

Acronym

Acamo

Project

Active Aircraft Monitor

Doctype

Requirements

Author

Kai Warendorf

Contact

Kai.Warendorf@hs-esslingen.de

Client

Esslingen University

Contact

Faculty of Information Technology

Version

2.0

Date

9 Sept. 2017

Contents

1	Project Drivers	3
1.1	Purpose of the Project.....	3
1.1.1	Vision Statement.....	3
1.1.2	Project Outcomes	3
1.1.3	Learning Objectives	3
1.2	Stakeholders	4
1.2.1	Project Team.....	4
1.2.2	Product Users	4
2	Functional Requirements	5
	Acamo.F.10 Select Active Aircraft	5
3	Non-Functional Requirements	6
3.1	Look and Feel Requirements	6
	Acamo.NF.10 Graphical User Interface (GUI)	6
3.2	Performance Requirements	7
	Acamo.NF.20 Timing	7
3.3	Implementation-Specific Requirements	7
	Acamo.NF.50 Test Driven Development.....	7
3.4	Maintainability Requirements	7
	Acamo.NF.70 Documentation.....	7
	Acamo.NF.80 Cohesion and Coupling.....	7
	Acamo.NF.90 OO Design Principles.....	7

Chapter 1

Project Drivers

1.1 Purpose of the Project

1.1.1 Vision Statement

This project aims at developing an application that shows the active aircraft in range of the ADS-B receiver.

1.1.2 Project Outcomes

The Java application reads aircraft messages.

The Java application transforms aircraft message data into aircraft data.

The Java application displays aircraft data.

1.1.3 Learning Objectives

After having completed this project, as student, you can ...

- develop and integrate Java classes.
- identify and solve domain problems through advanced Java programming.
- develop rudimentary graphical user interfaces with Java.

1.2 Stakeholders

1.2.1 Project Team

Various members and roles.

1.2.2 Product Users

Local Flight Control Engineer, User. Priority: **Key User.**

Chapter 2

Functional Requirements

2.1 Acamo Functional Requirements

Acamo.F.10 Select Active Aircraft

essential

Feature In order to get an overview of the local flight traffic, as a flight control engineer, I want to be able to observe the aircraft that are currently active.

Scenario

Given the application shows aircrafts in range

When I select to observe an active aircraft

Then the application should show the active aircraft

Active aircraft send any arbitrary new ADS-B message within a time span of 4 minutes.

Feature Each active aircraft shall be shown with the following information:

- ICAO of the aircraft
- Timestamp of the last activity
- Airline of the aircraft if available
- Most recent 3-dimensional position with latitude, longitude and altitude
- Most recent velocity with horizontal and heading

Feature The application shall show the active aircraft upon application startup.

Scenario

Given the application is off

When I start the application

Then the application should show the aircrafts in range

Chapter 3

Non-Functional Requirements

3.1 Look and Feel Requirements

Acamo.NF.10 Graphical User Interface (GUI)

essential

Feature The application user interface shall be realized as graphical user interface.

Feature The GUI window shall be organized in terms of a list of the aircrafts in range and a pane for the selected aircraft.

Active Aircrafts						Selected	...	Aircraft:
icao	operator	posTime	coordin...	speed	tr	icao		4008e6
4008e6	BAW3KG	Sun Jun ...	48,28 / 9...	238.59	12...	operator		BAW3KG
4076cd	BAW856K	Sun Jun ...	48,87 / 8...	238.71	12...	posTime		Sun Jun 16 10:53:10 CE...
3c667a	DLH1162	Sun Jun ...	49,11 / 8...	224.61	19...	coordinate		48,28 / 9,59
4b1611	SWR1325	Sun Jun ...	48,59 / 9...	209.48	21...	speed		238.59
4070ea	EXS89RC	Sun Jun ...	48,63 / 8...	232.32	12...	trak		124.55
440057	EJU57FM	Sun Jun ...	48,80 / 8...	226.69	31...			
3d67ca	DFLOC	Sun Jun ...	49,23 / 9...	99.99	29...			
4b16b9	SWR169A	Sun Jun ...	49,14 / 8...	227.26	19...			
40624f	BAW644X	Sun Jun ...	48,08 / 1...	237.61	12...			
44034b	EJU98YZ	Sun Jun ...	49,36 / 8...	221.24	18...			
3c9432	AXG1507	Sun Jun ...	48,49 / 8...	251.28	14...			
ab7ae4	DAL117	Sun Jun ...	49,00 / 9...	186.8	35...			

3.2 Performance Requirements

Acamo.NF.20 Timing

essential

Feature The list of active aircraft shall be updated at least once per second.

3.3 Implementation-Specific Requirements

Acamo.NF.50 Test Driven Development

essential

In order to ascertain sufficient testing of the product, the implementation must be carried out following a test-driven development approach.

3.4 Maintainability Requirements

Acamo.NF.70 Documentation

essential

In order to ascertain high understandability, the source code must be self-explanatory.

Acamo.NF.80 Cohesion and Coupling

essential

In order to support high maintainability, the modules of the system must be realized with high-cohesion and low coupling.

Acamo.NF.90 OO Design Principles

essential

In order to support high maintainability, the other well-known principles of good object-oriented design must also be applied.

4.1 How to start

Try to solve the lab in the following order:

1. Write `ActiveAircrafts.java`:
 - a. Implement `ActiveAircraftsInterface.java`
 - b. Use a generic `HashMap` with the correct types for `K`, `V`
 - c. Complete constructor and methods
 - d. Note: there is a new test class for testing the hashmap
2. Complete `BasicAircraft.java` in Messer:
 - a. Complete static get methods
 - b. Use fields API or hard code it
3. Write `Acamo.java` (`start()`)
 - a. Add `Acamo` and `ActiveAircrafts` to Messer's observer list
 - b. Fill the column header of the table using the attribute list from `BasicAircraft`
 - c. Create your GUI layout using panes (e.g. `VBox`, `HBox`, ...), labels or text for the selected plane etc.
 - d. Add the event handler (for table) to find out which line was selected
 - e. Add aircraft list to table
4. In `Acamo.java` (`update()`)
 - a. Add all `activeAircrafts` to the aircraft list.
 - b. Solve the exception problem
5. In `Acamo.java`

When an aircraft is selected, display its data in the selection panel