

License Management System for Little One Eduworld

Nurul Izzati Syazleen Razalli¹, Shahreen Kasim^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: shahreen@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.016>

Article Info

Received: 12 June 2025

Accepted: 3 November 2025

Available online: 30 November
2025

Keywords

License Management, Kindergarten
Licensing, Digital Application System,
Educational Technology, Automated
Document Handling

Abstract

This project presents the development of a License Management System for Little One Eduworld, aimed at overcoming the inefficiencies of a manual, paper-based kindergarten licensing process. The system offers a web-based platform that enables online submission, document uploads, real-time application tracking, and streamlined communication between applicants, verification officers, and administrative staff. Developed using Laravel, Vue.js, Inertia.js, and Tailwind CSS, the system was built under the Agile methodology to allow iterative improvement through continuous stakeholder feedback. Key features include secure user authentication, automated notification delivery, and PDF report generation. Functional, security, and user acceptance testing confirmed the system's effectiveness, reliability, and user-friendliness. Overall, the project successfully enhances operational efficiency, reduces processing delays, and supports MZR Global Sdn Bhd's efforts in expanding its kindergarten network through digital transformation.

1. Introduction

The License Management System for Little One Eduworld was developed as a final year project in collaboration with MZR Global Sdn Bhd, the primary stakeholder. This system addresses the growing need for a more efficient and transparent process for managing kindergarten (*Tadika*) license renewals. Previously, the renewal process was handled manually, involving extensive paperwork and in-person submissions, which often led to delays, difficulties in tracking application statuses, and increased administrative workload. These challenges resulted in slower renewals, a higher risk of errors, miscommunication, and lost documentation, ultimately affecting compliance and service quality for both applicants and staff.

To overcome these issues, the License Management System aims to digitize and automate the entire license renewal workflow. The main objectives are to provide a centralized platform for submitting, reviewing, and approving license renewal applications to enhance communication between applicants and officers through timely notifications and to reduce the administrative burden by minimizing manual data entry and paperwork. The system is designed for *Kerani* (applicants), *Pentadbir* (administrative officers), *Pegawai Penyemakan* (reviewing officer) and *Pegawai Perlulusan* (approval officers) within Little One Eduworld, ensuring that all stakeholders can efficiently manage and track the progress of license renewals.

The scope of the system includes online submission and management of renewal applications, officer review and approval workflows, real-time search and filtering, and status tracking. It also incorporates automated notifications and reminders, including a planned email notification feature, to keep users informed of important

updates and required actions. Applicants and officers can generate and download application documents in PDF format for record-keeping and compliance purposes.

The latest version of the system includes several key modules. The user registration and login module ensure secure access for all users. The application submission module allows users to submit new and renewal applications online, while the document upload feature enables easy management of supporting documents. real-time application tracking [9] keeps both applicants and officers updated on the status and progress of each application. The application review and approval module allow officers to process, comment on, and make decisions regarding applications within the system. Additionally, the generate report feature provides automated reporting for application statistics and compliance tracking. Finally, the renew license feature streamlines the process for renewing existing licenses, further reducing manual intervention. Together, these modules create a seamless, user-friendly, and reliable license management experience for Little One Eduworld, with support and guidance from MZR Global Sdn Bhd.

2. Related Work

This section reviews literature and existing technologies relevant to the development of the License Management System for Little One Eduworld. The review aims to provide a comprehensive understanding of the background, technologies, and systems that inform and influence the proposed solution.

2.1 Licensing System

A licensing system is designed to manage the application, renewal, and tracking of licenses for organizations or individuals. Such as systems streamline the process of submitting applications, verifying documents, and monitoring license statuses. In the context of educational institutions like Little One Eduworld, an efficient licensing system is crucial for ensuring compliance with regulatory requirements and supporting organizational growth.

2.2 Technology Stack

To develop the License Management System for Little One Eduworld, a modern and efficient technology stack was selected to ensure scalability, maintainability, and a seamless user experience. The backend is powered by Laravel [1], a robust PHP framework known for its elegant syntax, built-in security, and support for rapid application development. Laravel [1]'s MVC architecture separates business logic from presentation, making the system easier to maintain and extend.

On the frontend, Vue.js [2] is used to create dynamic and responsive user interfaces. Its component-based structure allows for modular development and real-time updates, ensuring users have a smooth and interactive experience when managing license applications. Tailwind CSS [3] is integrated for styling, providing a utility-first approach that accelerates design workflows and ensures a consistent, visually appealing interface across the system.

Inertia.js [5] acts as a bridge between Laravel [1] and Vue.js [2], enabling the development of single-page applications (SPAs) without the complexity of a full API. This integration allows for fast navigation and real-time updates, reducing load times and enhancing the overall user experience.

For local development and deployment, Laragon is used as the development environment. Laragon simplifies the setup and management of dependencies, making the deployment process more efficient and developer-friendly.

This technology stack provides a flexible, scalable, and user-friendly foundation for the License Management System, supporting the project's goal of digitizing and streamlining the *Tadika* license renewal process for Little One Eduworld.

2.3 Comparison with the Existing System

Several existing systems were analysed to draw insights for the proposed solution. The PETRONAS Licensing Management System (PLMS) [11] excels in managing vendor licenses but is tailored to the oil and gas industry, lacking features for educational institutions. Similarly, the eLESEN system [12] and MalaysiaBiz [13] portal are effective for business licensing but do not cater to educational workflows or multi-agency integration. By addressing these gaps, the proposed License Management System offers real-time tracking, extensive user interface customization, integration capabilities, and additional features tailored to the educational sector.

Table 1 Comparison Table

Features	PLMS[11]	eLESEN[12]	MalaysiaBiz[13]	Proposed System
Online Application Submission	Yes	Yes	Yes	Yes
Real-Time Status Tracking	Yes	Yes	Yes	Yes
Document Management	Yes	Yes	Yes	Yes
User Interface Customization	Limited	Limited	Limited	Extensive
Cost-Effective Maintenance	Moderate	Moderate	Moderate	Moderate
Email Notification	No	No	No	Yes
Renewal Workflow	No	Limited	Limited	Yes
Dashboard & Reporting	Limited	Limited	Limited	Extensive
PDF Generation	No	No	No	Yes
Role-Based Access Control	Yes	Yes	Yes	Yes

3. Methodology

The development of the License Management System for Little One Eduworld adopted an Agile methodology [7], emphasizing flexibility, continuous improvement, and close collaboration with the stakeholder, MZR Global Sdn Bhd. Throughout the project, meetings were held every two to three weeks to review progress, discuss improvements, and prioritize requirements, ensuring the system evolved according to stakeholder feedback and user needs. The process began with requirements gathering and analysis, followed by translating these into wireframes, architectural models, and database schemas. Development utilized Laravel [1] for the backend, Vue.js [2] for the frontend, Tailwind CSS [3] for consistent styling, and Inertia.js [5] for seamless integration. Iterative testing including unit, integration, and user acceptance tests was conducted to ensure reliability and functionality, with feedback from each review cycle incorporated into subsequent development. The project concluded with system deployment, documentation, and stakeholder training, resulting in a robust, user-friendly, and scalable solution tailored to the needs of Little One Eduworld.

3.1 Analysis and Design

The analysis and design phase are crucial in ensuring that the License Management System for Little One Eduworld meets both user and stakeholder requirements. This phase involves understanding the business processes, identifying user roles, and translating requirements into technical specifications and system architecture. Through iterative discussions with MZR Global Sdn Bhd, the system's workflow, data management, and user interactions were mapped out, forming the foundation for subsequent development and implementation.

3.2 System Requirement Analysis

The system is designed to support multiple user roles, including *Kerani* (Clerk), *Pegawai Penyemakan* (Review Officer), *Pegawai Perlulusan* (Approval Officer), and *Pentadbir* (Administrator). Key functional requirements include:

Table 3 Key functional and non-functional requirements

Functional	Non-functional
-Submission and renewal of license applications by clerks.	-Responsive and user-friendly interface.
-Review and approval workflows for officers.	-Data security and privacy.
-Role-based access control and user management by administrators.	-Scalability to support future expansion.
-Real-time status tracking and notifications including email.	-Integration with email systems for notifications.
-Document management and PDF generation.	

Table 3: (Cont)

- Dashboard and reporting features for monitoring application progress.
- Secure authentication and password management.

3.3 System Analysis

3.3.1 Context Diagram

The context diagram in Figure 1 provides a high-level overview of the system’s interactions with external entities. The main system, Pengurusan Lesen Tadika, interacts with users (Kerani, Pegawai Penyemakan, Pegawai Perlulusan) and the email system. Each user role communicates with the system to perform tasks such as submitting applications, reviewing, approving, managing users, and receiving notifications.

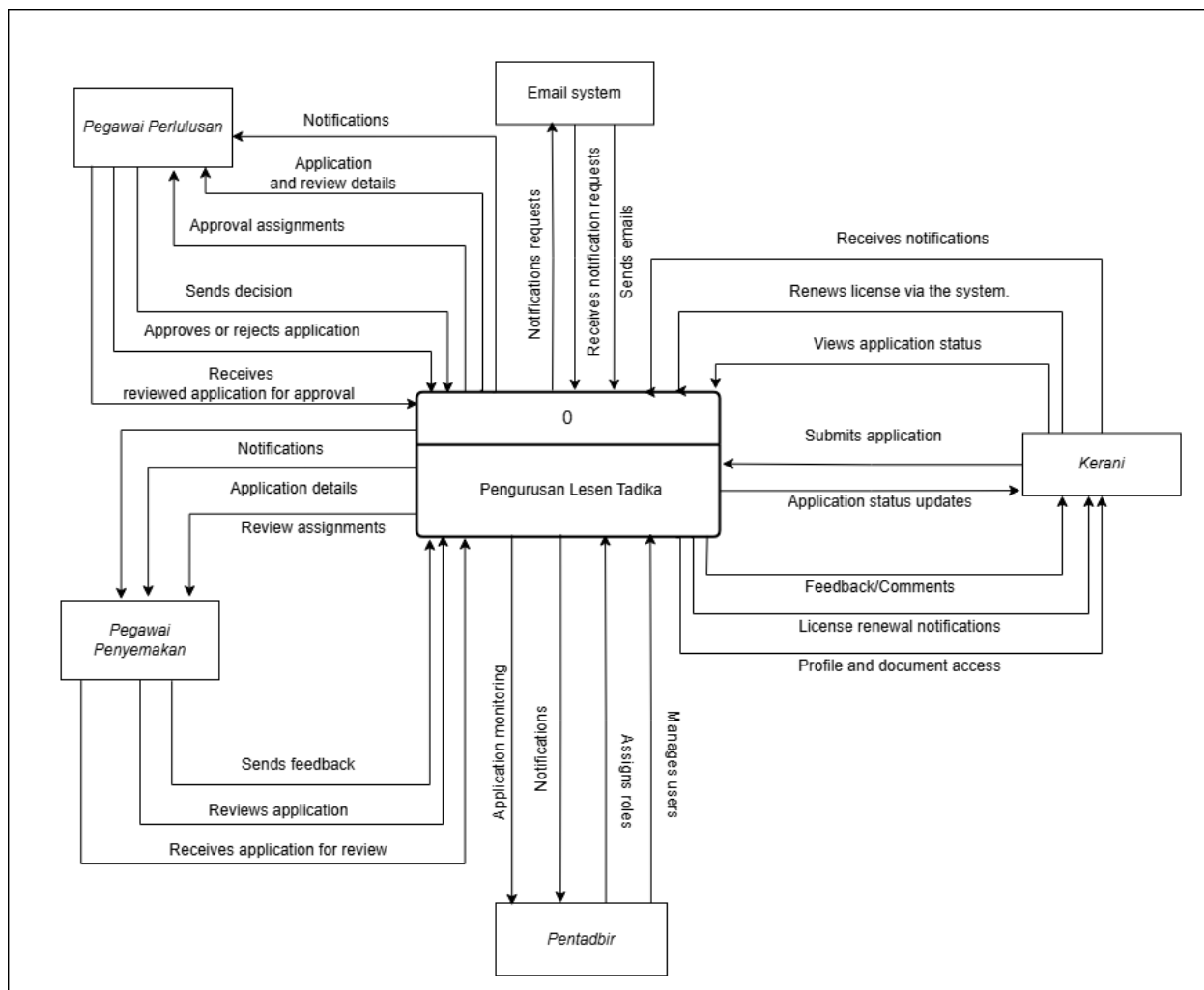


Fig. 1 Context diagram

3.3.2 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) Level 0 illustrates just like in Appendix A, the main processes within the License Management System, highlighting the flow of information between users and the system. The process begins when the *Kerani* submits new or renewal license applications, which are then stored and referenced in the system. Next, the *Pegawai Penyemakan* reviews these applications, updates their verification status, and provides feedback or requests corrections as needed. Once reviewed, the *Pegawai Perlulusan* examines the verified applications and makes final approval decisions, updating the application status accordingly. The *Pentadbir* is

responsible for managing user accounts, assigning roles, and monitoring overall system activity to ensure smooth operations. Throughout these processes, the system actively sends notifications and email alerts to users, keeping them informed about application statuses and any required actions, this is for ensuring efficient communication and workflow management.

3.3.3 Entity Relationship Diagram (ERD)

The ERD in Appendix B, is the database structure, showing entities such as users, roles, applicants, renew_license, documents, and related tables. Relationships are established between users and roles, applicants and their applications, and documents linked to applications. This structure supports efficient data management, role-based access, and traceability of all license-related activities.

3.4 System Design

The system design phase translates requirements and analysis into detailed workflows and process logic, represented by flowcharts for each user role:

Table 4 System design

System Design	Description	Appendix
Login and Registration Flow	Users register and log in using email and password, with options for password reset and recovery	C
<i>Pentadbir</i> (Administrator) Flow	Administrators log in, view pending jobs, manage applications, roles, and categories, and perform CRUD operations on system data	C
<i>Pegawai Perlulusan</i> (Approval Officer) Flow	Approval officers log in, review pending applications, make final decisions, and manage their workflow based on job status	C
<i>Pegawai Penyemakan</i> (Review Officer) Flow	Review officers log in, review new and renewal applications, update statuses, and provide feedback	C
<i>Kerani</i> (Clerk) Flow	Clerks register, log in, submit new or renewal applications, edit previous applications, and track application status	C

3.5 Testing Phase

Testing was conducted in multiple stages to ensure the system worked reliably and met user expectations. Unit testing was performed on individual features like user registration and form validation to confirm each worked correctly. Integration testing checked that different modules communicated properly, such as how submitted applications were reviewed by officers. User Acceptance Testing (UAT) involved real users from MZR Global Sdn Bhd testing the system's workflows. Feedback from these sessions helped identify and fix usability issues, ensuring a smoother experience for all user roles.

4. Result and Discussion

4.1 Implementation

The License Management System for Little One Eduworld was successfully implemented using Laravel [1] for backend development, Vue.js [2] for frontend interfaces, Tailwind CSS [3] for styling, and Inertia.js [5] for seamless integration. The system supports multiple user roles that are *Kerani*, *Pegawai Penyemakan*, *Pegawai Perlulusan*, and *Pentadbir*. Each of their roles with tailored workflows for license application, review, approval, and administration. Key modules such as application submission, document management [9], notification delivery, and PDF generation were developed and iteratively refined based on stakeholder feedback. The system was deployed in a live environment and is now fully operational.

4.2 Interface Design and Coding Function

In this section, the main functional modules of the system interface and their coding implementations are described. Each module is designed to address a specific operational requirement, ensuring smooth and efficient user experience. The modules covered include user registration and login module, applicant management for submitting and tracking applications, application review and approval for officer processing, search and filtering for efficient data retrieval, a notification system for timely alerts, PDF generation for document downloads, a dedicated renew license feature, and dashboard and reporting for monitoring application statistics and pending

tasks. The following subsections will present the interface design and key code segments for each module, demonstrating how the system supports users and officers in managing license applications effectively.

4.2.1 User Registration and Login module

This module provides secure user authentication and registration. The registration form collects user details such as name, email, and password, with password confirmation and validation. The login form allows users to enter their credentials and access the system. Both forms use Vue's useForm for reactive state and Inertia.js [5] for seamless server communication. Error handling and field validation are integrated for a smooth user experience.

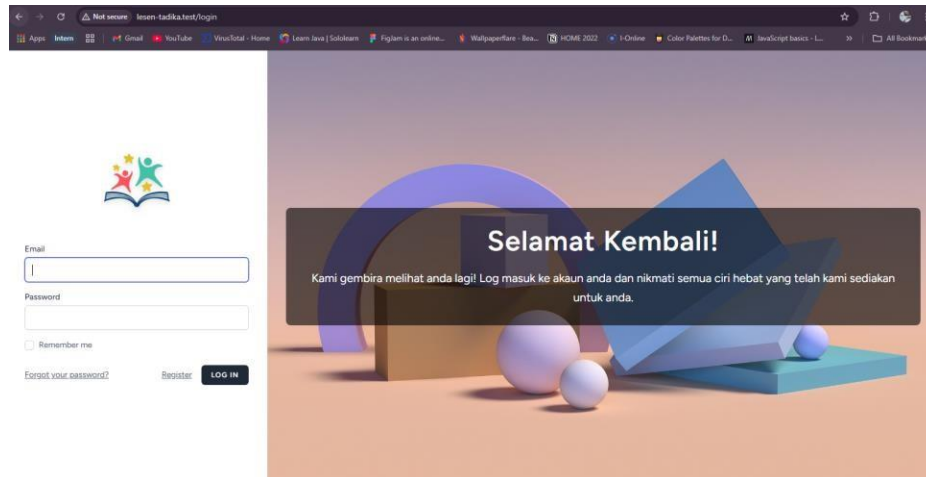


Fig. 2 Login

```

<script setup>
import { Head, Link, useForm } from '@inertiajs/vue3';
import AuthenticationCardLogo from '@/Components/AuthenticationCardLogo.vue';
import Checkbox from '@/Components/Checkbox.vue';
import InputError from '@/Components/InputError.vue';
import InputLabel from '@/Components/InputLabel.vue';
import PrimaryButton from '@/Components/PrimaryButton.vue';
import TextInput from '@/Components/TextInput.vue';

defineProps({
  canResetPassword: Boolean,
  status: String,
});

const form = useForm({
  email: '',
  password: '',
  remember: false,
});

const submit = () => {
  form.transform(data => ({
    ...data,
    remember: form.remember ? 'on' : '',
  })).post(route('login'), {
    onFinish: () => form.reset('password'),
  });
};
</script>

```

(a)

```

<script setup>
import { Head, Link, useForm } from '@inertiajs/vue3';
import AuthenticationCardLogo from '@/Components/AuthenticationCardLogo.vue';
import Checkbox from '@/Components/Checkbox.vue';
import InputError from '@/Components/InputError.vue';
import InputLabel from '@/Components/InputLabel.vue';
import PrimaryButton from '@/Components/PrimaryButton.vue';
import TextInput from '@/Components/TextInput.vue';

const form = useForm({
  name: '',
  email: '',
  password: '',
  password_confirmation: '',
  terms: false,
});

const submit = () => {
  form.post(route('register'), {
    onFinish: () => form.reset('password', 'password_confirmation'),
  });
};
</script>

```

(b)

Fig. 3 User(a)Login module;(b) Register module

4.2.2 Application Submission

This module, Figure 4 enables users to submit new license applications. The form collects applicant information such as name, identification number, and contact details. Vue's useForm is used for two-way data binding and form submission. The interface provides clear labels and input fields, ensuring users can easily provide all required information for their application.

The screenshot shows a web browser window with the URL 'lesen-tadika.test/applicant/create'. The page title is 'Pengurusan Lesen Tadika'. The main content area is titled 'Tambah Pemohonan Baru' and contains a form with the following fields:

- (a) Nama: Input field containing 'Kerani'
- (b) No. Kad Pengenalan: Empty input field
- (c) Warganegara: Empty input field
- (d) Alamat Rumah: Empty input field
- (e) Alamat pos jika berlainan daripada (d): Empty input field

At the bottom of the form, there is a blue alert message: 'Jika tiada, sila tulis "Tiada" di ruang kosong.'

Fig.4 Application Submission

```
<script setup>
import AppLayout from '@/Layouts/AppLayout.vue';
import { ref } from 'vue';
import { useForm } from '@inertiajs/vue3';
import Swal from 'sweetalert2';

const props = defineProps({
  errors: Object,
  user: Object,
});

const permission = ref(['kerani', 'admin', 'Pegawai Perlulusan', 'Pegawai Penyemakan']);
const form = useForm({
  nama: props.user.name,
  kad_pengenalan: null,
  warganegara: null,
  alamat_rumah: null,
  alamat_berlainan: null,
  email: props.user.email,
  telefon_r: null,
  telefon_b: null,
  telefon_p: null,
  faks: null,
  user_id: props.user.id,
});
```

Fig. 5 Application Submission code

4.2.3 Document Upload

This document upload module, Figure 6, allows users to upload required documents such as applicant's IC, child information and worker list as part of their application. The backend validates file types and sizes, and stores documents securely. The interface provides file input fields and feedback on upload status, ensuring users can manage their documents efficiently.

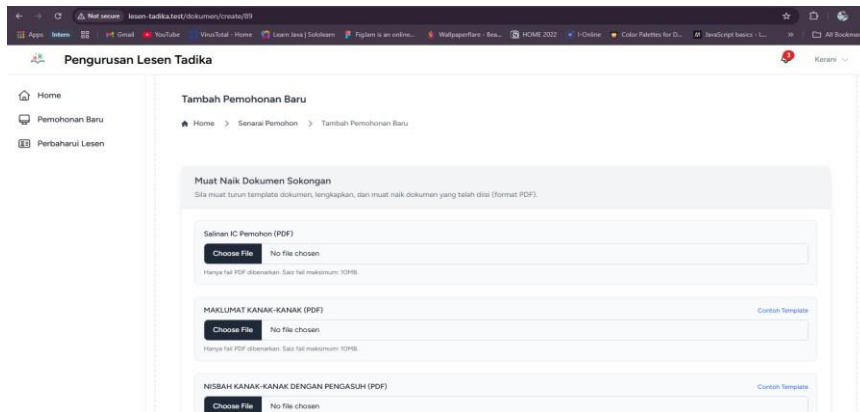


Fig. 6 Document Upload

```

class DocumentController extends Controller
{
    public function store(Request $request)
    {
        $request->validate([
            'applicant_id' => 'required|exists:applicants,id',
            'salinan_ic' => 'nullable|file|mimes:pdf|max:10240',
            'maklumat_kanak' => 'nullable|file|mimes:pdf|max:10240',
            'nisbah_kanak' => 'nullable|file|mimes:pdf|max:10240',
            'senarai_pekerja' => 'nullable|file|mimes:pdf|max:10240',
        ]);

        $applicantId = $request->input('applicant_id');

        // Salinan IC Pemohon
        if ($request->hasFile('salinan_ic')) {
            $file = $request->file('salinan_ic');
            $filePath = $file->store('documents', 'public');
            Document::create([
                'applicant_id' => $applicantId,
                'nama_dokumen' => 'Salinan IC Pemohon',
                'file_dokumen' => $filePath,
            ]);
        }

        // MAKLUMAT KANAK-KANAK
        if ($request->hasFile('maklumat_kanak')) {
            $file = $request->file('maklumat_kanak');
            $filePath = $file->store('documents', 'public');
            Document::create([
                'applicant_id' => $applicantId,
                'nama_dokumen' => 'MAKLUMAT KANAK-KANAK',
                'file_dokumen' => $filePath,
            ]);
        }
    }
}
    
```

Fig. 7 Document Upload code

4.2.4 Real-Time Application Tracking

This module, real-time application tracking [9] lets users and officers track the status of applications in real time. The Figure 8 interface displays the current status, comments from officers, and allows status updates. Conditional rendering is used to show relevant actions and information based on the application’s status and user role, improving transparency and communication.

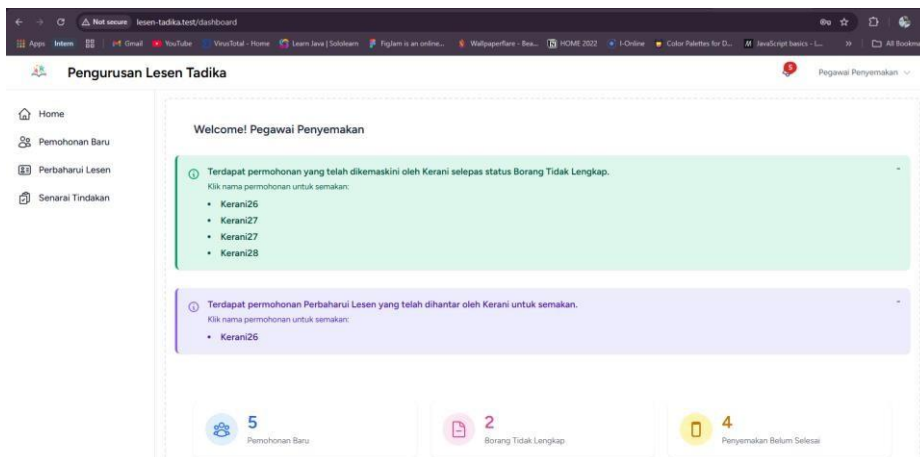


Fig. 8 Real-Time Application Tracking

```

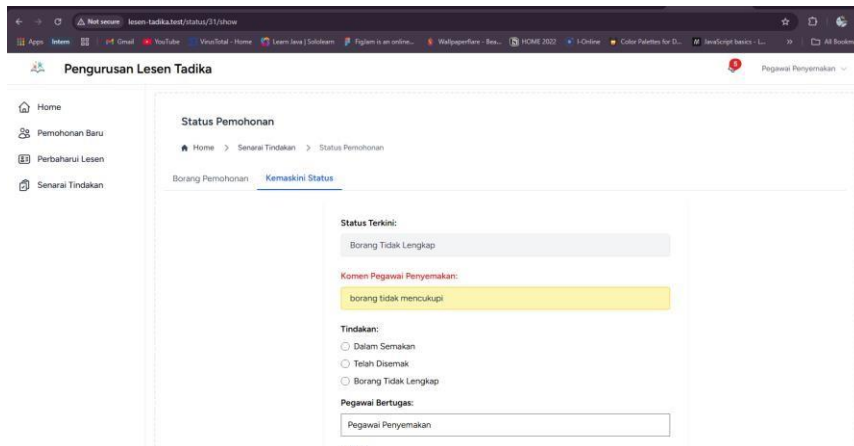
<!-- Pegawai Perluasan Notification Block: Telah Disemak -->
<div
  v-if="(userRole === 'Pegawai Perluasan' || userRole === 'Admin') && dashboardData && dashboardData.perlulusNotificati
  class="mb-8"
  >
  <div class="bg-blue-100 border-1-4 border-blue-500 text-blue-800 p-4 rounded-lg shadow flex items-start relativ
    <svg class="w-6 h-6 text-blue-500 mr-3 mt-1 flex-shrink-0 fill="none" stroke="currentColor" viewBox="0 0 24 24"
      <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M13 16h-1v-4h-1m12 20a8 8 0 0 1
    </svg>
    <div class="flex-1 v-show="!perlulusTelahDisemakNotiBlockMinimized">
      <div class="font-semibold mb-1">
        Terdapat permohonan yang telah disemak oleh Pegawai Penyemakan dan menunggu kelulusan.
      </div>
      <div class="text-sm mb-2">Klik nama permohonan untuk proses kelulusan:</div>
      <ul class="list-disc list-inside space-y-1">
        <li v-for="noti in dashboardData.perlulusNotifications" :key="noti.id">
          <a
            :href="route('status.show', noti.id)"
            class="font-semibold text-blue-900 hover:underline hover:text-blue-700 transition"
          >
            {{ noti.nama }}
          </a>
        </li>
      </ul>
    </div>
    <div class="flex-1 v-show="perlulusTelahDisemakNotiBlockMinimized">
      <div class="font-semibold">
        Terdapat permohonan telah disemak oleh Pegawai Penyemakan.
      </div>
    </div>
  </div>

```

Fig. 9 Real-Time Application Tracking code

4.2.5 Application Review and Approval

This module is for officers to review, comment on and approve or reject applications. Notifications are shown in Figure 10 for applications that require attention. The interface lists pending applications and provides quick access for officers to process them. This ensures efficient workflow and timely decision-making.

**Fig.10** Application Review and Approval

```

<!-- Kemaskini Status Tab -->
<div v-else class="max-w-2xl mx-auto bg-white p-8 rounded-lg shadow">
  <!-- Current Status Display -->
  <div class="mb-6">
    <label class="block font-semibold mb-2">Status Terkini:</label>
    <div class="px-4 py-2 bg-gray-100 rounded border border-gray-200 text-gray-700">
      {{ Applicants.status || 'Tiada status' }}
    </div>
  </div>
  <!-- Show komen if status is Borang Tidak Lengkap and komen exists -->
  <div v-if="Applicants.status === 'Borang Tidak Lengkap' && Applicants.komen" class="mb-6">
    <label class="block font-semibold mb-2 text-red-600">Komen Pegawai Penyemakan:</label>
    <div class="px-4 py-2 bg-yellow-100 rounded border border-yellow-300 text-gray-800">
      {{ Applicants.komen }}
    </div>
  </div>
  <form @submit.prevent="updateStatus">
    <div class="mb-4">
      <label class="block font-semibold mb-2">Tindakan:</label>
      <div class="flex flex-col space-y-2">
        <!-- For Pegawai Perlulusan, show renew approve/reject options if status is Perbaharui Lesen Telah -->
        <template v-if="Applicants.status === 'Perbaharui Lesen Telah Disemak' && (userRole?.toLowerCase() === 'pegawai-perlulusan')">
          <label class="inline-flex items-start break-words max-w-full">
            <input
              type="radio"
              class="form-radio mt-1"
              v-model="form.status"
              value="Perbaharui Lesen Diluluskan"
            />
          </label>
        </template>
      </div>
    </div>
  </form>
</div>

```

Fig. 11 Application Review and Approval code

4.2.6 Generate Report

This module enables officers and administrators to generate and download application reports in PDF format. The Figure 12 interface provides a button to trigger PDF generation. The backend processes the request and returns a downloadable file. This feature supports record-keeping and reporting requirements.

```

const downloadPdf = async (id) => {
  try {
    const response = await axios.post(
      '/generate-pdf',
      { id: id },
      { responseType: 'blob' }
    );
    const url = window.URL.createObjectURL(new Blob([response.data]));
    const link = document.createElement('a');
    link.href = url;
    link.setAttribute('download', 'applicant.pdf');
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
  } catch (error) {
    console.error('Error generating PDF:', error);
  }
};
</script>

```

Fig. 12

4.2.7 Renew License Feature

This renew license feature module allows users to renew their licenses. When a renewal is initiated, the system checks for existing renewals to prevent duplicates, then clones the applicant's data and related records for the new renewal process. The Figure 13 interface guides users through the renewal steps and ensures all necessary information is carried over, streamlining the renewal workflow.

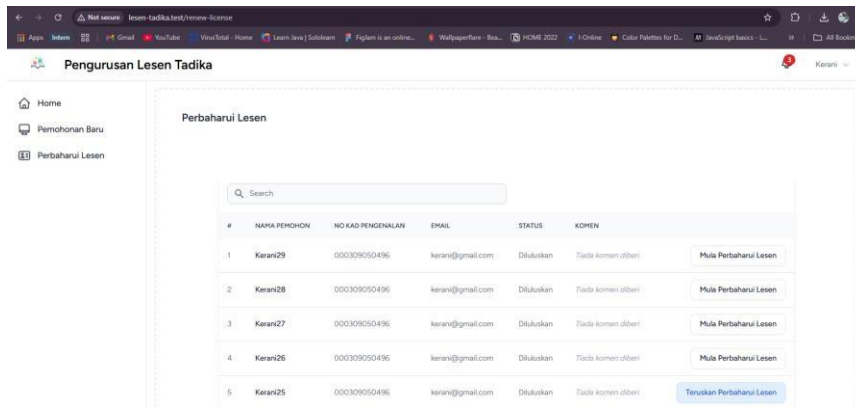


Fig. 13 Renew License Feature

```
public function renew($id)
{
    if (auth()->user()->role !== 'kerani') {
        abort(403);
    }

    $old = Applicant::findOrFail($id);

    // Check if a renewal already exists for this old applicant and is not completed
    $existingRenew = \App\Models\RenewLicense::where('previous_applicant_id', $old->id)
        ->whereHas('applicant', function($q) {
            $q->where('status', 'Perbaharui Lesen');
        })
        ->first();

    if ($existingRenew) {
        // Redirect to edit the existing renewal, do not clone again
        return redirect()->route('applicant.edit', $existingRenew->applicant_id);
    }

    // Clone applicant (except id, created_at, updated_at, status, etc)
    $new = $old->replicate(['id', 'created_at', 'updated_at', 'status']);
    $new->status = 'Perbaharui Lesen'; // <-- Use the new kategori status exactly as in your config
    $new->save();
}
```

Fig. 14 Renew License Feature code

5. Result and Discussion

5.1 Test Plan Result

Comprehensive testing was conducted to ensure the system’s reliability, security, and usability. All test cases for user, administrator and officer functionalities, as well as security requirements, passed successfully.

This Table 5 evaluates how well the system supports end-users (*Kerani*) in completing core actions such as registration, login, application submission, and status tracking. From the results, it confirms that the interface is intuitive and that backend processes accurately respond to user input, ensuring a reliable application experience.

Table 5 Functionality Test Plan Result for Kerani (User)

Module	Test Plan	Expected Result	Result
Registration	User enters incomplete data	Error message displayed	Pass
Registration	User enters valid data	Registration successful	Pass
Login	User enters incorrect email/password	Error message displayed	Pass
Login	User enters correct email/password	Login successfully, redirect to dashboard	Pass

Table 5: (Cont)

Application	User submits new/renewal application	Application submitted, status updated	Pass
Application	User views application status	Status displayed correctly	Pass
Application	User edits previous application	Application updated successfully	Pass
Notification	User receives email/notification on status change	Notification received	Pass

This Table 6 demonstrates the system’s capability to support *Pegawai Penyemakan* (Review Officers) in reviewing and verifying application data. It confirms that officers can access submitted applications, update verification statuses, provide comments, and trigger applicant notifications. All of the key elements in an efficient review workflow.

Table 6 Functionality Test Plan Result for Pegawai Penyemakan

Module	Test Plan	Expected Result	Result
Login	Officer logs in with valid credentials	Login successful	Pass
Review	Officer reviews new/renewal applications	Application details displayed	Pass
Review	Officer updates verification status	Status updated, applicant notified	Pass
Review	Officer adds comments/requests correction	Comments saved, applicant notified	Pass
Notification	Officer receives notification for new applications	Notification received	Pass

Next, this test Table 7 ensures that *Pegawai Perlulusan* (Approval Officers) can access and approve verified applications. It validates that status updates, approvals, rejections, and related notifications function as expected, ensuring that only complete and accurate applications proceed to final approval.

Table 7 Functionality Test Plan Result for Pegawai Perlulusan

Module	Test Plan	Expected Result	Result
Login	Officer logs in with valid credentials	Login successful	Pass
Approval	Officer reviews verified applications	Application details displayed	Pass
Approval	Officer approves/rejects application	Status updated, applicant notified	Pass
Notification	Officer receives notification for applications	Notification received	Pass

Table 8 verifies that administrative users can manage core backend operations such as logging in, user management (CRUD), assigning roles, monitoring system activity, and receiving notifications. These functionalities are essential to maintaining the system's integrity and workflow continuity.

Table 8 *Functionality Test Plan Result for Administrator*

Module	Test Plan	Expected Result	Result
Login	Admin logs in with valid credentials	Login successful	Pass
User Management	Admin creates/updates/deletes user accounts	User accounts managed successfully	Pass
Role Management	Admin assigns roles	Roles assigned correctly	Pass
Monitoring	Admin views dashboard and system activity	Data displayed correctly	Pass
Notification	Admin receives system notifications	Notification received	Pass

Lastly, Table 9 test plan confirms the implementation of essential security measures including proper error handling, password complexity and obscuration, SQL injection prevention, and role-based access control. Passing these tests demonstrates that the system is robust and secure for user data protection.

Table 9 *Security Test Plan Result*

No	Check List	Result
1	Error messages do not reveal sensitive info	Pass
2	Password complexity enforced	Pass
3	Password length minimum 8 characters	Pass
4	Passwords obscured in textbox	Pass
5	Role-based access enforced	Pass
6	Prevent SQL injection	Pass

5.2 User Acceptance Testing

User acceptance testing (UAT) was conducted with representatives from each user role. All acceptance criteria were met, and users confirmed that the system effectively streamlined the license management process.

Table 10 documents result from User Acceptance Testing (UAT). This is actual users validating the system's ability to meet their needs. The outcomes confirm that users can register, login, submit and track applications, and receive notifications, ensuring the system's practicality in real-world use.

Table 10 *User Acceptance Form Result*

Test Plan	Result
User can register account	Pass
User can login into the system	Pass
User can update their information	Pass
User can submit new/renewal application	Pass
User can view application status	Pass

Table 10: (Cont)

User receives notifications on status changes	Pass
User can reset password	Pass

This result from Table 11 confirms that administrators tested and validated the critical backend management functionalities, including login, user account management, role assignments, system monitoring, and access to reporting features, indicating operational readiness from the admin's perspective.

Table 11 Administrator Acceptance Form Result

No	Test Plan	Result
1	The administrator can log into the system.	Pass
2	The administrator can perform CRUD on user accounts.	Pass
3	The administrator can assign roles.	Pass
4	The administrator can monitor system activity.	Pass
5	The administrator can view reports and dashboards.	Pass

Continue with Table 12 shows acceptance results from *Pegawai Penyemakan*, confirming that the review features meet expectations, including access to applications, ability to update verification statuses, leave comments, and receive alerts for new submissions.

Table 12 Pegawai Penyemakan Acceptance Form Result

No	Test Plan	Result
1	Officers can log into the system.	Pass
2	Officers can review new/renewal applications.	Pass
3	Officers can update verification status.	Pass
4	Officers can add comments or request corrections.	Pass
5	Officer receives notifications for new applications.	Pass

The final acceptance test Table 13 verifies that *Pegawai Perlulusan* can access verified applications, approve or reject them, and receive timely notifications. It confirms the system's support for streamlined final approval workflows.

Table 13 Pegawai Perlulusan Acceptance Form Result

No	Test Plan	Result
1	Officers can log into the system.	Pass
2	Officers can review verified applications.	Pass
3	Officers can approve or reject applications.	Pass
4	Officer receives notifications for applications ready for approval.	Pass

6. Conclusion

In conclusion, the License Management System for Little One Eduworld has effectively transformed the manual kindergarten licensing process into a streamlined, digital solution that improves efficiency, reduces errors, and enhances transparency. Developed using Laravel [1], Vue.js [2], Tailwind CSS [3], and Inertia.js [5] within an Agile framework, the system supports key functionalities such as online application submission, real-time status tracking, document management [9], and role-based workflows. Comprehensive testing validated its performance and usability, ensuring it meets the needs of applicants, verification officers, and administrators. By addressing the limitations of the previous manual system, this project not only supports the operational goals of MZR Global

Sdn Bhd but also sets a strong foundation for future enhancements, including broader regulatory integration and scalability across additional sectors.

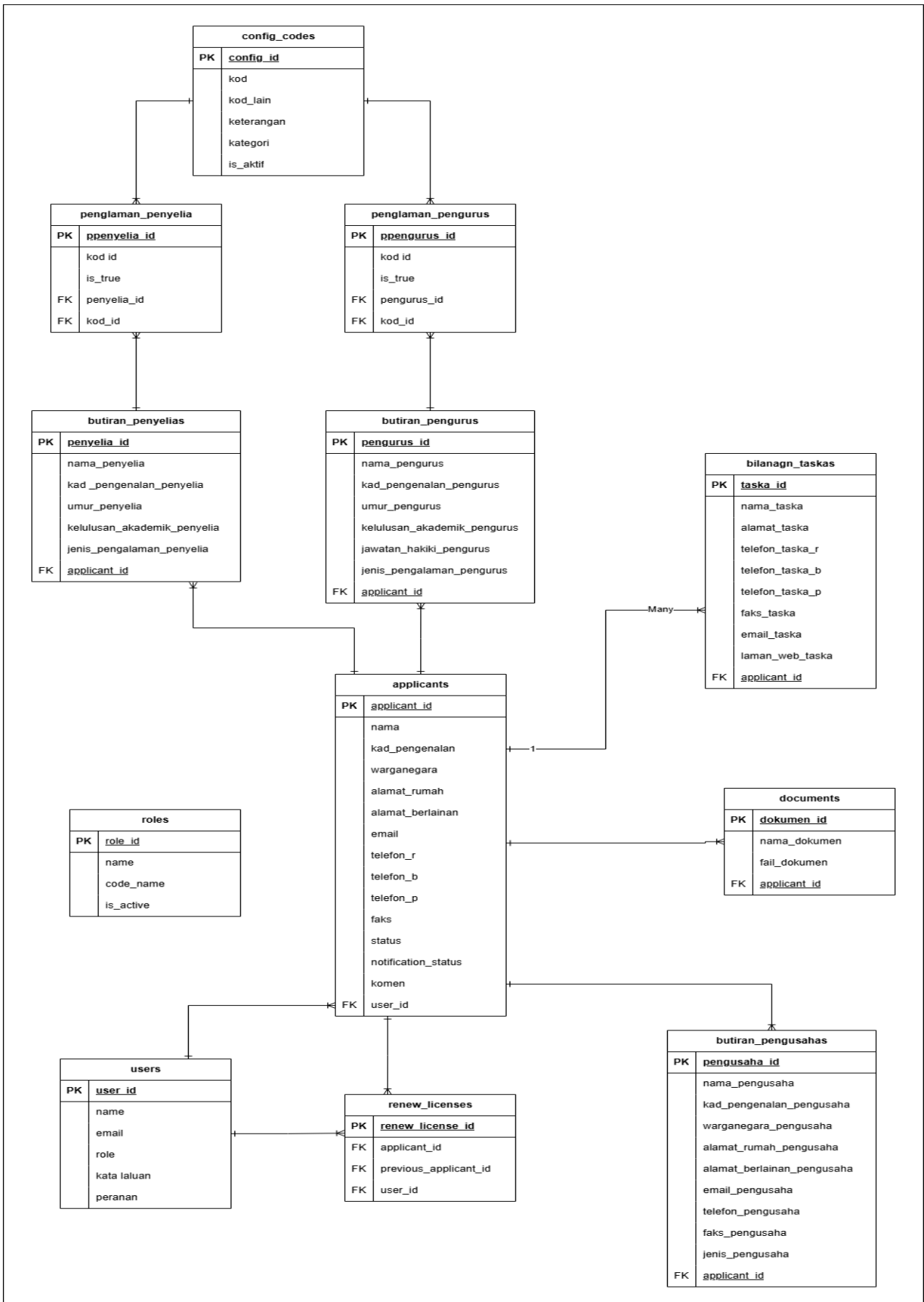
Acknowledgement

I am deeply grateful to the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, for their unwavering support and resources throughout this project. My heartfelt thanks go to my supervisor, PM Dr. Shahreen Kasim, whose invaluable guidance, insights, and encouragement have been instrumental in shaping this research. I also extend my sincere appreciation to MZR Global Sdn Bhd, the stakeholder company, for their collaboration and support, which provided real-world perspectives that enriched the development of this system.

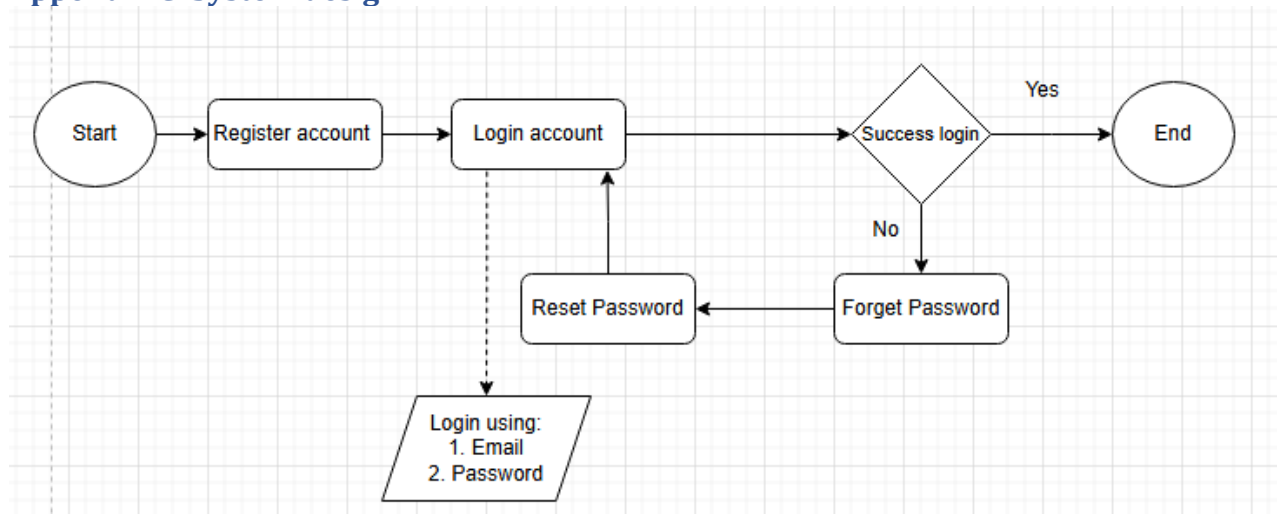
References

- [1] Crooks, M. (2024, September 10). Why Laravel is the best choice in PHP web development. Medium. <https://medium.com/@crooksm/why-laravel-is-best-choice-in-php-web-development-2024>
- [2] Piastou, M. (2023). Comprehensive performance and scalability assessment of front-end frameworks: React, Angular, and Vue.js. World Journal of Advanced Engineering Technology and Sciences, 9(2), 366–376. <https://doi.org/10.5281/zenodo.10011292>
- [3] Rahman, H. (2025, May 6). Modern Tailwind CSS in 2025: A utility-first revolution refined. Medium. <https://medium.com/@hrahman/tailwind-utility-first-2025>
- [4] Varenek, M. (2025, June 5). Tailwind CSS: The utility-first revolution in frontend development. GeeksforGeeks. <https://www.geeksforgeeks.org/tailwind-css-utility-first-2025>
- [5] Tannenbaum, J. (2024, December 16). Announcing Inertia 2.0: Redefining frontend development for Laravel. Laravel News. <https://laravel-news.com/inertia-2-announcement>
- [6] Majithiya, D. (2025, January 29). Why Vue.js is essential for modern web development in 2025. ShivLab. <https://shivlab.com/vuejs-2025>
- [7] Askarpour, M., Kokaly, S., & Rafeie, S. (2024, September 19). An exploration of agile methods in the automotive industry: Benefits, challenges and opportunities. arXiv. <https://arxiv.org/abs/2409.10200>
- [8] SPS at Wake Forest University. (2023, December). 5 benefits of Agile project management. <https://sps.wfu.edu/blog/benefits-of-agile-project-management>
- [9] Alade, S. M. (2023). Design and implementation of a web-based document management system. International Journal of Information Technology and Computer Science, 15(2), 35–53. <https://doi.org/10.5815/ijitcs.2023.02.04>
- [10] Laravel. (2024). Laravel documentation (v10). <https://laravel.com/docs/10.x>
- [11] PETRONAS. (2020). PETRONAS Licensing Management System (PLMS). Retrieved from <https://www.petronas.com>
- [12] Majlis Bandaraya Ipoh. (2021). eLESEN: Sistem Perlesenan Perniagaan. Retrieved from <https://mbi.gov.my>
- [13] MAMPU. (2020). MalaysiaBiz: Business Licensing Portal. Retrieved from <https://malaysiabiz.mampu.gov.my>

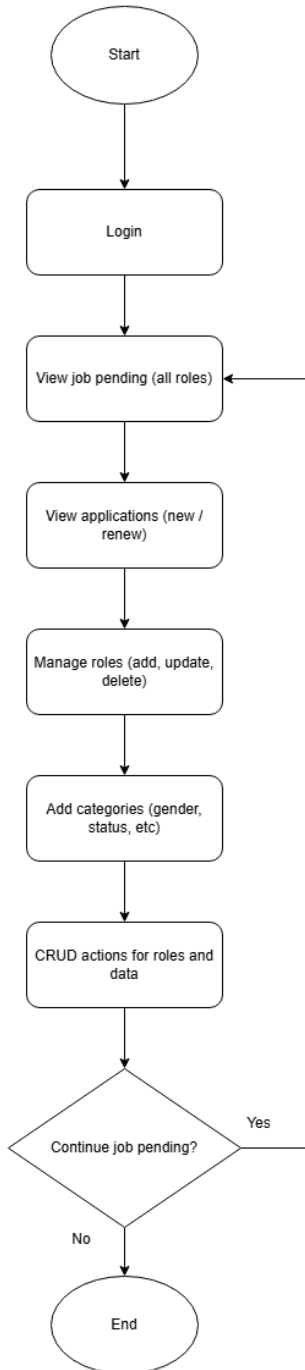
Appendix A: DFD



Appendix C: System design



Pentadbir



Pegawai Perlulusan

