Ansvarig: UG Uppgjord: UG Datum: 2023-03-22 Dok.nr: PUSP234206 Version: 1.1

**Högnivådesign**

**(STLDD)**

**ETSF20 Grupp 2**

[**1. Inledning**](#_ah2sxysjpvoo) **2**

[**2. Referens dokument**](#_xm3ksbtn4fxn) **2**

[**3. Översikt**](#_1qbo85t570kp) **2**

[**4. Klassdiagram**](#_qpetuuxm0wus) **4**

[**5. Databas**](#_1e2iypi0notz) **6**

[**6. Information förvarad i sessions**](#_6qllgi5m2h2v) **9**

[**7. Sekvensdiagram**](#_42l3rth9xo0) **10**

[7.1 Inloggning](#_97bczbrxz630) 10

[7.2 Ny användare](#_epxu698nb2fa) 10

[7.3 Borttagning av användare](#_atuzvsrnbwod) 10

[7.4 Tidrapportering](#_35tf8cgdrhuw) 10

[7.5 Skapa projekt](#_nbhbpbinwrzf) 10

[7.6 Ta bort projekt](#_yu1nbv13ure0) 10

[7.7 Uppdatering av användarinformation](#_z608vfgj4aby) 10

# **1. Inledning**

Detta dokument beskriver designen av ett tidrapporteringssystem, vilket är ett system med inloggningsfunktion på en webb-server. Inom systemet finns olika roller som användarna kan anta, vilket är följande: användare, projektledare och admin. En användare kan delta i olika projekt, tidrapportera, se statistik av tidrapportering samt se en profilsida. Projektledaren är en utökad användare och har därmed mer privilegier. Projektledaren kan skapa projekt, lägga in användare i projekten, dela ut roller inom ett projekt. En administratör är en utökad projektledare. Admin har tillgång till en administratörssida där admin ser alla projekt och användare. Admin kan lägga till användare och projekt till systemet men även ge användare roller som projektledare eller admin.

Detta dokument går igenom det material som behövs för att bygga tidrapporteringssystemet. Dokumentet innehåller text, bilder, sekvensdiagram, klassdiagram och SQL-kod som beskriver klasser, funktioner och uppbyggnaden av databasen för detta system.

# **2. Referens dokument**

1. SRS - Software Requirement Specification Version 1.0:

# **3. Översikt**

Systemet är implementerat i en Tomcat (version 10.x) server genom följande klasser och HTML filer:

**class User** Denna klass beskriver en användare i systemet.

**class UsersController** Denna klass används för att kunna hämta, lägga till, ta bort och redigera användare inom systemet. Vilket görs genom en databas. Administratören använder sig av denna klassen.

**class UsersRepository** Denna klass är ett CRUD (Create Read Update Delete) hjälpmedel för att hantera User tabellen i databasen.

**class Role** Denna klass beskriver en roll inom ett projekt, dessa roller skapas till specifika projekt av projektledaren eller av administratören. Rollerna tilldelas senare till användarna av projektet från projektledare eller administratörer.

**class RolesController** Denna klass används för att hämta, skapa, uppdatera och ta bort roller i ett projekt. Detta görs genom en databas. Projektledare samt admin har privilegiet att skapa roller.

**class RolesRepository** Denna klass är ett CRUD (Create Read Update Delete) hjälpmedel för att hantera Role tabellen i databasen.

**class Activity** Denna klass beskriver en aktivitet inom ett projekt.

**class ActivitiesController** Denna klass används för att hämta, skapa, uppdatera och ta bort aktiviteter i ett projekt. Detta görs genom en databas.

**class ActivitiesRepository** Denna klass är ett CRUD (Create Read Update Delete) hjälpmedel för att hantera Activity tabellen i databasen.

**class Project** Denna klass beskriver projekten i systemet som användarna kan ta del i.

**class ProjectsController** Denna klass används för att hämta, skapa, uppdatera och ta bort projekt i systemet. Detta görs igenom en databas. Projektledare och administratörer kan skapa nya projekt i systemet.

**class ProjectsRepository** Denna klass är ett CRUD (Create Read Update Delete) hjälpmedel för att hantera Project tabellen i databasen.

**class Clockin** Denna klass beskriver en tidrapportering för projekten.

**class ClockinsController** Denna klass används för att hämta, skapa, uppdatera och ta bort tidrapporteringar i systemet. Detta görs igenom en databas.

**class ClockinsRepository** Denna klass är ett CRUD (Create Read Update Delete) hjälpmedel för att hantera ClockIn tabellen i databasen.

**class AuthController** Denna klass är en klass för att kontrollerna inloggning, vid inloggning kontrolleras om en användare finns med det angivna användarnamnet samt lösenordet.

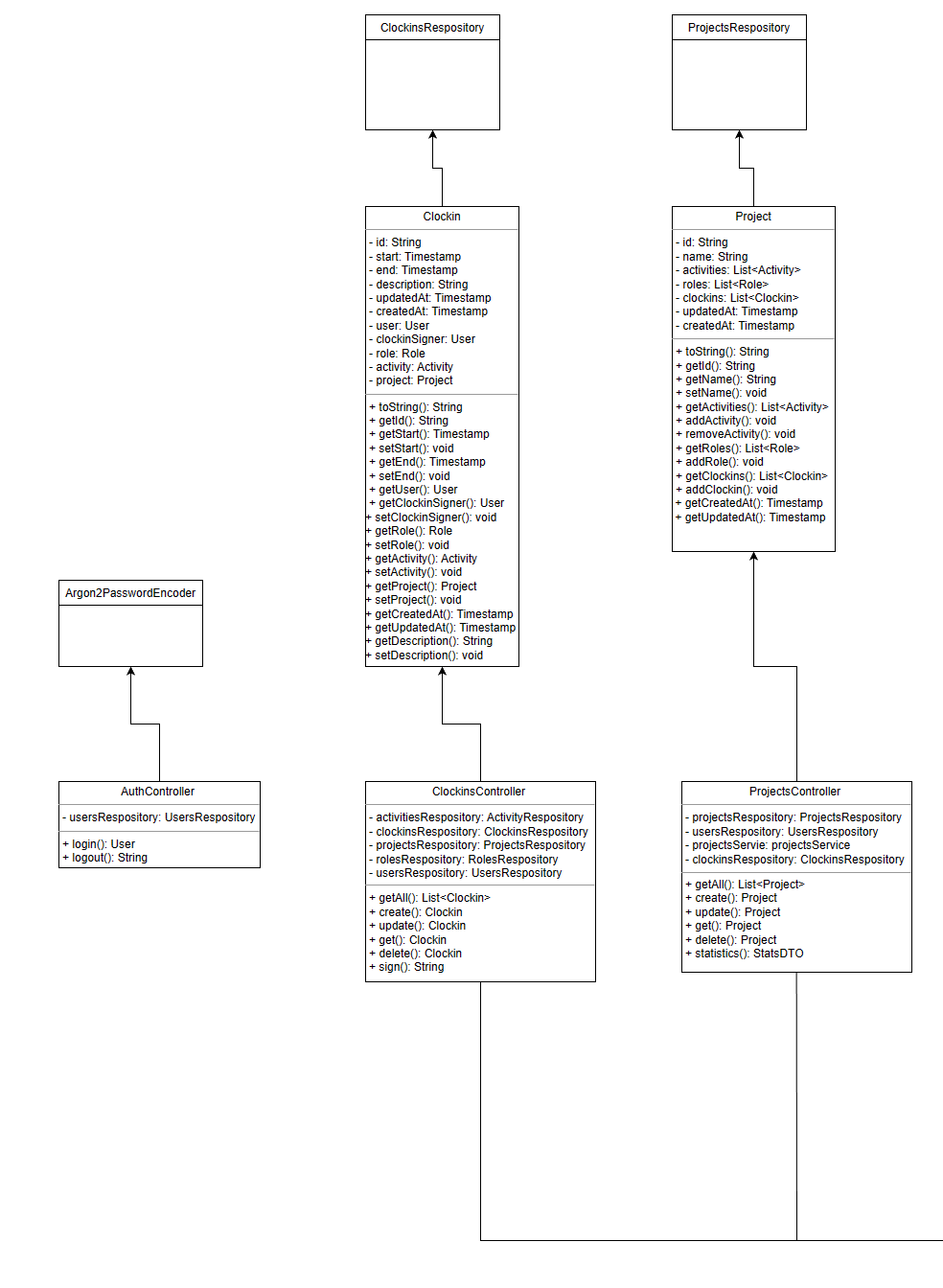
**file index.html** Denna html fil innehåller sidorna för systemet.

**file bundle.js** Denna javascript fil innehåller all funktionalitet för systemet.

**file bundle.css** Denna css fil innehåller designen för systemet.

# **4. Klassdiagram**

Ett klass diagram som visar de viktigaste klasserna i systemet visas i Figur 1 och Figur 2

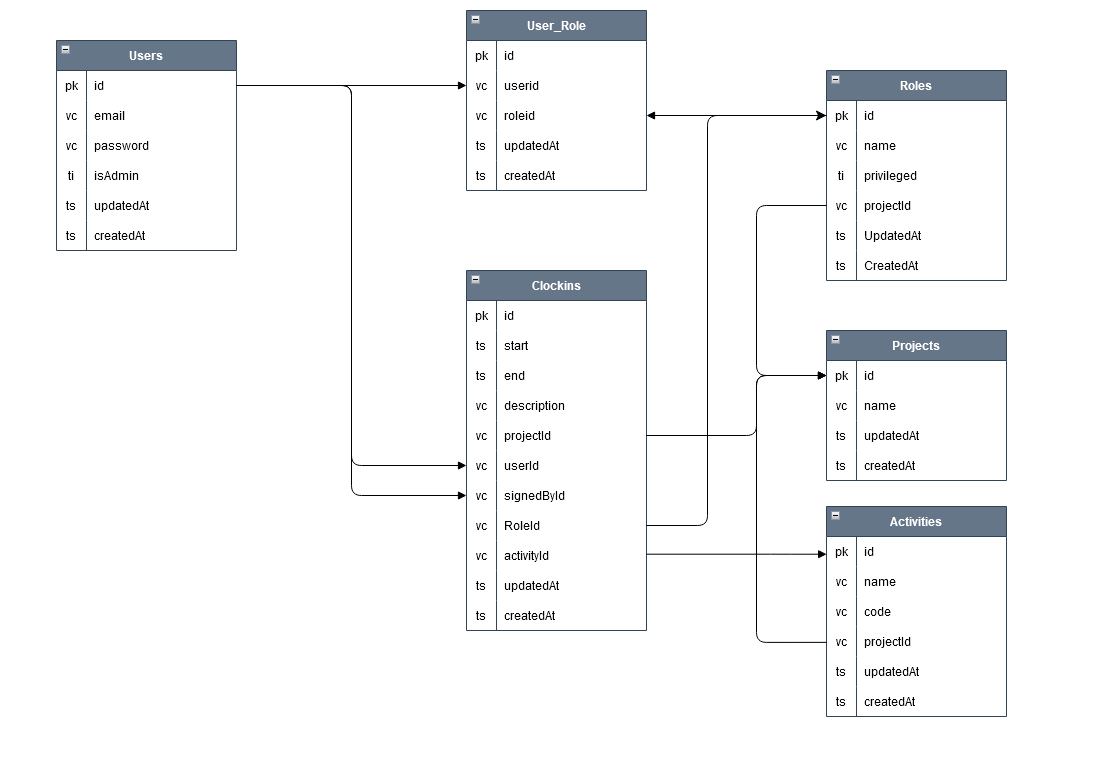
****

Figur 1: Klassdiagram vänster halva

# Figur 2: Klassdiagram höger halva

# **5. Databas**

Databasen definieras enligt Figur 3:



Figur 3: E/R - diagram

Databasen kan konstrueras från grunden med följande syntax enligt Figur 3:

CREATE TABLE Users (

id VARCHAR(50) PRIMARY KEY,

username VARCHAR(20) UNIQUE NOT NULL,

password VARCHAR(20) NOT NULL,

email VARCHAR(50),

name VARCHAR(30),

phoneNumber VARCHAR(12),

isAdmin TINYINT NOT NULL,

updatedAt TIMESTAMP NOT NULL,

createdAt TIMESTAMP NOT NULL

);

CREATE TABLE User\_Role (

id VARCHAR(50) PRIMARY KEY,

userId VARCHAR(50) NOT NULL,

roleId VARCHAR(50) NOT NULL,

updatedAt TIMESTAMP NOT NULL,

createdAt TIMESTAMP NOT NULL,

FOREIGN KEY (userId) REFERENCES Users(id),

FOREIGN KEY (roleId) REFERENCES Roles(id)

);

CREATE TABLE Roles (

id VARCHAR(50) PRIMARY KEY,

name VARCHAR(50) UNIQUE NOT NULL,

privileged TINYINT NOT NULL,

projectId VARCHAR(50) NOT NULL,

updatedAt TIMESTAMP NOT NULL,

createdAt TIMESTAMP NOT NULL,

FOREIGN KEY (projectId) REFERENCES Projects(id)

);

CREATE TABLE Clockins (

id VARCHAR(50) PRIMARY KEY,

start TIMESTAMP NOT NULL,

end TIMESTAMP NOT NULL,

description VARCHAR(200) NOT NULL,

projectId VARCHAR(50) NOT NULL,

userId VARCHAR(50) NOT NULL,

updatedAt TIMESTAMP NOT NULL,

clockinSigner VARCHAR(50),

createdAt TIMESTAMP NOT NULL,

FOREIGN KEY (projectId) REFERENCES Projects(id),

FOREIGN KEY (clockinSigner) REFERENCES Users(id)

);

CREATE TABLE Projects (

id VARCHAR(50) PRIMARY KEY,

name VARCHAR(100) UNIQUE NOT NULL,

updatedAt TIMESTAMP NOT NULL.

createdAt TIMESTAMP NOT NULL

);

CREATE TABLE Activities (

id VARCHAR(50) PRIMARY KEY,

name VARCHAR(50) UNIQUE NOT NULL,

code INTEGER UNIQUE NOT NULL,

projectId VARCHAR(50) NOT NULL,

updatedAt TIMESTAMP NOT NULL,

createdAt TIMESTAMP NOT NULL,

FOREIGN KEY (projectId) REFERENCES Projects(id)

);

INSERT INTO Users(id, username, password, isAdmin, updatedAt, createdAt)

VALUES ("123e4567-e89b-12d3-a456-426614174000", "Admin", "AdminPass", 1, "2023-02-21 13:57:40, "2023-02-01 13:57:40");

# **6. Information förvarad i sessions**

Under varje session förvaras följande attribut:

**String userId:** Unik identifierare för varje användare, tex. ‘asdadsad-kasi-9789-dsa5-dsaefsdagfds‘

**boolean isAdmin:** kollar om användare är admin. Följande stadier är definierade:

**true:** användare är admin

**false:** användare är inte admin

# **7. Sekvensdiagram**

## **7.1 Inloggning**

Figur 4 visar hur en lyckad inloggning hanteras. Figur 5 visar hur en misslyckad inloggning hanteras.

## **7.2 Ny användare**

Figur 6 visar ett lyckat försök till att skapa en ny användare.

## **7.3 Borttagning av användare**

Figur 7 visar ett lyckat försök till att ta bort en användare. Figur 8 visar ett misslyckat försök till att ta bort en användare.

## **7.4 Tidrapportering**

Figur 9 visar ett lyckat försök vid en tidrapportering. Figur 10 visar ett misslyckat försök vid en tidrapportering.

Figur 17 visar en lyckad signering av en tidrapport. Figur 18 visar en misslyckad signering av en tidrapport.

## **7.5 Skapa projekt**

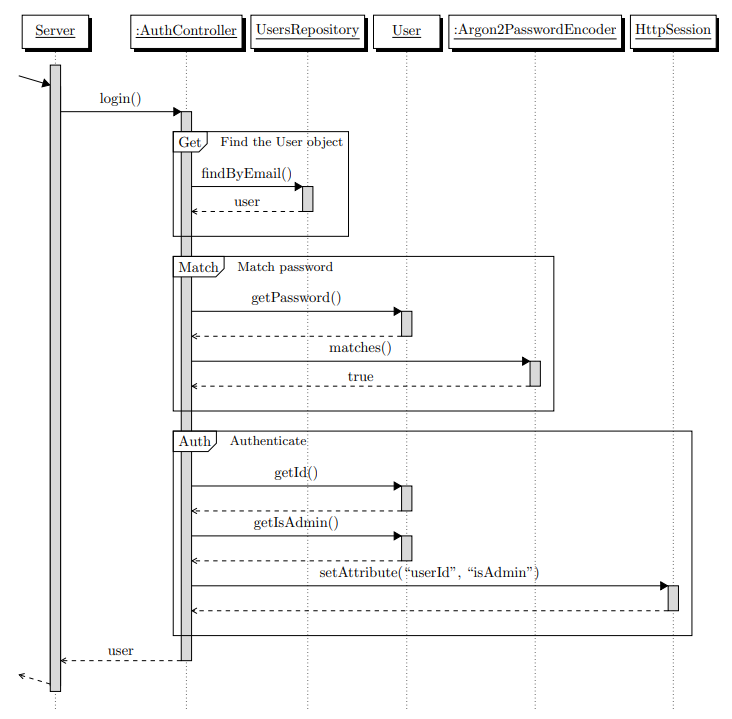
Figur 11 visar ett lyckat försök till att skapa ett projekt. Figur 12 visar ett misslyckat försök till att skapa ett projekt.

## **7.6 Ta bort projekt**

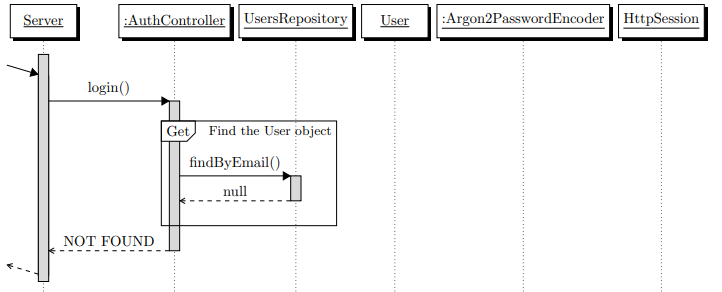
Figur 13 visar ett lyckat försök till att ta bort ett projekt. Figur 14 visar ett misslyckat försök till att ta bort ett projekt.

## **7.7 Uppdatering av användarinformation**

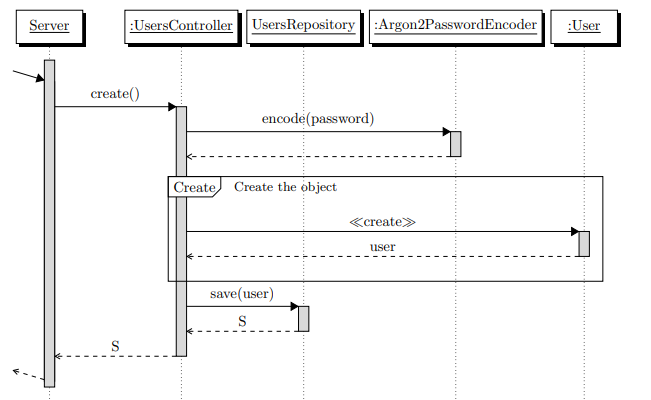
Figur 15 visar ett lyckat försök att uppdatera information för en användare. Figur 16 visar ett misslyckat försök att uppdatera information för en användare.



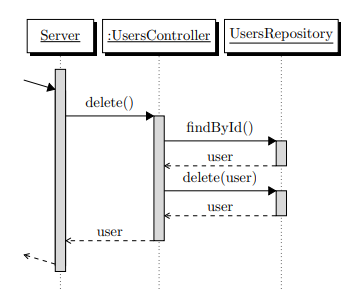
Figur 4: Sekvensdiagram som beskriver processen av en lyckad inloggning



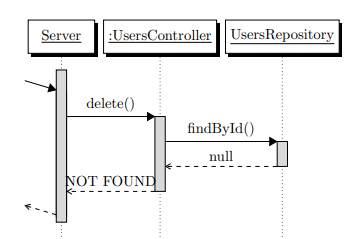
Figur 5: Sekvensdiagram som beskriver processen av en misslyckad inloggning



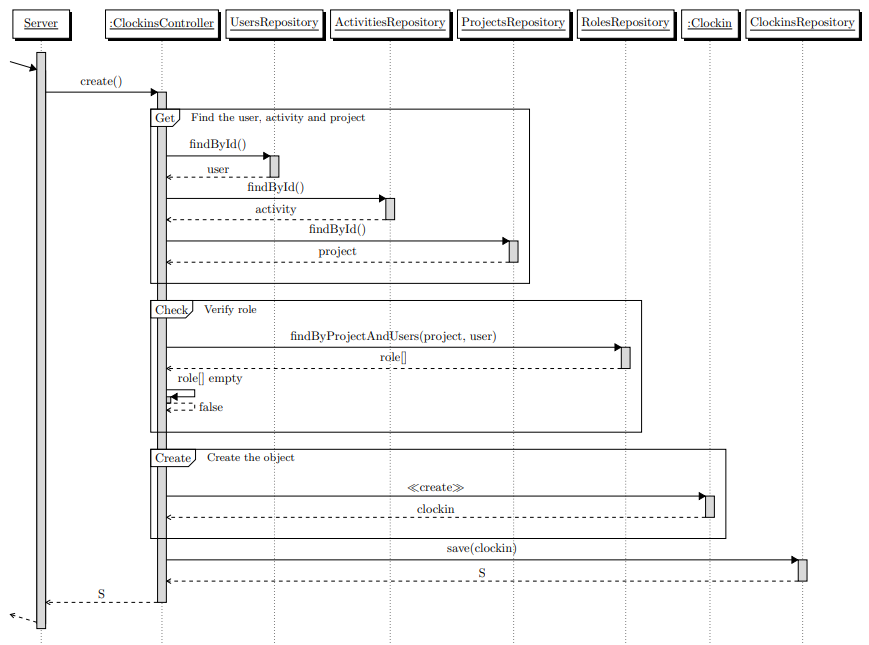
Figur 6: Sekvensdiagram som beskriver en lyckad process av att skapa en ny användare till systemet



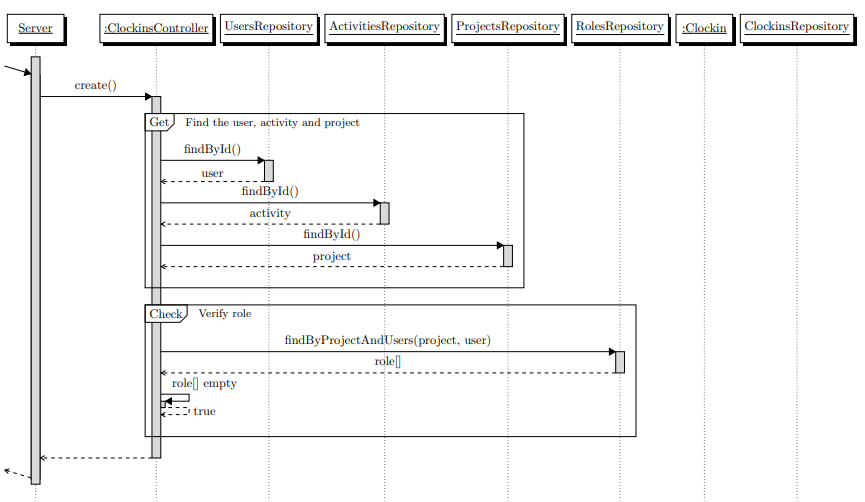
Figur 7: Sekvensdiagram som beskriver en lyckad process att ta bort en användare från systemet



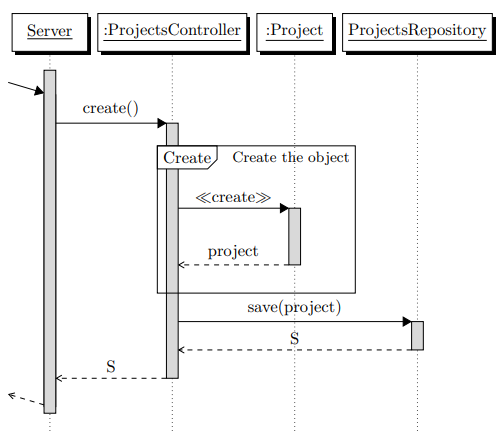
Figur 8: Sekvensdiagram som beskriver en misslyckad process att ta bort en användare från systemet



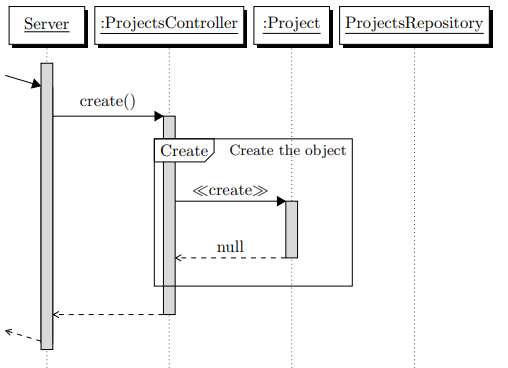
Figur 9: Sekvensdiagram som beskriver när en användare lägger till en tidrapport och processen lyckas



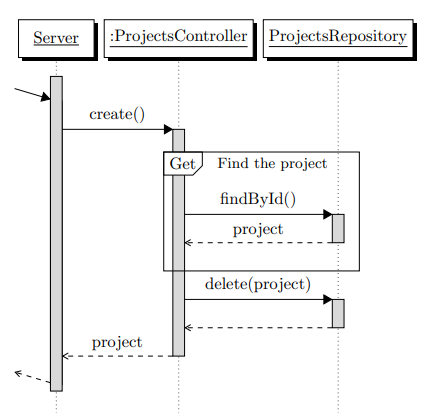
Figur 10: Sekvensdiagram som beskriver när en användare lägger till en tidrapport och processen misslyckas



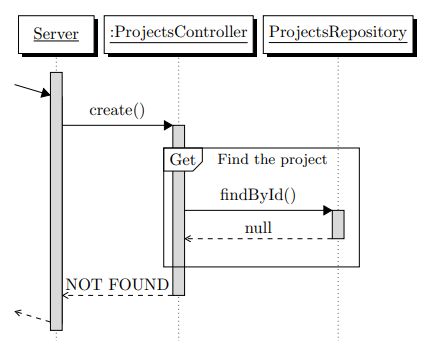
Figur 11: Sekvensdiagram som beskriver när en projektledare eller admin skapar ett projekt och processen lyckas



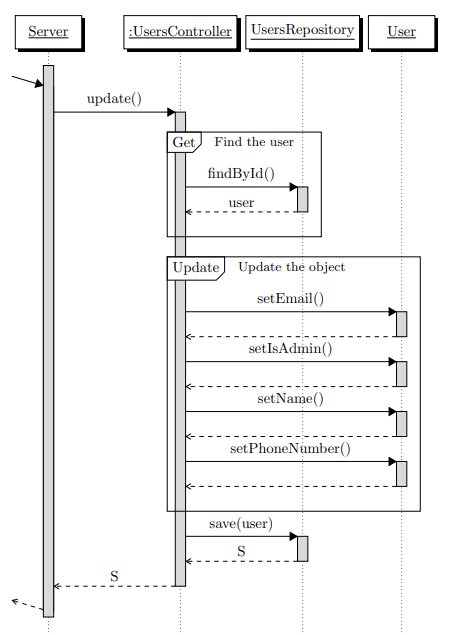
Figur 12: Sekvensdiagram som beskriver när en projektledare eller admin skapar ett projekt och processen misslyckas



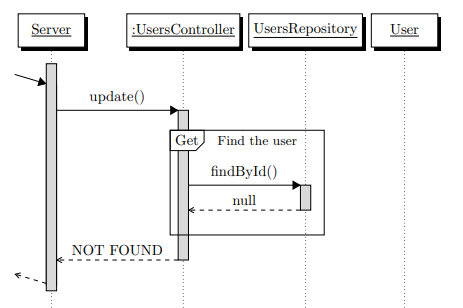
Figur 13: Sekvensdiagram som beskriver när en projektledare eller admin tar bort ett projekt och processen lyckas



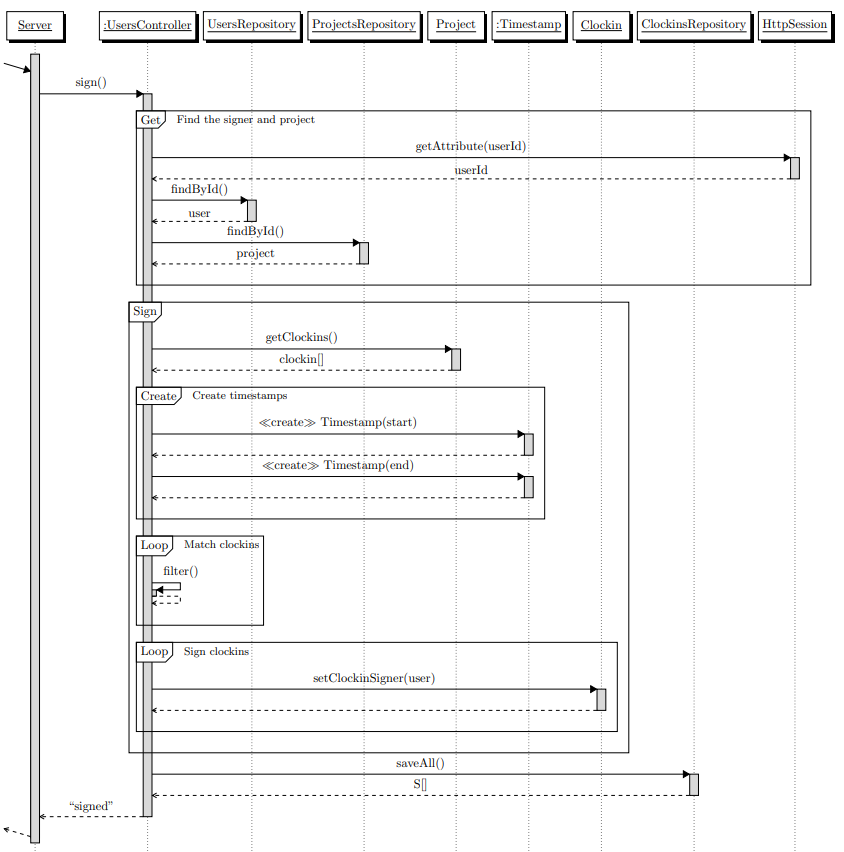
Figur 14: Sekvensdiagram som beskriver när en projektledare eller admin tar bort ett projekt och processen misslyckas



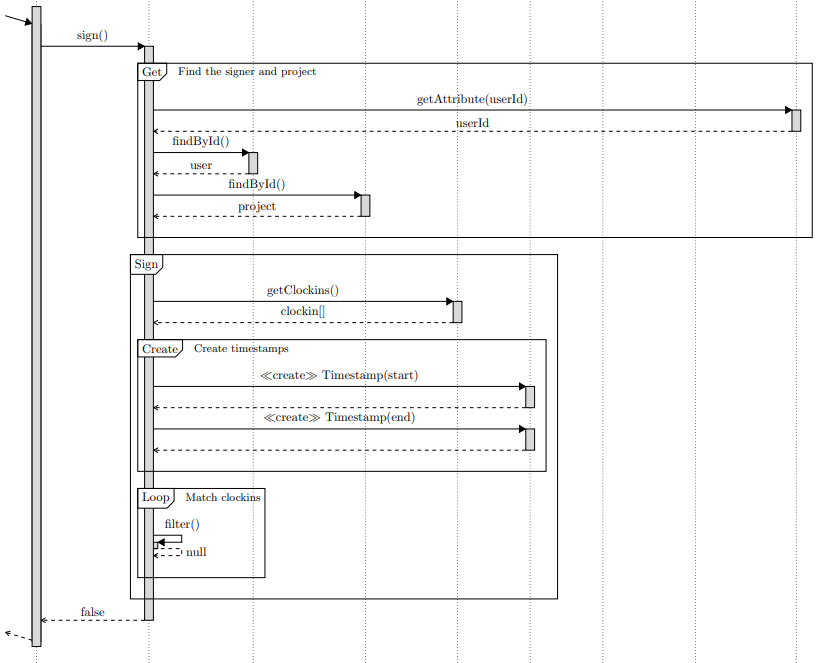
Figur 15: Sekvensdiagram som beskriver ett lyckat försök när en användares information uppdateras. Detta gäller, ändring av lösenord, mejl samt telefonnummer



Figur 16: Sekvensdiagram som beskriver ett misslyckat försök när en användares information uppdateras. Detta gäller, ändring av lösenord, mejl samt telefonnummer



Figur 17: Sekvensdiagram som beskriver processen av signering av tidrapport



Figur 18: Sekvensdiagram som beskriver ett misslyckat försök att signera en tidrapport