Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

# Desain User Interface 'Disaster Spatial Learning Global Partnership'

**Aaron Christoperus Prayogo Lauw[1*], Meme Susilowati[2], Budi Handoyo[3]**

*[1, 2]Ma Chung University, Villa Puncak Tidar Blok N No.1 (N-01), Karangwidoro, Dau, Malang, Jawa Timur, Malang, Indonesia*
*[3] UM Malang (Universitas Negeri Malang), Jl. Semarang No. 5, Malang 65145, Jawa Timur, Malang, Indonesia*
*lauwaaron@gmail.com[1]*

**Abstract**

This article presents the design of the User Interface (UI) for the DSL Global Partnership (DSLGP) system, a Web-GIS–based educational platform developed to support disaster literacy and digital learning. The study focuses on translating the Software Requirements Specification into structured UI components, including the Home page, role-based dashboards, and key transactional workflows such as guest registration, assessment distribution, resource validation, and data import operations. The UI is designed using a user-centered approach, emphasizing clarity, accessibility, and role separation to ensure intuitive navigation for research teams, teachers, students, administrators, and public users. This work provides a concise and systematic overview of the UI structure that can guide subsequent implementation and evaluation stages within the DSLGP development cycle.

***Keywords***: *dashboard; disaster education; user interface design; Web-GIS.*

## 1. Introduction

The DSL Global Partnership (DSLGP) system was developed to support disaster education with a spatial basis via a Web-GIS module, integrated assessments, and a publicly accessible resource repository for teachers, students, and research teams [1]. A well-designed user interface (UI) is critical to ensure accessibility, instructional effectiveness, and accurate completion of assessment instruments on an educational platform of this type [1], [2]

The Software Requirements Specification (SRS) document serves as the formal foundation guiding UI design and interoperability among modules in the development of DSLGP. Standard guidance such as ISO/IEC/IEEE 29148:2018 is recommended to produce a consistent, testable SRS [4]. In addition, curricular guidelines and spatial learning scenarios formulated in the draft DSL book and local research proposals provide the implementation context necessary to design use cases, user stories, and functional priorities at the local level [3], [4], [5].

The quality of the SRS, covering completeness, consistency, and testability—affects implementation success and software engineering cost; therefore, measuring SRS quality metrics and conducting pedagogical validation of DSL content are recommended practices in evidence-based projects such as DSLGP [6], [7].

## 2. Methods

The research and article development approach follows three primary phases drawn from the Software Development Life Cycle (SDLC) framework: Analysis, Design, and Implementation/Verification. These three phases were chosen to keep the study focused on requirements design and interface aspects.

### 2.1. Verification Phase (Requirements Elicitation & Documentation)

In this phase, functional and non-functional requirements were collected through literature review, stakeholder forums (teachers, administrators, research team), and field observations at the partner site (SMA Negeri 1 Dampit). The results of this phase were captured in the SRS and use-cases, which directly informed the UI design.


**Figure 1:** Verification Phase Documentation

### 2.2. Design Phase (UI/UX & Information Architecture)

Based on the SRS, the main UI components were designed—Home page, Virtual DSL module, role-based dashboards, and critical transaction flows (Guest registration, assessment creation/distribution, resource upload & validation, bulk data import). The design follows usability and user-centered design principles supported by the literature and is adapted to local needs as formulated in the proposal and draft book, together with software engineering practices described in SDLC methodology sources [8].

### 2.3. Implementation & Verification Phase (Prototyping and Review)

Hasil desain diimplementasikan sebagai prototipe antarmuka (mockups / screenshot) untuk diuji fungsionalitas dasar dan dilengkapi dokumentasi alur (workflow) serta sequence diagrams untuk transaksi utama; verifikasi dilakukan dengan review internal Tim Riset dan pemeriksaan konsistensi SRS terhadap implementasi awal. Praktik ini sejalan dengan rekomendasi SDLC untuk mengurangi biaya perbaikan pada tahap akhir [9], [10].

## 3. Content and Discussion

This chapter discusses the contents of the study.

### 3.1. UI Workflow

#### 1. Overview

The user interaction flow at the interface level follows a general pattern: public (anonymous) → Home (landing) → Learn More / Guest registration → validation process by the Research Team → Guest account activation → authentication → redirect to the role-appropriate dashboard. In addition, there are internal flows for role management (Superadmin provisioning Admin/Research Team), test distribution (Research Team → Teacher/Student), and content validation flow (Teacher/Student → Research Team validation → publish to Resources).

#### *2.* Feature and Action Details

To clarify user interaction flows within the system, this section presents a detailed breakdown of the interface structure based on the designed sitemap and features. Each page is described together with the actors that interact with it and the actions available via buttons or related UI components. Presenting this information in tabular form provides a structured overview of inter-page relationships, the access rights of each user role, and available operational functions. Consequently, this table can serve as a reference for development, requirements validation, and test-scenario specification.

**Table 1:** Feature and Action Details

| No | Actor | Page (Feature) | Actions (Button / Feature) |
|---|---|---|---|
| 1 | Public | Home (Home Page) | Navigation menu (Home, Virtual DSL, DSL, Resources, Learn More, Sign In); GPS permission request (Allow / Deny); Announcement widget (open / download attachment); Weather widget (view history & forecast); Links to Learn More / Resources / Virtual DSL; Sign In button. |
| 2 | | Virtual DSL | Open project description; status & roadmap links; participate/follow or project detail link (primarily navigation/links). |
| 3 | | DSL (content page) | Filter / search content; open content item; download attachments; display placeholder if empty (“No DSL content yet”). |
| 4 | | Resources (repository page) | Filter / search; open resource; download file; for authenticated users — link to Upload / Resource Sharing form. |
| 5 | | Learn More (company profile) | “Register Now!” button → open modal/registration form (Name, Address, Phone, Email, location icon button); Cancel / Submit buttons. |
| 6 | | Sign In | Username / Password fields; Sign In button; after login changes to Profile / Sign Out. |

**Table 2**: Continued Feature and Action Details

| No | Actor | Page (Feature) | Actions (Button / Feature) |
|---|---|---|---|
| **7** | Research Team | Research Team Dashboard | Filters (Gender, Class, Academic Year, Test Type); visualizations (charts/pie/bar); export data/charts; links to management pages (Manage Tests, etc.). |
| **8** | | Manage Tests (test list) | Add Test (popup); View Questions; Status (change: active / paused / finished); Edit; Duplicate; Delete; View Results (for finished tests). |
| **9** | | Add Test (popup/form) | Fields: Test Name, Instrument Type, Test Type; Cancel / Add buttons. |
| **10** | | View Questions / Add Question | Question list; Add Question (popup) with fields: Question, Disaster Type, Category, Attachment, Answer Type; Save / Cancel buttons. |
| **11** | | Manage Disaster Types | Add Disaster (popup); disaster type list; Status (active/inactive); Edit; Delete. |
| **12** | | Add Disaster Type (popup) | Fields: Disaster Name, Description; Cancel / Add buttons. |
| **13** | | Manage Assessments | View assessment parameters (Ready / Less Ready / Not Ready); Add Parameter (if permitted); per-parameter actions: Status / Edit / Delete. |
| **14** | | Add Parameter (popup) | Fields: Parameter Name, Minimum Value, Maximum Value; Cancel / Add buttons. |
| **15** | | Manage Content (Announcements / Resources / DSL) | Add Content (popup); content list; Status (active/inactive); Edit; Delete; upload file/link. |
| **16** | | Add Content (popup) | Dropdown Content Type; Title; Body/Description; Upload File / Link; Cancel / Add buttons. |
| **17** | | Content Validation | Pending Validation widget (submission list); Approve / Reject buttons; on Reject → rejection-reason popup; View File (download); Validation History. |
| **18** | | Master Indicators | Add Indicator (popup); indicator list with Status / Edit / Delete. |
| **19** | | Administrative Unit (TU) Settings | Add TU Admin (popup); TU Admin list: View / Edit / Delete. |
| **20** | | Guest Settings — Discussion Inbox | Guest message list; open message; Reply field (optional attachment); Send Reply / Cancel buttons. |
| **21** | | Guest Settings — Manage Guests | Guest list; Add Guest (popup); Validate Guest (Accept/Reject); Edit; Delete; Registrations Pending Validation list + Validate button. |
| **22** | Admin | TU Admin Dashboard | Filters (Gender, Class, Academic Year); summary widgets (counts of students/teachers/classes, active AY); charts; data export. |
| **23** | | Manage Student Data | Add Student (popup); Import XLSX (Import / Preview / Commit / Rollback); Download Template; Search; per-row actions: Status / Edit / Delete. |
| **24** | | Add Student (popup) | Fields: Name, Student ID (NIS/NISN), Date of Birth, Gender, Address, Username, Password; Cancel / Add buttons. |
| **25** | | Manage Teacher Data | Add Teacher (popup); Import XLSX; Download Template; Search; per-entry: Status / Edit / Delete. |
| **26** | | Manage Classes | Add Class (popup); class list; Status / Edit / Delete. |
| **27** | | Manage Academic Years | Add Academic Year (popup); set active Status; Edit; Delete (only one AY active at a time). |
| **28** | | Manage SKT (Student–Class–Academic Year) | Filter by Academic Year → Class; display student list; Edit assignment (move class/AY); Save changes. |
| **29** | | Manage SKT & Teachers (SKTG) | Class list + assigned responsible teacher; Edit button to assign / reassign. |
| **30** | Teacher | Teacher Dashboard | Observation progress widget; class completion progress; filters (Gender / Class / AY); average charts; links to Student Observations & Resource Sharing. |
| **31** | | Fill Student Observations | Filter Class → Student; checkbox select (multiple / select all); per-student observation form; Save Draft / Submit buttons. |
| **32** | | Resource Sharing (Teacher) | Upload Resources button (popup); submission list (status: Pending Validation / Approved / Cancelled); Cancel Submission button. |
| **33** | Student | Pre-test (list) | List of distributed tests; Start (Begin) button; status indicators. |
| **34** | | Pre-test (test-taking / assessment) | Questions (multiple types); Save Draft; Submit; upload attachment (if any). |
| **35** | | Post-test (list & test-taking) | Same as Pre-test: list; Start; Save Draft; Submit; view results after submission. |
| **37** | Guest | Guest Discussion | New message (+) button (popup: Title, Body, Attachment) — Cancel / Send; open thread; add attachment; Reply (Send Reply). |
| **38** | Superadmin | Research Team Menu (dropdown) | Dropdown access to all Research Team menus (Dashboard, Manage Tests, etc.); open / assume Research Team pages; view / assist functions. |
| **39** | | TU Admin Menu (dropdown) | Dropdown access to all TU Admin menus (Dashboard, Manage Student Data, etc.); open / assist. |

### 3.2. Home Page

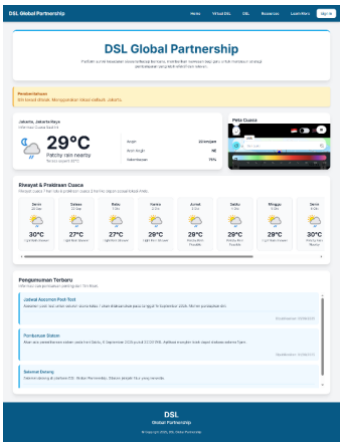


**Figure 2:** Home Page User Interface

The Home page is the public front page of the DSL Global Partnership application that presents a summary of functions and services without requiring authentication. Primary content includes an integrated weather widget (real-time conditions, 7-day history, and 2-day forecast) that adapts its display based on device coordinates when the user grants location permission; if location permission is denied, the system falls back to a default location of DKI Jakarta. The page also contains a "Latest Announcements" panel for assessment notifications, system updates, or important materials from the Research Team, and a Guest registration call-to-action (CTA) button that opens a modal form. After the registration form is submitted, users receive a confirmation and the registration entry is assigned status "Pending Validation" by the Research Team.
Other pages reachable from the Home header include:

1. **Virtual DSL:** an interactive, map/simulation module enabling users to visually explore disaster scenarios and run simulation exercises (real-time or scheduled).
2. **DSL:** a collection of domain-specific learning modules (materials, manuals, reference documents) organized by topic/disaster type to support learning.
3. **Resources:** a repository of validated content (teaching materials, guides, media) available for download or access according to user permissions.
4. **Learn More:** an informational page for the public containing program objectives, participation guidelines, FAQs, and short case studies.
5. **Sign In:** the authentication entry point for registered users; upon successful login users are redirected to their role-appropriate dashboard.

C. Dashboard per User

**Table 3:** Role Details

| Role | Primary Purpose | Main Widgets | Key Transactions |
|---|---|---|---|
| Research Team | Design, analyze, and validate research data on disaster preparedness | Pre/Post comparison charts; sample filters (class / AY / gender); respondent distribution map; resource validation panel; export controls | Create / Manage / Distribute tests; Validate resources; Export data; Manage assessment indicators |
| School Admin | Manage school administrative data and mass administration | Counts summary (students / teachers / classes / AY); XLSX import with preview; audit log | Import / Preview / Commit / Rollback bulk data; CRUD Students / Teachers / Classes; Generate reports |
| Teacher | Conduct classroom observations, assign tasks, and upload teaching materials | Class list; student observation progress; My Tests list; upload resource panel | Distribute assignments; Fill & Submit observations; Upload resources for validation |
| Student | Access and complete assessments and upload work | Active tests list (My Tests); submission status; individual results; access to resources | Take tests (Pre/Post); Submit; Upload work / portfolio |
| Guest | Interact with Research Team and access limited public content | Discussion inbox (threaded); registration status; notifications | Send messages / threads; Receive replies; View validation status |
| Superadmin | Provision accounts, monitor the system, and perform role interventions | Per-role preview dropdown; provisioning panel; audit view | Create Admin / Research Team accounts; Reset passwords; View audit logs |

Additional brief descriptions for each user:
1) The Research Team is responsible for assessment design, data analysis, and validation of incoming content. Their dashboard centers visual analytics (Pre/Post comparisons, demographic distribution) accompanied by granular filters to slice samples and a panel that holds resource items pending validation. Functionally, the Research Team dashboard displays Pre vs Post comparison charts with sample sizes, a respondent distribution map, a Pending Items panel, and quick controls to create, duplicate, activate assessments, and export data. Core workflows include assessment creation (metadata → add questions → map indicators → save/draft → activate), distribution to teachers/students, aggregation of assessment_submissions, and the resource validation process (preview → accept/reject → notification). System requirements for this role include support for autosave in the question editor, schema validation during indicator mapping, audit logging for all validation actions, and version rollback capability for resources.

For accessibility and security the dashboard must provide alternative text for charts, keyboard navigation for filters, and strict access control. Useful KPIs include average validation time, student response rate, and number of active assessments per academic year.

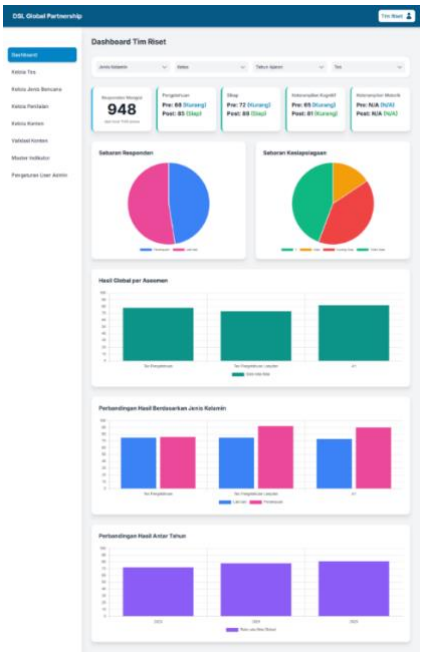


**Figure 3:** Research Team Dashboard User Interface

Figure 3 shows the Research Team user interface on the Dashboard page. Overall, the interface must facilitate test creation, trialing, and activation, while also providing data export and validation-action logging functions.

2) The School Admin or Administrative Unit manages school administrative data such as students, teachers, classes, and academic years, and performs bulk data imports. The dashboard emphasizes concise statistics and an XLSX import module with preview and rollback features to reduce data-entry risk. Short audit features and access to reports ease monitoring and data recovery when needed. Practically, the TU Admin module provides KPI summaries (counts of active students/teachers/classes), an XLSX import panel (upload → schema validation → column mapping preview → commit/rollback), a recent-change table, and report generation tools.

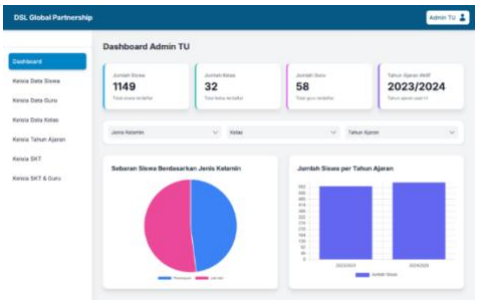


**Figure 4:** School Admin Dashboard User Interface

Figure 4 shows key transactions that include batch import processes that write to the students, teachers, and classes tables and record jobs in import_jobs / import_errors, plus CRUD operations with bulk-action options. Error handling must report problematic rows, limit file size, and provide snapshot/transaction_id for rollback. Security measures include access restrictions, file checksum logging, and input sanitization to prevent form injection. Recommended metrics include import success rate, import processing time, and number of manual corrections after import.

3) Teachers use the dashboard to monitor class progress and perform student observations. The primary view contains class lists, per-student observation status, and quick access to create assignments or upload teaching resources. The UI should present a streamlined flow for completing observations (checkboxes or short scales), saving drafts, and submitting results.
Operationally, the teacher dashboard includes class / academic-year listings, class progress summaries, a student table with observation status, quick buttons to create assignments/upload materials, and a notification area for assignments and assessment results. Important flows include assignment distribution (select class → template → schedule → send), filling observations with Save Draft and Submit Final options, and resource upload that enters resource_submissions for moderation. For user experience, features should include autosave, upload progress indicators, and schedule-conflict detection; on the data side, observations are stored in observations / draft_observations and assignments in class_assignments. Accessibility considerations include clear labels, large touch targets, and high contrast. Relevant KPIs are assignment submission rate, class average score, and percentage of completed observations.

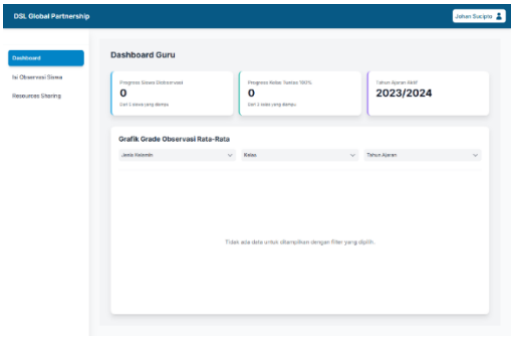


**Figure 5:** Teacher Dashboard

Figure 5 depicts a compact, functional Teacher dashboard. The left panel contains navigation (Dashboard, Fill Student Observations, Resource Sharing); the top area shows summary cards (Student Observation Progress, Class Completion Progress, Active Academic Year). The main area contains an Average Observation Grade chart with filter controls (Gender, Class, Academic Year); placeholders appear when data are unavailable. The dashboard is designed to help teachers monitor class progress, quickly fill observations, and access related learning resources.

4) The student dashboard focuses on tasks and assessments to be completed and individual results. Students can open My Tests to view assigned Pre/Post tests, complete questions, and see result summaries after submission. Reliable autosave and instant feedback (score/comments) improve the student experience. Specifically, the My Tests panel shows active tests, schedule, timer, autosave of answers, and submit options; the results area displays personal summaries and teacher comments. Core flows include starting a test (periodic autosave → timer → submit → summary), uploading work to student_portfolio, and receiving notifications. The system must handle offline conditions with local storage and synchronization when online, provide submit confirmations, and protect against double submissions; data are stored in assessment_submissions and temp_answers. For privacy and security, students can view only their personal results unless otherwise permitted. Primary KPIs include completion rate, mean score, and autosave success rate.

5) Guest functionality includes registration via the Home modal (creating entries in guest_registrations with status pending), review by the Research Team, and account activation upon approval. Guests may create discussion threads that, if necessary, go into moderation_queue before publication; messages and statuses are stored in guest_messages with a moderation_status flag. Error handling encompasses duplicate email detection, message length limits, and anti-spam measures; email verification and logging of privacy/GPS consent are required for data security. Operational metrics to monitor include approval rate, average validation time, and number of public interactions per period.

6) The Superadmin provisions accounts (TU Admin, Research Team), monitors the system, and may preview other role dashboards for debugging or audit purposes. The provisioning panel must record an audit trail when creating accounts or resetting credentials.

## 3. Conclusion

The DSLGP user interface design employs a standardized, consistent, and role-oriented structure. This approach ensures that all modules, from the Home page and per-role dashboards to critical transactions such as Guest registration, resource validation, and assessment distribution, are accessible and intelligible to users with varying levels of digital literacy. Applying user-centered design principles and SRS-based requirements mapping produced clear, low-redundancy workflows that effectively support disaster education.

Role-segregated dashboards create a more focused user experience, reduce UI complexity, and increase data-entry accuracy by teachers, students, and the research team. Integration of essential features—dynamic location permission handling, bidirectional validation, bulk data import, and analytics panels, strengthens DSLGP's role as a data-driven educational platform.

Overall, the DSLGP UI design not only improves interaction quality but also supports consistent implementation, system scalability, and maintainability, making it an appropriate model for Web-GIS-based e-learning systems going forward.

## References

[1] B. Handoyo, A. K. Putra, R. L. Mkumbachi, and A. N. Fadila Gastama, "Students' geo-capabilities skills improvement: The role of differentiated spatial project learning development using QGIS," *Jurnal Pendidikan Geografi: Kajian, Teori, dan Praktek dalam Bidang Pendidikan dan Ilmu Geografi*, vol. 30, no. 1, pp. 163–177, Jan. 2025, doi: 10.17977/2527-628X.1184.

[2] S. S, A. Shrivastava, and S. Karmakar, "Identifying interface design factors impacting user experience in digital learning platforms-A pilot study," *Social Sciences & Humanities Open*, vol. 11, p. 101662, 2025, doi: 10.1016/j.ssaho.2025.101662.

[3] B. Handoyo, S. Rizal, A. N. F. Gastama, M. Susilowati, and L. A. C. Prayogo, *Disaster Spatial Learning Pendekatan Terpadu untuk Mitigasi Bencana*. Kota Malang: Unpublished, 2025.

[4] B. Handoyo, A. K. Putra, R. L. Mkumbachi, and A. N. Fadila Gastama, "Students' geo-capabilities skills improvement: The role of differentiated spatial project learning development using QGIS," *Jurnal Pendidikan Geografi: Kajian, Teori, dan Praktek dalam Bidang Pendidikan dan Ilmu Geografi*, vol. 30, no. 1, pp. 163–177, Jan. 2025, doi: 10.17977/2527-628X.1184.

[5] M. R. R. Ramesh and Ch. S. Reddy, "Metrics for software requirements specification quality quantification," *Computers & Electrical Engineering*, vol. 96, p. 107445, Dec. 2021, doi: 10.1016/j.compeleceng.2021.107445.

[6] B. Handoyo, H. Soekamto, A. K. Putra, P. A. Kamil, and F. Wulandari, "Disaster preparedness: The role of spatial disaster learning using geospatial technology," *Jàmbá: Journal of Disaster Risk Studies*, vol. 16, no. 1, Nov. 2024, doi: 10.4102/jamba.v16i1.1576.

[7] M. Susilowati, “Proposal: Software Requirement Specification pada Disaster spatial learning bagi Siswa SMA Daerah Bencana di Kabupaten Malang,” Kabupaten Malang, 2025.
[8] T. Jokela, N. Iivari, J. Matero, and M. Karukka, “The standard of user-centered design and the standard definition of usability,” in *Proceedings of the Latin American conference on Human-computer interaction*, New York, NY, USA: ACM, Aug. 2003, pp. 53–60. doi: 10.1145/944519.944525.
[9] B. W. Boehm, “A spiral model of software development and enhancement,” *Computer (Long Beach Calif)*, vol. 21, no. 5, pp. 61–72, May 1988, doi: 10.1109/2.59.
[10] E. Bjarnason, F. Lang, and A. Mjöberg, “An empirically based model of software prototyping: a mapping study and a multi-case study,” *Empir Softw Eng*, vol. 28, no. 5, p. 115, Sep. 2023, doi: 10.1007/s10664-023-10331-w.