

Application architecture

Repositories, services, entry points, and features for the applicant, student, lecturer, and staff portals on top of the existing campus CMS.

Working draft. Not a commercial proposal. Owners, kickoff date, and the old-CMS product name are still to be confirmed. Cyberjaya is the first acceptance campus.

Date	19 September 2026
Organisation	raisd-campus
Control plane	github.com/raisd-campus/control-plane
Design baseline	docs/sdd/ SDD-00 through SDD-12
Source of features	Daily Checklist, detailed tab, 112 rows
Hosting	k3s, Singapore. Stage 1 is one 4 GB VPS. Size steps up per stage.

1. What this system is

Campuses need one digital operating surface for applicants, students, lecturers, and staff, while student records stay in the existing CMS. This is not a replacement of every legacy function. New webs talk to a Portal API. The Portal API writes to the existing CMS and to a file store. A screen is not Live until the save is durable, permission is enforced on the server, and a staff or lecturer path exists — or the old CMS already does that job.

Product decisions already recorded on 7 September 2026:

1. Cyberjaya first.
2. Admissions before semester registration.
3. Core mobile at launch means a phone-friendly web, not a native app.
4. Keep the existing CMS as the initial backend.
5. Deliver by Applicant, Student, Lecturer, Admin / Staff, and Shared System.

The deepest UI today is a student demo (baseline 6811e88): sample data, no live CMS, no durable save. Applicant, lecturer, and staff portals are largely not started. Backend is Needs checking on almost every capability.

Progress at 19 September 2026

Counted from Faid Zamin's Daily Checklist, downloaded that day. The sheet is the backlog. Status cells match the 18 September copy. A row is complete only when it is Live. Live count is 0. Kickoff and every target week are TBC, so the charts below are sequence and volume, not a calendar.

MILESTONE	APPLICABLE	FRONTEND STARTED	NOT STARTED	LIVE
1. Confirm rules	0	0	0	0
2. Admissions launch	18	4	14	0
3. Core student portal	51	16	35	0
4. Cyberjaya acceptance	0	0	0	0
5. Campus expansion	34	12	22	0

Started means Demo, Partial, or Placeholder. The student UI is ahead of the sequence. Admissions is still the first delivery, and its applicant and staff rows are not started. Each mark is one applicable row. A mark is a start, not a completion.

Admissions #####..... 4 started, 14 not started, 0 Live
Core student #####..... 16 started, 35 not started, 0 Live
Expansion #####..... 12 started, 22 not started, 0 Live

Sequence, not a schedule. Each later milestone starts only after the one before it. Bar length is a placeholder because the sheet has no agreed duration.

ORDER	MILESTONE	TARGET ON THE SHEET
1	Confirm rules and integration	TBC
2	Admissions launch	TBC
3	Core student portal	TBC
4	Cyberjaya pilot acceptance	TBC

ORDER	MILESTONE	TARGET ON THE SHEET
5	Campus expansion	TBC

Admissions tasks still to do

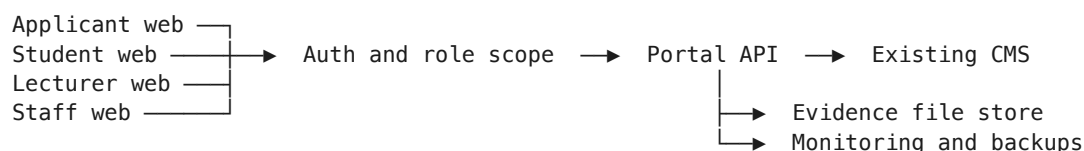
These close milestone 2. None is Live.

ID	TASK	NOW
CAP-02	Applicant application form	Not started
CAP-03	Applicant evidence upload	Not started
CAP-07	Track, offer, accept enrolment	Not started
CAP-01	Applicant sign-in. Student sign-in is only a placeholder	Not started / Placeholder
CAP-03 to 07	Staff review, qualifications, offer, first enrolment	Not started
CAP-53	Portal API to the existing CMS	Partial
CAP-37 to 39	Monitoring, backups, server-enforced access	Not started

After that: 35 core-student rows are not started, including every lecturer row. Expansion has 22 applicable rows not started. Owners are TBC on all 112 rows.

2. Architecture

Four doors, one campus. The webs and the Portal API run as Kubernetes deployments. Authentication and campus scope sit in front of the API. The API is the only component that writes to the CMS. The first install does not run all four doors: stage 1 is the applicant web and the API only. Student, lecturer, and staff deployments are added at the launch stage that needs them.



Modules inside the Portal API, not separate repositories: admissions, student records, teaching and files, finance, campus configuration.

Call rules

1. A portal web never opens Postgres, never stores uploads, and never calls the CMS.
2. The Portal API reports success only after the CMS acknowledges the write.
3. Authorisation is server-enforced. Hiding a menu is not security.
4. A staff token cannot be reused in the student shell. Each portal has its own origin.
5. Bank-transfer proof is not a payment until Bursary verifies and allocates it.
6. Staff screens are built only where the old CMS does not already run the workflow.

Data ownership

DOMAIN	SYSTEM OF RECORD	PORTAL RESPONSIBILITY
Applications, offers, enrolment	Existing CMS	Capture, display, handoff

DOMAIN	SYSTEM OF RECORD	PORTAL RESPONSIBILITY
Student master, documents	Existing CMS	Constrained edits; approval where required
Timetable, attendance, curriculum	Existing CMS	Student view; lecturer or staff write if the CMS supports it
Learning files	CMS or the file store	Authorised publish and download
Marks	Existing CMS	Display published results only until the staff workflow is Live
Fees, invoices, allocations	Existing CMS / finance ledger	Display; proof intake; never self-allocate
Announcements, policies	Owning department in the CMS	Display; acknowledgement if the campus requires it

3. Repositories

Five private repositories in the GitHub organisation `raisd-campus`. The existing CMS and mail have no repository here. The control plane is one repository. On the cluster it becomes three objects: the API deployment, a Postgres StatefulSet, and an evidence volume. Do not split the API into more repositories. Only `applicant-portal` and `control-plane` are deployed in the minimum stage.

REPOSITORY	WHAT IT OWNS	LOCAL PATH
<code>control-plane</code>	Campus configuration, identity and roles, Portal API, Postgres schema, evidence volume, this design set	<code>~/src/raisd/control-plane</code>
<code>applicant-portal</code>	Online registration web. Apply, evidence, track, accept enrolment	<code>~/src/raisd/applicant-portal</code>
<code>student-portal</code>	Enrolled-student web. Study, register, fees, results, support	<code>~/src/raisd/student-portal</code>
<code>lecturer-portal</code>	Teaching web. Publish, attendance, marking, class communication	<code>~/src/raisd/lecturer-portal</code>
<code>staff-portal</code>	Admin and staff web. New screens only for verified CMS gaps	<code>~/src/raisd/staff-portal</code>

Git remotes are `git@github.com:raisd-campus/<name>.git`. There is no application to install in any of them yet. The student demo recorded as `6811e88` lives in a separate codebase and is not these repositories.

4. How it is hosted

Workloads run on Kubernetes (k3s, Traefik included). The CMS and mail stay off the cluster. The cluster starts on the smallest server that can hold it, then grows one step at a time. Do not start on the 16 GB Premium machines.

Singapore only, same region as `cp8.sgp1.ultacp.com`. Prices are the 24-month catalogue rates on `ulthost.com` on 19 September 2026, not a quote. VPS Basic (1 GB) and VPS Business (2 GB) are below the floor: k3s, Postgres, and the API need about 4 GB before there is any headroom.

Server requirements

STAGE	NODES	REQUIREMENT	SMALLEST PLAN	CLUSTER TOTAL
1. Minimum	1	3 vCPU, 4 GB, 75 GB	VPS Professional, about \$15.87 / mo	3 vCPU, 4 GB
2. Admissions launch	1	4 vCPU, 6 GB, 100 GB	VPS Enterprise, about \$20.69 / mo. Resize the same node	4 vCPU, 6 GB
3. Core student	2	Edge: 3 vCPU, 4 GB, 75 GB. Data: 4 vCPU, 8 GB, 250 GB	Edge: a second VPS Professional. Data: VDS POWER Plus, about \$37.72 / mo. The VPS line ends at 6 GB	7 vCPU, 12 GB
4. Pilot acceptance	2	Same as stage 3	No new server	7 vCPU, 12 GB
5. Campus expansion	2	Edge unchanged. Data: 8 vCPU, 16 GB, 450 GB, only when the 8 GB node is full	VDS POWER Premium, about \$68.43 / mo, data node only	11 vCPU, 20 GB

The POWER Plus already owned (4 cores, 8 GB, 250 GB) is the stage 3 data node, not the stage 1 node. One Premium is the stage 5 data node or the CMS. The other Premium is the CMS and never joins the cluster. Running stage 1 on the Plus box is allowed, because it is larger than 4 GB. It is not the requirement.

Launch stages

Each stage adds workloads. Replicas stay at 1. Dates follow the milestones in SDD-03 and are still TBC.

STAGE	WHEN	CLUSTER	RUNNING	PUBLIC NAMES
1. Minimum	Before admissions	One node, VPS Professional	Ingress, applicant-portal, portal-api, Postgres, evidence volume	apply.raisd.co api.raisd.co
2. Admissions launch	Milestone 2	Same node, resized to VPS Enterprise	Stage 1 wired to the real CMS. Backup CronJob. staff-portal only if a CMS gap is verified	Same, plus staff only if deployed
3. Core student	Milestone 3	Add a 4 GB edge node. Data node becomes POWER Plus	student-portal and lecturer-portal. Ingress moves to the edge node	student.raisd.co teach.raisd.co
4. Pilot acceptance	Milestone 4	Same two nodes	Restore drill. No new cluster product	Unchanged
5. Campus expansion	Milestone 5	Grow the data node to 16 GB only when it is full	Remaining features. Not a second control plane	Unchanged

Stage 1 – one VPS, 3 vCPU, 4 GB

Browser

| HTTPS to that one node



Traefik ingress

apply.raisd.co → applicant-portal (1 replica)

api.raisd.co → portal-api (1 replica)

├─ Postgres StatefulSet

├─ evidence volume

└─ private call to the existing CMS

Not deployed: student-portal, lecturer-portal, staff-portal

Not used yet: POWER Plus, both POWER Premiums

Still to confirm: which Premium already runs the old CMS. That machine never joins k3s.

5. Entry points

Traefik, installed with k3s, routes by hostname. In stages 1 and 2, DNS points at the single data node. From stage 3, DNS points at the edge node and the data node no longer accepts public 443. Do not publish a name before its Deployment exists. The names come from the mailbox domain `raisd.co` and have not been created yet.

ENTRY POINT	WHO	REPOSITORY	FIRST SERVED
<code>https://apply.raisd.co</code>	Applicants	<code>applicant-portal</code>	Stage 1
<code>https://api.raisd.co</code>	Portal browsers, bearer token	<code>control-plane</code>	Stage 1
<code>https://staff.raisd.co</code>	Registry, Faculty, Bursary, QA, Marketing	<code>staff-portal</code>	Stage 2 only if a gap is verified, otherwise stage 3
<code>https://student.raisd.co</code>	Students	<code>student-portal</code>	Stage 3
<code>https://teach.raisd.co</code>	Lecturers	<code>lecturer-portal</code>	Stage 3
Existing CMS URL	Staff who already use it	None	Unchanged. Not an Ingress rule.
<code>https://cp8.sgp1.ultacp.com:2096/</code>	Mailbox holders	None	Webmail on UltraCP. Not in the cluster.

Every Service except the ingress is ClusterIP. Postgres has no host port. Do not set a session cookie on `Domain=.raisd.co`.

One admissions request

This is the stage 1 path. Later stages keep the same hostnames. Only the node that owns port 443 changes.

1. The browser loads `https://apply.raisd.co`. Traefik sends it to the `applicant-portal` Service.
2. The browser calls `https://api.raisd.co` with the applicant token. Traefik sends it to `portal-api`. The browser never sees a pod address.
3. `portal-api` checks role and campus, stores the file on the evidence volume, and writes the application to the CMS.
4. If the CMS does not acknowledge, the API returns a failed save. Postgres alone is not success.
5. Registry reviews in the existing CMS. `staff.raisd.co` is not part of stage 1.

6. Functionalities

Delivery order is fixed. A milestone is when a capability must work, not when investigation starts. Dates are TBC.

ORDER	MILESTONE	WHAT MUST WORK
1	Confirm rules and integration	Campus rules, named owners, legacy capabilities, integration access. Not a UI milestone.
2	Admissions launch	Applicant account, application, evidence, track, offer, accept, Registry review, first Portal API slice, auth, backups.
3	Core student portal	Learning, timetable, assignments, records, semester registration, fees, with a lecturer or staff path.
4	Cyberjaya pilot acceptance	Integrated pilot, staff training, campus sign-off.
5	Campus expansion	Remaining CMS, quizzes, live class, library, visa, housing, scholarships, online payment, other campuses.

A capability is Live only when all of these are true: real CMS data, durable save, server-enforced permission, a paired staff or lecturer path or a verified old-CMS equivalent, a named owner, and core mobile accepted where the capability is in the mobile set.

Applicant portal

Entry point `apply.raisd.co`. Repository `applicant-portal`. Milestone: admissions launch. Registry owns the decision. Marketing may publish announcements and does not issue offers.

Primary path: sign in or create an account, complete the application, upload evidence and English-entry documents, accept policies, track status, view an offer, accept enrolment. The API provisions the applicant in the CMS, persists the application, and writes the first enrolment only after acceptance.

8 capabilities. Frontend status is the 18 September 2026 checklist snapshot. Backend is Needs checking on almost every row. The same CAP ID on two portals is one capability, two role surfaces.

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-02	Registry	Online application form for new applicants	Admissions launch	Not started
CAP-03	Registry	Upload application documents and English-entry evidence	Admissions launch	Not started
CAP-07	Registry	Track application, view offer and accept enrolment	Admissions launch	Not started
CAP-55	Registry	Read applicant policies and declarations	Admissions launch	Not started
CAP-16	Marketing	Read applicant announcements and admissions updates	Admissions launch	Not started
CAP-01	TBC	Applicant sign-in and account access	Admissions launch	Not started
CAP-35	TBC	Accessible application forms and assistance	Admissions launch	Not started

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-36	TBC	Mobile online-registration shell	Admissions launch	Not applicable

Student portal

Entry point `student.raisd.co`. Repository `student-portal`. Sign-in and the phone shell are admissions (milestone 2). Learning, records, and fees are the core student portal (milestone 3). Quizzes, live class, library, visa, housing, and scholarships are campus expansion.

The student selects modules. The API validates rules as one transaction and the CMS returns the invoice. The student may upload bank-transfer proof. Bursary verifies and allocates. The proof itself never changes the ledger. Assignments are uploaded as bytes the lecturer can open, not as metadata alone.

Campus-requested student features: timetable, announcements, study guides, notes, briefs, assignment submit, revision, past papers, policies.

41 capabilities. Frontend status is the 18 September 2026 checklist snapshot. Backend is Needs checking on almost every row. The same CAP ID on two portals is one capability, two role surfaces.

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-08	Registry	Personal details and emergency contacts	Core student portal	Demo
CAP-09	Registry	Student documents library	Core student portal	Partial
CAP-10	Registry	Semester subject and class registration	Core student portal	Demo
CAP-13	Registry	Grades, results and student progress	Core student portal	Demo
CAP-15	Registry	Graduation clearance and transcripts	Campus expansion & remaining CMS	Placeholder
CAP-16	Registry	Important announcements	Core student portal	Partial
CAP-55	Registry	University policies for students	Core student portal	Partial
CAP-45	Bursary	Finance balances, invoices and payment history	Core student portal	Demo
CAP-46	Bursary	Bank-transfer proof submission	Core student portal	Demo
CAP-47	Bursary	Online payment and financial documents	Campus expansion & remaining CMS	Placeholder
CAP-48	Bursary	Scholarships and incentives	Campus expansion & remaining CMS	Demo
CAP-30	Quality Assurance	Student feedback and course evaluation	Campus expansion & remaining CMS	Placeholder
CAP-11	Faculty	Class timetable	Core student portal	Demo
CAP-12	Faculty	Attendance records	Core student portal	Demo
CAP-14	Faculty	Study plan and prerequisites	Core student portal	Demo
CAP-17	Faculty	Study guides	Core student portal	Not started
CAP-18	Faculty	Lecture notes and learning-media access	Core student portal	Placeholder
CAP-19	Faculty	Assignment briefs	Core student portal	Partial

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-20	Faculty	Submit, replace and withdraw assignments	Core student portal	Demo
CAP-21	Faculty	Revision work and practice activities	Core student portal	Not started
CAP-22	Faculty	Past examination papers	Core student portal	Not started
CAP-23	Faculty	Course information and learning outcomes	Core student portal	Partial
CAP-24	Faculty	Online quizzes and examinations	Campus expansion & remaining CMS	Placeholder
CAP-25	Faculty	Student-lecturer and student-student communication	Campus expansion & remaining CMS	Demo
CAP-26	Faculty	Live online classes	Campus expansion & remaining CMS	Not started
CAP-27	Faculty	Self-paced learning and release schedules	Campus expansion & remaining CMS	Partial
CAP-28	Faculty	View marks and lecturer feedback	Core student portal	Not started
CAP-29	Faculty	Assessment identity checks and integrity guidance	Core student portal	Not started
CAP-01	TBC	Sign-in, logout and student account access	Admissions launch	Placeholder
CAP-31	TBC	Digital library and e-resources	Campus expansion & remaining CMS	Not started
CAP-32	TBC	Student orientation and digital-learning skills	Core student portal	Not started
CAP-33	TBC	Learner support and IT helpdesk	Core student portal	Partial
CAP-34	TBC	Study-centre and regional support	Campus expansion & remaining CMS	Not started
CAP-35	TBC	Accessibility and special-needs support	Admissions launch	Demo
CAP-36	TBC	Mobile portal shell and device access	Admissions launch	Not applicable
CAP-39	TBC	Privacy, permissions and cybersecurity	Admissions launch	Partial
CAP-49	TBC	Resume and portfolio builders	Campus expansion & remaining CMS	Demo
CAP-50	TBC	Immigration and visa processing	Campus expansion & remaining CMS	Placeholder
CAP-51	TBC	Student requests, complaints and academic appeals	Campus expansion & remaining CMS	Placeholder
CAP-52	TBC	Accommodation management	Campus expansion & remaining CMS	Placeholder
CAP-54	TBC	Campus services information	Campus expansion & remaining CMS	Demo

Lecturer portal

Entry point `teach.raisd.co`. Repository `lecturer-portal`. There is no lecturer application in the current baseline. Student Materials in the demo is not an LMS.

Core teaching, required before student learning can be called Live: workspace, timetable and class list, record and correct attendance, publish guides, notes and briefs, receive submissions, mark and publish feedback, identity and integrity checks, submit materials for QA. Later: quizzes and exams, live class, class communication, self-paced release, class analytics, a mobile workspace.

19 capabilities. Frontend status is the 18 September 2026 checklist snapshot. Backend is Needs checking on almost every row. The same CAP ID on two portals is one capability, two role surfaces.

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-11	Faculty	View teaching timetable and class allocations	Core student portal	Not started
CAP-12	Faculty	Record and correct class attendance	Core student portal	Not started
CAP-17	Faculty	Publish study guides	Core student portal	Not started
CAP-18	Faculty	Publish lecture notes and learning media	Core student portal	Not started
CAP-19	Faculty	Publish assignment briefs and deadlines	Core student portal	Not started
CAP-20	Faculty	Receive and review submitted assignments	Core student portal	Not started
CAP-21	Faculty	Publish revision work and practice activities	Core student portal	Not started
CAP-22	Faculty	Prepare past papers for authorised publication	Core student portal	Not started
CAP-23	Faculty	Maintain course information and learning outcomes	Core student portal	Not started
CAP-24	Faculty	Prepare and conduct online quizzes and examinations	Campus expansion & remaining CMS	Not started
CAP-25	Faculty	Communicate with students and teaching groups	Campus expansion & remaining CMS	Not started
CAP-26	Faculty	Deliver live online classes	Campus expansion & remaining CMS	Not started
CAP-27	Faculty	Schedule self-paced learning and content release	Campus expansion & remaining CMS	Not started
CAP-28	Faculty	Mark assessments and publish lecturer feedback	Core student portal	Not started
CAP-29	Faculty	Check assessment identity and academic integrity	Core student portal	Not started
CAP-40	Faculty	Review class learning analytics	Campus expansion & remaining CMS	Not started
CAP-43	Faculty	Submit course materials for quality review	Core student portal	Not started
CAP-44	Faculty	Lecturer workspace and course access	Core student portal	Not started

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-36	TBC	Mobile lecturer workspace	Campus expansion & remaining CMS	Not applicable

Staff portal

Entry point `staff.raisd.co`. Repository `staff-portal`. Build a new screen only for a verified gap. If the old CMS already runs the workflow, deep-link or omit the screen. A new administrator chrome is not required for admissions if Registry can already issue offers in the old CMS.

- Registry owns applications, offers, student master, registration, result release, policies. Foreign awards are never auto-equivalent.
- Faculty staff own the timetable grid and programme rules. Day-to-day teaching is the lecturer portal.
- Bursary owns the student ledger and proof verification. Finance owns reconciliation and official documents.
- Quality Assurance records programme and site approvals before applications are taken, and later approves materials.
- Marketing publishes recruitment copy only.

38 capabilities. Frontend status is the 18 September 2026 checklist snapshot. Backend is Needs checking on almost every row. The same CAP ID on two portals is one capability, two role surfaces.

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-03	Registry	Review application documents and entry evidence	Admissions launch	Not started
CAP-05	Registry	Verify qualification levels, credits and duration	Admissions launch	Not started
CAP-06	Registry	Review foreign equivalency and professional recognition	Admissions launch	Not started
CAP-07	Registry	Review applications, issue offers and create first enrolment	Admissions launch	Not started
CAP-08	Registry	Maintain student records and emergency contacts	Core student portal	Not started
CAP-09	Registry	Publish and manage student documents	Core student portal	Not started
CAP-10	Registry	Configure and approve semester registration	Core student portal	Not started
CAP-13	Registry	Approve and release results and progress records	Core student portal	Not started
CAP-15	Registry	Approve graduation and issue official transcripts	Campus expansion & remaining CMS	Not started
CAP-16	Registry	Publish approved student announcements	Core student portal	Not started
CAP-55	Registry	Approve and publish student policies	Core student portal	Not started
CAP-45	Bursary	Manage student invoices, balances and payment records	Core student portal	Not started
CAP-46	Bursary	Verify bank-transfer proofs and allocate payments	Core student portal	Not started

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-48	Bursary	Assign and review scholarships and incentives	Campus expansion & remaining CMS	Not started
CAP-47	Finance	Reconcile online payments and issue financial documents	Campus expansion & remaining CMS	Not started
CAP-04	Quality Assurance	Record institution, programme and delivery-site approvals	Admissions launch	Not started
CAP-30	Quality Assurance	Manage student feedback and course evaluations	Campus expansion & remaining CMS	Not started
CAP-40	Quality Assurance	Review institutional learning analytics and quality reports	Campus expansion & remaining CMS	Not started
CAP-42	Quality Assurance	Record copyright permissions and provider approvals	Core student portal	Not started
CAP-43	Quality Assurance	Approve course materials and LMS-readiness evidence	Core student portal	Not started
CAP-11	Faculty	Schedule classes and allocate lecturers and rooms	Core student portal	Not started
CAP-14	Faculty	Maintain study plans, prerequisites and programme rules	Core student portal	Not started
CAP-22	Faculty	Approve access and publication rights for past papers	Core student portal	Not started
CAP-23	Faculty	Approve course information and learning outcomes	Core student portal	Not started
CAP-24	Faculty	Administer examinations, access and approved arrangements	Campus expansion & remaining CMS	Not started
CAP-16	Marketing	Publish recruitment and applicant communications	Admissions launch	Not started
CAP-31	TBC	Administer digital-library access and licences	Campus expansion & remaining CMS	Not started
CAP-32	TBC	Publish orientation and digital-learning guidance	Core student portal	Not started
CAP-33	TBC	Manage learner-support and IT-helpdesk requests	Core student portal	Not started
CAP-34	TBC	Coordinate study-centre and regional support	Campus expansion & remaining CMS	Not started
CAP-35	TBC	Manage accessibility and special-needs arrangements	Campus expansion & remaining CMS	Not started
CAP-36	TBC	Mobile staff and administrator workspace	Campus expansion & remaining CMS	Not applicable

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-44	TBC	Staff and administrator workspace and role access	Admissions launch	Not started
CAP-49	TBC	Manage career, resume and portfolio guidance	Campus expansion & remaining CMS	Not started
CAP-50	TBC	Process immigration and student-visa cases	Campus expansion & remaining CMS	Not started
CAP-51	TBC	Handle student requests, complaints and academic appeals	Campus expansion & remaining CMS	Not started
CAP-52	TBC	Manage accommodation allocation and records	Campus expansion & remaining CMS	Not started
CAP-54	TBC	Publish campus services information	Campus expansion & remaining CMS	Not started

Shared system

Not a fifth website. These capabilities gate Live. They run with the control plane on Premium 1, except training, which is an operating task before Cyberjaya acceptance.

6 capabilities. Frontend status is the 18 September 2026 checklist snapshot. Backend is Needs checking on almost every row. The same CAP ID on two portals is one capability, two role surfaces.

ID	DEPARTMENT	FEATURE	MILESTONE	FRONTEND
CAP-42	Quality Assurance	Copyright and external-provider operating controls	Confirm rules & integration	Not applicable
CAP-37	TBC	Connectivity, capacity and service monitoring	Admissions launch	Not applicable
CAP-38	TBC	Backups and continuity	Admissions launch	Not applicable
CAP-39	TBC	Privacy, permissions and cybersecurity	Admissions launch	Not applicable
CAP-41	TBC	Staff training and operating readiness	Cyberjaya pilot acceptance	Not applicable
CAP-53	TBC	Existing CMS integration and campus configuration	Admissions launch	Partial

7. Roles

ACTOR	TYPICAL DESK	NEEDS
Applicant	Registry / Marketing	Account, application, evidence, offer
Student	Registry, Faculty, Bursary, support	Records, learning, fees, services
Lecturer	Faculty	Class list, content, marking
Registry officer	Registry	Applications, student master, enrolment, results release

ACTOR	TYPICAL DESK	NEEDS
Faculty scheduler	Faculty	Classes, rooms, programme rules
Bursary / Finance	Bursary, Finance	Invoices, proof verification, reconciliation
QA	Quality Assurance	Approvals, materials readiness, evaluations
Marketing	Marketing	Applicant communications only

Minimum roles on the token: Applicant, Student, Lecturer, Registry, Faculty, Bursary, Finance, QA, Marketing, Admin. Campus and programme scope ride on the token.

8. What this document does not decide

- The old-CMS product name, and which Premium already hosts it.
- Anything beyond Ubuntu, k3s, Traefik, Postgres 16, and one replica per deployment. No service mesh and no second control plane in the minimum install.
- Named owners, kickoff date, and milestone dates.
- A native mobile app, a new identity system if CMS login already works, or a finance-engine replacement.
- Buying another server for admissions.

Compiled 19 September 2026 from docs/sdd/ in raised-campus/control-plane, including SDD-01, SDD-02, SDD-03, SDD-04, SDD-05, SDD-06, SDD-07, SDD-08, SDD-11, and SDD-12. Feature rows are the detailed Daily Checklist. This PDF is a working draft for the working group, not an instruction to replace a CMS function that already works.