Hugbúnaðarverkefni
Final Presentation

Team 1
Ragnar, Óli, Egill and Daníel

Demo

Main use cases:

–UC1: Clock in/Clock out

–UC2: Edit employee shifts

–UC3: Manage tasks

System architecture

Components and communication

- Controllers receive requests from the User
- Information is then sent as a Data Transfer Object to Service
- Service processes the data (like checking validity or interpreting the data)
- Services then send an Entity to the Repository

Components and communication (cont.)

- An Entity is a Java Interpretation of a database row
- The Repository is a JPA repository since we’re using Java Spring
- A response (either another Entity or void) is then sent to Service
- Service turns the Entity into a Data Transfer Object and sends it as a response to the User

Storing and Accessing Data

- Data is stored using a PostgreSQL database
- The database is hosted using Render
- We use FlyWay to migrate the database to our desired schema
- FlyWay is also used to insert dummy data into the database

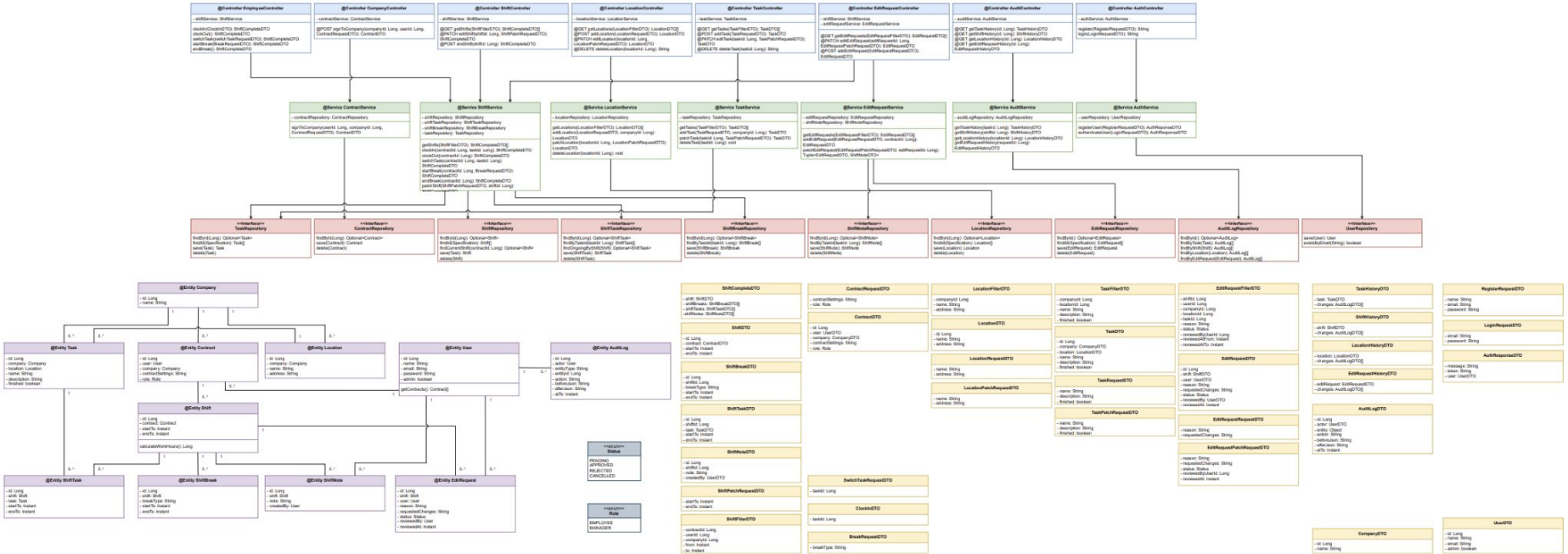
System Highlights

- We chose PostgreSQL + JPA for relational schema and type safety
- We chose Flyway for migrations instead of Spring’s internal schema initialization
- We chose specification based filtering to avoid custom SQL queries per filter

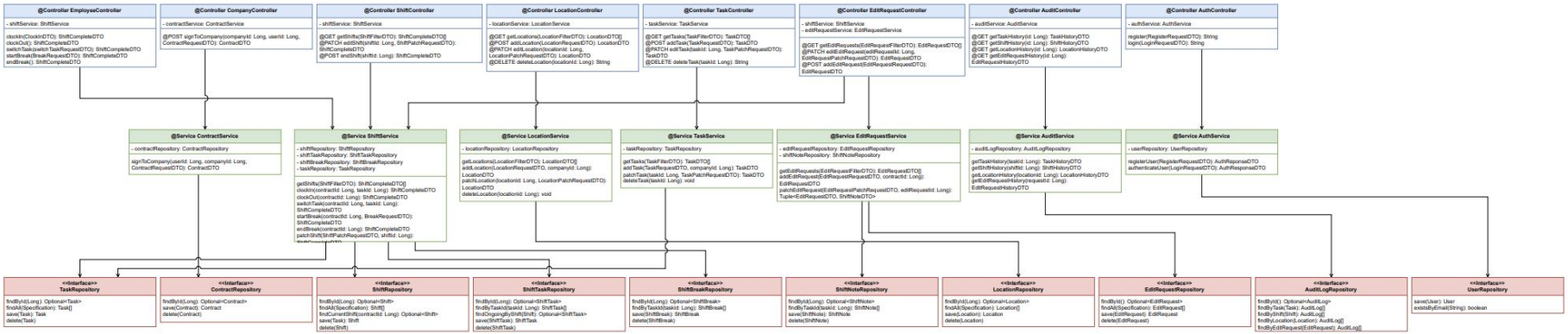
System highlight: Custom filter system

- Additional filter parameters can be specified with the request
- System creates a Specification object from the filter
- Specification automatically creates an SQL command to match the filter requested
- The Specification is created from a custom SpecificationUtils class which resolves the filters

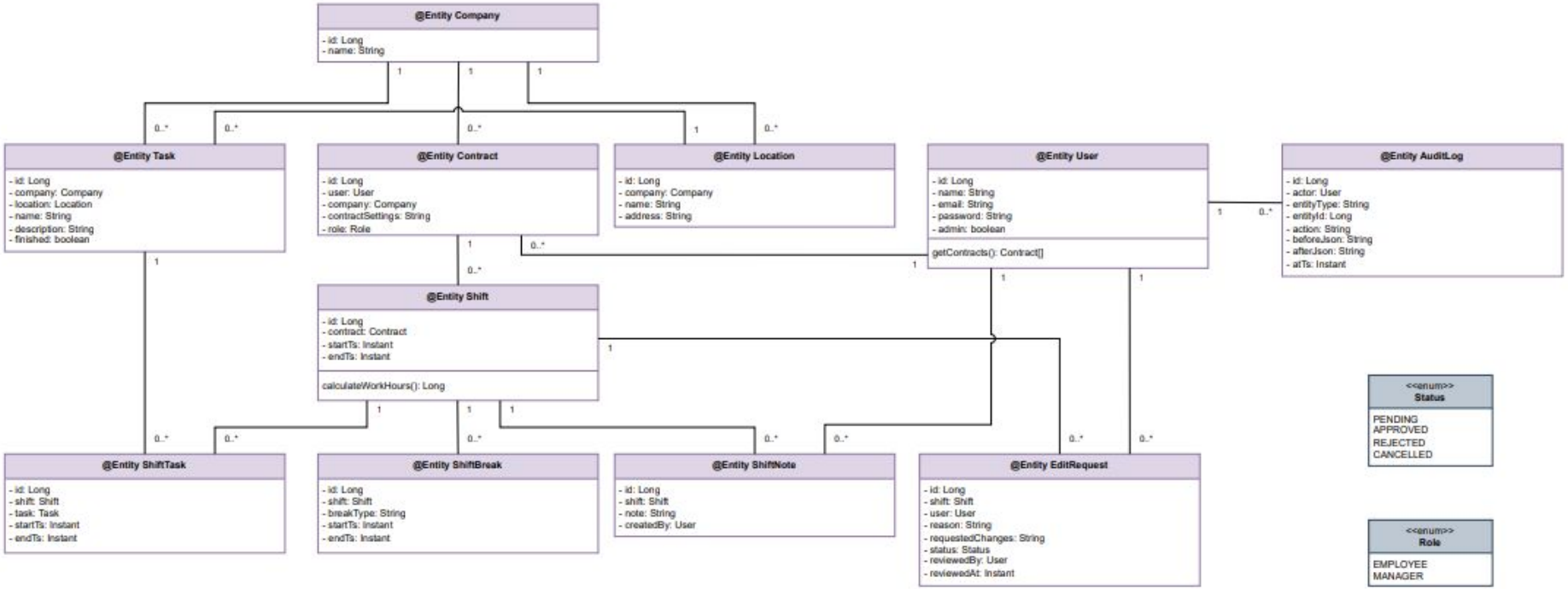
Class Diagram



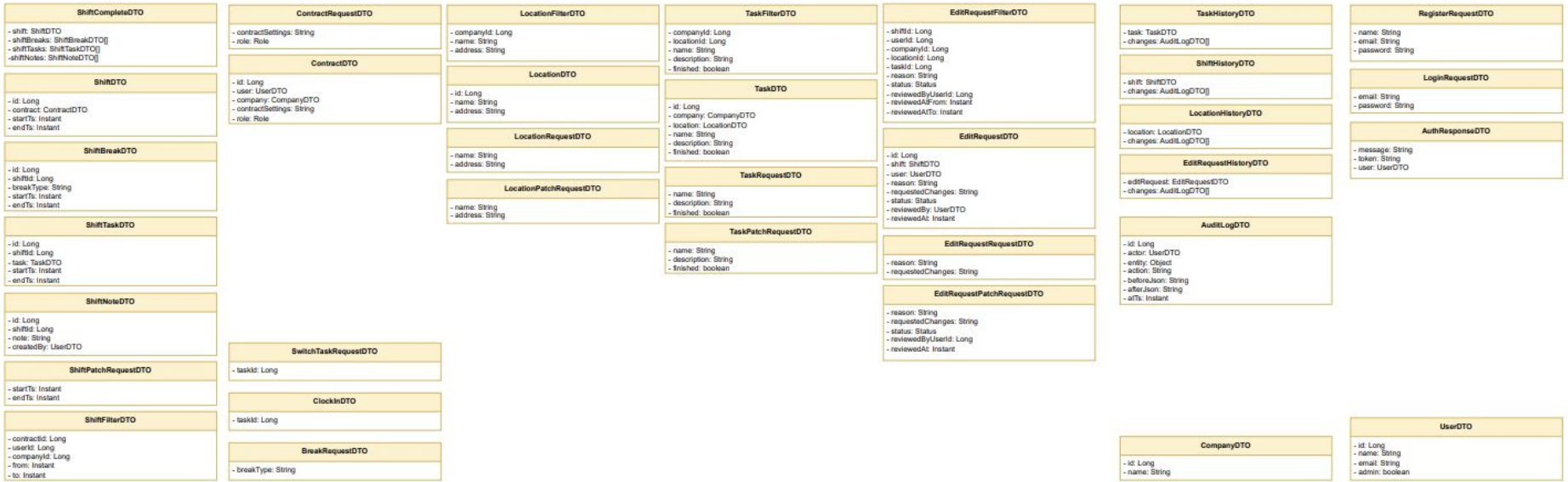
Controller, Service and Repository Diagram



Entity Diagram and Enums



DTO


ShiftCompleteDTO
- shift: ShiftDTO
- shiftBreaks: ShiftBreakDTO[]
- shiftTasks: ShiftTaskDTO[]
-shiftNotes: ShiftNoteDTO[]
ShiftDTO
- id: Long
- contract: ContractDTO
- startTs: Instant
- endTs: Instant
ShiftBreakDTO
- id: Long
- shiftId: Long
- breakType: String
- startTs: Instant
- endTs: Instant
ShiftTaskDTO
- id: Long
- shiftId: Long
- task: TaskDTO
- startTs: Instant
- endTs: Instant
ShiftNoteDTO
- id: Long
- shiftId: Long
- note: String
- createdBy: UserDTO
ShiftPatchRequestDTO
- startTs: Instant
- endTs: Instant
ShiftFilterDTO
- contractId: Long
- userId: Long
- companyId: Long
- from: Instant
- to: Instant
ContractRequestDTO
- contractSettings: String
- role: Role
ContractDTO
- id: Long
- user: UserDTO
- company: CompanyDTO
- contractSettings: String
- role: Role
SwitchTaskRequestDTO
- taskId: Long
ClockInDTO
- taskId: Long
BreakRequestDTO
- breakType: String
LocationFilterDTO
- companyId: Long
- name: String
- address: String
LocationDTO
- id: Long
- name: String
- address: String
LocationRequestDTO
- name: String
- address: String
LocationPatchRequestDTO
- name: String
- address: String
TaskFilterDTO
- companyId: Long
- locationId: Long
- name: String
- description: String
- finished: boolean
TaskDTO
- id: Long
- company: CompanyDTO
- location: LocationDTO
- name: String
- description: String
- finished: boolean
TaskRequestDTO
- name: String
- description: String
- finished: boolean
TaskPatchRequestDTO
- name: String
- description: String
- finished: boolean
EditRequestFilterDTO
- shiftId: Long
- userId: Long
- companyId: Long
- locationId: Long
- taskId: Long
- reason: String
- status: Status
- reviewedByUserId: Long
- reviewedAtFrom: Instant
- reviewedAtTo: Instant
EditRequestDTO
- id: Long
- shift: ShiftDTO
- user: UserDTO
- reason: String
- requestedChanges: String
- status: Status
- reviewedBy: UserDTO
- reviewedAt: Instant
EditRequestRequestDTO
- reason: String
- requestedChanges: String
EditRequestPatchRequestDTO
- reason: String
- requestedChanges: String
- status: Status
- reviewedByUserId: Long
- reviewedAt: Instant
TaskHistoryDTO
- task: TaskDTO
- changes: AuditLogDTO[]
ShiftHistoryDTO
- shift: ShiftDTO
- changes: AuditLogDTO[]
LocationHistoryDTO
- location: LocationDTO
- changes: AuditLogDTO[]
EditRequestHistoryDTO
- editRequest: EditRequestDTO
- changes: AuditLogDTO[]
AuditLogDTO
- id: Long
- actor: UserDTO
- entity: Object
- action: String
- beforeJson: String
- afterJson: String
- atTs: Instant
CompanyDTO
- id: Long
- name: String
RegisterRequestDTO
- name: String
- email: String
- password: String
LoginRequestDTO
- email: String
- password: String
AuthResponseDTO
- message: String
- token: String
- user: UserDTO
UserDTO
- id: Long
- name: String
- email: String
- admin: boolean

Retrospective

What went well? What difficulties did we encounter?

- Implementation went relatively smoothly and we’re satisfied with the final product
- It was difficult to set up a login system with JWT Authentication
- Audit log troubles
- Java Spring too limiting in SQL queries for database initialization => refactor to FlyWay was required

Structure and planning of work

- **Óli** project manager, Structural lead, Technical solutions. **Ragnar** main programmer. **Egill** implemented the SQL query filter and assisted in programming, **Daniel** implemented flyway and the intuitive shift patcher
- Decided early what to do and how to do it, but not *when* to
- Rush to get things done when deadlines were approaching
- 3 / 4’s of the team were moving to Japan for exchange studies, which complicated things

If we could do it again

- Establish weekly meetings + Trello board to avoid last minute rushes
- Start with FlyWay right away instead of internal Java Spring SQL parsing to avoid painful refactoring later
- Better structuring of controller classes
- Clearer Individual responsibilities for the project
- Most importantly, selecting another framework other than Java spring

Thank you for listening!