

**I4SWT-02 Software test**

**Handin 2**

**Gruppe: SWT E2018 9**

**ATM Del 1**

Christoffer Kjellerup Jakobsen - 201610457 (au569759)

Simon Møller Hindkær - 201609970 (au570382)

Marius Møller - 201610460 (au566306)

Tobias Damm Henrichsen - 201611660 (au553636)

Link til Jenkins build job: <http://ci3.ase.au.dk:8080/job/TeamSWT09AirTrafficMonitorBuildCoverage/>

Link til GitHub repository: <https://github.com/ChristofferKJ/I4SWT/tree/master/ATM>

Indhold

[Indledning 2](#_Toc526422961)

[Designet 2](#_Toc526422962)

[Klassediagram 2](#_Toc526422963)

[Sekvensdiagram 3](#_Toc526422964)

[Opdeling af implementering af klasser og test 3](#_Toc526422965)

[Jenkins og GitHub (Continuos integration) 3](#_Toc526422966)

[GitHub 3](#_Toc526422967)

[Jenkins 4](#_Toc526422968)

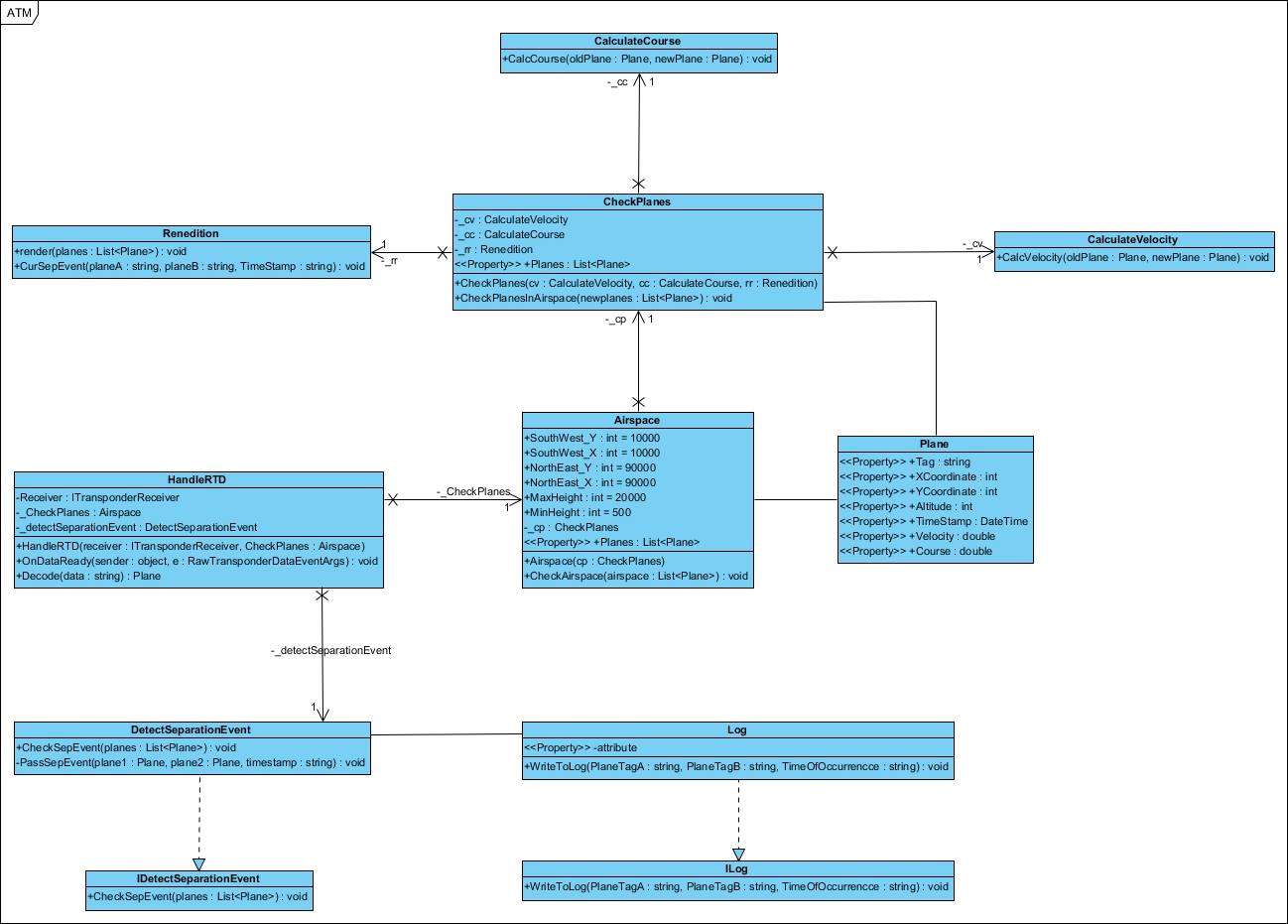
Air Traffic Monitor

# Indledning

I forbindelse med Handin 2 i I4SWT, er der blevet udarbejdet en journal, samt en implementation, med tests, af en Air Traffic Monitor, som skulle overvåge og decode en række genererede flydata fra en ”Transponder Reciever”. Til opgaven er der blevet anvendt continuos integration, ved hjælp fra værktøjerne GitHub og Jenkins (Links på forsiden).

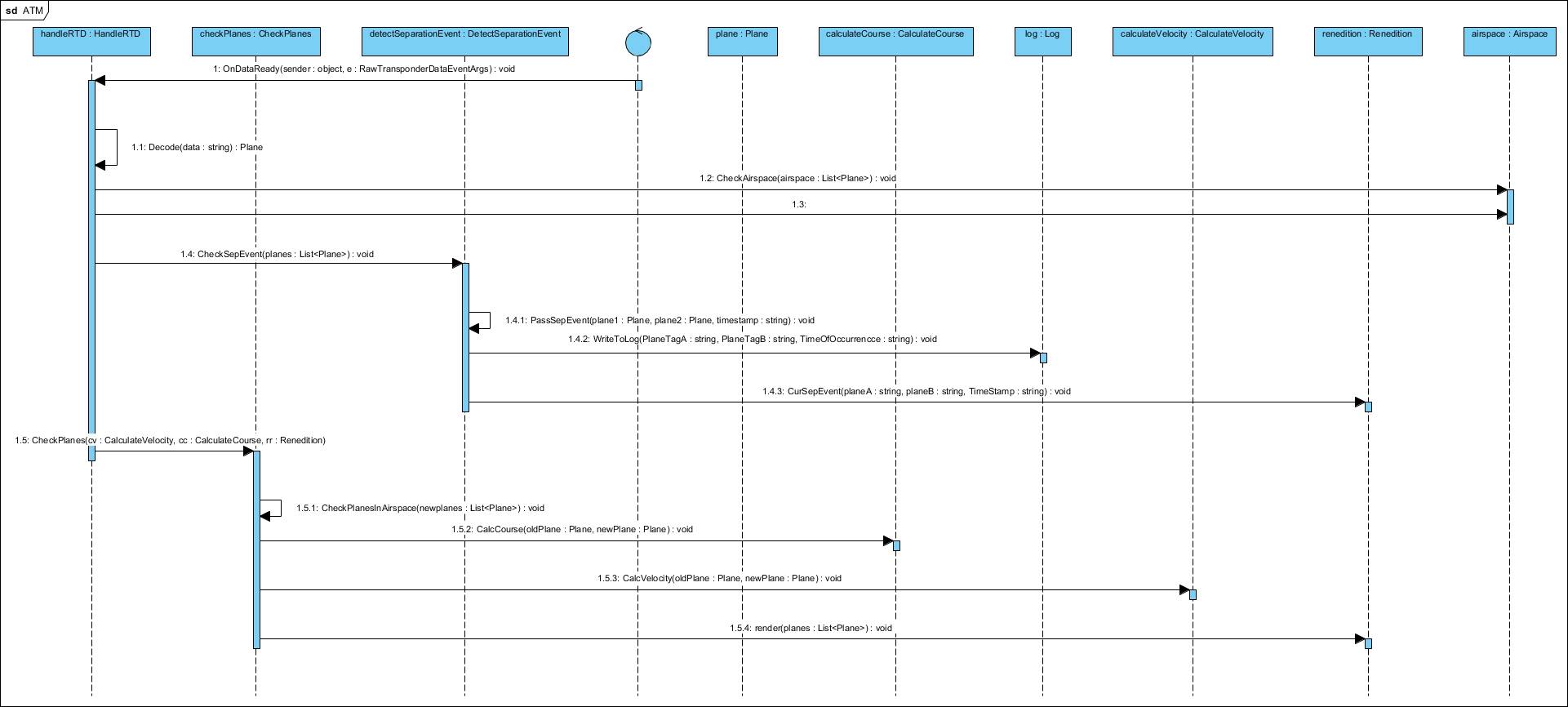
# Designet

## Klassediagram



Figur 1 - Klassediagram for ATM (AirTrafficMonitor)

## Sekvensdiagram



Figur 2 - Sekvensdiagram for ATM hvor vi modtager Transponderdata

# Opdeling af implementering af klasser og test

I starten læste hele gruppen opgaven igennem og udarbejdede et klassediagram sammen. Herefter blev klasserne fordelt ud på gruppens medlemmer, og implementeringen gik i gang. Undervejs blev klasserne oprettet, rettet til og til sidst færdigudarbejdet af de medlemmer der var sat på de forskellige klasser. Som udgangspunkt, var det en person der stod for én eller flere bestemte klasser, så der ikke blev arbejdet i de samme filer, af flere forskellige medlemmer.

Ved implementeringen af vores tests, blev der nu byttet rundt, så vi lavede tests for hinandens klasser. På den måde, blev man nødt til at tænke ud af boksen når testene skulle implementeres.

# Jenkins og GitHub (Continuos integration)

## GitHub

Overordnet set, har GitHub hjulpet os positivt, idet vi er en gruppe på 4 mennesker, der typisk arbejder i den samme solution, på samme tid. Ved at uddele de forskellige klasser, har vi altså kunne arbejde på det samme projekt på samme tid, hvilket jo er en kæmpe fordel når man laver gruppearbejde. Idet vi har været gode til at ”pushe” vores filer op på vores repository, har det givet alle gruppens medlemmer et godt overblik over fremskridtet hos de andre personer, hvilket både er med til at indbyde til kritik og ros på det skrevne kode. Der opstod dog et problem, hvor to personer arbejdede på den samme fil, hvilket resulterede i at vi tabte noget kode. Her fulgte et større tidsspild, på at få genfundet den tabte kode, idet vi stadig er forholdsvist nye på GitHub.

## Jenkins

Brugen af Jenkins, har i denne opgave hjulpet os med at få styr på om kvaliteten af de tests vi har lavet, har været tilstrækkelig. Vi har forsøgt at få en coverage på 100%, altså hvor 100% af vores kode bliver testet, og med netop dette har Jenkins været en kæmpe hjælp til at opnå et resultat så tæt på 100% som muligt.

Alt i alt er CI noget vi mener burde bruges til ethvert gruppeprojekt, og der er rigtig mange positive aspekter, ved at anvende det. Dog har den største ulempe, været den anvendte tid der er blevet brugt på at fixe mindre problemer.