International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Student University Residence Life Protocols: Design, Architecture and Implementation (Case of IUGB)

Dr. Bayomock Linwa André Claude[1], Mr. MEA Assoumou Michael[2]
Department Of Computer Science,
International University Of Grand-Bassam

ABSTRACT- University residences concentrate students in shared living environments where safety, accountability and access to information depend on reliable procedures for movement control, room accountability and incident handling. In Côte d'Ivoire, demand for university housing exceeds supply, and publicly documented residence procedures combine early online services, mainly for admission, with physical formalities on site [1], [2], [3]. This paper presents the design, architecture and implementation of IUGB In/Out, a multi-platform residence management system developed for the International University of Grand-Bassam (IUGB). The contribution is a set of six formally described residence life protocols (exit and return, overdue management, room inspection, incident handling, behavior evaluation, and identity and governance), together with the data model, the three-tier architecture and the implementation that enforce them. The protocols rely on role-based access control for students, supervisors and administrators, on cryptographically signed single-use QR tokens, on a persistent audit log, on real-time Socket.IO events, on Firebase push notifications, and on an AI assistant grounded in an administrator-managed knowledge base. The system is implemented with Flutter, React, Node.js/Express, Sequelize and MySQL. We position it against the publicly documented practices of Ivorian university residence authorities (CROU) and report the functional results obtained. No field-scale quantitative evaluation has been carried out yet; operational benefits are therefore reported as expected benefits, and limitations and future work are discussed.

Keywords – University Residence, Student Life, Protocols, QR Code, Role-Based Access Control, Traceability, Real-Time Systems, Côte d'Ivoire, IUGB

I. INTRODUCTION

Student housing is more than a supply of beds. A university residence is a regulated community in which hundreds of young adults live under shared rules, and the institution that operates it is expected to know, at any moment, who is inside, who is away with permission, who is late, which rooms are in which condition, and which incidents are open. These expectations are protocols in the strict sense: sequences of steps, performed by identified actors, that must produce the same outcome and leave the same trace every time. When such protocols are carried on paper, verbal agreement or informal messaging, they are difficult to enforce, to audit and to summarize for decision makers.

The Ivorian context makes the question pressing. Public sources describe a university housing sector in which the number of beds remains small relative to the student population, with 24,574 beds reported for more than 363,000 students in May 2026, a coverage that the government itself puts at about 6.5% and intends to raise to 15% by 2030 [1], [4]. Where beds are scarce, rules on who may occupy a room, who may enter, and who has left become safety and fairness issues as much as administrative ones; the government has publicly stated that nearly 35% of beds had been occupied illegally before control operations were conducted [5]. Residence authorities have started to digitize their front office: the CROU Abidjan 2 and CROU Bouaké, for instance, run online admission platforms with a personal housing space for each applicant [2], [3]. The documented procedures nevertheless remain hybrid, mixing online registration with physical dossier deposit [3], and we found no public description of a digital protocol for the daily life inside the residence (exits, returns, room inspections and incident handling).

This paper addresses that gap for a concrete institution, the International University of Grand-Bassam (IUGB), which is listed among the universities of the Ivorian higher-education sector [6]. According to the requirements analysis conducted for this work, residence operations at IUGB relied on manual or fragmented processes: paper passes for outings, no centralized view of who is off campus, late detection of overdue returns, and manual handling of room condition reports, maintenance requests and conduct evaluations [7]. We designed and implemented IUGB In/Out, a digital platform that replaces these practices by a small set of explicit protocols executed by three human roles (students, supervisors and administrators) and by the application itself.

The contributions of this paper are fourfold. First, we formalize six residence life protocols with their actors, steps, state transitions and invariants. Second, we present the data model that makes these protocols traceable. Third, we describe a three-tier architecture that combines mobile and web clients, a REST backend, real-time channels, push notifications and an AI assistant. Fourth, we situate the

© 2026 IJSRET

proposal against documented Ivorian residence practices and state clearly what has, and what has not, been evaluated.
The remainder of the paper is organized as follows. Section 2 states the motivations. Section 3 reviews related work and the Ivorian residence context. Section 4 describes the problem and its requirements, and Section 5 outlines the solution approach. Section 6 presents the protocols, Section 7 the data models and use cases, Section 8 the architecture and Section 9 the implementation. Section 10 reports the results and limitations, and Section 11 concludes.

1. Motivations

Three groups of motivations drive this work: safety and accountability, managerial visibility, and student service.

2. Safety And Accountability

Movement control is time-sensitive. A student who leaves for a weekend and does not return when expected is a situation that supervisors must learn about quickly, not at the next manual register check. Paper passes can be lost, altered or forged, and a register is hard to query when an incident requires reconstructing who was where. The capstone analysis underlying this paper identified delayed awareness of overdue exits and inconsistent reporting of incidents and room conditions as concrete field challenges [7]. A digital protocol with signed, short-lived credentials and an immutable log addresses both the prevention (the credential cannot be forged or reused) and the post-event audit (every transition is recorded).

3. Managerial Visibility

Housing management needs to answer simple questions reliably: how many students are currently out, which residences generate the most incidents, how do absences evolve over a semester. These questions cannot be answered from scattered paper or chat messages. Public figures on Ivorian residences also show that capacity pressure is structural [1], [8], which increases the importance of knowing precisely how existing beds and rooms are used. A platform that persists every movement, report and decision in a relational database, and that produces periodic snapshots, turns operational activity into planning data.

4. Student Service And Fairness

Students need to request an exit, follow its status, sign a room inventory, report a broken facility and understand the residence rules without queuing at an office. Fairness also matters: a room report signed by both the supervisor and the student protects the student from unjustified damage claims and the institution from unreported damage. An assistant that answers questions from the official residence rules reduces repeated inquiries to staff. These motivations also fit the national direction of travel, in which residence authorities are moving some services online and the ministry is investing in residence rehabilitation and construction [4], [9].

5. Research Status

We organize the state of the art in four parts: digital hostel management systems, protection of credentials and real-time communication, AI assistance in education, and the Ivorian residence context. The section closes with a comparison between documented Ivorian practices and the system proposed here.

6. Digital Hostel And Residence Management Systems

Digital hostel management is an established application domain, mostly documented for Asian and commercial settings. Heda et al. present DHMS, a MERN-stack platform that automates room allotment, grievance redressal, gate pass handling and a chatbot, with role-based access control and Socket.IO for real-time updates; the reported results (92% satisfaction in room allocation, chatbot response time under one second) come from simulation tests, not from a deployment [10]. Commercial products follow the same pattern: a cloud hostel management offering advertises attendance and leave management with gate pass workflows and parent-based leave approval [11]. These works confirm that gate passes, grievance handling and room management are recurring modules, and they motivate the integration of these functions in a single platform. They differ from the present work in three respects: they do not describe a cryptographically verifiable credential for physical entry and exit validation, they are not situated in an Ivorian institutional context, and the published evidence is simulation-based or promotional.

7. Credentials, Authentication And Real-Time Communication

Three standard building blocks support the credential and communication layers of residence protocols. Keyed-hash message authentication (HMAC) provides integrity and authenticity of a short message using a shared secret [12]; it is well suited to a server that both issues and validates QR payloads. JSON Web Tokens give a compact format for access and refresh tokens that carry claims between a client and an API [13], and adaptive password hashing such as bcrypt protects stored credentials against offline guessing [14]. For live dashboards, the WebSocket protocol offers full-duplex communication over a single connection [15]. The design question is not the availability of these primitives but their composition into protocols with clear invariants (single use, expiry, residence scope, revocation), which is the focus of Section 6.

8. AI Assistance In Education

Chatbots are increasingly used in education to answer repetitive questions and to provide guidance. A systematic review of chatbot applications in education highlights their potential for immediate student support and also notes the need to ground answers in reliable content [16]. Retrieval-

© 2026 IJSRET

augmented generation combines a language model with an external knowledge source so that generated answers can rely on curated, updatable documents rather than on model parameters alone [17]. In a residence, the relevant documents are rules, schedules and memoranda that change over time; the assistant must therefore consult a knowledge base that administrators can update, which is the approach taken in Section 8.5.

II. RESIDENCE MANAGEMENT IN CÔTE D'IVOIRE

Public university residences in Côte d'Ivoire are operated by the Regional Centers of University Services (CROU), public establishments under the supervision of the Ministry of Higher Education and Scientific Research (MESRS) [9], [6]. The Abidjan 2 center, for example, derives its attributions and organization from decree 2013-438 of 13 June 2013, and a dedicated sub-directorate for reception and housing manages admissions, residences, restaurants and security, with separate housing, reception, and security-and-risk-prevention services [18], [19]. The CROU Abidjan 1 organizes housing in a similar way, with a housing service that receives the dossiers of applicants and a reception and accommodation service that supervises residence management [20].

- **Access to housing.** Admission is handled through yearly admission and readmission operations. At CROU Abidjan 2, the 2025–2026 online registration window ran from 9 January to 8 February 2026; applicants register on a platform that issues a login and gives access to a personal housing space where they follow their request, consult the result, download forms and upload supporting documents such as a legalized commitment form and the identity document of a parent or guardian [19], [2]. At CROU Bouaké, applicants register online, then book an appointment to deposit the physical dossier in a residence [3]. Housing access is therefore partly digital, but with physical formalities.
- **Entry and exit control.** The documented control practices are rule-based and human-operated. The CROU Abidjan 2 residence regulation requires a resident who enters the residence after midnight to submit to checks by the security service; it also requires an inventory of furniture in the resident's presence when a room is allocated, an inventory at the end of the academic year before the keys are returned, and retention of the deposit in case of loss, damage or rent arrears [18]. A press report on the CROU Abidjan rules quotes its director announcing that concierges can enter rooms at any time for unannounced checks [21]. We did not find, in public sources, a description of a systematic electronic record of each exit and return for the CROU residences, nor comparable information for non-CROU Ivorian residences; we therefore make no claim about them.
- **Capacity and challenges.** The 2012 review of the higher-education sector by MESRS listed insufficient and dilapidated residences and overcrowding among the sector's problems, and mentioned a capacity of 9,907 beds [22]. In May 2026 the minister reported 24,574 beds for more than 363,000 students and a coverage of 6.5% [1]. Another official source reports a capacity of 16,033 beds in 2023 for 31 residences and restaurants across eight cities [8]; figures differ across sources, probably because of scope and reporting dates, and we therefore treat them as orders of magnitude. The government has also reported that nearly 35% of beds were occupied illegally, by students who did not meet the conditions or by outsiders, and that control teams freed rooms and dismantled illegal structures [5]. These elements show that identity, eligibility and occupancy control are central operational problems.
- **Digital services.** The CROU Abidjan 2 website exposes an online housing space (admission request, file tracking and online payment) and, in its navigation, services to report an incident and to follow a report [9]. These services show that online incident reporting is already part of the front-office offer for residents. They concern the center's services in general and do not describe per-residence supervision workflows.
- **Personal data.** Any system that records the movements and conduct of students processes personal data. Ivorian law n° 2013-450 of 19 June 2013 on the protection of personal data entrusts the supervision to the ARTCI, and the regulator has stated that processing must be declared or authorized beforehand, including surveillance and geolocation systems [23], [24]. This is a deployment requirement for any institution adopting such a platform and is discussed in Section 10.

1. Comparison With Documented Practices

Table 1 compares the documented practices of the CROU residences with the approach of IUGB In/Out. The left column only contains information supported by the cited sources; where public sources are silent, the cell says so.

The comparison shows both similarities and differences. The similarities concern the objectives (controlled access, accountable use of rooms, handling of incidents) and the institutional logic (a resident community under rules and under the authority of staff). The differences concern the mechanism: IUGB In/Out records every movement as an authorized, signed and traceable event, whereas the documented practices rely on rules and human controls. The system is also deliberately narrower in one respect: it does not handle admission, payment or eligibility, which the CROU platforms do. It should thus be seen as a complement to a housing admission process rather than a substitute.

© 2026 IJSRET

Table 1: Documented Ivorian CROU practices compared with IUGB In/Out

Dimension	Documented CROU practice	IUGB In/Out approach
Organization	Public establishments under MESRS; housing, reception and security services within a sub-directorate [19], [20]	One university, three roles (student, supervisor, administrator); supervisors bound to residences
Access to housing	Annual online admission and readmission with a personal housing space, then physical deposit of documents [2], [3]	Not an admission platform: student accounts are pre-approved from an imported roster; rooms are assigned by administrators
Entry and exit control	After-midnight check by security staff [18]; unannounced room checks by concierges [21]; no public description of an electronic exit and return record	Each exit is a request approved by a supervisor, validated at the gate with a signed single-use QR token, and recorded with planned and actual times
Late returns	Not documented publicly	Automatic detection every 10 minutes, one deduplicated alert, push reminder and live dashboard signal
Room inventory	Inventory and état des lieux in the resident's presence at allocation and at year end [18]	Digital move-in and move-out reports with item checklist, student signature or rejection, and link between move-out and move-in
Incident reporting	Online incident reporting and tracking offered by CROU Abidjan 2 at center level [9]	Residence-scoped incident reports with photo or video, routed to the supervisors of the student's residence, with status feedback
Management data	Not documented publicly	Persistent logs, live counters and Friday statistics snapshots
Student information	Regulation published on the center's website [18]	AI assistant answering from an administrator-managed knowledge base

2. Positioning

From the review, three gaps emerge. First, published hostel systems rarely specify how physical entry and exit are cryptographically validated. Second, the Ivorian documented digital offer covers admission and front-office services but, as far as public sources show, not the daily protocols inside the residence. Third, evidence in the literature is mostly simulation or vendor material. IUGB In/Out targets the first two gaps with an implemented

system and is explicit about the third: its evidence is a working implementation, not yet a field evaluation.

III. PROBLEM DESCRIPTION

1. Operational Context And Stakeholders

A residence involves three groups of direct users. Students request exits, present QR credentials, report incidents, sign room reports and ask questions. Supervisors oversee one or more assigned residences: they validate movement requests, scan credentials at the gate, process incidents and room reports, record evaluations and receive overdue alerts. Administrators govern the platform: they control which accounts are approved, configure semesters and residence structures, observe aggregate statistics and manage the knowledge that feeds the AI assistant [7]. Housing management, security teams, academic administration and IT support are indirect stakeholders that depend on the data the platform produces.

2. Problem Statement

The requirements analysis identified five related problems.

- Operational fragmentation. Related tasks performed by students, supervisors and administrators are carried on disconnected channels, which creates discrepancies and loss of continuity.
- Low traceability. When movements and incidents are not logged centrally, post-event audits are difficult and accountability weakens. Physical passes in particular can be lost, altered or forged.
- Delayed detection of critical states. Without automated checks, supervisors discover late returns slowly, and there is no instant answer to how many students are off campus.
- Incomplete information for management. Without structured persistence and periodic snapshots, leadership cannot reliably answer basic operational questions.
- Limited student support. Static documents are insufficient to answer recurrent questions on rules, curfews and procedures.

3. Requirements

© 2026 IJSRET

The functional and non-functional requirements derived from these problems are listed in Table 2.

Table 2: Functional (FR) and non-functional (NFR) requirements

ID	Requirement
FR1	User authentication with a secure token lifecycle
FR2	Role-specific access and navigation for administrators, supervisors and students
FR3	Exit (sortie) request, approval and refusal workflows
FR4	QR-based control of outbound and return events
FR5	Detection and handling of overdue returns
FR6	Incident submission and its treatment by supervisors or administrators
FR7	Room report creation, signing, rejection and closure
FR8	Semester and residence management for administrators
FR9	Administrative statistics and weekend snapshots
FR10	AI assistant with dynamic knowledge-base content control
NFR1	Security hardening and controlled access
NFR2	Data integrity and consistency across workflows
NFR3	Reasonable response performance under normal load
NFR4	Maintainable, modular codebase
NFR5	Usability on desktop web and mobile form factors
NFR6	Observability through logs and health checks

4. Scope

The current version covers authentication and account-state control, residence structure management, the sortie lifecycle, QR token generation and validation, incident reporting, room reports, administrative statistics with weekend snapshots, and the AI assistant. It deliberately excludes biometric gate hardware, multi-campus federation, an offline-first synchronization engine and predictive risk scoring based on machine learning [7]. Admission to housing and payments are also outside the scope.

5. Solution Approach

5.1 Design Principles

Five principles guide the solution. (i) *Protocol first*: each residence process is modeled as an explicit workflow with identified actors, valid states and transitions, so that behavior is the same for every user. (ii) *Least privilege*: every request passes through authentication, account-status, role and residence-scope checks before any business logic. (iii) *Verifiable credentials*: physical validation relies on short-lived, signed, single-use tokens rather than on static passes. (iv) *Everything leaves a trace*: significant transitions are persisted and key events are written to an audit log. (v) *Immediate visibility*: state changes are pushed to the relevant users in real time and by notification, so that nobody needs to poll or to ask.

6. Solution Overview

IUGB In/Out is a digital, centralized platform made of a Flutter mobile application for students, a React web portal for supervisors and administrators, and a Node.js backend that exposes a REST API, a Socket.IO gateway and background schedulers over a MySQL database. External services provide federated identity (Google), push delivery (Firebase Cloud Messaging) and language-model inference (Google Gemini). The platform shifts residence operations from a reactive, manual model to an automated and data-driven one [7].

7. From Problems To Components

Table 3 maps each problem to the components that answer it and to the protocol that realizes it (protocols are detailed in Section 6).

8. Student University Residence Life Protocols Description And Design

This section presents the central contribution. We first identify the actors, then give an overview of the six protocols (P1 to P6), then describe each one with its steps, its state machine and its invariants.

Table 3: Mapping of problems to solution components

Problem	Solution component	Protocol
Fragmentation	Role-specific clients on a single backend and database	All
Low traceability	Status machines, timestamps, persistent audit log (LogAction)	P1, P2, P6
Forgeable passes	HMAC-signed, single-use, time-limited QR tokens	P1
Late detection	Scheduled overdue sweep, push reminder, live dashboard signaling	P2

© 2026 IJSRET

Problem	Solution component	Protocol
Room and facility disputes	Move-in and move-out reports with dual signature; incident reports with media	P3, P4
Conduct follow-up	Scored evaluations bound to the supervisor's residences	P5
Management information	Live counters and weekend statistics snapshots	P2, P5
Unauthorized access	Roster whitelist, account approval, dual-token JWT, role and residence scope	P6
Student support	AI assistant grounded in an administrator-managed knowledge base	AI

9. Actors And Their Relationships

Figure 1 summarizes the actors and their interactions with the application. Table 4 lists their responsibilities and scope.

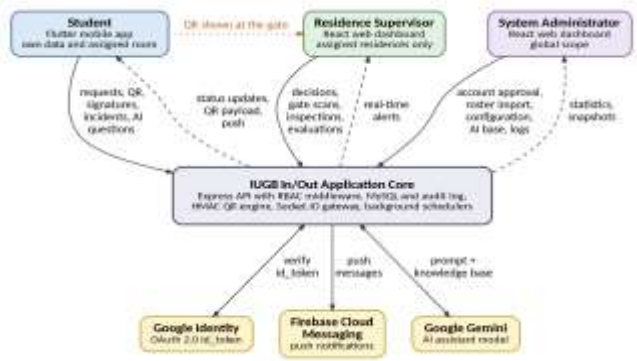


Figure 1: Actors, application core and external services

The relationships are hierarchical in authority and scoped in data. The student interacts only with his or her own records. The supervisor acts on the students and rooms of the residences to which he or she is bound. The administrator acts across residences and, in the use case model of Section 7, inherits the operational capabilities of supervisors. Students and supervisors also meet physically at the residence gate, where the QR token is presented and validated; this physical interaction is the only one that does not transit through the application until the scan is submitted. The application core is both an actor and the arbiter: it is the sole component that changes state, it triggers the overdue protocol without human initiative, and it notifies the human actors.

Table 4: Actors, scope and responsibilities

Actor	Client and scope	Main responsibilities
Student	Flutter mobile application; own data and assigned room	Request exits; generate and present QR tokens; sign or reject room reports; submit incident reports; chat with the AI assistant
Supervisor	React web dashboard; assigned residences only (SupervisorResidence)	Approve or refuse exits; scan QR tokens at the gate (or use manual fallback); justify overdue returns; create room reports; treat incidents; record evaluations
Administrator	React web dashboard; global scope	Import the student roster; approve or reject accounts; configure semesters, residences, buildings, rooms and assignments; manage the AI knowledge base; consult statistics, snapshots and audit logs
Application core	Express API, Socket.IO gateway, schedulers	Enforce authentication, roles and rules; issue and verify QR tokens; persist data and logs; detect overdue returns; emit real-time events and push notifications
External services	Google Identity, Firebase Cloud Messaging, Google Gemini	Verify Google identity tokens; deliver push messages; generate assistant answers

10. Protocol Overview

Table 5: Summary of the residence life protocols

© 2026 IJSRET

Protocol	Objective	Initiating actor	Other actors	Key artifacts
P1 Exit and return	Control and trace every departure and return	Student	Supervisor, application core	Sortie, QrToken, LogAction
P2 Overdue management	Detect late returns and escalate once	Application core (scheduler)	Student, supervisor, administrator	Sortie, LogAction (SORTIE_OVERDUE, SORTIE_JUSTIFIED)
P3 Room inspection	Establish accountability for room condition	Supervisor	Student	RoomReport, RoomReportItem
P4 Incident handling	Route and resolve facility incidents	Student	Supervisor, administrator	IssueReport, attachments
P5 Behavior evaluation	Record structured conduct assessments	Supervisor or administrator	Student (subject)	Evaluation
P6 Identity and governance	Control who may use the platform and with what rights	Administrator	Student, supervisor, application core	StudentImport roster, User, tokens

11. P1 – Exit And Return Protocol

Objective. Replace paper passes by a digital workflow in which every departure and return is authorized, validated by a verifiable credential and recorded. A student exit is called a *sortie*.

Steps. The protocol has six steps, shown in Figures 2 and 3.

1. Step 1, request initiation. The student submits a destination, a reason, a planned departure date and a planned return date. The backend resolves the active semester, verifies that the current time falls within the semester bounds (with a two-day grace period) and that no earlier request is still active (pending, or accepted without a recorded return). The request is stored with status PENDING.
2. Step 2, supervisor decision. The supervisors assigned to the student's residence review the queue. Approval sets the status to ACCEPTED and refusal to REFUSED. In both cases a Socket.IO event refreshes the dashboards and a push notification informs the student.
3. Step 3, QR generation and signing. For an accepted request a dynamic QR token is generated by the student or by the supervisor. The server creates a QrToken record (32-character hexadecimal token, type SORTIE or RETOUR, residence and expiry, with a configurable lifetime whose default is 600 seconds) and computes an HMAC-SHA256 signature over the token, type, residence identifier and expiry using a server secret, so that the code cannot be altered or reused in another residence [12].
4. Step 4, gate validation of the exit. At the gate the supervisor scans the student's code (or the student scans a supervisor gate code). The validation endpoint checks the token's existence, expiry, residence scope and signature. An atomic update that sets the use timestamp only if it is still empty prevents double use and replay. The actual departure time is recorded and the student's location status becomes OUT.
5. Step 5, gate validation of the return. A RETOUR token is validated the same way. The actual return time is recorded, the status becomes RETURNED, the student's location status reverts to IN and the net hours and nights spent outside the campus are computed.
6. Step 6, manual fallback. If the camera or connectivity fails, authorized staff can apply the same transitions through manual scan-out and scan-in operations, which leave an audit entry.

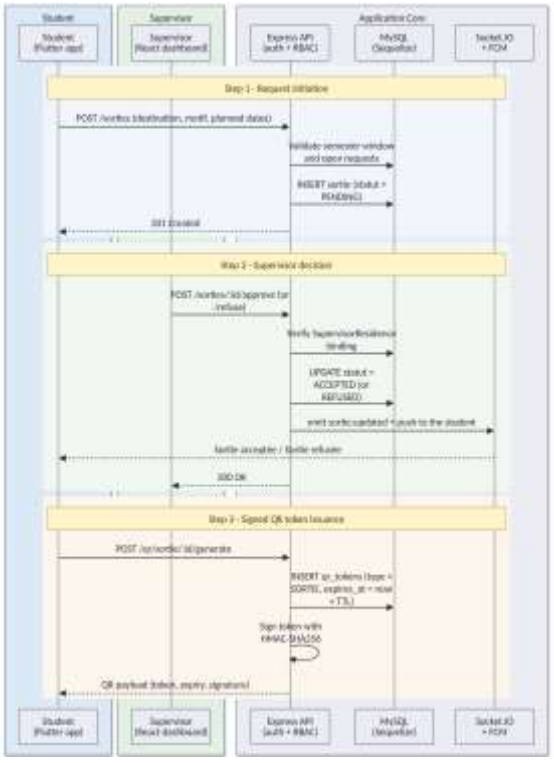


Figure 2: Exit request, supervisor decision and QR token issuance (steps 1 to 3)

© 2026 IJSRET

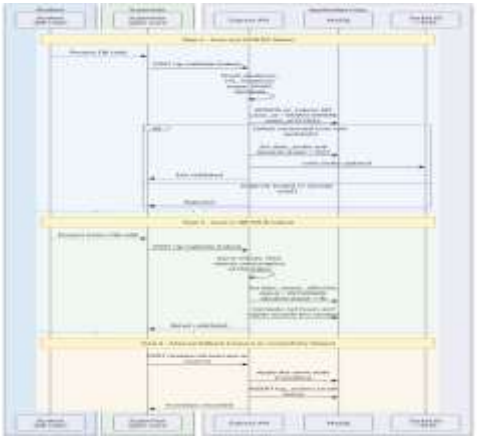


Figure 3: Gate validation of exit and return, and manual fallback (steps 4 to 6)

State machine. Figure 4 shows the lifecycle of a sortie. A request starts as PENDING and ends either as REFUSED or, after the scan-out and the scan-in, as RETURNED. The overdue condition is not set by a user decision; it is derived by the background process of P2.

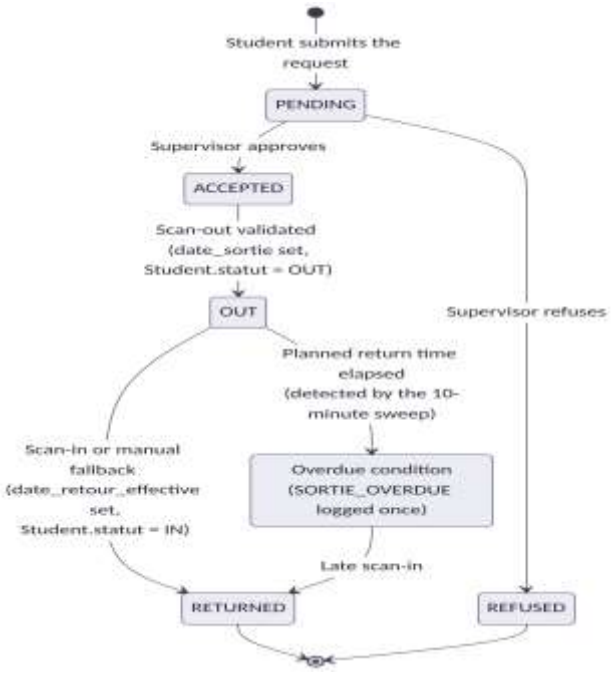


Figure 4: Lifecycle of a sortie

Invariants. (i) A student cannot hold two active sorties. (ii) A supervisor can decide only on students of his or her residences. (iii) A token is valid only for its residence, until its expiry and for a single use. (iv) Manual operations apply the same transitions as scans and are always logged.

12. P2 – Overdue Management Protocol

Objective. Make late returns visible quickly without flooding supervisors with repeated alerts.

Steps. A scheduler runs every 10 minutes (configurable). It selects the sorties that have a recorded departure, no recorded return, a planned return time in the past and a status other than RETURNED or REFUSED. For each one it checks the audit log for an existing SORTIE_OVERDUE entry for that sortie; if there is none it inserts one, sends a push reminder to the student and broadcasts a real-time event to the administrators' and supervisors' rooms. The dashboard then increments its overdue counters, highlights the row and plays an audio signal. A supervisor or administrator can justify a legitimate delay (for example a transport failure) with a comment, which is stored as a SORTIE_JUSTIFIED entry. Figure 5 shows the sequence.

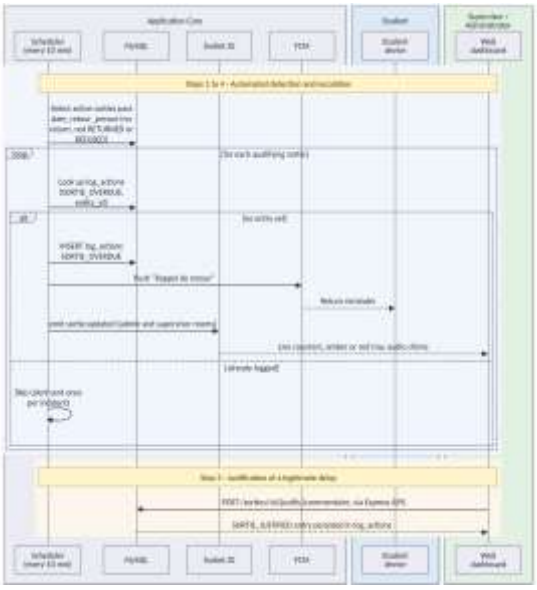


Figure 5: Overdue sweep, alert escalation and justification

Invariants. The alert is emitted exactly once per overdue incident (deduplication through the audit log), and a late scan-in closes the sortie as RETURNED.

13. P3 – Room Inspection Protocol

Objective. Establish accountability for the condition of a room at move-in and move-out.

Steps. After a room is assigned to a student, a supervisor creates a MOVE_IN report listing the fixtures and furniture, each with a condition rating (good, average or poor) and remarks. The report is sent to the student's application for review. If the student agrees, the signature is recorded and the report becomes final; if the student disputes an item, the rejection is recorded and the supervisor must re-inspect and revise the report. At the end of the semester or at checkout, a MOVE_OUT report is created and linked to the original

© 2026 IJSRET

MOVE_IN report, so that the two states of each item can be compared. Figure 6 shows the states.

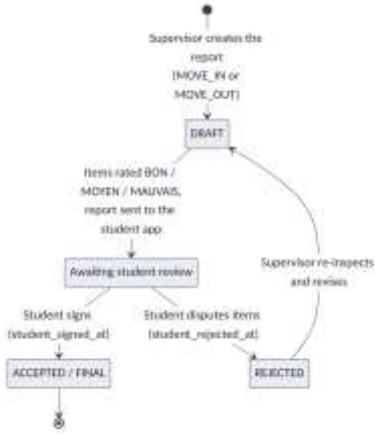


Figure 6: Workflow of a room inspection report

Invariants. A report is final only after a student signature; a rejection returns the report to the draft state; a MOVE_OUT report references its parent MOVE_IN report.

14. P4 And P5 – Incident Handling And Behavior Evaluation

P4, incident handling. A student submits an incident report with a title and a description, and may attach photos or videos (up to 5 MB per file, restricted to image and video types) stored on the server. The API dispatches the report to the supervisors assigned to the student's residence. Supervisors or administrators assign the person in charge and update the status along the sequence OPEN, IN_PROGRESS, RESOLVED and CLOSED. Every transition emits a real-time notification to the student's application and updates the badge counters of the supervisor dashboard.

P5, behavior evaluation. Supervisors or administrators record evaluations of students living in their residences, with integer scores from 0 to 10 for general discipline and for community life, a qualitative indicator of substance use with its frequency, and comments. Middleware enforces the residence scope, and each evaluation stores the identity of the evaluator and a creation timestamp, giving a tamper-resistant history that can inform housing renewal and disciplinary reviews. Because these records are sensitive, they raise data protection requirements that are discussed in Section 10.

15. P6 – Identity, Authentication And Governance Protocol

Objective. Ensure that only legitimate people use the platform and that each acts within a defined scope.

Steps. (1) Public self-registration of students is disabled: administrators import institutional rosters (Excel) that pre-populate student codes, emails, names, parent phone numbers and semester bindings. (2) Authentication uses a dual-token scheme: a short-lived access token (15 minutes) and a refresh token whose validity is tied to a per-user version counter, so that logout, password change or access revocation invalidate existing sessions; Google sign-in is available under account-linking constraints. (3) Accounts are in one of three states (pending, approved, rejected) and only approved accounts reach protected resources; a pending or rejected account receives an explicit error. (4) Three authority levels apply: students act on their own data, supervisors act within their residences, administrators govern globally. Figure 7 shows the chain of checks applied to every request.

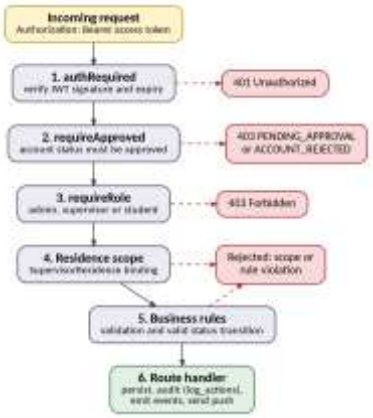


Figure 7: Chain of access checks applied to protected requests

Table 6 gives the resulting permissions

.

Operation	S	V	A
Request a sortie and track own sorties	●		
Approve or refuse a sortie		●	●
Generate a QR token	●	●	
Validate a QR token, manual scan-out and scan-in		●	●
Justify an overdue sortie		●	●
Create a room report		●	●
Sign or reject a room report	●		
Submit an incident report	●		

© 2026 IJSRET

Operation	S	V	A
Assign and update an incident report		●	●
Record a behavior evaluation		●	●
Chat with the AI assistant	●		
Import the roster and approve accounts			●
Configure semesters, residences, rooms, supervisors			●
Manage the AI knowledge base			●
Consult statistics, snapshots and audit logs			●

Table 6: Permissions by role (S = student, V = supervisor, A = administrator)

Supervisor operations are limited to the residences to which the supervisor is bound, whereas administrator operations are global.

IV. DATA MODELS AND USE CASES

1. Use Cases

Figure 8 presents the use cases grouped by the actor that mainly performs them. The student-facing use cases cover authentication, requesting and tracking sorties, generating a QR token, signing room reports, submitting incidents and chatting with the assistant. The supervisor use cases cover decisions, gate validation, overdue follow-up, room reports, incident treatment and evaluations. The administration use cases cover roster import, account approval, structure configuration, supervisor assignment, the knowledge base, statistics and audit logs. The administrator inherits the supervisor use cases with a global scope. Tags P1 to P6 refer to the protocols of Section 6, and AI to the assistant.

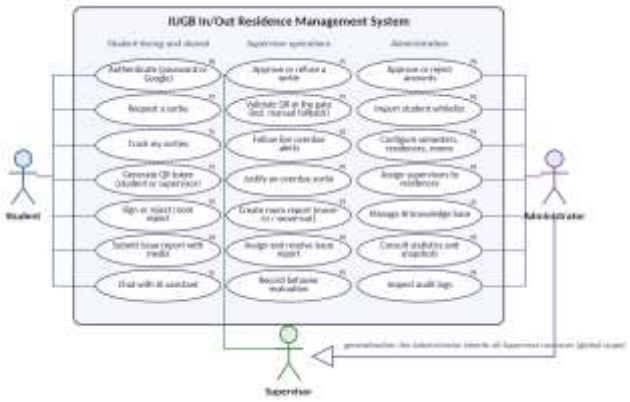


Figure 8: Use case diagram of IUGB In/Out

2. Data Model

The database contains nineteen entities, each implemented as a Sequelize model: User, Student, Semestre, Residence, SupervisorResidence, Building, Room, RoomAssignment, Sortie, QrToken, IssueReport, IssueReportAttachment, RoomReport, RoomReportItem, Evaluation, AdminStatSnapshot, UserDeviceToken, AiKnowledge and LogAction [7]. We present them in four views. Table 7 summarizes their roles.

Table 7: Entity groups and roles

Group	Entities	Role
Identity and access	User, Student, UserDeviceToken, SupervisorResidence	Accounts with role, status and token version; student profile with code and location status (IN or OUT); push device tokens; supervisor-to-residence binding
Residence structure	Semestre, Residence, Building, Room, RoomAssignment	Academic period, residences (with gender type and capacity), buildings, rooms and the timeline of room assignments
Movement	Sortie, QrToken	Exit request with planned and actual dates and status; signed QR token with type, expiry and use timestamp
Reporting and governance	RoomReport, RoomReportItem, IssueReport, IssueReportAttachment, Evaluation, LogAction, AdminStatSnapshot, AiKnowledge	Room condition reports, incidents with media, evaluations, audit entries, statistics snapshots and assistant knowledge

Figure 9 shows the identity and residence-structure view. A user has an optional student profile; supervisors reach residences through the SupervisorResidence bridge table; a residence belongs to a semester and contains buildings and

© 2026 IJSRET

rooms; students are linked to rooms through RoomAssignment, which keeps a start and end date so that occupancy has a history.

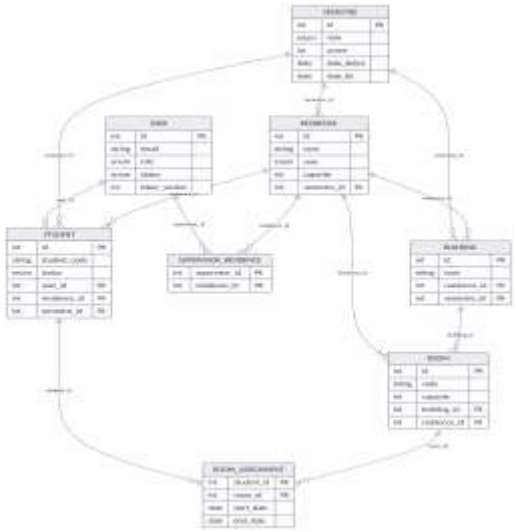


Figure 9: Data model, identity and residence structure

Figure 10 shows the movement view. A Sortie belongs to a student and references the QrToken used for its validation; a QrToken belongs to a residence, has a type (SORTIE or RETOUR), an expiry and a use timestamp.

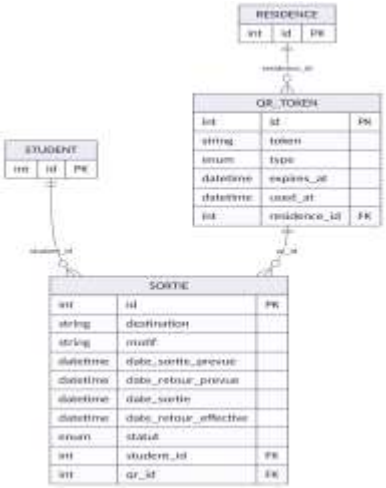


Figure 10: Data model, movement and QR tokens

Figure 11 shows the room reports and incidents. A RoomReport belongs to a student and a creator and may reference a parent report (the MOVE_IN report of a MOVE_OUT report); it owns checklist items. An IssueReport belongs to a student, may be assigned to a supervisor or administrator, and owns attachments.

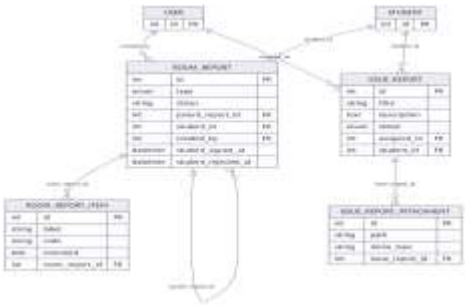


Figure 11: Data model, room reports and incidents

Figure 12 shows the evaluation, audit and device-token entities. LogAction records the actor, the action name, the target entity and its identifier, with a timestamp; it is the basis of the audit trail and of the deduplication of overdue alerts. AdminStatSnapshot (one row per time slot, with a payload) and AiKnowledge (title, content, active flag and order) are standalone tables.

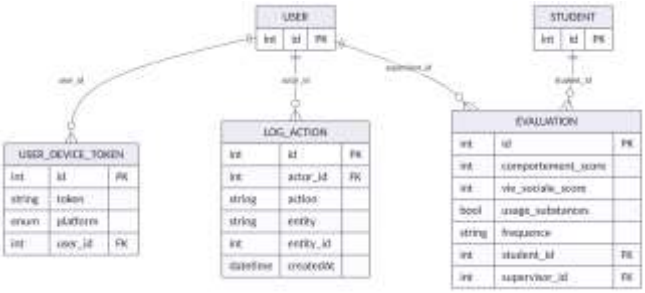


Figure 12: Data model, evaluations, audit log and device tokens

3. Integrity Considerations

Integrity is preserved by controlled status transitions that prevent invalid states, by server-side validation before persistence, by foreign keys between the entities above, by a unique constraint on the snapshot time slot and on the token value, and by checks that reject role and state mismatches. Planned and actual timestamps are stored separately on every sortie, which allows durations outside the campus to be computed rather than declared.

V. ARCHITECTURE

1 Three-Tier Architecture

Figure 13 shows the architecture, organized in three tiers. The presentation tier contains the Flutter mobile application and the React web portal. The application tier is a Node.js and Express backend that combines security middleware, REST route modules, domain services, a Socket.IO gateway and background schedulers. The data and services tier

© 2026 IJSRET

contains the MySQL database accessed through Sequelize, a file storage for incident media and the cloud services.

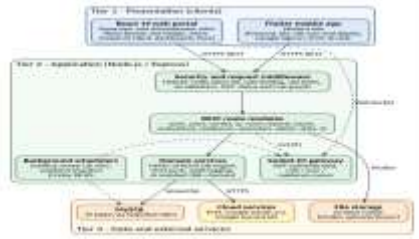


Figure 13: Three-tier architecture of IUGB In/Out

2. Backend

At startup the backend composes, in sequence, proxy trust settings, security headers, origin filtering, body parsers, request logging and authenticated static serving of uploads, so that every request crosses the same safety layers before reaching business logic. Routes are segmented by domain: authentication, semesters, residences, students, supervisors, sorties, QR, incident reports, room reports, rooms, administration, roster import, devices, AI, AI knowledge and health [7]. Background services run independently of requests: the overdue sweep, the weekly statistics snapshot (every Friday at 18:30, with a loop that recreates missing snapshots after downtime) and the push reminder dispatch.

3. Clients

The React web portal uses protected routes by role with dedicated layouts. The administrator experience includes the dashboard, weekend statistics, roster import, residence, building and supervisor governance, account approvals and knowledge-base management. The supervisor experience includes the overview of assigned students, rooms, the sortie queue, incident and room report processing, evaluations and QR validation. Live badge counters, visibility-aware refresh and audio signals reduce reaction time. The Flutter application uses Riverpod providers, a repository pattern and a shared API 0client; its navigation covers the home dashboard, sorties, QR scanning, incident reports and profile, with the AI chat available from a floating button. It supports credential login and Google sign-in, session continuity, and synchronization of the push token after authentication.

5.4 Data Flows

Figure 14 shows the main data flows. Clients send requests to the API over HTTPS (F1); the API reads and writes the database and the audit log (F2); domain events are published to Socket.IO rooms addressed to a role, a user or a residence, and received by the connected clients (F3); push messages are sent through Firebase Cloud Messaging to registered device tokens (F4); and assistant questions are completed with the active knowledge base before being sent to the language model (F5). The schedulers read and write the database and use F3 and F4 to alert users.

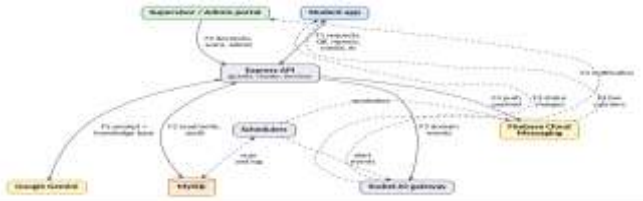


Figure 14: Main data flows

5. Ai Assistant And Knowledge Base

The assistant is designed to give students immediate support on residence rules, practical procedures, campus life and general educational guidance within safety boundaries. Knowledge is stored in the AiKnowledge table, which administrators can create, update, activate, deactivate and delete as policies evolve. When a student asks a question, the backend assembles a structured system prompt that enforces grounding for residence topics and permits broader support for general academic questions, includes the active entries ordered by priority, and sends the whole to the Gemini model through the Google generative AI SDK. Reliability controls include a message length limit, per-user throttling and a fallback model used when the primary model reaches quota limits. This follows the retrieval-augmented principle of conditioning generation on curated external documents [17], in a simple form (all active entries are injected in the prompt rather than retrieved by similarity).

6 . Security Design

Authentication uses a 15-minute access token and a revocable refresh token whose validity depends on a per-user version counter [13]; passwords are hashed with bcrypt [14]. Authorization combines role middleware with account-status and residence-scope checks (Figure 7). The API is hardened with request size limits, strict CORS origin filtering, security headers and global and per-route rate limiting; server start-up validates critical secrets and refuses weak defaults. QR tokens are signed with HMAC-SHA256 [12] and consumed atomically. Socket.IO connections are authenticated with the same tokens and subscribed only to the rooms matching the user's role, identity and residence. Key events such as logins, logouts and overdue detections are logged.

VI. IMPLEMENTATION

1. Technology Stack

Table 8 lists the technologies used in each layer. The code base is organized in three directories: the backend (API, models, routes, services, scripts), the web client and the Flutter student application.

Table 8: Technology stack

© 2026 IJSRET

Layer	Technologies
Backend	Node.js [25], Express [26], Sequelize ORM [27] over MySQL [28], jsonwebtoken, bcrypt, Joi validation, Helmet, CORS, express-rate-limit, Multer (uploads), Socket.IO [29], Firebase Admin, Google generative AI SDK
Web client	React [30], React Router, Ant Design, Axios, Socket.IO client, React Markdown (assistant answers), chart libraries, Excel import and export
Mobile client	Flutter [31], Riverpod, Dio, Firebase Messaging with local notifications, QR scanner and generator packages, Google Sign-In [32]
External services	Firebase Cloud Messaging [33], Google Identity [32], Google Gemini [34]

2. Development And Integration

The web and mobile clients were developed against the same backend revision to avoid inconsistencies, and environment-specific values are externalized. Setup is script-driven: dependencies are installed, the environment file is configured, the MySQL database is created, seed scripts are run where needed and the server is started; the web client is then connected to the API base URL, and the mobile client is built after Firebase configuration. A roster import script restricts registration to whitelisted students, and diagnostic scripts support troubleshooting.

3. Implementation Of The Main Protocols

The sortie workflow is implemented as route handlers that enforce the sequence of checks of Figure 7 and update the Sortie and Student records in one flow. QR issuance creates a QrToken row and signs the payload with a secret held only by the server; validation performs the existence, expiry, residence and signature checks, then marks the token used with a conditional update so that two concurrent scans cannot both succeed. The overdue sweep is a scheduled task whose interval is configurable and defaults to 10 minutes, and which uses the audit log as its deduplication store. Real-time events are emitted by handlers after persistence and are targeted to rooms; push notifications are sent to the device tokens registered after login. Incident media are uploaded through Multer with size and type restrictions and served through an authenticated static route. The weekly snapshot job aggregates the operational statistics shown on the administrator dashboard, such as students by residence and sex, exits during the weekend, presence and exit rate.

4. Quality Assurance

The test strategy combines endpoint tests organized by role, workflow scenarios (request, approval, scan-out; late return flagged as overdue; rejected account denied; knowledge-base update reflected in answers; room report sign and reject paths), user interface checks on web and mobile, and security tests on authentication failure, forbidden access and rate limiting. Quality practices include structured error responses, centralized validation, script-driven diagnostics and release checks. These are the planned and documented scenarios; their systematic execution with recorded results is part of future work (Section 10.4).

VII. RESULTS

1. What The Project Delivers

The implemented platform provides centralized digital control of residence operations, with role-specific task paths for each user type, QR-secured confirmation of movements that removes the risk of forged paper passes, faster visibility of late returns and incidents, and improved availability of data for governance [7]. Table 9 relates the delivered functions to the requirements of Table 2.

Table 9: Requirements and delivered functions

Requirement	Delivered function
FR1, FR2, NFR1	Dual-token authentication, Google sign-in, account states, role and residence guards, hardened API
FR3, FR4	Sortie request, decision, signed single-use QR validation at the gate, manual fallback
FR5	Scheduled overdue sweep with once-only alert, push reminder, live signaling and justification
FR6	Incident submission with media, residence-scoped routing, status tracking with real-time feedback
FR7	Move-in and move-out reports, student signature or rejection, parent link
FR8, FR9	Semester and residence administration; statistics page and Friday snapshots
FR10	Administrator-managed knowledge base and AI chat with fallback model
NFR2, NFR4, NFR6	Controlled status transitions; modular route structure; logs and health route
NFR5	Web portal for desktop and Flutter application for mobile

© 2026 IJSRET

2. Improvements To Residence Processes

Relative to the manual processes described in Section 4.2, the system changes five things. The credential for an exit becomes a signed token with a lifetime of minutes, instead of a static paper pass. The record of who is out becomes a database state updated at every scan, instead of a register. Late returns are detected by a scheduler and signaled once, instead of being discovered during rounds.

Room condition and incidents are captured with a defined workflow and a history, instead of ad hoc messages. Management data are persisted and summarized weekly.

In operational terms the system is expected to reduce paperwork, improve procedural consistency across residences, shorten reaction time to incidents and overdue cases, and strengthen accountability through automatic logs [7]. We stress that these are expected benefits derived from the design: no before-and-after measurement (for example of processing time or of detection delay) has been carried out.

3. Comparison With Documented Practices

In relation to Table 1, the system adds an electronic exit and return record, automated overdue detection and digital room reports to the controls that the CROU residence regulations describe (after-midnight checks, inventories, unannounced inspections) [18], [21].

It does not replace them: guards and supervisors remain necessary at the gate, and the platform does not manage admission, eligibility or payments, which are covered by the CROU online platforms [2]. A natural integration path is to feed the roster from an admission platform so that eligibility decisions and daily residence protocols share the same identities.

4. Limitations And Threats

- No field evaluation. The test scenarios are documented but no quantitative results (latency, throughput, detection delay, user satisfaction) are reported. Simulation-based figures in related work [10] should not be read as comparable.
- Connectivity. Continuous real-time behavior depends on stable Internet access; manual fallback covers gate scans but there is no offline-first synchronization.
- Third-party dependence. The assistant depends on external quotas and availability, and the assistant injects all active knowledge entries in the prompt, which will not scale to a large knowledge base without retrieval.
- Reliance on disciplined use. Some aspects of reliability still depend on users and staff acting as expected (for example scanning at every exit).
- Data protection. The platform stores personal data (identity, parent phone numbers, movements, evaluations including substance-use indicators). Deployment requires a declaration or authorization with the ARTCI under law 2013-450 [23], [24], a retention policy, and a review of who may read evaluations and logs.
- 6. Comparison limits. Public information on other Ivorian residences, in particular non-CROU ones, is scarce; the comparison is limited to what the cited sources document.
- 7. Operational risks. Configuration drift between environments, manual data entry quality and variable push delivery on some devices should be monitored.

5. Future Work

Planned directions are biometric or smart-gate integration, predictive analytics for occupancy and risk, tools to balance supervisor workload, advanced filters across semesters and residences, documented public API contracts, multilingual and accessibility improvements, and offline-assisted mobile capture with delayed synchronization. In addition, a pilot with recorded measurements, a privacy impact assessment, and retrieval-based selection of knowledge entries for the assistant would strengthen the evidence and the scalability of the system.

VIII. CONCLUSION

This paper presented IUGB In/Out, a residence management system built around six explicit residence life protocols. The exit and return protocol combines supervisor approval, signed single-use QR tokens and atomic validation; the overdue protocol detects late returns and alerts once; room inspection and incident handling give accountability for the state of rooms and facilities; behavior evaluation and identity governance complete the picture under role and residence scoping. The protocols are supported by a nineteen-entity relational model, a three-tier architecture with real-time and push channels, and an AI assistant grounded in a managed knowledge base.

Placed against the publicly documented practice of Ivorian CROU residences, which combine online admission with human-operated controls and increasingly offer online incident reporting at center level, the system contributes the missing layer of daily, traceable residence protocols.

Its evidence is, for the moment, an operating implementation and documented test scenarios, not a measured field study. The next step is therefore a supervised pilot at IUGB with quantitative indicators, coupled with the data protection steps required by Ivorian law.

© 2026 IJSRET

REFERENCES

1. AIP, "Enseignement supérieur : Adama Diawara relance la réhabilitation des cités universitaires pour améliorer les conditions de vie des étudiants," Agence Ivoirienne de Presse, Abidjan, 16 May 2026. [Online]. Available: https://www.aip.ci
2. CROU Abidjan 2, "Plateforme d'admission en cité universitaire – Espace Logement," logement.croua2.ci, accessed Oct. 2026.
3. CROU Bouaké, "Logement : admission et réadmission en résidence universitaire," logement.croubouake.ci, accessed Oct. 2026.
4. Koaci, "Côte d'Ivoire : Université Nangui-Abrogoua, un nouveau siège moderne pour le CROU Abidjan II afin de renforcer les services aux étudiants," Koaci.com, 15 May 2026.
5. Afrique Sur 7, "Côte d'Ivoire – FESCI : le gouvernement fait de surprenantes révélations," afrique-sur7.fr, accessed Oct. 2026.
6. MESRS, Direction des Ressources Humaines, "Sigles et définitions" (list of public universities, schools and CROU), drh.enseignement.gouv.ci, accessed Oct. 2026.
7. Mea, "IUGB In/Out Residence Management System," CSC 4499 Capstone Report, International University of Grand-Bassam (IUGB), supervised by Dr. A. C. Bayomock Linwa, 2026.
8. Portail de l'économie ivoirienne, "Enseignement supérieur," economie-ivoirienne.ci, accessed Oct. 2026.
9. CROU Abidjan 2, "Les admissions dans les cités universitaires," nouveau.croua2.ci, accessed Oct. 2026.
10. R. Heda et al., "DHMS: A Digital Hostel Management System Integrating Campus ChatBot, Predictive Intelligence, and Real-Time Automation," arXiv:2507.17759 [cs.HC], 2025.
11. Digiicampus, "Hostel Management System" (vendor product page), digiicampus.com, accessed Oct. 2026.
12. H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, IETF, Feb. 1997.
13. M. Jones, J. Bradley and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, May 2015.
14. N. Provos and D. Mazières, "A future-adaptable password scheme," in Proc. USENIX Annual Technical Conference, 1999, pp. 81–91.
15. I. Fette and A. Melnikov, "The WebSocket Protocol," RFC 6455, IETF, Dec. 2011.
16. C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," Computers and Education: Artificial Intelligence, vol. 2, art. 100033, 2021.
17. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459–9474.
18. CROU Abidjan 2, "Règlement intérieur des résidences," croua2.ci/reglement-interieur-des-residences, accessed Oct. 2026 (decree n° 2013-438 of 13 June 2013 cited therein).
19. CROU Abidjan 2, "Accueil et Logement" (Sous-direction de l'Accueil et des Logements), croua2.ci/accueil-et-logement, accessed Oct. 2026.
20. CROU Abidjan 1, "Sous-direction de l'accueil et du logement," crouabidjan1.ci, accessed Oct. 2026.
21. Abidjan.net, "Résidences universitaires : le DG du CROU annonce les nouveaux prix et nouvelles mesures," Abidjan.net News, article 469613, undated.
22. MESRS, Côte d'Ivoire: enseignement supérieur et recherche scientifique, bilan et perspectives, Government of Côte d'Ivoire, 2012. [Online]. Available: https://www.gouv.ci (publication "Perspectives enseignement sup").
23. Republic of Côte d'Ivoire, Loi n° 2013-450 du 19 juin 2013 relative à la protection des données à caractère personnel.
24. CIO Mag, "ARTCI – Le traitement des données personnelles désormais soumis à une autorisation préalable," cio-mag.com, accessed Oct. 2026.
25. OpenJS Foundation, "Node.js Documentation," https://nodejs.org/en/docs.
26. "Express Documentation," https://expressjs.com.
27. "Sequelize Documentation," https://sequelize.org.
28. Oracle, "MySQL Documentation," https://dev.mysql.com/doc.
29. "Socket.IO Documentation, v4," https://socket.io/docs/v4.
30. Meta, "React Documentation," https://react.dev.
31. Google, "Flutter Documentation," https://docs.flutter.dev.
32. Google, "Google Identity and OAuth," https://developers.google.com/identity.
33. Google, "Firebase Cloud Messaging," https://firebase.google.com/docs/cloud-messaging.
34. Google, "Google Generative AI Documentation," https://ai.google.dev.

© 2026 IJSRET