En kode som skal lage en gui for en todolist ble skrevet i python. En modell av den objektorienterte koden presentert i oppgaven ble seende slik ut:

Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse

Fra klassediagrammet kan vi se flere eksempler på ting som kunne ha blitt implementert bedre, og hvordan klassen kunne støttet videre utvikling av funksjonalitet.

Kommentar til metoder/struktur:

Det første jeg må kommentere på er at brukergrensesnittet blir laget i \_\_init\_\_ metoden til To-Do-List appen. Dette er ikke bra fordi init ikke skal blokke for andre programmer som prøver å kjøre. Denne metoden burde heller åpnes til applikasjonen gjennom en public run metode eller noe lignende. Selve «create\_gui» metoden burde ikke være offentlig.

Kommentar til medlemsvariabler:

En annen åpenbar ting er at medlemsvariabelen «root» i den øverste klassen ikke burde være public, siden det ikke er hensiktsmessig at andre klasser/kode skal kunne endre på «root» direkte. Det samme gjelder for objektet «toDoList» som man heller kan gi gjennom en «getter».

Kommentar til klasser/struktur

Når det gjelder struktur er koden generelt sett robust til å kunne implementere filtere for henting av tasks og videre funksjonalitet. Det er hensiktsmessig å skille gui-applikasjonen fra selve To-Do-listen, slik som er gjort her. Vi ser også at innhenting av tasks skjer gjennom getter-metoden «get\_tasks» («Tasks» i den originale koden, men dette burde ALDRI en metode hete). Dette tillater at vi kan gi forskjellige filtere til metoden for å hente ut tasks basert på parametere. Eksempler på parametere som To-Do-List kan sortere/filtere etter er bruker id og status til oppgaven.