UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Monitoring of Medication Boxes Using Wireless Sensors

Manuel Luís Gama Meruje

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Pedro Ricardo Morais Inácio
Supervisor: Eng.º Filipe Miguel Monteiro da Silva e Sousa

Covilhã, outubro de 2016

Acknowledgements

The order of the acknowledgements does not convey any meaning regarding the importance of these persons in my life.

I would like to thank my supervisor, Professor Doutor Pedro Ricardo Morais Inácio, for the opportunity to integrate his team and for following my academic path since my Bachelor's degree.

I would like to thank Filipe Sousa, Paulo Silva, Ricardo Graça, Sílvia Rêgo, and the whole Fraunhofer Assistive Information and Communication Solutions (AICOS) team for the opportunity they gave me to collaborate in their work, and for all the support they gave me while I was developing this project and dissertation.

A special thanks to the persons on *Centro de Convívio para Idosos do Bonfim* that helped me to gather data for my datasets, Adelaide, Ludovina, Jaime, Estrela, Beatriz, Felismina, Rosa, Silvina, Diamantina, António, Virgília, José, Ivone, Maria, and Elvira. From the bottom of my heart, thank you for your patience and help.

A different and important acknowledgement goes to my parents, Joaquim Meruje and Margarida Meruje, my sister, Ana Rita Meruje, my goddaughter, Gabriela Eusébio, my brother-in-law, Luís Eusébio, my mother-side greatparents, Joaquim and Rosária, and my father-side greatparents, António e Rosário. Thank you for all your love, for all your efforts, and for supporting me every day since I was a little child.

My next thoughts go to my friends, for their patience, and for constantly motivating me in this long journey: João Mateus, Rafael Robalo, Flávio Gomes, Joana Nova, Jessica Ferreira, José Godinho, Renato Lopes, Leopoldo Ismael, Pedro Querido, Patrícia André, Filipa Belo, Andreia Marques, Bruno Coelho, Susana Correia, Catarina Mendonça, Diogo Camilo, Pedro Maleitas, João Capinha, Vitor Pereira, Cátia Malhadas, Carolina Roxo, André Baptista, Bernardo Sequeiros, Acácio Correia, João Neves, and all the others I forgot to mention: thank you.

Last, but never the least, an acknowledgement to the special person that always motivated and encouraged me, each day since the beginning, to continue, and to finish this stage of my academic life: my girlfriend, Filipa Mendonça. Thank you for all of your patience and kindness.

To all of you, a wholeheartedly thank you.

Resumo

Uma baixa adesão à terapêutica é um problema real entre os adultos que pode levar a sérias repercussões nas suas vidas. A adesão à terapêutica é definida pela Organização Mundial de Saúde como *a medida em que o comportamento de uma pessoa coincide com as recomendações de um prestador de cuidados de saúde*. Uma baixa adesão a uma determinada terapêutica pode comprometer, em muitos casos, os benefícios do tratamento. Além disso, tomar medicação errada pode levar a efeitos secundários não desejados, condições de saúde adversas e visitas a hospitais.

Esta dissertação descreve um trabalho focado na concepção, desenvolvimento e investigação de uma solução para a monitorização de caixas de medicação com caixas de sensores a elas acopladas. As principais contribuições deste trabalho incluem o desenvolvimento de uma aplicação móvel, um estudo em como classificar dados de gestos de caixas de medicação, uma implementação do algoritmo que obtém dados das caixas de sensores e a integração do algoritmo de aprendizagem automática na aplicação móvel. Foi desenvolvida uma prova-de-conceito de alarmes de medicação no âmbito deste projecto de Mestrado. Os dados dos sensores são recebidos pelo protótipo através de um módulo que integra a ligação e transferência de dados das caixas de sensores via ligação sem fios. Outro módulo implementa funções de extração de métricas que serão usadas sobre os dados dos sensores inerciais contidos nas caixas de sensores. As métricas calculadas, também chamadas de características, são passadas para um algoritmo de aprendizagem automática, que está integrado no módulo de classificação de dados e extração de características, para posterior identificação de dados.

No desenvolvimento da solução, foi feito um estudo aprofundado sobre como classificar dados inerciais de gestos de caixas de medicação. Este estudo incluiu a criação de dois conjuntos de dados com diferentes características que, depois de serem pré-processados, foram submetidos a diferentes algoritmos de aprendizagem automática, sendo os seus resultados analisados neste documento. O processo de coleção de dados foi feito em dois locais distintos, correspondendo a um ambiente controlado e um ambiente não controlado. Os resultados obtidos mostram que é possível identificar os gestos considerados no ambiente controlado, tendo os melhores resultados chegado a 97.9% de taxa de acerto. Os resultados obtidos para o conjunto de dados do ambiente não controlado (que contou com a participação dos utilizadores alvo da aplicação) demonstraram que ainda há aspetos a melhorar antes de produzir uma versão final da solução.

Palavras-chave

Acelerómetro, adesão à terapêutica, aplicação smartphone, aprendizagem automática, caixas de medicação, caixas de sensores, giroscópio, reconhecimento de actividades, reconhecimento de padrões, sensores inerciais.

Resumo alargado

Esta secção resume a dissertação de uma forma mais abrangente que a secção Resumo e na Língua Portuguesa. A sua estrutura segue o encaminhamento do corpo do documento.

Introdução

O primeiro capítulo tem como objetivo enquadrar o trabalho descrito ao longo desta dissertação, definindo o enquadramento, o problema e os objetivos da mesma. Ainda neste capítulo, é apresentada a abordagem adotada para a resolução do problema proposto e as principais contribuições do trabalho.

Enquadramento, Descrição do Problema e Objectivos

A adesão a uma determinada terapêutica é definida como *a medida em que o comportamento de uma pessoa coincide com as recomendações de um prestador de cuidados de saúde*. Uma baixa adesão à terapêutica de uma certa prescrição pode comprometer os benefícios do tratamento em muitos casos. Adicionalmente, tomar medicação errada pode levar a efeitos secundários indesejados, como agravamento da saúde ou visitas a hospitais. O problema pode ser ainda mais detalhado, referindo que é estimado que cerca de 30% a 50% das prescrições nunca são tomadas, como defendido em [HRM+15]. Existem vários fatores diferentes que podem levar à não adesão de uma determinada terapêutica, sendo alguns deles a falta de lembrança do horário de toma de medicação, a falha da toma da medicação a horas, a falha em antecipar a toma da medicação e a falha na preparação da medicação para futuras tomas [AOM+11].

Para além destas, outras causas que agravam a adesão à terapêutica são referidas em [HRM+15], como a duração do tratamento, o custo da medicação, a frequência da dosagem, a complexidade da prescrição e ainda a má comunicação com o paciente. É também referido que o nível de conhecimento do paciente, ideias e experiências, bem como as da família e amigos, mostram estar correlacionadas com a adesão à terapêutica. Em [Far99] os métodos de medição de medicação são categorizados em dois grupos: os métodos diretos, que correspondem à deteção de drogas em algum dos fluidos biológicos (normalmente urina ou sangue) e observação direta do paciente a tomar a medicação; e os métodos indiretos, que compreendem a utilização de dispositivos eletrónicos de monitorização, revisão da prescrição e relato do paciente sobre a sua medicação.

O trabalho descrito neste documento é focado no estudo de soluções para monitorização de medicação e na investigação e desenvolvimento de um sistema protótipo para monitorização de medicação, baseado numa aplicação móvel combinada com caixas de sensores e algoritmos de aprendizagem automática. A solução consiste na utilização de caixas de sensores acopladas a

caixas tradicionais de medicação para identificar quando e como um utilizador interage com a caixa de medicação. Isto permite que a solução seja de baixo custo, uma vez que o utilizador não tem de comprar uma caixa de medicação nova e pode reutilizar a caixa de sensores em outras caixas de medicação. Um objetivo importante é perceber como os algoritmos de aprendizagem automática podem ajudar na identificação das aberturas ou fechos da caixa através de dados inerciais dos sensores. Outro objetivo importante é desenvolver uma prova-de-conceito que integre as ferramentas, tecnologias e funcionalidades necessárias para o funcionamento básico da solução.

Abordagem e Principais Contribuições

Esta secção apresenta a abordagem a ser seguida durante o desenvolvimento da solução descrita nesta dissertação e as principais contribuições científicas resultantes do trabalho desenvolvido. A abordagem seguida é descrita da seguinte forma:

- Contextualização com o problema descrito nesta dissertação e investigação de trabalho relacionado nesta área de conhecimento, que inclui uma revisão de soluções do estado da arte;
- Contextualização com as ferramentas e tecnologias necessárias na implementação da aplicação Android;
- Engenharia de Software, Arquitetura de Sistema e delineação de passos para o desenvolvimento do protótipo;
- Desenvolvimento da aplicação protótipo para o sistema operativo Android;
- Recolha de dados junto de dois grupos distintos;
- Testar os conjuntos de dados e o protótipo desenvolvido.

As principais contribuições podem então ser descritas da seguinte forma:

- Uma aplicação para lembretes de medicação para o sistema operativo Android, que esteja preparada para ligar à caixa de sensores que recolhem dados de caixas de medicação tradicionais. A engenharia de software e detalhes da implementação da aplicação são os temas principais dos capítulos 3 e 4, respetivamente;
- Um estudo de como classificar dados das caixas de medicação. Este estudo e os resultados mais promissores são descritos no capítulo 5;
- Uma implementação do algoritmo que recebe os dados das caixas de medicação;
- A integração do algoritmo de classificação de dados recebidos das caixas de sensores para a aplicação móvel desenvolvida.

O código fonte da aplicação foi entregue à Fraunhofer AICOS, no âmbito de um protocolo de colaboração, onde este trabalho foi feito. A aplicação móvel usa bibliotecas desenvolvidas e fornecidas pela Fraunhofer.

Estado da Arte

O capítulo 2 faz uma breve introdução à monitorização de medicação, explora as soluções existentes e explicita as tecnologias e ferramentas utilizadas no desenvolvimento da dissertação.

No tema de sistemas de monitorização de medicação, alguns dos trabalhos mais relevantes permitem perceber que tipo de soluções já foram exploradas e como o sistema deve ser desenhado. Algumas soluções, como as de Asai *et. al.*, [AOM+11], Pang *et. al.*, [PTC14] e Li *et. al.*, [LPNF14], são compostas por vários módulos e capazes de monitorizar para além de medicação. Outras mais simples, como a de Morena *et. al.*, [DM14], consistem apenas num frasco de medicação inteligente capaz de se ligar à *cloud*, agregando informação em tempo real sobre o próprio frasco.

Algumas das tecnologias brevemente descritas incluem: uma biblioteca com elementos de *design* disponibilizada pela Fraunhofer; o sistema *Pandlet* – as caixas de sensores – também fornecidas pela Fraunhofer, para o desenvolvimento do projeto; o *bundle* de software Android Studio, que contém o software necessário para o desenvolvimento de aplicações móveis para o sistema operativo Android; o formato de dados JavaScript Object Notation (JSON); *framework* Waikato Environment for Knowledge Analysis (WEKA), uma ferramenta para análise de conhecimento e conjuntos de dados.

Este capítulo apresenta a revisão de literatura, bem como o trabalho relacionado com o que é desenvolvido no âmbito deste projeto, ajudando a perceber quais as soluções já existentes e possíveis vantagens e desvantagens. Para além disso, as tecnologias e ferramentas utilizadas são brevemente introduzidas, bem como os motivos para as escolhas destas.

Arquitetura do Sistema

O capítulo 3 da dissertação define o funcionamento do protótipo, os objetivos a serem atingidos com a sua implementação e fornece uma ideia do funcionamento da aplicação desenvolvida. É ainda descrita a Análise de Requisitos, uma breve descrição do sistema e da engenharia de software. Esta análise tem como objetivo a descrição das funcionalidades que são implementadas no sistema a ser desenvolvido, e como este se deve, ou não, comportar.

É feita ainda uma breve descrição da solução, onde se pode verificar que esta é composta por três módulos principais: o módulo de alarmes de medicação, o módulo de obtenção de dados e o módulo de classificação de dados e extração de características. O módulo de alarmes de medicação integra o sistema de calendários do sistema operativo Android através de um pro-

vedor de conteúdos e da classe *CalendarContract*. O módulo de obtenção de dados integra o sistema *Pandlet*, através de uma biblioteca desenvolvida pela Fraunhofer AICOS, na aplicação protótipo. O módulo de classificação de dados e extração de características integra os algoritmos de aprendizagem automática, através das bibliotecas da *framework* WEKA e implementa as funções que obtêm as características através dos dados de sensores.

A secção da engenharia de software apresenta e descreve alguns diagramas *Unified Modeling Language (UML)* definidos para o sistema e discute ainda, em múltiplas subsecções, os casos de uso, diagramas de atividades, diagrama de instalação e componentes. Para além disso, são incluídos alguns *mockups* para uma ideia visual acerca do que esperar da aplicação.

Os casos de uso definem uma lista de funcionalidades que o sistema deve fornecer quando interage com os atores do sistema. Estes são descritos, quer na forma de diagrama UML, quer na forma escrita, para a aplicação desenvolvida no âmbito da dissertação. Para além disso, é ainda feita uma descrição dos possíveis atores do sistema. Neste caso, os possíveis atores do sistema são o *provedor de cuidados de saúde* e o *paciente*. Os diagramas de atividade apresentam possíveis fluxos de interação entre o ator do sistema e o próprio sistema a ser desenvolvido. O diagrama de instalação contém o esquema geral da arquitetura do sistema, mostrando quais os componentes que nele estão inclusos.

Desenvolvimento do Protótipo

O capítulo 4 foca-se na fase de desenvolvimento do protótipo da solução proposta, descrevendo a aplicação Android desenvolvida e fazendo um breve resumo do funcionamento da aplicação. É ainda discutido um teste preliminar ao protótipo, onde apenas algumas das funcionalidades são experimentadas e a sua forma de operar é explicada.

Para além da aplicação principal, foi desenvolvida uma segunda aplicação Android simples que integra a biblioteca das caixas de sensores para que a recolha de dados possa ser feita em separado. Esta não é descrita uma vez que foi utilizada apenas para este propósito.

A aplicação Android de lembretes de medicação é um dos principais resultados deste projeto de Mestrado. É feita uma pequena introdução a cada módulo, de modo a que alguns detalhes de implementação sejam descritos neste capítulo. Na descrição dos módulos são explicados alguns trechos de código que foram considerados importantes. São ainda descritos alguns testes feitos à aplicação, onde se pode verificar que as funcionalidades principais foram corretamente implementadas.

Conjuntos de Dados e Análise dos Resultados

O capítulo 5 foca-se na forma como os conjuntos de dados para teste foram gerados e nos resultados obtidos após teste aos diferentes algoritmos de aprendizagem automática tidos em

conta neste trabalho.

É feita uma descrição dos gestos considerados para o estudo de como são retirados comprimidos de caixas de medicação. É ainda descrito o pré-processamento aplicado aos conjuntos de dados na fase de preparação para os algoritmos de aprendizagem automática e introduzido o conceito de característica. Para além disso, são ainda apresentadas as características extraídas para efeitos de classificação de dados.

Dois conjuntos de dados (contendo dados inerciais de gestos aplicados a caixas de medicação) distintos foram criados, tendo um deles sido recolhido em ambiente controlado, nomeadamente nas instalações da Fraunhofer AICOS, e outro em ambiente não-controlado, nomeadamente no Centro de Convívio para Idosos do Bonfim. Os dados foram recolhidos com diferentes frequências de amostragem e as caixas de sensores testadas em mais que uma posição quando acopladas a caixas de medicação. Quatro algoritmos de aprendizagem automática foram testados no âmbito deste projeto de Mestrado: *J48*; Naïve-Bayes (NB); Sequential Minimal Optimisation (SMO); e k-Nearest Neighbor (kNN). Para além disto, foram utilizadas duas formas de validação dos algoritmos: divisão dos conjuntos de dados em treino (66%) e teste (33%); e *k-fold cross-validation* ($k = 10$).

No total, foram efetuados cerca de 3456 testes para o conjunto de dados criado em ambiente controlado e cerca de 576 para o conjunto de dados criado em ambiente não-controlado. Foi possível concluir que os melhores resultados foram atingidos pelo algoritmo SMO para os conjuntos de dados recolhidos com frequências de amostragem entre 50 Hz e 100 Hz. Quando os mesmos conjuntos de dados foram submetidos ao algoritmo NB, este mostrou ter o pior desempenho entre os algoritmos escolhidos para teste.

Conclusão

O capítulo 6 enumera as principais conclusões que podem ser retiradas do trabalho desenvolvido durante o projeto de Mestrado. Para além disso, são mencionadas algumas direções para possível trabalho futuro.

Após a revisão da literatura, foi concluído que seria necessário fazer uma investigação para perceber como os algoritmos de aprendizagem automática podiam ajudar a identificar os dados provenientes dos sensores. Depois da análise aos resultados dos testes feitos, foi possível perceber que várias configurações permitiram o reconhecimento de gestos nos conjuntos de dados.

Foi possível perceber, de uma maneira geral, que o desempenho do algoritmo SMO se destacou mais que os outros, e que o algoritmo NB teve um pior desempenho. Verificou-se ainda que, quando se tenta discriminar qual dos compartimentos das caixas de medicação é aberto ou

fechado, os resultados são piores do que no caso em que estes não se discriminam. Os resultados para o caso em que os gestos de *pegar na caixa de medicação* e *pousar a caixa de medicação* não foram considerados foram, de uma forma geral, altos.

Algumas das possíveis causas para os piores resultados são ainda descritas: a discrepância entre o contexto dos dois conjuntos de dados recolhidos; as diferenças entre a maneira como cada sujeito faz os gestos; as diferenças nos tamanhos dos conjuntos de dados; a diferença entre o tempo em que um sujeito demora a fazer os gestos; e a possível existência de atrasos na transmissão de dados através da pilha Bluetooth Low Energy (BLE).

A maior parte dos objetivos iniciais do trabalho foi atingida. Contudo, ao longo do caminho colocaram-se alguns desafios que mereceram maior atenção, levando a que, por exemplo, se fizesse um estudo mais aprofundado acerca da aplicação de algoritmos de aprendizagem automática do o inicialmente esperado. Os resultados deste trabalho de mestrado contribuíram para a solução pretendida, embora ainda haja aspetos a melhorar antes de produzir a versão final do sistema.

Abstract

Medication adherence is a real problem among older adults which can lead to serious repercussions on their health and life. Adherence is defined by the World Health Organization as *the extent to which the behavior of a person corresponds with recommendations from a health care provider*. A low medication adherence to a certain prescription can undermine the treatment benefits in many cases. Moreover, taking wrong medication may lead to unwanted secondary effects, adverse health conditions, and visits to the hospital.

This dissertation describes the work focused on the design, development, and research of a solution for monitoring medication boxes using attached sensors. The main contributions of this work include the development of a mobile application, a study on how to classify data from medication box gestures, an implementation of the algorithm that retrieves data from sensor boxes, and an integration of the data classification algorithm into the mobile application. A medication reminder proof-of-concept was developed in the scope of this Master's project. Sensor data is received by the prototype through a module that integrates the connection and data transference from the sensor boxes via wireless communication. Another module implements metric extraction functions that are applied to the inertial sensor data retrieved from the sensor box. The calculated metrics, herein corresponding to *features*, are passed to a machine learning algorithm, integrated in the data classification and feature extraction module, for posterior data identification.

An in-depth analysis on how to classify inertial data from medication box gestures was conducted during the development of the solution. This in-depth analysis included the creation of two datasets with different characteristics which were preprocessed and fed to several machine learning algorithms. The analysis of the results outputted by the algorithms is included in this document. The dataset collection took place in two different locations, corresponding to a controlled environment and to a non-controlled environment. The obtained results showed that it is possible to identify the gestures in the dataset for the controlled environment, with the best results achieving a true positive rate of 97.9%. The results obtained for the dataset of the non-controlled environment (which was created with target users) showed that there are still many aspects that need to be improved before a final version of the solution is released.

Keywords

Accelerometer, activity recognition, gyroscope, inertial sensors, machine learning, medication adherence, medication boxes, pattern recognition, sensor box, smartphone application.

Contents

List of Figures

List of Tables

Acronyms

ACM Association for Computing Machinery

ADB Android Debug Bridge

AICOS Assistive Information and Communication Solutions

ANN Artificial Neural Network

BDM Bayesian Decision Making

BLE Bluetooth Low Energy

CCS Computing Classification System

DTW Dynamic Time Warping

GPIO General Purpose Input/Output

GUI Graphic User Interface

GPRS Global Packet Radio Service

GPS Global Positioning System

HCI Human-Computer Interaction

IC Integrated Circuit

IDE Integrated Development Environment

IoT Internet-of-Things

JSON JavaScript Object Notation

kNN k-Nearest Neighbor

LED Light Emitting Diode

NB Naïve-Bayes

MAC Media Access Control

MEMS Microelectromechanical System

NDEF NFC Data Exchange Format

NFC Near Field Communication

OS Operating System

PDA Personal Digital Assistant

RBA Rule Based Algorithm

RFID Radio-Frequency Identification

RMS Root Mean Square

SDK Software Development Kit

SMA Signal Magnitude Area

SMO Sequential Minimal Optimisation

SVM Signal Vector Magnitude

SVM Support Vector Machines

SMS Short Message Service

UBI Universidade da Beira Interior

UI User Interface

UML Unified Modeling Language

UUID Universally Unique Identifier

WBSN Wireless Biomedical Sensor Network

WEKA Waikato Environment for Knowledge Analysis

XML eXtensible Markup Language

Chapter 1

Introduction

This chapter defines the focus, problem, and objectives addressed by the research work described in this dissertation, as well as the steps taken to solve that problem. The next-to-last section presents the main contributions for the advance of scientific knowledge and the last section describes the contents of each chapter of this dissertation.

1.1 Focus and Scope

This Master of Science (M.Sc.) project is focused on a new approach to medication monitoring systems. The project, described in this document, is the result of a collaboration between Universidade da Beira Interior (UBI) and Fraunhofer Assistive Information and Communication Solutions (AICOS).

Adherence and *compliance* are two terms frequently used in the literature to describe medication-taking behaviours, and also used while measuring the reliability and effectiveness of a certain medication treatment on a patient. Although they have a similiar meaning, they are slightly different. The World Health Organisation has defined *Adherence* as "the extent to which a person's behaviour - taking medication, following a diet, and/or executing lifestyle changes - corresponds with agreed recommendations from a health care provider" in [Sab03]. A strong emphasis was placed on the need to differentiate *adherence* from *compliance*. This is due to the fact that the term *compliance* suggests that a person is passively following orders from a doctor orders, rather than actively collaborating in the treatment process, while *adherence* emphasizes the need for agreement and that the patient is free to decide whether or not to adhere to a specific prescription recommended by the prescriber and, as such, failure to do so should not be a reason to blame the patient.

A low medication adherence to a certain prescription can undermine the treatment benefits in many cases. Additionally, taking a wrong medication, or a correct one at a wrong time, can lead to visits to the hospital, adverse health conditions or unwanted secondary effects. There are many different factors that lead to medication non-adherence, being some of them the failure to remember correct regimen, the failure to recall taking medication on time, the failure to antecipate taking medication and the failure to prepare medication for future use, as stated by [AOM+11]. This is a problem being addressed by many [DM14] [PTC14] [CDF+15], leading to a wide spectrum of solutions, ranging from simple smartphone applications that spawn alarms

to take the medication on time, to custom smart medication boxes that will alert the users via auditory or visible signals. Both solutions represent a great burden to the users. On the first case, users must dismiss the alerts on the smartphone, even when they remember to take the medication. If they are not near the medicine at the time of the alarm, they may as well dismiss it, but, subsequently, they may also forget to take the medicine. On the second case, the solution ends up being too expensive and requires the user or the caregiver to constantly refill the device with the correct medication.

The work described herein is focused on the study of existing solutions to medication monitoring, and on the research and development of a prototype of a medication monitoring system, based on a mobile application combined with sensor boxes and levering machine learning algorithms. Its scope falls within the specific research area of assisted living, and signal processing of the Computer Science discipline. In the Association for Computing Machinery (ACM) Computing Classification System (CCS), this work would fall in:

- *Human-centered computing~User interface programming*;
- *Human-centered computing~Ubiquitous and mobile devices*;
- *Computing methodologies~Machine learning*.

1.2 Problem Statements and Objectives

The problem addressed in this Master's project is the one of correctly and efficiently identify when a patient takes a certain medication. If a patient carefully follows his prescription, it has a higher chance of improving his health conditions. A prescription is a document written by a physician, to a pharmacist, containing medication instructions for a specific patient. By having data on medication intakes, a caregiver can understand whether, given the prescription, the health of the patient has improved or not, as a result of the adherence to the scheduled medication.

The problem can be further detailed by referring that is estimated that between 30% to 50% of drug prescriptions are never taken, as stated by the authors of [HRM+15], and which can lead to serious repercussions on the health of the patient. Some factors that also lead to a low adherence prescription are referred in [VHVRD01], such as the duration of the treatment, the cost of the medication, the frequency of dosing, the complexity of the regimen, and poor communication with the patient.

Other referred factors contributing to non-compliance include unresolved concerns of the patient, the diagnosis, the absence of symptoms, the time between taking the drug and its effect, and the fear of adverse secondary effects. It is also stated that the knowledge of the patients,

ideas and experiences, as well as those of family members and friends, have been shown to correlate with the medication adherence.

Starting from that, one can understand that there is a need to improve the adherence of a patient to its prescription. Another concerning problem is the reliability of the existing methods of medication adherence measurements. Methods for medication measurement can be categorized, as stated on [Far99], into two different groups: the direct methods and the indirect methods. Direct methods provide proof that the medication has been taken by the patient: detection of the drug in some biological fluid (usually urine or blood), and direct observation of patient taking the medication are part of this group. Indirect methods are the most commonly used, and the respective methods that are part of this group are, e.g., self-report of the patient, use of electronic monitoring devices and prescription record review.

The main objective of this Master's project is to create a solution for medication monitoring imposing a lighter burden on the users. The solution will consist in the usage of wireless sensors boxes, attached to traditional medication boxes, to identify when and how the user interacts with the medication box. This allows the solution to be low cost, since the user does not have to buy a new medication box and can re-use the sensor box in multiple medication boxes. An important goal is to understand how machine learning can help on the identification of box openings or closures by using inertial data of the sensor boxes. Another important objective is to develop a proof-of-concept that integrates the tools, technologies, and functionalities necessary for the basic functioning of the solution. Another objective is to implement a functionality, in the system, that allows a caregiver to generate a report on the gestures that were made with the medication box.

Please note the system will monitor the medication box. The behaviour of the caretaker and what he or she does with the medication after taking the medicine out of the medication box is out of the scope of this dissertation.

1.3 Adopted Approach for Solving the Problem

To achieve the objectives described in the previous section, the research work was divided into the following phases:

1. Contextualization with the problem addressed in this dissertation and with related research work on this field of knowledge, which included a review on state-of-the-art solutions;
2. Contextualization with tools and libraries needed in the implementation of the Android application;
3. Software Engineering and System Architecture and guidelines to develop the preliminary

prototype;

4. Development of the prototype application for Android Operating System (OS);
5. Gathering data for datasets, among two different groups;
6. Testing the datasets, and the developed prototype.

1.4 Main Contributions

The main contribution of this dissertation is the proposal of a new solution to medication monitoring using a sensor box that is attached to medication boxes and a smartphone. Other important contributions can be summarized as follows:

1. A medication reminder application for Android OS that is able to connect to a sensor box that collects data from a traditional medication box. The software engineering and implementation of the application are the main subjects of the chapter 3 and 4, respectively;
2. A study on how to classify data from the medication boxes. This study and the most promising results are described in chapter 5;
3. An implementation of the algorithm that retrieves data from the sensor box;
4. The integration of the algorithm to classify data retrieved from the sensor box into the developed mobile application.

The source code of the application was delivered to Fraunhofer AICOS, under the umbrella of the collaboration protocol in which this work was performed. The mobile application uses libraries developed and provided by Fraunhofer as well.

1.5 Dissertation Overview

The body of the dissertation is composed by five chapters. The contents of each one of the chapters can be summarized as follows:

- Chapter 1 - Introduction — presents the focus and scope of the work described in this dissertation, as well as the addressed problem and objectives. It includes subsections describing the objectives and the main problem, and the adopted approach for solving the problem and fulfilling the objectives;
- Chapter 2 - State-of-the-Art — contains a discussion on available solutions and techniques to tackle the problem addressed by this project, along with the enumeration of their respective advantages and disadvantages. It also contains a broader revision of the literature on this field of knowledge, with focus on some related works;

- Chapter 3 - System Architecture — describes the architechture of the proposed system, including the requirement analysis, software engineering, system architechture and use case diagrams;
- Chapter 4 - Prototype Development — discusses important points concerning the development of the prototype, includes a brief description of the involved algorithms, libraries, technologies, prototype approaches, and describes a preliminary test made to the prototype;
- Chapter 5 - Datasets and Analysis of the Results — discusses the creation of the datasets that were used for the research on how to classify data from sensor boxes attached to medication boxes. It also describes how the experiments were conducted and presents their most promising results;
- Chapter 6 - Conclusion — ends this Master's dissertation by drawing the main conclusions, with focus on the results, and pointing out directions for future work.

Apart from the main chapters described above, this dissertation contains two appendices. The contents from the appendices can be summarized as follows:

- Appendix A - Software Engineering Diagrams — contains some of the diagrams elaborated during the Software Engineering of the prototype. These diagrams are mentioned in chapter 3;
- Appendix B - Prototype Development Listings — includes a small set of listings containing relevant code excerpts described in chapter 4 of the dissertation.

Chapter 2

State-of-the-Art and Technologies

2.1 Introduction

Medication monitoring is a common technique that allows to evaluate the adherence of a patient to a certain medicine. Knowing, with some degree of certainty, the adherence level to a medication regimen is important in medical research and clinical practise. Results of clinical trials cannot be interpreted without adherence information. Variations on adherence can distort the response rate and alter the number of patients required to detect a significant difference between treatment and placebo. The authors of [Far99] state that when a therapy fails to achieve a desired clinical outcome, it is often assumed that drug failure has ocurred. Besides, the more accurate and reliable the information on adherence is, the easier it is to follow and understand the health state of a patient, and understand whether, given a specific prescription, it improves or detriorates.

Two types of measures can be used to assess medication adherence, as stated by authors of [Far99]: direct measures, that provide proof that the drug has been taken by the patient - including drug assays of blood or urine, use of drug markers with the target medication, and direct observation of the patient; and indirect measures, which include self-reporting forms filled by the patient, pill counts, use of electronic monitoring devices, and review of the prescription records and claims.

Low medication adherence can have different origins. Among others, the lack of memory from the patients on taking the medication, the failure to prepare the next medication taking, the duration of the treatment, the cost of the medication, the frequency of dosing, and the complexity of the regimen are some of the main origins on what the literature focus on, as stated by [AOM+11].

This makes medication monitoring a hard task to do because most of the existing solutions are not able to correct all the different factors that can lead to a low adherence rate. While measuring the adherence through a form, where the caregiver asks a series of questions to the patient, the patient can lie to the caregiver and compromise the whole diagnostic. The same applies to the pill counting method, where the caregiver can not know whether the patient really took a specific medicine or not.

According to the adopted approach, one of the initial stages of this work was to get acquainted with the subject at hands and study related works.

This chapter of the Master's dissertation presents the literature review on state-of-the-art solutions that are related with the work developed in the scope of this project. As such, on section 2.1 will briefly elaborate on medication monitoring. Section 2.2 contains the literature review on available and related solutions. Then, section 2.3 will describe the tools and technologies used while developing the Master's project.

2.2 Literature Review

Some of the present solutions described in the next section make use the *mHealth* concept. *mHealth* is defined in [KST11] as the medical and public health practice supported by mobile devices, such as mobile phones, patient monitoring devices, Personal Digital Assistants (PDAs) and other wireless devices. *mHealth* involves the use of the core utility of mobile phones – voice and short messaging – as well as more complex functionalities and applications including Global Packet Radio Service (GPRS), Global Positioning System (GPS) and Bluetooth technology.

The authors of [VLvW+12] examined the effectiveness of interventions using electronic reminders in improving the adherence of patients to chronic medication. Thirteen different studies met an inclusion criteria: the intervention was aimed to patients who were prescribed chronic medication; the intervention involved an electronic reminder system aimed at improving medication adherence; the reminder was directed to patient; the study design was either a randomized controlled trial or a controlled clinical trial; the study was published in English. Personal and active reminders such as telephone calls and emails from healthcare providers revealed positive effects on adherence rates. However, these reminders can require an extensive time investment from healthcare providers. On the other hand, electronic reminders are automatically sent to patient without the interference of a healthcare provider. Examples are reminder messages sent to the mobile phone of the patient via Short Message Service (SMS) or electronic reminder devices that provides visual and audio alerts. In this study were reviewed thirteen different ones: seven on electronic reminder devices, four on SMS reminders and two on *pagers*. This type of reminders does not require additional effort from professionals and may be easy to integrate in the daily life of the patients. Although there are evidences of short-term effectiveness on electronic reminders and SMS reminders, long-term effects remain unclear.

The authors of [DM14] state that USA medication adherence rates average is of 60%, which is low. The low adherence rates leads to losses of around 300 billion dollars on healthcare in the USA and near 100 billion dollars to pharmaceutical companies of this country. Wireless and low-power embedded computing technologies are enabling the design of real-time pill bottles in order to increase medication adherence. A low-power smart pill bottle is introduced in this

article. The device is connected to a cloud infrastructure and, by using sensors, it is able to aggregate infomations (timestamp and content measurement) in real-time so that patients can be notified later to take their medication at appropriate times. Data is sent from bottles to servers where it is analysed in real-time. If a dose is missed, the system reminds the patient via automated phone call or text messages - as well as on-bottle lights and chimes. If the system notices prolonged non-adherence, it can solicit feedback via text message or call, asking them why the dose was missed.

Most common medication errors are done by patients and include under or over doses due to erratic medication consumption or misundertanding of medication posology, as stated by the authors of [SLBM+13]. It presents an *mHealth* software-based solution for medication self-management that can be installed on a smartphone or tablet. *SapoMed* is capable of preventing medication errors by tracking and managing all the prescribed medication. It is based on web services architecture, a feature that is essential to satisfy the m-Health paradigm of healthcare services. The application uses the device camera to capture the barcode, then it connects to database to retrieve the medication data. *SapoMed* is capable of interacting with the user through visual, sound, and vibration alerts. It also has a calendar view where the user can check its intakes. Furthermore, there is an emergency center capable of sending text messages or e-mails to a predefined contact, which can be an healthcare professional or a familiar person. One of the advantages of this approach is the ability to inform the patient of which is the right medicine to take at a specific hour, by scanning the medicine box with the device camera. On the other hand, the lack of a hardware-based device that confirms whether the patient takes the right medicine or not and the fact that the device always needs Internet connection to retrieve the medication data are disadvantages of the system.

iMedBox is an Internet-of-Things (IoT) solution presented by the authors of [PTC14] consisting of a third-party medication box that is able to register, to collect statistics of all medicine utilities (by reading Radio-Frequency Identification (RFID) tags on them), to record and to prevent medication non-compliance. This objective is achieved by reminding patients to take medicine on time through different alarm types, an electronic pharmaceutical package (*iPackage*) and an electronic wearable that contains biomedical sensors (*iTag*). The box is able to connect to a public area (e.g., an hospital, the medicine supply chain and the emergency help center) through wireless internet, and to the intelligent pharmaceutical package and the electronic wearable through RFID links and Wireless Biomedical Sensor Network (WBSN). *iMedBox* comes with a tablet device with a specific applicaton specially developed to handle data, and to present different data collected from medication and the peripheral devices. The user has the possibility to customize views. The application has two pre-defined views: (i) *Signal View*, for real-time monitoring of biomedical signals, and (ii) *Medication View*, for medication management. The system has a backend system for prescription management, case history management and re-

mote diagnosis. When a new prescription is issued in a hospital, corresponding records are added in by dispatching one prescription into multiple dose lists. Then, when the user opens *iMedBox*, a timely dose list that should be taken in the respective day is retrieved. Finally, medications records will be updated to the database, no matter if the doses are taken correctly or not.

In terms of monitorization, this solution seems to be very complete, but the fact it has many different devices can be confusing to the user. Also, if only one of the features is needed, the specific feature is always dependant on the medication box to interpret data, which can be another disadvantage.

Article [AOM+11] presents a solution based on four different modules:

1. the sensing module, that collects data from real world either by asking users the name of the medication and whenever they take them or by using various kinds of sensing technologies;
2. the inferring module, which translates raw data from the sensors to behaviour data of the users;
3. the ruling module, that is used to analyse behaviours of the users and activities actions for users and then to create an action based on this behaviour;
4. the actuating module, consisting of an output to the user, i.e., a notification to mobile phone of the patient, changing his behaviour;
5. the sensing module consists on a medication table, that is able to weight and identify the medication, and a mobile phone that is used to know when the user leaves home and when it is consuming medication outside their home.

The inferring module is used to infer user behaviour by analysing the sensor events from the sensing module. The ruling module receives user behaviour events from the sensing module and then sends commands to the actuating module. The actuating module consists of: a mobile phone - that eases operation by using SMS as medium to remind the user; a computer - that shows all the information; an information globe - a small device that contains Light Emitting Diode (LED) lights inside to inform different status to the user with corresponding colours. JBoss Drools is a framework that provides an open source business rules engine and business rules management system. This framework is used in both inferring and ruling module. The cited solution uses multicolor LED lights to inform the patient making it more user friendly, which can be an advantage. On the other hand, this solution is not portable, which means that it needs to be set up in a static place.

In paper [CDF+15], a smart pill-dispenser is described. Authors mainly focused on the low-

cost requirement in order to overcome one of the limitations of currently available devices. The novelty of this approach is the adoption of a smart device not only for providing a user-friendly human-machine interface, but also for automatic identification of the patient. This solution uses features of smart devices such as Near Field Communication (NFC) wireless link, to exchange data between the pill-dispenser and the smart device, and the fact that these devices are connected to the Internet. The smart device operating system chosen to this study was the Android OS, since it allows to use the device either as a NFC reader or as a NFC writer. Since the pill-dispenser is not significant to demonstrate the applicability of this approach, a pill-dispenser emulator with an *Explorer 16 Development Board* and a NFC microchip that is able to read standard NFC Data Exchange Format (NDEF) messages was developed. The Android application is used to configure and to interact with the pill-dispenser emulator. When the smart device is approached to the emulator, the application sends a synchronisation command with local date and time information. Then, if an event occurs (either scheduled or manually triggered), the application sends commands to show the user which medication should be taken. User and medication are represented by LED lights. It is left for future work the usage of new technologies such as Bluetooth Low Energy (BLE). The dispenser will only work when the scheduled alarm triggers and the device is tapped on the NFC module. This prevents the patient from taking pills out of the schedule, which can be an advantage by reducing the human error. On the other hand, a possible drawback of this solution is the need to be in a static place.

Article [AAG+12] describes a solution based on a smart pillbox and two associated mobile phone applications (web and native). These components are interconnected through a common online data source. This data source can be only written by the web application and the smart pillbox. The web application is designed to be used by an informal caregiver, possibly the patient, to fill in with information about the medication on the smart pillbox. The smart pillbox is made of four rows of chambers as most people do not need to take pills more than four times a day. These chambers were designed to be large enough for elderly patients remove medication with two fingers. It is powered by an *Arduino* instrumented with LED backlights, used to notify the patient when medication is scheduled to be taken, and sensors that record when and whether the patient takes the medication. Each container chamber, corresponding to each medication, has five possible states: empty, not taken, take now, taken and missed. The mobile application is only able to read data from the online data source and to trigger alarms whenever there is medication to take. When these alarms are triggered, a light is produced on the smart pillbox so the patient can identify which pill needs to be taken. One of the possible drawbacks of the implementation of this solution is that the pillbox has a fixed number of containers. If the user has to take more than four pills per day at different times, this may not be the best solution. Besides this, the pillbox needs to be refilled when the medication is over.

In aged societies, there are lot of elderly recipients who take medicines everyday. Elderly people

frequently forgot to take medicine due to their cognitive detrioration. The authors of [SN14] state that convencional portable reminders are unable to recognize the presence of medicines in the storage spaces. Smartphones are regarded as one of the most important personal items along with a wallet and house keys. As so, the solution developed consists of sensorless smartphone case that has a pill-organizer embbeded. By using the smartphone camera and image processing, the medicine case is able to detect medicines in each storage compartment of the pill organiser. The intelligent pill organiser sends results of the image processing to a database server that can be acessed through a website for medication monitoring. Furthermore, the smartphone has a GPS module. This allows the user to be notified when it goes to certain places (geofencing). For example, if the user goes to a restaurant in a certain time window and needs to take medicine before meals, the system can send an alert based on the time and place. One of the main goals of this project, the detection whether a patient takes the right pill or not, is achieved by this solution. Being portable and having a geofencing mode are a plus that can lead to medication adherence increase. There are possible drawbacks anyway, e.g., the processing resources required to the computer vision algorithms, that detect which pill was taken, are high and may incurr in delayed processing in some smartphones.

Scientists at the University of California developed a solution based in a two-step system for monitoring the patient adherence that is described in [KALS15]. The first step of the system is the detection that the medicine bottle has been opened, using force-sensitive resistors. This information is coupled with the next step: the detection of a pill being swallowed using a smart necklace, which includes a piezoelectric sensor resting in the lower trachea. The skin motion during the swallow of a medication has unique pattern that can be used to confirm that the medication has been ingested after the bottle is opened. Data acquired from the necklace is transmitted to an Android application via BLE, where classification algorithms are used to distinguish between swallowed medication and other types of swallows, such as saliva and water. The novelty of this approach is that, besides creating an alarm to remind the patient of taking the medicine and detecting the act of opening the bottle, the system is able to detect whether the pill is consumed or not. This may be a solution that can effectively detect whether the pill is taken or not, which is a good feature. But, on the other hand, the necklace can be irritating to the patient and if one takes it off, it can result in bad measurements of the medication adherence. Besides this, another possible drawback is the need of a third-party device to retrieve data.

A state-of-the-art solution consisting of ingestible sensors, a skin-worn receiver patch, and a mobile device based user interface is described in [HRM+15]. The microfabricated sensor is designed to be incorporated into every ”digital” capsule during pharmaceutical manufacturing. Upon ingestion and contact with the stomachal fluid, each sensor communicates a unique and private binary number representing the medication and the corresponding dose of interest. The

binary number is stored in the non-volatile memory of the Integrated Circuit (IC). To ensure that the electric signal does not intrefere with common medical instruments, consumer electronics, the electrophysiological signals of the body, and that cells or tissue are not stimulated, the electric signal is sent twice per second in a specific frequency. In combination with a wearable sensor patch and a mobile device based user interface, the sensor provides a system for real-time and continuous measurement of medication adherence. The solution is able to detect whether the patient takes the pill or not. In the other hand, the solution can not detect, *a priori*, if the patient took the right medicine. Besides, it can raise bioethic questions about the privacy and security of the data transmitted by the digital pill.

Authors of [LPNF14] have developed an interoperable system combining in three subsystems: a smart pillbox, a computer, and a wristband. All subsystems are able to wirelessly communicate. Whenever it is time to take a medicine, the computer will send a signal to the wristband so the user is notified to take the medicine. The pillbox is equipped with a RFID reader and the wrist-band is equipped with a RFID tag and different sensors (temperature, gyroscope, accelerometer and compass). This will allow the pillbox to detect when the user is opening the it. Besides this, when the user approaches the pillbox, data from sensors is sent to the computer to be analysed later. Then, when the user takes the pill, a pair of diodes and a photodiode, embedded in each compartment of the pillbox, detects the empty space in the compartment and the lid closes. The pillbox updates the status of the compartment and sends a wireless signal to the computer to remind the user or caregiver to refill it. This system is able to tell if the pill was removed from the pillbox and to predict if it was taken by collecting data of sensors on a wristband by analising the data, which can be an advantage. On the other hand, the fact that it uses a third party system, the computer, making the solution impossible to be portable, may be a disadvantage.

2.3 Brief Introduction on Used Technologies

This section contains a brief introduction to the technologies is used in the scope of this project. SC-Lib is described in the subsection 2.3.1. Subsection 2.3.2 will briefly describe the Pandlets system. Then, subsection 2.3.3 will briefly introduce the Android Studio software. The subsection 2.3.4 introduces the JavaScript Object Notation (JSON) standard. Finally, subsection 2.3.5, swiftly introduces the Waikato Environment for Knowledge Analysis (WEKA).

2.3.1 SC-Lib

SC-Lib (Smart-Companion Library) is a library, developed at Fraunhofer Portugal AICOS, containing Human-Computer Interaction (HCI) elements specially designed and optimized for senior population. One of the main purposes of this library is to facilitate the interaction between the target group and Fraunhofer applications running on Android devices. On the other hand, it makes it easier for developers to deal with the HCI design and provides a standard (uniform) visual identity for different applications. Elements such as buttons, text fields, simple icons

and menus are included in this library.

Since this Master's project is inserted in a Fraunhofer AICOS project that uses the same library for the layout elements, this library was used in the development of the Android application prototype, as well.

2.3.2 Pandlets

Pandlet is a small modular sensor box, developed and manufactured at Fraunhofer AICOS. The device consists in a microprocessor along with inertial (gyroscope, magnetometer and accelerometer) and pressure sensors, General Purpose Input/Output (GPIO), and an Android library that allows connection via BLE to an Android device. For nomenclature clarification, in the remaining part of this document, when *inertial sensors* are referred, only the gyroscope and accelerometer will be considered, since magnetometer was not used in the scope of this Master's project.

While developing the solution, Pandlets were used in the data collection module of the Android application prototype. Although there are similiar hardware in the market, Fraunhofer AICOS provided this hardware for the development.

2.3.3 Android Studio and Android Debug Bridge

Android Studio [Inc11] is the official Integrated Development Environment (IDE) for Android applications development. It is based on the IntelliJ IDEA IDE, and it often comes bundled with the Android Software Development Kit (SDK) Manager, which allows the management of Android OS images, emulator, Java libraries, and all the necessary software for development. One of the main components that is used while developing Android applications is the Android Debug Bridge (ADB), a service that allows the communication between the computer and the Android device, or emulator, via command-line interface.

The solution herein presented in this Master's dissertation is developed in the scope of a Fraunhofer AICOS project that is focused for Android devices. As such, this software was used throughout the development the Android application prototype.

2.3.4 JSON

JSON [Int13] is a lightweight data-interchange format. It is commonly used as an alternative to the eXtensible Markup Language (XML). It is able to deal with the basic JavaScript data types such as number, string, boolean, array, object, and null. A Java library, `org.json`, was used to facilitate the use of JSON within the Android development. This library was mostly used in the medication reminder module of the application.

2.3.5 WEKA

Waikato Environment for Knowledge Analysis (WEKA) [atUoW11] is a popular machine learning workbench that contains a collection of algorithms for data mining operations. One of the main advantages of this framework is it can either be used with a Graphic User Interface (GUI) or be invoked from Java code. Another characteristic of WEKA is that it is released as a open source software.

This software was mainly used in the classification module of the application. Although there is similiar software in the Internet, this was chosen since it is open-source and it offers a relatively simple implementation in Java programming language.

On the proof-of-concept prototype developed in the scope of this dissertation, an Android reduced port of WEKA, developed *Institut für Pervasive Computing* [fPC12], was used.

2.4 Conclusion

Throughout this chapter, the research on the literature was discussed. A review on state-of-the-art solutions, related with the work developed in the scope of this project, was presented as well. This helps to understand what solutions are already available in the community, and to understand which possible advantages and disadvantages they may have.

Furthermore, in this chapter, the used technologies in the development of the solution were briefly introduced, as well as some of the reasons for the choices made to them.

According to the adopted approach, System Architecture and guidelines for the development of the solution will be described and discussed in the next chapter.

Chapter 3

System Architecture

3.1 Introduction

The intermediate stage of this work was to devise a system comprised of a mobile application and associated sensor components and communication. This chapter of the dissertation is going to be focused on the System Architecture of the proposed solution. This is an important part of the project because it defines the functioning of the system, and some of the goals to be reached, as well as providing an idea of the working of the application developed. Section 3.2 will elaborate on Requirements Analysis, describing how the solution must behave from the user and from the system perspective. Then, in section 3.3, there will be a brief description of the system, while section 3.4 describes the Software Engineering of the solution.

3.2 Requirements Analysis

The Requirements Analysis has the objective of describing functionalities that will be implemented in the system to be developed and how it must or must not behave. There are two types of requirements that need to be identified: the Functional Requirements, referring to functionalities the system should provide and be available to the user, and the Non-functional Requirements, describing system restrictions. Both will be presented in this section of the chapter.

3.2.1 Functional Requirements

The functional requirements refer to the features that the system should provide. For the Android application, the functional requirements identified for this system are as follows:

- The application shall allow scheduling of new alarms;
- The application shall allow editing scheduled alarms;
- The application shall allow removing existing alarms;
- The application shall allow seeing the history of medication;
- The application shall allow removing, individually, past alarms;
- The application shall not allow editing past alarms;

- The application shall not allow adding alarms on past dates and hours;
- The application shall generate an alarm or notification when the medication is almost over, depending on the pill count of each box;
- The application shall allow adding new pills to an existing box/change the sensor to another new box;
- The application shall generate an alarm or notification when the scheduled hour for a medicine is approaching;
- The application shall generate an alarm or notification when the user picks the wrong medication at a scheduled hour;
- The application shall generate an alarm or notification when the user picks the any medication out of the scheduled hour.

3.2.2 Non-functional Requirements

Non-functional requirements describes the environment where the system is inserted and what restrictions it has. Different types of non-functional requirements are described in this subsection.

- The application will be developed only for the Android OS and it cannot be used in another system;
- Application data is to be used offline only;
- The application shall allow the access to its data via another calendar application, i.e., Google Calendar;
- The application shall be available when the device is available as well;
- The application shall run either in foreground and background;
- The application shall be used only in portrait mode;
- The application shall be allowed to run when a BLE module is detected on the mobile device;
- The mobile device shall have at least 50 megabytes of free disk space;
- The mobile device shall have a quad-core processor able to run the each modules in different threads.

3.3 Architecture

The system is composed of several modules, namely: a *Classification and Feature Extraction* module, a *Data Collection* module, and the *Medication Reminder* itself. The *Medication Reminder* module accesses the calendar data via *ContentProvider* and *CalendarContract* classes of the Android system. Each medication reminder is considered as a generic calendar event of the Android platform. As such, several methods are implemented to guarantee the main functions of a basic calendar, such as: Insert Calendars, List Events, Insert Events, Update (Edit) Events, and Delete Events.

Since all tests performed during data collections were made using the WEKA framework, the Classification functions of the *Classification and Feature Extraction* module consists on the use of the Java library implementation of machine learning algorithms that WEKA has available. On the *Feature Extraction* functions, several metric calculation methods were implemented in order to transform the raw data from the sensors into *features*. The term *feature* is later explained in the chapter 5, as well as the discrimination of the calculated metrics.

The *Data Collection* module consists of the implementation of the Pandlets Android library. This module implements transparent Bluetooth Low Energy (BLE) stack functions and sensor integration that enables the application to search and retrieve data from Pandlet devices from Fraunhofer AICOS. The module will look for events with associated devices, then tries to connect to those devices, storing the sensor events in a file.

3.4 Software Engineering

This section elaborates further on the software engineering of the system to be developed. It will resort to Unified Modeling Language (UML) diagrams for that purpose. Several subsections discuss or provide more detail in use cases, activities and components. Mockups will be also included, so as to provide a visual idea of what to expect.

3.4.1 Use Cases

This subsection of the chapter presents the Use Cases of the system to be developed. Use Cases are a list of functionalities that the system must provide when interacting with the actors of the system. Figure 3.1 identifies the use cases resorting to the UML diagram, along with the Use Cases, the actors of the system, and their relations. The Use Cases are described with more detail in the following sections.

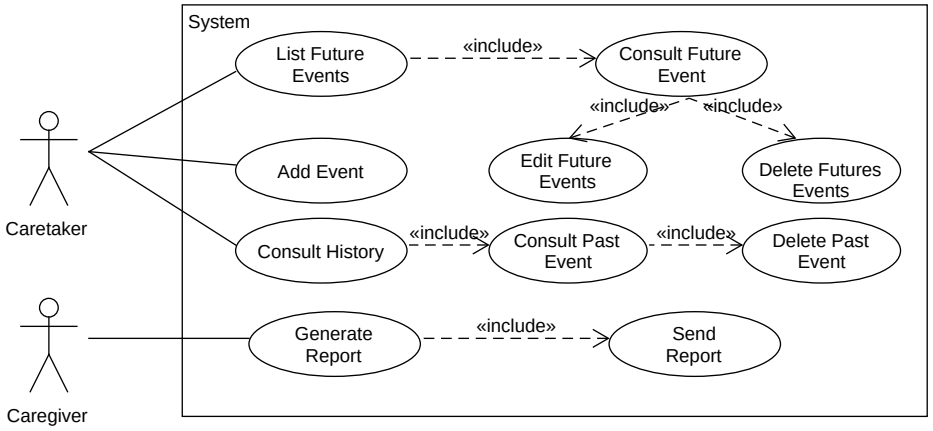

Figure 3.1: Use cases diagram.

3.4.2 Actor Description

The diagram in figure 3.1, identifies two actors that will interact with the application: the *caretaker*, and the *caregiver*. In this application, the *caretaker* will have three main functionalities while the *caregiver* has one. The *caretaker* will be able to consult, edit and delete future events, to add new events and to consult the history and delete past events. On the other hand, the *caregiver* will only be able to generate and send reports via e-mail.

3.4.3 Use Cases Description

There are two main types of users for the system being designed. The *caretaker* is the most important one and, for the sake of the explanation, it will be referred to as the *user* afterwards. The *medication reminder* will be simply referred to as event.

User Functionalities

The functionalities offered to the user can be better described as follows:

- **List Future Events** - The user opens the application and, if any exist, future events in the calendar are presented for the time interval of one month. This user is able to navigate between present and future months;
- **Consult Future Event** - Once the event list is available, the user can choose one of the available events and the details of the event will be displayed. From this functionality, a user can edit or delete future events;
- **Edit Future Events** - This use case is included in the *Consult Future Event* functionality. The user shall be allowed to edit all the future events related to the selected one;

- **Delete Future Events** - This use case is included in the *Consult Future Event* functionality. The user shall be allowed to delete all the future events related to the selected one;
- **Add Event** - The user shall be allowed to add new events to the calendar. When adding a new event to the calendar, the user is able to choose whether it wants to monitorize the medication or not;
- **Consult History** - If there are any past events, the user shall be allowed to consult the history of events that already ocurred;
- **Consult Past Event** - If there are any past events, the user can choose one of the available events by tapping on it and the details of the event will be displayed. From this functionality, a user can delete the chosen past event;
- **Delete Past Event** - This use case is included in the *Consult Past Event* functionality. The user shall be allowed to remove the chosen past event.

Caregiver Functionalities

The functionalities to be provided to the *caregiver* are as follows:

- **Generate Report** - The *caregiver* shall be able to generate a report with the past medication list and each medication status;
- **Send Report** - This use case is included in the *Generate Report* functionality. The *caregiver* shall be able to send the generated report via e-mail.

3.4.4 Activity Diagrams

The diagrams in the referenced figures present the possible interaction flow between the user and the system to be developed. For the sake of the aestetics of the present document, the diagrams contained on this subsection of the chapter will be presented in the appendix chapter A.1.

On the following figure 3.2, it can be observed the flow for the main application loop flow, starting on the *Presents Event List and Available Options*. One can also observe that, when the chosen functionality ends its execution, the flow will return to the state where the loop started.

Figure 3.3 presents the *Add and Edit Event* activity flow. In the *Add and Edit Event* functionality, the user is able to define which are the parameters of the alarm that will be triggered. Parameters that can be modified are the name of the medication and the recurrence of the alarm. The time of the alarms and respective dosages are allowed to be modified on the *Add and Edit Intakes*.

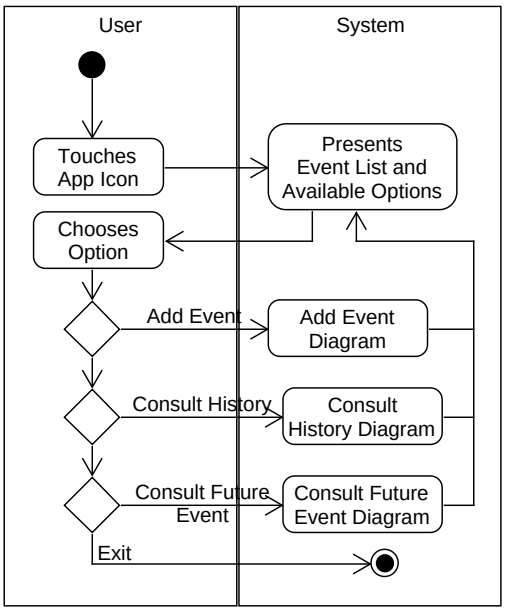

Figure 3.2: System activity diagram.

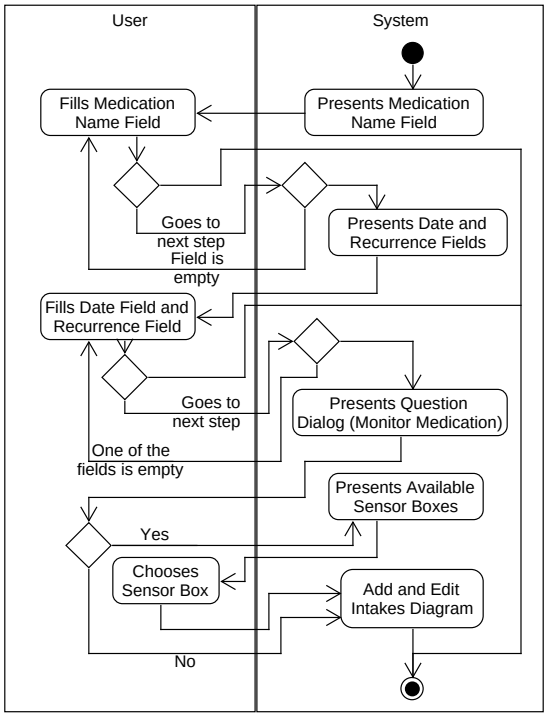

Figure 3.3: Add and Edit Event activity diagram.

For each event, the alarm times and the dosages must be saved so the system can trigger the alarms. The flow for the *Add and Edit Intakes* Use Case is depicted on figure A.3. Note that this flow represents a possible scenario of the *Add and Edit Event* Use Case. It can be noticed that, after the intake list being presented, a user has to add intakes so the system allows the insertion of the medication reminder in the calendar. For all cases herein described, fields are verified for empty data as well.

Figure A.1 presents the flow associated with the *Consult History* Use Case. The user is allowed to consult the list of past events of the medication reminder, as well as the details of an individual past intake. Furthermore, entries from the list are allowed to be deleted while consulting the past event list.

A possible activity flow for the *Consult Future Intake* Use Case is depicted in the following figure A.2. The *caretaker* is allowed to consult the list of future events of the medication reminder, as well as the details of an incoming individual intake. The diagram emphasizes that the caretaker has additional options to edit or delete the medication intake from the medication reminder.

3.4.5 Installation Diagram

Figure 3.4 contains a scheme of the general architecture by means of an installation diagram. As it can be seen in the figure, the application of the system does not need connectivity to the

Internet to operate. The use of an external device, the sensor box, is optional and requires a connection via bluetooth. The internal library to access the calendars database is also a component of the system, as well as the WEKA library, that allows the use of the classification algorithms.

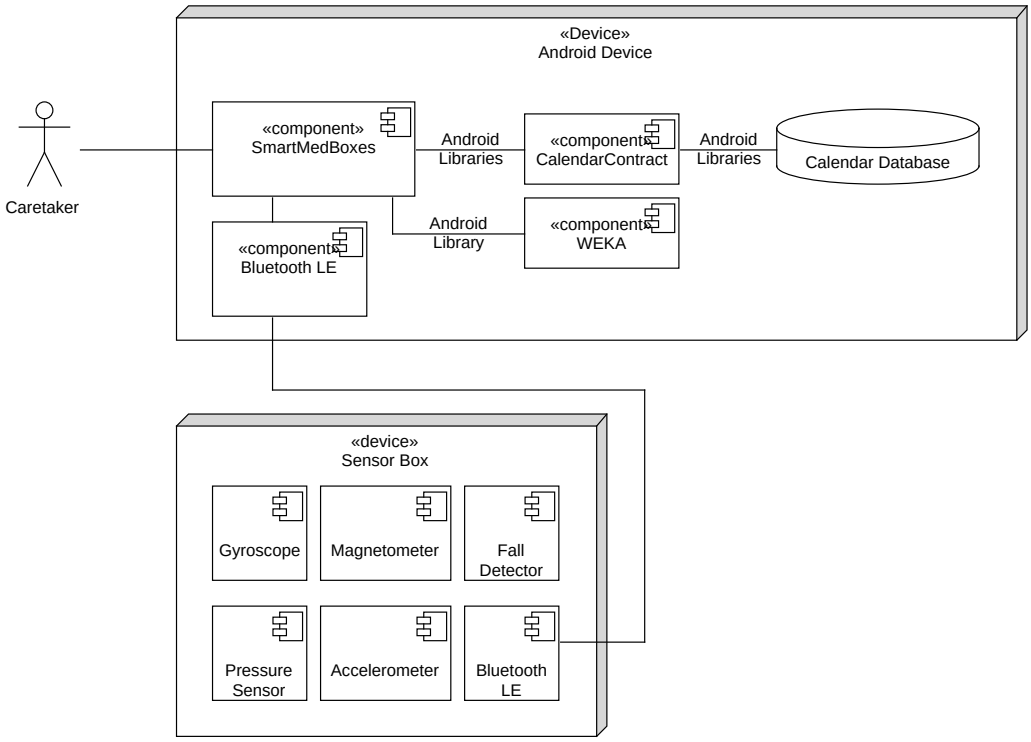

Figure 3.4: Installation diagram.

3.4.6 Android Application Mockup

For a better understanding of what is intended for the application layout and development of the prototype, a mockup was designed. The schematics in figure 3.5 represent a preliminary layout of the main screen, as well as the screen where a future medication intake is detailed. To keep the size of this chapter manageable, the full set of mockups for the application was included in the appendix A.

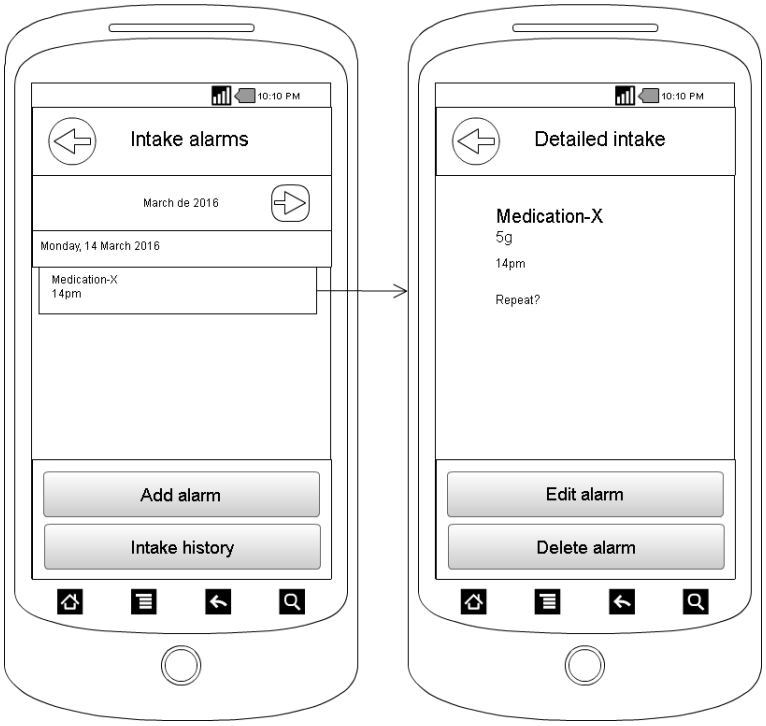

Figure 3.5: Main screen mockup.

Figure A.4 contains the mockups for the *Add Alarm* and *Add Intake Hours* functionalities. Figure A.5 has the detailed future intake functionality mockup, where one can see the *Edit Alarm* and *Delete Alarm* button. Finally, figure A.6 illustrates how the past intakes list should look like, as well as the past intake detail. An example of a notification generated by the application is still presented in this figure, as well as a question dialog.

3.5 Conclusion

This chapter describes the requirements analysis, and contains a brief description of the architecture of the system and the software engineering process. The Software Engineering subsection includes the Use Cases and its respective description, Actor Description, Activity Diagrams, Installation Diagrams and Mockups for the Android application of the system. All components analysed herein are considered the main pieces supporting the development of a software product. The correct identification of the components is fundamental to a enable a well structured development, allowing all the required features to be implemented in the system. This will allow, in the end, to assess if the final result matches the initial project.

Chapter 4

Prototype Development

4.1 Introduction

This chapter of the dissertation is focused on the prototype development phase of the proposed solution, which followed the conceptual part described in the previous chapter. The system is composed of an Android application, along with the environment where it runs, and a sensor bundle, with which the application communicates. Development (and testing) was thus mostly focused on the mobile application, which is the main subject of the next 3 sections. Section 4.2 describes the development of the Android application prototype. Then, section 4.3 describes the functioning of the application resorting to screenshots and section 4.4 provides details on tests made to the preliminary prototype.

4.2 Android Application

The Android application is one of the main outputs of this Master's project and it is composed of three main modules, all implemented in the scope of this work. The *Data Collection* module will be described in subsection 4.2.1, while subsection 4.2.2 discusses the *Classification and Feature Extraction* module. The Medication Reminder is then described in subsection 4.2.3.

A secondary Android application was developed in the scope of this project. Its purpose was to simply integrate the Pandlet library and obtain values from the sensors. It has a very basic design, because it was only used by the developer to test communications and gather data. As such, it is not further discussed herein.

4.2.1 Data Collection Module

The *Data Collection* module integrates and makes extensive use of the Pandlets library. This library was designed in a manner that makes the Bluetooth Low Energy (BLE) stack and sensor integration to be very simple and similar to the Android Sensors framework. On this proof-of-concept, the *Data Collection* module is able to retrieve data from the 3-axis accelerometer and 3-axis gyroscope sensors of the Pandlet device.

An accelerometer sensor is a Microelectromechanical System (MEMS) that measures the acceleration of the resulting forces applied to the device at a specific moment in time. In other words, an accelerometer is able to measure the gravitational acceleration and the acceleration

that is produced by any movement of the device. It can also measure the acceleration imposed to a structure upon a change in motion. The readings returned by this sensor use the meter per squared second (m/s^2) standard unit. Since the device is part of the sensor box that will be attached to the medication box, structural stress (e.g., the *click* of the opening of a compartment) imposed to the medication box will be reflected in the measurements of the sensor. Note that accelerometers are inertial sensors, which means they can be affected by calibration errors and by external noise also.

On smartphones and digital devices, a gyroscope sensor is a Microelectromechanical System (MEMS) that measures the angular velocity in respect to its three axis, x, y, and z. By reading the values of a gyroscope, one can understand the amount of rotational motion that the device is being subjected to in a given moment in time. When the sensor is static, the reading values from the three axis of the gyroscope shall be 0. Nonetheless, since a gyroscope is an inertial sensor, and similarly to the accelerometers, it can be affected by calibration errors and external background noise. The standard unit of reading from this type of sensors is radian per second (rad/s). Analogously to the accelerometer, this device is part of the sensor box attached to the medication box. It is expected that structural stress imposed to the medication box will be reflected in the sensor readings, allowing one to measure the angular velocity of the sensor box and, consequently, of the medication box.

Data collected from these sensors is used in the classification module to understand which gestures are being made with the medication box. Description of the sets of gestures considered in the scope of the Master's project are further discussed in chapter 5.

4.2.2 Classification and Feature Extractor Module

The *Classification and Feature Extractor* module consists mainly of the logic enabling the integration of the Android application with WEKA libraries and the usage of the algorithms they provide. All the available algorithms on the desktop framework are also available in this module, but their usage required the implementation of methods to adapt the output of the sensors to WEKA. As such, the module contains methods to load the training dataset, whose *features* were extracted and whose models were created on the desktop, to classify new instances that were created with the calculated *features*, and to convert data retrieved from the inertial sensors of the sensor box to the *features* needed in the classification engine.

Listing B.1, included in the appendix due to size limitations, presents the class constructor and the instance classification method for the *Classification and Feature Extractor* module. Several details are highlighted in this excerpt of code, namely that the classifier is instantiated using `SMO()` and that the `instanceClassification(String)` method guarantees that the correct format of the strings before submitting them to WEKA.

Once the method `instanceClassification(String)` is executed, the module will return a string containing the possible label of the *feature vector* that was classified with the highest probability. This output is handed out to the *Medication Reminder* module. Then, the *Medication Reminder* module will write the results of new instance classifications to a file.

4.2.3 Medication Reminder

The *Medication Reminder* module consists of the implementation of methods that enable the integration of the *CalendarContract* library into the application, which will ultimately allow a user to retrieve, add, and edit calendars and events in the system. To prevent methods from blocking the User Interface (UI), these methods were implemented using the *AsyncTask* class from Android, a class that allows the use of background tasks in different threads of the main UI thread. The *CalendarContract* Android library [Inc12] was used for the application serving as the proof-of-concept, mostly because it allows the solution to be self-contained, in terms of calendar data.

A class specially devoted to handle the medication details when using the calendars was developed. Listing B.2, in the appendix, contains an excerpt of code of such class, emphasizing its most important the variables and methods. It is possible to see that the application stores the name (`medName`), a description (`medDesc`), the dosage (`medDosage`) and the recurrence of prescription for each registered medicine, along with other details.

Due to size limitations, the code in the interior of the methods of the listing was removed, but they can be briefly described as follows. Methods `toJSON()` and `fromJSON()` are used to convert the class to a JSON string and vice-versa, and are specially useful when passing the medication details between Android activities. As Android events are not designed to handle all the information associated with a medicine, such as *Dosage*, or Universally Unique Identifier (UUID), another two methods were developed to fill this gap: `generateDescription()`, which generates a JSON string with the medication details extra fields that will be attributed to the description of the events; and `parseDescription()` which, as the name states, parses the JSON string that is retrieved from the event description. The additional details that need to be conveyed to the event description are: (i) the pair intakes-dosage, attributed to the medication reminders; (ii) the medication status, namely if it is taken, not taken, or unknown; (iii) the MAC Address from the BLE module in the Pandlet device; (iv) the recurrence; and (v), the UUID, a unique value generated for each medication added to the calendar. Note that each medication may have multiple intakes and different dosages with the same UUID.

4.3 Application Overview

Apart from the underlying logic, the system is supported by several Android activities, some of which are shown and briefly described in this section. The main objective of this section

is to provide a better insight on how the application works by showing the interfaces used to interact with the user and how the functionalities are structured. The *Future Intakes List* is the main activity of the application and it is where the user navigates into automatically when the application is started. From there, the user is able to access all the functionalities of the application. Figure 4.1 shows the main activity for the application.

In figure 4.2, one may see the screenshot of the interface for the first step of adding a new medication reminder. The previous screen is followed by the one shown in figure 4.3, depicting the second step when adding a new medication reminder. In this activity, the user is able to choose the date and the timings of the reminder via date pickers and dropdown lists. Finally, figure 4.4 shows the screen summarizing the setup of a new medication reminder, offering the option to be added to the calendar, fine-tune the intake hour or cancel the reminder. It is believed that this compartmentalized flow benefits user-friendliness. Each step of the flow is identified at the top of each screen (e.g., `Step 3/3`).

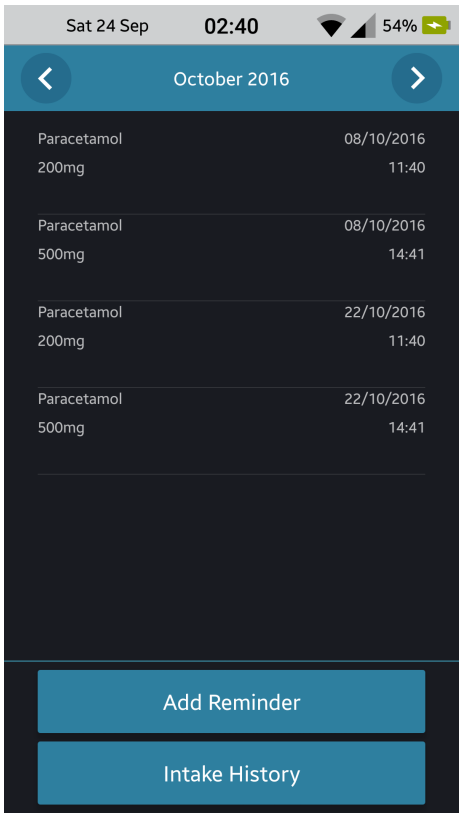

Figure 4.1: Screenshot of the first activity shown when the user starts the application.

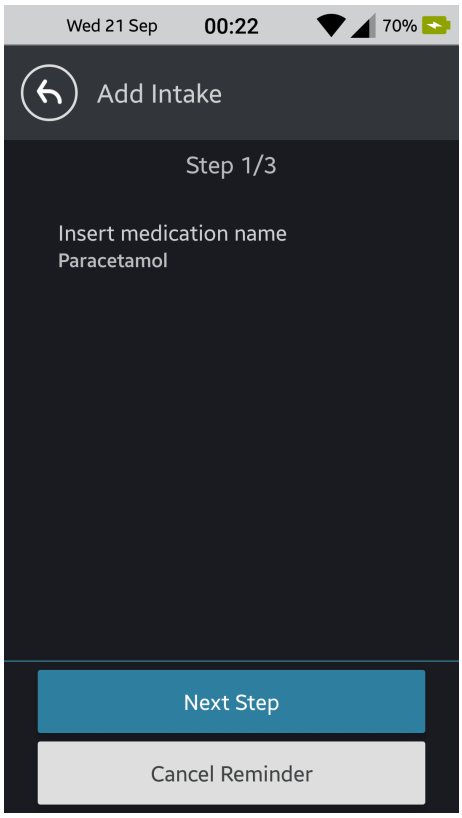

Figure 4.2: Screenshot of the application on the first step of adding a new medication reminder.

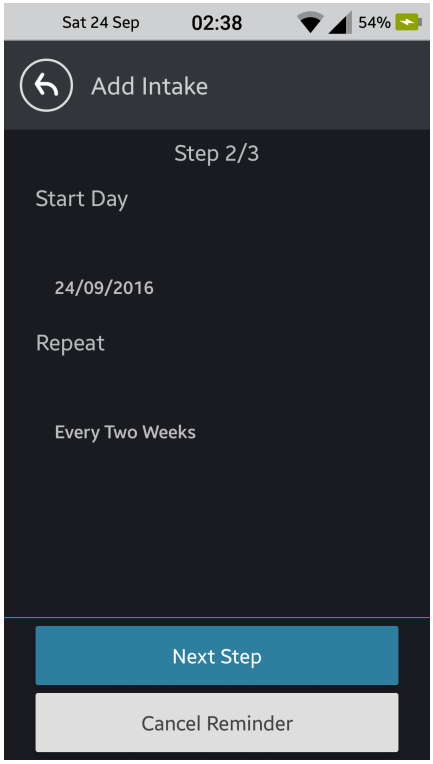

Figure 4.3: Screenshot of the application on the second step of adding a new medication reminder.

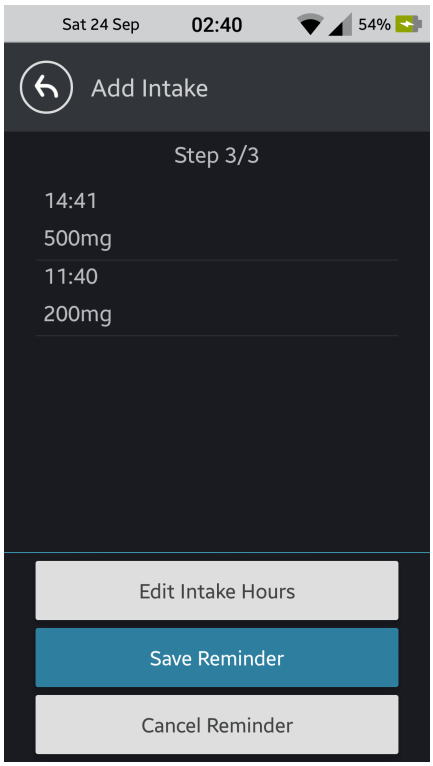

Figure 4.4: Screenshot of the application on the third step of adding a new medication reminder, after the intakes are added.

In the case the user chooses to edit the intake hour for a medication, it is presented with the screen shown in figure 4.5. As can be seen, the user is able to introduce the hour (and minutes) and the respective dosage in this screen. Different intakes can be added using this procedure. If the user wishes to add more intake timings to the list, he may choose the `Add Intake Hour` after confirming the current form, as shown in figure 4.6. Additionally, if the user wants to change one of the intakes, he may select one of the available intakes from the list.

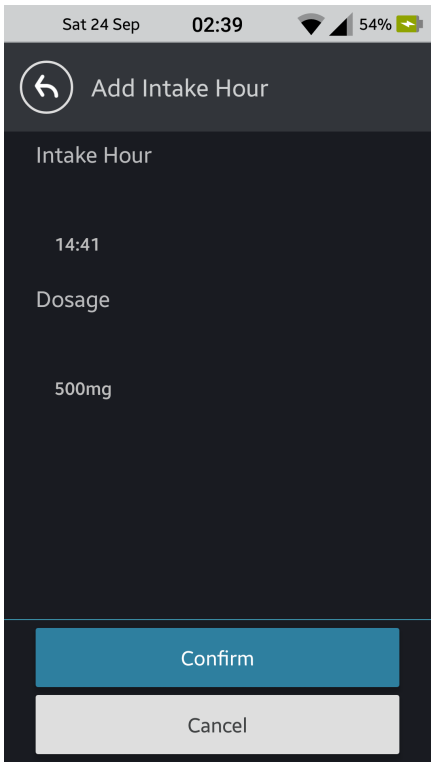

Figure 4.5: Screenshot of the application when one is adding hour and dosages to a new medication reminder.

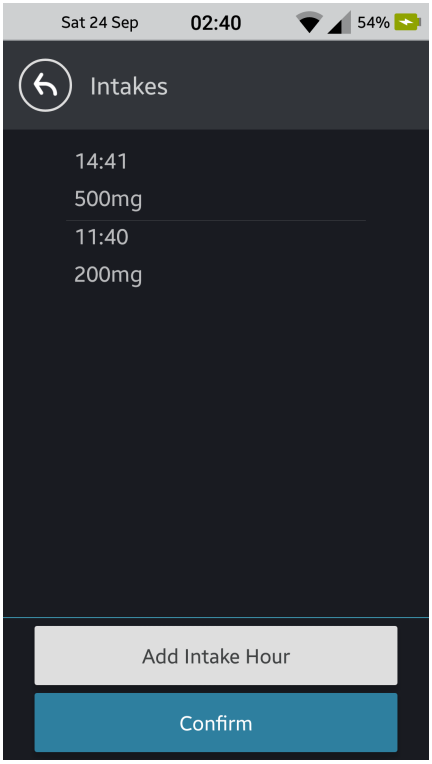

Figure 4.6: Screenshot of the application listing the intakes to be added to the medication.

Figure 4.7 presents a screenshot of a question dialog. This dialog is presented to the user when adding a new medication to the application. If one chooses *Yes*, the device will present the list of Pandlet devices that are available to connect, otherwise, the application will jump to the intake management screen. This is perhaps the best visual clue of the connection between the developed application and the medication box, as well as of the main objective of this project. At this point, the application is asking the user if it should monitor the intake via the external device or not. Additionally, the application will try to interact with the user at the intake times, by issuing alarms and actively asking if the intake has occurred, as shown in figure 4.8. When the user chooses one of the answers, the medication status will be updated.

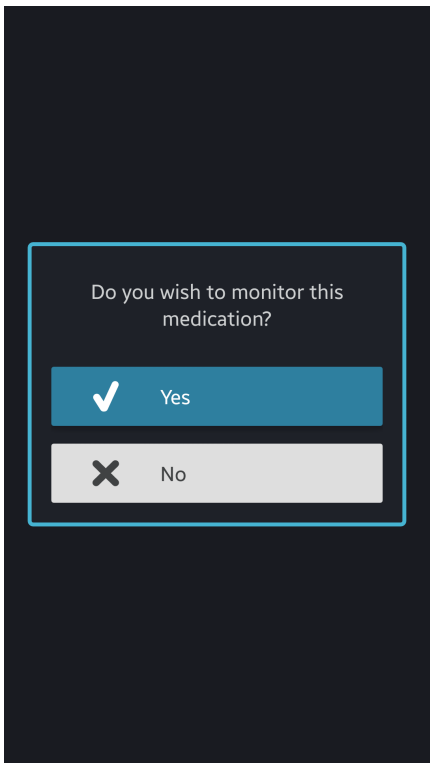

Figure 4.7: Screenshot of a dialog where the user is asked if he wants monitor a medication.

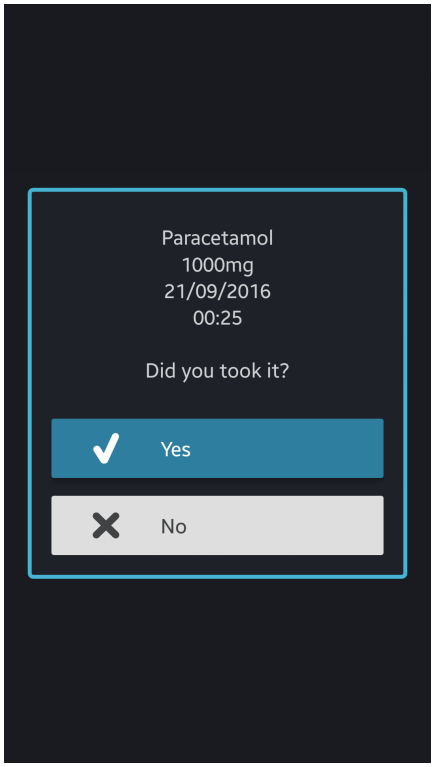

Figure 4.8: Screenshot of a dialog where the caretaker is asked if he took the medication at the time of the alarm.

4.4 Preliminary Testing of the Prototype

Apart from the unit tests performed along the implementation phase, the developed prototype was preliminarily tested in a Samsung Galaxy S4 (i9505) running Android OS 5.0.1, connected to a Pandlet device. They were mainly focused in assessing if the functionalities were functioning correctly and some possible anomalous scenarios. These tests were not exhaustive, since this was not the main focus of the work and the system was merely a prototype. Subsection 4.4.1 describes the test to the *Add Medication* functionality, while subsection 4.4.2 and subsection 4.4.3 discuss the tests to the *Edit Medication* and *Delete Medication* functionalities, respectively. The discussion that follows complements the previous description of the application in the sense that it also shows several secondary features that were added to handle anomalous scenarios.

4.4.1 Add Medication

One of the first details to be tested in the *Add Medication* functionality was the input and the contents on the field for the name of the medication. The application was implemented in such a way that, if the field is left empty, an alert is raised, preventing the user from going further into the procedure. The test showed that this safeguard is working.

Both fields of the date and recurrence type screen were tested also. When both or one of the

fields were not duly filled in, and the user pressed *Next Step*, another alert was raised, stopping the user from going to the next step. When all the fields are correctly filled and the user touches *Next Step*, a dialog asking the user to decide whether he or she wants to monitor the medication is raised. Both behaviors were expected and corroborated by the tests.

Tests also confirmed that the flow concerning the choice of monitoring the medication was also working properly. In this case, the Pandlet list appears when the user chooses to monitor the intake. In this list, it was possible to select the particular Pandlet (hence medication box) that should be monitored. In the case where the user tried to evolve to the next step without choosing any Pandlet device, an alert dialog prevented the situation, as expected. On the other hand, if the a Pandlet is chosen and the `Confirm` button is pressed, the activity with the intake list is correctly shown. After choosing a Pandlet, the devices becomes one of the options that may be associated with a given intake.

In the intake list activity, the user may either edit the intake hours or save the reminder. The situation where the user tries to save the reminder without properly configuring at least one intake hour was tested, and the application spawned an alert, as expected. In this case, the user must dismiss the alert and go into the list of intakes first, which was also tried.

When the `Add Intake` button is touched, the screen to add an hour and a respective dosage should come up. An user was asked to test this particular activity at will. For example, he did try to add the intake without having the fields duly filled up, which led to the spawning of a dialog, from where the user can only evolve to the very same screen. When the fields were correctly adjusted, the intake was successfully added to the medication reminder after pressing the `Confirm` button, which brings the user back to the intakes list activity. Once the user touched the *Save Reminder* button on that screen, the system added the events to the calendar, and then looked for the Pandlet device and started to retrieve and process data.

The application should spawn a (Android system wide) notification at the time of the intake. To test this scenario, the intake hour was set up to one of the minutes following the moment in which the user was trying the prototype. At the time of the alert, the dialog with the question *Did you take it?* came up in the screen. The LogCat – the logging module of Android – showed that the classification instances were reporting the created and classified instances. Although the classification results were not confirmed on the created file during these tests, all pointed out to that the main functionality is working.

4.4.2 Edit Medication

The second most important functionality of the application concerns the editing of medications. In order to perform tests on this feature, previous events were added to the calendar. The normal flow for such operation presumes that the user selects the medication from the main

activity, evolving to the view where the details such as timings and name of medication are shown. From there, the user can press the `Edit Reminder` to access this specific functionality.

The user tested the reaction to the modification of the name of the medicine. Like before, if the field was cleared and the form was submitted, an alert was raised as expected.

Modifying a medication follows the same step by step procedure described in the previous section. Once the name is modified (i.e., the respective field is non-empty), the application evolves to step `2/3`, where the date and recurrence type can be adjusted. The user tested different configurations for these values. If one or both fields were set to empty, then the user was not able to proceed to the next step, obtaining feedback on the problem via text message.

Step `3/3` concerns the choice of whether the user wants to monitor the medication or not. The user verified that the Pandlet list effectively appears in the affirmative case, and that he could select the medication box at that point. If no Pandlet device was chosen, a pop-up would take the foreground, preventing him from going further. Otherwise, the `Confirm` button was available, and the application changes focus to the intake list activity.

By pressing the `Edit Intake Hours` button, the same list of intakes is shown once again, but the user now has the option to add a new hour and dosage. At this point, the anomalous scenario where no values were set before pressing the `Add Intake` button was tested, with the successful outcome of not being able to proceed. A reminder was effectively added only when the fields were duly filled up.

When the user returned to the intake list activity, he pressed the *Save Reminder* button, and the system added the events to the calendar. Then, the system searched for the Pandlet device and started to retrieve and process data, as it could be seen via LogCat. Additionally, at the time of the alert, the dialog with the question *Did you took it?* came up on the screen. Although the classification results were not confirmed at this point, there were entries generated in the logs, and it was concluded that, from the user side, it all seemed to be working according to the specifications.

4.4.3 Delete Future Medication

The application offers the possibility to delete a future event and, to test the functionality, it was set up with several events in the calendar a priori. The user was then tasked with requirement to delete one. This could be achieved via the selection of the medication from the main activity where all the events are shown. A screen with the specific information is then showed, with the option to `Delete Reminder` at the bottom of the screen. As expected, after pressing that button, a dialog for confirmation was spawn. Pressing the button with the label `No` took the user to the underlying screen; pressing `Yes` led the user to the future events list, on

which it was possible to verify that the event was deleted.

4.5 Conclusion

This chapter started with a discussion of some important details concerning the modules composing the Android application and their implementation. The means used to integrate functionalities related with the sensors and with WEKA deserved special attention. The developed prototype was then presented using screenshots and by describing several tests to its main functionalities. Though it can be immediately concluded that the visual aspect of the prototype is not exactly equal to the mockups, it complies with the requirements identified during software engineering.

The prototype discussed herein allowed the testing of the solution, by implementing several classification algorithms, as discussed in the following chapter. Many choices regarding the look and feel of the prototype, as well as the underlying workflow, were influenced by the target audience, even though the application is still at a prototyping stage. The compartmentalized flow for adding or editing a medication is an example of such choices.

Chapter 5

Datasets and Analysis of the Results

5.1 Introduction

This chapter focuses on the creation of the datasets used to test the approach of using machine learning to identify gestures and also on the results obtained when trying out different algorithms on that data. As such, section 5.2 describes the gestures that need to be considered for the purpose of studying the retrieval of pills from a medication box. The means used to create the datasets are disscussed in section 5.2, along with their brief analyses. Section 5.3 describes the preprocessing that was applied to the datasets in preparation to the submission to the machine learning algorithms. A definition of *feature* is included as well, preceding the introduction of the extracted *features* for the purposes of classification. Section 5.5 contains a discussion on the results gathered after applying several machine learning algorithms to the datasets.

5.2 Gestures

Although the movement of taking a medicine from a medication box can be understood as a relatively simple gesture, it can be splitted into multiple ones. During the course of this project, it was decided that, for the generic case where a user just attaches the sensor box to a medication box, the movement should be considered as a combination of five different gestures. The gestures that were considered are the logical and intuitive steps of successfully taking a medicine out of a medication box, as follows:

- *pick the medication box*;
- *open a compartment*;
- *take the medicine out of the compartment*;
- *close the compartment*;
- *return the medication box*.

One can still consider two different (sub-)gestures for the *take(ing) the medicine out of the compartment* case, where the first gesture is described as *picking the medication with the fingers* and the second gesture as *dropping the medication on the hand*. This means that one should consider that some will try to take the medication with the fingers while others may try

to tilt the box towards the hand. One of the secondary objectives of this part of the project was to study if it would be possible to exactly discriminate which compartment was open or closed during the gesture *take the medicine out of the compartment* via the inertial sensor data. As such, data was gathered while taking that into account, by adding the information on the number of the compartment to metadata, and asking the subjects to open different compartments.

Datasets

A careful search on the Internet returned no useful results in terms of datasets with the specific characteristics needed for this work. As such, in order to test the approach of using machine learning for classification and the prototype, it was required to create datasets with medication boxes gestures.

Two different datasets were created. A regular medication box with four compartments was used while recording the inertial data from the sensor box. These boxes are usually cheap and can be bought at any local pharmacy (see Figure 5.1). The inertial data includes sensor readings from a 3-axis accelerometer and a 3-axis gyroscope. Two different placements of the sensor box in the medication box were considered for all tests. To record data from both positions, two different sensor boxes were used, as emphasized in the photo include in figure 5.1. This way, it is possible to potentially assess if the placement of the sensor box is influential for the results, and to identify which place is best to attach the sensors. Since real medication pills can not be used while retrieving data, small wrinkled papers were used instead of real pills. While recording the data, labels were used to identify each of the gestures.

Figure 5.1: Medication box and respective sensor boxes used for creating the datasets and for the tests.

For the sake of the explanation, two different names will be given to the devices attached to the medication box. The position of the sensor on the left side of the medication box will be

referred to as *position CE*, while the position of the sensor in the back of the medication box will be referred to as *position FF*. These names are given after the Media Access Control (MAC) Address of the Bluetooth Low Energy (BLE) modules of the sensor boxes.

The first dataset consists of inertial sensor data, collected at Fraunhofer Portugal AICOS office, with the help of some of its collaborators. The subjects that helped to create this dataset were five different right-hand dominant persons. The age group of the subjects ranged from 23 to 30 years old. Test subjects were told to reproduce the two different sequences of the gestures, corresponding to the intuitive order of taking the medication out of the medication box as described in section 5.2. The first sequence considers the *picking the medication with the fingers* gesture, while the second one considers the *dropping the medication to the hand* gesture. During these experiments, the test subjects left the pill on a table while the remaining part of the sequence was not over. The sequence was repeated three times per compartment of the medication box, for different sampling frequencies of the sensors. This means that each subject was asked to perform the same tasks at least six times (one per each combination of sequence of gestures with frequencies). For the sampling frequency, [SSMBC+16] states that 30 Hz is enough for capturing relevant information for human body motion. Since data is being collected from a object, three different sampling frequencies were used for this dataset: 8 Hz, 50 Hz, and 100 Hz. This way of proceeding will potentially allow to pinpoint the optimal sampling frequency to be used on the second dataset.

The second dataset consists of inertial sensor data, collected at the *Centro de Convívio para Idosos do Bonfim*, with the help of some of its users. This dataset corresponds thus to a real world scenario. The data recording sessions for this dataset were performed in the scope of *COLABORAR*, a Fraunhofer Portugal AICOS project that aims to improve the research and development of the solutions developed by the institution, by listening to opinions and challenges from the target users. Figure 5.2 presents a test subject with the medication box[1]. The subjects that helped to create this dataset are 15 different, right-hand dominant persons. The age group of the subjects ranged from 66 to 93 years old. For this test, test subjects were told to reproduce a single sequence of gestures. Each sequence was equal to one of the sequences simulated during the collection of the first dataset, with the exception of the step where the subjects take the medicine from the medication box. Since this subject group is a small sample of the target users, they were asked to do the gesture sequence in the way they feel to be natural. The idea was to obtain data accurately representing a real world scenario and also to test the algorithms against the data. Notice that, at the moment of collection of this dataset, tests had already been made to the first dataset. Because of that, each sequence was only reproduced two times: a first one, using the sampling frequency of 50 Hz, and a second one, using the sampling frequency of 100 Hz.

[1]This photo was taken and included here with her consent.

Figure 5.2: Data recording at *Centro de Convívio para Idosos do Bonfim*.

Notice that the datasets were created in such a way that they allow studying if it is possible to discriminate which compartment of the medication box is opened or closed, or to simply identify if one of the compartments was interacted with, without precisely knowing which one. Furthermore, some of the test subjects of the second dataset were not able to reproduce the *pick the medication box* and *return the medication box* gestures, during the collection of the inertial data. As such, some of the tests had to be made while considering these two gestures, while others did not.

5.3 Preprocessing

Data preprocessing is a common step within the scope of the application of machine learning techniques. It allows converting the original raw data into a representation that the algorithms are able to handle. Several methods may apply during this stage, depending on the data and algorithm to be applied. Data Normalisation, Feature Extraction, and Noise Reduction are some of the preprocessing steps that can be applied to the data samples. Data used in this project was subdued to Data Normalisation prior to the Feature Extraction step.

Data Normalisation consisted, in this case, on scaling the sensor data, by finding the maximum values in the datasets and dividing the respective readings by the absolute value of those. This procedure preserves the waveform of the signal in a different interval, namely $[-1; 1]$.

On machine learning algorithms, the term *feature*, as stated by [KP98], is the specification of an attribute and its corresponding value. The main purpose of features is to describe a sample through categorical features, through a finite number of discrete values, or through continuous values, which are usually a subset of real numbers. Machine learning algorithms then use sets of *features*, called *feature vectors*, to process and classify data.

As the specialized literature is not very rich in terms of how to extract *features* from data retrieved specifically from sensor boxes attached to objects, it was decided to follow some of the suggestions in [FDFC10] and [KWM11]. The features used in the scope of this work are described below. Some of them are exclusive to the accelerometer sensor and, as such, they were not extracted from the data concerning the gyroscope. Feature vectors were built in three different ways: (i) the first ones concern the accelerometer data only; (ii) the second ones concern the gyroscope data exclusively; (iii) and the third one, where the data from both sensors is combined by joining the feature vectors of the same gesture sample.

The list of features extracted from the data can be listed and described as follows:

- The average ($\bar{X}$) and standard deviation (σ), which have minimal processing and memory requirements, were calculated using the sensor data retrieved on the gesture time-window using the well-known formulas in equations (5.1) and (5.2):

$$\bar{X} = \frac{\sum_{i=0}^{n} x_i}{n}, \text{and} \quad (5.1)$$

$$\sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}, \quad (5.2)$$

  where x_i represents a sample value;

- The Root Mean Square (RMS) of the signals, is a statistical metric, which is normally associated with meaningful context information, that can be obtained via the formula in equation (5.3), where n denotes the number of samples used in the calculation:

$$x_{RMS} = \sqrt{\frac{\sum_{i=0}^{n} x_i^2}{n}}. \quad (5.3)$$

- The Pearson's product-moment coefficient, also known as sample correlation coefficient, which is calculated as the ratio between the covariance of the signals along the x-axis and the y-axis to the product of their standard deviations (equation 5.4):

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}; \quad (5.4)$$

- The Signal Magnitude Area (SMA) metric, also referred to as the *energy expenditure* in activities, is usually used to distinguish between a resting state and an active state and can be calculated using the formula in equation 5.5:

$$x_{SMA} = \frac{1}{t}\left(\int_0^t |x(t)|dt + \int_0^t |y(t)|dt + \int_0^t |z(t)|dt\right); \quad (5.5)$$

- The Signal Vector Magnitude (SVM) 5.6, sometimes referred to as Signal Magnitude Vector

to avoid confusion with the Support Vector Machines (SVM) algorithm, indicates the degree of the intensity of movements and it is calculated as shown by equation 5.6:

$$x_{SVM} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{x_i^2 + y_i^2 + z_i^2}; \quad (5.6)$$

- The energy and power of the signal for each axis, which are typically defined by equations 5.7 and 5.8:

$$x_{energy} = \sum_{i=0}^{n} x_n^2, \quad (5.7)$$

$$x_{power} = \frac{\sum_{i=0}^{n} x_n^2}{n}. \quad (5.8)$$

In addition to these features, the procedure to obtain *binned distributions* was also applied to the data. In order to do so, the range of each axis (which was defined as ranging between -1 and 1) was split in ten parts (the bins). This procedure allows obtaining a more precise idea of the fraction of the values that fell within each bin.

5.4 Classification Algorithms

There is a wide panoply of machine learning algorithms to choose from when considering classification of data. In 2010, the authors of [ABT10] compared different techniques for classifying different human activities that were performed using body-worn miniature inertial sensors. Some of the algorithms that were tested in this comparative study were Bayesian Decision Making (BDM), Rule Based Algorithm (RBA), Dynamic Time Warping (DTW), k-Nearest Neighbor (kNN), Support Vector Machines (SVM), and Artificial Neural Network (ANN). They ended up with the conclusion that BDM and kNN achieved higher true positive rates.

Another comparative study of classification techniques for human activity using body-worn miniature inertial sensors and a magnetic sensor is described in [YB11], in which *J48* (and other tree-based algorithms), Naïve-Bayes (NB), Support Vector Machines (SVM), and Artificial Neural Network (ANN) were tested. It is stated in the study that some of the best results were achieved by the ANN and SVM algorithms.

The choice of which machine learning algorithms were to be tested in the scope of this project was mostly influenced by the two aforementioned works and by the knowledge on the area, since no works focusing specifically on the application of the algorithms on data concerning gestures of objects was found. All algorithms that were tested were readily available in the WEKA framework, which basically permitted to abstract from the underlying implementation and issues. The list of four algorithms is as follows: J48, an open-source implementation of the C4.5

decision tree classification algorithm; Sequential Minimal Optimisation (SMO), a faster algorithm for training Support Vector Machines (SVM); Naïve-Bayes (NB), a statistic-based classifier; and k-Nearest Neighbor (kNN), a Euclidean Distance based classifier.

The algorithms were tested using the different datasets described above and for different combinations of the involved variables. For example, different positions of the sensor in the medication box were tested, as well as different sampling frequencies. The 8 Hz sampling frequency was exclusively tested for the first dataset, and the results were used to better understand which was the optimal sampling frequency to be used.

5.5 Results

The results obtained during the testing of the several algorithms applied to the datasets are presented in this section. As previously mentioned, four algorithms were tested for several gestures and sampling frequencies, generating a large diversity of results. k-Fold Cross-Validation and Split Percentage were used for each one of the different datasets with the default values on the WEKA framework. These values correspond to $k = 10$, for the k-Fold Cross Validation, and to a Split Percentage of 66% for the training set and 33% for test set.

First Dataset

Some of the gestures from the sensor boxes for the dataset collected at Fraunhofer AICOS had to be ruled out of the tests. This happened mainly because Pandlets may have lost connection with the mobile device when data was being collected, consequently corrupting some of the data samples from the gestures. This led to an unbalanced dataset, where some gestures have more samples than others. There would always be more samples of the *pick the medication box*, *take the medicine out of the compartment*, and *return the medication box* gestures, even if the aforementioned gestures had not been removed. This happens because these gestures are repeated and recorded each time one of the compartments are opened or closed. E.g., collecting two samples from opening the first compartment of the medication box and another two from the second compartment, corresponds to four samples of the *pick the medication box*, *take the medicine out of the compartment*, and *return the medication box* gestures. For this reason, the following tests were made either when trying to discriminate which of the compartment was opened and closed in the medication box, or when the compartment was not being discriminated:

- unbalanced dataset, only considering the *dropping the medication to the hand*;
- unbalanced dataset, only considering the *picking the medication with the fingers*;
- unbalanced dataset, considering both *picking the medication with the fingers* and *dropping the medication to the hand*;

- balanced dataset, only considering the *picking the medication with the fingers*;

- balanced dataset, only considering the *dropping the medication to the hand*;

- balanced dataset, considering both *picking the medication with the fingers* and *dropping the medication to the hand*.

Given the above division of the dataset, twelve different tests were made with the four algorithms: six using Cross-Validation and another six using Split Percentage. These twelve tests were performed two times. In the first run, all gestures were considered. In the second run, the gestures *pick the medication box* and *return the medication box* were not considered.

To keep the description shorter, the names *Drop*, *Pick*, and *Both* are used to refer to the gestures *dropping the medication to the hand*, *picking the medication with the fingers*, and both, respectively, in the following tables. Due to space limitations, a table containing all results obtained during this phase of the project is provided by request or as an external annex to the digital version of the dissertation. The most promising results are presented and discussed in this section. An explanation over the worst results and the behavior of the algorithms is also included below.

The first set of tests considered all gestures performed by the subjects, so as to obtain an idea on how the algorithms behave with all gestures. Table 5.1 presents the best results for the case where the compartment is being discriminated with the unbalanced dataset.

Table 5.1: Best results when discriminating the compartment with unbalanced dataset and considering all gestures.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	SMO	100 Hz	79	68,70%	32	82,05%
Pick	CE	Acc. + Gyr.	SMO	100 Hz	128	71,11%	43	70,49%
Both	CE	Acc. + Gyr.	SMO	100 Hz	250	70,42%	82	67,77%

While discriminating the compartment, best results were achieved by the SMO algorithm, using data samples gathered at a sampling frequency of 100 Hz and using the Accelerometer and Gyroscope data combined.

Table 5.2 presents the best results for the case where the compartment is being discriminated with the balanced dataset.

Table 5.2: Best results when discriminating the compartment with balanced dataset and considering all gestures.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	kNN	50 Hz	48	62,34%	16	61,54%
Pick	FF	Acc. + Gyr.	SMO	100 Hz	30	42,25%	13	54,17%
Both	FF	Acc. + Gyr.	kNN	50 Hz	83	57,64%	28	57,14%

In this case, the best results when trying to classify data from the balanced dataset are given by the Accelerometer and Gyroscope data using kNN, with a sampling frequency of 50 Hz, and SMO with a sampling frequency of 100 Hz.

The optimal results for the case where the compartments of the medication boxes are not discriminated and the dataset is unbalanced are presented in table 5.3. One can notice that, when the Accelerometer and Gyroscope data was combined, the best performance was achieved by the SMO algorithm with varying sampling frequencies, namely 50 Hz and 100 Hz.

Table 5.3: Best results when non-discriminating the compartment with unbalanced dataset and considering all gestures.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	SMO	50 Hz	200	85,11%	71	88,75%
Pick	FF	Acc. + Gyr.	SMO	100 Hz	158	87,78%	56	91,80%
Both	CE	Acc. + Gyr.	SMO	100 Hz	295	83,10%	101	83,47%

The best results for the case where the dataset is not discriminating the compartment of the medication box and has a balanced number of samples are presented in table 5.4. The highest true positive rate was achieved by SMO, with a value of $95,74\%$.

Table 5.4: Best results when non-discriminating the compartment with balanced dataset and considering all gestures.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	SMO	50 Hz	95	95,00%	31	91,1%
Pick	FF	Acc. + Gyr.	SMO	50 Hz	92	92%	32	94,12%
Both	FF	Acc. + Gyr.	SMO	50 Hz	128	92,75%	45	95,74%

The Naïve-Bayes (NB) algorithm was the algorithm with the worst behavior in the tests where all the gestures were being considered, achieving a maximum value of true positives rate of 64.71% and a minimum value of 6.25%. The *J48* algorithm behaved slightly better, achieving a maximum value of 85.51% and a minimum value of 7.69%. k-Nearest Neighbor (kNN) comes next, achieving a maximum value of 91.5% and a minimum value of 11.5% and, finally, Sequential Minimal Optimisation (SMO) has the best results, achieving a maximum value of $95,74\%$ and a minimum value of 15.38% success rate (not shown).

On the second run of the tests, the two gestures *pick the medication box* and *return the medication box* were not considered, in order to understand if, by removing them, there were any improvements in the behavior of the algorithms. Table 5.5 presents the best results for the case where the compartment is being discriminated and the dataset is unbalanced. As can be concluded from the analysis of the table, best results were given by the SMO algorithm when using a sampling frequency of 100 Hz and when the sensor box was in the *CE* position.

Table 5.5: Best results when discriminating the compartment with unbalanced dataset, not considering all gestures.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	CE	Acc. + Gyr.	SMO	100 Hz	70	66,67%	22	61,11%
Pick	CE	Acc. + Gyr.	SMO	100 Hz	59	54,63%	21	56,76%
Both	CE	Acc. + Gyr.	SMO	100 Hz	129	60,56%	42	58,33%

The best results for the case where the compartment is being discriminated and the dataset is balanced are presented in table 5.6. Contrarily to the previous case, the best results were obtained for 50 Hz sampling frequency (and not for the 100 Hz sampling frequency), while the best position of the sensor box has changed to *FF*. Nonetheless, the best algorithm was, once again, SMO.

Table 5.6: Best results when discriminating the compartment with balanced dataset, not considering all gestures.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	SMO	100 Hz	20	37,74%	11	61,11%
Pick	FF	Acc. + Gyr.	SMO	50 Hz	32	50,79%	9	42,86%
Both	FF	Acc. + Gyr.	SMO	50 Hz	61	50,83%	23	56,10%

Table 5.7 presents the best results obtained when the compartments of the medication boxes were not discriminated and the dataset was unbalanced. The *Drop* gesture is better classified than the *Pick* gesture. The algorithm achieving the best results was SMO.

Table 5.7: Best results when non-discriminating the compartment with unbalanced dataset.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	SMO	50 Hz	130	92,20%	47	97,92%
Pick	FF	Acc. + Gyr.	SMO	100 Hz	64	85,33%	22	88,00%
Both	FF	Acc. + Gyr.	SMO	100 Hz	129	89,58%	41	83,67%

Finally, table 5.8 summarizes the best results obtained for when the compartments of the medication box were not being discriminated and the dataset had a balanced number of samples. It emphasizes that the best classification ratios are obtained when the sampling rate is of 50 Hz and the algorithm is SMO.

Table 5.8: Best results when non-discriminating the compartment with balanced dataset.

Considered Gesture	Position	Sensors	Algorithm	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Drop	FF	Acc. + Gyr.	SMO	50 Hz	58	96,67%	18	90,00%
Pick	FF	Acc. + Gyr.	SMO	50 Hz	55	91,67%	18	90,00%
Both	FF	Acc. + Gyr.	SMO	50 Hz	87	94,57%	29	93,55%

Generally speaking, Naïve-Bayes (NB) is the algorithm with the worst results when the gestures *pick the medication box* and *return the medication box* are not considered, achieving a maximum true positives rate of 70.97%. The *J48* algorithm has a slightly better behavior than Naïve-Bayes (NB), achieving a maximum value of 85.5%. k-Nearest Neighbor (kNN) comes second, achieving a maximum true positives rate of 91.5% and, finally, Sequential Minimal Optimisation (SMO) presents best results by achieving a maximum true positives rate of 97.9%. It must be said that all four algorithms achieved a minimal true positive rate value of 3.23% in at least one test during these experiments. This happened mostly on the unbalanced subset when trying to discriminate the compartments and when only the *Pick* gesture was considered. For example, this value was reached in cases where the box was at *position CE* for the 8 Hz sampling rate and when using the split percentage for validation.

Second Dataset

At the time the second dataset was collected, problems related with communications with the sensor box were solved. As such, all samples gathered at *Centro de Convívio para Idosos do Bonfim* were considered. This dataset did not have the same problem as the first, where data might have been lost. Because of that, there is no unbalanced version of the dataset for the case where the compartment of the medication box is not discriminated. The following tests were made either when trying to discriminate which of the compartment was opened and closed in the medication box, or when the compartment was not being discriminated:

- discriminating, unbalanced dataset;
- discriminating, balanced dataset;
- non-discriminating, balanced dataset.

In total, six different tests were made with the four algorithms: three using Cross-Validation and another three using Split Percentage. As before, and for the sake of the explanation, only the most promising results are presented in this section.

Table 5.9 presents the best results for the case where all the gestures were considered. The most promising results from the case where the compartments were discriminated in the balanced dataset were given by the gyroscope sensor with a sampling rate of 100 Hz, on the *position CE* and

using the SMO algorithm. On the other hand, the best results for the unbalanced dataset were given by the *J48* the algorithm while combining the accelerometer and gyroscope sensors and with a sampling rate of 50 Hz on *position FF*. And, by last, when not discriminating which of the compartments was open or closed, the best result was given by combining the accelerometer and gyroscope sensors with a sampling rate of 50 Hz, on the *position FF* and using the SMO algorithm.

Table 5.9: Best results for the second dataset when considering all gestures.

	Position	Sensors	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Unbalanced (Discriminating)	CE	Gyro	100HZ	71	35,50%	29	42,65%
Balanced (Discriminating)	FF	AG	50HZ	23	20,91%	3	8,11%
Balanced (Non-Discriminating)	CE	AG	50HZ	111	55,50%	29	42,65%

Table 5.10 presents the best results for the case where the *pick the medication box* and *return the medication box* gestures were not considered, for both the unbalanced and balanced subsets and also for when the compartments were discriminated or not. Results from the case where one is discriminating which of the compartments was open or closed in the balanced dataset was given by the gyroscope sensor with a sampling rate of 100 Hz, on *position CE*. On the unbalanced subset, the best result was obtained when combining the accelerometer and gyroscope sensors with a sampling rate of 50 Hz, also on *position CE*. When discrimination of the compartments was not being considered, the best result was also obtained when combining the accelerometer and gyroscope sensors with a sampling rate of 50 Hz, though on *position FF*. The aforementioned results were all obtained by using the SMO algorithm.

Table 5.10: Best results for the second dataset when the *pick the medication box* and *return the medication box* gestures were not considered.

	Position	Sensors	Sampling Freq.	Correct Inst.	10 Fold	Correct Inst.	Percent. Split
Unbalanced (Discriminating)	CE	Acc. + Gyr.	50 Hz	69	38,0%	22	36,07%
Balanced (Discriminating)	CE	Gyroscope	50 Hz	7	22,58%	11	12,22%
Balanced (Non-Discriminating)	CE	Acc. + Gyr.	50 Hz	166	55,33%	44	43%

Contrary to the first dataset, all the results given by the algorithms were extremely low and not useful at all. This may have been caused by several different factors, such as the fact that the first dataset was collected in a controlled ambient while the second dataset was created in a ambient with real target users; differences in the way each subject made the gestures; the differences in the dataset sizes; the possible existence of sensor calibration problems; the

difference in duration of each sample; and the possible existence of delay on the transmission of data via the BLE stack.

5.6 Conclusion

A significant slice of the work conducted during the course of this project is described in this chapter. It presents how the datasets to analyze the application of machine learning algorithms to classify gestures on medication boxes were collected and used, as well as the respective experiments and *features* used in that analysis. This part of the work contributed to one of the major challenges of this Master's project.

Four different machine learning algorithms were tested against the datasets, involving different situations and scenarios. At the end, a total of 3456 different results for the *Fraunhofer AICOS* dataset were generated, while for the *Centro de Convívio para Idosos do Bonfim* dataset, 576 different results were obtained. They enabled drawing some important conclusions regarding the work at hands. For example, it was clear that Sequential Minimal Optimisation (SMO) was the best algorithm, and that it should be considered in future embodiments of this work. On the other hand, the worst algorithm was Naïve-Bayes (NB). It also became clear that the 50 Hz and 100 Hz sampling rates were better than the 8 Hz. Most of all, and specially the ones obtained for the second dataset, these results revealed the true dimension of the challenge underlying this work. The results for the second dataset were worst than for the first one, emphasizing that some work is still necessary before the application is ready for general release. A true positives rate below 95% is considered not optimal, since one is dealing with medication boxes that may have critical medication on it.

Chapter 6

Conclusion and Future Work

This chapter contains the final remarks of the work described in this dissertation (section 6.1). Potential directions for future work regarding the sensor data classification and the Android application will also be discussed in section 6.2.

6.1 Main Conclusions

The specialized literature on the area emphasizes that low medication adherence is still a problem among patients with a prescription. The work presented along this dissertation is an attempt to minimize that problem, by facilitating the medication intakes monitoring via a smartphone application that connects to a sensor box attached to medication boxes. The main idea was for the monitoring to be non-intrusive and for the application to estimate intakes from gestures on the box. Apart from the development of the application, it was also necessary to conduct research on the means by which such estimation would be possible, namely via machine learning.

The work evolved to the definition of the gestures that could be possibly be associated with the activity of taking a pill, and to the creation of datasets for analysis of the approach. The datasets were constituted by data obtained from the sensors attached to a common medication box. They were collected in two different occasions, in a controlled and in a non-controlled scenario. The subjects of the non-controlled scenario correspond to the target users. It was possible to recognize different gestures using machine learning, though there are many different variables and factors that need to be considered. It can be concluded that this study is still paving the way for a more complete solution to the problem.

Several positions of the sensors over the medication box and sampling rates were tried out against four different machine learning algorithms. Generally speaking, one can notice that the *FF position* achieved better results than the *CE position*. The results showed that the SMO algorithm had a better performance when compared to the remaining ones, and that Naïve-Bayes (NB) was the worst performing algorithm. The true positive rates were worse when trying to discriminate the exact compartment of the medication box that was accessed, when compared with the case where that discrimination was not done. Moreover, the true positive rates were also higher when the *pick the medication box* and *return the medication box* were not considered (i.e., classification improves when only the intermediate gestures are considered).

The results for the tests performed over the dataset collected at Frauhnofer AICOS were significantly better than the ones obtained for the dataset collected at the *Centro de Convívio para Idosos do Bonfim*. This showed the importance of conducting tests in non-controlled environments. There are several possible factors that can justify these results, such as:

- differences in the way the subject handles the box (fundamental differences in the gestures);
- the differences in the dataset sizes;
- the possible existence of sensor calibration problems;
- the difference in the duration of each sample;
- the possible existence of delay on the transmission of data via the BLE stack.

The implementation of a proof-of-concept prototype that allows for the identification of gestures on an Android OS device was one of the main objectives of this work. The basic functionalities and modules of the application were successfully implemented, namely the integration of the WEKA library in the app. Another objective was the implementation of a functionality for creating reports of the medication events in the system. While developing the prototype, special attention was given to the *caretaker* part and, mostly because of time limitations, this objective was not entirely achieved. The objective of detecting if a patient took the medication on time was not fully achieved, though the underlying study conducted with that purpose in mind fully compensates for that, in the opinion of the author. Although the medication reminder has the functionality to identify the gestures, it needs to be improved so that the result is presented to the user of the prototype. The functionality is nonetheless operational and it is saving all the results from the identification in the form of a text file on the device. The intake of a medication can be signaled manually to the application.

6.2 Future Work

Even though a lot of efforts have been applied to pattern recognition and machine learning in the last few years, there is still plenty of ground to cover on this area. While the objectives of the dissertation were clearly optimistic (as they should be), experiments in real-world scenarios provided clear indication on the difficulty of the problem at hands. This section contains potential lines of future work that may be pursued after the end of this project.

The Android application, being a proof-of-concept, can be further polished into a final application that includes all the planned features identified in the software engineering. This includes notifications for when the user is taking a medication on an unscheduled hour, as well as notifications when the user did not take a medication at a given time, and the functionality for a

caregiver to generate a report. Additionally, the usage of the variables is not optimal and can be improved.

As the application is a proof-of-concept, there are some implementation details that can be further optimized. For example, when an *Activity* is about to change, the *MedicationDetails* object, containing the details of the medication at the moment, is converted to JSON and then passed via an *Intent* to the new *Activity*. The new *Activity* will get and parse the JSON object from the *Intent* to *MedicationDetails* object back again. These steps may affect the performance of the application, since the *Activity* classes should merely deal with user inputs and present information to the user. This may be avoided in the future by using the *Application* class.

A state machine can be implemented in the Android application during the sensing of data to improve the identification of the gestures. This may avoid some unwanted or nonsensical results, by enforcing a certain order of gestures for taking a medication out of the box. E.g., a compartment can not be closed without being opened before.

The size of the datasets should be increased. Furthermore, the placement of several sensor boxes at on the medication box during the collection of data, for efficiency and testing purposes. Larger datasets will most likely lead to better results. Moreover, the datasets should be primarily obtained using target users as the subjects.

As emphasized by the discussion in chapter 5, the number of variables and conditions considered in the testing of the algorithms was large. Nonetheless, future works may consider trying other algorithms and combination of parameters, by building on some of the conclusions of this study. For example, future iterations should not need to consider the sampling frequency of 8 Hz, which consistently led to worst results. Additional sensors, such as heat or pressure, may be included in the equation. Moreover, since many machine learning algorithms allow fine tunning of the parameters, their optimization should also be considered. Feature engineering is a process that enables the identification of which of the extracted *features* will allow the machine learning algorithm to obtain better results. This process was not applied in the context of this work and may help improve the result as well.

Bibliography

[AAG+12] Brianna Abbey, Anahita Alipour, Logan Gilmour, Christopher Camp, Crys Hofer, Robert Lederer, Greig Rasmussen, Lili Liu, Ioanis Nikolaidis, Eleni Stroulia, et al. A remotely programmable smart pillbox for enhancing medication adherence. In *2012 25th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 1–4. IEEE, 2012. 11

[ABT10] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010. 40

[AOM+11] Daisuke Asai, Jarrod Orszulak, Richard Myrick, Chaiwoo Lee, Joseph F Coughlin, and Olivier L De Weck. Context-aware reminder system to support medication compliance. In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3213–3218. IEEE, 2011. vii, ix, 1, 7, 10

[atUoW11] Machine Learning Group at the University of Waikato. Weka 3 - data mining with open source machine learning software in java [online]. 2011. Available from: `http://www.cs.waikato.ac.nz/ml/weka/` [cited August 2016]. 15

[CDF+15] C Crema, A Depari, A Flammini, M Lavarini, E Sisinni, and A Vezzoli. A smartphone-enhanced pill-dispenser providing patient identification and in-take recognition. In *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 484–489. IEEE, 2015. 1, 10

[DM14] Dana DeMeo and Michael Morena. Medication adherence using a smart pill bottle. In *2014 11th International Conference & Expo on Emerging Technologies for a Smarter World (CEWIT)*, pages 1–4. IEEE, 2014. ix, 1, 8

[Far99] Kevin C Farmer. Methods for measuring and monitoring medication regimen adherence in clinical trials and clinical practice. *Clinical therapeutics*, 21(6):1074–1090, 1999. vii, 3, 7

[FDFC10] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and Joao MP Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010. 39

[fPC12] Institut für Pervasive Computer. Pervasive computing infrastructure [online]. 2012. Available from: `https://www.pervasive.jku.at/Teaching/lvaInfo.php?key=346&do=uebungen` [cited August 2016]. 15

[HRM+15] Hooman Hafezi, Timothy L Robertson, Greg D Moon, Kit-Yee Au-Yeung, Mark J Zdeblick, and George M Savage. An ingestible sensor for measuring medication adherence. *IEEE Transactions on Biomedical Engineering*, 62(1):99–109, 2015. vii, 2, 12

[Inc11] Google Inc. Android studio - the official ide for android [online]. 2011. Available from: `https://developer.android.com/studio/index.html` [cited August 2016]. 14

[Inc12] Google Inc. Calendarcontract | android developers [online]. 2012. Available from: `https://developer.android.com/reference/android/provider/CalendarContract.html` [cited September 2016]. 27

[Int13] Ecma International. Json [online]. 2013. Available from: `http://www.json.org/` [cited August 2016]. 14

[KALS15] Haik Kalantarian, Nabil Alshurafa, Tuan Le, and Majid Sarrafzadeh. Non-invasive detection of medication adherence using a digital smart necklace. In *Proc. IEEE PerCom, Smart Environ. Workshop*, 2015. 12

[KP98] Ron Kohavi and Foster Provost. Glossary of terms. *Machine Learning*, 30(2-3):271–274, 1998. 38

[KST11] Misha Kay, Jonathan Santos, and Marina Takane. mhealth: New horizons for health through mobile technologies. *World Health Organization*, 64(7):66–71, 2011. 8

[KWM11] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011. 39

[LPNF14] Jiajia Li, Shaun J Peplinski, Sarah Mostafa Nia, and Aydin Farajidavar. An interoperable pillbox system for smart medication adherence. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1386–1389. IEEE, 2014. ix, 13

[PTC14] Zhibo Pang, Junzhe Tian, and Qiang Chen. Intelligent packaging and intelligent medicine box for medication management towards the internet-of-things. In *2014 16th International Conference on Advanced Communication Technology (ICACT)*, pages 352–360. IEEE, 2014. ix, 1, 9

[Sab03] Eduardo Sabaté. *Adherence to long-term therapies: evidence for action*. World Health Organization, 2003. 1

[SLBM+13] Bruno M. Silva, Ivo M. Lopes, Mickael B. Marques, Joel J. P. C. Rodrigues, and Mario L. Proença. A mobile health application for outpatients medication management.

In *2013 IEEE International Conference on Communications (ICC)*, pages 4389–4393. IEEE, 2013. 9

[SN14] Takumi Suzuki and Yasushi Nakauchi. A smartphone mediated portable intelligent medicine case for medication management support. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3642–3645. IEEE, 2014. 12

[SSMBC+16] Rubén San-Segundo, Juan Manuel Montero, Roberto Barra-Chicote, Fernando Fernández, and José Manuel Pardo. Feature extraction from smartphone inertial signals for human activity segmentation. *Signal Processing*, 120:359–372, 2016. 37

[VHVRD01] Etienne Vermeire, Hilary Hearnshaw, Paul Van Royen, and Joke Denekens. Patient adherence to treatment: three decades of research. a comprehensive review. *Journal of clinical pharmacy and therapeutics*, 26(5):331–342, 2001. 2

[VLvW+12] Marcia Vervloet, Annemiek J Linn, Julia CM van Weert, Dinny H De Bakker, Marcel L Bouvy, and Liset Van Dijk. The effectiveness of interventions using electronic reminders to improve adherence to chronic medication: a systematic review of the literature. *Journal of the American Medical Informatics Association*, 19(5):696–704, 2012. 8

[YB11] Murat Cihan Yüksek and Billur Barshan. Human activity classification with miniature inertial and magnetic sensor signals. In *2011 19th European Signal Processing Conference*, pages 956–960. IEEE, 2011. 40

Appendix A

Software Engineering Diagrams

This appendix contains some of the Software Engineering diagrams developed along this work. These diagrams are mentioned in chapter 3, complementing the description therein contained. The appendix is divided as follows. Section A.1 contains the activity diagrams for the Android application prototype that was developed within the scope of this project. Section A.2 includes some of the mockups designed for the Android prototype.

A.1 Activity Diagrams

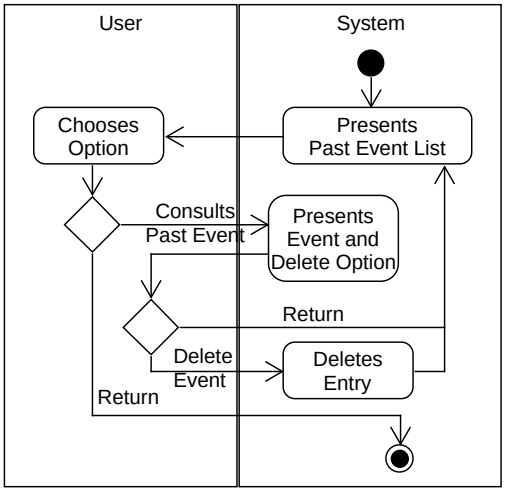

Figure A.1: *Consult History* activity diagram.

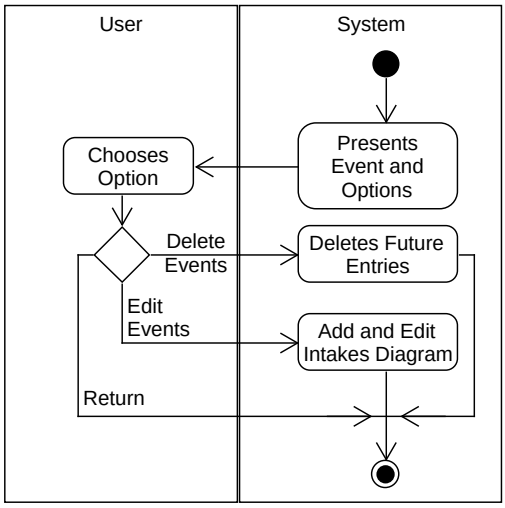

Figure A.2: *Consult Future Intake* activity diagram.

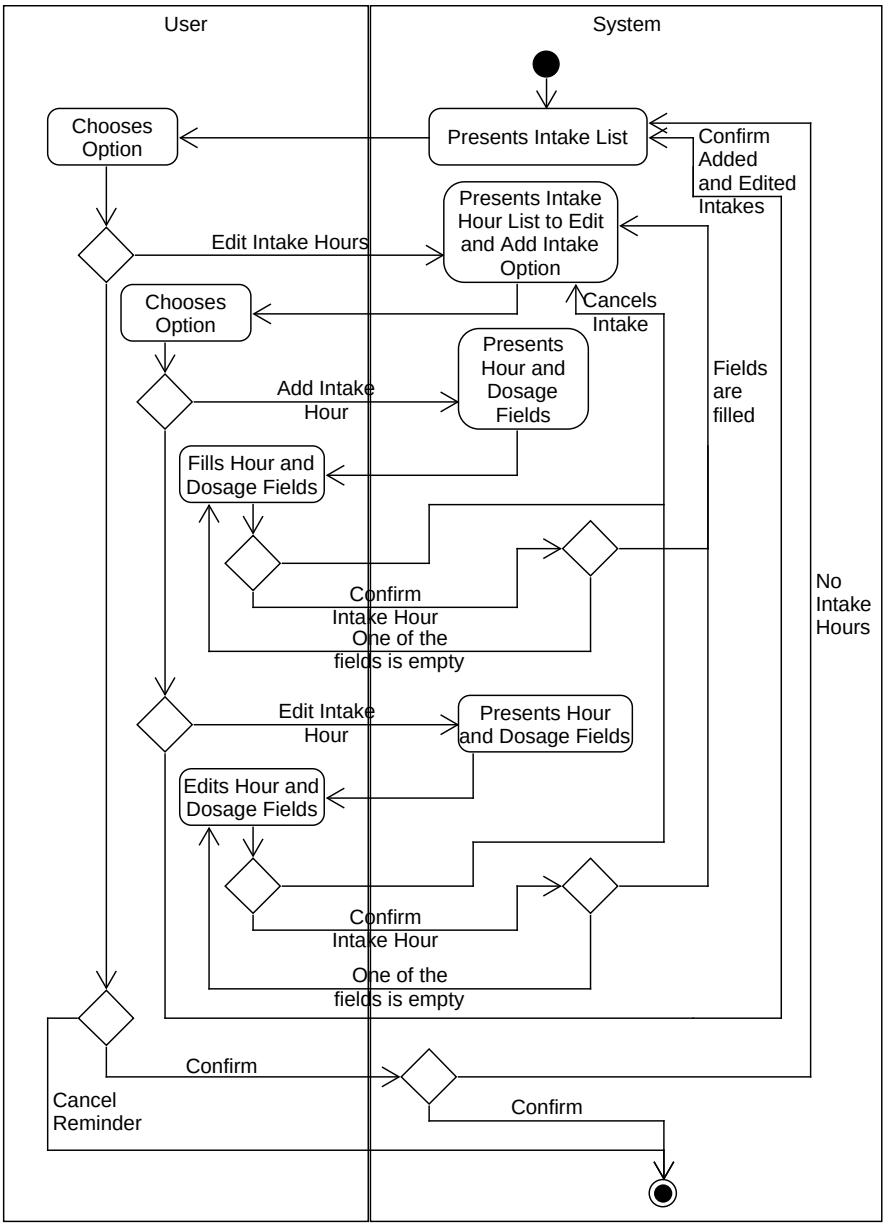

Figure A.3: *Add and Edit Intake* activity diagram.

A.2 Mockup

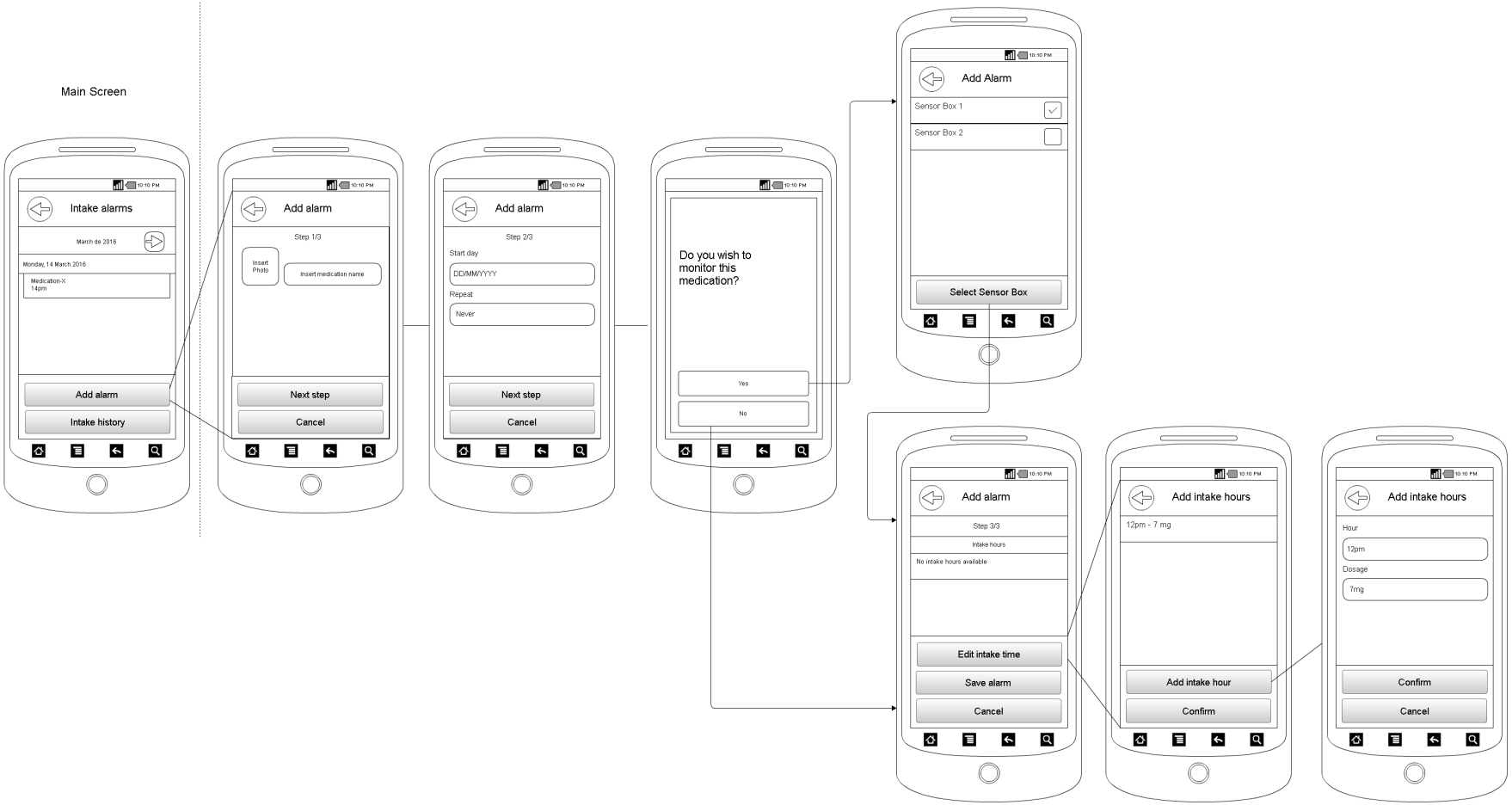

Figure A.4: *Add Alarm and Add Intake Hours* mockup.

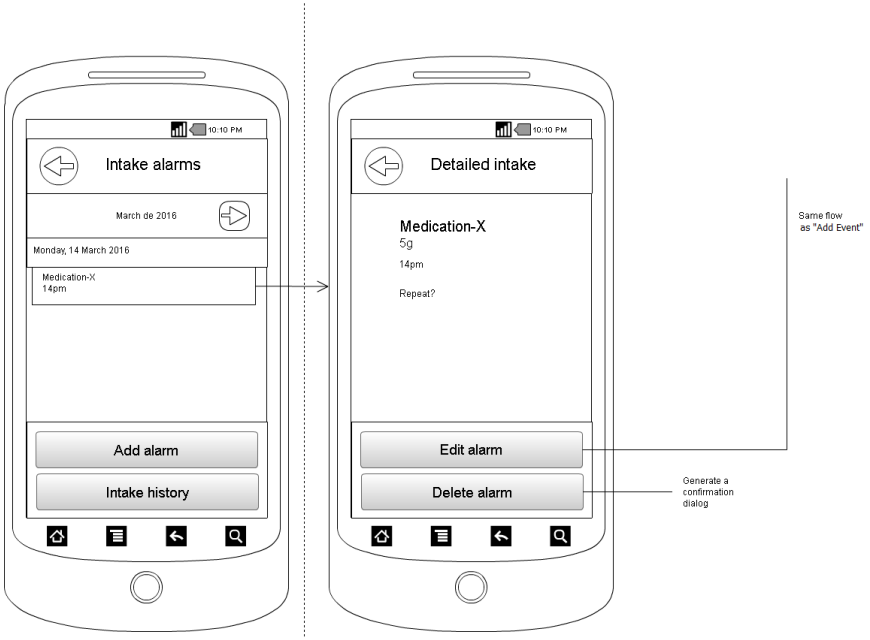

Figure A.5: *Consult Future Intake* mockup.

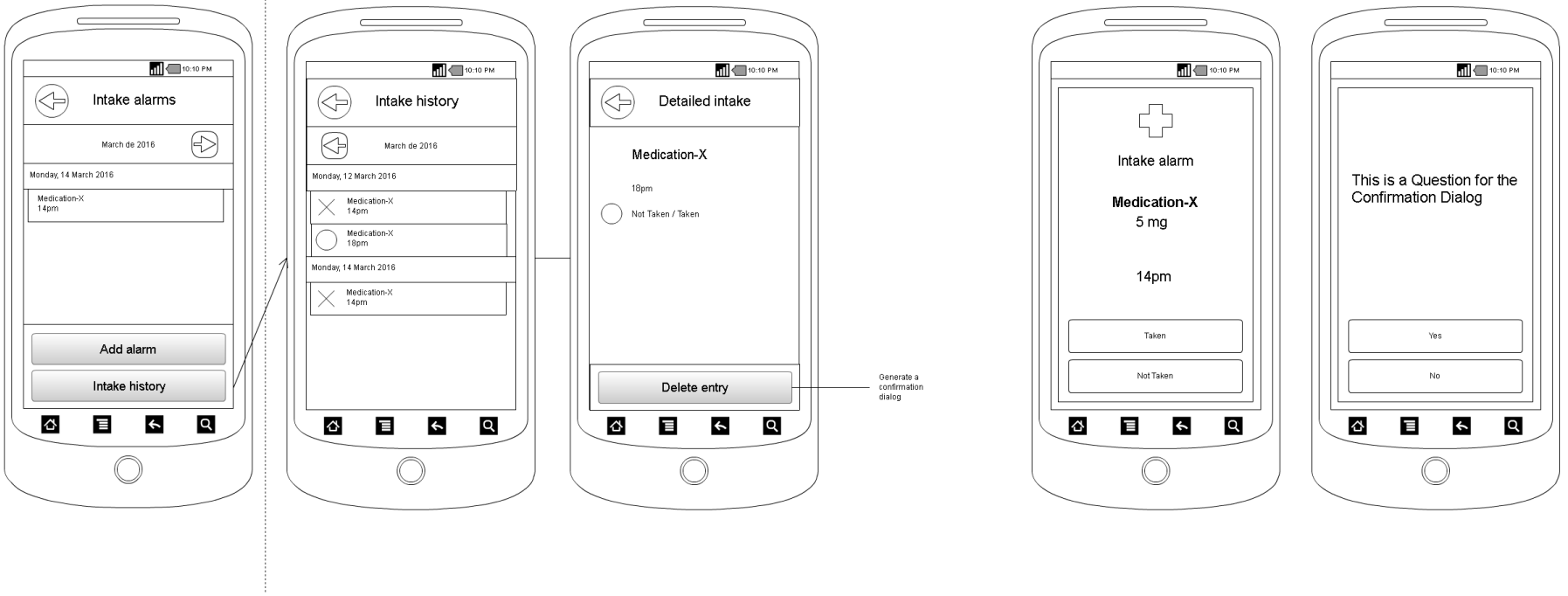

Figure A.6: *Past Intake List*, *Past Intake and Notifications* mockup.

Appendix B

Prototype Development Listings

This appendix contains listings with excerpts of code concerning the developed prototype. These excerpts of code are referred in chapter 4, complementing the description therein contained. This appendix has only one section containing a code excerpt for the Classification and Feature Extraction module B.1 of the Android application prototype, and a code excerpt for the `MedicationDetails` Java object B.2.

B.1 Relevant Listings

Listing B.1: Excerpt of code showing part of the Classification Module class implementation.

```
1  public class ClassificationModule {
2      private Classifier cClassifier = null;
3      public ClassificationModule() {
4          this.cClassifier = new SMO();
5          setTrainSet();
6      }
7      // instance classification
8      public String instanceClassification(String featureVector) {
9          // the variable header contains the attribute set
10         // in the format of ARFF files, the format that WEKA
11         // is able to process
12         String instanceStr = header + featureVector;
13         // (...)
14         Instances i = new Instances(br);
15         int cIdx = i.numAttributes() - 1;
16         i.setClassIndex(cIdx);
17         double[] prediction = cClassifier.distributionForInstance(i.
               firstInstance()); // predict instance
18         String str = "";
19         int max_i = -10;
20         double max = -10;
21         for (int j = 0; j < prediction.length; j++) {
22             if (prediction[j] > max) {
23                 max_i = j;
```

```
                max = prediction[j];
            }
            str += "p(\"" + i.classAttribute().value(j) + ") = " +
                Double.toString(prediction[j]) + " // ";
        }
        str += "\np(\"" + i.classAttribute().value(max_i) + "\") = " +
            Double.toString(prediction[max_i]) + "\n\n";
        return str;
    }
}
```

Listing B.2: Excerpt of code showing some of the data stored in MedicationDetails objects.

```
public class MedicationDetails {
  private String medName = "";       // Medication Name be here
  private String medDesc = "";       // This variable will have the
     dosage, pandlet,recurrence, medStatus, Instances, UUID , rrule
     (...)
  private String medDosage = "";     // This variable will have the
     dosage
  private String medRec = "";        // RRULE be here
  private String medLocation = "SmartMedBoxes-Home"; // Location of the
      event
  private String medPandlet = "";    // MAC address
  private String medInstances = ""; // instance times
  private long medStart = -1;        // Medication Event Start
  private long medEnd = -1;          // Medication Event End
  private long eventID = -1;         // event id (from calendar)
  private long instanceID = -1;      // event instance id (from calendar
     )
  private int medAlarm = 0;          // default value is 0 minutes (when
      med starts, it will alarm)
  private int medStatus = -1;        // 0 - not taken | 1 - taken | 2 -
     unknown | -1 - default
  private String medUUID = "";       // UUID
  // (...) // getters and setters are here
  public JSONObject toJSON() {}
  public static MedicationDetails fromJSON(String jsonString){}

  // (...) // more methods here
```

```
21
22    public void parseDescription() {}//(...)
23    public void generateDescription() {}//(...)
24 }
```