**LoggiLue**



**Grupp 2**

**Kravdokument**

**V. 2.0**

**2017-04-21**

# Dokumenthistorik

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Beskrivning** | **Författare** |
| 170314 | 0.1 | Skapande av första versionen | Kajsa, David, Hannes |
| 170315 | 0.2 | Fortsättning av första versionen | Kajsa, David, Hannes |
| 170315 | 0.3 | Angett motivering för krav FAK-2 | Oscar |
| 170316 | 0.4 | Fortsättning av första versionen | Kajsa, David, Hannes |
| 170321 | 1.0 | Fortsättning av första versionen | Kajsa, David, Hannes |
| 170331 | 1.1 | Ändringar efter retrospektmöte: versionsnummer, bytt upplägg för funktionella krav, förtydligande av krav FAK-4 & FAK-5, omformuleringar i produktbeskrivningen för objektivitet, förtydligat målgruppen, lagt till utvecklare under intressenter, förklaring av prefix för kvalitativa krav, versionsnummer i dokumenthistoriken. | Kajsa |
| 170411 | 1.2 | Borttagning av ”vi”. Korrigering av stavfel. Infogande av referens till MoSCoW-modellen. Motiveringar till FAK-2.1 FAK- 5, FAA- 3. | David, Oscar, Kajsa Hannes |
| 170421 | 2.0 | Små ändringar efter andra retrospektmötet och utifrån Kristinas kommentarer. | Kajsa |
| 170424 | 2.01 | Diverse små ändringar efter feedback från RS2. | David |
| 170426 | 2.1 | Ändringar efter retrospektmöte 2: Tagit bort sista överflödiga stycket i produktbeskrivningen, lagt till front-end, back-end, JavaScript, Python, HTML och CSS i ordlistan. | Kajsa |

**Innehåll**

[Dokumenthistorik 2](#_Toc481653444)

[Kravdokument 4](#_Toc481653445)

[Syfte 4](#_Toc481653446)

[Ordlista 4](#_Toc481653447)

[Referenser 4](#_Toc481653448)

[Produktbeskrivning 5](#_Toc481653449)

[Målgrupp 5](#_Toc481653450)

[Intressenter 5](#_Toc481653451)

[Funktionella krav 6](#_Toc481653452)

[Konto 6](#_Toc481653453)

[Anteckningar 7](#_Toc481653454)

[Schema 8](#_Toc481653455)

[Kvalitativa krav 9](#_Toc481653456)

[Produktkrav 9](#_Toc481653457)

[Organisationskrav 9](#_Toc481653458)

[Externa krav 10](#_Toc481653459)

# Kravdokument

# Syfte

Syftet med dokumentet är att kunna definiera alla krav och funktioner som projektet kommer att innehålla. Det kommer att ändras med projektets gång då nya krav och funktioner kommer uppkomma eller behöva omprioriteras. Detta dokument kommer endast innefatta funktioner och krav, inte planering eller liknande.

# Ordlista

*KronoX* Är ett övergripande system som används för schemaläggning såväl som bokningar för lokaler och dylikt, anpassat för användning av högskolor och universitet.

*It’s learning*  Webbplats för studenter och lärare.

*MySQL*  Databasplattform som används för att skapa databaser.

*YouTrack*  Webbplats för att logga tid i projekt.

*Procedurmanual*  Stilguide och verktygs förklaring. Vilka procedurer som kommer ingå i projektet.

*Front-end* Användargränssnittsorienterad bearbetning som ofta förknippas med språk som HTML, CSS och JavaScript

*Back-end* Själva basbearbetningen.

*HTML5* HyperText Markup Language, ett märkspråk för hypertext

*CSS3* Cascading style sheets, ett språk som beskriver presentationsstilen för ett strukturerat dokument

*Python* Ett programmeringsspråk.

*Javascript* Ett dynamiskt skriftspråk som främst används på klientsidan i webbtillämpningar.

# Referenser

[1] S. Eklund.(2010). *Arbeta i Projekt.* 3 upplaga. Studentlitteratur. Lund.

# Produktbeskrivning

Produkten som ska skapas är en dynamisk webbsida där man ska kunna dela med sig och ta del av anteckningar som har blivit uppladdade av andra studenter. Skulle man missa en föreläsning eller helt enkelt inte bara hann med att anteckna så ska möjligheten finnas att på denna webbsida kunna ta del av andras anteckningar samt dela med sig av egna anteckningar. Anteckningarna ska kopplas till föreläsningar och liknande händelser via Kronox så de enkelt kan hittas med hjälp av ett schema, detta för att införa kategorisering i informationsflödet.

Det blir enklare om man hjälper varandra, genom att dela med sig, samt införandet av ett belöningssystem så ska den som skapar bra anteckningar också få en gratifikation tillbaka på det. På webbsidan ska man med namn och lösenord kunna logga in på sitt konto. Kontot ska kopplas till det program användaren studerar (till en början med endast för informationsarkitekter men om tiden finns kan fler program implementeras). Därifrån ska man kunna välja vilken kurs, och i sin tur vilken föreläsning eller liknande händelse på schemat som är av intresse. Användaren ska där kunna se tillhörande inlägg från händelsen. Inläggen är i form av anteckningar, bilder och annat studiematerial man fått med sig från föreläsningen.

# Målgrupp

Målgruppen för LoggiLue kommer att vara studenter på Malmö högskola som läser programmet informationsarkitekt. Studenten har bra teknisk kunskap men hemsidan kommer vara fokuserad på användbarhet ändå för att tilltala så många användare som möjligt. Systemet kommer, som sagt, till huvudsaken att fokusera på studenter på informationsarkitektprogrammet och kommer därav att använda sig av ett enkelt gränssnitt för att det ska gå snabbt och enkelt för användaren att både hitta och ladda upp anteckningar.

# Intressenter

Högskolor kan ses som intressenter då kommuner och skolor eventuellt i framtiden skulle kunna köpa in LoggiLue för att erbjuda det till studenterna.

Studenterna på högskolorna blir slutanvändarna av LoggiLue.

Externa intressenter skulle kunna vara KronoX och It’s learning då de möjligtvis skulle vilja implementera funktionaliteten i sina system för att ha allt som gäller skolan på samma sida.  
Även utvecklarna, de som skapar produkten, kan ses som intressenter.

# Funktionella krav

Alla funktionella krav kommer prioriteras efter MoSCoW modellen[1]. Kraven är kopplade med ett bokstavsprefix där första bokstaven står för om det är ett funktionellt krav (F) eller ett kvalitativt krav(K). Andra bokstaven specificerar om det är ett användarkrav(A) eller ett systemkrav(S). De funktionella kraven specificerar de funktioner webbsidan ska innehålla. Dessa kategoriseras i följande underrubriker; konto(K), anteckningar(A) och schema(S). Sista bokstaven specificerar under vilken kategori gruppen har valt att sortera kraven i.

De funktionella kraven är uppdelade i två delar; systemkrav och användarkrav. Systemkraven är mer detaljerade krav och nedbrutna av användarkraven för att mer fungera som instruktioner för skapande av hemsidan. Alla systemkrav kommer för tillfället inte skrivas klart eftersom tekniska kunskaper om hur databaser och KronoX fungerar saknas. Dessa kommer därför att fyllas på under tiden som förståelsen ökar och de kommer alla att utgå ifrån användarkraven och bryta upp dem i mindre mer detaljerade systemkrav.

## Konto

Krav som involverar själva kontot för systemet. Hur de ska fungera och vad det ska innehålla.

### Användarkrav

**ID:** FAK-1 **Namn:** Registrering

**Prioritering:** Hög (must)

Man ska kunna registrera sig på webbsidan. När du registrerar skall man fylla i användarnamn, lösenord och välja vilken klass du tillhör.

*Utan registrering kan du inte göra någonting på sidan. Därför är det mycket viktigt.*

**ID:** FAK-2 **Namn:** Logga in

**Prioritering:** Hög (must)

Man ska kunna logga in på webbsidan med användarnamn och lösenord

*Inloggningen är nödvändig så att den registrerade ska få tillgång till alla funktioner som sidan använder.*

**ID:** FAK-2.1 **Namn:** Krav på inloggning

**Prioritering:** Hög (must)

Om man inte loggar in på hemsidan så kommer man inte kunna dela eller få tillgång till anteckningar.  
*Man måste bli medlem för att det ska finnas en kontroll över vad som publiceras och av vem.*

**ID:** FAK- 3 **Namn:** Logga ut

**Prioritering:** Hög (must)

Man ska kunna logga ut från sin användare från webbsidan.

**ID:** FAK-4 **Namn:** Logga ut automatiskt efter inaktivitet.

**Prioritering:** Låg (could)

Systemet skall logga ut användaren efter en bestämd tid ifall denna är inaktiv. Denna tid är inte än definierad.

*Så ingen missbrukar någons konto.*

**ID:** FAK-5 **Namn:** Återfå borttappat lösenord

**Prioritering:** Låg (could)

Systemet ska kunna ge tillbaka ett borttappat lösenord till användaren genom mailen.  
*Användaren måste ha möjlighet att återfå sitt konto utifall man glömmer eller tappar bort lösenordet.*

### Systemkrav

**ID:** FSK-1 **Namn:** Registrering

**Prioritering:** Hög (must)

En front-end del där användaren ska kunna skriva in sina uppgifter och välja vilket program den går.

**ID:** FSK-2 **Namn:** Registrering till databas

**Prioritering:** Hög (must)

En back-end del som hämtar det användaren skriver in och skickar det till databasen.

## Anteckningar

Krav som involverar anteckningarna och deras funktionalitet.

### Användarkrav

**ID:** FAA- 1 **Namn:** Lägga till en anteckning

**Prioritering:** Hög (must)

Kunna lägga upp en anteckning kopplad till en föreläsning från användarens schema.

**ID**: FAA-1.1 **Namn**: Ladda upp en fil

**Prioritering**: Medel (should)

Kunna ladda upp en anteckning om den är sparad i ett filformat eller en bild.

**ID:** FAA- 2 **Namn:** Visa en anteckning

**Prioritering:** Hög (must)

Kunna välja en anteckning från användarens schema och öppna den.

**ID**: FAA-2.1 **Namn**: Ladda ner anteckning

**Prioritering**: Medel (should)

Kunna välja alternativet att ladda ner en anteckning.

**ID**: FAA-3 **Namn**: Rankning av anteckningar

**Prioritering**: Låg (could)

Kunna ranka anteckningar baserat på deras kvalitet.  
*Vill ge beröm till bra anteckningar så att användaren känner att det är värt det.*

### Systemkrav

**ID:** FSA-1 **Namn:** Lagring av anteckningar

**Prioritering:** Hög (must)

Anteckningar kommer sparas i en databas som kommer ligga uppe på en server.

## Schema

Krav som involverar schemat.

**ID:** FAS- 1 **Namn:** Se schema som är kopplat till sitt program.

**Prioritering:** Hög (must)

Användaren skall kunna se sitt schema från sin utbildning på hemsidan.

*Produkten bygger på att kategorisera anteckningarna efter lektion/föreläsning.*

**ID:** FAS- 2 **Namn:** Kunna välja en föreläsning.

**Prioritering:** Hög (must)

Genom att trycka på föreläsningen visas alla anteckningar för den föreläsningen.

**ID**: FAS- 3 **Namn**: Visa anteckningar för kurs.

**Prioritering**: Medel (should)

Kunna sortera utifrån kurs och därmed kunna browsa bland alla de uppladdade anteckningarna för den kursen.

### Systemkrav

**ID:** FSS- 1 **Namn:** Hämta data från KronoX schema.

**Prioritering:** Hög (must)

Hämta data från KronoX schema.

*Produkten bygger på att kategorisera anteckningarna efter lektion/föreläsning.*

**ID:** FSS- 2 **Namn:** Spara data från KronoX schema.

**Prioritering:** Hög (must)

Spara data från KronoX-schemat.

*Produkten bygger på att kategorisera anteckningarna efter lektion/föreläsning.*

**ID:** FSS- 3 **Namn:** Presentera data från KronoX schema.

**Prioritering:** Hög (must)

Presentera data från KronoX schema på hemsidan.

*Produkten bygger på att kategorisera anteckningarna efter lektion/föreläsning.*

**ID:** FSS- 4 **Namn:** Lektionerna blir klickbara.

**Prioritering:** Hög (must)

Gör varje lektion/föreläsning klickbar med ett grafiskt gränssnitt.

**ID:** FSS- 5 **Namn:** Unik sida för varje lektion.

**Prioritering:** Hög (must)

Varje lektion/föreläsning är en unik sida för anteckningar.

**ID**: FSS- 6 **Namn**: Bläddra bland anteckningar

**Prioritering**: Medel (should)

Hämta och visa alla anteckningar kopplat till den kurs man valt.

# Kvalitativa krav

Alla kvalitativa krav kommer prioriteras efter MoSCoW modellen[1]. Kvalitativa krav (alternativ icke-funktionella krav) är krav som påverkar systemet som helhet. De delas in i kategorien produktkrav, organisationskrav samt externa krav. Prefixerna är givna på liknande sätt som för de funktionella; den första bokstaven anger om det är ett kvalitativt krav (K) eller ett funktionellt krav (F). Den andra bokstaven är baserade på underkategorierna; produktkrav (P), organisationskrav (O) och externa krav (E).

## Produktkrav

Krav som specificerar beteendet av mjukvaran.

**ID:** KP-1 **Namn:** Användarvänlig

**Prioritering:** Hög (must)

Sidan ska visas med hjälp av en användarvänlig design och gränssnitt.

*Viktigt att sidan är användarvänlig då det är ett stort fokus för hela produkten.*

**ID:** KP-2 **Namn:** Tillgänglig

**Prioritering:** Hög(must)

Sidan ska alltid vara åtkomlig.

*En webbsida förväntas alltid finnas online och vara tillgänglig för kunden.*

## Organisationskrav

Vissa organisationer har information angående sina krav presenterade i en stilguide eller en procedurmanual.

**ID:** KO-1 **Namn:** Front-end

**Prioritering:** Hög (must)

Front-end kommer att skrivas med HTML5, CSS3 & Javascript.

*Det är dem språken projektgruppen behärskar*

**ID:** KO-2 **Namn:** Back-end

**Prioritering:** Hög (must)

Back-end kommer att skrivas med Python 3 tillsammans med en databas i MySQL.

*Python 3 är det språket projektgruppen behärskar. MySQL passar bra tillsammans med Python 3.*

**ID:** KO-3 **Namn:** Versionshantering

**Prioritering:** Hög (must)

Versionshantering av systemet kommer att skötas via Git.

**ID:** KO-4 **Namn:** Tid-logging.

**Prioritering:** Hög(must)

Verktyget YouTrack används av samtliga i gruppen för att logga tiden som varje medlem arbetar.

*Viktigt att alla använder samma verktyg för att logga arbetstid.*

## Externa krav

Krav som kommer från utsidan. **Just nu finns inga externa krav.**