

Summary Specification	
BIL. SEBUTHARGA/QUOTATION REF.:	<u>UBD/Q/046/2026 (OAVC (ULG) ) - [G]</u>
TARIKH DIKLANKAN :	<u>02 Jun 2026 [Selasa]</u>
TARIKH TUTUP :	<u>09 Jun 2026 [Selasa]</u> JAM: 2 Petang
DOCUMENT FEE : B\$ 10.00	

**Important Note:**

The Official PDF quotation/tender form will be send via email once UBD Official receipt issued and send to vendors/supplier for confirmation of payment received.

Please take note that the document fee for each of the New and Extended quotations will be BND 10.00 each

**Payment Information**

Quotations/tenders are identified as follows:

[G] – Kerajaan (Government Account)

Please ensure payment is made to the correct account:

[G] → UNIVERSITI BRUNEI DARUSSALAM [KERAJAAN]

Payment details and account numbers are available under “Mode of Payment” on the UBD website

**Note:** Please ensure that you provide the correct details when making the payment.

Please email proof of the above payment to [helpdesk.epay@ubd.edu.bn](mailto:helpdesk.epay@ubd.edu.bn) , [finance@ubd.edu.bn](mailto:finance@ubd.edu.bn), [eprocurement.support@ubd.edu.bn](mailto:eprocurement.support@ubd.edu.bn).

TITLE :		<u>TO DELIVER A SOFTWARE SYSTEM FOR ACADEMIC SCHEDULING, STUDENT REGISTRATION &amp; ASSESSMENT MANAGEMENT, UNIVERSITI BRUNEI DARUSSALAM</u>
No.	Quantity	SPECIFICATIONS
1		Please refer to the attached document for the full specification

**UNIVERSITI BRUNEI DARUSSALAM**

AI SEnse Unit

---

**OPEN QUOTATION**

Request for Proposal — Technical Specification

Ref: UBD-OQ-2026-002

## To Deliver A Software System for Academic Scheduling, Student Registration & Assessment Management

Quotation Reference                      **UBD-OQ-2026-002**

Date of Issue

---

Quotation Closing Date

---

Classification                              **CONFIDENTIAL — For Quotation Respondents Only**

## 1. Introduction

Universiti Brunei Darussalam (UBD) invites quotations from suitably qualified vendors for the design, development, testing, and delivery of **specific software modules**, aligned to the GenBESTARI undergraduate curriculum framework effective from Academic Year 2026/2027.

This document constitutes the complete technical specification for this Open Quotation (Phase 2B). The scope defined herein is self-contained and independent. Vendors are expected to deliver all items listed in Section 2 as a complete, integrated, and production-ready system.

Attribute	Details
System Name	Academic Scheduling, Student Registration & Assessment Management Systems
Institution	Universiti Brunei Darussalam (UBD)
Curriculum Framework	GenBESTARI — Undergraduate Degree Rules & Regulations, Senate-approved January 2026
Target Users	Students, Lecturers, Programme Leaders, Faculty Administrators, Exam Office, Registry, Deans
Platform	Web-based; desktop and mobile responsive; hosted on university infrastructure

## 2. Scope of Work

The vendor shall design, develop, test, document, and deliver all items listed in the table below. This constitutes the complete scope of this quotation. Each item is a distinct deliverable subject to the acceptance criteria in Section 5.

No.	Module / Component	Key Functionalities	Qty
1	Module Offering Management	<ul style="list-style-type: none"> <li>– Faculty module offering submission per semester</li> <li>– Not all modules run every semester — offering list is the authoritative gate</li> <li>– Configure total weekly contact hours per module</li> <li>– Define block breakdown: Lecture hrs + Tutorial hrs + Practical hrs</li> <li>– Create separate timetable blocks per component (each block has own day, time, room type)</li> <li>– Option to schedule blocks back-to-back (same session) or separate days</li> <li>– Set room type requirement per block (lecture theatre, computer lab, science lab, studio)</li> <li>– Set expected enrolment quota and number of parallel sections</li> <li>– Academic / Exam Office review &amp; approval workflow</li> <li>– Room assignment upon approval (matched by capacity &amp; equipment)</li> <li>– Publish approved offering &amp; timetable blocks to Registration module</li> <li>– Offering amendment request after approval (with re-approval)</li> <li>– Historical offering archive per module per semester</li> </ul>	1
2	Class & Tutorial Management	<ul style="list-style-type: none"> <li>– Timetable compiled from approved Module Offering blocks</li> <li>– Each module entry shows all blocks: lecture block(s) + tutorial/practical block(s)</li> <li>– Blocks on separate days OR back-to-back — both supported</li> <li>– Room confirmed per block at offering approval stage</li> <li>– Lecturer / tutor assignment per block</li> <li>– Published timetable visible to students before and during registration</li> <li>– Individual student timetable auto-assembled from registered modules</li> <li>– Weekly calendar view (web &amp; mobile), export to PDF / iCal</li> </ul>	1

		<ul style="list-style-type: none"> <li>– Timetable amendment workflow (post-publish changes require re-approval)</li> <li>– Make-up class scheduling with student notification</li> <li>– Blended / hybrid session type support (on-campus / online flag per block)</li> <li>– Lecturer timetable view and room occupancy view</li> </ul>	
3	<b>Module Registration</b>	<ul style="list-style-type: none"> <li>– Timetable-first model: approved blocks visible to students before registration opens</li> <li>– Student registration portal with live weekly calendar view</li> <li>– Force-registered modules auto-added to calendar first</li> <li>– Adding a module simultaneously plots ALL its blocks (lecture + tutorial/practical) on calendar</li> <li>– Real-time visual clash detection — conflicting slots highlighted before confirmation</li> <li>– System prevents registration if any block clashes with existing calendar entries</li> <li>– Prerequisite &amp; antirequisite validation at point of registration</li> <li>– Enforce GenBESTARI credit load: Min 20 MC / Max 24 MC per semester (except final semester); overload &gt;24 MC requires Dean approval + Grade B average</li> <li>– Module quota enforcement with waitlist management</li> <li>– Multi-section selection for large modules</li> <li>– Add / drop period with configurable deadlines</li> <li>– Late registration with advisor / department approval workflow</li> <li>– Overload registration request (approval required)</li> <li>– Module withdrawal with 'W' grade management</li> <li>– Repeat / resit module registration</li> <li>– Registration summary &amp; confirmation notification</li> <li>– Batch force-registration by admin for compulsory modules</li> </ul>	1
4	<b>Results &amp; Assessment Management</b>	<ul style="list-style-type: none"> <li>– Lecturer mark entry portal (coursework &amp; exam components)</li> <li>– Component-wise mark entry with configurable weightings</li> <li>– Grade computation engine (letter grade, GPA point, pass/fail) — GenBESTARI 10-point scale</li> <li>– Mark submission deadline management</li> <li>– Programme Leader review &amp; moderation workflow with audit log</li> <li>– Faculty broadsheet generation (per module, per programme)</li> <li>– Faculty Assistant Registrar broadsheet sign-off</li> <li>– Faculty Dean endorsement workflow</li> <li>– Exam Office approval post-senate meeting</li> <li>– Controlled results release to students</li> <li>– Incomplete (IC) grade management &amp; resolution</li> <li>– Grade appeal &amp; review request workflow (3 working day window from results publication)</li> <li>– cGPA &amp; semester GPA calculation engine with level weightage: Level 1000 = 30%, Level 2000 = 30%, Level 3000 = 40%</li> <li>– Transcript generation (official &amp; unofficial)</li> <li>– Dean's List computation per semester</li> <li>– Academic standing classification (Good Standing, Probation, Dismissal)</li> <li>– Supplementary assessment management: eligible for marks 30.0–39.9% OR valid absence (ABS grade)</li> <li>– Two-attempt maximum enforcement for Degree Core / Major Core (failure = programme termination)</li> <li>– Exam barring: automatic after 3 absences (consecutive or cumulative)</li> <li>– Discovery Year grade recording (excluded from cGPA)</li> </ul>	1
5	<b>Database Design &amp; Modelling</b>	<ul style="list-style-type: none"> <li>– Define all core entities and their relationships (ER diagram)</li> <li>– Design database schema per module (tables, columns, data types)</li> </ul>	1

		<ul style="list-style-type: none"> <li>– Establish primary key, foreign key, and index strategy</li> <li>– Define naming conventions: tables (snake_case), columns, constraints</li> <li>– Design audit trail tables (created_by, updated_by, timestamps)</li> <li>– Soft-delete convention (is_deleted flag vs hard delete)</li> <li>– Multi-tenancy / faculty isolation strategy</li> <li>– Database migration and versioning strategy (e.g. Flyway, Liquibase)</li> <li>– Data archival and retention policy per entity</li> <li>– Seed data and reference data tables (grading scales, room types, etc.)</li> </ul>	
6	UI/UX Design System	<ul style="list-style-type: none"> <li>– Define brand colour palette mapped to system roles and states</li> <li>– Typography: primary font, heading scale, body text, monospace</li> <li>– Iconography library selection and usage rules</li> <li>– Spacing, grid, and layout system (8px grid)</li> <li>– Component library design: buttons, inputs, dropdowns, modals, alerts, badges, tables, cards, calendars</li> <li>– State designs: empty, loading, error, success for all components</li> <li>– Responsive breakpoints (mobile, tablet, desktop)</li> <li>– Accessibility standards compliance: WCAG 2.1 AA minimum</li> <li>– Dark mode / light mode consideration</li> <li>– Design tokens (variables for colours, spacing, radii, shadows)</li> <li>– Prototype &amp; usability testing of core flows before development</li> </ul>	1
7	Client UAT, FAT & Go-Live	<ul style="list-style-type: none"> <li>– User Acceptance Testing (UAT) Support</li> <li>– Final Acceptance Testing (FAT) Support</li> <li>– User Documentations &amp; Technical Documentations</li> <li>– Go Live and Sign off from Client</li> </ul>	1

### 3. Technology Requirements

All development delivered under this quotation must strictly conform to the technology stack defined in this section. Vendors must confirm compliance with every item listed. Any proposed deviation must be explicitly declared in the submission and is subject to UBD approval before work commences. Non-declared deviations discovered during delivery or testing shall constitute a breach of contract.

#### 3.1 Backend Framework

Component	Mandated Technology	Requirements
MVC Framework	Laravel (PHP) — minimum v11.x	Eloquent ORM, service/repository pattern, Artisan command structure. Strict PHP typing enabled.
PHP Version	PHP 8.2 or above	Strict mode; type declarations on all methods; no deprecated functions.
Dependency Management	Composer	composer.json and composer.lock must be submitted with all deliverables.
Background Jobs	Laravel Queues — Redis driver preferred	Asynchronous processing for: notifications, broadsheet PDF generation, cGPA recalculation, mark computation jobs, and bulk registration operations.
Real-Time Events	Laravel Echo + WebSocket (Pusher or Soketi)	Mandatory for Module Registration real-time clash detection. Clash events must be broadcast to the client within 300ms of the student adding a module — this is a hard performance SLA.

<b>Task Scheduling</b>	Laravel Scheduler (cron-backed)	Automated: registration window open/close, mark submission deadline enforcement, grade appeal window closure (3 working days), exam barring threshold checks.
------------------------	---------------------------------	---

### 3.2 Database Layer

GenBESTARI employs a polyglot persistence strategy. The vendor must implement data access across all three database engines for the modules in this scope.

Engine	Version	Role	Usage in This Scope
<b>MySQL</b>	8.0+	Primary transactional store — all ACID-compliant academic data	Module offerings, timetable blocks, room assignments, student registrations, attendance records, mark records, grade results. InnoDB; UUID primary keys; UTF8MB4 charset.
<b>PostgreSQL</b>	15+	Read-heavy reporting, analytics, materialised views	Broadsheet generation queries, cGPA computation datasets with level weightage, academic standing classification, Dean's List computation, module pass/fail statistics. Window functions and partial indexes used where appropriate.
<b>MongoDB</b>	6.0+	Semi-structured event data, audit logs, activity streams	Student registration activity logs, timetable clash detection event streams, notification delivery records, audit change logs with nested old/new values, API rate-limit tracking. TTL indexes on transient collections.

Database access must use Laravel Eloquent (MySQL), the Laravel PostgreSQL driver (PostgreSQL), and jenssegers/mongodb or Laravel's native MongoDB driver (MongoDB). No hardcoded credentials — all connection strings via .env files.

### 3.3 Frontend & UI Layer

Component	Mandated Technology	Requirements
<b>Template Engine</b>	Laravel Blade	All server-rendered views in Blade. Blade components and slots for reusable elements. No mixing of template engines.
<b>CSS</b>	Custom CSS — CSS Custom Properties (BEM)	No third-party CSS frameworks (e.g. Bootstrap, Tailwind) unless explicitly approved in writing by UBD. All design tokens as CSS variables. BEM class naming throughout.
<b>JavaScript</b>	jQuery 3.7+ with Vanilla JS (ES6+)	jQuery mandatory for DOM manipulation, AJAX calls, and event handling. No SPA frameworks. Code must run natively in modern browsers without transpilation.
<b>Calendar Component</b>	FullCalendar.js v6+	Mandatory for Module Registration and Timetable weekly views. Must support real-time conflict visualisation (red clash highlight), iCal export, and mobile-responsive layout.
<b>PDF / Document Output</b>	Laravel DomPDF or Browsershot (Puppeteer-backed)	Required for M11 broadsheet generation, official and unofficial transcripts, and exam slips. Vendor must confirm library choice and provide a sample rendered output at demonstration.

### 3.4 API Layer

All modules must expose RESTful APIs for integration. The following standards are mandatory across all API endpoints.

Standard	Specification
<b>Architecture</b>	RESTful over HTTPS. Resource-based URL naming: plural nouns, lowercase-kebab-case. All endpoints versioned under /api/v1/.
<b>Documentation</b>	OpenAPI 3.0 (Swagger) mandatory. All endpoints documented with request/response schemas, error codes, and authentication requirements. Swagger UI hosted in staging environment.
<b>Authentication</b>	Laravel Sanctum — SPA token-based authentication. All endpoints protected. Token expiry and refresh strategy must be documented.
<b>Response Format</b>	Standard JSON envelope: { "status": "success error", "data": {}, "meta": { "pagination": {} }, "errors": [] }. HTTP status codes semantically correct (200, 201, 400, 401, 403, 404, 422, 500).
<b>Pagination</b>	Cursor-based pagination preferred. Laravel built-in paginator with resource transformations. Pagination metadata in meta block.
<b>Rate Limiting</b>	Laravel throttle middleware. Default: 60 requests/minute per authenticated user. Configurable per route group.
<b>Outbound Webhooks</b>	Results & Assessment Management must expose outbound webhooks for results publication events. Payload format, retry policy, and signature verification must be documented in the OpenAPI specification.
<b>API Gateway</b>	All endpoints must be registerable in a central API gateway (Kong or equivalent). Vendor must supply declarative gateway configuration files as part of the delivery package.

### 3.5 Security & Code Quality

Standard	Requirement
<b>Authorisation</b>	Role-Based Access Control (RBAC) via Laravel Gates and Policies. All routes explicitly guarded. Permission matrix documented for all system roles.
<b>Input Validation</b>	All API inputs validated via Laravel Form Requests. Server-side validation mandatory. SQL injection, XSS, and CSRF protection enforced through Laravel's built-in middleware stack.
<b>Audit Logging</b>	All create, update, and delete operations on academic records must be logged with: actor ID, timestamp, IP address, old values, new values. Stored in MongoDB audit collection; immutable once written.
<b>Encryption</b>	All data in transit via TLS 1.2 minimum (TLS 1.3 preferred). Sensitive fields (grades before publication, student identifiers) encrypted at rest using AES-256.
<b>Code Quality</b>	PHPStan level 8 and Laravel Pint (PSR-12 code style) must pass with zero errors on all submitted code. CI pipeline must include lint and static analysis gates.
<b>Test Coverage</b>	Minimum 80% unit test coverage (PHPUnit or Pest). Feature tests required for all critical workflows — especially clash detection, mark computation, and grade appeal. Coverage report submitted with each milestone.
<b>Secrets Management</b>	No credentials or API keys hardcoded in source. All secrets via .env files. Documented .env.example submitted with deliverables.

## 4. Testing, Documentation & Go-Live

The vendor is responsible for supporting the full testing lifecycle, documentation, and go-live process for all delivered modules.

Activity	Requirement
<b>User Acceptance Testing (UAT)</b>	Vendor must attend UAT sessions, respond to defect reports, and deliver fixes within agreed SLA. All Priority 1 (Critical) and Priority 2 (High) defects must be resolved before UAT sign-off.
<b>Final Acceptance Testing (FAT)</b>	A full regression test of all delivered modules. Vendor must resolve all outstanding Priority 1 and Priority 2 defects before FAT sign-off is granted.
<b>User Documentation</b>	End-user guides for each delivered module covering all key workflows in plain English — suitable for non-technical staff and students.
<b>Technical Documentation</b>	System architecture overview, API documentation (Swagger), database schema guide, deployment guide, .env.example, and developer onboarding guide. A developer unfamiliar with the codebase must be able to set up a local environment within 30 minutes.
<b>Go-Live</b>	Vendor must provide production deployment assistance and be available on go-live day for immediate support.
<b>Hypercare</b>	Minimum 2-week hypercare period post go-live. On-call response time no more than 4 business hours for Priority 1 issues.
<b>Client Sign-Off</b>	Formal written sign-off from the client (UBD) required before the final payment milestone is released.

## 5. Acceptance Criteria

No.	Deliverable	Acceptance Criterion
1	<b>Module Offering Management</b>	All key functionalities in \$2 operational in staging. Approval workflow end-to-end tested. Approved offering blocks correctly received by Registration and Timetable modules.
2	<b>Class &amp; Tutorial Management</b>	Timetable correctly assembled from offering blocks. Student individual view, staff view, and room occupancy view all functional. PDF and iCal exports working.
3	<b>Module Registration</b>	Timetable-first model confirmed: no module appears for registration without an approved timetable. Clash detection within 300ms confirmed by load test. Credit load rules enforced. Waitlist functional. Registration audit log complete.
4	<b>Results &amp; Assessment Management</b>	Six-stage approval chain end-to-end tested. cGPA formula with level weightage verified against manual calculations. Supplementary assessment rules correctly enforced. 3-working-day appeal window enforced by scheduler. Broadsheet and transcript PDF output confirmed.
5	<b>Database Design &amp; Modelling</b>	All migrations run cleanly on a fresh MySQL 8.0+ and PostgreSQL 15+ instance. MongoDB validators applied to all collections. ER diagram covers all entities in this scope. Data dictionary complete.
6	<b>UI/UX Design System</b>	All components functional at all three responsive breakpoints. WCAG 2.1 AA: zero Level A or AA failures in accessibility audit. Design token values in CSS match Figma definitions.
7	<b>UAT, FAT &amp; Go-Live</b>	All Priority 1 and Priority 2 defects resolved before FAT. System live on production. User and technical documentation delivered. Client sign-off obtained.
All	<b>Code Quality</b>	PHPStan level 8 passes with zero errors. Laravel Pint passes. Unit test coverage $\geq 80\%$ . OpenAPI spec delivered and Swagger UI live in staging. No undocumented endpoints.

## 6. Quotation Pricing

Vendors must provide a fixed-price quotation against each item below. Partial quotations (covering only some items) will not be considered. All prices in Brunei Dollars (B\$), inclusive of all applicable taxes and levies.

No.	Qty	Description	Unit (B\$)	Total (B\$)
1	1	Module Offering Management		
2	1	Class & Tutorial Management		
3	1	Module Registration		
4	1	Results & Assessment Management		
5	1	Database Design & Modelling		
6	1	UI/UX Design System		
7	1	Client UAT (User Acceptance Testing), FAT (Final Acceptance Testing) and Go-Live <ol style="list-style-type: none"><li>1. User Acceptance Testing Support</li><li>2. Final Acceptance Testing Support</li><li>3. User Documentations &amp; Technical Documentations</li><li>4. Go Live and Sign off from Client</li></ol>		
<b>TOTAL QUOTED AMOUNT (B\$)</b>				

End of Document — Open Quotation UBD-OQ-2026-002