



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

# Mobilapplikation för intranät

Examensarbete inom högskoleingenjörsprogrammet i Datateknik

Filip Svensson

Ernst Näslund

**INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK**

## **Abstract**

Broccoli Engineering AB is a consultant company based in Gothenburg. Currently their internal communication is mainly by email. For their company events a lot of administrative work is created from sending out invitations, reminders and summarizing a list of the participants. This work aims to create a mobile application that can be used by the employees to respond to events, receive reminders and have an overview of current and upcoming events. The report describes the implementation and the design that has been used to create this application. The result is a prototype that handles events and reminders. Preparations for further development are made in form of graphical components but needs functionality that has not yet been implemented due to the existing intranet being under development. The prototype can be used to automate a part of the administrative work and free resources.

**Keyword:** Mobile application, cross-platform, Flutter, HR-management, handle events

## **Sammanfattning**

Broccoli Engineering AB är ett konsultbolag baserat i Göteborg. I dagsläget sker deras interna kommunikation huvudsakligen via epost. Vid event på företaget skapas mycket administrativt arbete med att maila ut inbjudningar, påminnelser och summering av deltagare. Detta arbete syftar till att ta fram en mobilapplikation som kan användas av företagets anställda för att svara på event, få påminnelser samt få en överblick över kommande event. Rapporten beskriver konstruktionen och designen som använts vid applikationens skapande. Resultatet är prototyp som hanterar event och påminnelser. Förberedelser för vidareutveckling är gjorda i form av grafiska komponenter, men kräver funktionalitet som inte implementerats än då deras intranät är under utveckling. Prototypen kan användas för att automatisera en del av det administrativa arbetet och på så sätt frigöra resurser.

**Nyckelord:** Mobilapplikation, cross-platform, Flutter, HR-management, eventhantering

## **Förord**

Denna rapport är skriven av Filip Svensson och Ernst Näslund och gjordes som ett examensarbete om 15 högskolepoäng under vårterminen 2023

Vi vill tacka vår handledare Björn Bergholm samt Åsa Vikström, HR-chef, på Broccoli Engineering AB för stöd och återkoppling under arbetets genomförande.

Vi vill även rikta ett stort tack till Anna Manfredsson, Anders Truong och Elma Esati som har varit till stor hjälp under utvecklingen.

Till sist vill vi tacka Pelle Evensen som har varit vår handledare på Chalmers och hjälpt oss med denna rapport samt gett kontinuerlig feedback under hela projektet.

## Terminologi

<b>Backend</b>	Mjukvara för applikationen, användaren kan inte se eller interagera med denna
<b>Frontend</b>	Det grafiska som visas på applikationen som användaren ser och kan interagera med
<b>Backlog</b>	En lista prioriterad lista med uppgifter som är planerade att utföras
<b>IDE</b>	Mjukvaruprogram som används för att skriva och köra kod.
<b>Android</b>	Operativsystem för smartphones
<b>iOS</b>	Operativsystem för smartphones
<b>Firebase</b>	En online plattform med verktyg för att bygga och utveckla applikationer för Android, Web och iOS.
<b>FCM</b>	En tjänst inom Firebase som gör det möjligt att skicka ut notifikationer till användarna av en applikation.
<b>Flutter</b>	Cross-platform-ramverk utvecklat av Google
<b>Dart</b>	Objektorienterat programmeringsspråk
<b>Molnservice</b>	Ett lagringsutrymme som existerar på internet
<b>Emulator</b>	Mjukvara som efterliknar funktioner hos en viss hård- och mjukvara

# Innehåll

<b>1. Inledning</b> .....	7
1.1 Syfte.....	7
1.2 Mål.....	7
1.3 Avgränsningar .....	7
<b>2. Teknisk Bakgrund</b> .....	8
2.1 Flutter.....	8
2.2 Android Studio .....	8
2.3 Firebase.....	8
2.4 Scrum.....	9
2.5 Figma .....	9
<b>3. Metod</b> .....	10
3.1 Förberedande arbete .....	10
3.2 Agilt arbetssätt.....	10
3.3 Utvärdering och utformning .....	10
<b>4. Utveckling</b> .....	11
4.1 Navigeringsflöde .....	11
4.2 Inloggning.....	12
4.3 Hemskärm.....	13
4.4 Kalenderskärm.....	14
4.5 Eventskärmar .....	15
4.6 Aviseringsskärm .....	16
4.7 Sidomeny .....	17
<b>5. Resultat</b> .....	18
<b>6. Diskussion</b> .....	19
6.1 Utvärdering av arbetssätt och resultat .....	19
6.2 Påverkan på samhället .....	19
6.2.1 Samhälleliga Aspekter .....	19
6.2.2 Etiska Aspekter .....	19
6.2.3 Ekologiska Aspekter .....	19
6.3 Framtida Utvecklingsmöjligheter .....	20
6.3.1 Vidareutveckling av notifikationer .....	20
6.3.2 Tidigare implementering av Firebase .....	20
6.3.3 Testning på iOS hårdvara.....	20
6.3.4 Omkonstruktion med annat ramverk.....	20

<b>7. Slutsats</b> .....	21
<b>Referenser</b> .....	22
<b>Appendix A – Gantt-schema</b> .....	23

## 1. Inledning

Broccoli Engineering AB är ett konsultbolag inom hård- och mjukvaruutveckling baserat i Göteborg. I dagsläget sker företagets interna kommunikation huvudsakligen via e-post. Detta resulterar i mycket administrativt arbete vid exempelvis interna evenemang. I dessa fall behöver HR-avdelningen skicka ut inbjudan, följt av eventuella påminnelser för att sedan summera svaren på dessa. För att underlätta kommunikationen vill Broccoli utveckla en mobilapplikation som de anställda kan installera på sina jobbtelefoner. Denna applikation ska spegla intranätet och tillhandahålla funktionalitet för kommunikation och administrativa uppgifter.

### 1.1 Syfte

Syftet med projektet är att utveckla en plattformsoberoende mobilapplikation som är integrerad med Broccolis befintliga intranät och som tillhandahåller en effektiv kommunikationskanal mellan företagets avdelningar och anställda. Applikationen syftar till att underlätta kommunikationen till anställda genom broadcast funktioner samt effektivisera arbetet för HR-avdelningen genom att erbjuda en helhetslösning för evenemangshantering.

### 1.2 Mål

Målen för applikationen är följande:

- Applikationen ska anpassas för att kunna köras både på Android och iOS operativsystemen
- Applikationen ska innehålla en kalenderfunktion som hämtar data om evenemang från en gemensam databas
- Applikationen ska visa anställdas kontaktuppgifter och nuvarande uppdrag
- Applikationen ska innehålla en broadcast funktion som kan meddela alla anställda
- Applikationen ska hantera anmälan till olika event

### 1.3 Avgränsningar

Applikationens fokus kommer att ligga på framtagandet av en enkel frontenddesign samt implementation av evenemang- och broadcast-funktionalitet. Eftersom applikationen kommer användas av Broccolis intranät, under konstruktion, kommer en prototyp produceras där funktionaliteten kan komma att begränsas av Broccolis interna utvecklingsprocess.

## 2. Teknisk Bakgrund

Detta kapitel innehåller en beskrivning av de verktyg och metoder som använts vid framställandet av applikationen.

### 2.1 Flutter

Då specificeringen efterfrågar en lösning som ska fungera både på iOS och Android utvärderades möjligheten att använda sig av ett cross-plattform-ramverk för att spara tid under arbetet. Valet av ramverk föll på Flutter då det uppfyller kraven som ställs och ansågs vara kapabelt till att erbjuda en helhetslösning.

Flutter [1] är ett cross-plattform-ramverk från Google och använder sig av programmeringsspråket Dart. Ramverket tillåter utvecklaren att använda en stor del av kodbasen gemensamt för flera plattformar så som iOS och Android. Detta sparar tid då behovet av att utveckla flera kodbaser försvinner och utvecklingsprocessen går fortare.

För att producera en lösning som är visuellt likadan på olika plattformar utnyttjar Flutter inte några inbyggda grafiska komponenter hos de olika systemen som vissa andra cross-plattform-ramverk gör. I stället erbjuder ramverket en uppsättning egna komponenter för att kunna rendera dessa exakt likadant på alla enheter som Flutter kan köras på. Dessa komponenter kallas för widgets och utgör grundkomponenterna för hur applikationer byggs med Flutter.

En annan funktion Flutter erbjuder är så kallad "Hot Reload" vilket gör det möjligt att se kodjusteringar nästan direkt på telefonen eller simulatorn. Detta underlättar och påskyndar utvecklingsprocessen då endast delarna som påverkas av förändringen behöver kompileras om vid varje ändring.

### 2.2 Android Studio

Android Studio är en så kallad IDE (Integrated Development Environment) anpassad för att utveckla mobilapplikationer för Android. Android Studio erbjuder en emulator vilket ger möjlighet att se applikationens utseende och funktion utan fysisk hårdvara [2]. Eftersom det finns officiellt stöd för Flutter inom Android Studio valdes denna IDE för att smidigt kunna exekvera koden och testa applikationen.

### 2.3 Firebase

Firebase är en uppsättning med tjänster som underlättar implementationen av olika funktioner i en applikation, till exempel användarautentisering, databaser och push-notifikationer. Denna tjänst drivs av Google och all information sparas i deras molnservice, möjliggör en direkt kommunikation med backend utan att behöva bygga en egen mellanhand[3].

FCM [4] (Firebase Cloud Messaging) är en tjänst inom Firebase som hanterar meddelanden till användare över alla plattformar, vare sig det är web, Android eller iOS. Tjänsten består av fyra lager. Först komponeras meddelandet som ska skickas med tillhörande text, plattform och till vilka användare som ska ta emot notifikationen på Firebase hemsida, sedan hanteras meddelandet i Firebases egna backend innan det skickas vidare till transportlagret där meddelandet vet vilken plattform det ska levereras till. Push-notifikationer är möjliga att göra

i Flutter utan någon tredjepart, dock gör Firebase det väldigt enkelt att implementera eftersom man inte behöver implementera backend logik. Därför valdes Firebase.

## 2.4 Scrum

Scrum är ett arbetssätt skapat för att öka produktivitet inom projekt [5]. I Scrum finns det en "Product Owner" vilket projektet levereras till, en "Scrum master" som styr grupper med utvecklare vars ansvar är att utföra olika uppgifter. Arbetet sker agilt och delas upp i så kallade "sprints". En sprint består av en mängd uppgifter som står skrivna i en "backlog" och som ska utföras inom en viss tidsram. Efter en sprint diskuterar utvecklare framsteg och eventuella problem tillsammans med scrum master och product owner för att bestämma nästa sprint. Scrum valdes som arbetsmetod för att få hög flexibilitet och kontinuerlig feedback under arbetet.

## 2.5 Figma

Figma [6] är ett onlinebaserat designverktyg som används för att skapa hemsidor, applikationer, logotyper med mera. Verktöget är molnbaserat med möjlighet till samarbete mellan olika användare samtidigt. Broccoli har redan en uppsatt arbetsyta på Figma med de färgkoder och ikoner som ska användas av applikationen.

### **3. Metod**

Detta avsnitt beskriver det förberedande arbetet följt av det agila arbetssättet som använts. Därefter nämns hur utvärderingen gått till genom projektets gång.

#### **3.1 Förberedande arbete**

För att ta fram prototypen delades arbetet upp i två primära delar, en planeringsfas där tillvägagångssätt av programmeringsspråk och möjliga designmönster bestämdes och en implementationsfas där programmeringen utfördes. Ett Gantt-schema gjordes för att bestämma tidsfördelningen av projektet med möjlighet till mindre ändringar efter behov (se appendix A). Tiden för utvecklingen av applikationen spenderades till 100% på Broccolis kontor efter förfrågan från företaget.

#### **3.2 Agilt arbetssätt**

För att uppnå hög effektivitet kommer vi använda oss av scrum-metoden där onlinevertyget Trello används för att hålla en backlog på de olika uppgifter som skall utföras. Möten kommer ske veckovis med både handledare och vår product owner, HR-avdelningen, på företaget för kontinuerlig feedback på funktionalitet och design. Ungefär var fjärde vecka kommer längre möten ta plats med en full demo av den nuvarande applikationen för att få feedback på allmän funktionalitet och användarvänlighet.

#### **3.3 Utvärdering och utformning**

Under utvecklingen testades ny funktionalitet vid mötena med HR-avdelningen som direkt kunde ge feedback och önskemål kunde justeras. Utvecklingen av applikationen gjordes med hjälp av ramverket Flutter och Android Studio. Applikationen utvecklades med riktlinjer om att hålla navigeringen enkel. Det skulle vara lätt att hitta informationen och genom få knapptryck snabbt få åtkomst till de funktioner man ville.

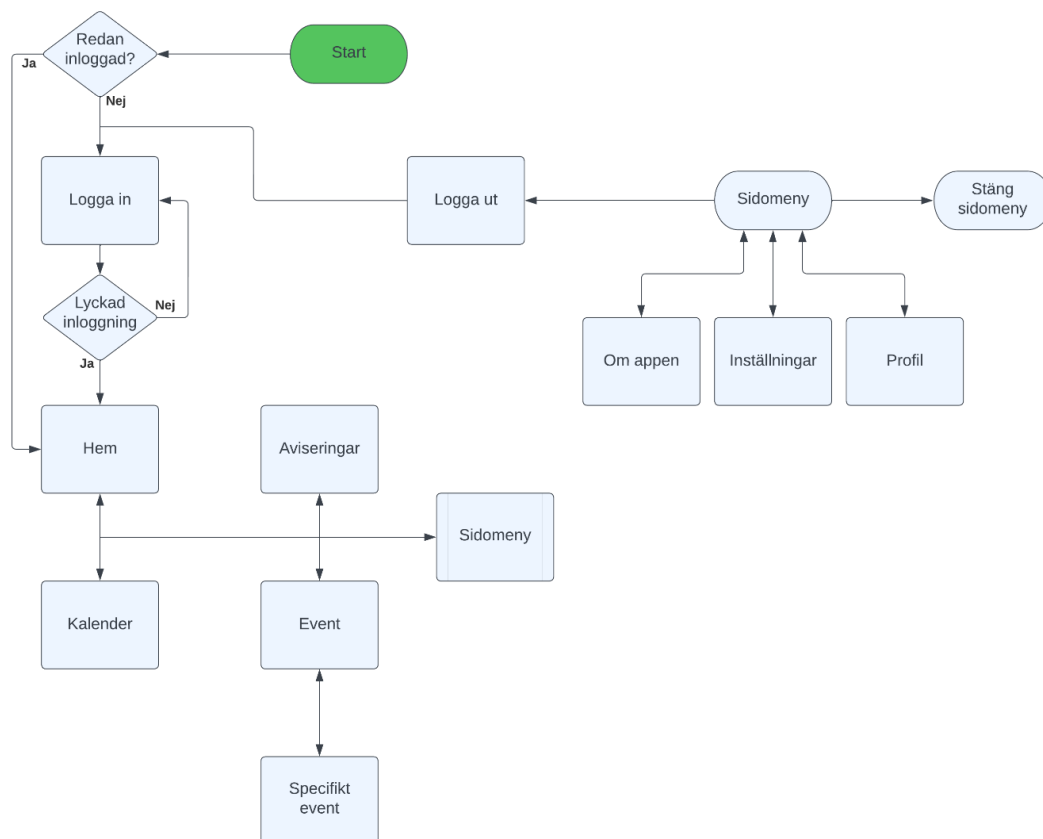
## 4. Utveckling

Utvecklingen började med framtagandet av olika exempelapplikationer med hjälp av Flutter för att bekanta sig med ramverket samt dess programmeringsspråk, Dart. Dessa applikationer fokuserade enbart på att implementera grundläggande funktioner så som knapptryck, inmatning från användare, hämtning av data från extern källa samt uppdatering av vad som visas på skärmen. Sedan påbörjades arbetet med vad som skulle bli slutprodukten och en ny applikation skapades. Arbetet med att implementera de önskade funktionerna började med att ta fram en fungerande inloggningsmetod. Applikationen har strukturerats efter MVVM-designmönstret (Model-View-ViewModel) då detta underlättar för företaget vid fortsatt utveckling efter det att arbetet överlämnats.

### 4.1 Navigeringsflöde

Ett flödesschema över hur det tänkta navigeringsflödet skulle vara togs fram för att användas som ett hjälpmedel under utvecklingsprocessen. Flödesschemat bidrog till en överblick över applikationens beteende redan innan någon kod skrivits. Flödesschemat visas i figur 1.

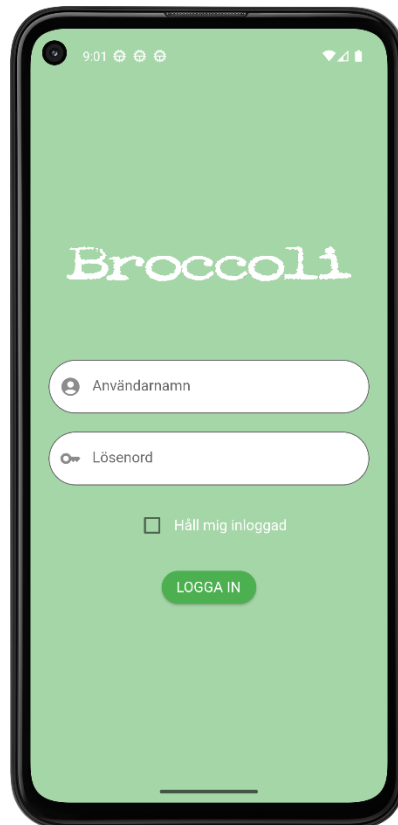
En regel vi utgick från är den så kallade ”Three Click rule” [7]. Detta fanns som vägledning, men var ingenting som följdes strikt, under utvecklingen för att hålla sig inom ramen för vad som anses vara enkel och snabb navigering.



Figur 1, flödesschema över navigationen

## 4.2 Inloggning

Första målet var att hämta data från företagets befintliga backend. För att uppnå det behövdes en inloggningsrutin som ger tillgång till övriga resurser. Första skärmen skapades med två fält för användarnamn respektive lösenord samt ett alternativ för att förbli inloggad vid lyckad inloggning, se figur 2. Eftersom applikationen ska spegla det existerande intranätet används befintliga inloggningsuppgifter. Här gjordes valet att inte implementera möjligheten för en användare att återställa sina uppgifter då detta inte var något som önskades.

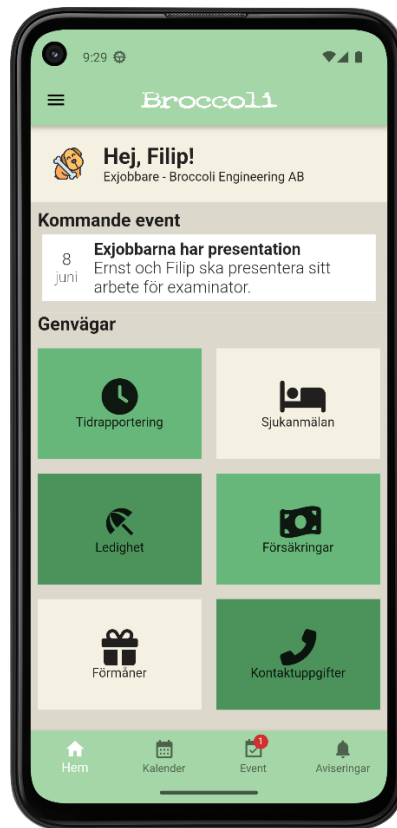


Figur 2, inloggningsskärm

När användaren trycker på "Logga in" skickas uppgifterna till företagets server som besvarar förfrågan. Skulle kombinationen av användarnamn och lösenord vara fel, eller inte existera, kommer en röd text visas under logotypen som påpekar detta och användaren förblir på inloggningsskärmen. Om inloggningen lyckas navigerar applikationen till en hemskärm som är startpunkten för en inloggad användare. Om användaren har markerat "håll mig inloggad" vid en lyckad inloggning kommer nödvändiga uppgifter som krävs för att få tillgång till företagets backend sparas i telefonens minne. Detta åstadkoms med hjälp av ett bibliotek vid namn Flutter Secure Storage, som är designat för att använda de inbyggda lösningarna för säker lagring på Android respektive iOS [8]. Vid en utloggning raderas dessa uppgifter från telefonen innan applikationen stängs ner. Kontinuerlig inloggning uppnås genom att applikationen söker i minnet efter dessa uppgifter, skulle de inte finnas kan användaren anses vara utloggad och inloggningsskärmen visas.

### 4.3 Hemskärm

Hemskärmen är den första skärm som visas efter start av applikationen när användaren är inloggad. Tanken var att denna skärm ska utgöra flera funktioner, se figur 3.



Figur 3, hemskärm

Överst finns ett applikationsfält med företagets logotyp centrerad. Denna återfinns på alla skärmar i inloggat läge förutom vid ett events egen skärm. Till vänster finns en ikon med tre horisontella linjer som utgör knappen för att öppna sidomenyn. Under applikationsfältet välkomnas den inloggade och deras profilbild visas. Underrubriken utgörs av befattningen på nuvarande uppdrag samt vilket företag konsulten är placerad på.

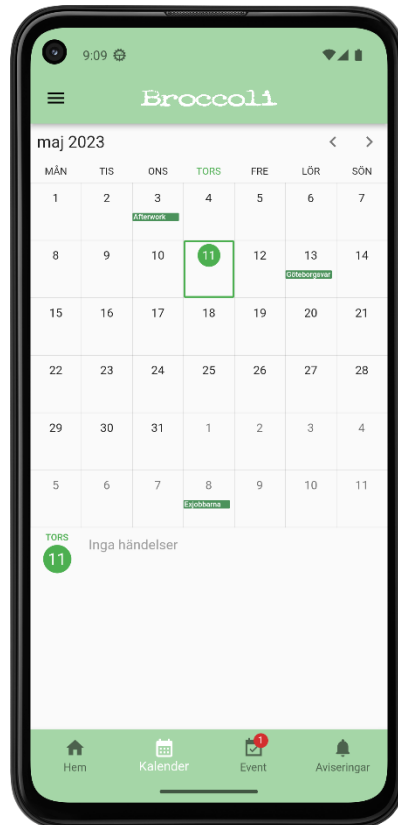
Under den personliga informationen finns snabb åtkomst till nästa event som användaren tackat ja till. Det går att trycka på det vita området för att komma till det eventets egen sida.

Majoriteten av hemskärmen består av olika genvägar till diverse funktioner som den anställda kan tänkas nyttja. Dessa är medvetet placerade på den nedre delen av skärmen då det faller inom området som är lättast att nå med en tumme när telefonen hålls med en hand [9]. Genvägsrutorna utgör för tillfället ingen funktion, utan är endast förberedda för senare implementation när Broccolis intranät är färdigutvecklat.

Längst ned finns navigationsmenyn som, likt applikationsfältet, visas på alla skärmar i inloggat läge med undantag för ett events egen skärm. I den finns ikoner som kan användas till att växla mellan de olika huvudskärmarna. Skulle det finnas event eller aviseringar som användaren inte har svarat på visas en röd cirkel med antalet obesvarade ovanpå skärmens ikon i navigationsmenyn.

#### 4.4 Kalenderskärm

Kalenderskärmen är tänkt att ge en mer överskådande vy över kommande, men även passerade, evenemang. Kalenderfunktionaliteten har utvecklats med hjälp av ett bibliotek vid namn Flutter Calendar [10] som erbjuder en widget där endast datakällan behöver anpassas. När användaren navigerar till sidan är dagens datum markerat som standard, och eventuella händelser för dagen visas under kalendern. Den markerade dagen indikeras i kalendern med en grön gräns. Användaren kan markera vilken dag de vill, och informationen för det datumet visas i stället under rutnätet. Se figur 4.



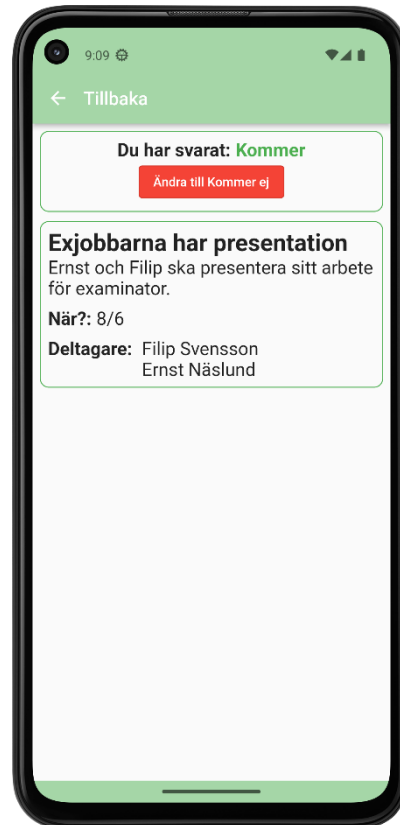
Figur 4, kalenderskärm

## 4.5 Eventskärmar

Vid navigering till "Event" presenteras olika event i en kolumn som är grupperad i tre olika fält, se figur 5. Överst visas de event som en användare inte lämnat något svarsbesked till ännu. Under det visas eventen som tackats ja till, och sist visas de event som tackats nej till. Efter önskemål från företaget går det att ändra svarsbesked fritt antal gånger. Eventskärmen visar eventets titel, två rader beskrivning samt datumet. För att få mer information och även ge svarsbesked kan en användare trycka på eventet för att komma till eventets egen skärm, se figur 6.



Figur 5, eventskärm

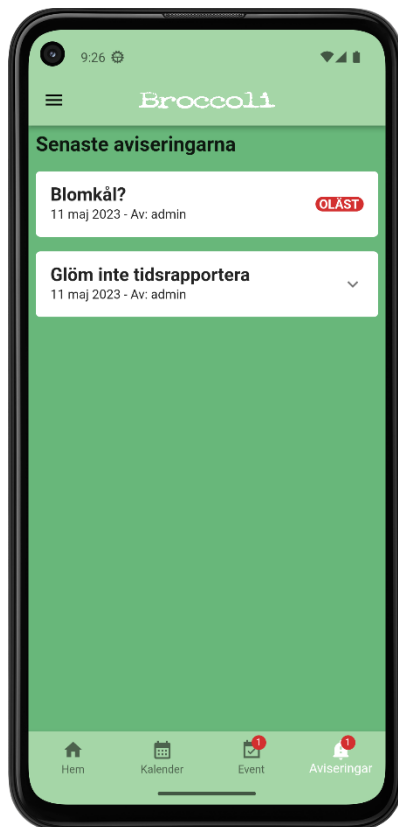


Figur 6, ett valt events egen skärm

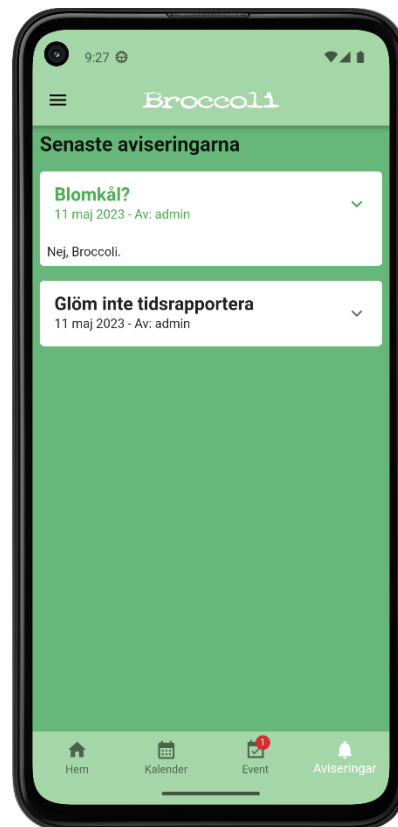
På ett events egen skärm visas nuvarande svarsbesked överst, med en knapp för att ändra till det andra alternativet. Är det obesvarats finns det två knappar för de båda alternativen, kommer eller kommer ej. Vidare listas titel, hela beskrivningen av eventet, datumet samt en lista på de användare som hittills tackat ja.

## 4.6 Aviseringskärm

För att möjliggöra broadcast-funktionaliteten implementerades en skärm som visar alla meddelanden som administratörer kan publicera på intranätet. Detta gör att en användare kan läsa en avisering på intranätet via webbläsaren och då även få den markerad som läst i applikationen. Layouten består av rutor som ordnas i en kolumn med datumordning där den senaste hamnar överst. En ruta visar titeln på aviseringen där underrubriken innehåller publiceringsdatum och skapare. Om aviseringen inte har blivit markerad som läst visas en röd bubbla med texten ”oläst” till höger i rutan. Här fick vi även delta i utvecklingen av intranätet då det behövdes logik i företagets backend för att markera ett meddelande som läst eller inte läst. För att markera en avisering som läst, eller för att läsa eventuell tillhörande beskrivning, går det att trycka på en ruta som då expanderar för att kunna visa hela innehållet, se figur 7.



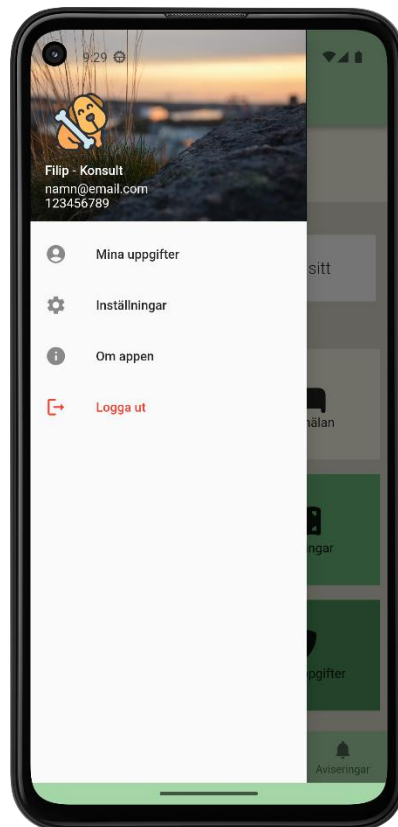
Figur 7, a) aviseringsskärm



b) med expanderad ruta

## 4.7 Sidomeny

Sidomenyn innehåller funktioner som ansågs inte vara lika frekvent använda som funktionerna i navigationsmenyn. Som tidigare nämnt kan användaren komma åt den genom att trycka på ikonen med tre streck uppe till vänster i applikationsfältet. När menyn är utfälld visas användarens profilbild överst i större format än på hemschärmen, se figur 8. Under bilden finns förnamn, den anställdes roll, arbetsmejladressen och arbetstelefonnumret. Därefter finns olika alternativ för att se sina profiluppgifter, komma åt inställningar, få information om applikationen samt möjligheten att logga ut.



Figur 8, sidomeny

## 5. Resultat

Arbetet resulterade i en prototypapplikation för iOS och Android som speglar Broccolis nuvarande stadiet i utvecklingen av deras intranät. Applikationen uppfyller de funktioner satta som mål i planeringsrapporten. Testning av applikationen har bara skett på Android-plattformen. Inloggning på applikationen kräver ett konto som skapas av företaget. Det går att välja att förbli inloggad även om applikationen stängs ner.

Data i applikationen om event och anställda på företaget hämtas från Broccolis databas och visas på de olika skärmarna. Varje gång en användare byter skärm uppdateras datan genom att hämtas på nytt. Designen med ikoner och färger är hämtade från företagets Figma-fil som de använder i sin utveckling. I inloggat läge går det att nå en godtycklig funktion inom tre tryck.

Det är möjligt att tacka ja eller nej till kommande event samt se datumet för eventet och andra anställda som har tackat ja till att delta. Kommande evenemang som man tackat ja till syns på huvudsidan av applikationen. Genvägar på huvudsidan är för tillfället inaktiva då funktionerna inte har implementerats på intranätet ännu.

Det finns en kalenderfunktion som ger en överblick över passerade samt kommande händelser.

Pushnotiser kan skickas ut till användare via FCM vilket sker manuellt genom Firebasekonsolen, där titel och text på notifikationen kan specificeras samt tid för utskick och om den skall upprepas dagligen eller veckovis.

## 6. Diskussion

Detta kapitel diskuterar först en utvärdering av arbetssättet och resultatet baserat på de uppsatta målen. Efter det diskuteras påverkan på samhället samt framtida utvecklingsmöjligheter.

### 6.1 Utvärdering av arbetssätt och resultat

Planeringen som gjordes visade sig vara bra bortsett från rapportskrivningen som påbörjades först i april och fortsatte till mitten av maj. Scrum-metoden visade sig passa arbetet bra då det ibland krävde omprioriteringar av funktionaliteten med hänsyn till Broccolis utveckling av sitt intranät. Eftersom det heller inte fanns en given design att följa har de regelbundna mötena bidragit till att designen har kunnat itereras fram baserat på feedbacken som getts.

Applikationen når upp till samtliga uppsatta mål. Då Broccoli utvecklat sitt intranät parallellt under arbetets gång har vissa element lagts till utan att fylla någon funktion; alternativen i sidomenyn samt genvägarna på hemskärmen. Dessa följer dock det tema som önskades och genvägarna använder sig av samma ikoner som de som är under utveckling.

### 6.2 Påverkan på samhället

Detta kapitel tar upp de samhällliga, etiska och ekologiska aspekterna för projektet.

#### 6.2.1 Samhällliga Aspekter

Genom att samla intranätet i en mobilapplikation gör man det enkelt för anställda att svara på kommande evenemang i stället för att arbetstid läggs på att skicka ut påminnelse mejl eller söka upp folk fysiskt för att få svar om de kan komma eller inte. Detta kan leda till en ökning i produktivitet på företaget då behovet av dessa arbetsuppgifter försvinner.

#### 6.2.2 Etiska Aspekter

Den etiska aspekt som behöver vara i åtanke är hanteringen av känslig information som adress, namn och telefonnummer vilket applikationen får direkt tillgång till genom kopplingen mot intranätet. Applikationen lagrar inga av dessa uppgifter på telefonen utan de hämtas regelbundet från intranätet och visas direkt på skärmen. Det enda som lagras på telefonens minne är de uppgifter som krävs för att verifiera att en användare som är kontinuerligt inloggad blivit autentiserad korrekt. Lagringen av dessa uppgifter använder sig av de olika operativsystemens inbyggda lösningar, vilket bör anses vara det säkraste alternativet för att förhindra att obehöriga kommer åt informationen.

#### 6.2.3 Ekologiska Aspekter

Det finns inga direkta ekologiska aspekter av arbetet. Alla anställda på Broccoli har redan tillgång till en mobiltelefon och inga ytterligare resurser kommer krävas för att använda applikationen.

## 6.3 Framtida Utvecklingsmöjligheter

Detta kapitel beskriver vad som kan vidareutvecklas inom applikationen

### 6.3.1 Vidareutveckling av notifikationer

Push notiser från applikationen kommer för tillfället från manuella utskick. Dessa kan göras regelbundna så att en specifik notifikation skickas till alla användare efter angivet intervall. Automatiserade notifikationer hade kunnat utnyttjas för att skicka påminnelser om event som man inte svarat på efter en viss tid. Det hade även varit möjligt att dela upp användare i olika grupper för olika syften så som frukost- och lunchgrupper där notiser kan gå ut till en specifik grupp. För att uppnå denna funktionalitet krävs det implementering på företagets backend.

### 6.3.2 Tidigare implementering av Firebase

Firebase var hjälpsamt för att lösa push-notifikationer som går ut till användare. Denna tjänst upptäcktes en bit in i utvecklingen och blev därför inte implementerad från början. Hade en tidigare implementering skett kunde det bidragit till mindre buggar med notifikationer som skickades men inte levererades till enheterna.

### 6.3.3 Testning på iOS hårdvara

Då simulering och testning av iOS kräver en dator från Apple kunde detta inte testas. Eftersom Flutter gör att applikationen ser likadan ut på olika system kan den antas vara visuellt samma på en iOS enhet. Däremot går det inte att garantera att biblioteket Flutter Secure Storage samt push notifikationer fungerar direkt då dessa funktioner använder sig av målsystemets hårdvara som kan se olika ut och därför kräva specifika konfigurationer.

### 6.3.4 Omkonstruktion med annat ramverk

Då arbetet inte analyserade de möjliga implementationsmetoderna på djupet bör en sådan analys övervägas vid en eventuell rekonstruktion. Olika ramverk har olika för- och nackdelar. Eftersom hela applikationen blir begränsad av möjligheterna ett ramverk erbjuder så bör noggrannare förstudier genomföras i en nästa iteration.

## **7. Slutsats**

Projektet har resulterat i en prototyp av en mobilapplikation för intranätet på Broccoli Engineering. Applikationen är lätt att navigera genom och utgör flera användbara funktioner för både administratörer och användare. Överblicken av event och deltagare underlättar arrangörers arbete med planering. Designen följer företagets färger, typsnitt och ikoner.

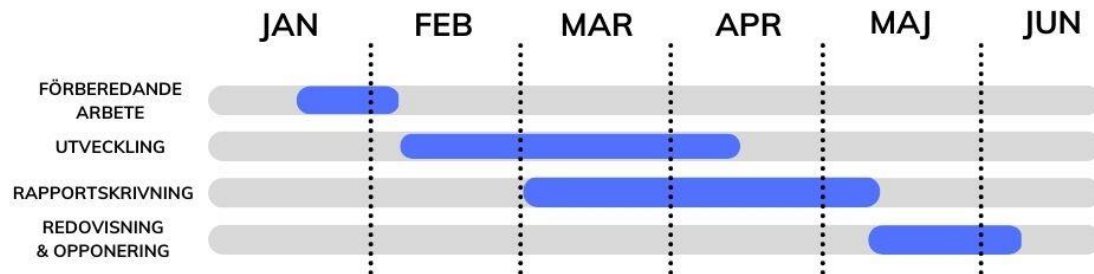
Applikationen är inte fullt redo att bli släppt till anställda i nuläget eftersom den behöver funktioner från intranätet som ännu inte existerar. Arbetet med att ta fram en mobilapplikation som fullständigt speglar intranätet har påbörjats med implementationen av flera funktioner och en stabil grund för vidareutveckling är skapad.

## Referenser

- [1] Flutter, "Flutter architectural overview", 2023. [Online] Tillgänglig: <https://docs.flutter.dev/resources/architectural-overview> (hämtad: 2023-05-10).
- [2] Android, "Meet Android Studio", 2023. [Online] Tillgänglig: <https://developer.android.com/studio/intro> (hämtad: 2023-05-10)
- [3] Firebase, "Developer Documentation for Firebase", 2023. [Online] Tillgänglig: <https://firebase.google.com/docs> (hämtad 2023-05-10)
- [4] Firebase, "Firebase Cloud Messaging", 2023. [Online] Tillgänglig: <https://firebase.google.com/docs/cloud-messaging> (hämtad: 2023-05-10)
- [5] Scrum, "What is Scrum?", 2023. [Online] Tillgänglig: <https://www.scrum.org/learning-series/what-is-scrum> (hämtad 2023-05-08)
- [6] Figma, "It's time to design on your own terms", 2023. [Online] Tillgänglig: [www.figma.com/design/](http://www.figma.com/design/) (hämtad 2023-05-08)
- [7] Tiller, "3-Click Rule", 2023. [Online] Tillgänglig: <https://tillerdigital.com/glossary/3-click-rule/> (hämtad 2023-05-15)
- [8] Flutter, "flutter\_secure\_storage", 2023. [Online] Tillgänglig: [https://pub.dev/packages/flutter\\_secure\\_storage](https://pub.dev/packages/flutter_secure_storage) (hämtad 2023-05-07)
- [9] Medium "Mobile App Design: Keeping the Thumb Zone in Mind", 2020. [Online] Tillgänglig: <https://philwijs.medium.com/mobile-app-design-keeping-the-thumb-zone-in-mind-c1d4fafcd127> (hämtad 2023-05-10)
- [10] Flutter, "syncfusion\_flutter\_calendar". 2023 [Online] Tillgänglig: [https://pub.dev/packages/syncfusion\\_flutter\\_calendar](https://pub.dev/packages/syncfusion_flutter_calendar) (hämtad 2023-06-11)

## Appendix A – Gantt-schema

### MOBILAPPLIKATION FÖR INTRANÄT OCH HR-MANAGEMENT GANTT SCHEMA



INSTITUTIONEN FÖR DATA- OCH  
INFORMATIONSTEKNIK  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2023  
[www.chalmers.se](http://www.chalmers.se)



UNIVERSITY OF  
GOTHENBURG



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY