システム方式設計書

1. プロジェクト名

- 社内ドキュメント・資産・人材管理ができる社内インフラシステム

2. アーキテクチャ概要

- ウェブベースのアプリケーション

- クライアント、サーバー、データベースの3階層アーキテクチャ構造

- プレゼンテーション階層 : JQuery, Ajaxを使用してユーザーインターフェースを構成

- ビジネスロジック階層 : Springを使用して主要ロジックを処理

- データ階層 : PostgreSQLを使用して権限、ユーザー、メニュー、会社情報を保存·

管理

3. 技術スタック

- 開発言語 : Java1.8, html, css, javascript

- フレームワーク及びライブラリー : JQuery, Ajax(フロントエンド),

Spring(バックエンド)

- データベース : PostgreSQL

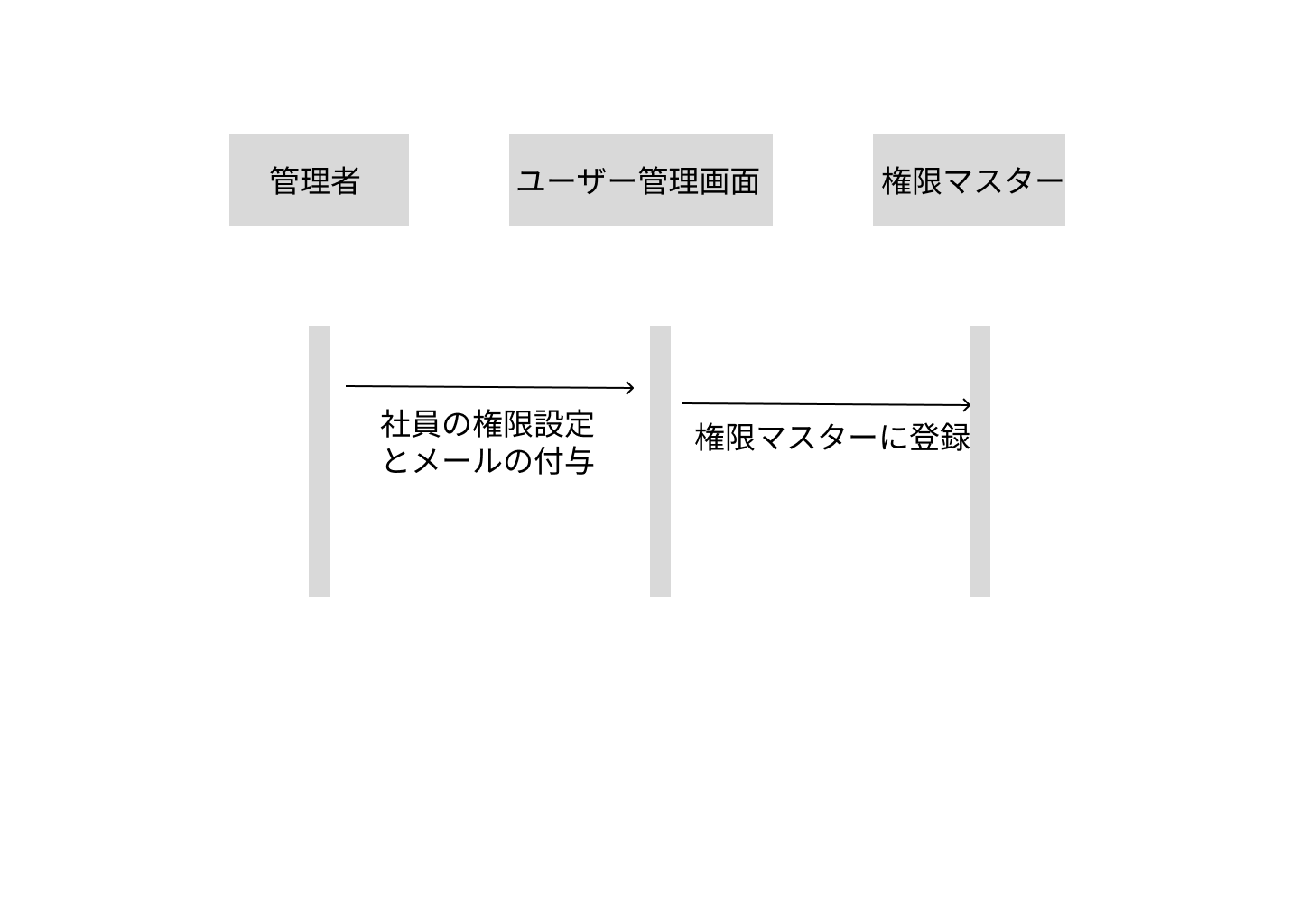
- サーバー : EC2インスタンスでTomcatをインストールして配布

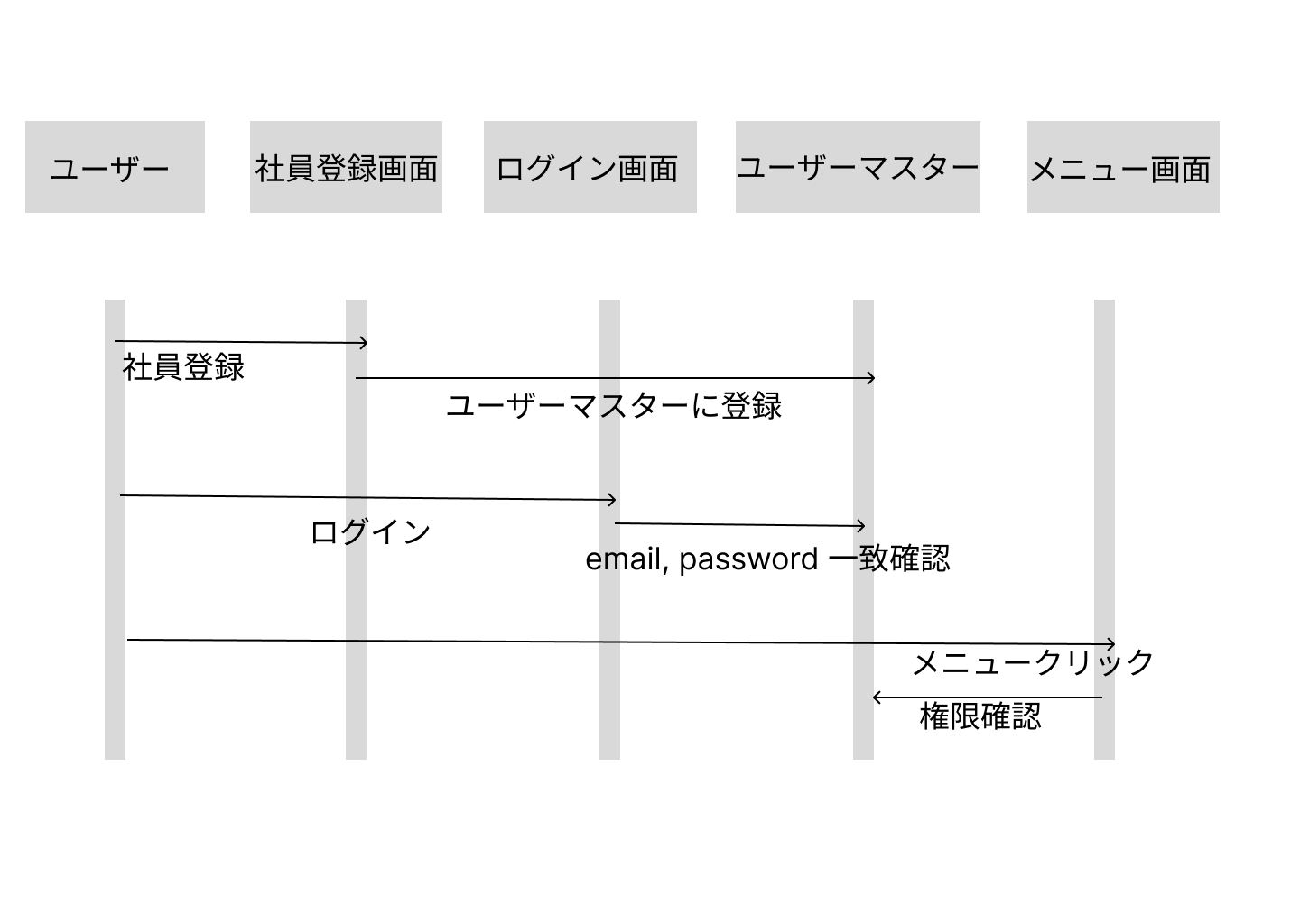
- 配布環境 : AWS EC2

- CI/CD : GitHub Actions

4. インターフェース設計

- SequenceDiagram





5. セキュリティ設計

- 重要なデータは、スプリング セキュリティを使用して暗号化し、保存

- すべてのユーザーはシステムアクセス時に認証が必要であり、JWTトークンを使用し てセッションの有効性を検査

· spring security

メリット

- @PreAuthorize、@Securedのようなアノテーションを使用してコードレベルで権限 を制御可能

- BCryptなど安全なパスワードハッシュアルゴリズムを使用してパスワ ード

セキュリティを強化

デメリット

- spring securityの設定がやや複雑

- テストコード作成時、認証·認可設定による追加コードが必要

· JWT

メリット

- 認証要請を処理するたびにDB照会なしにトークンだけで必要な情報を確認すること ができ、性能向上

- 必要時に敏感な情報を暗号化して転送可能

- サーバーのリソース節約が可能

デメリット

- 有効期限が切れるまでは、ユーザーがログアウトしてもサーバーから直ちに無効に

することが難しい

- クライアントにJWTが保存されるので、該当トークンが奪取される場合、他のユーザ ーがこれを利用することができる。

6. メンテナンス計画

- モニタリング : log4jでエラー状況を記録し、今後の問題についてデバッグ

- バックアップおよび復旧 : 障害発生時にバックアップデータを使って迅速に復旧

できるように手順を用意