POLITECNICO
MILANO 1863

Department of Computer Science and Engineering

Design Document (DD)

TrackMe

- v1.0 -

Authors:

Avila, Diego	903988
Schiatti, Laura	904738
Virdi, Sukhpreet	904204

December 10th , 2018

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

TrackMe develops health-monitoring devices devoted to measure and record different parameters related to the health status of a person (i.e. body temperature, blood pressure, heart pulse rate and percentage of O2 in the blood) and also their location. TrackMe health smartwatch is synchronized with an app that gives users access to their data and stats. This document discuss more technical aspects of working of the TrackMe application.

1.2 Purpose

The purpose of this document is to give more technical details than the RASD about TrackMe application system. While the RASD presented a general view of the system and what functions the system is supposed to execute, this document aims to present the implementation of the system including components, run-time processes, deployment and algorithm design. It also presents in more details the implementation and integration plan, as well as the testing plan.

More precisely, the document presents:

- Overview of the high level architecture
- The main components and their interfaces provided one for another
- The Runtime behaviour
- The design patterns
- The algorithm design of the most critical parts of the application
- Implementation plan
- Integration plan
- Testing Plan

The purpose of this document is to provide an overall guidance to the architecture of the software product.

1.3 Scope

The TrackMe environment will be composed by three systems, whose scope are defined in this section.

Data4Help is a system capable to store the physical health data of any individual using any TrackMe wearable device. All the stored data will be available to any third party company, after request and approval of its use is sent to the TrackMe wearable device user, who will be able to accept or reject the request, under no condition. Also, third party companies will be able to request data of a group of users, and Data4Help will provide that information under the solely condition of be able to anonymize it.

In addition, **AutomatedSOS** is a system that will be built on top of Data4Help, and will send requests to every elderly individual in order to access its physical health parameters. It will monitor the physical health status of all of its users in order to establish their health condition, based on previously defined thresholds. If any of the user's health parameters is under or above its threshold, the system will contact the associated health-care service for that user. AutomatedSOS will do its best effort to contact the health-care service, but under no circumstances will follow-up the status of the health-care response, meaning that it will not know if the health-care service answered the notification or not.

Finally, **Track4Run**, also a system built on top of Data4Help, will send requests to every user in order to access their location and let them participate in a defined running circuit. Track4Run organizer users, will be able to setup a race, define its running circuit and send invitations to all the TrackMe wearable device users. Also, the race spectators will be able to follow the participants' location using the Track4Run web site. The location of every participant will only be available during the race.

1.4 Definitions, Acronyms, Abbreviations

1.4.1 Definitions

- **Data trading**: Generate revenue from user data in a much more direct way, by selling user data to a third party.

1.4.2 Acronyms

- DD: Design Document
- D4H: Data4Help
- ASOS: AutomatedSOS
- T4R: Track4Run
- GUI: Graphical User Interface
- MVC: Model View Controller is a design pattern used for GUIs
- RASD: Requirements Analysis and Specifications Document

1.4.3 Abbreviations

- $[Gn]$: n-goal.

1.5 Revision history

It is important to keep track of the revisions made to this document:

Version	Last modified date
1.0	10^{th} December, 2018

Table 1.1: Revision history timeline

1.6 Document structure

This document is divided in seven parts, each one devoted to approach each one of the steps required to apply requirements engineering techniques.

- Chapter 1 gives an introduction of the design document. It contains the purpose and the scope of the document, as well as some abbreviation in order to provide a better understanding of the document to the reader.
- Chapter 2 deals with the architectural design of the application. It gives an overview of the architecture and it also contains the most relevant architecture views: component view, class view, deployment view, runtime view and it shows

the interaction of the component interfaces. Some of the used architectural designs and designs patterns are also presented here, with an explanation of each one of them and the purpose of their usage.

- Chapter 3 refers to the mock-ups already presented in the RASD document.
- Chapter 4 explains how the requirements that have been defined in the RASD map to the design elements that are defined in this document.
- Chapter 5 presents the algorithm design. It includes the most critical parts of the application and the algorithms designed to deal with them. All of the algorithms are written in pseudo code in order not to make any restriction on the implementation language and stay focused on the most important parts.
- Chapter 6 shows the effort spent by each group member while working on this project.
- Chapter 7 includes the reference documents.

Chapter 2

Architectural design

High-level components and their interaction

2.1 Overview

2.2 Component view

2.2.1 Data4Help component diagram

In the Figure 2.1 it can be seen the component diagram of D4H, with all its external interfaces. It can be noticed two main components: **DataBase** and **D4HBackend**, where the former one refers to the TrackMe database, and which provides an interface used by the **DBManager** component.

On the other hand, **D4HBackend** component, contains all the components related to D4H, which are needed to provide the interfaces used by the third parties and the web site. The following interfaces are used by the web site: **SignupWeb**, **LoginWeb**, **SearchWeb**, **RequestWeb** and **SubscriptionWeb**, while the **RequestAPI** interface is used by the third parties. In this case, D4H provides the interface in order to let the third parties send requests for accessing the health status and location of the individuals to them.

Furthermore, **AuthenticatorService** component has the responsibility of validate the different credentials, and to provide the secret codes to the third parties, to do so, it interacts with the **TokenDB** component using the **TokenConnector**

Finally, it can be seen the different third parties related to D4H. It worth mentioning **Track4Run** and **AutomatedSOS** components which, even though they are part of the TrackMe environment, they are treated as third parties in the sense they are completely decoupled of D4H. Moreover, all third parties must provide an interface to D4H in order to let it communicate the incoming changes of the subscriptions.

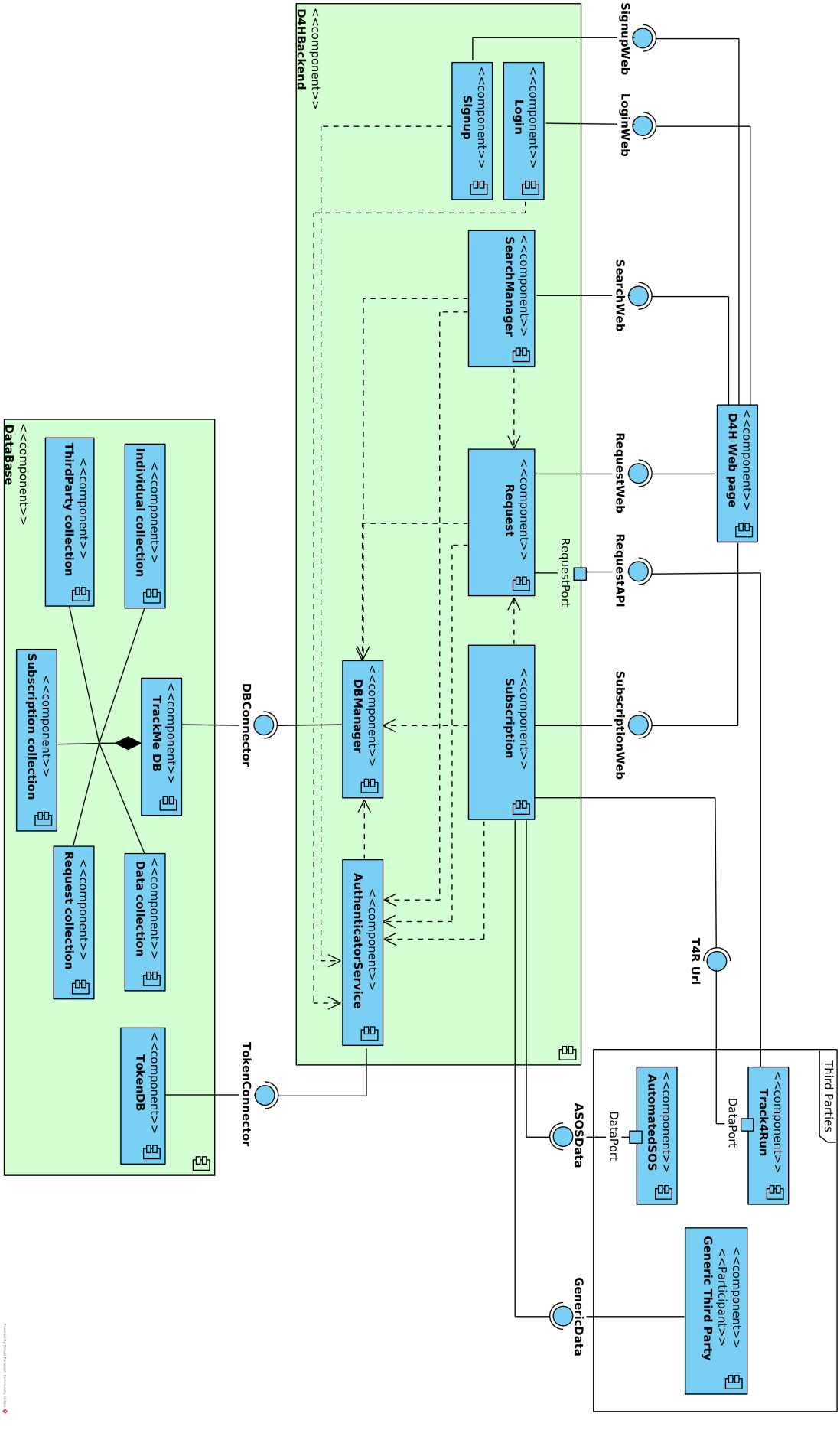

Figure 2.1: Data4Help Component Diagram

2.2.2 Track4Run component diagram

In the Figure 2.2 it can be seen the T4R components. Unlike the D4H component diagram, here the *DataBase* component is part of the *T4RBackend* component, this is so because T4R owns it in the sense that it has full responsibility on it. Besides this detail, the database provides a connector in order to let the **DBManager** component communicate to it.

The main structure of the system is similar to the structure seen in D4H component diagram. The web site communicates with the back-end using the following interfaces: **LoginWeb**, **SignupWeb**, **UserWeb** and **NotificationWeb**. On the other hand, T4R provides the **DataAPI** interface, which is responsibly of receiving all the data of the participants, and uses the **RequestAPI** interface, in order to send the requests for accessing the individuals' location and health status.

Finally, as in D4H, the **AuthenticatorService** component is responsible of validate the users' credentials.

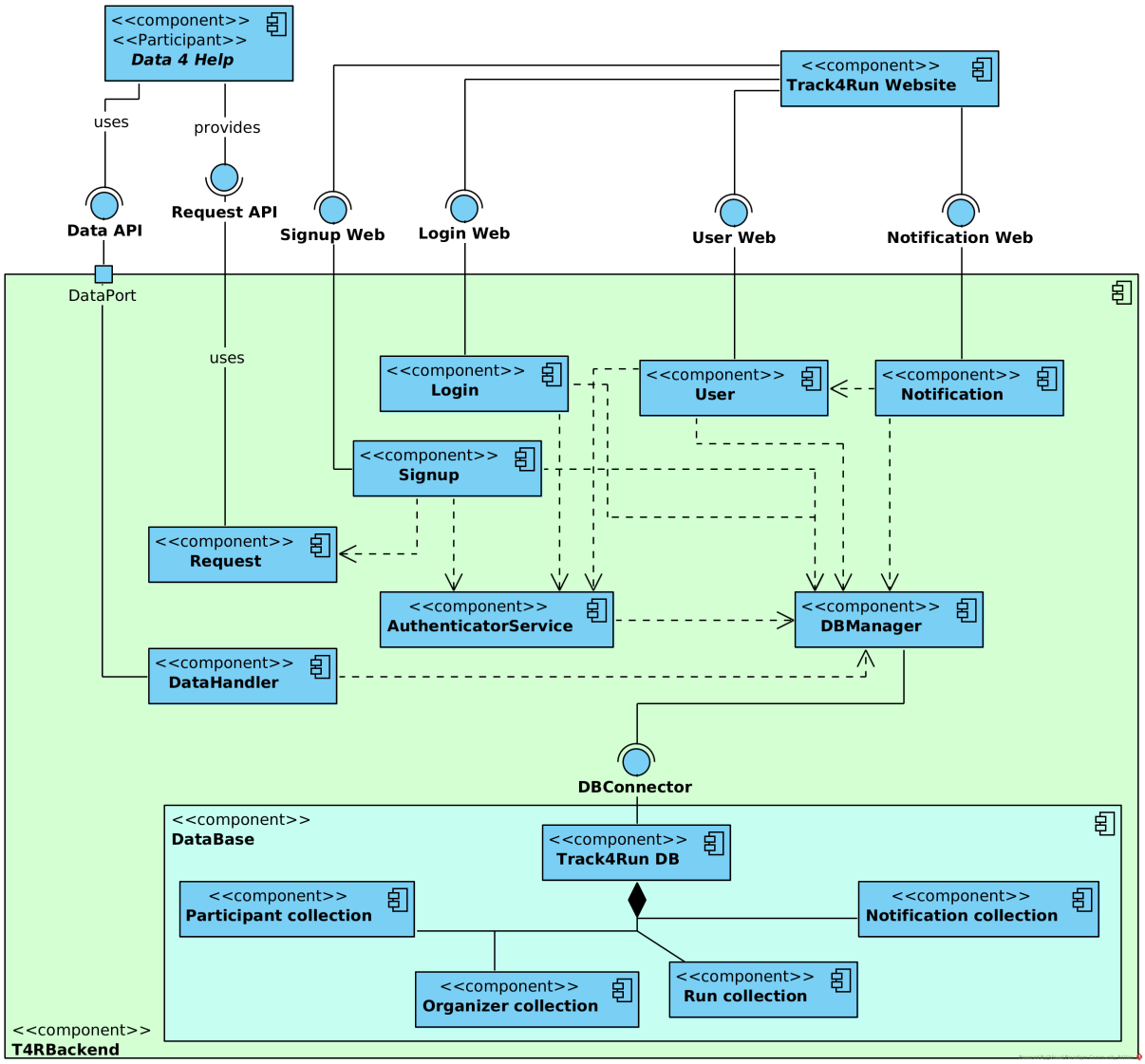

Figure 2.2: Track4Run Component Diagram

2.2.3 AutomatedSOS component diagram

In the Figure 2.3 it can be seen the ASOS components. It can be noticed that the system has a similar structure of T4R: the **ASOSBackend** component owns the **DataBase** component, and provides a **DataAPI** interface to receive the updated information of the individuals.

On the other hand, **ASOSBackend** component sends notifications to the different **Health Care Service** components using the provided **Alarm Interface**.

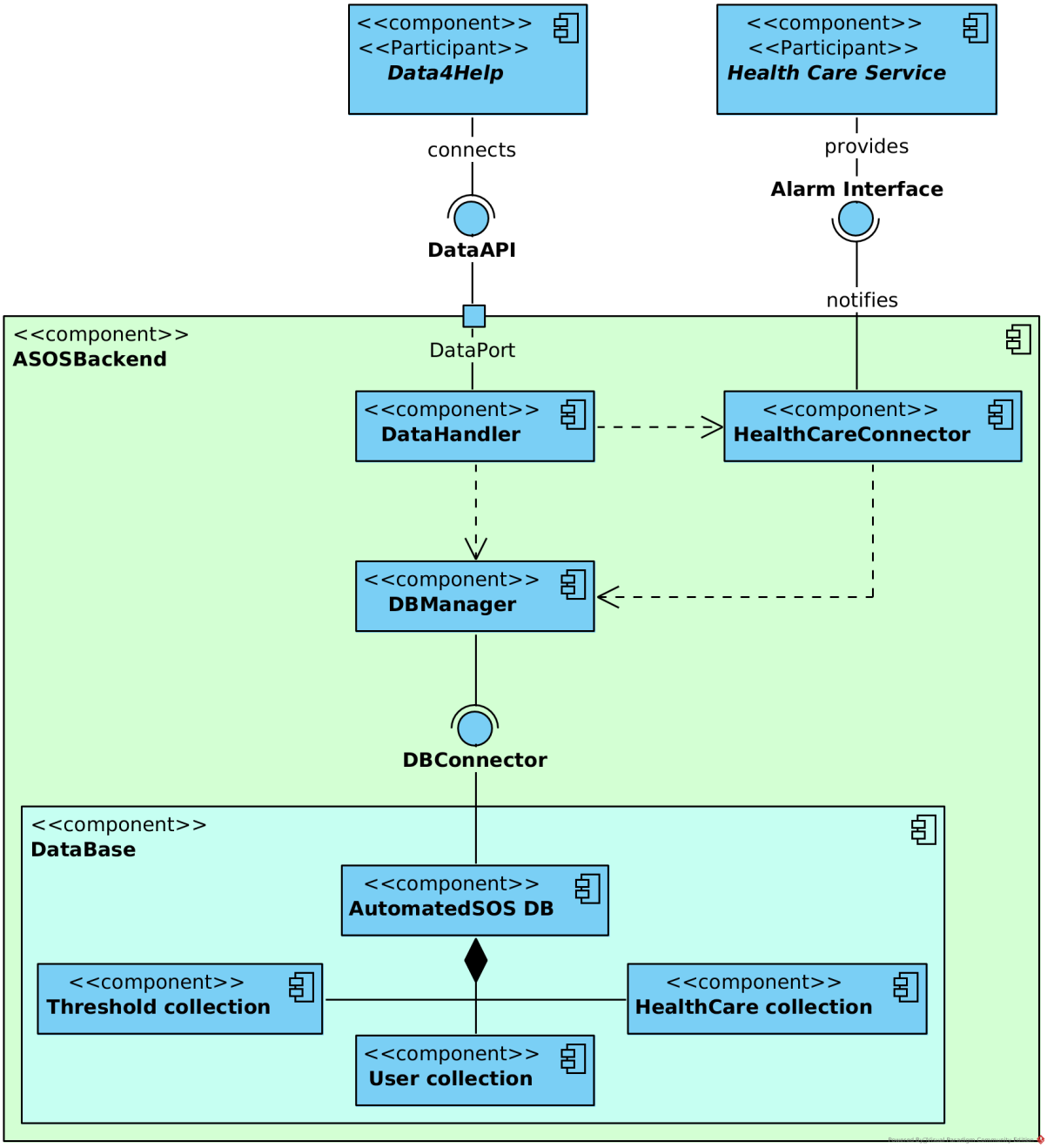

Figure 2.3: AutomatedSOS Component Diagram

2.3 Deployment view

2.4 Runtime view

You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases

2.5 Selected architectural styles and patterns

Please explain which styles/patterns you used, why, and how

2.6 Other design decisions

Chapter 3

User interface design

Provide an overview on how the user interface(s) of your system will look like; if you have included this part in the RASD, you can simply refer to what you have already done, possibly, providing here some extensions if applicable.

Chapter 4

Requirements traceability

Explain how the requirements you have defined in the RASD map to the design elements that you have defined in this document.

Chapter 5

Implementation, integration and test plan

Identify here the order in which you plan to implement the subcomponents of your system and the order in which you plan to integrate such subcomponents and test the integration.

Chapter 6

Effort spent

Team Work	
Task	**Hours**
Understanding the problem	3
Brainstorming	2
World and shared phenomena	2
Definitions, acronyms, abbreviations	1
Software system attributes	2
Alloy coding	7
Checking document	4
Total	20

Table 6.1: Time spent by all team members

Individual Work					
Diego Avila		**Laura Schiatti**		**Sukhpreet Kaur**	
Task	**Hours**	**Task**	**Hours**	**Task**	**Hours**
X	X	Layout	X	X	X
Total	X	**Total**	X	**Total**	X

Table 6.2: Time spent by each team member

Chapter 7

References

- Requirement Analysis and Specification Document: AA 2017-2018.pdf". Version 1.0 - 26.10.2017

- Henriksen, A., Haugen Mikalsen, M., Woldaregay, A. Z., Muzny, M., Hartvigsen, G., Hopstock, L. A., Grimsgaard, S. (2018)
  Using Fitness Trackers and Smartwatches to Measure Physical Activity in Research: Analysis of Consumer Wrist-Worn Wearables. Journal of medical Internet research, 20(3), e110. doi:10.2196/jmir.9157.
  Retrieved from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5887043/

- IEEE. (1993). IEEE Recommended Practice for Software Requirements Specifications (IEEE 830-1993).
  Retrieved from https://standards.ieee.org/standard/830-1993.html

- Sloane, A. M. (2009). Software Abstractions: Logic, Language, and Analysis by Jackson Daniel, The MIT Press, 2006, 366pp, ISBN 978-0262101141.