

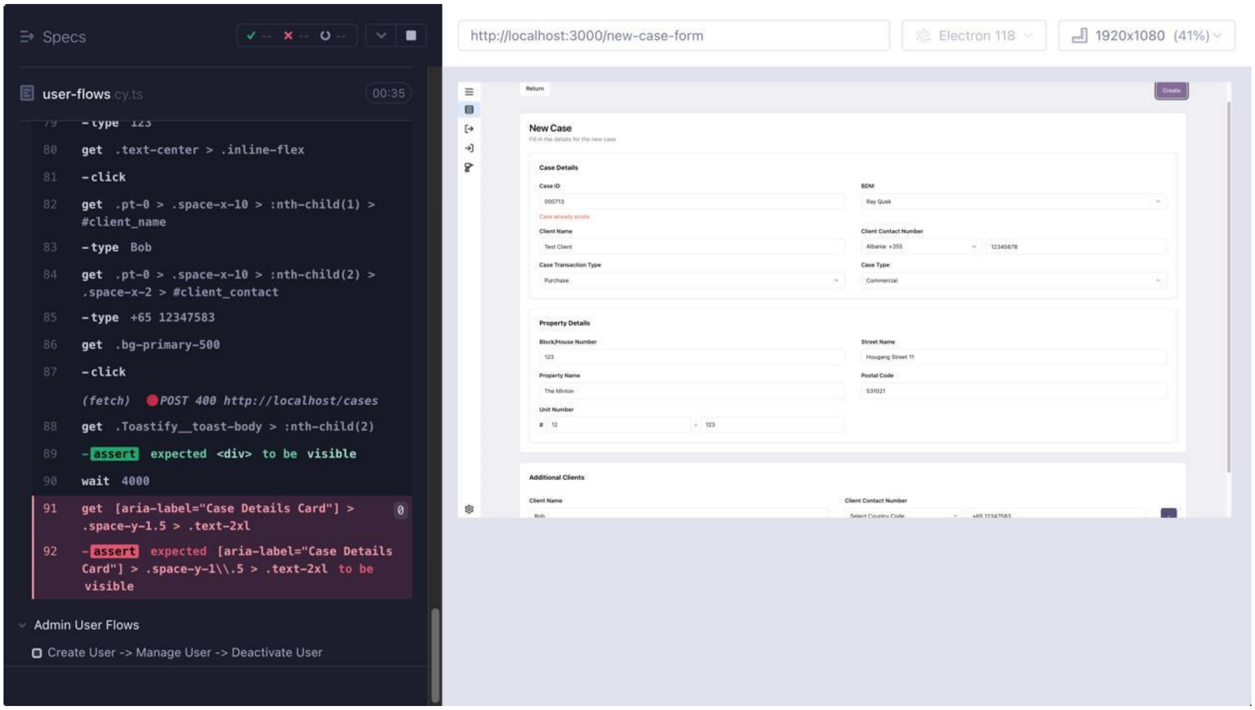


```
 % Coverage report from v8
-------------------------------------------|---------|----------|---------|---------|
File                                       | % Stmts | % Branch | % Funcs | % Lines |
-------------------------------------------|---------|----------|---------|---------|
All files                                  |   82.41 |    73.34 |      50 |   82.41 |
 app                                       |     100 |      100 |     100 |     100 |
  page.test.tsx                            |     100 |      100 |     100 |     100 |
  page.tsx                                 |     100 |      100 |     100 |     100 |
 app/case                                  |   81.54 |    84.69 |   68.75 |   81.54 |
  page.tsx                                 |   81.54 |    84.69 |   68.75 |   81.54 |
 app/case/components                       |   92.65 |    84.78 |   58.82 |   92.65 |
  columns.tsx                              |     100 |      100 |     100 |     100 |
  confirmation-modal.tsx                   |   96.66 |    88.88 |     100 |   96.66 |
  data-table-pagination.tsx                |      85 |    71.42 |      25 |      85 |
  data-table.tsx                           |   92.92 |    76.92 |      50 |   92.92 |
  line-item.tsx                            |     100 |      100 |     100 |     100 |
 app/change-password                       |   72.39 |    84.37 |   64.28 |   72.39 |
  page.tsx                                 |   72.39 |    84.37 |   64.28 |   72.39 |
 app/dashboard                             |   82.32 |    64.86 |      60 |   82.32 |
  page.tsx                                 |   82.32 |    64.86 |      60 |   82.32 |
 app/dashboard/components                  |    87.5 |    74.19 |   75.75 |    87.5 |
  columns.tsx                              |   91.86 |    59.09 |   91.66 |   91.86 |
  data-table-pagination.tsx                |      96 |    92.85 |    62.5 |      96 |
  data-table-row-actions.tsx               |     100 |      100 |     100 |     100 |
  data-table.tsx                           |   84.95 |    69.23 |      50 |   84.95 |
  search.tsx                               |   69.47 |       80 |    62.5 |   69.47 |
 app/edit-case-form                        |   72.02 |       72 |   53.33 |   72.02 |
  page.tsx                                 |   72.02 |       72 |   53.33 |   72.02 |
 app/edit-case-form/component              |   91.26 |    67.79 |   27.27 |   91.26 |
  client-case-detail.tsx                   |   99.36 |    55.17 |   30.76 |   99.36 |
  completion-case-buc-detail.tsx           |   88.51 |    88.88 |      75 |   88.51 |
  completion-case-pp-detail.tsx            |   98.42 |    92.85 |     100 |   98.42 |
  finance-case-detail.tsx                  |   83.33 |    40.35 |    12.9 |   83.33 |
  general-case-detail.tsx                  |     100 |       90 |      10 |     100 |
  property-case-detail.tsx                 |   97.12 |    85.71 |      15 |   97.12 |
 app/login                                 |   78.57 |       60 |      80 |   78.57 |
  page.tsx                                 |   78.57 |       60 |      80 |   78.57 |
 app/new-case-form                         |   69.58 |    64.86 |   36.36 |   69.58 |
  page.tsx                                 |   69.58 |    64.86 |   36.36 |   69.58 |
 app/new-ocr-case-form/component           |   83.33 |    83.33 |   66.66 |   83.33 |
  form-upload.tsx                          |   83.33 |    83.33 |   66.66 |   83.33 |
 app/user-management-dashboard             |   82.48 |    44.44 |   42.85 |   82.48 |
  page.tsx                                 |   82.48 |    44.44 |   42.85 |   82.48 |
 app/user-management-dashboard/components  |   83.29 |    68.53 |   58.69 |   83.29 |
  columns.tsx                              |   94.73 |     90.9 |      70 |   94.73 |
  create-user.tsx                          |    97.6 |       76 |   85.71 |    97.6 |
  data-table-pagination.tsx                |      85 |    71.42 |      25 |      85 |
  data-table-row-actions.tsx               |   69.09 |    28.57 |      40 |   69.09 |
  data-table.tsx                           |   84.95 |    69.23 |      50 |   84.95 |
  manage-user.tsx                          |   72.91 |    63.63 |      80 |   72.91 |
  search.tsx                               |   85.71 |       50 |      25 |   85.71 |
-------------------------------------------|---------|----------|---------|---------|
```

Appendix II. Cypress Automated Test Report & User Interface



```
Running:  user-flows.cy.ts                                                  (1 of 1)


Secretary/BDM User Flows
  √ Manual Case Creation -> Edit Case -> Delete Case (39940ms)

Admin User Flows
  √ Create User -> Manage User -> Deactivate User (17108ms)


2 passing (59s)


(Results)

┌────────────────────────────────────────────────────────────────────────────────┐
│ Tests:        2                                                                │
│ Passing:      2                                                                │
│ Failing:      0                                                                │
│ Pending:      0                                                                │
│ Skipped:      0                                                                │
│ Screenshots:  0                                                                │
│ Video:        false                                                            │
│ Duration:     59 seconds                                                       │
│ Spec Ran:     user-flows.cy.ts                                                 │
└────────────────────────────────────────────────────────────────────────────────┘


==================================================================================


(Run Finished)


     Spec                                   Tests  Passing  Failing  Pending  Skipped
┌────────────────────────────────────────────────────────────────────────────────┐
│ √  user-flows.cy.ts                00:59        2        2        -        -        - │
└────────────────────────────────────────────────────────────────────────────────┘
  √  All specs passed!               00:59        2        2        -        -        -
```



```
79  -type  123
80  get  .text-center > .inline-flex
81  -click
82  get  .pt-0 > .space-x-10 > :nth-child(1) >
    #client_name
83  -type  Bob
84  get  .pt-0 > .space-x-10 > :nth-child(2) >
    .space-x-2 > #client_contact
85  -type  +65 12347583
86  get  .bg-primary-500
87  -click
    (fetch)  ● POST 400 http://localhost/cases
88  get  .Toastify__toast-body > :nth-child(2)
89  -assert  expected <div> to be visible
90  wait  4000
91  get  [aria-label="Case Details Card"] >                0
    .space-y-1.5 > .text-2xl
92  -assert  expected [aria-label="Case Details
    Card"] > .space-y-1\\.5 > .text-2xl to be
    visible
```

Admin User Flows

□ Create User -> Manage User -> Deactivate User

New Case

Fill in the details for the new case

Case Details

Case ID: 000713

Case already exists

BDM: Ray Quek

Client Name: Test Client

Client Contact Number: Albania: +355 12345678

Case Transaction Type: Purchase

Case Type: Commercial

Property Details

Block/House Number: 123

Street Name: Hougang Street 11

Property Name: The Minton

Postal Code: 531021

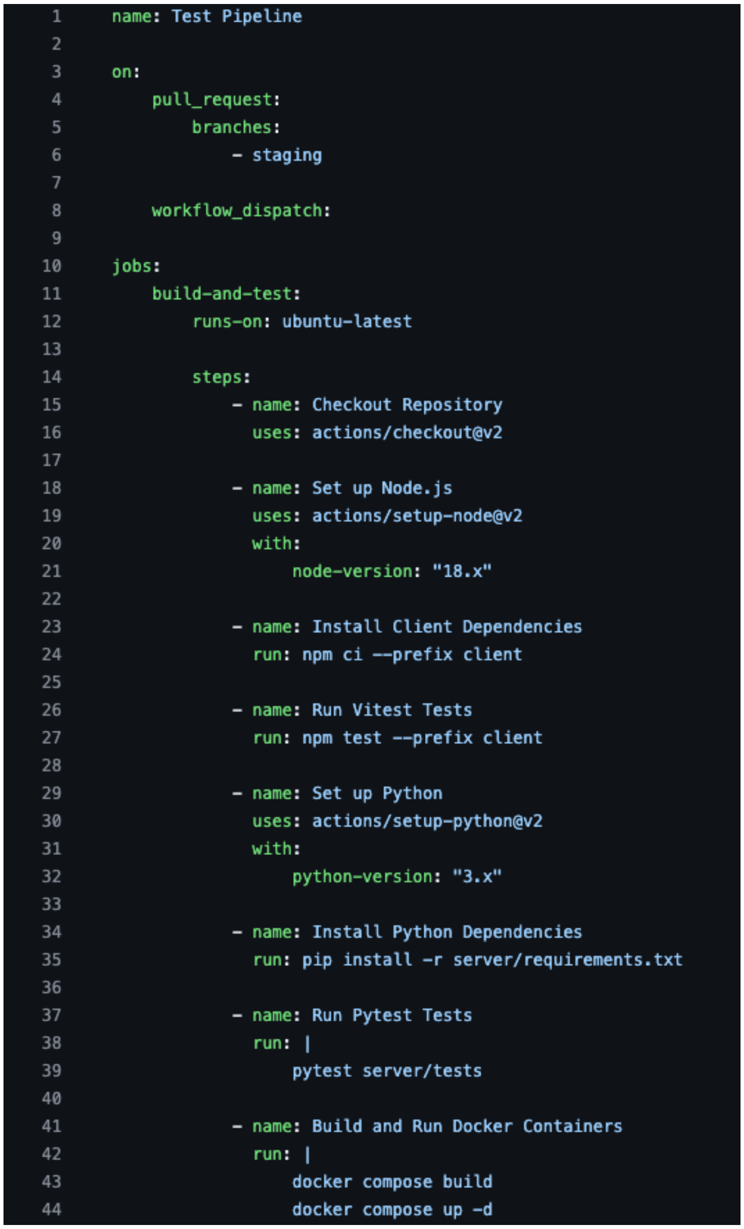
Unit Number: # 12 - 123

Additional Clients

Client Name: Bob

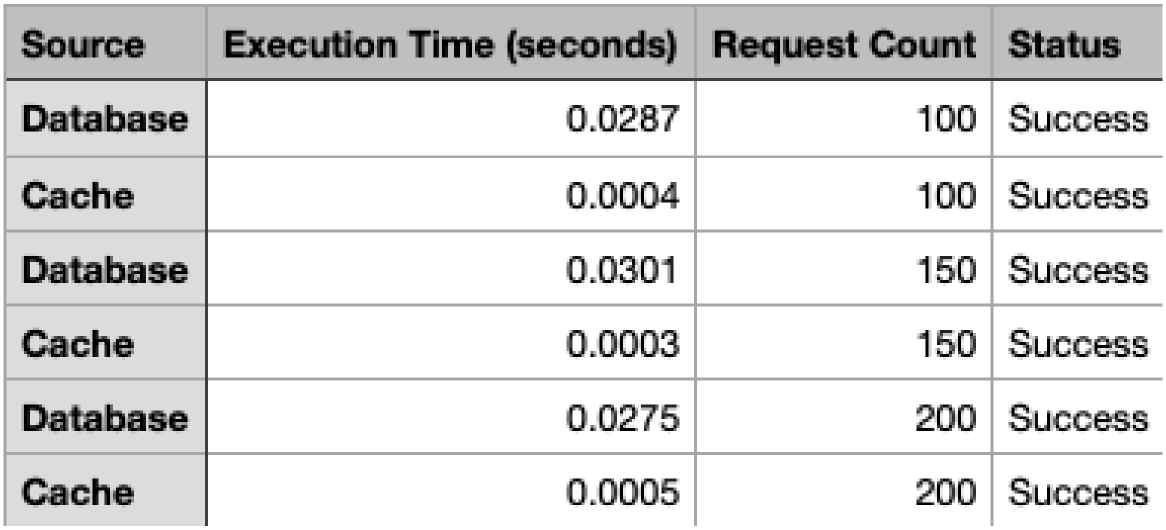
Client Contact Number: Select Country Code +65 12347583

Appendix JJ. GitHub Actions Workflow



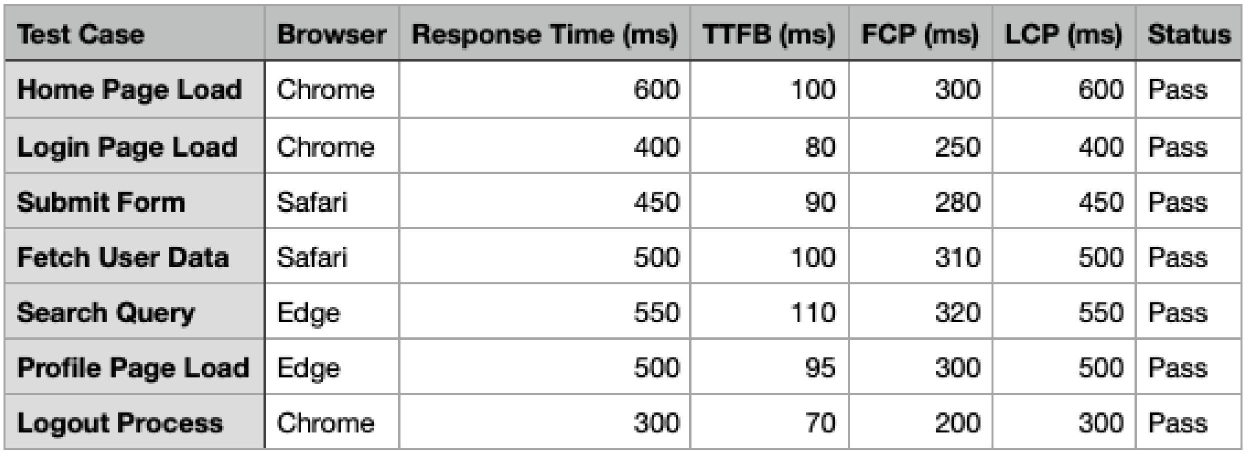
```yaml
name: Test Pipeline

on:
    pull_request:
        branches:
            - staging

    workflow_dispatch:

jobs:
    build-and-test:
        runs-on: ubuntu-latest

        steps:
            - name: Checkout Repository
              uses: actions/checkout@v2

            - name: Set up Node.js
              uses: actions/setup-node@v2
              with:
                  node-version: "18.x"

            - name: Install Client Dependencies
              run: npm ci --prefix client

            - name: Run Vitest Tests
              run: npm test --prefix client

            - name: Set up Python
              uses: actions/setup-python@v2
              with:
                  python-version: "3.x"

            - name: Install Python Dependencies
              run: pip install -r server/requirements.txt

            - name: Run Pytest Tests
              run: |
                  pytest server/tests

            - name: Build and Run Docker Containers
              run: |
                  docker compose build
                  docker compose up -d
```

Appendix KK. Caching Time Improvement



Source	Execution Time (seconds)	Request Count	Status
Database	0.0287	100	Success
Cache	0.0004	100	Success
Database	0.0301	150	Success
Cache	0.0003	150	Success
Database	0.0275	200	Success
Cache	0.0005	200	Success

Appendix LL. Browser Response Time Test



Test Case	Browser	Response Time (ms)	TTFB (ms)	FCP (ms)	LCP (ms)	Status
Home Page Load	Chrome	600	100	300	600	Pass
Login Page Load	Chrome	400	80	250	400	Pass
Submit Form	Safari	450	90	280	450	Pass
Fetch User Data	Safari	500	100	310	500	Pass
Search Query	Edge	550	110	320	550	Pass
Profile Page Load	Edge	500	95	300	500	Pass
Logout Process	Chrome	300	70	200	300	Pass

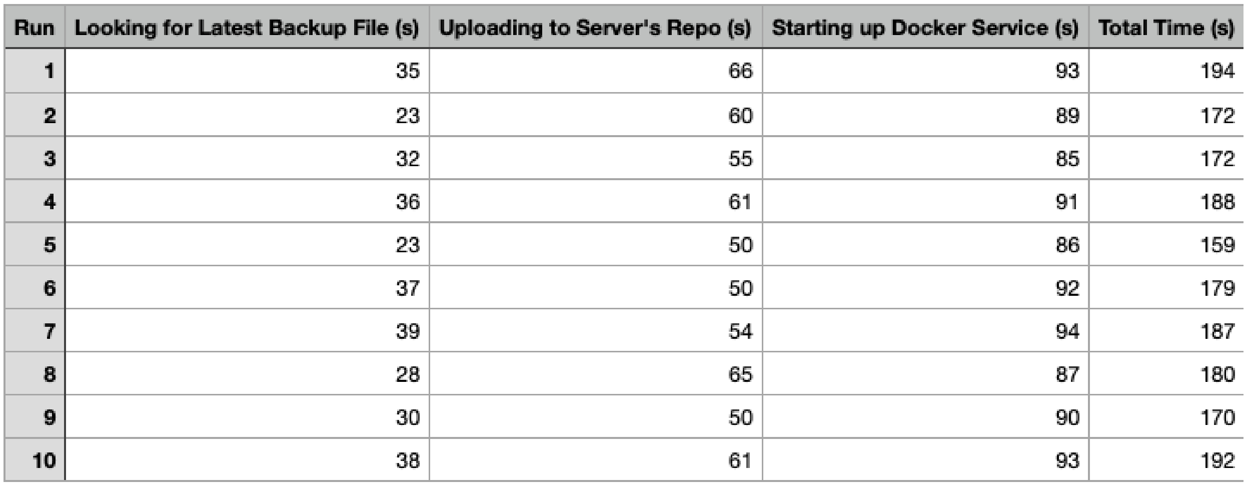
- **Response Time (ms)**: The total time taken (in milliseconds) for the browser to complete the specific test case.

- **TTFB (ms)**: Time to First Byte, the time (in milliseconds) it takes for the browser to receive the first byte of data from the server after making the request.

- **FCP (ms)**: First Contentful Paint, the time (in milliseconds) taken for the browser to render the first visible content on the page.

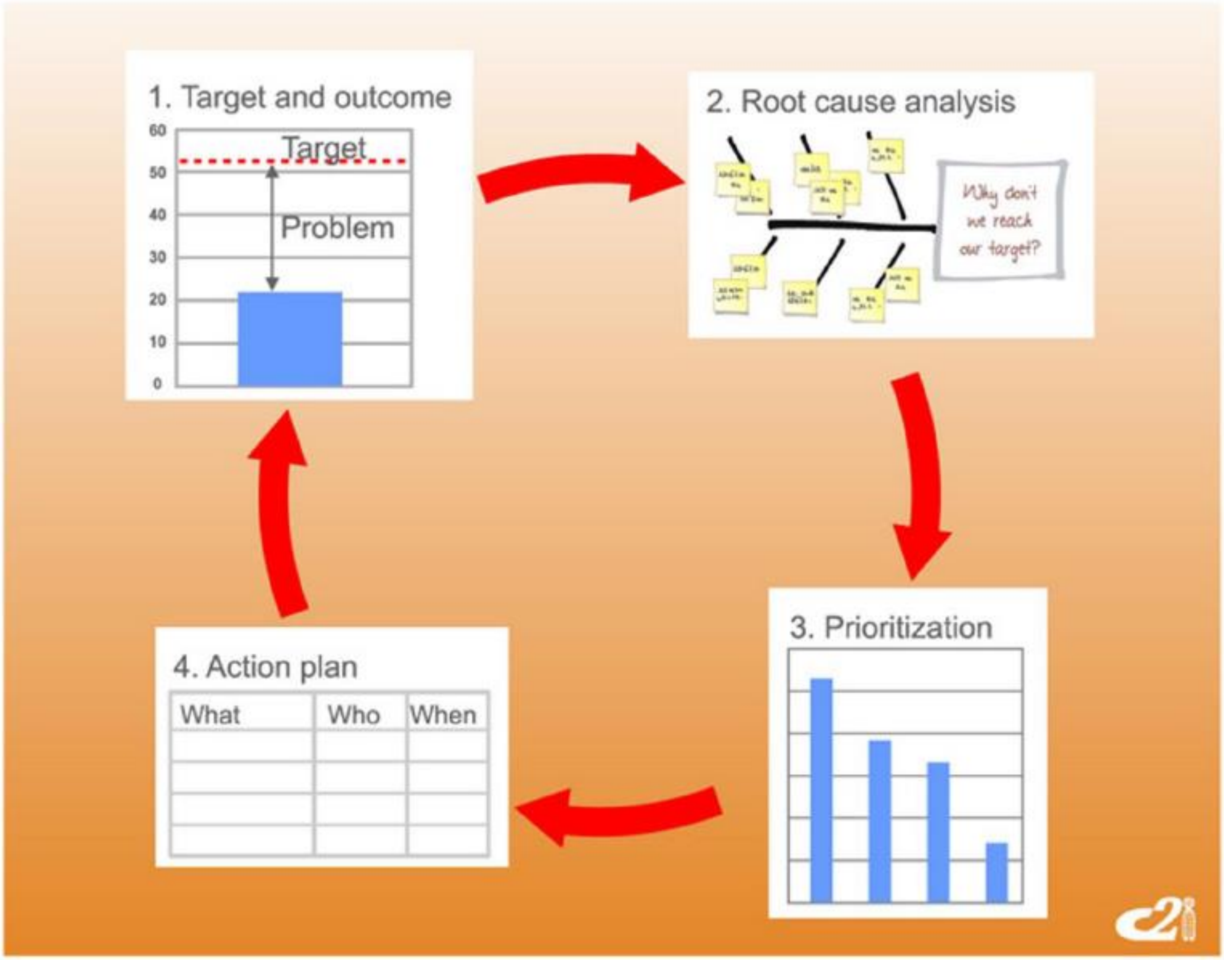
- **LCP (ms)**: Largest Contentful Paint, the time (in milliseconds) taken to render the largest visible content on the page.

Appendix MM. System Recovery Time Test



Run	Looking for Latest Backup File (s)	Uploading to Server's Repo (s)	Starting up Docker Service (s)	Total Time (s)
1	35	66	93	194
2	23	60	89	172
3	32	55	85	172
4	36	61	91	188
5	23	50	86	159
6	37	50	92	179
7	39	54	94	187
8	28	65	87	180
9	30	50	90	170
10	38	61	93	192

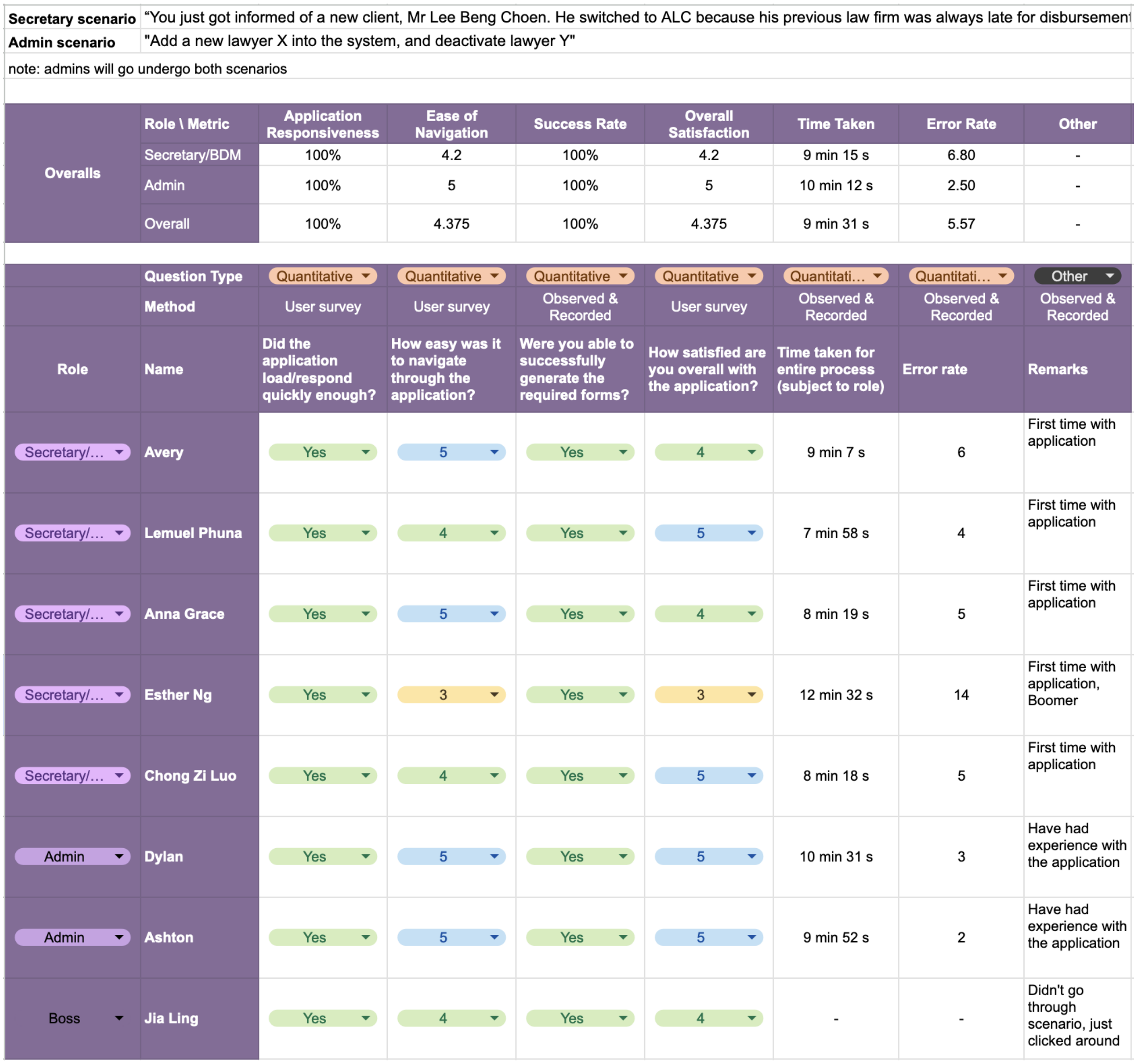
Appendix NN. Feedback to Actionable Process



What	Who	When

The "improvement loop" from the book *Getting the Right Things Done* by Pascal Dennis

Appendix OO. UAT Results



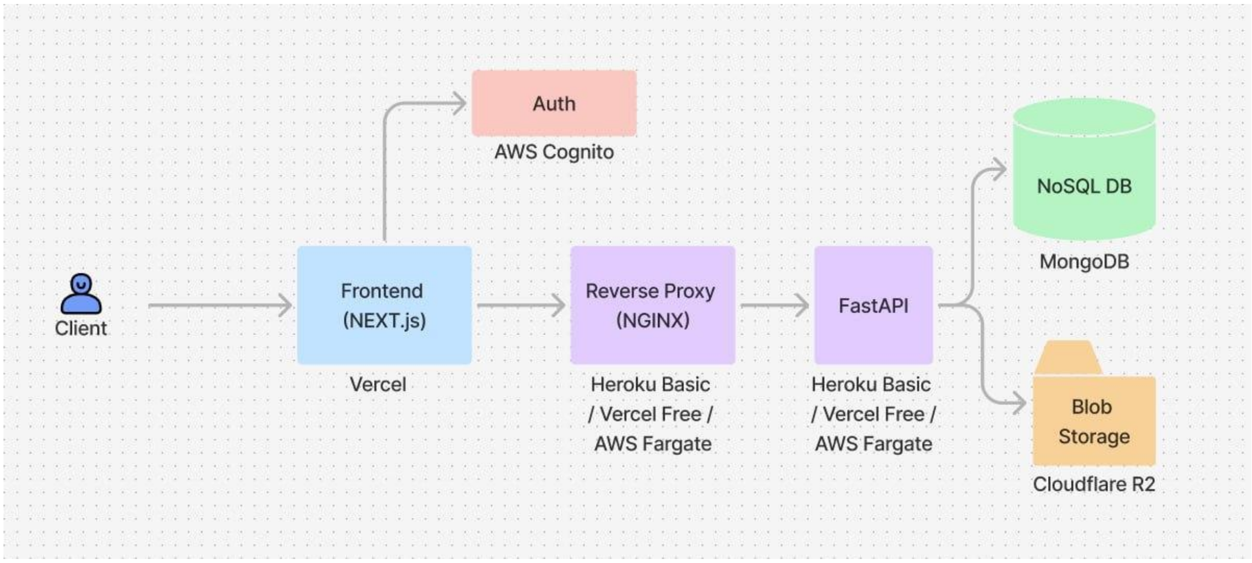
Secretary scenario	"You just got informed of a new client, Mr Lee Beng Choen. He switched to ALC because his previous law firm was always late for disbursement
Admin scenario	"Add a new lawyer X into the system, and deactivate lawyer Y"
note: admins will go undergo both scenarios	

	Role \ Metric	Application Responsiveness	Ease of Navigation	Success Rate	Overall Satisfaction	Time Taken	Error Rate	Other
Overalls	Secretary/BDM	100%	4.2	100%	4.2	9 min 15 s	6.80	-
	Admin	100%	5	100%	5	10 min 12 s	2.50	-
	Overall	100%	4.375	100%	4.375	9 min 31 s	5.57	-

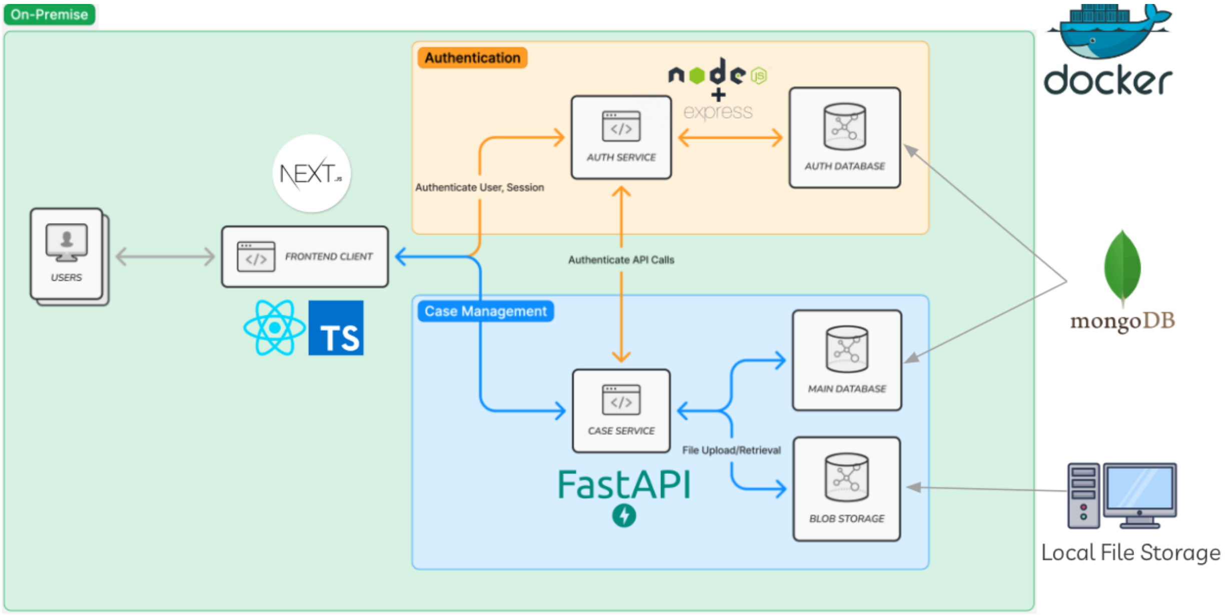
	Question Type	Quantitative	Quantitative	Quantitative	Quantitative	Quantitati…	Quantitati…	Other
	Method	User survey	User survey	Observed & Recorded	User survey	Observed & Recorded	Observed & Recorded	Observed & Recorded
Role	Name	Did the application load/respond quickly enough?	How easy was it to navigate through the application?	Were you able to successfully generate the required forms?	How satisfied are you overall with the application?	Time taken for entire process (subject to role)	Error rate	Remarks
Secretary/…	Avery	Yes	5	Yes	4	9 min 7 s	6	First time with application
Secretary/…	Lemuel Phuna	Yes	4	Yes	5	7 min 58 s	4	First time with application
Secretary/…	Anna Grace	Yes	5	Yes	4	8 min 19 s	5	First time with application
Secretary/…	Esther Ng	Yes	3	Yes	3	12 min 32 s	14	First time with application, Boomer
Secretary/…	Chong Zi Luo	Yes	4	Yes	5	8 min 18 s	5	First time with application
Admin	Dylan	Yes	5	Yes	5	10 min 31 s	3	Have had experience with the application
Admin	Ashton	Yes	5	Yes	5	9 min 52 s	2	Have had experience with the application
Boss	Jia Ling	Yes	4	Yes	4	-	-	Didn't go through scenario, just clicked around

Appendix PP. Cloud to On-Premise Architecture Change

Before



After



Appendix QQ. User Guide

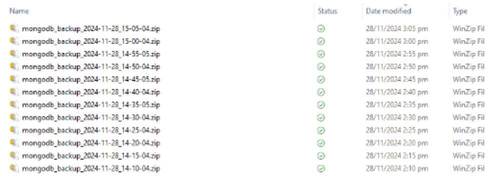
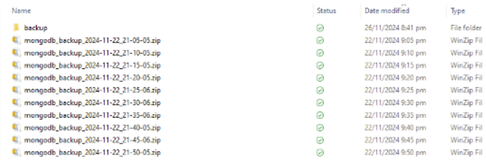
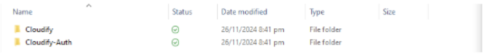
Data Backup User Guide

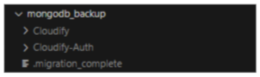
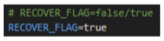
<table>
<tr><th>Steps</th><th>Description</th></tr>
<tr><td>1.</td><td>Environment configuration</td></tr>
<tr><td></td><td><b>BACKUP_SCHEDULE</b>
Cron expression for scheduling the backup process. Default is daily at 1 AM.

BACKUP_SCHEDULE="*/5 * * * *"
<pre>
* * * * *
| | | | |
| | | | +---- Day of the week (0 - 7) [Sunday = 0 or 7]
| | | +------- Month (1 - 12)
| | +---------- Day of the month (1 - 31)
| +------------- Hour (0 - 23)
+---------------- Minute (0 - 59)
</pre>
<b>LOCAL_MONGO_BACKUP_DIRECTORY</b>
Specify the directory where MongoDB backup files will be saved each time the cron job executes.

LOCAL_MONGO_BACKUP_DIRECTORY=C:\Users\cheng\OneDrive - Singapore Management University\localdb_backup</td></tr>
<tr><td>2.</td><td>Run the cloudify_backup container</td></tr>
<tr><td>3.</td><td>The backup will occur at the specified BACKUP_SCHEDULE time and interval.</td></tr>
</table>

Recovery User Guide

<table>
<tr><th>Steps</th><th>Description</th></tr>
<tr><td>1.</td><td>Open OneDrive localdb_backup folder</td></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Date modified</th><th>Type</th></tr>
<tr><td>mongodb_backup_2024-11-28_15-05-04.zip</td><td>✓</td><td>28/11/2024 3:05 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_15-00-04.zip</td><td>✓</td><td>28/11/2024 3:00 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-55-05.zip</td><td>✓</td><td>28/11/2024 2:55 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-50-04.zip</td><td>✓</td><td>28/11/2024 2:50 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-45-05.zip</td><td>✓</td><td>28/11/2024 2:45 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-40-04.zip</td><td>✓</td><td>28/11/2024 2:40 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-35-05.zip</td><td>✓</td><td>28/11/2024 2:35 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-30-04.zip</td><td>✓</td><td>28/11/2024 2:30 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-25-04.zip</td><td>✓</td><td>28/11/2024 2:25 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-20-04.zip</td><td>✓</td><td>28/11/2024 2:20 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-15-04.zip</td><td>✓</td><td>28/11/2024 2:15 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-28_14-10-04.zip</td><td>✓</td><td>28/11/2024 2:10 pm</td><td>WinZip Fil</td></tr>
</table></td></tr>
<tr><td>2.</td><td>Unzip the latest backup file</td></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Date modified</th><th>Type</th></tr>
<tr><td>backup</td><td>✓</td><td>26/11/2024 8:41 pm</td><td>File folder</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-05-05.zip</td><td>✓</td><td>22/11/2024 9:05 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-10-05.zip</td><td>✓</td><td>22/11/2024 9:10 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-15-05.zip</td><td>✓</td><td>22/11/2024 9:15 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-20-05.zip</td><td>✓</td><td>22/11/2024 9:20 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-25-06.zip</td><td>✓</td><td>22/11/2024 9:25 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-30-06.zip</td><td>✓</td><td>22/11/2024 9:30 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-35-06.zip</td><td>✓</td><td>22/11/2024 9:35 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-40-05.zip</td><td>✓</td><td>22/11/2024 9:40 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-45-06.zip</td><td>✓</td><td>22/11/2024 9:45 pm</td><td>WinZip Fil</td></tr>
<tr><td>mongodb_backup_2024-11-22_21-50-05.zip</td><td>✓</td><td>22/11/2024 9:50 pm</td><td>WinZip Fil</td></tr>
</table></td></tr>
<tr><td>3.</td><td>Copy both Cloudify and Cloudify-Auth folders from the unzip backup folder</td></tr>
<tr><td></td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Date modified</th><th>Type</th><th>Size</th></tr>
<tr><td>Cloudify</td><td>✓</td><td>26/11/2024 8:41 pm</td><td>File folder</td><td></td></tr>
<tr><td>Cloudify-Auth</td><td>✓</td><td>26/11/2024 8:41 pm</td><td>File folder</td><td></td></tr>
</table></td></tr>
<tr><td>4.</td><td>Go to the repository and replace both existing Cloudify and Cloudify-Auth folders in the monfdb_backup folder.

Note: Do not delete the .migration_complete file</td></tr>
<tr><td></td><td>
<pre>
⌄ mongodb_backup
  › Cloudify
  › Cloudify-Auth
  .migration_complete
</pre></td></tr>
<tr><td>5.</td><td>Go to the .env file and make sure RECOVER_FLAG is true</td></tr>
<tr><td></td><td>
<pre>
# RECOVER_FLAG=false/true
RECOVER_FLAG=true
</pre></td></tr>
</table>