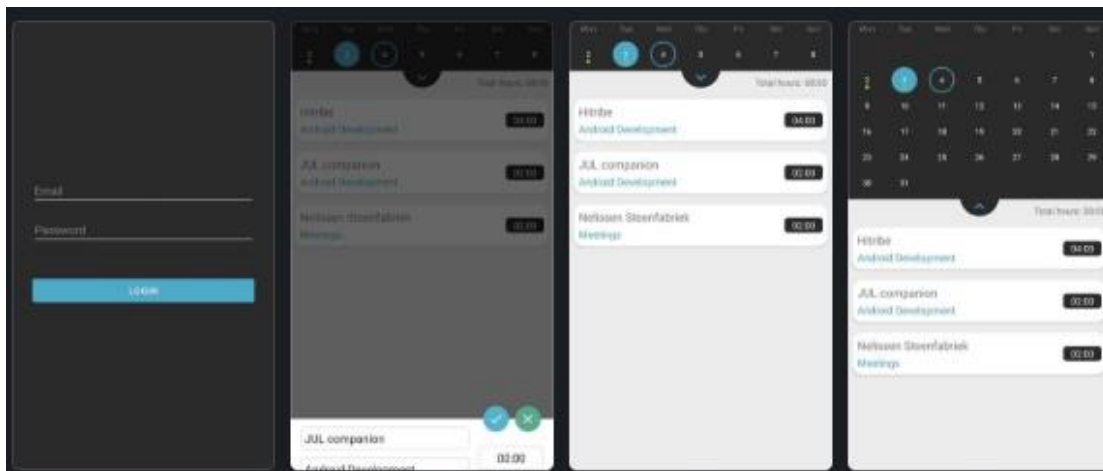


## Realisatie stageopdracht 1: Overtime applicatie

In het eerste deel van mijn stage heb ik een bestaande app van Appwise opnieuw moeten maken met de nieuwste technologieën die nu gebruikt worden binnen Appwise. De Overtime app wordt binnen het bedrijf gebruikt om per dag de uren die gespendeerd worden aan bepaalde projecten bij te houden. Daarbij hoort dan ook een taak die uitgevoerd is voor het project en daarnaast nog de mogelijkheid om notities toe te voegen.

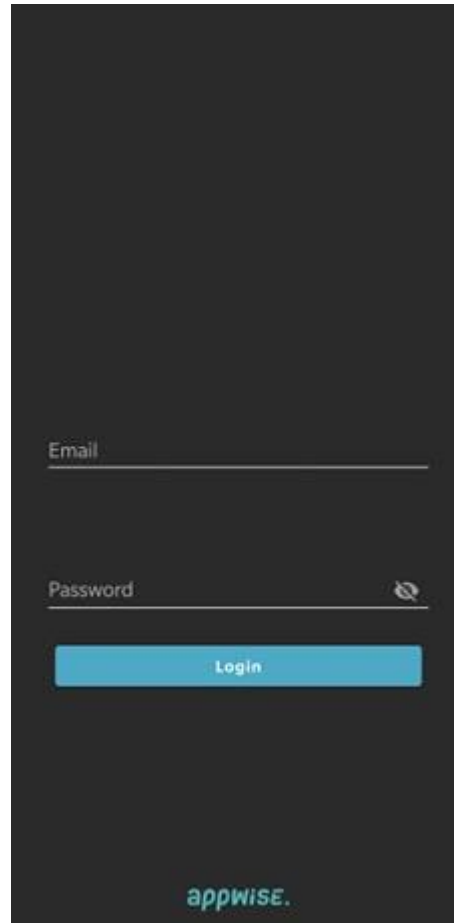
Het doel van deze opdracht was om mezelf kennis te laten maken met de werkwijze die gehanteerd werd binnen het Android-team van Appwise. Daarbij hoorden natuurlijk ook verschillende nieuwe technologieën waarmee ik eerder nog niet had gewerkt. Zo waren er bijvoorbeeld verschillende *features* van Kotlin (waaronder de coroutines) en ook het *DBMS Realm*, dat gebruikt wordt om lokaal belangrijke data bij te houden op het Android apparaat. Ik heb de broncode van een app (JUL Companion) gekregen die nog in development was om mij een voorbeeld te geven van hoe ze die verschillende technologieën toepassen. Daarnaast heb ik ook screenshots gekregen van hoe de huidige app eruitziet zodat ik mijn front-end daarop kon baseren.



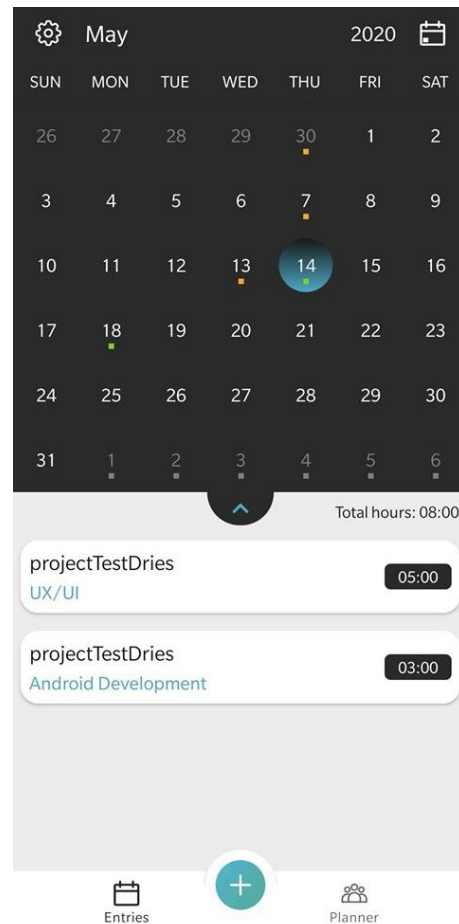
Nadat ik mijn tijd had genomen om de voorziene documentatie te bestuderen, heb ik eerst de nodige code voor de database klassen geschreven omdat ik al wist welke klassen er gingen nodig zijn voor de applicatie.



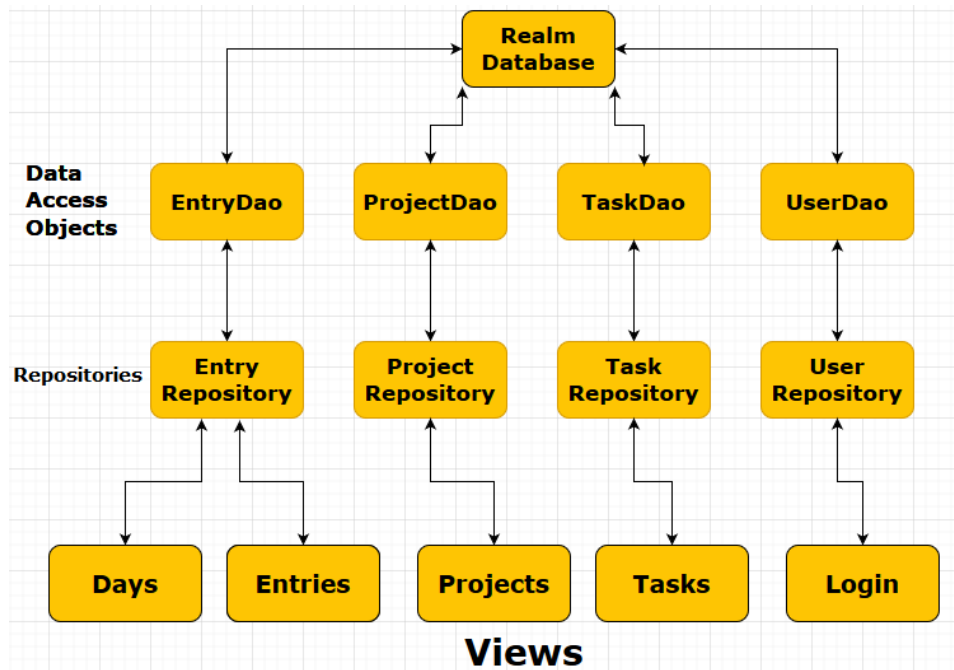
Daarna ben ik begonnen met de Overtime app feature per feature uit te werken. Ik maakte voor elke feature eerst het nodige scherm en begon daarna met de achterliggende code. Een voorbeeld hiervan is het login scherm.



Op vraag van Jesse, mijn stagementor, heb ik een nieuwe *library* gezocht om de kalender van de applicatie te verwezenlijken. Hiervoor werd eerder de *Material library* gebruikt maar deze bleek te zwaar te zijn. Na een korte zoektocht kwam ik terecht op de *CalendarView library* van “kizitonwose” op GitHub. Het is een uitgedunde *library* die zich enkel focust op de belangrijke aspecten die een kalender nodig heeft. Omdat de nieuwe *library* nog niet was gebruikt op de manier die ik nodig had, werd het een leuke uitdaging om dit zelf uit te zoeken en te implementeren. Hierdoor heb ik veel bijgeleerd over het gebruik van Kotlin en heb ik ervoor kunnen zorgen dat de applicatie minder opslag- en werkgeheugen inneemt op het apparaat van de gebruiker.



De volgende uitdaging was *Realm*. Ik had al veel bijgeleerd over *Realm* toen ik de JUL app aan het bekijken was. Er was veel kennis overdraagbaar maar het andere deel heb ik mezelf aangeleerd. Ik heb gebruik gemaakt van de basis klassen die in elke nieuwe applicatie worden gebruikt waarop ik dan zelf kon verder bouwen. *Realm* is gebaseerd op *NoSQL* en heeft al veel ingebouwde functies waarvan ik gebruik kon maken. Dit was handig toen ik bezig was met de *Data Access Objects* en de *repositories* om de gebruiker via de schermen wijzigingen te laten maken aan de lokale database.



Nadat ik de applicatie functioneel helemaal in orde had gebracht, ben ik ermee naar mijn stagementor gegaan om zijn feedback te vragen via een *pull request* op Github. Op die manier kon Jesse stukken van mijn code selecteren om daar notities bij te zetten, zo kon ik dan ook gemakkelijk zien waar ik fout zat om dit zo aan te passen. Nadat ik de nodige aanpassingen had gedaan, heb ik de app helemaal getest en nog enkele bugs opgelost. Waarna de app helemaal klaar was en ik klaar was voor mijn volgende opdracht.

Ik heb in totaal 6 weken nodig gehad om de Overtime app te ontwikkelen. Door deze app opnieuw te maken, heb ik veel bijgeleerd over de nieuwe *features* van Kotlin, waarmee ik eerder nog niet had gewerkt, en ook over *Realm* en hoe een *NoSQL* systeem werkt voor Android apparaten.