

Nyttjande av sensordata för optimerad lokalanvändning i högskolemiljö

Utveckling av ett API-baserat statistikverktyg för närvarodata vid Chalmers tekniska högskola

Kandidatarbete vid Industri- och materialvetenskap

Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari, Oscar Lindqvist, Andreas Maass Magnusson och Erik Persson

KANDIDATARBETE 2025

Nyttjande av sensordata för optimerad lokalanvändning i högskolemiljö

Utveckling av ett API-baserat statistikverktyg för närvarodata vid
Chalmers tekniska högskola

Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari,
Oscar Lindqvist, Andreas Maass Magnusson och Erik Persson



CHALMERS

Institutionen för industri- och materialvetenskap
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2025

Nyttjande av sensordata för optimerad lokalanvändning i högskolemiljö
Utveckling av ett API-baserat statistikverktyg för närvarodata vid Chalmers tekniska högskola

Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari, Oscar Lindqvist,
Andreas Maass Magnusson och Erik Persson

© Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari, Oscar Lindqvist,
Andreas Maass Magnusson och Erik Persson, 2025.

Handledare: Erik Hulthén, Institutionen för industri- och materialvetenskap
Examinator: Andreas Dagman, Institutionen för industri- och materialvetenskap

Kandidatarbete 2025
Institutionen för för industri- och materialvetenskap
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: Flödesschema för konceptet som presenteras i rapporten.

Göteborg 2025

Förord

Ett stort tack vill vi rikta till vår handledare och uppdragsgivare Erik Hulthén, som visat stort engagemang och givit värdefulla insikter som hjälpt detta kandidatarbete styras i rätt riktning. Vi vill också tack Ella Dahlström och Lucas Fosse, parallellt arbetande med befintliga systemet Find A Room och sensorer på Chalmers, som gav mycket feedback inför utformning av detta projekt.

Tack även till de som deltog och ställde upp på intervjuer som gav både insikter och värdefullt underlag för projektets resultat.

Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari, Oscar Lindqvist,
Andreas Maass Magnusson och Erik Persson

Göteborg, Maj 2025

Nyttjande av sensordata för optimerad lokalanvändning i högskolemiljö
Utveckling av ett API-baserat statistikverktyg för närvarodata vid Chalmers tekniska högskola

Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari, Oscar Lindqvist,
Andreas Maass Magnusson och Erik Persson
Institutionen för industri- och materialvetenskap
Chalmers Tekniska Högskola

Sammanfattning

Detta kandidatarbete bygger vidare på tidigare projekt vid Chalmers tekniska högskola med syftet att utveckla ett verktyg för att tillgängliggöra närvarostatistik för nyttjandegrad av lokaler inom högskolemiljön. Utgångspunkten var det befintliga systemet Find A Room, som mäter lokalnärvaro med hjälp av sensorer, men som i dagsläget saknar ett statistikverktyg för vidare analys och optimering av lokalanvändning. Projektets mål har varit att undersöka hur ett sådant statistikverktyg kan utformas för olika målgrupper samt att belysa etiska och säkerhetsmässiga aspekter vid hantering av närvarodata.

En behovsanalys genomfördes genom en kvalitativ undersökning i form av intervjuer i samband med granskning av resultat från tidigare arbetens behovsanalyser. Användarkrav prioriterades med hjälp av metoder som MoSCoW och Analytic Hierarchy Process. Detta låg till grund för konceptgenerering och val av en API-baserad lösning.

Arbetet resulterade i en kravspecifikation, systemarkitektur och en fungerande prototyp i form av ett REST-API. Prototypen integrerar närvarodata och föreläsningsbokningar. Den visualiserar även förhållandet mellan registrerade och faktiskt närvarande studenter samt historisk närvaro per kurstillfälle.

Slutsatsen är att ett modulärt API-baserat system skapar goda förutsättningar för vidareutveckling och integration i befintliga plattformar. Arbetet identifierar även viktiga etiska överväganden såsom anonymitet, behörighet och datasäkerhet, och föreslår riktlinjer för hur dessa kan hanteras i framtida vidareutveckling.

Nyckelord: API, lokalanvändning, närvaromätning, statistik, undervisningslokaler

Utilization of Sensor Data for Space Optimization in a Higher Education Environment

Development of an API-Based Analytics Tool for Attendance Data at Chalmers University of Technology

Melker Gustafsson, Jonatan Jageklint, Senja Kortesaari, Oscar Lindqvist, Andreas Maass Magnusson and Erik Persson

Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

This bachelor's thesis builds upon previous projects at Chalmers University of Technology with the goal of developing a tool to make occupancy statistics for room utilization in higher education environments accessible. The foundation was the existing system Find A Room, which measures room occupancy using sensors, but currently lacks a statistical tool for further analysis and optimization of space usage. The objective of the project has been to investigate how such a statistical tool can be designed for different target groups, as well as to highlight ethical and security-related aspects in the handling of occupancy data.

A user analysis was conducted through a qualitative study in the form of interviews in conjunction with a review of results from previous projects' needs assessments. User requirements were prioritized using methods such as MoSCoW and Analytic Hierarchy Process. This formed the basis for generating a concept and the selection of an API-based solution.

The work resulted in a requirements specification, system architecture, and a functioning prototype in the form of a REST API. The prototype integrates occupancy data and lecture bookings. It visualizes the relationship between registered and actually attending students as well as the historical attendance per course occurrence.

The conclusion is that a modular API-based system provides a solid foundation for further development and integration into existing platforms. The work also identifies important ethical considerations such as anonymity, access control, and data security, and proposes guidelines for how these could be managed in future development.

Keywords: API, educational facilities, statistics, space utilization, occupancy

Nomenklatur

Analytic Hierarchy Process Att vikta olika krav genom att parvis jämföra dem. 7, 12, 13

användarkrav Användarbehov. 7

API Application Programming Interface. 5–7, 15, 16, 21, 27, 29, X, XI

ASP.NET Ett ramverk för att utveckla applikationer. 16

backend Backend inkluderar servern bakom en applikation, som ansvarar för att hantera data. Tar emot, bearbetar, och skickar tillbaka information till frontend. 6, 13, 30

baskrav Ett grundläggande krav som måste uppfyllas men som ofta tas för givet av användaren. 7

brainstorm En teknik för att generera idéer och lösa problem. 11

Campus Maps Ett program som utvecklats av Chalmers och används för att hitta lokaler på campus. 18, 25

Canvas En lärplattform som används för att hantera kurser och utbildningsmaterial. 18

controller Den del av RSC som hanterar förfrågningar från klienter. 6, 7

CSV Comma-Separated Values, ett filformat som används för att lagra data i tabellform. 14, 15

databas Används för att hantera, lagra och hämta data på ett effektivt sätt. Data baseras på information som till exempel siffror eller text. 13, 14

endpoint En specifik URL där en klient kan få tillgång till resurser eller tjänster från ett API. 5, 7, 27, 29

- enhetstest** Ett test som körs på enskilda komponenter eller moduler i en mjukvaraapplikation för att säkerställa att de fungerar korrekt. 16
- Figma** Ett designverktyg för att designa prototyper. 13
- Find A Room** Ett system för att hitta lediga grupprum. 1–3, 14, 20, 33, IX
- frontend** Frontend är den synliga delen av en applikation, det vill säga det gränssnitt som användare interagerar med. 6, 13, 21, 30
- funktionella krav** Beskriver vilka funktioner systemet ska ha, det vill säga vad användaren ska kunna göra i systemet. 6
- GitHub** Webbaserad plattform som tillåter utvecklare att spara programkoden tillsammans med dess tidigare versioner, och möjliggör samarbete mellan flera utvecklare genom inbyggd konfliktshantering. 15
- Grafana** Ett statistikverktyg som används för att visualisera data och går att koppla till egna databaser och servrar. 13, 25, 27, 33
- heatmap** En grafisk representation av data där värden representeras med färger. 17, 20, VI
- hårdvara** Fysiska komponenter i en dator. 2
- icke-funktionella krav** Beskriver systemets egenskaper, såsom prestanda, säkerhet, tillgänglighet och underhållbarhet. 6
- InfluxDB** En tidsseriedatabas, vilket innebär att den är optimerad för att lagra tidsstämplad data samt mäta hur den förändras över tid. 5, 14
- JSON** JavaScript Object Notation. 6, 15, 27
- kanbantavla** Ett verktyg för att visualisera arbetsprocesser. 5, 15
- Kesselringmatris** En metod för att prioritera och utvärdera olika alternativ baserat på viktade kriterier. 8
- klient** En dator eller applikation som ansluter till en server för att få tillgång till tjänster eller resurser. 5
- Kårappen** En app som innehåller information och tjänster från Chalmers studentkår. 18

- Ladok** Ett system för studieadministration vid svenska universitet och högskolor. 6, 13
- mjukvara** En samling av instruktioner som körs av en dator, ofta ett program. 2
- mock** Simulerad data eller beteende som ersätter verkliga beroenden vid testning eller utveckling. Används exempelvis för att ersätta externa system eller databaser . 16
- Moq** Bibliotek för mocking i .NET. 16
- MoSCoW** Ett sätt att prioritera krav, där man delar upp de i must, should, could och won't have. 13, 21, 22
- OpenAPI 3.0** En specifikation för att beskriva REST-baserade API:er på ett standardiserat sätt. . 15
- Power BI** Ett verktyg från Microsoft för att visualisera och analysera data. 25, 33
- Python** Ett högnivåprogrammeringsspråk som används för många typer av utveckling, inklusive webb, dataanalys och maskininlärning. 14
- QlikView** Ett statistikverktyg som används för att visualisera data och går att koppla till egna databaser och servrar. 18, 25
- repository** Den del av RSC som hanterar kopplingen till datakällor. 6
- Repository-Service-Controller** Repository-Service-Controller. 6, 25, 30
- REST** Representational State Transfer. 5, 27
- Scrum** Ett ramverk för produktutveckling. 9, 15
- service** Den del av RCS som hanterar affärslogik och således hämtar och kombinerar nödvändig data från repository. 6, 7
- sprint** En tidsbestämd arbetsperiod. 9, 15
- Swagger** Ett verktyg som tillåter utvecklare att dokumentera strukturen av ett API på ett standardiserat sätt. 15
- systemtest** Ett test som körs på hela systemet för att säkerställa att det fungerar korrekt och uppfyller kraven. 16

TA Teaching Assistant, en anställd som hjälper till med undervisning. 17, 19, 20, 23, IX

TimeEdit En schemadatabas för högskolor. 7, 13–15, 18

user story Användarperspektiv på funktionalitet av produkt. 9, 15

web scraping En metod för att extrahera data från webbplatser. 32

Innehåll

Sammanfattning	iii
Abstract	iii
1 Inledning	1
1 Problem	2
2 Syfte	3
3 Frågeställningar	3
4 Avgränsningar	3
2 Teori	5
1 Teknisk bakgrund	5
1.1 Kanbantavla	5
1.2 API	5
1.3 REST	5
1.4 JSON	6
1.5 Kvalitetsattribut för mjukvara	6
1.6 Designmönster	6
1.7 Definition av krav	7
2 Metodramverk och utvecklingsmodeller	7
2.1 Analytic Hierarchy Process	7
2.2 Mocking	7
2.3 Kesselringmatris	8
2.4 MoSCoW	8
2.5 Agil Utveckling	8
2.6 V-modellen	9
3 Metodik	10
1 Behovsanalys	11
1.1 Kvalitativ undersökning	11
1.2 Kvantitativ undersökning	12
1.3 Användarkrav och tekniska krav	12
2 Konceptgenerering och eliminering	12
3 Slutgiltig kravspecifikation	12
4 Koncept	13

5	Prototyp	13
5.1	Anpassningar av systemarkitekturen	13
5.2	Prototyputvecklingsprocess	15
6	Utvärdering	15
6.1	Enhetstester	16
6.2	Integrationstester	16
6.3	Systemtester	16
4	Resultat	17
1	Behovsanalys	17
1.1	Intervjuer med målgrupper	17
1.2	Studenters behov	18
1.3	Etiska aspekter	18
1.4	Övriga identifierade behov	18
1.5	Användarkrav	19
2	Koncept	21
2.1	Systemdesign	24
3	Prototyp	25
3.1	Avgränsning av krav	26
3.2	Systemarkitektur i prototypen	26
3.3	API	27
3.4	Exempel på frontend-integration	27
4	Utvärdering av koncept	29
5	Diskussion	30
1	Etiska aspekter	31
2	Förslag till framtida arbete	32
6	Slutsats	34
1	Vilken närvarostatistik är relevant för olika målgrupper på en högskola?	34
2	Hur kan insamlad data göras tillgängligt för användare på ett effektivt vis?	34
3	Vilka etiska aspekter bör tas hänsyn till vid hantering och visualisering av närvarodata och hur kan de aspekterna hanteras?	35
	Referenser	36
A	Val av koncept	I
B	Figmaskisser	III
C	Slutgiltig kravspecifikation	VII
1	Funktionella krav	VIII
2	Icke-funktionella krav	X
D	API-endpoints med exempelsvar	XII
1	Närvaroinformation	XII
2	Kurser	XIII

Innehåll

3	Undervisningstillfällen	XIV
4	Kurstillfällen	XVI
5	Lokaler	XVII

Kapitel 1

Inledning

Högskolelokaler som grupprum och föreläsningssalar utgör en central del av högskolemiljön. De möjliggör en studiemiljö som är avgörande för studenters lärande och spelar en betydelsefull roll både akademiskt och för det sociala samarbetet mellan studenter. Många universitet arbetar för att optimera sin lokalanvändning, och Chalmers tekniska högskola är inget undantag. Chalmers bedriver sin verksamhet uppdelat på två campus, Lindholmen och Johanneberg. Under 2023 beslutades det att samla nästan all verksamhet på campus Johanneberg [1], vilket medför att användningen av högskolans lokaler förväntas öka. I samband med denna flytt blir behovet av en bättre överblick över lokalernas nyttjandegrad samt optimering av lokalanvändningen särskilt aktuellt.

Redan 2023 utfördes ett kandidatarbete som syftade på att undersöka vilka teknologier som kunde användas för att mäta närvaro i Chalmers lokaler. Arbetet identifierade vilka typer av sensorer som var lämpliga att använda med hänsyn till studenters integritet. Mätdata skulle användas för att studenter lättare skulle kunna hitta lediga grupprum och därav öka lokalnyttjandet. På sikt skulle det även användas för att analysera i vilken mån lokalerna faktiskt används [2]. Konceptet vidareutvecklades under 2024, då ytterligare två kandidatarbeten utfördes parallellt, där fokuset låg på grupprum. Ett av dessa kandidatarbeten resulterade i utvecklingen av systemet *Find A Room*, vilket ligger till grund för detta kandidatarbete. Find A Room består av ett system sensorer som kan registrera upp till sex personer i ett grupprum. Insamlad närvarodata lagras i en databas och presenteras på en webbapplikation där studenter i realtid kan se vilka grupprum som är lediga eller upptagna. Applikationen inkluderar även en statistikpanel riktad till Chalmers anställda. Där presenteras statistik i form av en graf som visar hur många personer som vistats i grupprummen den senaste timmen [3].

Närvaromätningen i Find A Room utvecklas fortfarande, men statistikdelen är än så länge begränsad och behöver vidareutvecklas. I dagsläget saknas ett effektivt statistikverktyg för att visualisera närvarodata i Chalmers lokaler. Statistiken kan nyttjas av olika yrkesroller för att ge en bättre förståelse kring hur grupprum och

föreläsningssalar nyttjas. Detta kan minska risken för outnyttjad kapacitet samtidigt som behovet av nya ytor kan bedömas mer precist. Genom att exempelvis analysera antalet studenter som faktiskt deltar på föreläsningar kan schemaläggare anpassa salarnas storlek utifrån det faktiska behovet. Även ekonomiska aspekter skulle kunna gynnas, då närvarodata kan möjliggöra en mer behovsanpassad lokalvård, vilket skulle innebära en effektivare arbetsplanering och resursanvändning. Det skulle i sin tur kunna minska kostnader. En effektivare lokalanvändning kan även optimera energiförbrukning, vilket bidrar till en minskad miljöpåverkan, exempelvis genom att undvika uppvärmning av lokaler som inte används och ett minskat behov av lokalytor [4].

För att kunna dra full nytta av denna typ av statistik krävs dock en förståelse för vilken data som är mest relevant för olika målgrupper och hur den ska tillgängliggöras på ett användarvänligt vis. Ett statistikverktyg skulle möjliggöra en rad olika användningsområden, däremot öppnar det även upp för potentiella risker. Närvarodata kan uppfattas som känslig information, särskilt om det går att kopplas till individer eller grupper. Detta öppnar även upp för frågan om det finns negativa följder kring om sådan data görs allmänt tillgänglig. Under utvecklingen av ett sådant verktyg bör därför etiska- och säkerhetsaspekter beaktas.

1 Problem

Find A Room-systemet har redan implementerad hårdvara och mjukvara för att samla in närvarodata, men saknar ett effektivt verktyg för att omvandla denna data till användbar statistik. För att systemet ska kunna fungera som stöd för schemaläggare och andra målgrupper krävs ett användarvänligt gränssnitt som tydligt visualiserar relevant information. En central utmaning är att analysera den insamlade data och identifiera vilken typ av statistik som är relevant att presentera. Här behöver datas tillförlitlighet och eventuella begränsningar beaktas, liksom hur informationen kan användas. En felaktig användning av statistiken kan leda till missvisande beslut, vilket gör det viktigt att belysa risken för felanvändning.

En annan viktig aspekt är hur statistiken bör anpassas för olika målgrupper. Olika användare har ofta varierande behov och intressen, vilket gör det nödvändigt att utforska vilka målgrupper som behöver vilken typ av data och hur den bör presenteras för att vara mest användbar.

Vid hantering av närvarodata uppstår etiska frågor, exempelvis kopplade till integritet och anonymitet. Det är viktigt att undersöka vilka etiska dilemman som kan uppstå under projektets gång och vilka konsekvenser de kan få. Därtill behöver säkerhetsaspekter beaktas; en central fråga är hur man säkerställer att endast behöriga användare får tillgång till relevant statistik och vilka risker det finns om denna insamlade data skulle hamna i fel händer.

2 Syfte

Arbetet syftar till att undersöka hur ett statistikverktyg kan utformas för att tillgängliggöra närvarodata för olika målgrupper och utifrån detta utveckla en prototyp. Detta innebär inte bara att framställa information som målgruppen har störst behov av, utan även ta hänsyn till aspekter som behörighet och etik, samt att projektet ska vara långsiktigt hållbart med tanke på underhåll och skalbarhet. Arbetet utförs i samarbete med Chalmers med sikt på att optimera deras lokalanvändning. Målet är att detta ska främja studenter genom att öka möjligheterna för Chalmers att skapa en effektiv studiemiljö.

3 Frågeställningar

Med utgångspunkt i problembeskrivningen har tre frågeställningar formulerats.

1. Vilken närvarostatistik är relevant för olika målgrupper på en högskola?
2. Hur kan insamlad data göras tillgänglig för användare på ett effektivt vis?
3. Vilka etiska aspekter bör tas hänsyn till vid hantering och visualisering av närvarodata och hur kan de aspekterna hanteras?

4 Avgränsningar

Vid utveckling av prototypen har gruppen valt att avgränsa sig till att inte undersöka en optimal utformning av ett grafiskt gränssnitt för att visualisera närvarostatistik. I det ingår exempelvis att inte forska kring bäst lämpade typer av grafer för olika nyckeltal. Detta beror på att det finns befintliga visualiseringsverktyg som kan integreras med prototypen för att uppnå arbetets syfte. Många av dessa verktyg har etablerade sätt att presentera data på ett användarvänligt sätt. Detta val möjliggör också ett effektivare arbetssätt, där fokus kan läggas på själva kärnan i arbetet, det vill säga att sammanställa relevant närvarodata på ett sätt som gör den lätt att visualisera och analysera.

Find A Room-systemet, som detta arbete baseras på, involverar flera viktiga hårdvarukomponenter som sensorer för närvaromätning och servrar för datalagring. Eftersom arbetet grundar sig i ett redan befintligt system som dessutom fortfarande utvecklas, kommer inga ytterligare hårdvarukomponenter att vidareutvecklas eller hanteras.

Prototypen som utvecklas avgränsas till att endast hantera universitetslokaler. Ett färdigutvecklat statistikverktyg skulle i framtiden potentiellt kunna hantera närvarodata från andra typer av lokaler, men detta ligger utanför ramarna för det aktuella arbetet.

Ytterligare en avgränsning handlar om projektets målgrupp. I detta arbete har gruppen beslutat att endast fokusera på Chalmers anställda och studenter, vilket bland annat omfattar schemaläggare och lokalansvariga, men inte allmänheten.

Kapitel 2

Teori

Arbetet innefattar en mängd tekniska begrepp och metodramverk som beskrivs nedan.

1 Teknisk bakgrund

1.1 Kanbantavla

En *kanbantavla* är ett verktyg för att visualisera arbetsflöden och organisera projekt, och kan vara både fysiska och digitala. Tavlorna består av kolumner som representerar utvecklingen i arbetsprocessen. Uppgifter blir upplagda på tavlan, får tilldelade personer och flyttas mellan kolumnerna under arbetsprocessen [5].

1.1.1 InfluxDB

InfluxDB är en tidsseriedatabas, vilket innebär att den är optimerad för att lagra tidsstämplad data. Detta innebär datapunkter som är kopplade till en specifik tidpunkt, och tidsseriedatabaser är optimerade för att effektivt mäta hur dessa förändras över tid [6].

1.2 API

API är ett gränssnitt som tillåter flera system att kommunicera och dela data med varandra. Den fungerar som en brygga mellan två system, där den tar emot förfrågningar från en klient, vidarebefordrar dem till en server och returnerar svaren tillbaka till klienten [7].

Endpoints är komponenter inom ett API och är specifika adresser som pekar ut platser för specifika resurser på API-servern. API-klienten, alltså själva mjukvaran som efterfrågar information, använder endpoints för att efterfråga information från API-servern [8].

1.3 REST

REST är en arkitekturstil för att designa nätverksbaserade system. Det bygger på att resurser, såsom dataobjekt eller tjänster, adresseras via URL:er och manipuleras genom standard HTTP-metoder som GET, POST, PUT och DELETE. REST följer

principer som gör det möjligt för system att vara enkla att integrera och skala. Detta görs oberoende av tillstånd, det vill säga att all nödvändig data skickas med varje förfrågan. Genom att använda enhetliga och väl definierade gränssnitt, görs det enkelt att kommunicera mellan olika system.

RESTful API:er är vanligt förekommande i moderna webbapplikationer där olika komponenter kommunicerar med varandra, till exempel mellan frontend och backend [9].

1.4 JSON

JSON är ett format för att representera strukturerad data. *JSON* används ofta för att utbyta data mellan klient och server, särskilt inom webbutveckling, eftersom det är lättläst för både människor och maskiner. Formatet är textbaserat och bygger på att data representeras i nyckel-värde-par, vilket gör det väldigt flexibelt och lätt att använda i en mängd olika programmeringsspråk [10].

1.5 Kvalitetsattribut för mjukvara

Under utveckling av mjukvarusystem ges ofta stort fokus på att uppfylla *funktionella krav*, men det är också avgörande att beakta olika kvalitetsattribut för att säkerställa att produkten blir korrekt, säker, hållbar och användbar [11]. Dessa kvalitetsattribut formuleras vanligtvis som *icke-funktionella krav* i kravspecifikationen och är ofta mer tekniskt orienterade. En särskilt viktig egenskap hos dessa attribut är att de kan utvärderas kvantitativt genom testning [12].

Ett av de mest centrala attributen är tillförlitlighet, vilket innebär att mjukvaran ska uppvisa korrekt beteende och tillhandahålla sin tjänst utan fel eller avbrott. Nära kopplat till detta är underhållbarhet, som beskriver hur enkelt systemet kan återställas till operativt skick efter exempelvis ett systemfel eller en krasch. Tillgänglighet kombinerar dessa två aspekter och definieras som sannolikheten att systemet är i fungerande tillstånd vid ett givet tillfälle.

För användaren är användbarhet ett viktigt kvalitetsattribut. Det syftar på att systemet ska vara intuitivt och effektivt att använda, något som kan mätas genom användartester för att verifiera att produkten är enkel att förstå och hantera. Slutligen är skalbarhet en egenskap som beskriver systemets förmåga att anpassas till ökad användning, exempelvis genom att klara fler användare eller större datamängder utan att försämra prestandan.

1.6 Designmönster

För att uppfylla de tekniska kraven på modulär arkitektur samt möjliggöra effektiv och skalbar dataåtkomst bygger programvaran som hanterar beräkningarna på ett designmönster känt som *Repository-Service-Controller*. Damyanov m. fl. [13] skriver att mönster delar upp systemet i de tre huvudsakliga komponenterna *repositories*, *services* och *controllers*.

Repositories ansvarar för att hämta och formatera data från olika datakällor. I systemet finns exempelvis ett *repository* för att hantera data från Ladok, ett annat

för TimeEdit, samt ett för sensordata från databasen. Dessa repositories strukturerar informationen så att den kan användas av övriga systemkomponenter.

Services utgör systemets affärslogik och ansvarar för att bearbeta insamlad data. Det är tjänsterna som beräknar nyckeltal, sammanställer information och kopplar samman data från de olika källorna. Exempelvis kan tjänster hantera analys av föreläsningsdata och närvarostatistik.

Slutligen fungerar controllers, eller kontroller, som gränssnitt mellan systemets endpoints i API:et och de underliggande tjänsterna. Kontrollerna tar emot inkommande förfrågningar, förmedlar dessa till rätt tjänst och returnerar sedan det bearbetade svaret till användaren.

1.7 Definition av krav

En kravspecifikation kan vara uppdelad i funktionella krav respektive icke-funktionella krav. De två definitionerna som används här liknar hur Fatimatu Zahra m. fl. [14] beskriver dem. Funktionella krav beskriver vilka funktioner systemet ska ha, det vill säga vad användaren ska kunna göra, medans icke-funktionella krav handlar om systemets egenskaper, till exempel prestanda, tillgänglighet, säkerhet, pålitlighet, skalbarhet och underhåll. De påverkar hur väl systemet uppfyller användarnas förväntningar men beskriver inte direkt någon specifik funktion.

I arbetet har *användarkrav* definierats som en sammanställning av användarnas behov. De kan sedan överföras som funktionella och icke-funktionella krav i en kravspecifikation.

Baskrav i rapporten syftar på grundläggande krav som utgör ett minimum av funktionalitet, kvalitet eller prestanda som måste finnas på plats, oavsett övriga önskemål eller prioriteringar, för att ett system ska vara användbart.

2 Metodramverk och utvecklingsmodeller

2.1 Analytic Hierarchy Process

Analytic Hierarchy Process är en metod som kan appliceras inom beslutsanalys, och kan användas för att genomföra parvisa jämförelser av olika kriterier eller krav inför ett beslut [15]. Kraven viktas numeriskt, exempelvis om ett krav är viktigare än ett annat sätts värdet till 1, om de bägge är lika viktiga 0,5. Ett krav kan även sättas till 0 om ett annat krav är viktigare [16]. Summan av varje krav som fås genom en sådan parvis jämförelse räknas samman och delas med totalsumman för alla kraven. Detta ger respektive kravs vikt.

2.2 Mocking

Mocking innebär att man ersätter riktiga komponenter, data eller beroenden, med simulerade versioner under utveckling och testning. Detta används ofta i situationer där det är opraktiskt eller omöjligt att använda det faktiska systemet, exempelvis

vid begränsad åtkomst till externa API:er eller databaser. Mocking möjliggör utveckling av systemets övriga delar utan att vänta på att de externa delarna blir tillgängliga [17].

2.3 Kesselringmatris

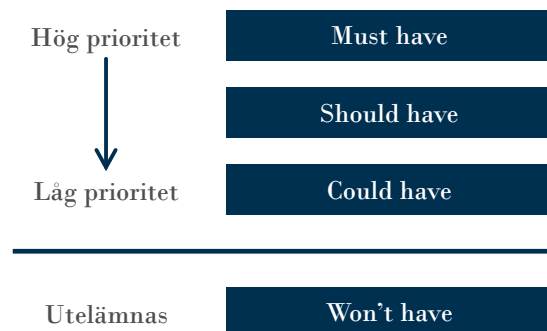
En Kesselringmatris är en metod som systematiskt utvärderar och rangordnar koncept eller lösningsförslag [18]. Den kan tillämpas under produktutveckling i fall där ett flertal alternativ på lösningar behöver jämföras mot användarkrav och önskemål.

I en Kesselringmatris placeras utvärderingskriterier mot konceptförslag. Kriterier tilldelas en viktning på en skala mellan 1 och 5, där 5 innebär att konceptet löser kravet utmärkt, medan 1 innebär att konceptet inte löser kravet alls. Varje koncepts betyg räknas sedan samman som en viktad summa med varje kriteriers respektive vikter. Konceptet med högst viktad summa blir då det koncept som bör användas.

2.4 MoSCoW

MoSCoW-metoden är en etablerad och ofta använd metod för kravprioritering inom mjukvaruutveckling. Efter att krav och önskemål samlats in från användare, struktureras dessa i fyra prioriteringsnivåer för att tydligt kunna avgöra vilka funktioner som är mest kritiska för systemet [16]. Metodens namn är en akronym som representerar de fyra kategorierna: must, should, could och won't have, vilket illustreras i figur 2.1.

Kategorin must have innefattar krav som är absolut nödvändiga för att systemet ska vara användbart; dessa funktioner är produkten i praktiken inte genomförbar. Should have-kraven är viktiga och ska eftersträvas, men inte helt avgörande för att produkten ska kunna användas. I could have-kategorin återfinns krav som är önskvärda men inte nödvändiga, vilket innebär att de kan implementeras om tid och resurser tillåter. Slutligen omfattar won't have-kategorin de krav som bedöms vara mindre relevanta för den aktuella utvecklingscykeln och därmed utesluts, men som eventuellt kan beaktas vid framtida vidareutveckling.



Figur 2.1: MoSCoW-metodens fyra kategorier och deras prioritering.

2.5 Agil Utveckling

Inom mjukvaruutveckling så är agil utveckling en grupp av utvecklingsstrategier som alla syftar till att skapa flexibilitet i utvecklingsprocessen och tillåter anpassning efter behov och tekniska förutsättningar [19].

2.5.1 Scrum

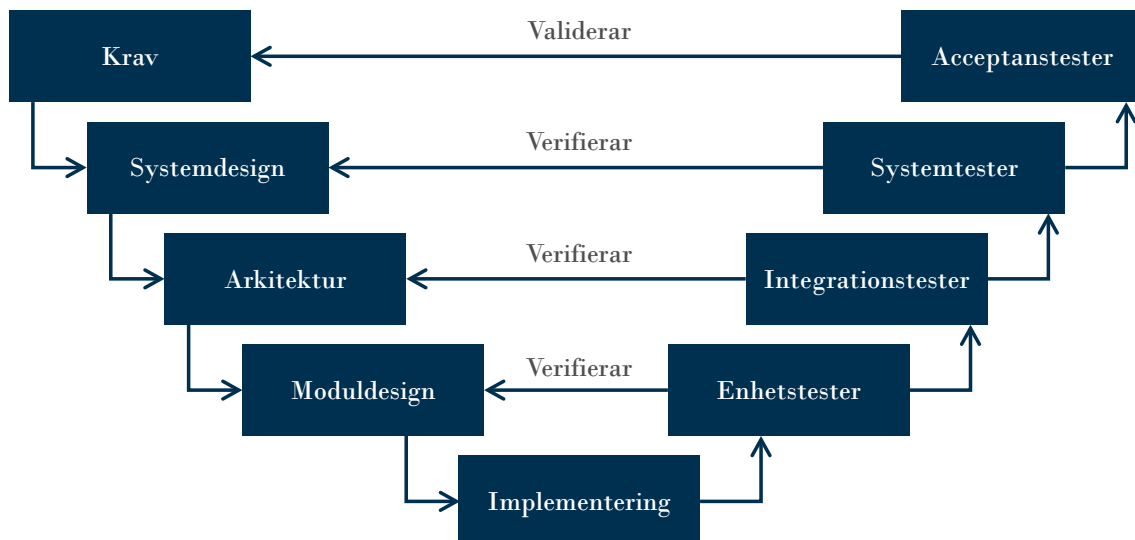
Scrum är en metod inom agil utveckling där utveckling och implementation av krav och lösningar sker iterativt i cyklar så kallade *sprints*, som vanligen pågår ett fåtal veckor. Inför varje sprint så prioriteras ett urval av krav listade som uppgifter, eller *user stories*, vilka är korta beskrivningar av funktionalitet utifrån ett användarperspektiv. De implementeras och utvärderas sedan inför nästa sprint [19].

2.6 V-modellen

V-modellen, se figur 2.2, är en linjär och sekventiell utvecklingsmodell som används för att strukturera och kvalitetssäkra systemutvecklingsprojekt [20]. Modellen kännetecknas av sin V-form, där den vänstra sidan representerar planering, kravställning och design, medan den högra sidan representerar olika nivåer av testning och verifiering. Verifikation innebär att en produkt löpande granskas för att säkerställa att den överensstämmer med tekniska specifikationer och krav. Syftet med modellen är att säkerställa att varje utvecklingssteg har en motsvarande testaktivitet, vilket gör det möjligt att verifiera att systemet uppfyller de specificerade kraven i varje fas.

En central idé i V-modellen är att testning inte enbart ses som en avslutande aktivitet, utan som en integrerad del av hela utvecklingsprocessen. Exempelvis motsvaras kravspecifikationen av en valideringsprocess av acceptanstestning, systemdesign av systemtestning och detaljerad design av enhetstestning. Detta systematiska förhållningssätt bidrar till att fel kan upptäckas tidigt, vilket i sin tur ökar effektiviteten och kvaliteten i utvecklingsarbetet.

Validering görs av exempelvis acceptanstester, feedbacksessioner och iterativa förbättringar baserat på verklig användning. Detta utgör då en viktig metod för att säkerställa att slutliga produkten faktiskt möter användarnas behov och förväntningar. Genom att involvera användarna tidigt i utvecklingsprocessen efter en fungerande prototyp framställts, säkerställs att produkten inte bara fungerar tekniskt utan även uppfyller de faktiska krav som målgruppen har.



Figur 2.2: V-modellen, fritt översatt från Rodriguez m. fl. [20].

Kapitel 3

Metodik



Figur 3.1: Flödesschema för processen.

Detta kapitel beskriver de processer och moment som projektet genomgick, samt motiverar de metoder som tillämpades för att genomföra projektet. Nedan redogörs för behovsanalys, identifiering av användarkrav utifrån samlad kvantitativ rådata samt ytterligare identifierade krav utifrån tekniska och etiska aspekter, prioritering och viktning av krav, konceptgenerering, eliminering och val av koncept, slutgiltig kravspecifikation och systemdesign, utveckling av prototyp, och hur prototypen sedan utvärderades, se figur 3.1.

Processen och metoder som appliceras följer i stort strukturen som beskrivs av Ulrich m. fl. [21]. Produktutveckling följer dessa steg som förklarat ovan. I de olika kapitlen redogörs för hur processen applicerats på produktutvecklingen inom detta projekt.

1 Behovsanalys

En behovsanalys utfördes i syfte att identifiera krav och önskemål bland olika målgrupper på Chalmers som hade kunnat nyttja information om närvaromätning i högskolans lokaler. För att identifiera relevanta yrkesroller utfördes en *brainstorm* inom projektgruppen, samt diskussion med uppdragsgivare för projektet som även är representant för Chalmers i detta arbete.

1.1 Kvalitativ undersökning

För att samla in kvalitativ data genomfördes målinriktade intervjuer med schemaläggare, programansvariga och lokalansvariga på Chalmers. Intervju som en form för insamling av behov och krav är ett etablerat sätt att genomföra en behovsanalys, att föredra i många fall över text-baserade undersökningar [21, s. 119-125]. Författarna beskriver hur intervjuer tillåter en att vara uppmärksam på icke-verbala signaler, upptäcka latenta och gömda behov hos slutanvändarna och förstå deras prioriteringar. Dessa intervjuer syftade till att förstå målgruppernas specifika behov och krav på funktionalitet hos systemet. Intervjuerna utfördes enligt semi-strukturerat upplägg, vilket öppnar för att ställa följdfrågor, anpassa sig för intervjuobjektet och ger möjlighet för diskussion som underlättar att förstå helhetsbilden hos användarnas behov [22].

1.1.1 Intervjufrågor

Intervjuerna genomfördes med övergripande intervjuguide, med följande struktur:

- Presentation av intervjuerna och kort beskrivning av projektets syfte
- En kort beskrivning av den intervjuades yrkesroll och ansvarsområden på högskolan
- Deras arbetssätt, metoder och eventuella system de använder i dagsläget?
- Huruvida de upplever behov för ytterligare verktyg som kan underlätta för deras yrkesroll
- Deras föreställningar kring hur de kan se användning för data och statistik på lokalanvändning och närvaromätning, eventuellt specifika nyckeltal?
- Hur de skulle föredra att ha tillgång till sådan data, exempelvis via en existerande plattform eller en ny?
- Om de har åsikter på etiska överväganden eller reservationer på visualisering av närvarostatistik
- Åsikter om vilka målgrupper som bör ha tillgång till närvarodata och närvarostatistik, finns det målgrupper som sådan data inte bör göras tillgänglig för?

Intervjuerna utfördes med två personer från projektets arbetsgrupp, spelades in med intervjuobjektens samtycke samt dokumenterades genom att under tiden föra anteckningar.

1.2 Kvantitativ undersökning

En betydelsefull målgrupp för behovsanalysen var studenter, som innefattade en stor mängd potentiella slutanvändare. I detta steg gjordes, istället för egna utförda undersökningar, studier på tidigare kvantitativa undersökningar som gjorts i samma kontext [3], [23]. De två kandidatarbeten som detta arbete bygger vidare på har gjort omfattande insamlingar av studenters åsikter och behov i förhållande till nyttjande av sensordata på Chalmers. Därför gjordes en sammanställning på denna redan insamlade rådata från deras behovsanalys, som i stor utsträckning bestod av enkätundersökningar.

1.3 Användarkrav och tekniska krav

Rådata från datainsamling sammanställdes i kalkylblad. Önskemål och behov kategoriserades in enligt funktionella krav och icke-funktionella krav i fall där behov eller åsikter relaterade till etiska aspekter, säkerhetsaspekter eller krav relaterade till kvalitetsattribut för mjukvara var önskade.

Ytterligare så inkluderades även vissa krav av icke-funktionell kategori enligt grundläggande krav på mjukvarukvalitet och etiska aspekter som internt beslutades som baskrav för att i senare skede kunna viktas mot funktionalitet.

2 Konceptgenerering och eliminering

Efter att användarkrav hade identifierats genomfördes en konceptgenerering med syfte att utforska olika tekniska lösningar som skulle kunna tillgodose dessa behov. Lösningar till de olika delarna av användarkraven togs fram genom brainstorming inom projektgruppen. De olika dellösningarna kunde sedan kombineras och resulterade då i ett antal konceptförslag som dokumenterades för vidare utvärdering.

De koncept som var orimliga att genomföra med avseende på kvalitetsattribut och andra baskrav togs sedan bort med hjälp av en elimineringsmatris [21]. Därefter eliminerades resterande koncept ytterligare genom att projektgruppen betygsatte de olika koncepten utifrån de identifierade användarkraven i en Kesselringmatris [18]. För att bestämma vikter för respektive användarkrav jämfördes dessa parvis i enlighet med Analytic Hierarchy Process. Dessa erbjöd hög precision och möjliggjorde viktning av kraven i relation till varandra på ett strukturerat vis [16]. Alla tre konceptvalmatriser finns listade i bilaga A.

3 Slutgiltig kravspekifikation

Efter att ett koncept valts ut kunde en slutgiltig kravspekifikation tas fram. Med användarkraven som grund kunde dessa anpassas för att vara mer specifika utifrån det koncept som valdes. Därtill lades helt nya krav till som uppkom med anledning av det valda konceptet. Här gjordes även en rangordning av kraven med hjälp

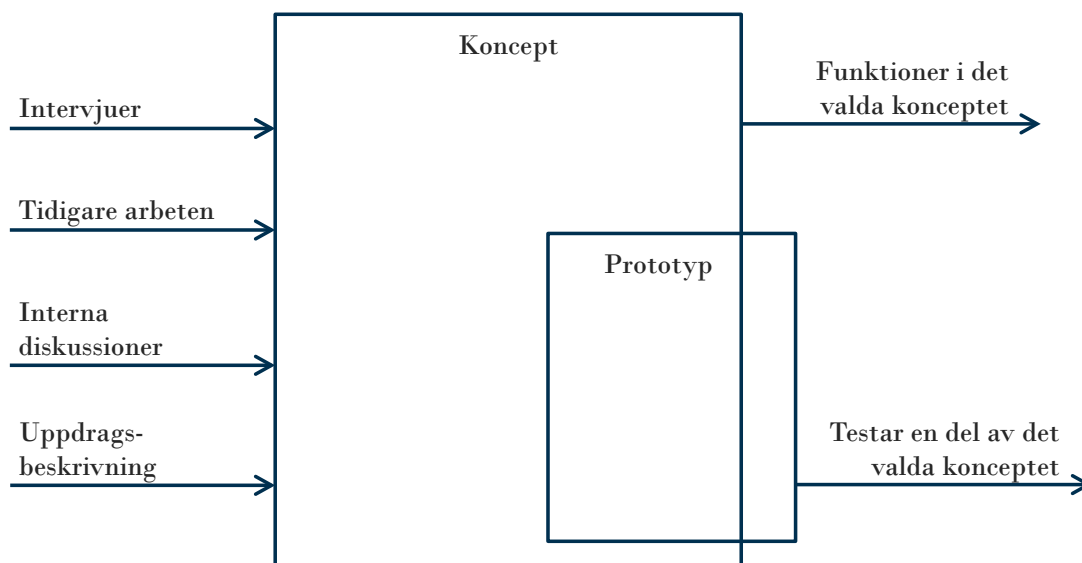
av MoSCoW, för att få en övergripande prioritering. Den tidigare viktningen av användarkraven enligt Analytic Hierarchy Process var vägledande för besluten.

4 Koncept

En lösning för att implementera konceptet togs fram och dokumenterades. En arkitektur för systemet ritades upp i form av ett flödesschema. För att illustrera delar av konceptet gjordes skisser i Figma. Detta kunde sedan utgöra en målbild för hur den färdiga produkten skulle kunna se ut.

5 Prototyp

En prototyp utvecklades för att visa på en förenklad men fungerande version av det valda konceptet och hade som fokus att demonstrera kärnfunktionaliteten. Detta gjordes för att bevisa den tekniska genomförbarheten. Den avsågs inte vara en fullständig implementation av alla krav, utan snarare ett verktyg för att validera och utvärdera konceptet. Se figur 3.2.



Figur 3.2: Utvärdering av en del av det valda konceptet genom en prototyp.

5.1 Anpassningar av systemarkitekturen

Prototypen delades upp i de tre huvudsakliga delarna databaser, frontend och backend. En begränsning för prototyputvecklingen var att databasen inte innehöll aktuell sensordata. En annan begränsning blev tydlig när ett användarkrav identifierades och visade sig vara beroende av data från de externa systemen TimeEdit och Ladox, till vilka vi inte hade direkt tillgång. En sista begränsning var tillgången till de visualiseringsverktyg som används på högskolan, då dessa är belagda med licenser, varför det ersattes med Grafana. Detta ledde till anpassningar som gjordes för att ändå visa på konceptets genomförbarhet och som tillåter att produkten i senare skede kan kopplas upp emot dessa system och databaser i framtida slutimplementation. Dessa anpassningar beskrivs i följande avsnitt.

5.1.1 Slumpgenerering av närvarodata

Databasen för närvarodata ersattes med en tillfällig InfluxDB-databas. Den valdes då det är samma typ av databas som används i det befintliga Find A Room-systemet. Då testdatabasen syftar till att efterlikna aktuella sensordatabasen är valet av denna typ väsentlig för att den ska nyttjas. Den tillfälliga databasen fylldes med slumpade data som skapades för att simulera verkliga scenarier i lokalerna.

Slumpade data skapades med hjälp av ett Python-skript som skrev simulerade mätpunkter till databasen för var femte minut under de tider det fanns en bokning enligt TimeEdit-CSV:n. Skriptet byggdes upp på så vis att det för varje bokning beräknar hur många som fanns registrerade på den eller de kurser som bokningen hörde till. Därtill räknades det ut hur stor andel av kursens föreläsningstillfällen som har passerat, vilket sedan användes för att sätta ett intervall på hur stor andel av de registrerade studenterna som kan tänkas närvara vid bokningen. Det sistnämnda gjordes för att simulera den väl kända effekten av att studenter är mer benägna att närvara vid föreläsningar i början av en kurs än i slutet av den.

Sammanfattningsvis kan beräkningen av närvaron N_f vid ett givet föreläsningstillfälle f beskrivas med följande ekvation:

$$N_f = R \cdot \left(\alpha - \beta \cdot \frac{f}{F} \right) + \epsilon \quad (3.1)$$

där:

- R : Antal registrerade studenter på kursen.
- α : Maximal andel av registrerade studenter som kan tänkas närvara (vid kursens första föreläsningstillfälle, $f = 0$).
- β : Minskningstakt för närvaron över föreläsningstillfällen.
- f : Antal passerade föreläsningstillfällen.
- F : Totalt antal föreläsningstillfällen i kursen.
- ϵ : Slumpmässig störning för att simulera variation i närvaron.

5.1.2 Utdrag från TimeEdit

Med hjälp av en schemaläggare på högskolan kunde ett utdrag från TimeEdit göras för de tre föregående läsåren. Utdraget låg sedan till grund för en CSV-fil som kunde läsas in av programmet och representera alla bokade undervisningstillfällen. Ungefär 140 000 tillfällen ingick i utdragen, vilket gjorde det onödigt svårhanterat i prototypen. Som en lösning på det reducerades antalet tillfällen till 23 535 genom att endast kurser som var bokade i ett urval av salar togs med.

5.1.3 Simulering av information från Ladok

Alla kurser och kurstillfällen som ingick i TimeEdit-CSV:n sammanställdes i en JSON-fil. Informationen som ingick för varje kurs var kurskod och vilka kurstillfällen som hade gått. För varje kurstillfälle ingick ett genererat id, år, termin och läsperiod. Därtill gjordes en separat JSON med alla kurstillfällen tillsammans med hur många som var registrerade på respektive tillfälle. Antalet registrerade sattes till ett slumpat heltal mellan 20 och 250.

5.2 Prototyputvecklingsprocess

Inför och under utveckling av prototypen lades delvis fokus på att se till att dokumentation kring kravspecifikationen var utförligt beskriven, vilket innebar att funktionalitet definierat av krav kunde ges som en väl definierad uppgift i form av en user story. Utvecklingen gjordes i form av Scrum, en agil utvecklingsmetod, som gynnade projektets korta tidsram, och tillät högt prioriterad funktionalitet att implementeras först. Utvecklingen dokumenterades även i en kanbantavla vilket gjorde arbetet transparent och kunde följas löpande av resterande i arbetsgruppen som var tilldelade övriga uppgifter. Återkoppling kunde då ske effektivt efter en så kallad sprint, vilken pågick ett fåtal veckor, och resurser kunde fördelas utefter behov.

För versionshantering och samarbete i kodbasen användes *GitHub*, vilket möjliggjorde parallellt arbete, kodgranskning och spårbar historik över ändringar och tillägg. Genom nyttjande av plattformen kunde ny funktionalitet implementeras och integreras utan konflikter.

För att dokumentera API:et användes OpenAPI 3.0-struktur i kombination med Swagger. Användningen av dessa verktyg möjliggjorde ett webb-gränssnitt där API:et kunde testas direkt i en webbläsare, vilket underlättade utveckling, testning och framtida integration. Det tillåter också att API:et är enklare att förstå och kan vidareutvecklas på.

5.2.1 Användning av generativ AI

Under utveckling av prototypen användes generativ AI i mindre mån för att tidseffektivisera kod-processen genom exempelvis förslag för att avsluta ett kodstycke och gav förslag för att testa olika funktionaliteter inom koden. Användning skedde enligt riktlinjer och föreskrifter för lärosätet Chalmers tekniska högskola [24], och diskuterades med handledare och examinator inför- och under vår arbetsgång. Vi gav hänsyn till risken för upphovsrättsskyddad kod och såg till att kärnfunktionalitet för den prototyp vi utvecklade bestod av egna lösningar.

6 Utvärdering

Testning skedde både kontinuerligt under utvecklingen och genom slutliga tester för att verifiera att funktionalitet och prestanda var i enlighet med kravspecifikationen. Genom systematiska tester kunde potentiella brister identifieras och åtgärdas i ett tidigt skede, vilket minskade risken för fel i den slutliga produkten. Denna verifieringsprocess skedde genom löpande tester under den iterativa arbetsprocessen för varje separat arbetsuppgift under en sprint som utvecklingsprocessen tillämpade.

Detta möjliggjorde upptäckande och eliminering av fel tidigt, vilket gav löpande kvalitetssäkring för produkten.

V-modellen lämpar sig särskilt väl i projekt med tydligt definierade krav och en relativt stabil utvecklingsmiljö, och användes i detta arbete som en vägledande struktur för både planering och utvärdering av den utvecklade prototypen.

6.1 Enhetstester

Enhetstester utvecklades under implementationen för att isolerat kontrollera funktionaliteten hos de enskilda komponenterna. Fokus låg på logiska enheter såsom affärslogik och hjälpfunktioner. För att möjliggöra isolering användes mocking-verktyget *Moq*, vilket gjorde det möjligt att testa metoder utan beroenden till externa system såsom databaser eller externa API:er. Dessa tester kördes automatiskt vid varje ny kodändring och hjälpte till att identifiera problem tidigt i utvecklingscykeln, vilket förbättrade systemets stabilitet och pålitlighet.

6.2 Integrationstester

För att säkerställa att de olika komponenterna i systemet fungerade korrekt tillsammans genomfördes integrationstester. Syftet med dessa tester var att validera dataflödet mellan frontend, backend och databas – alltså att systemet inte bara fungerar i isolerade delar utan som en helhet.

Testerna genomfördes med hjälp av verktyget *WebApplicationFactory*, vilket är en del av testmiljön i ASP.NET. *WebApplicationFactory* gör det möjligt att skapa en in-memory-instans av hela webbtjänsten, så att HTTP-anrop kan simuleras precis som de skulle fungera i en verklig driftmiljö – men utan att faktiskt starta upp applikationen i en webbläsare eller använda produktionsdata.

Detta tillvägagångssätt möjliggör att testa till exempel att API-endpoints returnerar rätt HTTP-statuskoder, att rätt data returneras beroende på inmatad input och att olika systemkomponenter samverkar korrekt.

Testmiljön var konfigurerad med mockad testdata, så att resultaten kunde kontrolleras och var oberoende av faktisk driftdata. Detta bidrog till mer pålitliga och reproducerbara tester.

6.3 Systemtester

Systemtester genomfördes som en helhetsgranskning av prototypen för att verifiera att systemet uppfyllde de funktionella kraven. Dessa tester omfattade realistiska användarscenarier, exempelvis att hämta statistik, jämföra närvarodata mellan år och att visualisera deltagarantal för klienter. Testningen skedde manuellt och följde ett antal förutbestämda testfall. Resultaten dokumenterades och användes som grund för vidare förbättringar inför potentiella framtida iterationer. Genom denna testnivå kunde helhetsperspektivet säkerställas och användarupplevelsen utvärderas.

Kapitel 4

Resultat

Nedan redogörs för resultaten från projektets genomförande med fokus på identifierade behov, det valda konceptet och en utvärdering genom en prototyp.

1 Behovsanalys

De av arbetet identifierade målgrupperna är lokalansvariga, fastighetsägare, schemaläggare, lärare, programansvariga, TA:s, studenter, beslutsfattande roller på institutioner och säkerhetsansvariga.

1.1 Intervjuer med målgrupper

Ett målinriktat urval resulterade i tre intervjuer med representanter från följande målgrupper: lokalansvariga, schemaläggare, programansvariga och lärare. Dessa roller bedömdes som särskilt relevanta utifrån deras koppling till lokalvård, schemaläggning och kursutvärdering.

Samtliga målgrupper uttryckte att närvarostatistik skulle vara värdefull i deras arbete, men med olika syften och användningsområden. Den lokalansvarige betonade att statistik över studentflöden i lokalerna skulle kunna möjliggöra mer behovsanpassad lokalvård samt fungera som underlag vid diskussioner kring lokaloptimering. Han framhöll särskilt vikten av att statistiken är lättillgänglig och överskådlig, exempelvis genom användning av heatmap eller tydliga översikter.

Schemaläggaren framhöll att statistik kan användas för att se om lokaler verkligen används enligt plan. Denna information kan i sin tur ligga till grund för en mer effektiv resursfördelning. En särskilt viktig aspekt enligt henne var möjligheten att kunna jämföra bokad närvaro med faktisk närvaro, vilket är relevant vid kurser med låg beläggning.

Programansvarig, som även arbetar som lärare, lyfte fram att närvarodata kan vara ett värdefullt stöd vid kursutvärdering och planering, särskilt om den kan kopplas till scheman och specifika kurser. Han betonade samtidigt att data inte bör användas i övervakande syfte, utan i syfte att förbättra kvalitet och planering. Han

påpekade även att det inte är självklart att studenter skulle acceptera sensorer i grupprum, med hänvisning till att personal skulle motsätta sig att sensorer installerats i personalutrymmen och kontor. Dessutom framhöll han vikten av att inte dra långtgående slutsatser baserat enbart på närvarostatistik, såsom att använda den som argument för om en kurs fungerar eller inte. Exempelvis lyfte han risken att en bristfällig närvaro på en kurs skulle kunna åläggas studenterna. Avslutningsvis påpekade intervjupersonen att det bör finnas en ”röd kapp” som direkt kan stänga ner systemet vid pågående attack.

Samtliga intervjuobjekt var överens om att statistiken bör göras tillgänglig via en användarvänlig digital plattform, helst integrerad med befintliga system som QlikView, TimeEdit eller Canvas.

1.2 Studenters behov

De centrala önskemål som har identifierats utifrån enkätundersökningar bland studenter på högskolan [3], [23] är följande. Många studenter upplevde att det ofta är svårt att hitta tillgängliga grupprum, vilket skulle kunna bero på att rummen ibland är bokade utan att faktiskt användas. Det ligger till grund för önskemålet om att i realtid kunna se vilka grupprum som verkligen är lediga. Ett annat återkommande önskemål var att tjänsten bör integreras i befintliga plattformar. En majoriteten föredrog att funktionen skulle finnas i Kårappen, följt av Campus Maps och TimeEdit. Flera studenter svarade även att grupprummens storlek ofta upplevs som olämplig i förhållande till gruppens behov. Utöver detta framförde många en känsla av obehag kring att det skulle finnas sensorer installerade i grupprummen.

1.3 Etiska aspekter

Anonymisering av data samt personlig integritet är viktiga etiska aspekter att ta hänsyn till. Ingen individ ska på så vis kunna identifieras eller känna sig kränkt eller hotad av systemet. Därtill är datasäkerhet även en viktig aspekt, för att bevara en tilltro till systemet och att närvarodata ska skyddas. Vidare är även behörigheter viktigt, så att användare endast når den data de är behöriga till. Det är väsentligt att systemet hanterar alla dessa aspekter innan det släpps som produkt för att säkerställa att systemet följer etiska riktlinjer.

1.4 Övriga identifierade behov

Eftersom studenterna tycker att det ofta är svårt att hitta tillgängliga grupprum, finns ett behov av att systemet automatiskt avbokar grupprum som inte tagits i anspråk inom 30 minuter, i linje med Chalmers befintliga regelverk [25]. Prediktioner baserade på historisk data över grupprumsanvändning skulle också kunna hjälpa studenter att snabbare hitta lediga platser.

För lokalansvariga och beslutsfattare finns ett tydligt behov av att utvärdera hur effektivt lokalerna används. Det handlar om att mäta faktisk beläggning, alltså hur stor andel av tillgängliga platser som nyttjas, samt hur ofta lokaler bokas. Denna statistik kan synliggöra lokaler som är över- eller underutnyttjade och därmed fungera som grund för lokalvård, planering och optimerad bokning.

Det finns även ett behov av att följa utvecklingen över tid för att kunna utvärdera

effekter av förändringar. Jämförelse av beläggningsdata före och efter exempelvis ombyggnationer, förändrade rutiner eller införande av ett statistiksystem kan visa om åtgärderna haft önskad effekt. Det kan exempelvis handla om att schemaläggare optimerat lokalnyttjandet eller att studenter, genom ökad information, använder lokalerna mer effektivt.

1.5 Användarkrav

Följande användarkrav har identifierats:

- A1. Använda befintliga program för att utforska närvaroinformation**
Närvaroinformation, inklusive statistik och relevanta nyckeltal, ska kunna integreras i universitetets befintliga program och verktyg som redan används av användarna. Syftet är att göra informationen lättillgänglig där användaren redan arbetar, och därmed undvika behovet av ytterligare en applikation. Det möjliggör också att kombinera närvarostatistik med annan data som hanteras i dessa verktyg.
- A2. Se historiskt deltagande kopplat till kurser och kurstillfällen**
Användaren ska kunna se historisk närvarodata för specifika kurstillfällen samt få en överblick över närvarostatistik för hela kurser. Det ska även vara möjligt att analysera förhållandet mellan antalet antagna studenter och det faktiska deltagandet, liksom att jämföra deltagande mellan olika kurstillfällen och kurser. Detta krav är centralt för att ge programansvariga och lärare möjlighet att använda närvarodata som underlag vid kursutvärderingar. För lärarna innebär det även stöd i planering av föreläsningar, handledningspass och laborationer baserat på faktisk närvaro, snarare än registrerade antal eller erfarenhetsbaserade antaganden. Studierektorer och schemaläggare kan dessutom använda data för att optimera resursfördelning, särskilt för TA:s, samt för att fatta beslut kring lokalbokning och eventuell kontrollerad överbokning.
- A3. Se användning av en lokal över tid**
Systemet ska tillhandahålla möjlighet att analysera närvarodata baserat på lokal, vilket kan ge lokalansvariga insikter i hur ofta och hur intensivt olika lokaler används. Denna information ska kunna brytas ned på lokal-, byggnads- och campusnivå samt analyseras över olika tidsintervall.

- A4. Mäta lokalernas beläggningseffektivitet**
Systemet ska kunna mäta hur stor andel av platserna som faktiskt nyttjas i en lokal under en bokning. Denna information ska kunna brytas ned på lokal-, byggnads- och campusnivå samt analyseras över olika tidsintervall. Syftet är att ge lokalansvariga, schemaläggare och fastighetsförvaltning möjlighet att fatta datadrivna beslut kring lokaloptimering, resursfördelning och strategiska investeringar. Funktionen är även viktig för att följa upp om införandet av ett närvarosystem lett till förbättrat lokalnyttjande.
- A5. Se studentflöden på campus, byggnader och lokaler**
Användaren ska kunna se sammanställd statistik över studentflöden i byggnader och lokaler. Detta möjliggör mer behovsanpassad lokalvård samt utgör ett bättre underlag för diskussioner kring lokaloptimering och driftplanering. Statistiken ska vara översiktlig och förslagsvis inkludera en heatmap, som visar var flest folk har befunnit sig på campus.
- A6. Se beläggning i lokaler, grupprum och datorsalar i realtid**
Att kunna visa realtidsdata över beläggning i olika typer av utrymmen – såsom föreläsningssalar, handledningsrum, grupprum och datorsalar – bedöms ha stort potentiellt värde. Denna funktion skiljer sig från Find A Room genom att realtidsdata kan kopplas till schemat, vilket gör det möjligt för användaren att söka efter närvarodata i specifika lokaler baserat på kurs. För grupprum finns dessutom möjlighet till automatisk avbokning om rummet inte tas i anspråk inom en viss tid.
För studenter innebär detta att de enklare kan identifiera var det finns lediga platser, medans för lärare och TA:s möjliggör funktionen prioritering av insatser, vid till exempel handledning, baserat på var flest studenter befinner sig.
- A7. Prediktion av lediga grupprum baserat på historisk data**
Systemet ska kunna analysera historisk användningsdata för grupprum och ge sannolikhetsbaserade uppskattningar om när ett rum sannolikt kommer bli ledigt i en byggnad. Informationen ska kunna hjälpa studenter att snabbare hitta tillgängliga studieplatser, när samtliga grupprum är bokade.
- A8. Möjlighet att stänga ned systemet omedelbart**
Systemet ska innehålla en funktion för omedelbar avstängning i händelse av kritiska fel eller allvarliga säkerhetsincidenter. Funktionen ska vara avgränsad, men lättillgänglig för systemadministratörer, säkerhetsansvariga eller annan ansvarig.

- A9. Behörighetssystem**
Systemet ska innefatta ett flexibelt och rollbaserat behörighetssystem för att säkerställa att endast auktoriserade användare får tillgång till specifika delar av systemet som de har behov av. Behörighetssystemet ska vara kopplat till ett centralt rollsystem för högskolan, för att säkerställa att åtkomstnivåer alltid speglar aktuella behörigheter enligt organisationens strukturer. Administratörer ska kunna hantera alla behörighetsnivåer och snabbt kunna justera åtkomsträttigheter vid behov.
- A10. Skyddad data**
Närvarostatistik och annan data ska vara skyddad och endast åtkomlig för behöriga användare. Detta för att data endast ska kunna användas i det syfte som systemet ämnar, för att minimera risken för missbruk och säkerställa personlig integritet.
- A11. Anonymiserad data**
Data kopplat till närvaro ska vara anonym, med hänsyn till den personlig integritet.
- A12. Information om hur data används**
Det ska framgå tydligt vilken data som samlas in, hur den används och i vilket syfte. Syftet är att upprätthålla förtroende och motverka känslan av övervakning bland studenter och personal.

2 Koncept

Konceptgenereringen resulterade i ett koncept som innebär att information ska göras tillgängligt via ett API och därigenom integreras i olika visualiseringsverktyg. Data hämtas direkt från datakällorna när behovet uppstår, såsom sensordatabaser och studieadministrationsprogram, istället för att samla all data i en separat databas. Detta innebär även att inget fristående och helintegrerat system med egen frontend utvecklas.

Det valda konceptet resulterade i en slutgiltig kravspecifikation som visas i tabell C.1. Kraven är prioriterade enligt MoSCoW-modellen och saknar målvärde. För en kravspecifikation med beskrivningarna i sin helhet se bilaga C.

Tabell 4.1: De funktionella kraven i den slutgiltiga kravspecifikationen tillsammans med respektive kravs prioritering enligt MoSCoW. Observera att målvärde saknas för samtliga krav.

Funktionella krav	
F1 Integrera data i andra system och verktyg	Must
F2 Se historiskt deltagande kopplat till kurser och kurstillfällen	Should
F3 Se användning av en lokal över tid	Could
F4 Mäta lokalernas beläggningseffektivitet	Should
F5 Se studentflöden på campus, byggnader och lokaler	Could
F6 Se beläggning i lokaler, grupprum och datorsalar i realtid	Could
F7 Prediktion av lediga grupprum baserat på historisk data	Won't
F8 Möjlighet att stänga ned systemet omedelbart	Must
F9 Behörighetssystem	Must
F10 Information om hur data används	Must

Tabell 4.2: De icke-funktionella kraven i den slutgiltiga kravspecifikationen tillsammans med respektive kravs prioritering enligt MoSCoW. Observera att målvärde saknas för samtliga krav.

Icke-funktionella krav	
I1 Säkerhet: data är skyddad	Must
I2 Underhåll: lätt att underhålla, anpassa	Must
I3 Prestanda: Snabb hämtning av data från API:n	Must
I4 Pålitlig: Information stämmer, systemet är konsekvent	Must
I5 Skalbarhet: lätt att bygga på	Must
I6 Användarvänlighet: lätt att använda	Must
I7 Anonymiserad data	Must

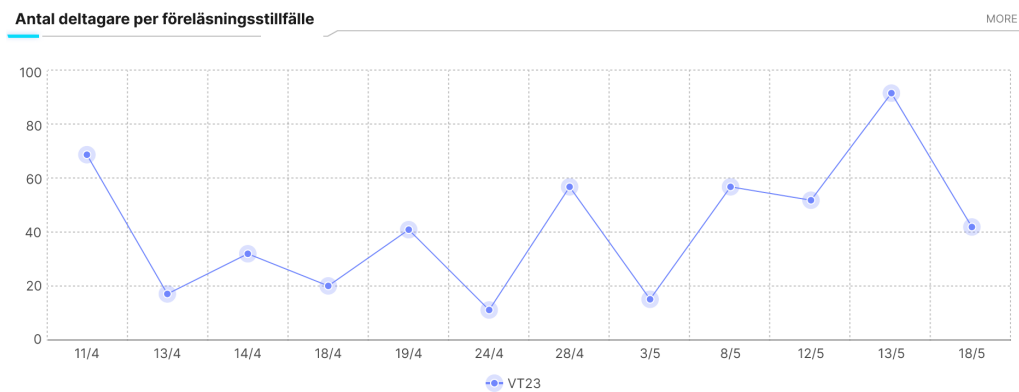
Systemet som ingår i det valda konceptet syftar till att tillgängliggöra närvarodata på ett sätt som stödjer både undervisningsplanering, kursutvärdering och resursfördelning. Systemet möjliggör att användare, såsom lärare, schemaläggare och programansvariga, kan se historisk närvaro kopplad till specifika kurser och kurstillfällen. Det är även möjligt att få en överblick på kursnivå, jämföra deltagande

mellan olika tillfällen, och analysera relationen mellan antalet registrerade studenter och faktisk närvaro.

Utöver kursbaserad statistik stödjer systemet också analys av lokalers användning över tid. Lokalansvariga kan se hur ofta och hur intensivt olika rum används, samt utvärdera belägningsgraden i förhållande till lokalernas kapacitet. Systemet visualiserar studentflöden i byggnader och på campusnivå, vilket ger möjlighet för effektivare planering av lokalvård och underhåll.

Utöver historisk statistik stödjer systemet även visualisering av lokalers beläggning i realtid. Användare kan snabbt identifiera lediga rum, även om de är bokade men inte används, vilket underlättar både spontana studier och effektiv resursanvändning. För studenter innebär detta ökad tillgång till grupprum och arbetsytor. För lärare och TA:s innebär det att de under undervisningspass med ett flertal bokade lokaler kan se vart studenter befinner sig och placeras utifrån behov.

Vald kurs: **TDA357** Kurstillfälle: VT24, lp3



Föreläsningar

Antal deltagare:
 Högst: 80 personer (90%)
 Genomsnitt: 60 personer (83%)

Laborationer

Antal deltagare:
 Högst: 50 personer (57%)
 Genomsnitt: 30 personer (34%)

63% mindre deltagande i laborationer jämfört med föreläsningar.

50% mindre deltagande efter halva kursen.

Visa detaljerad data

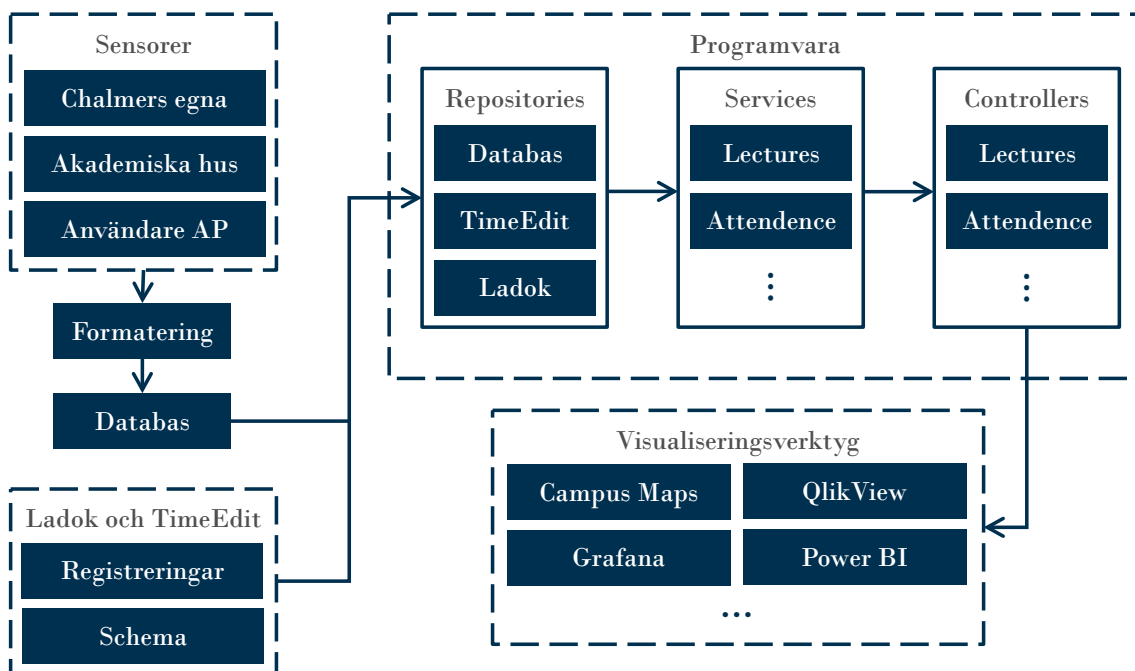


Figur 4.1: Skissen visar ett möjligt sätt att realisera att användaren ska kunna se historiskt deltagande av ett kurstillfälle, vilket är en del av kravet F2: *Se historiskt deltagande kopplat till kurser och kurstillfällen.*

2.1 Systemdesign

Systemet är utformat för att fungera tillsammans med befintliga plattformar och verktyg, vilket minimerar behovet av ytterligare applikationer. Informationen ska presenteras på ett sätt som är lättförståeligt och relevant för varje målgrupp, med nyckeltal och statistik som kan kombineras med annan tillgänglig data, som exempelvis i figur 4.1. Fler exempel finns i bilaga B.

För att säkerställa att endast behöriga användare har åtkomst till systemets data används ett externt behörighetssystem, vilket gör att användarhanteringen blir konsekvent på högskolan. Genom systemet kan användare identifieras och kopplas till sina respektive roller på Chalmers. Genom detta får användaren enbart tillgång till en begränsad del av samtlig information, anpassad efter användarens roll.



Figur 4.2: Illustration av informationsflödet i konceptet. Data kommer från sensorer, Ladok och TimeEdit och hämtas av en programvara som i sin tur gör beräkningar och kombinerar de olika källorna. Programvaran gör sedan beräkningarna tillgängliga att hämtas av olika visualiseringsverktyg.

Systemarkitekturen har utformats för att realisera dessa krav genom en modulär struktur med fokus på insamling, bearbetning och tillgängliggörande av data via ett API. Designen möjliggör återanvändning av data i olika gränssnitt och underlättar integrering i existerande digitala miljöer på Chalmers. figur 4.2 visar en översikt av arkitekturen, som består av flera huvudkomponenter och dataflöden.

Systemet grundas på insamling av data från olika sensorer, såsom Chalmers egna sensorer, sensorer från fastighetsägaren Akademiska Hus och antalet användare av det trådlösa nätverket. Sensordata skickas till ett API som ansvarar för att formatera och lägga till metadata, såsom ID på rum och byggnad, innan det lagras i en databas.

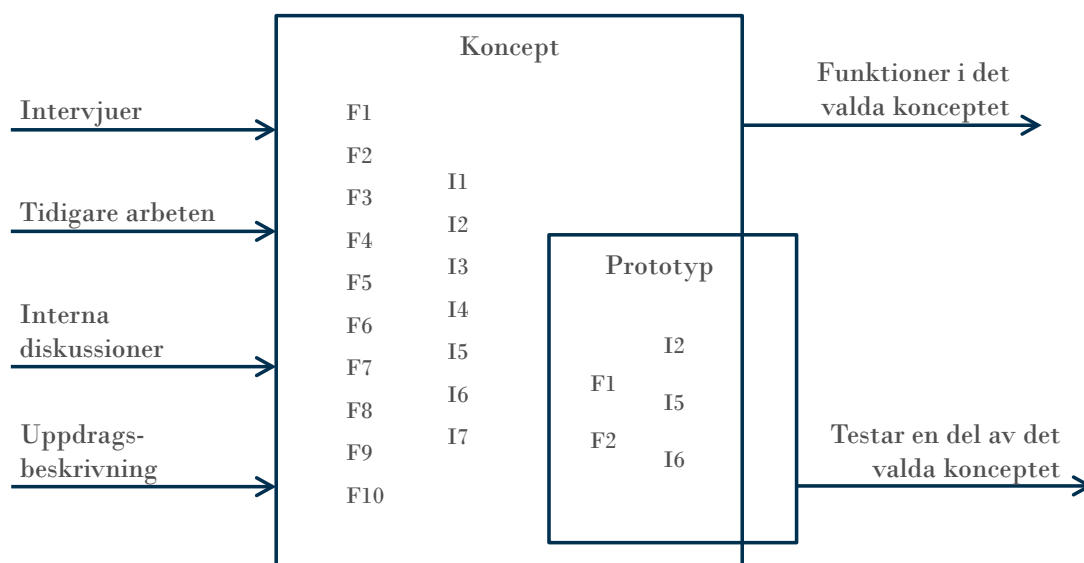
Utöver sensordata integrerar systemet data från de externa källorna Ladok och TimeEdit. Ladok tillhandahåller data om vilka kurser som går och registreringar på dessa, medan TimeEdit tillhandahåller schemainformation.

Repository-Service-Controller-mönstret används för att uppfylla krav I5 *skalbarhet* och I2 *Underhåll*. Mönstret säkerställer att strukturen på det interna systemet är lättförståelig utan konstant kommunikation för nuvarande och framtida utvecklare.

Slutligen finns det olika komponenter som kan användas för att visualisera data för slutanvändarna. Dessa inkluderar verktyg som Campus Maps, QlikView, Grafana och Power BI. Dessa verktyg gör det möjligt för användare som schemaläggare, programansvariga, lokalvårdare och studenter att interagera med och analysera data på ett sätt som är anpassat till deras specifika behov. Till exempel kan data från närvaromätning i grupprum integreras i Campus Maps för att ge studenter en översikt över vilka grupprum som är lediga just nu. Samtidigt kan en schemaläggare få in data i till exempel QlikView där de redan har annan data för att på så sätt integrera även närvarodata.

3 Prototyp

Utöver den originella avgränsningen av målgrupper har målgrupperna avgränsats återigen för att specificera just de målgrupper som prototypens funktionalitet ska riktas emot. Konzeptets målgrupper avgränsades till endast de målgrupper som har behov av funktionerna som uppfyller de högst prioriterade kraven. Dessa målgrupper innehåller Lärare, Schemaläggare, Programansvariga och institutionen. Alla dessa målgrupper efterfrågade *Integrering av data i andra system och verktyg* vilket var krav F1. De har även främst nytta av krav F2 *Se historiskt deltagande kopplat till kurser och kurstillfällen*.



Figur 4.3: Prototypen testar en begränsad del av alla krav.

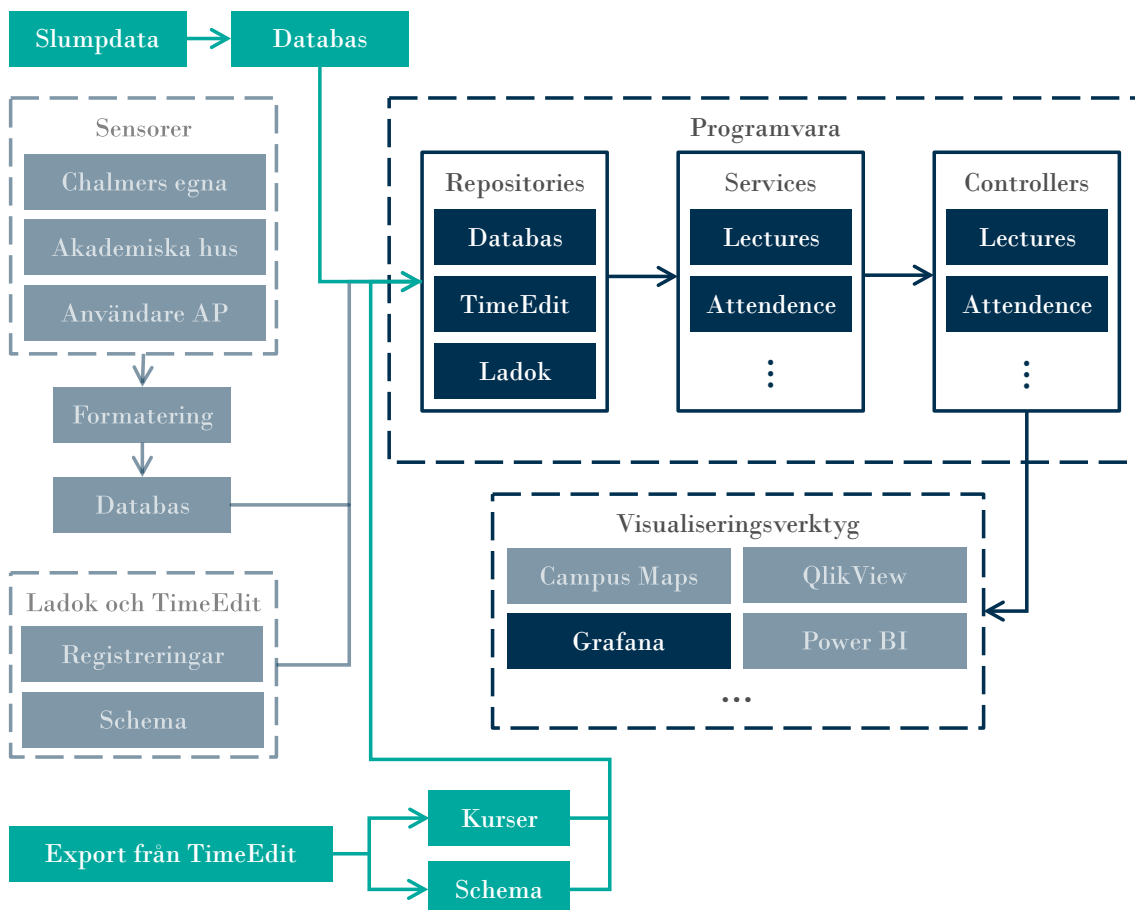
3.1 Avgränsning av krav

Följande krav valdes för att implementeras i prototypen, se figur 4.3.

- F1 Integrering av data i andra system och verktyg
- F2 Se historiskt deltagande kopplat till kurser och kurstillfällen
- I2 Underhåll: lätt att underhålla, anpassa
- I5 Skalbarhet: Lätt att bygga ut och integrera
- I6 Användarvänlighet: Lätt att använda och integrera

3.2 Systemarkitektur i prototypen

Prototypen är uppbyggd på samma sätt som den övergripande systemarkitekturen, med vissa undantag som beskrivs i kapitel 3 avsnitt 5.1.



Figur 4.4: Illustration av informationsflödet i prototypen. Anpassningar efter begränsningar är ritade i turkos medan delar av konceptet som inte används till prototypen är grå-markerade. Jämför figur 4.2.

3.3 API

Systemet inkluderar ett antal API-endpoints som möjliggör åtkomst till resurser såsom kurser, föreläsningar, närvaro och lokaler. En fullständig översikt över alla endpoints, finns i bilaga D.

Närvarodata kan hämtas på flera nivåer. Det finns endpoints för att hämta sammanställd närvarostatistik för ett specifikt kurstillfälle, samt detaljerad närvaro per föreläsning inom ett sådant tillfälle. Dessutom kan man hämta alla kurstillfällen med föreläsningar för en viss kurs.

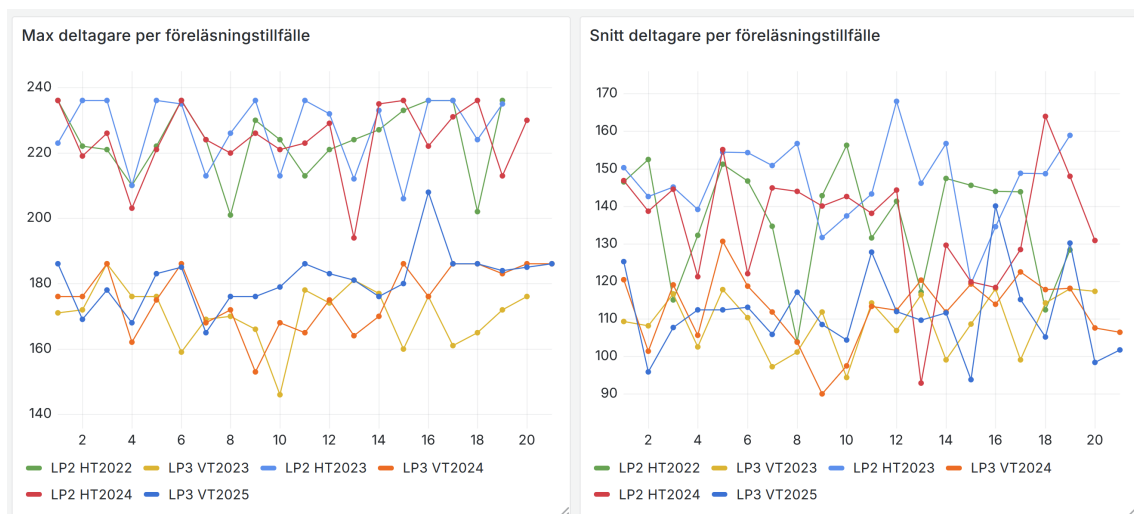
För kursrelaterad information finns endpoints som returnerar alla kurser, information om en specifik kurs, samt alla kurstillfällen kopplade till en viss kurs. Föreläsningar hanteras genom endpoints som tillåter hämtning av samtliga föreläsningar eller enskilda föreläsningar baserat på ID. På liknande sätt finns endpoints för att hämta alla kurstillfällen, ett specifikt kurstillfälle, samt alla föreläsningar kopplade till ett visst tillfälle.

Även lokaler hanteras via API:et, där man kan hämta en lista över alla lokaler eller detaljerad information om en specifik lokal.

Samtliga endpoints returnerar strukturerad JSON-data och följer etablerade REST-konventioner. Felhantering sker med hjälp av standardiserade HTTP-statuskoder, däribland 404 Not Found för saknade resurser, 408 Request Timeout vid långsam serverrespons, samt 500 Internal Server Error vid oväntade fel på serversidan.

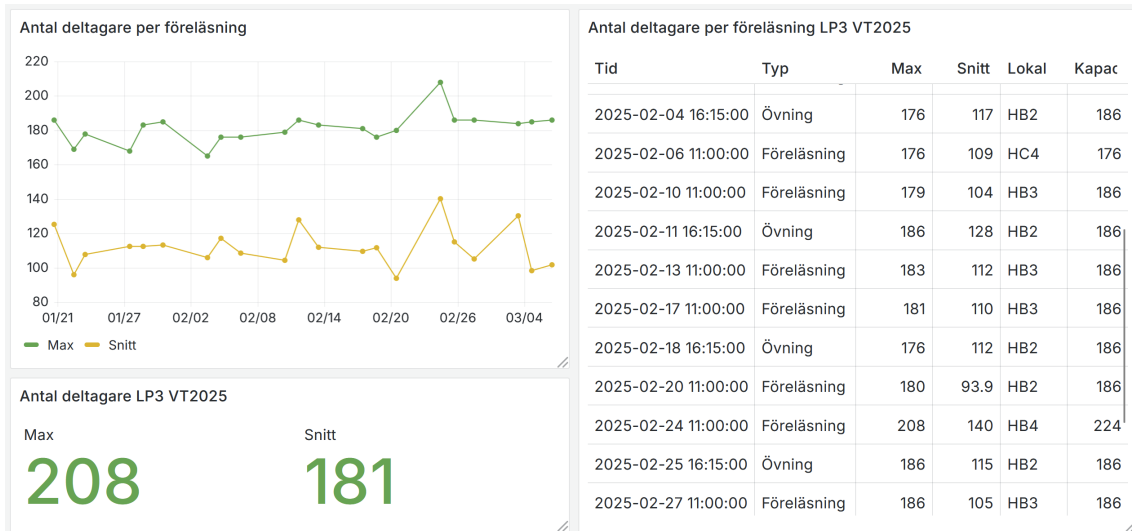
3.4 Exempel på frontend-integration

Gränssnittet i Grafana visualiserar data i form av paneler, grafer och tabeller. Det består av tre huvuddelar. En översiktsvy visar närvaron vid varje föreläsningstillfälle i en kurs, jämfört med motsvarande tillfälle under tidigare år. Detta gör det möjligt att analysera trender och förändringar i studentdeltagande över tid. Vyn visas i figur 4.5.

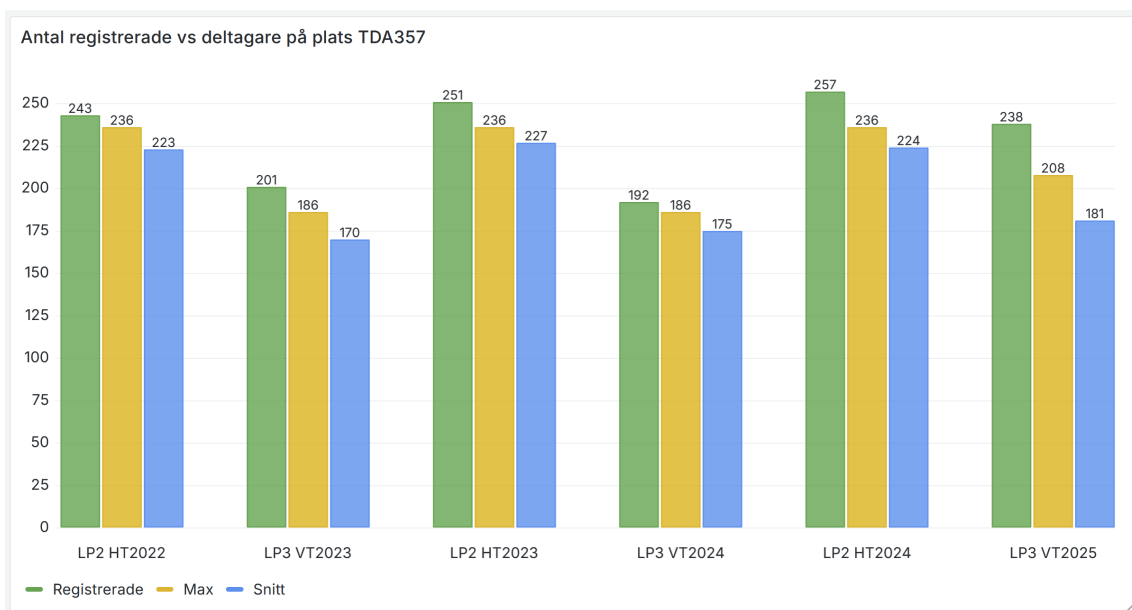


Figur 4.5: Översiktsvy för en kurs med flera kurstillfällen.

Vidare finns en detaljvy för ett specifikt kurstillfälle som visar detaljerad information om varje föreläsningens deltagarstatistik. Den visas i figur 4.6. En planeringssvy visar avslutningsvis antal antagna studenter till en kurs i relation till faktiskt deltagande. Här presenteras både max- och medelantalet deltagare, vilket ger en bild av kursens utnyttjandegrad. Se figur 4.7.



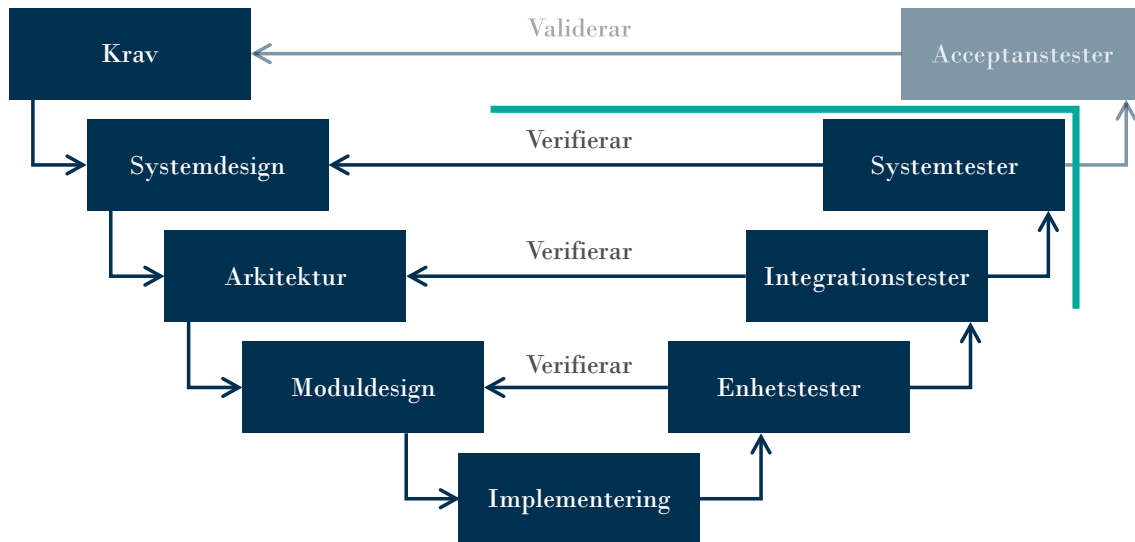
Figur 4.6: Detaljvy för ett specifikt kurstillfälle med information om varje undervisningstillfälle. Observera att snitt för hela kurstillfället (181, angivet längst ner) är baserat på maximalt antal deltagare per föreläsning. Snitt per föreläsning baseras däremot på deltagandet under föreläsningstiden och eventuell paus är inkluderad, vilket gör att det värdet är mycket lägre.



Figur 4.7: Vy med antagningsstatistik i förhållande till närvaroinformation.

4 Utvärdering av koncept

Den utvecklade prototypen har använts för att utvärdera konceptet. Utvärderingen har utförts enligt V-modellen till och med nivån för systemtester, se figur 4.8. Därmed har verifieringen genomförts, men inte valideringen i ett skarpt användarsammanhang.



Figur 4.8: Prototypen är utvärderad enligt V-modellen upp till och med systemtesterna. Således är alla verifieringssteg genomförda till skillnad från valideringssteget.

4.0.1 Enhetstester

Enhetstesterna visade att logiken i systemets centrala funktioner fungerade korrekt för olika typer av indata. Fel i filtreringen av rum och hanteringen av närvarodata kunde upptäckas tidigt och åtgärdades innan de påverkade andra delar av systemet. Den tidiga upptäckten av regressionsfel förbättrade systemets stabilitet över tid.

4.0.2 Integrationstester

Testerna visade att systemets olika komponenter fungerade korrekt tillsammans. Bland annat verifierades att API:ets endpoints svarade som förväntat, att svaren innehöll relevant data som var strukturerat på ett korrekt sätt och att logiken var oberoende av de underliggande databasstrukturerna. Inga allvarliga brister i interaktionen mellan modulerna kunde identifieras.

4.0.3 Systemtester

Systemtesterna speglade verkliga användarflöden och fokuserade på att säkerställa att data från API:et kunde visualiseras på ett tydligt och användarvänligt sätt i gränssnittet. Testningen visade att prototypen fungerade stabilt i sin helhet och uppfyllde de funktionella kraven.

Kapitel 5

Diskussion

Detta arbete har genomförts inom ramen för Chalmers tekniska högskola. Detta innebär att de slutsatser som dragits riskerar att felaktigt generaliseras till att gälla alla lärosäten, trots att projektet endast undersökt en lösning specifik för Chalmers. Det är troligt att olika lärosäten har varierad utformning på deras organisationer, skiftande tekniska förutsättningar eller helt enkelt andra behov, vilket kan påverka både relevansen och genomförbarheten av det föreslagna systemet. Samtidigt är det värt att notera att ett API-baserat system är mycket flexibelt och kan anpassas till olika miljöer, vilket minskar risken för att lösningen blir alltför specifik kring Chalmers. Dock behöver ett API inte nödvändigtvis vara den mest lämpliga lösningen i alla kontexter. Det kan uppstå integrationsproblem om ett lärosäte redan har låsta system kopplade till sensorer, behörigheter och roller, eller om det till och med helt saknas verktyg där det är lämpligt att visualisera närvarodata. I så fall kan ett fristående, helintegrerat program vara mer ändamålsenligt.

Valet att utveckla ett API i detta projekt grundades utifrån behovsanalysen, där målgrupper uttryckte en önskan om att undvika ännu ett nytt verktyg att interagera med. På högskolor där sådana verktyg saknas eller där användarna har andra preferenser kan ett fristående system vara att föredra. På så vis är det viktigt att valet av koncept anpassas efter den specifika organisationen. En fördel i detta projekt är att programvaran har strukturerats enligt designmönstret *Repository-Service-Controller*. Det innebär att varje del av systemet har ett tydligt definierat ansvar, vilket i sin tur gör det enklare att anpassa och vidareutveckla. Till exempel kan en *frontend* enkelt bytas ut eller ersättas med en mer integrerad lösning utan att *backend* behöver skrivas om. Denna flexibilitet är särskilt värdefull om systemet i framtiden ska omformas till en helintegrerad applikation. För att undersöka mer kring vilka sätt ett koncept kan utformas för att vara en lösning som går att kategoriskt appliceras på lärosäten rekommenderas ytterligare studier med fokus på andra universitet och deras specifika förutsättningar.

En bredare målgrupp i behovsanalysen hade kunnat ge ytterligare perspektiv och insikter kring användarbehov och potentiella användningsområden för systemet. Det

hade varit intressant att inkludera samtliga identifierade målgrupper i behovsanalysen och eventuellt intressenter från andra institutioner eller administrativa funktioner som inte direkt berörs av lokalplanering eller undervisning. Samtidigt representerar de intervjuade personerna flera olika delar av organisationen, från schemaläggning och lokalansvar till programledning och undervisning, vilket ger en god förutsättning för olika perspektiv. Vår bedömning är alltså att vi ändå har identifierat de mest centrala behoven hos målgrupperna som är mest relevanta för systemets syfte.

En annan aspekt som förtjänar vidare reflektion är avsaknaden av målvärden och verifieringsmetoder i kravspecifikationen. Detta utgör en svaghet i resultatet, eftersom det försvårar möjligheten att objektivt utvärdera om ett krav faktiskt är uppfyllt. Samtidigt har vi i detta skede av projektet inte haft tillräcklig information för att kunna formulera rimliga och välgrundade målvärden. Exempelvis behöver antalet användare och mängden data uppskattas för att kunna sätta ett målvärde på prestanda. Det är därför möjligt att argumentera för att det inte är nödvändigt att fastställa målvärden redan i ett tidigt utvecklingsstadium, särskilt då arbetet främst syftat till att ta fram ett koncept och en prototyp. Däremot är det något som bör undersökas vidare i framtida iterationer, särskilt inför implementation i skarp miljö, där tydliga verifieringskriterier blir avgörande för kvalitetssäkring och fortsatt utveckling.

Vid utveckling av prototypen valde vi att fokusera på ett begränsat antal funktioner, snarare än att försöka implementera en stor mängd krav för att försöka nå hög kvalitet på prototypen. Detta tillvägagångssätt möjliggjorde en mer genomarbetad och stabil lösning, där de utvalda funktionerna kunde testas och utvärderas ordentligt. Samtidigt innebär detta en begränsning: eftersom alla krav inte testades i prototypen är det svårt att fullt ut avgöra vilka funktioner som är mest värdefulla för användarna. Det finns en risk för att vissa behov underskattats eller förbises, särskilt om de inte prioriterades i den initiala kravanalysen. För framtida utveckling vore det därför värdefullt att successivt utöka prototypen med fler funktioner och dessutom genomföra användartester för att validera och omvärdera kravens relevans i praktiken.

1 Etiska aspekter

En central etisk utmaning med systemet är risken för att information om deltagande kan korsrefereras med extern information, exempelvis scheman eller kursplaner, vilket potentiellt kan avslöja var en individ befunnit sig vid en viss tidpunkt. Även om systemet i sig inte innehåller information som kan användas för att identifiera personer, garanterar det inte anonymitet. I kombination med andra datakällor kan det dras slutsatser om var individer har befunnit sig eller troligen kommer att befinna sig. Därför är det inte bara viktigt att hantera behörigheter till intern närvarodata utan också se över behörigheter till data kopplat till universitet som inte är inkluderad i systemet, såsom kursregistreringar. Det räcker alltså inte att titta enbart på själva systemet utan även på vilka konsekvenser det kan få i en större kontext.

Ett sätt att minska riskerna är att arbeta med aggregerad statistik, där data sammanställs på en nivå som inte möjliggör spårning av individer eller specifika händelser. Sådan data är mindre känslig och kan därför göras tillgänglig för fler roller inom organisationen. Däremot föreslås all tillgång till detaljerad närvarodata begränsas till personer som genomgått säkerhetsprövning eller har särskild behörighet. Det är avgörande att säkerställa att endast de med ett legitimt behov får tillgång till känslig information, och att det finns tydliga rutiner för hur denna åtkomst hanteras och dokumenteras.

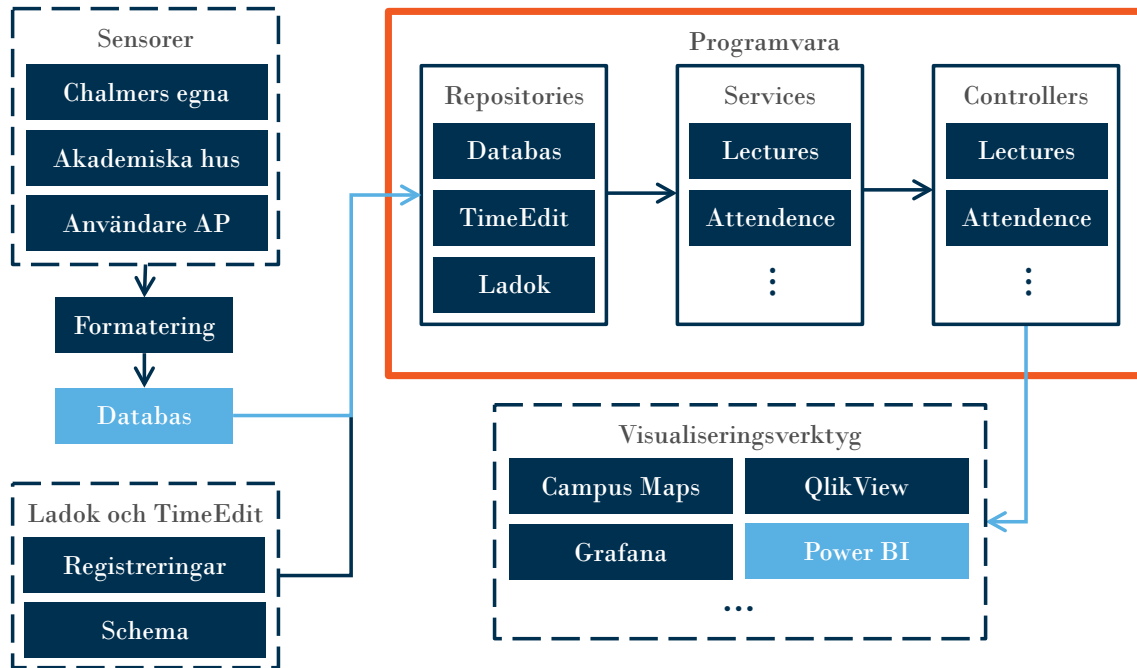
Även utan direkt identifierbar information kan systemets data användas för att förutsäga var personer sannolikt kommer att befinna sig, särskilt om det kombineras med historiska mönster eller extern information. Detta öppnar upp för allvarliga risker, där illasinnade aktörer kan använda data för att planera brott, såsom övergrepp, inbrott eller stölder, genom att till exempel identifiera tider då lokaler är tomma eller när väktare inte är närvarande. Historisk data blir därmed särskilt känslig, eftersom den kan användas för att bygga prediktiva modeller. Även realtidsdata är känslig, inte minst eftersom den i praktiken kan samlas in över tid via *web scraping* och därmed omvandlas till historisk data.

Oavsett hur väl systemet är utformat finns emellertid alltid en risk att data läcker, antingen genom brister i programvaran eller genom angrepp. Den enda säkra metoden för att helt undvika dessa risker är att inte samla in data alls. Då det är en förutsättning för att ett sådant här system ska fungera är det givetvis inte möjligt, men trots det är tanken intressant att ta upp för diskussion. Det vore även värdefullt att genomföra en särskild studie som undersöker vilken typ av information som faktiskt kan extraheras genom att korsreferera systemets data med andra källor. En sådan studie skulle ge bättre underlag för att bedöma risknivåer och utforma lämpliga skyddsåtgärder.

En annan aspekt är den psykologiska påverkan som datainsamling kan ha på studenter och personal. Även om det är anonym data, kvarstår risken att studenter och personal upplever en känsla av övervakning, då exempelvis flera studenter inte kände sig bekväma med sensorer. Bland lärarna kan det finnas en oro för att närvarostatistik kan användas som ett prestationsmått, vilket i sin tur kan skapa press och påverka undervisningen. Därför bör systemet inte enbart utvärderas utifrån funktionell nytta, utan även hur det uppfattas av personal och studenter. Trots att ett system kan bidra till förbättring i mätbara nyckeltal, som resursutnyttjande eller lokaloptimering, kan det samtidigt ha en negativ påverkan på mer abstrakta värden som arbetsmiljö, tillit och psykologisk trygghet. Det innebär att nettoeffekten av systemet potentiellt inte är så entydigt positiv som den kan framstå i kvantitativa analyser. För att uppnå verklig förbättring krävs därför en helhetssyn där både konkreta och abstrakta värden vägs in i utvärderingen.

2 Förslag till framtida arbete

För att kunna genomföra en mer tillförlitlig utvärdering av det föreslagna konceptet vore det värdefullt att vidareutveckla prototypen. Detta arbete består av två huvud-



Figur 5.1: Systemdesign i konceptet där programvaran är markerad i orange och en integration från databasen till visualisering i Power BI.

sakliga delar. Den första delen handlar om att vidareutveckla själva programvaran, se orange markering i figur 5.1. Eftersom programvaran är fristående från externa systemkomponenter med hjälp av anpassningarna i prototypen, är den relativt enkelt att bygga vidare på. Det gör den till ett bra verktyg för att visa på konceptets potential för projektets intressenter, vilket skulle kunna möjliggöra framtida investeringar i projektet. Den andra delen handlar om att integrera programvaran med de databaser och system som används i praktiken. Ett exempel skulle vara att integrera hela vägen från sensordata i Find A Room-databasen till att ett nyckeltal visualiseras i Power BI, se ljusblåa markeringar i figur 5.1. En svaghet i den nuvarande metoden är nämligen att prototypen bygger på testdata. Detta var förvisso ett rimligt val givet att vi inte hade tillgång till den riktiga databasen, men trots det begränsas möjligheten att dra slutsatser om systemets potential i praktiken. Att implementera en integration genom hela systemet möjliggör genomförandet av acceptanstester som kan visa på ifall konceptet faktiskt har potential. Sammanfattningsvis skulle vidare arbete med båda delarna av prototypen skapa bättre förutsättningar för en mer heltäckande utvärdering av konceptet.

Ett annat intressant område för framtida arbete är att undersöka mer hur närvarodata bäst kan visualiseras för olika typer av analyser. Det kan handla om att jämföra olika diagramtyper och utvärdera vilka som bäst stödjer tolkning av trender, jämförelser eller avvikelser kring närvaro, nyttjandegrad och andra nyckeltal. Det skulle också vara intressant att studera hur olika visualiseringsprinciper påverkar användarens förståelse och om det påverkar beslut. Därtill skulle framtida studier av specifika visualiseringsverktyg inkludera en jämförelse mellan exempelvis Grafana och Power BI för att se hur väl de lämpar sig för konceptet.

Kapitel 6

Slutsats

Under arbetets gång har tre frågeställningar undersökts genom att utveckla ett koncept som uppfyller syftet med arbetet. Utifrån det kan följande slutsatser dras.

1 Vilken närvarostatistik är relevant för olika målgrupper på en högskola?

Behovsanalysen visar att olika målgrupper inom högskolan har skilda behov när det gäller närvarostatistik, beroende på deras roller och ansvar. För programansvariga, lärare, schemaläggare och beslutsfattande roller på institutioner är det avgörande att kunna se historisk närvarodata kopplad till specifika kurstillfällen. Detta möjliggör utvärdering av studentengagemang över tid samt förbättring av kursupplägg och schemaläggning. Även förändringar i närvaro under kursens gång är av intresse för dessa grupper, då det kan indikera problem med kursstruktur, undervisningskvalitet eller studenters arbetsbelastning.

Lärare, lärarassistenter och studenter själva har behov av realtidsstatistik i grupprum och datorsalar under övningspass, vilket kan stödja planering och resurstilldelning under undervisningstillfällen. Studenter efterfrågar också information om vilka grupprum som inte nyttjas trots att de är bokade, för att kunna nyttja utrymmen mer effektivt. Lokalansvariga behöver statistik över studentflöden i campusmiljön för att optimera lokalanvändning, säkerhet och underhållsplanering.

2 Hur kan insamlad data göras tillgängligt för användare på ett effektivt vis?

Baserat på behovsanalysen som genomförts i detta arbete och i föregående kandidatarbeten ansågs ett API vara den mest effektiva lösningen. Slutsatsen grundar sig på att målgruppen inte önskade någon ny programvara att interagera med. De menade att de redan använde flera olika system i sitt arbetsflöde, och att ytterligare ett verktyg skulle öka komplexiteten och därmed minska effektiviteten. Genom att

utveckla ett API och leverera statistiken via detta undviks just detta problem utan att kompromissa funktionaliteten.

3 Vilka etiska aspekter bör tas hänsyn till vid hantering och visualisering av närvarodata och hur kan de aspekterna hanteras?

Arbetets undersökning resulterade i att det fanns tre etiska aspekter som berör projektet. Till att börja med innebar anonymitet att säkerställa att individer inte kan identifieras med närvarodata eller i kombination med extern data. Detta utförs med hjälp av ett behörighetssystem samt aggregering av data över längre perioder av tid. Vidare berörde integritetsaspekten att undvika användarens känsla av att vara övervakad både av institution och chefer. Detta kan förebyggas via behörighetskrav av sådan data, transparens kring datainsamlingsmetoder samt regler emot att använda närvarostatistik för utvärdering av läraryrket. Sist medför datasäkerhetsaspekten att läckage av data måste undvikas samt att det är viktigt att behålla systemets trovärdighet. Detta åtgärdas genom att hålla data skyddat samt att ett behörighetssystem införs.

Litteratur

- [1] Chalmers tekniska högskola, *Förbereder för fler studenter på Johanneberg och färre på Lindholmen*, Hämtad 2025-02-03, 2024. URL: <https://www.chalmers.se/aktuellt/nyheter/chalmers-forbereder-for-fler-studenter-pa-johanneberg-och-farre-pa-lindholmen/>.
- [2] A. Belle Nylén, J. Färenmark, J. Sandberg, A. Svedberg, H. Thörn och A. Wickman, "A Comparative Study of People Counting Technologies," 2023. URL: <https://chalmers.instructure.com/courses/22323/files/2764447?wrap=1>.
- [3] G. Asplund, E. Dahlström, O. Eriksson, L. Hakeskog, I. Nordin och H. Olsson, "Effektiv närvarodetektering och närvaroräkning för optimerad lokalanvändning," 2024. URL: <http://hdl.handle.net/20.500.12380/308653>.
- [4] A. Gustafsson och S. Park, "MIND THE GAPS - TILLFÄLLIG KARAKTÄR OCH SAMUTNYTTJANDE I EN STRATEGI FÖR EFFEKTIV LOKALANVÄNDNING," examensarb., Chalmers tekniska högskola, 2015.
- [5] McGraw-Hill, "Kanban," i *McGraw-Hill Dictionary of Scientific and Technical Terms*, 6. utg., Entry under "Kanban", industrial engineering term, New York: McGraw-Hill, 2003.
- [6] A. Dotis-Georgiou. "Getting Started: Writing Data to InfluxDB." Hämtad 2025-05-13. (2020), URL: <https://www.influxdata.com/blog/getting-started-writing-data-to-influxdb/>.
- [7] A. Butterfield, G. E. Ngondi och A. Kerr, *API*, 2016. DOI: 10.1093/acref/9780199688975.013.0160. URL: <https://www.oxfordreference.com/view/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-160>.
- [8] M. G. Dan Nosowitz. "What is an API endpoint?" Hämtad 2025-04-28. (2024), URL: <https://www.ibm.com/think/topics/api-endpoint>.
- [9] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Hämtad 2025-05-08, Ph.D. dissertation, University of California, Irvine, 2000. URL: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
- [10] Ecma International, *The JSON Data Interchange Format*, <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>, 1st Edition, October 2013, 2013.
- [11] K. C. Kapur, "Reliability, availability, and maintainability," *AccessScience*, 2020, Hämtad 2025-05-08. DOI: 10.1036/1097-8542.580500.

- [12] P. Gupta, "Software testing," *AccessScience*, 2024, Hämtad 2025-05-08. DOI: 10.1036/1097-8542.757613. URL: <https://www.accessscience.com/content/article/a757613>.
- [13] D. Damyantov, Z. Varbanov och S. Varbanova, "An improved approach of using data storage services in ASP.NET Core," 2022, s. 287–291. DOI: 10.1109/ICAI55857.2022.9959991.
- [14] N. Fatimatuzzahra, Y. Priyadi och R. Nurtantyana, "Functional and Non-Functional Requirement (FR and NFR) Formation Based on Requirement Elicitation Using Text Semantics in IdVar4CL Artifact," i *2025 10th International Conference on Signal Processing and Communication (ICSC)*, 2025, s. 421–426. DOI: 10.1109/ICSC64553.2025.10968323.
- [15] K. Pappu, G. Reddy, S. Supraja och V. Prasad, *Analytical Hierarchy Process approach – An application of engineering education*, febr. 2013.
- [16] A. Hudaib, R. M. T. Masadeh, M. H. Qasem och A. I. Alzaqebah, "Requirements Prioritization Techniques Comparison," *Modern Applied Science*, årg. 12, nr 2, s. 62, 2018. DOI: 10.5539/mas.v12n2p62.
- [17] D. Saff, S. Artzi, J. H. Perkins och M. D. Ernst, "Automatic test factoring for java," i *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '05, Long Beach, CA, USA: Association for Computing Machinery, 2005, s. 114–123, ISBN: 1581139934. DOI: 10.1145/1101908.1101927. URL: <https://doi.org/10.1145/1101908.1101927>.
- [18] L.-O. Bligård, "Utvecklingsprocessen ur ett människa-maskinperspektiv," Avdelningen för Design & Human Factors, Institutionen för Produkt- och Produktionsutveckling, Chalmers Tekniska Högskola, Göteborg, Teknisk rapport 96, 2015, Med stöd av Eva Simonsen. ISSN 1652-9243. DOI: 10.13140/RG.2.1.1954.4400. URL: <https://www.acd3.se>.
- [19] C. Fox, "Agile methods in software engineering," *AccessScience*, 2023, Hämtad 2025-05-12. DOI: 10.1036/1097-8542.014730. URL: <https://www.accessscience.com/content/article/a014730>.
- [20] M. Rodriguez, M. Piattini och C. Ebert, "Software Verification and Validation Technologies and Tools," *IEEE Software*, årg. 36, nr 2, s. 13–24, 2019. DOI: 10.1109/MS.2018.2883354.
- [21] K. T. Ulrich, S. D. Eppinger och S. Bengtsson, *Produktutveckling : konstruktion och design*, 1. uppl. Lund: Studentlitteratur, 2014, ISBN: 9789144074214.
- [22] U. Flick, K. Roulston och M. Choi, "Qualitative Interviews," i *The SAGE Handbook of Qualitative Data Collection*, London: SAGE Publications Ltd, 2018, s. 233–249. DOI: 10.4135/9781526416070. URL: <https://methods.sagepub.com/hnbk/edvol/the-sage-handbook-of-qualitative-data-collection/toc>.
- [23] F. Bjerhem Aronsson, O. Cronvall, A. Enström, M. Johannesson, T. Karlsson och L. Åkerfeldt, "Liveinformation om lediga grupprum," Examensarbete på kandidatnivå, Chalmers tekniska högskola, 2024. URL: <http://hdl.handle.net/20.500.12380/308652>.
- [24] Chalmers tekniska högskola, *Regler för användning av AI-verktyg*, Hämtad 2025-02-08, 2024. URL: [37](https://www.chalmers.se/utbildning/dina-</p></div><div data-bbox=)

studier/kandidat-och-examensarbete/regler-for-anvandning-av-ai-verktyg/.

- [25] Chalmers tekniska högskola, *Boka lokaler, gemensamma ytor och grupprum*, <https://www.chalmers.se/utbildning/dina-studier/planera-och-genomfora-studier/boka-lokaler-och-grupprum/>, Hämtad 2025-05-13, Göteborg, 2024. URL: <https://www.chalmers.se/utbildning/dina-studier/planera-och-genomfora-studier/boka-lokaler-och-grupprum/>.

Bilaga A

Val av koncept

Tabell A.1: Elimineringssmatris för koncept från idégenerering.

Koncept	Frontend	Datahantering	Underhåll och lätt att förstå	Pålitlighet	Användarvänlighet	Integrering med befintliga system	Analysera historisk data	Se realtidsdata	Behövrigheter	Säkerhet	Vidare
A	API	Samlat i egen databas	-1	-1	1	1	1	1	1	1	
B	API	Hämtas vid behov	1	1	1	1	1	1	1	1	Ja
C	Helintegrerat nytt program	Samlat i egen databas	-1	-1	1	1	1	1	1	1	
D	Helintegrerat nytt program	Hämtas vid behov	1	1	1	1	1	1	1	1	Ja
E	Heltintegrerat med Grafana	Samlat i egen databas	-1	-1	1	1	1	1	1	1	
F	Heltintegrerat med Grafana	Hämtas vid behov	1	1	1	1	1	1	1	1	Ja
G	Heltintegrerat med QlikView	Samlat i egen databas	-1	-1	-1	-1	1	?	?	-1	
H	Heltintegrerat med QlikView	Hämtas vid behov	-1	1	-1	-1	1	?	?	-1	
I	Heltintegrerat med Power BI	Samlat i egen databas	-1	-1	1	1	1	1	1	-1	
J	Heltintegrerat med Power BI	Hämtas vid behov	1	1	1	1	1	1	1	-1	
K	Bygga vidare på Find A Room	Samlat i egen databas	-1	-1	1	1	1	1	1	1	
L	Bygga vidare på Find A Room	Hämtas vid behov	1	1	1	1	1	1	1	1	Ja
M	Helintegrerat i kårappen	Samlat i egen databas	-1	-1	1	1	-1	-1	1	1	
N	Helintegrerat i kårappen	Hämtas vid behov	1	1	1	1	-1	-1	1	1	
O	Helintegrerat i Campus Maps	Samlat i egen databas	-1	-1	1	1	-1	-1	1	1	
P	Helintegrerat i Campus Maps	Hämtas vid behov	-1	1	1	1	-1	-1	1	1	

Tabell A.2: Viktning av användarkraven.

Krav	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	Summa	Relativ summa
A1	-	1	1	1	1	1	1	0	0	0	0	0	6	0,091
A2	0	-	1	0,5	1	1	1	0	0	0	0	0	4,5	0,068
A3	0	0	-	0	0	1	1	0	0	0	0	0	2	0,030
A4	0	0,5	1	-	1	0,5	1	0	0	0	0	0	4	0,061
A5	0	0	1	0	-	1	1	0	0	0	0	0	3	0,045
A6	0	0	0	0,5	0	-	1	0	0	0	0	0	1,5	0,023
A7	0	0	0	0	0	0	-	0	0	0	0	0	0	0,000
A8	1	1	1	1	1	1	1	-	0	0	1	1	9	0,136
A9	1	1	1	1	1	1	1	1	-	0	0,5	1	9,5	0,144
A10	1	1	1	1	1	1	1	1	1	-	0	1	10	0,152
A11	1	1	1	1	1	1	1	0	0,5	1	-	1	9,5	0,144
A12	1	1	1	1	1	1	1	0	0	0	0	-	7	0,106
													66	1,00

Tabell A.3: Val av bästa koncept genom en Kesselringmatris.

Krav	Vikt	Ideal		Koncept B		Koncept D		Koncept F		Koncept L	
		v	t	v	t	v	t	v	t	v	t
A1	0,09	5	0,45	5	0,45	1	0,09	1	0,09	1	0,09
A2	0,07	5	0,34	5	0,34	3	0,20	4	0,27	3	0,20
A3	0,03	5	0,15	5	0,15	4	0,12	4	0,12	4	0,12
A4	0,06	5	0,30	5	0,30	5	0,30	4	0,24	5	0,30
A5	0,05	5	0,23	3	0,14	4	0,18	4	0,18	4	0,18
A6	0,02	5	0,11	4	0,09	1	0,02	1	0,02	2	0,05
A7	0,00	5	0,00	5	0,00	5	0,00	4	0,00	3	0,00
A8	0,14	5	0,68	5	0,68	4	0,55	4	0,55	4	0,55
A9	0,14	5	0,72	4	0,58	4	0,58	2	0,29	4	0,58
A10	0,15	5	0,76	4	0,61	4	0,61	4	0,61	4	0,61
A11	0,14	5	0,72	4	0,58	4	0,58	4	0,58	4	0,58
A12	0,11	5	0,53	2	0,21	4	0,42	4	0,42	4	0,42
T		60	5,00	51	4,13	43	3,65	40	3,37	42	3,67
T/T_ideal		1,00	1,00	0,85	0,83	0,72	0,73	0,67	0,67	0,70	0,73
Medel		5,00	0,42	4,25	0,34	3,58	0,30	3,33	0,28	3,50	0,31
Std-avvikelse		0,00	0,23	0,75	0,20	0,96	0,20	1,00	0,17	0,83	0,20
Median		5,00	0,40	4,50	0,32	4,00	0,25	4,00	0,26	4,00	0,25
Antal svaga punkter		0		0		2		2		1	

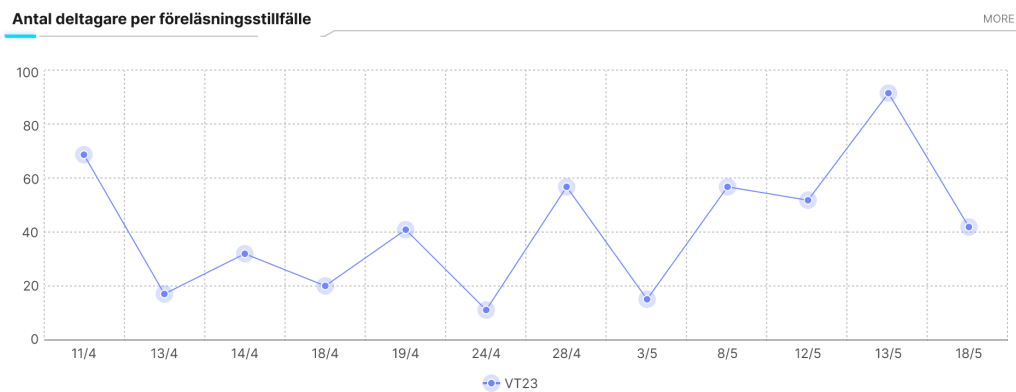
Bilaga B

Figmaskisser



Figur B.1: Skissen visar ett möjligt sätt att realisera funktionen där användaren kan se historiskt deltagande för en kurs, vilket är en del av krav F2: *Se historiskt deltagande kopplat till kurser och kurstillfällen.*

Vald kurs: **TDA357** Kurstillfälle: VT24, lp3



Föreläsningar

Antal deltagare:

Högst: 80 personer (90%)

Genomsnitt: 60 personer (83%)

Laborationer

Antal deltagare:

Högst: 50 personer (57%)

Genomsnitt: 30 personer (34%)

63% mindre deltagande i laborationer jämfört med föreläsningar.

50% mindre deltagande efter halva kursen.

Visa detaljerad data



Figur B.2: Skissen visar ett möjligt sätt att realisera att användaren ska kunna se historiskt deltagande av ett kurstillfälle, vilket är en del av kravet F2: *Se historiskt deltagande kopplat till kurser och kurstillfällen.*

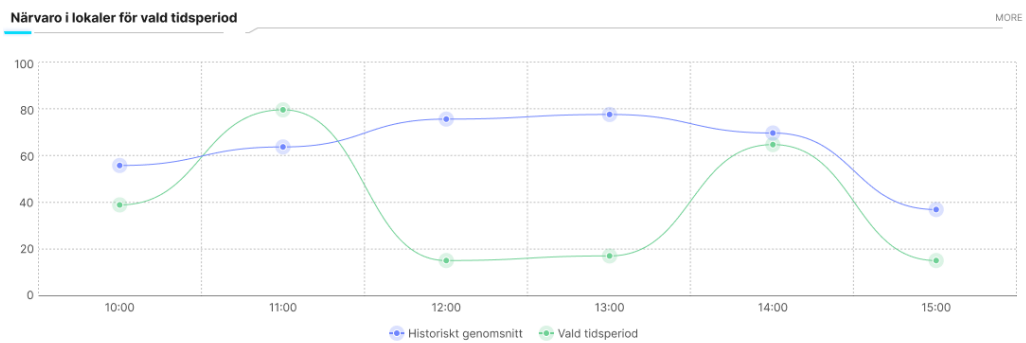
Visa detaljerad data ^

Föreläsning/laboration	Datum	Tid	Antal deltagare
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64
Föreläsning	11/4-2024	10:00-12:00	64

Figur B.3: Tabellen i figuren är en del av skissen i figur B.2.

Närvaro i lokaler

Filter:



Antalet besökare

Totalt: 1300 personer

Genomsnitt per timme: 45 personer

15% mindre än normalt vid vald tidpunkt.

Figur B.4: Skissen visar ett möjligt sätt att realisera krav F3: *Se användning av en lokal över tid.*



Figur B.5: Skisserna visar ett möjligt sätt att realisera en del av krav F5: *Se studentflöden på campus, byggnader och lokaler*, vilket delvis innebär att se heatmaps över lokalnyttjandet under en vald tidsperiod.

Bilaga C

Slutgiltig kravspecifikation

Tabell C.1: Sammanfattning av den slutgiltiga kravspecifikationen.

Funktionella krav
F1 Integrera data i andra system och verktyg
F2 Se historiskt deltagande kopplat till kurser och kurstillfällen
F3 Se användning av en lokal över tid
F4 Mäta lokalernas beläggningseffektivitet
F5 Se studentflöden på campus, byggnader och lokaler
F6 Se beläggning i lokaler, grupprum och datorsalar i realtid
F7 Prediktion av lediga grupprum baserat på historisk data
F8 Möjlighet att stänga ned systemet omedelbart
F9 Behörighetssystem
F10 Information om hur data används
Icke-funktionella krav
I1 Säkerhet: data är skyddad
I2 Underhåll: lätt att underhålla, anpassa
I3 Prestanda: Snabb hämtning av data från API:n
I4 Pålitlig: Information stämmer, systemet är konsekvent
I5 Skalbarhet: lätt att bygga på
I6 Användarvänlighet: lätt att använda
I7 Anonymiserad data

1 Funktionella krav

- F1. Integrera data i andra system och verktyg**
Must-have
Från A1
- Systemet ska tillgängliggöra all data via ett API, för att kunna integreras i universitetets befintliga program och verktyg som redan används av användarna. Det inkluderar närvaroinformation, inklusive statistik och relevanta nyckeltal. Syftet är att göra informationen lättillgänglig där användaren redan arbetar, och därmed undvika behovet av ytterligare en applikation. Det möjliggör också att kombinera närvarostatistik med annan data som hanteras i dessa verktyg.
- F2. Se historiskt deltagande kopplat till kurser och kurstillfällen**
Must-have
Från A2
- Användaren ska kunna se historisk närvarodata för specifika kurstillfällen samt få en överblick över närvarostatistik för hela kurser. Det ska även vara möjligt att analysera förhållandet mellan antalet antagna studenter och det faktiska deltagandet, liksom att jämföra deltagande mellan olika kurstillfällen och kurser. Detta krav är centralt för att ge programansvariga och lärare möjlighet att använda närvarodata som underlag vid kursutvärderingar. För lärarna innebär det även stöd i planering av föreläsningar, handledningspass och laborationer baserat på faktisk närvaro, snarare än registrerade antal eller erfarenhetsbaserade antaganden. Studierektorer och schemaläggare kan dessutom använda data för att optimera resursfördelning, särskilt för TA:s, samt för att fatta beslut kring lokalbokning och eventuell kontrollerad överbokning. Följande nyckeltal ska inkluderas:
Andel närvarande studenter:
- $$\left(\frac{\text{Antal närvarande studenter}}{\text{Antal registrerade studenter på kursen}} \right) \times 100 \quad (\text{C.1})$$
- där *antalet närvarande studenter* är det högsta registrerade värdet för närvaro under ett undervisningstillfälle.
Närvarograden för kursen som helhet:
- $$\frac{\sum \text{andel närvarande studenter}}{\text{Antalet undervisningstillfällen för kursen}} \quad (\text{C.2})$$
- F3. Se användning av en lokal över tid**
Could-have
Från A3
- Systemet ska tillhandahålla möjlighet att analysera närvarodata baserat på lokal, vilket kan ge lokalansvariga insikter i hur ofta och hur intensivt olika lokaler används. Denna information ska kunna brytas ned på lokal-, byggnads- och campusnivå samt analyseras över olika tidsintervall.

- F4.** **Mäta lokalernas beläggningseffektivitet**
 Should-have
 Från A4
- Systemet ska kunna mäta hur stor andel av platserna som faktiskt nyttjas i en lokal under en bokning. Denna information ska kunna brytas ned på lokal-, byggnads- och campusnivå samt analyseras över olika tidsintervall. Syftet är att ge lokalansvariga, schemaläggare och fastighetsförvaltning möjlighet att fatta datadrivna beslut kring lokaloptimering, resursfördelning och strategiska investeringar. Funktionen är även viktig för att följa upp om införandet av ett närvarosystem lett till förbättrat lokalnyttjande. Nyckeltalet beläggningsgrad vara inkluderat:
- $$\frac{\text{Antalet närvarande}}{\text{Antalet platser i lokalen}} \quad (\text{C.3})$$
- F5.** **Se studentflöden i byggnader och lokaler samt på campus.**
 Could-have
 Från A5
- Användaren ska kunna se sammanställd statistik över studentflöden i byggnader och lokaler. Detta möjliggör mer behovsanpassad lokalvård samt utgör ett bättre underlag för diskussioner kring lokaloptimering och driftplanering. Statistiken ska vara översiktlig och förslagsvis inkludera en heatmap, som visar var flest folk har befunnit sig på campus.
- F6.** **Se beläggning i lokaler, grupprum och datorsalar i realtid**
 Could-have
 Från A6
- Att kunna visa realtidsdata över beläggning i olika typer av utrymmen – såsom föreläsningssalar, handledningsrum, grupprum och datorsalar – bedöms ha stort potentiellt värde. Denna funktion skiljer sig från Find A Room genom att realtidsdata kan kopplas till schemat, vilket gör det möjligt för användaren att söka efter närvarodata i specifika lokaler baserat på kurs. För grupprum finns dessutom möjlighet till automatisk avbokning om rummet inte tas i anspråk inom en viss tid. För studenter innebär detta att de enklare kan identifiera var det finns lediga platser, medans för lärare och TA:s möjliggör funktionen prioritering av insatser, vid till exempel handledning, baserat på var flest studenter befinner sig.
- F7.** **Prediktion av lediga grupprum baserat på historisk data**
 Won't-have
 Från A7
- Systemet ska kunna analysera historisk användningsdata för grupprum och ge sannolikhetsbaserade uppskattningar om när ett rum sannolikt kommer bli ledigt i en byggnad. Informationen ska kunna hjälpa studenter att snabbare hitta tillgängliga studieplatser, när samtliga grupprum är bokade.

- F8. Möjlighet att stänga ned systemet omedelbart**
Must-have
Från A8
Systemet ska innehålla en funktion för omedelbar avstängning i händelse av kritiska fel eller allvarliga säkerhetsincidenter. Funktionen ska vara avgränsad, men lättillgänglig för systemadministratörer, säkerhetsansvariga eller annan ansvarig.
- F9. Behörighetssystem**
Must-have
Från A9
Systemet ska innefatta ett flexibelt och rollbaserat behörighetssystem för att säkerställa att endast auktoriserade användare får tillgång till specifika delar av systemet som de har behov av. Behörighetssystemet ska vara kopplat till ett centralt rollsystem för högskolan, för att säkerställa att åtkomstnivåer alltid speglar aktuella behörigheter enligt organisationens strukturer. Administratörer ska kunna hantera alla behörighetsnivåer och snabbt kunna justera åtkomsträttigheter vid behov.
- F10. Information om hur data används**
Must-have
Från A12
Det ska framgå tydligt vilken data som samlas in, hur den används och i vilket syfte. Syftet är att upprätthålla förtroende och motverka känslan av övervakning bland studenter och personal.

2 Icke-funktionella krav

Samtliga icke-funktionella krav är kategoriserade som *must*, eftersom de ses som grundläggande för systemet. Målvärden för kraven är inte specificerade.

- I1. Säkerhet: data är skyddad**
Must-have
Från A10
Det ska framgå tydligt vilken data som samlas in, hur och varför den används. Syftet är att upprätthålla förtroende och motverka känslan av övervakning bland studenter och personal.
- I2. Underhåll: lätt att underhålla, anpassa**
Must-have
Nytt krav
Systemet ska följa en modulär arkitektur för att underlätta vidareutveckling och flexibilitet vid förändringar, såsom att byta ut eller lägga till nya datakällor.
- I3. Prestanda: Snabb hämtning av data från API:n**
Must-have
Nytt krav
Systemet ska kunna leverera efterfrågad statistik nära realtid via ett API. Det ska även kunna hantera parallella förfrågningar från samtliga Chalmers studenter och anställda utan märkbar prestandaförlust.
- I4. Pålitlig: Information stämmer, systemet är konsekvent**
Must-have
Nytt krav
Systemet ska kunna leverera korrekt och fullständig närvarodata utan avbrott. Felhantering ska vara robust och tydlig, samt felhändelser ska loggas.

- I5.**
Must-have
Nytt krav
- Skalbarhet: Lätt att bygga ut och integrera**
Systemet ska vara byggt för att enkelt kunna hämta nya typer av data, hantera nya dataformat och kombinera flera datakällor. Det ska vara lätt att bygga ut med nya nyckeltal och analyser. Systemet ska även kunna integrera nya analysverktyg och metoder utan behov av omstrukturering.
- I6.**
Must-have
Nytt krav
- Användarvänlighet: Lätt att använda och integrera**
API:et ska vara enkelt att använda, med tydlig dokumentation som visar vilken typ av data som skickas för olika förfrågningar. Statistiken och nyckeltalen ska vara lättbegripliga, relevanta och anpassade efter användarnas behov och förväntningar.
- I7.**
Must-have
Från A11
- Anonymiserad data**
Data kopplat till närvaro ska vara anonym med hänsyn till personlig integritet.

Bilaga D

API-endpoints med exempelsvar

1 Närvaroinformation

GET /api/attendance/occurrence/{occurrenceId}

Hämtar sammanställd närvarostatistik för ett specifikt kurstillfälle.

Parametrar

occurrenceId	ID för kurstillfället
"ppu156_lp2_ht24"	

Svar

```
{
  "occurrenceId": "ppu156_lp2_ht24",
  "startTime": "2024-11-04T08:00:00",
  "endTime": "2024-12-09T17:00:00",
  "maxAttendance": 186,
  "meanAttendance": 115.8
}
```

GET /api/attendance/occurrence/{occurrenceId}/lecture

Hämtar närvarodata per föreläsning inom ett specifikt kurstillfälle.

Parametrar

occurrenceId	ID för kurstillfället
"ppu156_lp2_ht24"	

Svar

```
[
  {
    "lectureId": "0018011",
    "startTime": "2024-11-04T08:00:00",
    "endTime": "2024-11-04T09:45:00",
    "max_attended_students": 164,
  }
]
```

```
    "mean_attended_students": 109.3,  
    "lectureOrderNumber": 1  
  },  
  ...  
]
```

GET /api/Attendance/course/{courseId}/occurrence/lecture

Hämtar kurstillfällena med föreläsningar för en viss kurs.

Parametrar

courseId	ID för kursen
"ppu156"	

Svar

```
[  
  {  
    "occurrenceId": "ppu156_lp2_ht24",  
    "startTime": "2024-11-04T08:00:00",  
    "endTime": "2024-12-09T17:00:00",  
    "maxAttendance": 186,  
    "meanAttendance": 115,  
    "lectures": [  
      {  
        "lectureId": "0018011",  
        "startTime": "2024-11-04T08:00:00",  
        "endTime": "2024-11-04T09:45:00",  
        "max_attended_students": 164,  
        "mean_attended_students": 109.3,  
        "lectureOrderNumber": 1  
      },  
      ...  
    ]  
  },  
  ...  
]
```

2 Kurser

GET /api/courses

Hämtar alla kurser.

Svar

```
[  
  {  
    "courseId": "ppu156",  
    "name": "Datorstodd maskinkonstruktion (CAD)"  
  }  
]
```

```
    },  
    ...  
  ]
```

GET /api/courses/{courseId}

Hämtar information om en specifik kurs baserat på dess ID.

Parametrar

courseId	ID för kursen
"ppu156"	

Svar

```
{  
  "courseId": "ppu156",  
  "name": "Datorstodd maskinkonstruktion (CAD)"  
}
```

GET /api/Courses/{courseId}/occurrences

Hämtar alla kurstillfällen för en specifik kurs baserat på dess ID.

Parametrar

courseId	ID för kursen
"ppu156"	

Svar

```
[  
  {  
    "occurrenceId": "ppu156_lp2_ht24",  
    "courseId": "PPU156",  
    "year": 2024,  
    "semester": "ht",  
    "studyPeriod": 2,  
    "registeredStudents": 75  
  },  
  ...  
]
```

3 Undervisningstillfällen

GET /api/Lectures

Hämtar alla föreläsningar.

Parametrar

<code>startTime</code> "2024-11-01T00:00:00"	(Valfri) Starttid för att filtrera föreläsningar.
<code>endTime</code> "2024-11-30T23:59:59"	(Valfri) Sluttid för att filtrera föreläsningar.

Svar

```
[
  {
    "lectureId": "0018011",
    "occurrenceId": "ppu156_lp2_ht24",
    "courseId": "ppu156",
    "year": 2024,
    "semester": "ht24",
    "studyPeriod": 2,
    "lectureStart": "2024-11-04T08:00:00",
    "lectureEnd": "2024-11-04T09:45:00",
    "lectureType": "Introduktionsforelasning",
    "roomId": "hb3",
    "lectureOrderNumber": null
  },
  ...
]
```

GET /api/lectures/{lectureId}

Hämtar information om en specifik föreläsning baserat på dess ID.

Parametrar

<code>lectureId</code> "0018011"	ID för föreläsningen
-------------------------------------	----------------------

Svar

```
{
  "lectureId": "0018011",
  "occurrenceId": "ppu156_lp2_ht24",
  "courseId": "ppu156",
  "year": 2024,
  "semester": "ht24",
  "studyPeriod": 2,
  "lectureStart": "2024-11-04T08:00:00",
  "lectureEnd": "2024-11-04T09:45:00",
  "lectureType": "Introduktionsforelasning",
  "roomId": "hb3",
  "lectureOrderNumber": null
}
```

4 Kurstillfällen

GET /api/Occurrences

Hämtar alla kurstillfällen.

Svar

```
[
  {
    "occurrenceId": "ppu156_lp2_ht24",
    "courseId": "PPU156",
    "year": 2024,
    "semester": "ht",
    "studyPeriod": 2,
    "registeredStudents": 75
  },
  ...
]
```

GET /api/occurrences/{occurrenceId}

Hämtar information om ett specifikt kurstillfälle baserat på dess ID.

Parametrar

occurrenceId	ID för kurstillfället
"ppu156_lp2_ht24"	

Svar

```
{
  "occurrenceId": "ppu156_lp2_ht24",
  "courseId": "PPU156",
  "year": 2024,
  "semester": "ht",
  "studyPeriod": 2,
  "registeredStudents": 75
}
```

GET /api/occurrences/{occurrenceId}/lectures

Hämtar alla föreläsningar för ett specifikt kurstillfälle baserat på dess ID.

Parametrar

occurrenceId	ID för kurstillfället
"ppu156_lp2_ht24"	

Svar

```
[
  {
    "lectureId": "0018011",
    "occurrenceId": "ppu156_lp2_ht24",
  }
]
```

```
    "courseId": "ppu156",
    "year": 2024,
    "semester": "ht24",
    "studyPeriod": 2,
    "lectureStart": "2024-11-04T08:00:00",
    "lectureEnd": "2024-11-04T09:45:00",
    "lectureType": "Introduktionsforelasning",
    "roomId": "hb3",
    "lectureOrderNumber": null
  },
  ...
]
```

5 Lokaler

GET /api/rooms

Hämtar alla lokaler.

Svar

```
[
  {
    "roomId": "ha1",
    "roomName": "HA1",
    "type": "auditorium",
    "capacity": 263,
    "buildingId": "horsalslangan"
  },
  ...
]
```

GET /api/rooms/{roomId}

Hämtar information om en specifik lokal baserat på dess ID.

Parametrar

<code>roomId</code>	ID för lokalen
<code>"ha1"</code>	

Svar

```
{
  "roomId": "ha1",
  "roomName": "HA1",
  "type": "auditorium",
  "capacity": 263,
  "buildingId": "horsalslangan"
}
```