**AttendIt**

**Grupp 20**

**Designdokument**

**V. 1.1.0**

**17/03/22**

**Handledare** **Gruppmedlemmar**  
*Farid Naisan Axel Bengtsson  
 Malin Möller  
 Oskar Georgsson  
 Linus Håkansson  
 Pontus Knutsson*

# Dokumenthistorik

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Datum | Version | Beskrivning | Författare | Granskare | Status |
| 17/03/29 | 1.0 | Användningsfallsdiagram & Påbörjat användningsfallsbeskrivningar | Malin Möller  Linus Håkansson | Oskar Georgsson, Axel Bengtsson,Malin Möller | Klart |
| 17/03/30 | 1.0 | Användningsfallsbeskrivningar klart | Malin Möller  Linus Håkansson | Oskar Georgsson, Axel Bengtsson,Malin Möller | Klart |
| 17/03/30 | 1.0 | Användargränssnitt | Axel Bengtsson | Oskar Georgsson, Axel Bengtsson,Malin Möller | Klart |
| 17/03/11 | 1.1 | ER-diagram med kort beskrivning | Linus Håkansson  Malin Möller | Oskar Georgsson, Axel Bengtsson,Malin Möller | Klart |
| 17/04/12 | 1.1 | Uppdaterat användningsfallsdiagram | Malin Möller |  |  |
| 17/04/12 | 1.1 | Lagt till sekvensdiagram, Systemdiagram, | Oskar Georgsson |  |  |
| 17/04/12 | 1.1 | Lagt till klassdiagaram för server | Linus Håkansson |  |  |
| 17/04/13 | 1.1 | Uppdaterat ER-diagram | Linus Håkansson |  |  |

**Innehåll**

[**Dokumenthistorik**](#_gjdgxs) **2**

[**Designdokument**](#_1fob9te) **4**

[**Syfte**](#_3znysh7) **4**

[**Ordlista**](#_2et92p0) **5**

[**Systemdiagram**](#_buyj0rer7tc9) **6**

[**Användningsfallsdiagram**](#_1t3h5sf) **7**

[**Användargränssnitt**](#_2s8eyo1) **9**

[**Entity Relationship diagram**](#_17dp8vu) **12**

[**Klassdiagram**](#_rfdggcjy41kz) **13**

[**Sekvensdiagram**](#_6yy29saadrwx) **14**

# Designdokument

# Syfte

Syftet med detta dokument är att man ska få en överblick över hur systemet kommer att vara konstruerat gällande arkitektur. Detta dokument skall även underlätta för utvecklarna så att man har någon form utav mall att följa. Dokumentet kommer att innehålla systemdiagram, användningsfallsdiagram samt skisser över användargränssnitt.

# 

# 

# Ordlista

|  |  |
| --- | --- |
| ER-diagram | ER står för Entity Relationship är det är ett koncept hur man skriver en databas. |
| PK | PK står för Primary Key. Nyckel för att komma åt en rad i en databas. |
| Systemdiagram | Visar de komponenter som finns i projektet. |
| Användningsfallsdiagram | Visar de interaktioner som en användare kan göra med systemet. |
| Klassdiagram | Visar hur klasser interagerar. |
| Sekvensdiagram | Visar hur ett användningsfall genomförs. |

# 

# 

# 

# Systemdiagram

Systemdiagrammet nedan visar alla delar som ingår och är aktiva i projektet.

Server

Database

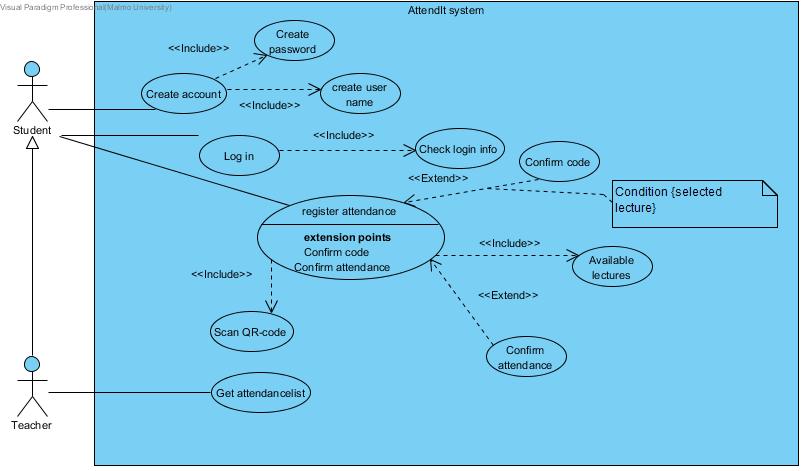
Client

Client

Client

“Ett systemdiagram finns som beskriver den övergripande arkitekturen av hur den är uppbyggd. Förstår att programmet använder sig av en databas, server och tre klienter samt hur dem hör ihop med varandra men inte mer än det. Ni hade kanske kunnat förklara de olika delarna, deras syfte/uppgift eller vad systemet innehåller mer ingående.”

# Användningsfallsdiagram



## Scenarion

I användningsfallsdiagrammet visas hur aktörerna “Student” och “Teacher” interagerar med systemet. Interaktionen beskrivs med olika scenarion för varje användningsfall där först det huvudsakliga flödet beskrivs, och därefter vilka alternativa flöden som kan uppstå.

### Create account

*Förberedelser: Applikationen ska vara installerad på enheten.*

*Förväntat resultat: Ett konto har skapats*

Huvudsakligt flöde:

1. Användare väljer att skapa nytt konto
2. Användaren fyller i användarnamn
3. Användaren fyller i lösenord
4. Uppgifter valideras
5. Konto skapas.

Alternativt flöde:

2a. Användarnamn ej giltigt

2a1. Användarnamn upptaget

2a2. Användarnamn innehåller för få bokstäver

2a3. Användarnamn innehåller för få siffror

2a4. Användarnamn innehåller specialtecken

3a. Lösenord ej giltigt

3a1. Lösenordet innehåller mindre än 8 tecken

5a. Konto ej kan skapas

5a1. Felmeddelande konto kan ej skapas

### Log in

*Villkor: Ett konto ska vara skapat i applikationen*

*Förväntat resultat : Användare blir inloggad*

Huvudsakligt flöde:

1. Användaren anger användarnamn
2. Användaren anger lösenord
3. Uppgifterna kontrolleras
4. Användaren är inloggad

Alternativa flöden:

3a. Uppgifter kontrolleras ej

3a1. Visa felmeddelande

3b. Uppgifter felaktiga

3b1. Visa felmeddelande

4a. Loggas inte in

4a1. Visa felmeddelande ingen internetanslutning

### Attendance

*Villkor: Användaren inloggad*

*Förväntat resultat : Användare har registrerats som närvarande vid vald aktivitet*

Huvudsakligt flöde:

1. Användaren scannar en QR-kod
2. Användaren väljer en föreläsning ur en lista
3. Användaren bekräftar sin närvaro
4. Närvaron registreras

Alternativa flöden:

1a. Scanning av QR-kod misslyckas  
 1a1. Användaren gör ett nytt försök  
 2a. Föreläsningen är obligatorisk  
 2a1. Användaren anger obligatorisk kod

### Attendancelist

*Villkor: Användare ska vara inloggad med admin-rättigheter  
Förväntat resultat: Användaren får fram en lista över studenter som närvarat vid viss aktivitet*

Huvudsakligt flöde:

1. Användare väljer alternativ statistik
2. Användare väljer föreläsning
3. Användare får fram närvarolista

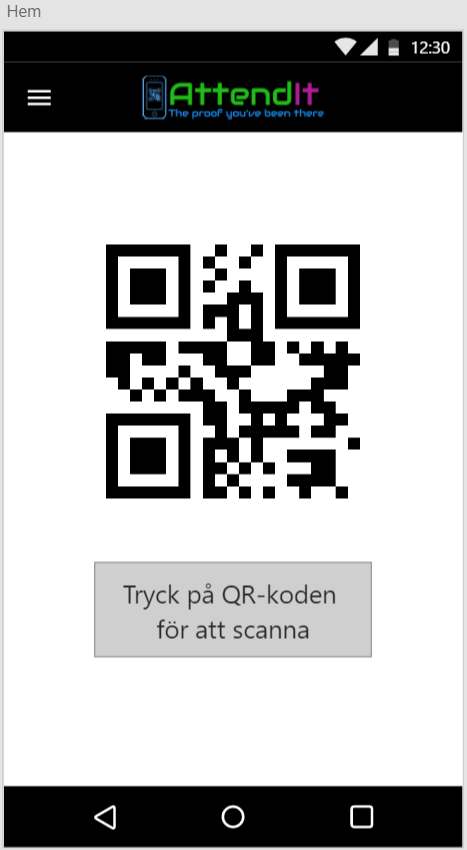
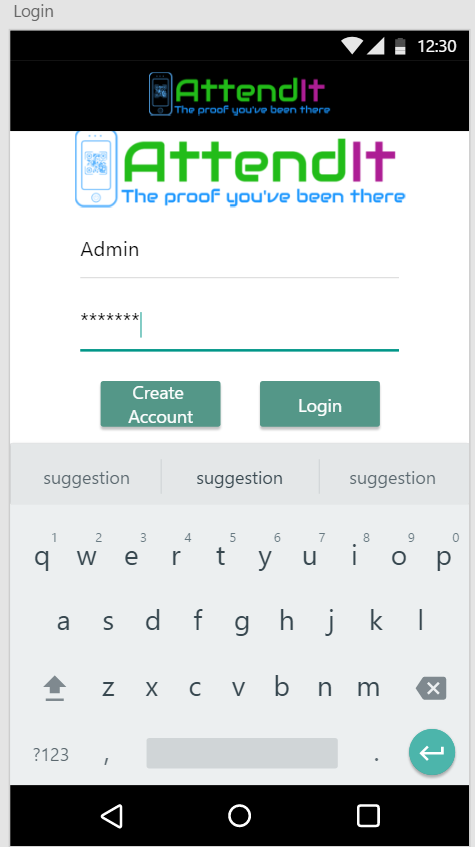
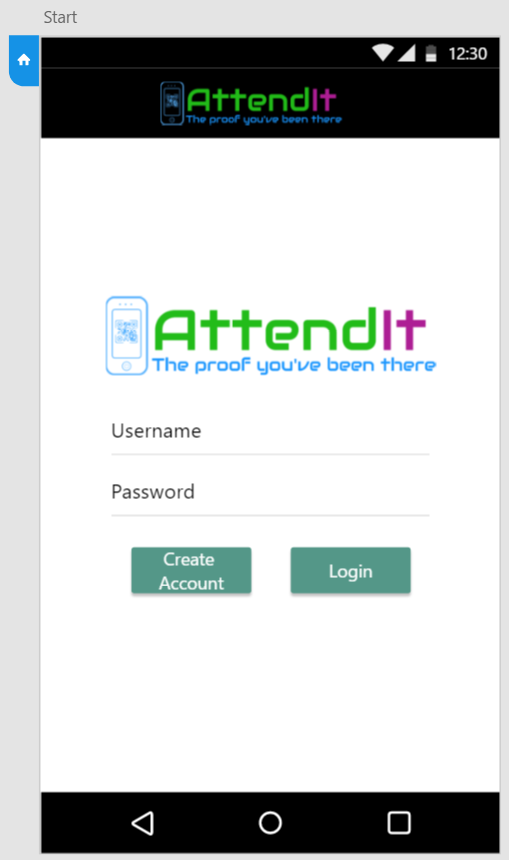
“Användningsfallsdiagrammet beskriver funktionaliteten i produkten. Lite överflödigt med info i usecase-diagrammet (diskuteras), Diagrammet ska visa vilka funktioner användaren kan göra, så kanske ta bort saker som check login info och available lectures. Istället för det kan ni skriva anmäla sig till lektioner. som systemet står för. Annars verkar det bra, tydligt och detaljerat och det finns beskrivningar för alla fallen i diagrammet. Respekt. Användarfallen förståeliga men inte jättetydliga, kan enkelt misstolkas för någon som inte är teknisk insatt exempelvis kan man förklara extension points och förtydliga vissa delar i diagrammet. Ser bra ut annars

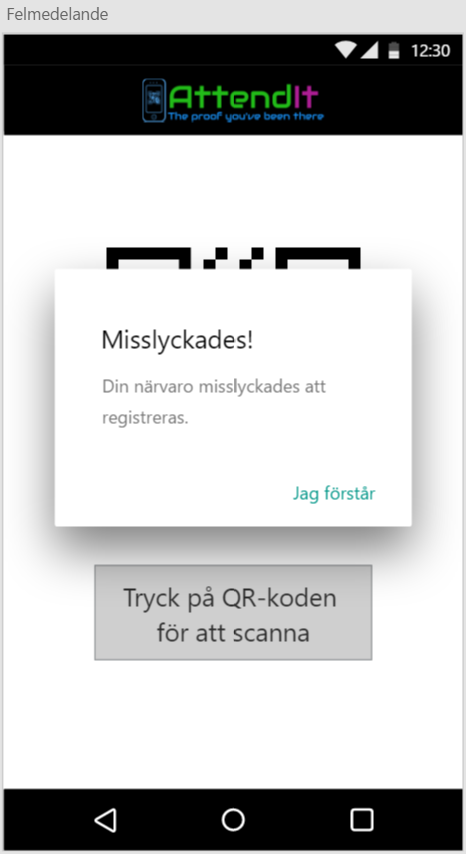
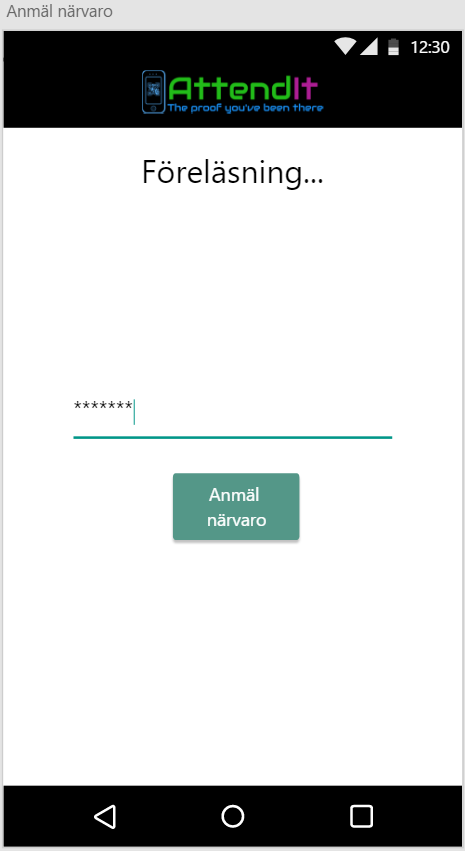
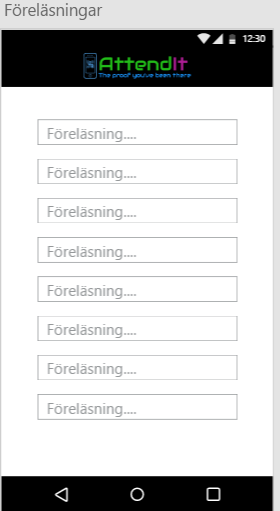
### 

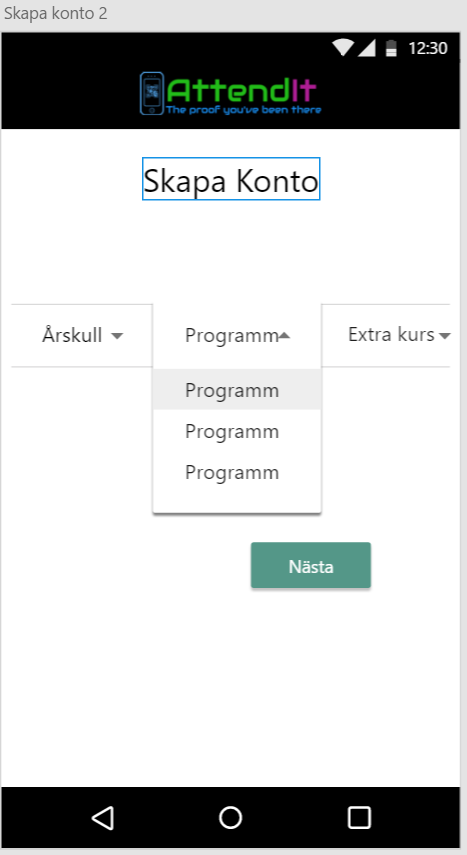
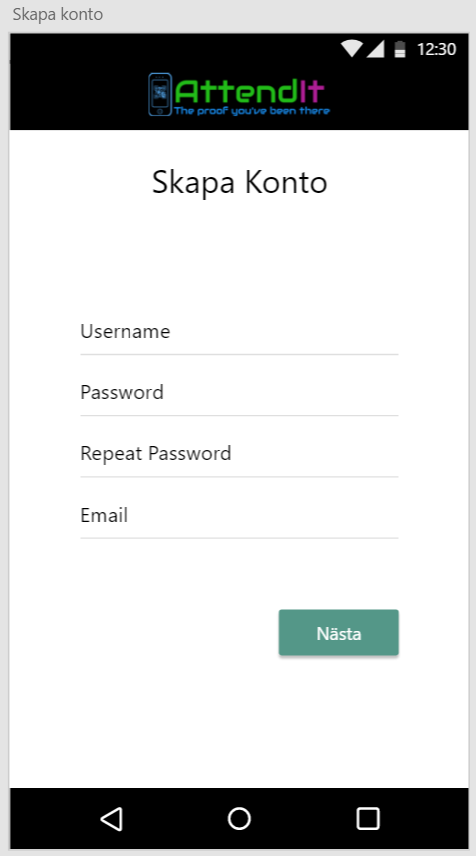
# Användargränssnitt

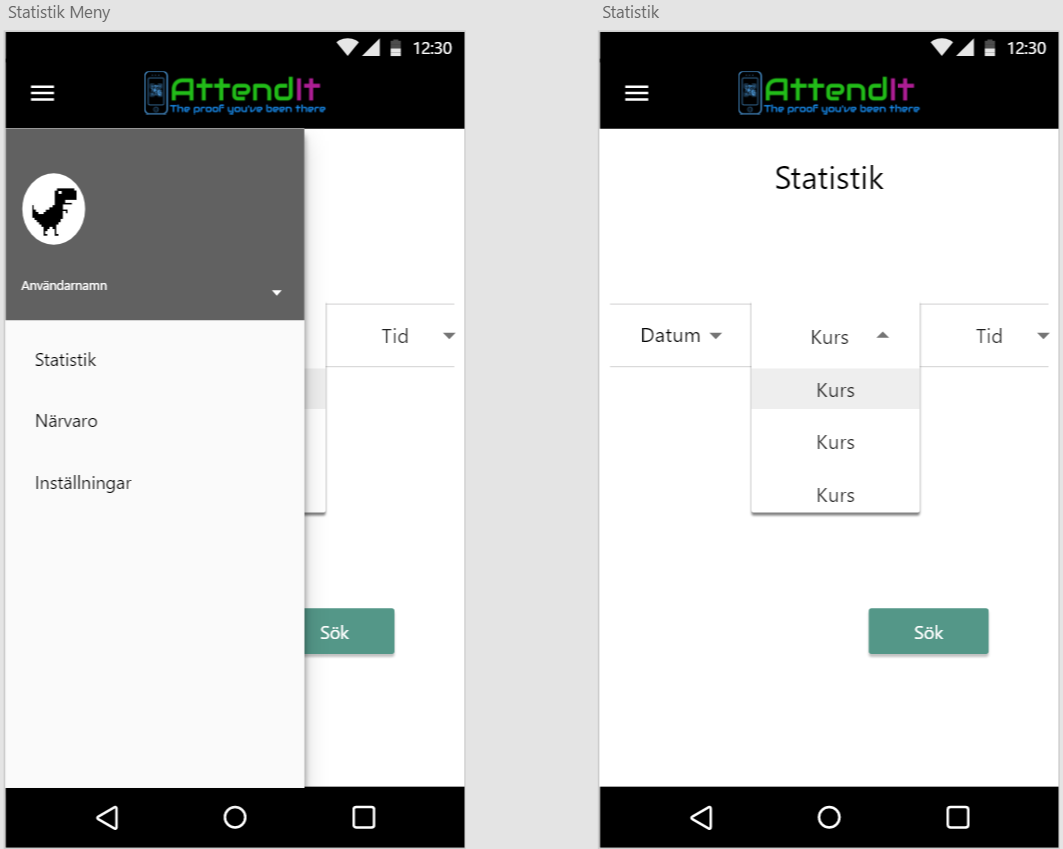
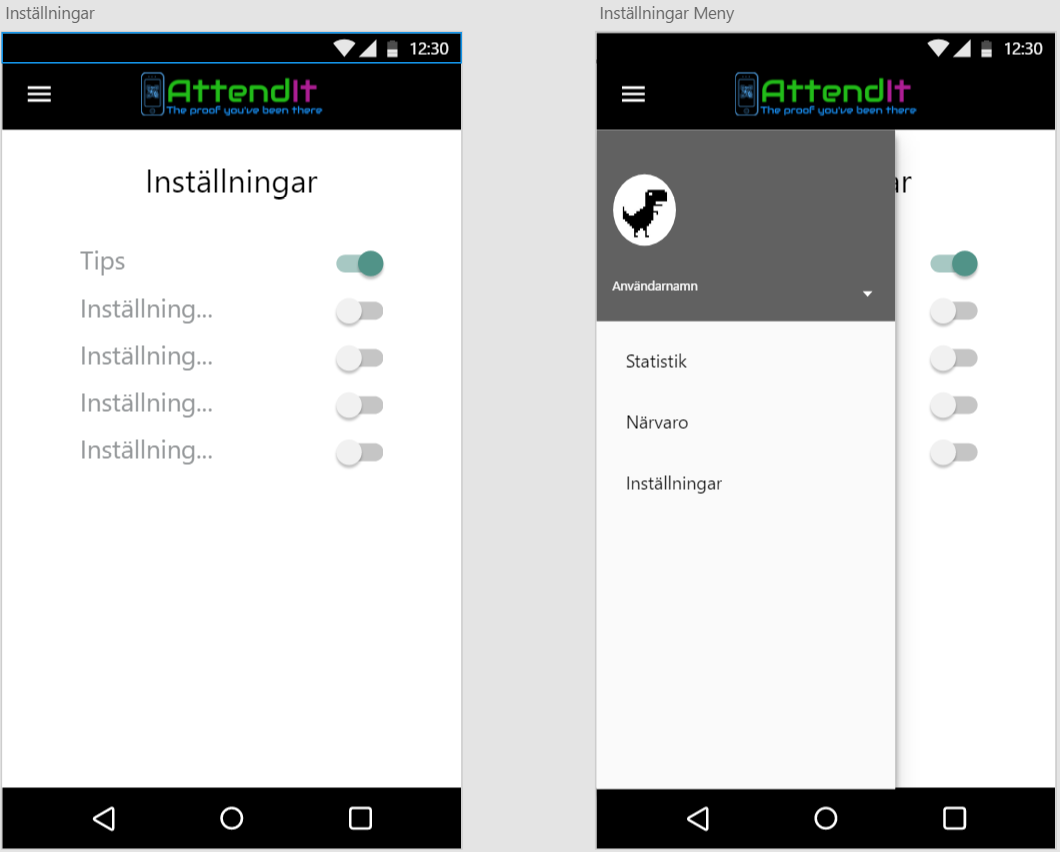
Utformandet av användargränssnittet har utförts med material design som senaste Android (7.1) är baserad på då detta är det nya sättet att designa applikationer på. Logiska vägar att välja och gå i appen är även något som har försökts att uppnå för att göra det enkelt för användaren att se vart hen befinner sig samt hur hen tar sig till ett visst ställe. Detta är inte den slutgiltiga designen men det är ett koncept som är värt att följa vid uttagandet av den slutliga produkten.

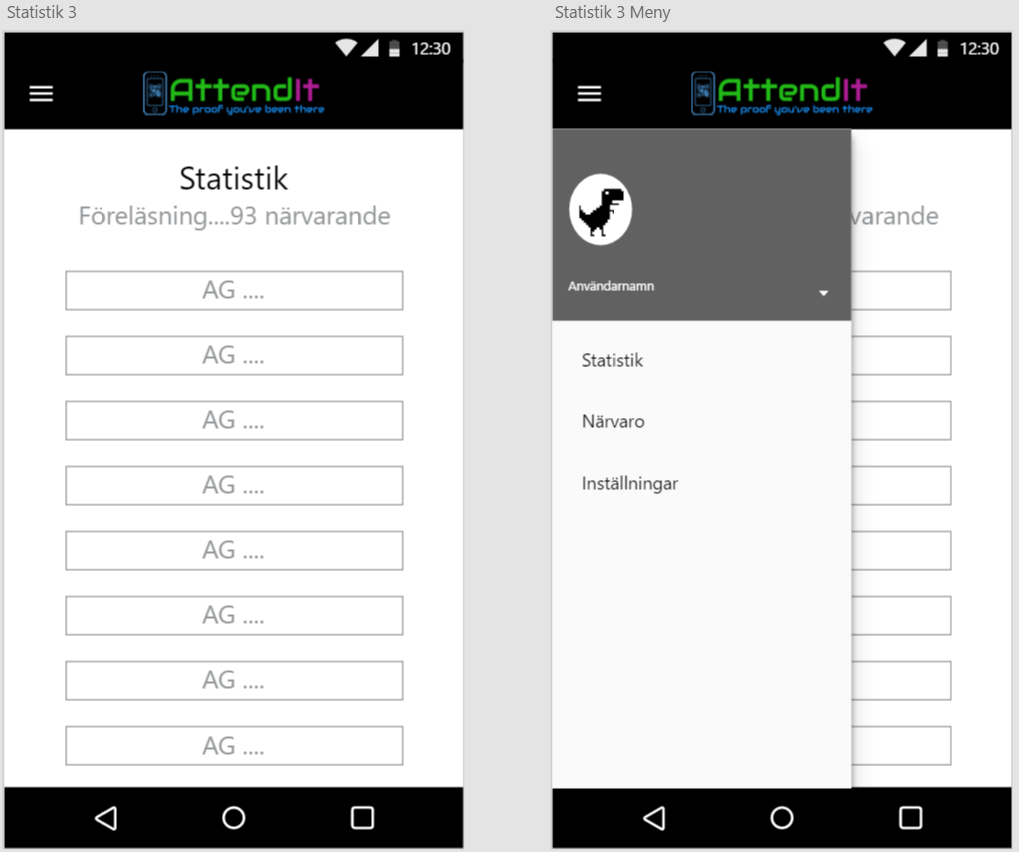
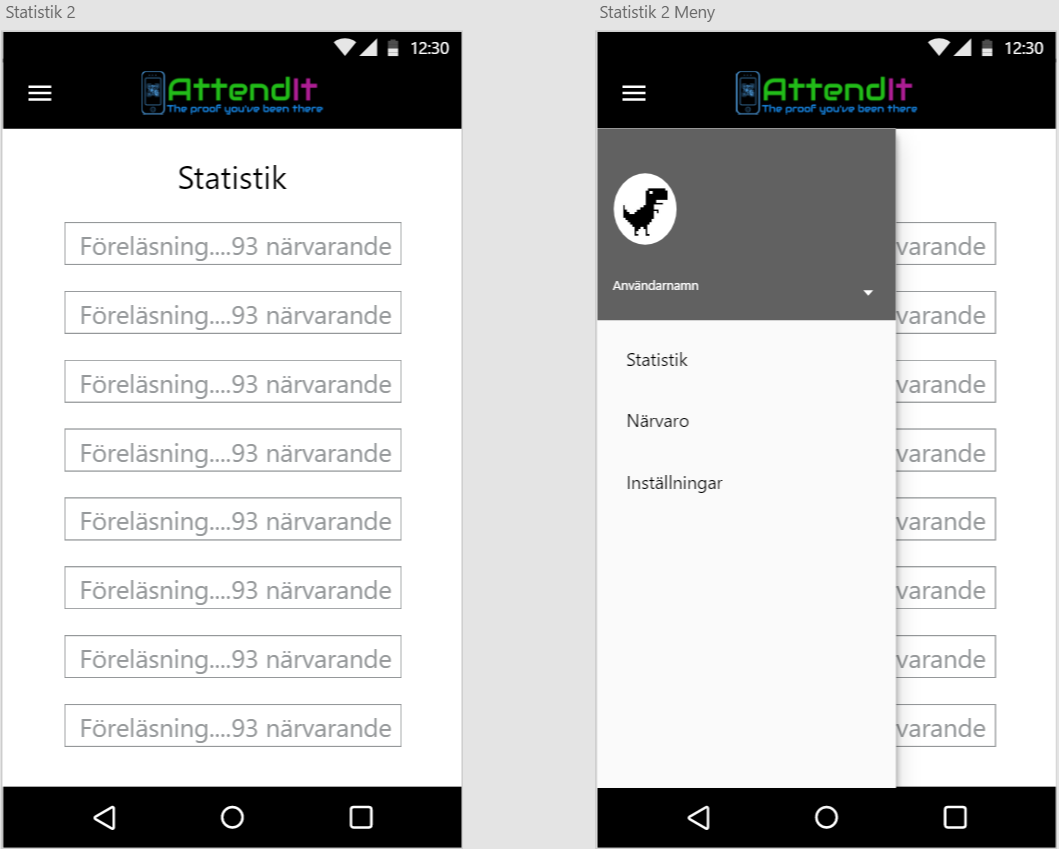
Nedan är ett par bilder som representerar de olika skärmarna i applikationen. Sista bilden representerar det tänkta händelseflödet mellan de olika bilderna från vänster till höger. Ovanför varje bild finns en rubrik som förtydligar vad bilden i fråga har för funktion.

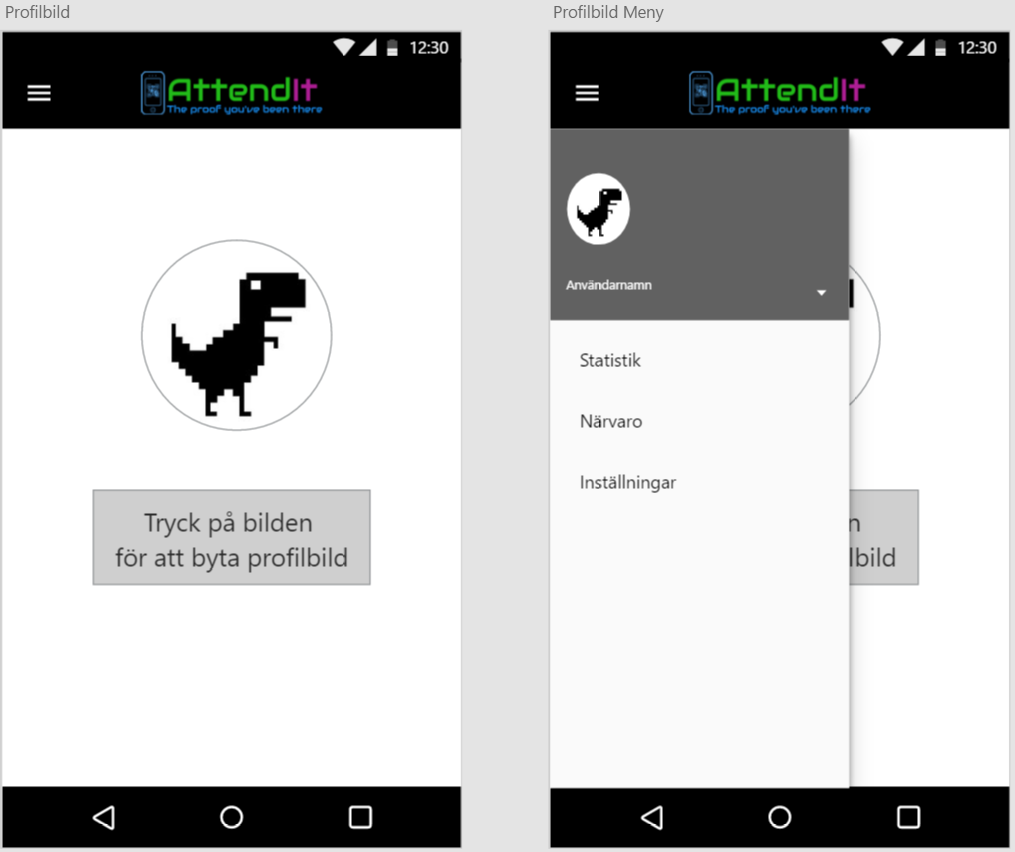
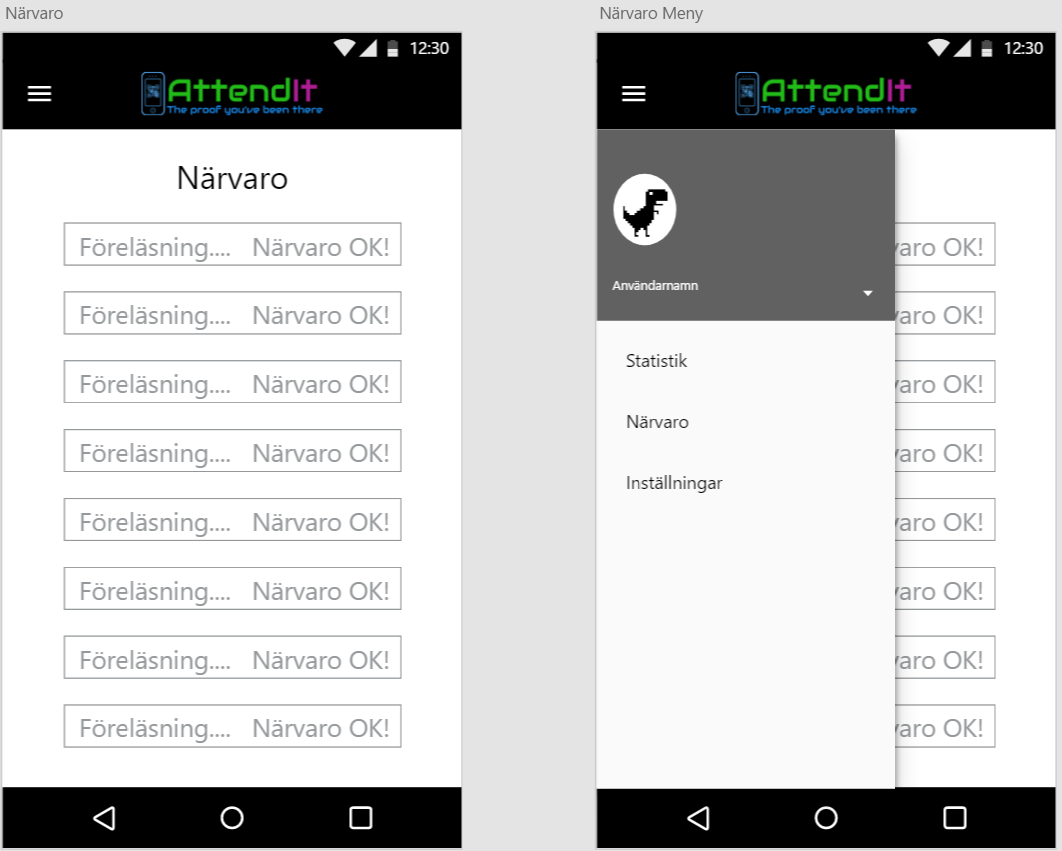














# 

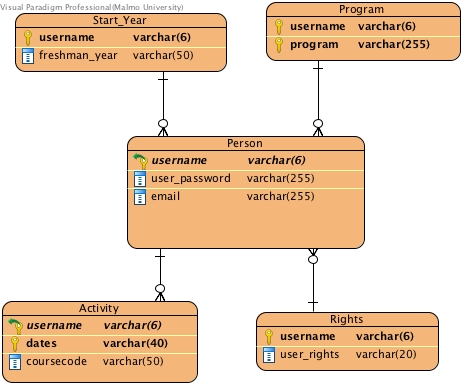
”En väldigt bra skiss av användargränssnittet finns om hur applikationen kommer att se ut, väldigt fint. Finns inget mer information utöver rubrikerna och bildernas kontenta som beskriver vad de olika delarna i applikationen gör. Gruppen av bilder med rubriker är lite väl små och svåra att tolka men det kompletteras med de större bilderna. Här kan ni skriva kort om vad varje sida visar.

# 

# 

# Entity Relationship diagram

Nedan finns ett ER-diagram som beskriver vilka entiteter som finns samt vilka attribut varje entitet har. Attributen är även beskrivna med vilken datatyp de är samt vilket/vilka attribut som är PK i varje entitet.



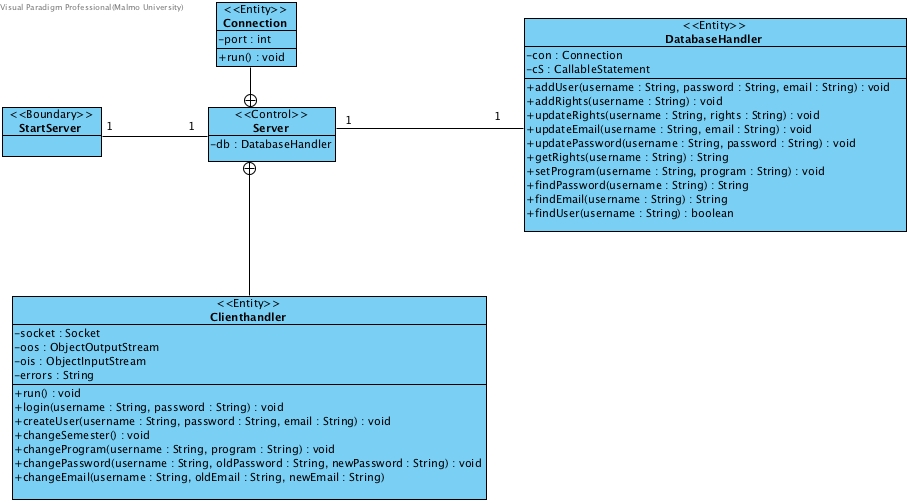
# 

”Ser bra ut, innehåller dess kontenta, inte så insatt i detta. Verkar rimlig och förståelig.

# Klassdiagram

Syftet med klassdiagrammen är att ge en överblick över hur klasserna kommer att se ut.

Klassdiagrammet nedan representerar arkitekturen över Server- och Databas-klassen. Databas-klassen innehåller metoder för att hantera data som finns lagrade, eller ska lagras, i databasen och vid anrop av dessa så utförs metoder i databasen.

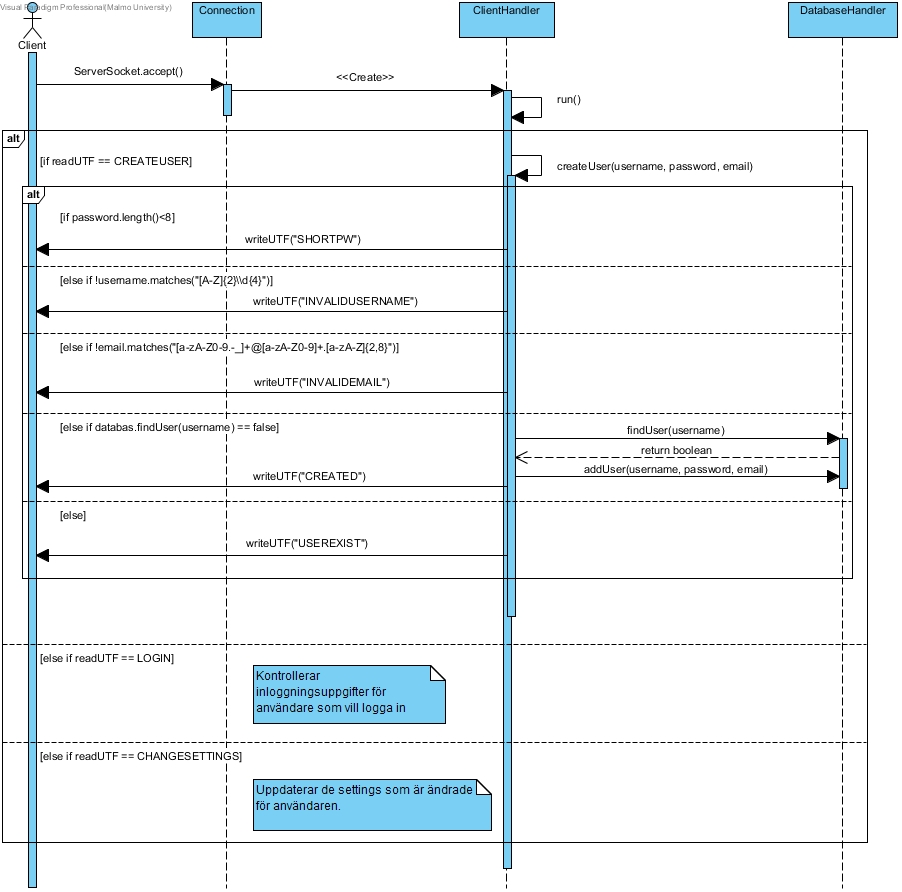


# 

# Sekvensdiagram

Syftet med sekvensdiagrammen är att ge en översikt hur vårt system kan och kommer fungera utifrån vissa användningsfall.

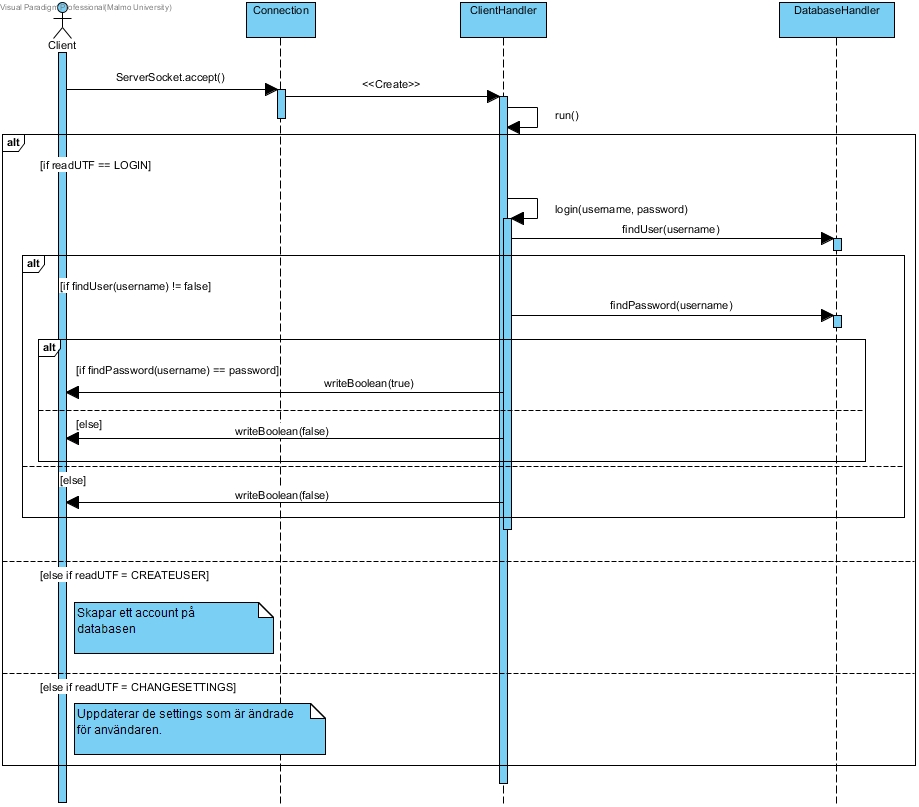
Sekvensdiagrammet nedan beskriver händelseförloppet när en användare skapas på servern.



# 

# 

Sekvensdiagrammet nedan beskriver händelseförloppet på servern när en klient försöker logga in.



”Diagrammen ser bra ut, förstår syftet med dessa vad de visar och vilken relevans de har för projektet/produkten. Hittade dock ett evil fel i de hela 😉. I sekvensdiagrammet för att skapa ett konto så finns det att man ska skapa namn, lösenord och e-mail och där så registreras man. Men i en av gränssnittsbilderna så har ni en skapa konto bild där man även anger årskull, program och extra kurs. Hmm hur hör detta ihop? 😉

Sekvensdiagrammet kanske inte är jättenödvändigt när man har resterande diagram ovan men förstår syftet.

Annars så ser allting bra ut, bra jobbat och fortsätt så. Fint, tydligt på bra detaljnivå.