



**DALHOUSIE
UNIVERSITY**

CSCI 5308-Adv Topics in Software Develop

G02-Dalhousie Stack Overflow System

System Architecture

Table of Content

System Overview.....	2
Tech Stack.....	2
Software (Web-only)	2
Hardware & Internet	2
Architecture	3
System Layering and Code Organization.....	3
Security Model.....	3
Key Data Flows.....	4
Deployment Topology and Scaling Strategy.....	4
Risks and Countermeasures	4
Project Structure	5
Reference.....	8

System Overview

We aim to build a Q&A platform for Dalhousie students, which is designed similar to Stack Overflow. We used a Modular Monolith + Multi-instance Horizontal Scaling as the architecture style. Students can post any questions (academic or non-academic) on this platform anonymously and they can get answers from their peers or teachers. In our system, students can comment, upvotes, tagging, upload images, and the system is designed with safety features.

Tech Stack

Software (Web-only)

- Application Frame: Spring Boot 3 (Java 17+) -- Production-ready applications that can be deployed directly
- UI: Vaadin -- rapidly build enterprise-grade user interfaces.
- Session/Horizontal Scaling: Spring Session + Redis
- Database: PostgreSQL
- Search: Elasticsearch -- keeping individual shards between 10–50GB for capacity planning.
- Object Storage: MinIO (S3-compatible)
- Monitoring: Spring Boot Actuator + Prometheus + Grafana
- Single Sign-On(SSO): OpenID Connect (Authorization Code + PKCE) + Keycloak -- SAML protocol is widely adopted for SSO in universities and enterprises.
- Content Moderation and Filtering: ICU4J + Aho–Corasick + Regular Expression
- Layering

Hardware & Internet

- HAProxy + VIP (Keepalived) -- HTTP load balancing, health checks, and HA methods.
- Caching: Redis
- Scalability: Web + Redis + SQL + Elasticsearch + HAProxy stack

Architecture

- Modular Monolith + Multi-instance Horizontal Scaling, with external Redis / PostgreSQL / Elasticsearch.
- Search Sharding & Capacity Planning -- typically limit single shards to 10–50 GB (or max ~2e8 documents)
- Session Consistency: Spring Session + Redis
- Object Storage Redundancy: MinIO Erasure Coding (ECC)

System Layering and Code Organization

- Layering (From the Top to the Bottom):
- Web: Vaadin + REST, handling I/O mapping and use case facades
- Domain: Entity models + Port interfaces
- Adapters: JPA (PostgreSQL) / Redis / ES / MinIO / OIDC (Keycloak) / observability
- Boot: Spring Boot
-

Security Model

- Authentication: OIDC (Authorization Code + PKCE) via Keycloak; Website uses HttpOnly+Secure+SameSite
- Authorization: Site roles student/staff/instructor/moderator/admin; configurable “tag moderators”.
- Anonymity: Anonymous display for students; admins can view real identities with audit trails.
- Input Protection: Malicious term filtering (ICU4J normalization + Aho–Corasick lexicon + regex), character limits (frontend prompts + @Size + DB CHECK/varchar(N)), Redis rate limiting.
- Uploads: MinIO private buckets, direct uploads via pre-signed URLs, MIME type whitelist and size limits
- CSRF/CORS: Cookie-based session CSRF protection enabled; pure API uses Bearer authentication with CSRF disabled.

Key Data Flows

- Query: Vaadin form → Server validation → PostgreSQL → Asynchronous write to ES index → Notification
- Search: Query → ES; Auto-complete → Return results.
- Upload: Request URL from backend → Client PUT to MinIO → Write object key to post.
- Governance: Trigger Level A block; Level B requires editing or queuing for review; Full audit trail of review process.

Deployment Topology and Scaling Strategy

- Initial (MVP) Topology:
- LB: HAProxy/NGINX ×2 + VIP (Keepalived), health checks.
- Application: Spring Boot 2–3 instances (stateless, horizontally scalable).
- Redis: 3 nodes (master-slave + Sentinel or Cluster).
- PostgreSQL: Primary database + 1 secondary (optional read-only).
- Elasticsearch: 3 nodes (master role + data), install smartcn, mount synonyms.
- MinIO: 1–2 nodes (scalable to 4–6 nodes later with 6+2 erasure coding).
- SSO: Keycloak ×2 high availability, integrated with campus IdP.
- Monitoring: Prometheus + Grafana; (optional) elasticsearch_exporter.

Risks and Countermeasures

Risks	Countermeasures
• Complex ES Configuration	<ul style="list-style-type: none">• Template-based indexing & synonyms; pre-release rebuild validation
• Governance False Positives/Negatives	<ul style="list-style-type: none">• Review queue + metric monitoring; versioned glossaries/whitelists
• Anonymous Abuse	<ul style="list-style-type: none">• Stricter throttling, private message ban, traceable disclosure (auditing)
• Data Growth	<ul style="list-style-type: none">• Sharding per 10–50GB; ILM for logs/auditing; periodic snapshot rotation

Project Structure

```
dal-qa/
├── README.md
├── LICENSE
└── .gitignore
└── .editorconfig
└── .gitattributes
└── .env.example
└── .github/
    └── workflows/
        └── ci.yml
        #Spring.Boot.3 (Java.17+) Modular.Monolith
└── backend/.....
    ├── dalqa-parent/
    │   └── pom.xml
    ├── dalqa-common/ #.General: Exceptions, DTOs, Pagination, Result
    │   └── src/main/java/.....
    ├── dalqa-domain/ #.Domain.Model.&.Port (.Moderation/Policy.Port)
    │   └── src/main/java/.....
    ├── dalqa-app/ #.UseCases: Register.Global.Interceptors: Rate.Limiting./.
    │   └── src/main/java/.....
    #Content.Governance./Length.Limiting.....
    ├── dalqa-web/ #.REST/
    │   └── src/main/java/.....
    ├── dalqa-persistence-jpa/
    │   └── src/main/resources/db/migration/
    │       ├── V1_baseline.sql
    │       ├── V2_fts.sql #.Postgres.FTS.....
    │       └── V3_content_limits.sql #.DB-level.length.constraints.&
    │           #.CHECK. (maximum.length.for.title/body)...
    #.dalqa-redis/#.Session/Rate.limiting/counting; publish.rate.limiting.metrics
    └── src/main/java/.....
    ├── dalqa-search-fts/
    │   └── src/main/java/.....
    ├── dalqa-search-es/ #Elasticsearch
    │   └── src/main/java/.....
    ├── dalqa-auth-oidc/#.Keycloak/OIDC.+ Site-wide.RBAC
    │   └── src/main/java/.....
    ├── dalqa-storage-minio/ #Pre-signed.upload/whitelist.policy.
    │   #is.reused.by.content.governance.
    │   └── src/main/java/.....
    ├── dalqa-observability/
    │   └── src/main/resources/logback-spring.xml
    ├── dalqa-boot/ #.Exposure.Limit.Length.Threshold/Switch. (Configuration.Item)
    │   └── src/main/resources/application.yml
    #.Rule-Based.Content.Governance
    ├── dalqa-moderation/.....
    │   ├── src/main/java/...
    │   │   ├── pipeline/ #.Standardization->.Matching->.Rule.Evaluation.Pipeline.....
    │   │   │   ├── normalizer/ #.ICU4J/Java.Normalizer
    │   │   │   ├── matcher/ #.Aho-Corasick.matching.+ regular.expression.templates
    │   │   │   ├── actions/ #.Block/Occlusion/Queue.for.Review/Alert
    │   │   │   └── metrics/ #.Export.Metrics. (Hits/False Positives/Latency)
    │   │   └── src/main/resources/wordlists/ #.Wordlists. (Supports.hot-reloading)
    │   │       ├── blocklist.txt #.Blacklist. (multilingual)
    │   │       ├── allowlist.txt #.Whitelist. (academic/course.exceptions)
    │   │       └── patterns.yaml #.Regular.expressions/templates.
    │   └── .....
    #(contact.info/external.links/spam)
```

```

...   .dalqa-content-policy/.#.Word.Limit/Publishing.Policy.Module
...     .src/main/java/...
...       limits/.#.Maximum.character.count.for.titles/body.text,.limits.on.code.blocks/images
...       validators/.#.JSR-380.Bean.Validation.wrappers.(@Size,.etc.)
...       hooks/.#.Entry.hooks:.Pre-submission.validation.for.questions/answers/comments
...
...   .dalqa-ai-common/.#.AI.foundation.(Prompt/Interceptors)
...   .dalqa-ai-assist/.#.AI.assistant:.Conversational.search/Summarization/Polishing.(optional)
...   .dalqa-ai-similarity/.#.Similarity/Duplicate.recall.(Vector)
...   .dalqa-ai-similarity-pgvector/.#.pgvector.adaptation
...   .dalqa-ai-similarity-opensearch/.#.OpenSearch.k-NN.adaptation
...   .dalqa-ai-moderation/.#.AI.secondary.review.(toxicity/exam.leakage,.optional)
...     .src/main/java/...
...
.frontend/.#.React.+.Vite.+.TypeScript
...   package.json./tsconfig.json./vite.config.ts
...   .src/
...     .pages/
...       Home.tsx./Search.tsx./QuestionDetail.tsx
...       AskQuestion.tsx.#.Frontend.character.limit.alerts.&.real-time.highlighting.of.prohibited.words
...       Admin.tsx.#.Review.queue.viewing/approval.(hit.records).
...     .components/
...       MarkdownEditor.tsx.#.Word.counter/progress.bar,,whole-question.posting.prohibition.alerts
...       ContentWarnings.tsx.#.Highlight/suggest.replacements.for.matched.words
...       RateLimitBadge.tsx.#.Rate.limit.warning.(anti-spam)
...     .api/
...       moderation.ts.#.Text.pre-check.API.(Optional)
...       policy.ts.#.Retrieve/Display.Current.Length.Policy
...     .store/
...       policy.store.ts.#.Cache.Length.Thresholds/Switches
...
...   .hooks/
...     useContentPolicy.ts.#.Frontend.length.validation
...     useModerationPreview.ts.#.Preview/alert
...   #.Infrastructure/Monitoring
...   .infra/
...     .docker-compose.yml.#.pg/redis/minio/keycloak/prom/grafana
...     .prometheus.yml
...     .grafana/
...       .dashboards/
...         spring-app.json
...         moderation.json.#.Hit.rate/false.positive.rate/processing.delay/approval.rate
...         rate-limit.json.#.Traffic.throttling.hits/rejections/source.distribution
...         provisioning/datasources/prometheus.yml
...     .keycloak/
...       realm-export.json
...     .k8s/
...       .base/
...         app-deployment.yaml
...         redis-statefulset.yaml
...         ingress.yaml
...         overlays/{dev,stg,prod}/
...       .scripts/
...         db-backup.sh
...         db-restore.sh
...         wordlists-sync.sh.#.(NEW).Word.list.synchronization/hot.update.script
...
...   .docs/
...     .architecture/
...       system-overview.md
...     .diagrams/.#.Pre-release.governance.sequence.diagram

```

```
|- api/openapi.yaml
|- operations/
|   |- runbook.md
|   |- review-queue.md #. Review.queue.flow.and.SLO
|   |- monitoring-guide.md #. Add.governance/throttling.metric.descriptions.
`- security/
   |- content-policy.md #. Character.Limit.Policy.(Title/Body/Code/Images)
   |- moderation-rules.md #. Rule-based.lexicon/regular.expressions/exemption.list
   |- upload-policy.md #. Coordinates.with.whitelist.(whole-question.ban).(Enhanced)
   |- audit-logging.md #. Log.hits/passes/appeal.chains.(Enhanced)
`- capacity/
   |- sizing-guide.md #. Post-enforcement.CPU.reservation.recommendations.(+5-10%)
`- ai/
   |- ai-safety-policy.md #. AI.boundaries.for.secondary.verification.only.(Enhanced)

.tests/
  |- e2e/
    |- moderation.spec.ts #. Path.for.blocking/warning/appealing.inappropriate.words
    |- length-limits.spec.ts #. Dual.frontend.and.backend.length.limits.
  `-. datasets/
```

Reference

Spring boot :: Spring boot. (n.d.).

<https://docs.spring.io/spring-boot/>

Vaadin Flow documentation | Comprehensive guide for developers. (n.d.).

<https://vaadin.com/docs/latest/>

Spring session - Spring Boot :: Spring session. (n.d.-b).

<https://docs.spring.io/spring-session/reference/guides/boot-redis.html?>

28.3. *Write-Ahead Logging (WAL)*. (2025, September 25). PostgreSQL

Documentation.

<https://www.postgresql.org/docs/current/wal-intro.html?>

Size your shards | Elastic Docs. (n.d.).

<https://www.elastic.co/docs/deploy-manage/production-guidance/optimize-performance/size-shards?>

Erasure Coding — MinIO Object Storage (AGPLV3). (n.d.).

<https://docs.min.io/community/minio-object-store/operations/concepts/erasure-coding.html?>

Micrometer Prometheus :: Micrometer. (n.d.).

<https://docs.micrometer.io/micrometer/reference/implementations/prometheus.html?>

Configure the Prometheus data source | Grafana documentation. (n.d.). Grafana Labs.

<https://grafana.com/docs/grafana/latest/datasources/prometheus/configure/>

OpenID Foundation. (n.d.). *How OpenID Connect Works - OpenID Foundation.*

OpenID Foundation - Helping People Assert Their Identity Wherever They Choose.

<https://openid.net/developers/how-connect-works/>

Nick Craver - Stack Overflow: The Architecture - 2016 edition. (2016, February 17).

<https://nickcraver.com/blog/2016/02/17/stack-overflow-the-architecture-2016-edition/>

Normalization. (n.d.). ICU Documentation.

<https://unicode-org.github.io/icu/userguide/transforms/normalization/>

Aho-Corasick. (n.d.).

<https://xlinux.nist.gov/dads/HTML/ahoCorasick.html>