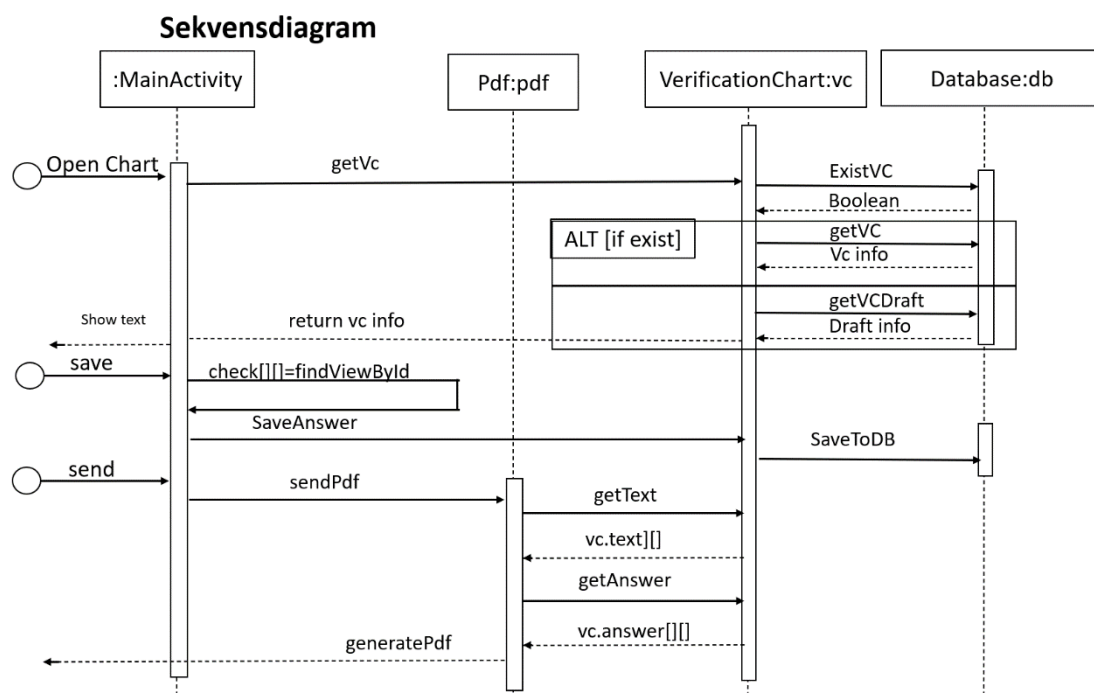


# Sekvens Diagram

Udarbejdelsen af sekvensdiagrammet har til formål at give en oversigt over en proces i programmet. Sekvensdiagrammet inkluderer flere objekter, og de viser også hvor vi er i sekvensen tidsmæssigt. Metoderne i tre klasser i programmet er begyndt at blive uoverskuelige, og derfor laver jeg nu et sekvensdiagram som jeg vil kunne arbejde ud fra. Jeg vil gerne have at diagrammet indeholder sekvensen for at der hentes tekst, at svar gemmes og at der genereres en pdf.



Klassen VerificationChart holder styr på data som har at gøre med tjeklisten. Som start i vores proces gemmes der i klassen VerificationChart, men senere vil VerificationChart få sine data fra en database. Databasen lukkes gentagne gange for at sikre at der ikke opstår konflikter når andre klasser laver forbindelse til samme database. Er man sikker på at der ikke vil opstå konflikt, vil databaseforbindelsen kunne skabes én gang og genbruges, her er den sikre løsning valgt.

Ved opstart af en tjekliste bliver der genereret et nyt objekt vc af VerificationChart. Fra vc hentes oplysninger i form af tekst som bruges til at generere tekster i det grafiske userinterface, der hentes også svar samt info om billeder hvis tjeklisten allerede eksisterer i databasen. Vc tager sig af at tjekke om tjeklisten allerede eksisterer i databasen, og er på diagrammet indrammet med en boks markeret med ALT for alternativ, hvis tjeklisten ikke eksisterer i databasen, vælges alternativet som er at generere en standart formular til indtastning. Standart formularen er endnu ikke gemt i databasen.

Når metoden save kaldes gemmes der information i det todimensionelle array check som allerede er defineret i klassen MainActivity. Informationen hentes fra det grafiske user interface, og MainActivity fungerer således som en controller for user interfacet. Efter at informationen er gemt i check kopieres den over i vc som allerede er oprettet og gemmes her i det todimensionelle array answer. Man kunne have ladet vc selv gemme den information direkte, men for at give rum for fremtidige ændringer holdes

kommunikationen med det grafiske userinterface og MainActivity adskilt fra vc. Skifter man til at andet grafisk userinterface som kræver oversættelse eller tilretning af informationen, vil det være naturligt at ændre det her i MainActivity inden det gemmes i vc.

Når funktionen send kaldes, oprettes et objekt af klassen Pdf og navngives pdf. I pdf kaldes metoden sendPdf. I første omgang var sekvensen planlagt til at hente information i databasen, som vist på det lille billede. Med det er der ikke nogen grund til når vc allerede er opdateret først i sekvensen og igen med save funktionen. Nu kan vc informationen kan hentes direkte. Var pdf ikke aktiveret i denne sekvens kunne det være nødvendigt at hente informationen fra databasen. generatePdf går ikke til MainActivity fordi pdf gemmes direkte på enheden. Senere tilføjes der måske en besked til MainActivity om at pdf'en er genereret, men for nu undlades den besked.

