

# Fickmonster

---

Projekt – Uppgift H

Adam El Soudi, Andreas Brundin, Gustav Grip, Marcus Skeholt och Valdemar Rydberg.

# Innehållsförteckning

Projekt – Uppgift H	1
Innehållsförteckning	2
Figurförteckning	3
Introduktion 1	4
Syfte 1.1	4
Uppgift 1.2	4
ER-diagram & DB-schema 2	5
Datan och Program 2.1	7
Queries 3	8
Q 1. 3.1	8
Q 2. 3.2	9
Q 3. 3.3	10
Q 4. 3.4	11
Q 5. 3.5	12
Q 6. 3.6	13
Q 7. 3.7	14
Q 8. 3.8	15
Q 9. 3.9	16
Q 10. 3.10	17
Arbetsrapport 4	18
Säkerhet, etik och integritet 5	19
Reflektioner 6	20
Källhänvisningar: 7	21

## Figurförteckning

Figur 1 - ER-diagramet.	5
Figur 2 - DB schema.	6
Figur 3 - Resultat av query 1.	8
Figur 4 - Resultat av query 2.	9
Figur 5 - Resultat av query 3.	10
Figur 6 - Resultat av query 4.	11
Figur 7 - Resultat av query 5.	12
Figur 8 - Vy till query 6.	13
Figur 9 - Resultat av query 6.	13
Figur 10 - Resultat av query 7.	14
Figur 11 - Resultat av query 8.	15
Figur 12 - Resultat av query 9 med en vän på ett gym.	16
Figur 13 - Resultat av query 9 om en användare inte har en vän.	16
Figur 14 - Resultat av query 10.	17

# 1. Introduktion

I denna rapport beskrivs tillvägagångssättet för en databaslösning för en mobilapplikation efter ett företags kriterier. Det som kommer finnas med är databasen och dess ER-diagram, 10 olika queries, säkerhet, etik och integritet, strategier för kriterier samt reflektioner.

## 1.1. Syfte

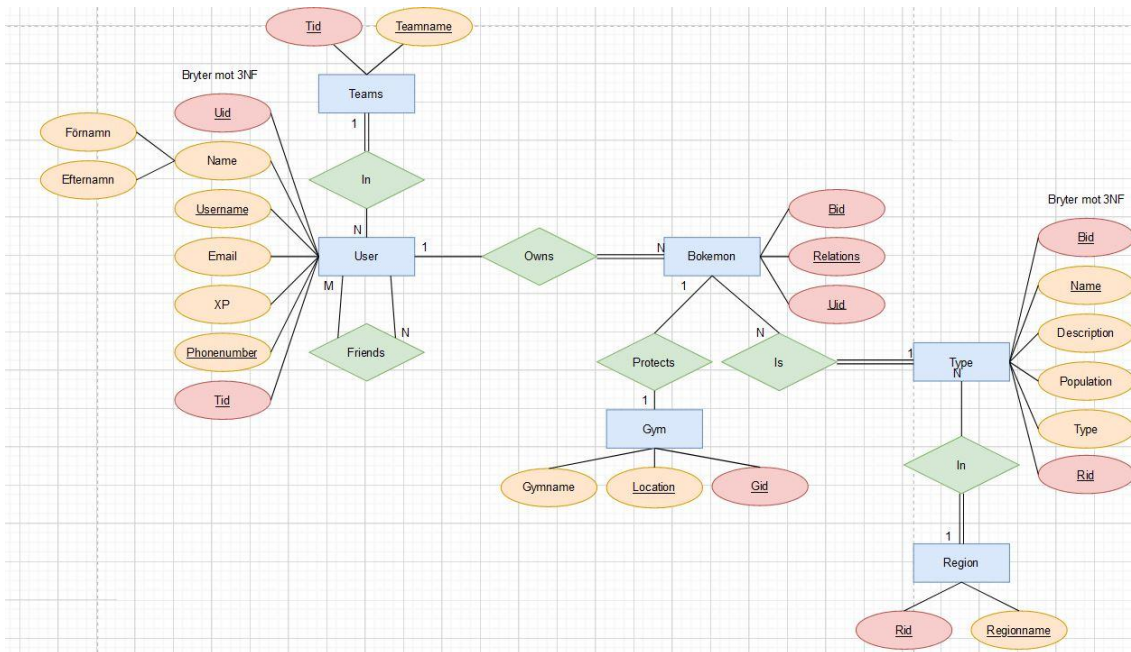
Syftet med uppgiften är att använda och fördjupa kunskaper i att göra ER-diagram, konvertera till tabeller, skapa en större databas i SQL, att i grupp argumentera och samarbeta på ett större problem för implementation, att skriva en arbetsrapport, presentera en databasdesign muntligt samt genomföra en teknikdemonstration.

## 1.2. Uppgift

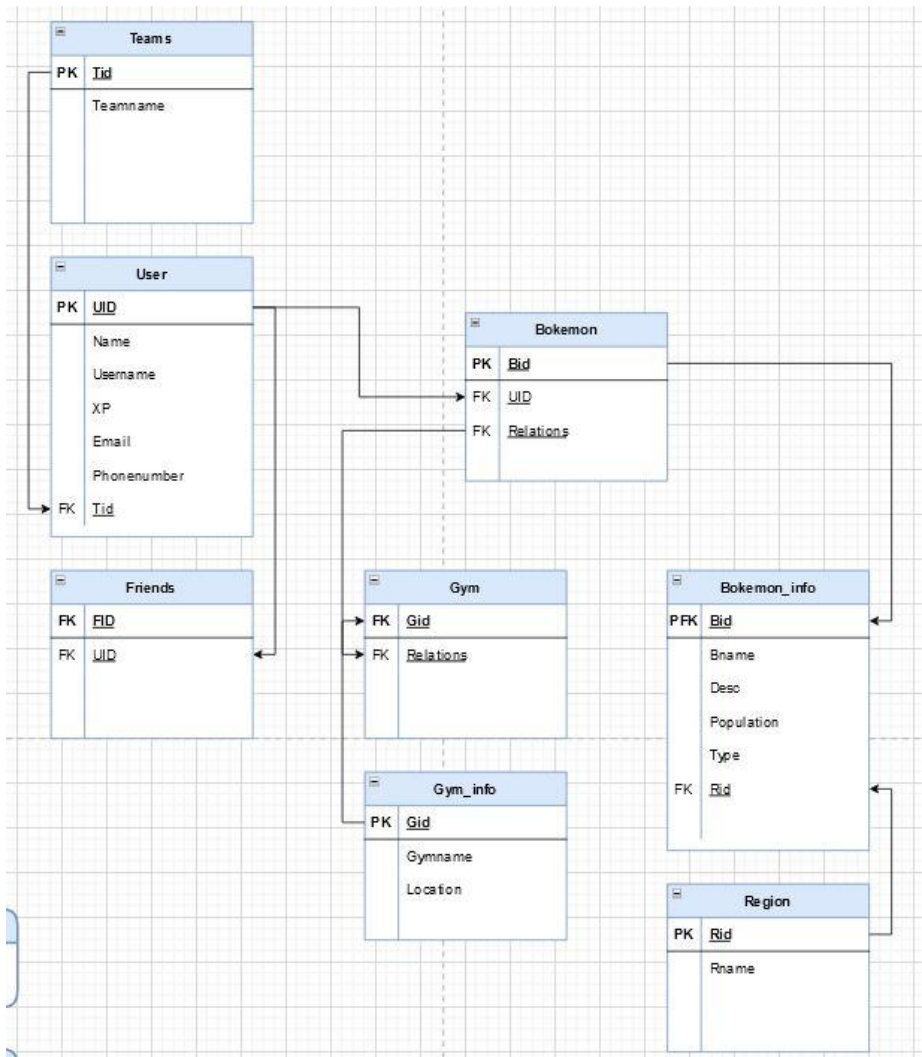
Uppdraget som tilldelats och som ska lösas är att bygga en databaslösning för en mobilapplikation där spelare samlar på olika monster även kallat Bokemon, databasen har olika kriterier som ska uppfyllas såsom att en spelare kan fånga och samla på Bokemon, tillhöra och vakta olika gym, tillhöra lag och vara vän med andra användare. Dessa kommer lösas genom olika queries vilket vi kommer fördjupa oss i senare i rapporten.

## 2. ER-diagram & DB-schema.

I detta avsnitt så kommer vi att gå igenom vårt ERD samt schema som vi skapat för att lösa denna uppgift. Men även det program som vi har utliserat för uppbyggnaden utav vår databas, samt den data som vi har använt oss utav.



Figur 1. ERD.



figur 2. Schemat.

För att kunna lösa denna uppgift så smidigt som möjligt har vi valt att lägga till våra egna "surrogatnycklar". Detta kan vara olika id:er såsom "Uid", "Fid" m.fl. Vi har markerat dessa med en röd bubbla i vårt ERD (se figur 1), så att man lättare kan urskilja dessa. Detta leder dock till att vi bryter mot den 3:e normalformen (3nf). Det betyder att det finns fler nycklar i en tabell istället för en enskild nyckel som är fullt funktionellt beroende. Som man kan se i figur 1 och 2 så bryter alla våra tabeller detta. Dock så har vi ansett det nödvändigt för att vi lättare ska kunna koppla ihop tabellerna. Då det kan bli svårt med de "naturliga nycklarna" såsom telefonnummer m.m. Där dessa inte refererar ordentligt till en annan tabell.

## 2.1. Datan och Program.

Programmet vi har använts oss utav har varit "SQLiteStudio". Detta då det var den som var lättast att ladda ner samt att aktivt kunna använda sig av. Det tog en liten stund innan vi helt förstod hur programmet beter sig samt fungerar till fullo, men efter lite labbande så förstod vi grunderna. För att se att vår databas som vi byggt upp aktivt fungerar så la vi in "mockdata". Detta är data som inte har någon riktig koppling till den riktiga världen. Utan används för det mesta att fylla ut databasen så att man kan se att den fungerar. Programmet som vi då använde oss utav var "Mockaroo". Det är ett ganska simpelt program där man som användare lätt kan skapa upp data som passar till de olika tabellerna. Problemet vi stötte på var just "Friends" tabellen som man kan se på figur 1 och 2. Detta då vi först tog olika siffror mellan 1-500 och slängde om dessa. Problemet var att om vän 1 är vän med vän 2 så måste vän 2 också vara vän med vän 1. Detta löste vi genom att först göra en lista med talen i storleksordning från 1-500 och sedan göra en ny rad med slumpmässiga siffror och la in dessa i vår databas. Därefter så tog vi raderna och vände på dessa i mockaroo och la in de igen i vår databas.


### 3. Queries

I detta avsnitt så kommer vi att visa våra queries som vi har fått i uppdrag att lösa. Det är totalt 10 stycken med unika resultat.

#### 3.1. Q1.

Lista namn på alla spelare samt hur många typer av monster de har “fångat”.

```
SELECT user.username AS Username, count(bokemon.uid) AS  
Total_Bokemons_Caught_per_Person  
FROM user  
JOIN bokemon  
ON user.uid = bokemon.uid  
GROUP BY user.username;
```

	Username	 Total_Bokemons_Caught_per_Person
1	pbrass19	26
2	awarn15	25
3	gmarlerp	24
4	alearman1u	23
5	brappport1p	23
6	mdegoey1g	23
7	kbenesevich2o	22
8	mfitzsymonds8	21
9	apenberthyw	20
10	hlorenzini1i	20
11	mwaddilove1d	20
12	nnaik1r	20

figur 3 resultat av q1



### 3.2. Q2.

Lista antal spelare som finns för de olika lagen.

```
SELECT teams.teamname AS Team_Name, count(user.tid) AS Users_per_Team
FROM teams
JOIN user
ON teams.tid = user.tid
GROUP BY teams.teamname;
```


	Team Name	Users per Team
1	Cyan	164
2	Magenta	171
3	Yellow	165

figur 4 resultat av q2

## 3.3 Q3.

Lista alla spelare med deras Level. Spelare med xp 0-1000 är level 1, XP 1001-3000 är level 2, xp 3001 - 6000 är level 3 osv.

```
SELECT user.username AS Username, user.xp AS XP,  
CASE  
WHEN user.xp < 1001 THEN "level 1"  
WHEN user.xp < 3001 THEN "level 2"  
WHEN user.xp < 6001 THEN "level 3"  
WHEN user.xp < 10001 THEN "level 4"  
WHEN user.xp > 10000 THEN "level 5"  
END AS "Level"  
FROM user  
GROUP BY user.username;
```

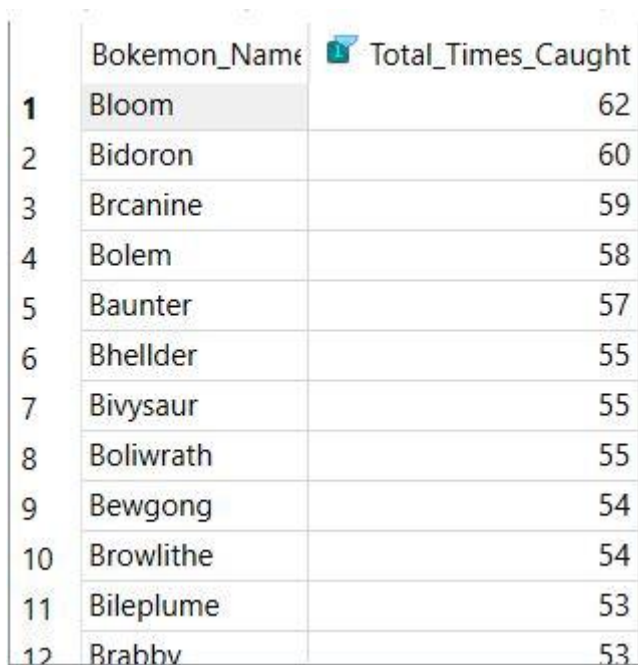
	Username	XP	 Level
24	mstopp5s	76	level 1
25	pfawderyb1	363	level 1
26	rrubinovitchdj	622	level 1
27	scallingham9k	71	level 1
28	smcgeoch6h	574	level 1
29	wbuserd7	4	level 1
30	wblunsden9g	507	level 1
31	aalston6e	2595	level 2
32	ahucknall1l	1505	level 2
33	asilverston53	2220	level 2
34	ballom37	1834	level 2
35	hbradburvcn	1867	level 2

figur 5 resultat av q3

## 3.4. Q4.

Lista alla typer av monster med namn och antal som finns i spelet (som är fångade av spelare)

```
SELECT bokemon_info.bname AS Bokemon_Name, count(bokemon.bid) AS  
Total_Times_Caught  
FROM bokemon_info  
JOIN bokemon  
ON bokemon_info.bid = bokemon.bid  
GROUP BY bokemon_info.bname;
```



	Bokemon_Name	Total_Times_Caught
1	Bloom	62
2	Bidoron	60
3	Brcanine	59
4	Bolem	58
5	Baunter	57
6	Bhellder	55
7	Bivysaur	55
8	Boliwrath	55
9	Bewgong	54
10	Browlithe	54
11	Bileplume	53
12	Brabbv	53

figur 6 resultat av q4

## 3.5 Q5.

Lista namn för den spelare som har flest vänner för varje lag.

```
SELECT * FROM  
(SELECT teams.teamname AS Team_Name, user.username AS Username, count(friends.fid)  
AS Most_Friends_per_Team  
FROM user  
JOIN teams
```

	Team Name	Username	Most Friends per Team
1	Cyan	drobardeti	7
2	Magenta	tsallery3	5
3	Yellow	mgritsaev78	7

figur 7 resultat av q5

## 3.6. Q6.

Lista alla typer av monster och antal av varje som den spelare som fångat flest monster har. I denna query så använde vi oss utav en vy. Man kan förklara en vy som en extra tabell som vi har skapat från andra queries och som man sedan kan joina in för att få ett resultat.

Vy:

```
CREATE VIEW best_dude_dudette AS
SELECT user.uid, count(bokemon.uid) AS numb_of_bokemons
FROM user
JOIN bokemon
ON user.uid = bokemon.uid
GROUP BY user.username
ORDER BY numb_of_bokemons desc
LIMIT 1;
```

	uid	numb_of_bokemons
1	46	26

figur 8 vy till q6

```
SELECT bokemon_info.bname AS Bokemon_Name, COUNT(bokemon_info.population) AS
Times_Player_Caught, user.username AS User
FROM user
LEFT OUTER JOIN bokemon ON bokemon.uid = user.uid
LEFT OUTER JOIN bokemon_info
ON bokemon_info.bid = bokemon.bid
JOIN best_dude_dudette
ON user.uid = best_dude_dudette.uid
GROUP BY bokemon_info.bname;
```

	Bokemon_Name	Times_Player_Caught	User
1	Bandshrew	2	pbrass19
2	Barasect	2	pbrass19
3	Barmeleon	2	pbrass19
4	Bentacool	2	pbrass19
5	Bulpix	2	pbrass19
6	Bachamp	1	pbrass19
7	Bachop	1	pbrass19
8	Badabra	1	pbrass19
9	Bagnemite	1	pbrass19
10	Baras	1	pbrass19
11	Bbra	1	pbrass19
12	Banmonth	1	pbrass19

figur 9 resultat av q6

### 3.7. Q7.

Lista namn på de tre typer av monster som finns minst antal av i spelet.

```
SELECT bokemon_info.bname AS Bokemon_Name, bokemon_info.population AS Population
FROM bokemon_info
ORDER BY population
ASC LIMIT 3;
```

	Bokemon Name	Population
1	Barfetch'd	51
2	Baunter	318
3	Baichu	428

figur 10 resultat av q7

## 3.8. Q8.

Lista medelvärde för antal fånga monster som spelare har av varje typ.

```
SELECT Bokemon_Name, ROUND (AVG(avg_caught)/1.0,2) AS  
Average_Number_Of_Times_Caught  
FROM
```

```
(SELECT bokemon.Uid AS UserID, bokemon_info.bname AS Bokemon_Name, COUNT(1)  
AS avg_caught  
FROM bokemon INNER JOIN bokemon_info ON bokemon.Bid = bokemon_info.bid  
WHERE Uid IS NOT NULL  
GROUP BY bokemon.Uid, bokemon_info.bname) AS MyView  
GROUP BY Bokemon_Name;
```

	Bokemon_Name	Average_Number_Of_Times_Caught
1	Bolbat	1.16
2	Plastpoise	1.16
3	Beepinbell	1.14
4	Bentacool	1.14
5	Bulpix	1.14
6	Bankey	1.12
7	Barasect	1.12
8	Baunter	1.12
9	Bddish	1.12
10	Bivysaur	1.12
11	Pulbasaur	1.12
12	Bonvta	1.11

figur 11 resultat av q8

## 3.9. Q9.

Visa lista med namnet på de gym som en användares kompisar har vakter på samt namnet på figuren som vaktar samt vännens namn.

```
SELECT user.username AS Friend_in_Gym, bokemon_info.bname AS
Friends_Bokemon_At_Gym, gym_info.gymname AS Gym_Name
FROM gym_info
JOIN bokemon_info
ON bokemon_info.bid = bokemon.bid
JOIN bokemon
ON bokemon.bid = bokemon_info.bid
JOIN gym_info AS gym1
ON gym1.gid = gym.gid
JOIN gym
ON gym.gid = gym_info.gid AND gym.relations = bokemon.relations
JOIN (SELECT user.uid AS friends_name
FROM user
JOIN friends AS J1
ON J1.uid = user.uid
JOIN friends AS J2
ON J2.uid = J1.fid
WHERE J1.fid = 25 /* 85 */
GROUP BY user.uid) as friend
ON friend.friends_name = bokemon.uid
JOIN user
ON user.uid = bokemon.uid
```

	Friend in Gvm	Friends Bokemon At Gvm	Gvm Name
1	gfursland1k	Bonyta	Resturant

figur 12 resultat av q9

Friend in Gvm	Friends Bokemon At Gvm	Gvm Name
---------------	------------------------	----------

figur 13 resultat av q9 om en användare ej har en vän på ett gym.



## 3.10. Q10.

Lista alla telefonnummer till alla spelare som har vakter på ett gym som är nära ett köpcentrum så de kan skicka specialerbjudanden om rabatter via sms.

```
SELECT user.phone AS User_Near_Mall
FROM user
JOIN bokemon
ON user.uid = bokemon.uid
JOIN gym
ON bokemon.relations = gym.relations
JOIN (SELECT gym_info.location, gym_info.gid
FROM gym_info
WHERE gym_info.location BETWEEN 40.6207889 AND 41.375722 /* 40.6207889 AND
41.375722*/ /* 10.3550571 AND 32.23483*/
GROUP BY gym_info.gymname) AS neargym
ON neargym.gid = gym.gid
GROUP BY user.phone
```

	User Near Mall
1	104-705-5515
2	136-935-9517
3	180-357-7895
4	285-406-5236
5	303-980-2698
6	411-112-8394
7	475-848-0716
8	536-338-5339
9	650-445-8374

figur 14 resultat av q10

## 4. Arbetsrapport

I detta kapitel kommer arbetets tillvägagångssätt, struktur och tidsuppskattning för de olika delarna att tas upp samt en kort reflektion om hur arbetet i gruppen har fungerat.

För att kunna lösa denna uppgift så har vi arbetat tillsammans för att få så mycket input vid varje steg som möjligt. Detta ansåg vi vara bättre än att dela upp oss då vi har kunnat kolla på varje steg från flera vinklar, diskutera och reflektera kring olika lösningar. Under vissa moment så har gruppen delat upp sig, för att lösa t.ex samma queries snabbare / effektivare. Det negativa med detta arbetssätt var att de olika momenten tog lite längre tid än om vi skulle delat upp oss mer frekvent och arbetat med en fas var, fördelarna ansågs vara större än nackdelarna då alla medlemmar i gruppen nu fick ta del och få lärdomar om hela uppgiften och inte bara inom vissa detaljerade moment.

Arbetet sköttes som sagt för det mesta i en enhet än enskilt. Tiderna som vi jobbade var oftast från klockan 11-17, med få avvikelser från tiderna. Samarbetet fungerade bra genom diskussioner inom gruppen vid problem och för att komma fram till en lösning som fungerar. Kommunikationen mellan gruppen sköttes primärt via Discord och fungerade i sin helhet bra för att spara och föra vidare information. Vi använde oss även av google drive för att spara och dela arbetet i gruppen.

Det spenderades olika tider på vardera moment. För ER-diagrammet så tog det runt 6-8 timmar för att få den klar. DB-schemat tog lite kortare tid, runt 1-2 timmar. Det majoriteten av tiden lades ner på var alla queries, efter många olika förslag så var alla queries klara efter 24 timmar arbete. Slutligen för att skriva rapporten tog omkring 5 timmar. Att lära sig grunderna i databashanteraren SQLiteStudio tog runt 2 timmar.

Allt som allt så fungerade arbetet bra, gruppens medlemmar gjorde det som krävdes. Arbetet kunde definitivt utföras mer effektivt såsom att dela upp de olika momenten lite mer, men i slutändan hinner vi klart till inlämnings- och presentationsdagen.

## 5. Säkerhet, etik och integritet

I detta avsnitt kommer säkerhet, etik och integritet att tas upp, vilka lagar som måste följas, hur man går tillväga för att följa reglerna samt hur arbetet har anpassat sig efter ovanstående punkter.

I denna uppgift fanns det några krav att förhålla sig till inom säkerhet, etik och integritet. För att kunna säkerställa dessa krav för användare så har vi använt oss utav funktionen "ON DELETE CASCADE", denna funktion fungerar så att om en användare tar bort sitt konto eller inte vill ha dess information kvar såsom personuppgifter och telefonnummer i databasen så kan vi ta bort allting gällande den specifika användare genom en vanlig delete istället för att manuellt gå igenom databasen för att ta bort allting som är relaterat till användaren.

Utöver detta så är det inget som är särskilt oetiskt kring det som vi samlar in om personerna i databasen. Den personliga datan som samlas in är för- och efternamn, mail samt telefonnummer. Vi samlar inte på några känsliga personuppgifter såsom etnicitet, religionstillhörighet m.fl.

De lagar som måste följas för att få denna applikation i drift är de GDPR lagarna som finns idag. Men även vilka rättigheter barn och unga personer har på nätet.

Det som behövs ta i hänsyn är vilka lagar och regler som gäller om man ska skapa en databas som hanterar olika personuppgifter då vissa uppgifter är otillåtna att samla på om man inte har en legitim anledning för att samla på dessa uppgifter. Sedan måste vi också ta hänsyn till GDPR. I denna lagstiftning så får man enbart samla på den datan som aktivt behövs, ingen onödig data ska förekomma från de personer som finns i databasen. Detta är något som vi har följt genom att enbart samla på telefonnummer, mail samt för- och efternamn. Ingen känslig data förekommer i vår databas.

[IMY integritet skyddsmyndigheten](#). 2021. Grundläggande Principer.

Även att vi har implementerat "ON DELETE CASCADE" vilket betyder att om en användare vill försvinna / tas bort från vår databas så kan detta ske snabbt och smidigt. Vi behöver inte modifiera vår design då när vi byggde den hade vi i åtanke att ta hänsyn till GDPR och dylikt.

Det som gäller för kund är naturligtvis att de följer de lagar som existerar idag. Såsom GDPR som nämnts tidigare i denna rapport. Men även då det som skapats är ett mobilspel så kommer det att locka yngre personer, då är det viktigt för vår kund att känna till barn och ungas digitala rättigheter. Några av de viktigaste punkterna som IMY tar upp på sin sida: [IMY Integritetsskydds myndigheten 2020. Barns och ungas rättigheter på digitala plattformar En vägledning till aktörer](#). Så finns det ett par viktiga punkter att känna till såsom:

“• Låt barnkonventionen och barnets bästa genomsyra er tjänst!.

- Ge information om hur uppgifterna om barn ska användas!  
Informationen ska vara barnanpassad när ni vänder er till barn.
- Motverka hot och hat på era plattformar och skydda barn från skadlig mediepåverkan”

## 6. Reflektioner

(den sista) En kompakt mening som tar upp flera saker - ni skulle enkelt kunna bryta ut i fler meningar och argumentera mer för de olika delarna; kunskapen har ökats - hur?

Samarbetet för problemlösning, på vilket sätt, vad var bra som ni gjorde?

Komma med förslag - hur hanterade ni det? Röstning eller som underlag till diskussion för vidareutveckling?

I detta stycke kommer reflektioner angående projektet att tas upp, vad fungerade bra respektive dåligt? Vad skulle vi kunna gjort annorlunda? Vad för kunskap och vilka lärdomar vi tar med oss.

Om vi skulle lösa uppgiften på nytt hade vi delat upp uppgiften bättre i början så vi kunnat hålla på med flera olika delar samtidigt för ett effektivare arbete. Vi skulle även ha lärt oss hur man delade databasen och enkelt lägga in den på en annan dator, nu gjorde vi i princip alla queries på en datorn som vi kopplade upp på en TV så alla såg vad som skedde och kunde komma med idéer.

De kunskaper vi tar med oss av detta projekt är många, det var en givande uppgift där lärdomar varit många. Kunskapen har ökats inom databaser, uppgifterna som skulle lösas var svårare än tidigare vilket medför att även lösningarna var mer komplicerade där ny kunskap inom området var tvunget att läras för att lyckas. Samtidigt var det viktigt att ha i baktanke att man inte använde någon olämplig data om användarna för att inte strida på GDPR som vi var inne på tidigare, detta medförde att man inte riktigt kunde knappa in det första som kom på tanken utan att i princip med alla val var man tvungen att ha detta i baktanke.

Detta var även första gången vi arbetat i en större grupp inom ämnet databaser vilket var nytt och spännande. Som vi var inne på ovan kunde uppdelningen varit bättre för att få mer effektivitet på arbetet men det är en lärdom vi tar med oss. Arbetet i gruppen fungerade i övrigt bra, samtliga var delaktiga och kom med idéer till lösningar på olika moment som vi testade och spann vidare på. Samarbetet fungerade även bra, det fanns en sund respekt gentemot varandra där alla var välkomna att komma med förslag om hur vi ska ta itu ett visst moment. I ett grupparbete är det lika viktigt att komma med förslag som att ta in och reflektera över andras tankar

## 7. Källhänvisningar

<https://www.imy.se/globalassets/dokument/ovrigt/barn-och-ungas-rattigheter-pa-digitala-plattfor-mar.pdf>

<https://www.imy.se/verksamhet/dataskydd/det-har-galler-enligt-gdpr/grundläggande-principer/>

<https://www.imy.se/nyheter/ny-vagledning-ska-starka-barns-och-ungas-rattigheter-pa-natet/>