*Lukas Orluff Knudsen*

[*eaaluok@students.eaaa.dk*](mailto:eaaluok@students.eaaa.dk) *| EAALUOK*

*Systemintegration Forår 2023*

EKSAMENSPROJEKT – DMU-21T

Indholdsfortegnelse

[Opgave 1 - Integration af forretningsprocesser (15 point) 2](#_Toc134630366)

[Registrering af tilstedeværelse ved hjælp af sit nye studiekort 2](#_Toc134630367)

[Opgave 2 - Service integrations arkitektur (15 point) 3](#_Toc134630368)

[Event Description Table 3](#_Toc134630369)

[Service Category Table 3](#_Toc134630370)

[Opgave 3 - Informations integrations arkitektur (10 point) 4](#_Toc134630371)

[Metadata Model – C-READ 4](#_Toc134630372)

[Metadata Model – PAYS 4](#_Toc134630373)

# Opgave 1 - Integration af forretningsprocesser (15 point)

## Registrering af tilstedeværelse ved hjælp af sit nye studiekort

![A picture containing text, diagram, screenshot, number

Description automatically generated]()

Figur 1: Aktivitetsdiagram af en C-READ forretningsproces

Figur 1 beskriver et forløb over registrering af tilstedeværelse, ved brug af det nye kortlæsningssystem (C-READ). Der antages at der er installeret en kortlæsningsterminal i hvert klasselokale. Dette øger installationsomkostninger, men forhindre misbrug af systemet. Det er også antaget at C-READ indeholder CPR, navn og kortnummer for hver studerende, derudover ved C-READ hvilket klasselokale kortlæsningsterminalen er installeret i.

Flowet følger en iterativ proces. Dette betyder at C-READ kun anmoder om data fra de andre systemer, og har derfor ALT ansvaret for processen. Dette kaldes høj samhørighed og lav kobling, da hvert system står for udførslen af KUN deres egne opgaver. Forløbet ender med en fejl- eller succesmeddelelse.

# Opgave 2 - Service integrations arkitektur (15 point)

## Event Description Table

|  |  |  |  |
| --- | --- | --- | --- |
| # | Business event | Description | Responses |
| 1 | Registrering af tilstedeværelse ved hjælp af sit nye studiekort | Studerende registrerer sin tilstedeværelse ved at scanne sit studenterkort på kortlæsningsterminalen i klasseværelset | 1. Verificere studenterkort 2. Verificere at den studerende er aktiv 3. Verificere at den studerende er i det korrekte klasseværelse 4. Registrerer tilstedeværelse i LMS 5. Bekræftelse af registrering |
| 2 | Køb af mad i kantinen ved at betale med sit nye studiekort | Studerende betaler ved at scanne sit studenterkort på kortlæsningsterminalen i kantinen | 1. Verificere studenterkort 2. Verificere at den studerende er aktiv 3. Tjekker om der er tilstrækkelige saldo på studenterkortet 4. Fratrækker mængde fra saldo 5. Bekræftelse af betaling |

## Service Category Table

|  |  |  |  |
| --- | --- | --- | --- |
| Response | Description | Category | Systems |
| 1.1 Verificere studenterkort | Tjekke at studenterkortet er tilknyttet en studerende | Studenterkort | C-READ |
| 1.2 Verificere at den studerende er aktiv | Tjekke at studerende er studieaktiv | Studenterkonti | STAS |
| 1.3 Verificere at den studerende er i det korrekte klasseværelse | Tjekke om der eksisterer en lektion for den givne klasse, i det givne klasseværelse, på den givne dato og tidspunkt | Skemastyrring | SS |
| 1.4 Registrerer tilstedeværelse i LMS | Send anmodning om at registrere den givne studerende som tilstede | Tilstedeværelse | LMS |
| 1.5 Bekræftelse af registrering | Meld tilbage til studerende at de er blevet registreret som tilstede | Tilstedeværelse | C-READ |

# Opgave 3 - Informations integrations arkitektur (10 point)

## Metadata Model – C-READ

|  |  |  |  |
| --- | --- | --- | --- |
|  | Data element name | <StudenterID> | <Studenterkortnummer> |
| Basic metadata | Data source | <IDM udtrækker løbende data fra CPR-registret. Studenter ID er direkte relateret til den studerendes CPR> | <C-READ autogenerere ét unikt studenterkortnummer for hver student> |
| Description | <Studenter ID bruges som den studerendes unikke ID. Dette er direkte relateret til den studerendes CPR. Studenter ID’et bruges til at tilknytte ét studenterkortnummer til den korrekte student> | <Studenterkortnummer bruges til at tilknytte ét studenterkort til den korrekt student. Studenterkortnummer identificere et studenterkort> |
| Format and data type | <DDMMYYXXXX – String> | <XXXXYYYYZZZZWWWW – String> |
| Canonical name | <Studerendes ID> | <Studenterkortnummer> |
| Canonical format | <DDMMYYXXXX> | <XXXXYYYYZZZZWWWW> |
| Transformation rules | <DDMMYY-XXXX til DDMMYYXXXX> | <XXXX-YYYY-ZZZZ-WWWW til XXXXYYYYZZZZWWWW> |
| Interface | <Web service, API, adapter, MQ, SQL> | <Adapter, MQ, SQL> |
| Semantic metadata | Integrity rules | <Skal være unikt og må kun være tilknyttet én person> | <Skal være unikt og der må kun være ét studenter ID, studenterkortnummer par> |

## Metadata Model – PAYS

|  |  |  |
| --- | --- | --- |
|  | Data element name | <Saldo> |
| Basic metadata | Data source | <Når en student opretter en PAYS konto, bliver saldo initialiseret til decimaltallet 0,0. Hvis en studerende indsætter eller fratrækker (betaler) kroner på/fra deres PAYS konto, bliver saldo opdateret med beløbet> |
| Description | <Saldo er mængden af kroner der er tilgængelig på den studerendes PAYS konto. Dette er kroner de kan bruge i kantinen> |
| Format and data type | <X.Y – Double> |
| Canonical name | <Saldo> |
| Canonical format | <X.Y> |
| Transformation rules | <> |
| Interface | <MQ, SQL> |
| Semantic metadata | Integrity rules | <Saldo er et decimaltal der repræsenterer antal tilgængelige danske kroner på den studerendes PAYS konto> |

# Opgave 4 - Enterprise integrationsmønstre (10 point)

## C-READ

## PAYS