UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

WILLIAM NIEMIEC

Takere: a no-code platform for the development of mHealth applications based on care plans

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Érika Fernandes Cota

Porto Alegre
October 2022

“Those who can imagine anything, can create the impossible.”

— Alan Turing

ACKNOWLEDGEMENTS

I thank God for giving me health and protection to complete the course.

I thank my family for all support provided and for always trying to create the best conditions possible for supporting my studies.

I thank my advisor Dr. Érika Fernades Cota for being always present in every important moment of my studies and my work and also for all knowledge and opportunities provided in my graduation.

I thank the Federal University of Rio Grande do Sul for the opportunity of graduating in my favorite area (computing and technology).

I thank so much my best friend Laura Galant Speggiorin for being my longest friendship, for all the provided support in hard moments, for the wonderful walks, for always being present in my life, and for making me feel good about myself.

I thank Andre Florencio Carini for supporting me in hard moments and for being a very present friend.

I thank Jeniffer Moreira Borges for showing me that is possible to be happy even in unimaginable moments, along with all provided support in hard moments.

I thank Maria Flavia Borrajo Tondo for being an inspiration to me, for having given me several moments of happiness in such a short time, for making college days so good and unforgettable that made me not want them to end, and for showing me how strong and amazing a person can be.

I thank Kaline B. F. Mesquita, who I consider a sister for feeling free to talk about anything, for considering me someone trustworthy, for the various advice given, and also for the great conversations we have.

I thank Keslley Lima da Silva for the many teachings passed and for always being a supportive and comprehensive friend.

Finally, I thank my coworkers and friends Nicole da Costa Davila and Rafael Fernandes Borges for being great partners, for the fun and unforgettable moments we had, and for making the work environment such a good place that it makes me feel sad when I leave it.

ABSTRACT

People on long-term treatment suffer from two problems: following the treatment correctly and staying engaged during it. The main cause of this problem is that each person has a specific engagement along with a lack of clarity in visualizing the progress of the treatment. In order to deal with this problem, mHealth applications have been developed. Although mHealth improves patient engagement, the unavailability of developers to develop these applications according to the treatment of each patient reduces its reachability. This work proposes a no-code platform that allows healthcare professionals to instantiate mHealth applications for their patients according to their care plan. Our goal is to allow healthcare professionals to instantiate mobile applications that show treatment progress and use elements that engage their patients. With this, we hope to contribute so that patients on long-term treatment do not abandon it over time and follow it correctly.

Keywords: No-code. mHealth. healthcare. cross-platform. care plan.

Mobilex: uma plataforma no-code para o desenvolvimento de aplicações mHealth baseadas em planos de cuidado

RESUMO

Pessoas em tratamento de longo prazo sofrem com dois problemas: seguir o tratamento corretamente e permanecer engajados durante ele. A principal causa disso é que cada pessoa possui um fator de engajamento específico aliada à falta de clareza na visualização do andamento do tratamento. Para lidar com este problema, foram desenvolvidas aplicações mHealth. Embora o uso de aplicações mHealth melhore o engajamento do paciente, a indisponibilidade dos desenvolvedores para desenvolver esses aplicativos de acordo com o tratamento de cada paciente reduz sua acessibilidade. Este trabalho propõe uma plataforma no-code que permite aos profissionais de saúde instanciarem aplicativos de mHealth para seus pacientes de acordo com seus planos de cuidados. Nosso objetivo é permitir que profissionais de saúde criem aplicativos móveis que mostrem o progresso do tratamento e usem elementos que engajem seus pacientes. Com isso, esperamos contribuir para que os pacientes em tratamento contínuo não o abandonem com o tempo e sigam ele corretamente.

Palavras-chave: No-code. mHealth. healthcare. cross-platform, plano de cuidado.

LIST OF FIGURES

Figure 2.1 NANDA - Sedentary lifestyle 18
Figure 2.2 NIC - Hypertension management 19
Figure 2.3 NOC - Patient engagement behavior 21
Figure 2.4 User dragging an item and dropping it where he/she wants 22
Figure 2.5 Connected and disconnected graphs 22
Figure 2.6 Directed and undirected graphs 23
Figure 2.7 Breadth-first search algorithm 24
Figure 2.8 HTML structure being parsed by a web browser 24
Figure 2.9 Parsed HTML with and without CSS 25
Figure 2.10 Microservice architecture - overview 27
Figure 2.11 API - overview 28
Figure 2.12 Event-driven architecture - overview 29

Figure 3.1 Approach general overview 34
Figure 3.2 User inserting the care plan element in the flow 35
Figure 3.3 Card generation from the care plan flow 35
Figure 3.4 Patient progress generated from the completed card 36
Figure 3.5 Board generation 37
Figure 3.6 The agenda idea is to filter the board and show only care plan elements close to their deadline 38
Figure 3.7 General overview of progress generation. Dashed lines represent a card that is not in the board anymore because it was already finished 39
Figure 3.8 A flow composed of two connected digraphs 39

Figure 4.1 Takere systems 46
Figure 4.2 Process of adding a new care plan element (structured as described in Section 3.4.2) in the back-end system and its propagation to the front-end systems 46
Figure 4.3 Takere - HCP: Architecture 47
Figure 4.4 Drag-and-drop example using Begin element 48
Figure 4.5 Two elements: X and Y, being Y a child of X 49
Figure 4.6 Begin element - configuration window 49
Figure 4.7 Care plan progress based on a created flow - HCP view 50
Figure 4.8 HCP view about one care plan element that has input fields 50
Figure 4.9 Takere - API: Architecture 51
Figure 4.10 Takere - Patient: Architecture 56
Figure 4.11 Takere - Patient: board 57
Figure 4.12 Takere - Patient: agenda screen 59
Figure 4.13 Takere - Patient: progress screen 60

Figure 6.1 Takere - HCP: Cancer care plan flow 76
Figure 6.2 Takere - HCP: Urolithiasis care plan flow 77
Figure 6.3 Takere - Patient: Patient answering a question of the quiz defined in Table B.6 79
Figure 6.4 Takere - HCP: Patients by care plan 80
Figure 6.5 Takere - HCP: Patient answer for a question of the quiz defined in Table B.6 80

LIST OF TABLES

Table 3.1 Structure of care plan elements 41
Table 3.2 Structure of each parameter item from parameters item from care plan elements 41
Table 3.3 Basic parameters of care plan elements. Note that the slug field was omitted due to being used only for internal system use 42
Table 3.4 Specific parameters of care plan elements. Note that the slug field was omitted due to being used only for internal system use 42

Table 4.1 Architecture modules of Takere - HCP 48
Table 4.2 Architecture modules of Takere - API 52
Table 4.3 Specific parameters semantics 53
Table 4.4 Input types 53
Table 4.5 Takere database. Note that "ObjectId" type is a reference to a record in another table 55
Table 4.6 Architecture modules of Takere - Patient 58

Table 5.1 Selected NICs for each disease 62

Table 6.1 Cancer disease - activities from "Exercise promotion" NIC 64
Table 6.2 Cancer disease - activities from "Self-esteem enhancement" NIC 65
Table 6.3 Urolithiasis disease - activities from "Fluid monitoring" NIC (part A) 66
Table 6.4 Urolithiasis disease - activities from "Fluid monitoring" NIC - part B 67
Table 6.5 Urolithiasis disease - activities from "Teaching: Prescribed diet" NIC 68
Table 6.6 All possible connections of elements 70
Table 6.7 Structure of the explanation element 71
Table 6.8 Structure of the medication control element 72
Table 6.9 Structure of the orientation element 73
Table 6.10 Structure of the quiz element 74
Table 6.11 Structure of the reminder element 75

Table B.1 Care plan for "Exercise promotion" NIC - parameters of EP01, EP02, EP03 and EP04 91
Table B.2 Care plan for "Exercise promotion" NIC - parameters of EP05 91
Table B.3 Care plan for "Exercise promotion" NIC - parameters of EP11 92
Table B.4 Care plan for "Exercise promotion" NIC - parameters of EP14 92
Table B.5 Care plan for "Exercise promotion" NIC - parameters of EP22 92
Table B.6 Care plan for "Fluid monitoring" NIC - parameters of FM06, FM07 and FM22 93
Table B.7 Care plan for "Fluid monitoring" NIC - parameters of FM23 93
Table B.8 Care plan for "Fluid monitoring" NIC - parameters of FM25 93
Table B.9 Care plan for "Fluid monitoring" NIC - parameters of FM26 94
Table B.10 Care plan for "Fluid monitoring" NIC - parameters of FM23 94

LIST OF ABBREVIATIONS AND ACRONYMS

WHO	World Health Organization
HCP	Healthcare professionals
IT	Information Technology
FBP	Flow-Based Programming
NANDA	North American Nursing Diagnosis Association
NIC	Nursing Interventions Classification
NOC	Nursing Outcomes Classification
ANA	American Nurses Association
DFS	Depth-first search
API	Application Programming Interface
POPEP	Periodic Of Periodic Element Problem
mHealth	Mobile health
HTML	Hyper-Text Markup Language
CSS	Cascading Style Sheets
VPL	Visual programming language
FBP	Flow-based programming
REST	Representational state transfer
DAO	Data Access Object
DTO	Data Transfer Object
JSON	JavaScript Object Notation
BSON	Binary JavaScript Object Notation

CONTENTS

1 INTRODUCTION 12
1.1 mHealth - current challenges 13
1.1.1 Generality of mHealth solutions 13
1.1.2 Difficulty for HCP to develop mHealth applications 13
1.2 Goals 14
2 BACKGROUND 16
2.1 Healthcare concepts 16
2.1.1 mHealth 16
2.1.2 Gamification 16
2.1.3 NANDA-NIC-NOC 17
2.1.3.1 NANDA 17
2.1.3.2 NIC 17
2.1.4 NOC 18
2.2 Computing concepts 20
2.2.1 Drag-and-drop 20
2.2.2 Graph concepts 20
2.2.2.1 Connected graph 22
2.2.2.2 Digraph 22
2.2.2.3 Breadth-first Search 23
2.2.3 Web technologies 23
2.2.3.1 HTML 24
2.2.3.2 Rich text 24
2.2.3.3 CSS 25
2.2.3.4 JavaScript 25
2.2.4 Visual programming language 25
2.2.5 Flow-based programming 26
2.2.6 Component-Oriented Programming 26
2.2.7 Internationalization in software architectures 26
2.2.8 Microservice architecture 27
2.2.8.1 Back end 27
2.2.8.2 Front end 28
2.2.8.3 Application Programming Interface 28
2.2.9 RESTful API 28
2.2.10 Event-driven architecture 29
2.2.11 Data modeling 30
2.2.11.1 Models 30
2.2.11.2 DAO 30
2.2.11.3 DTO 30
2.2.11.4 JSON 31
2.2.11.5 BSON 31
2.2.12 No-code platform 31
2.2.13 Job scheduler 31
2.2.14 Non-relational databases 32
2.3 Related Work 32
3 PROPOSED APPROACH 34
3.1 Board 36
3.2 Agenda 37
3.3 Progress 37

3.4 Care plan flow **38**
3.4.1 Building a flow 40
3.4.2 Care plan elements 40
3.4.2.1 Begin element 40
3.4.2.2 Conditional element 40
3.4.2.3 Periodic care plan elements 43
3.4.2.4 Non-periodic care plan elements 43
3.5 Care plan parser **43**
3.5.1 Non-periodic module 43
3.5.2 Periodic module 44
3.5.2.1 Periodicity Of The Periodic Element Problem 44
4 IMPLEMENTATION **45**
4.1 Takere - HCP **45**
4.1.1 Architecture 45
4.1.2 Care plan flow 47
4.1.3 Monitoring patients 50
4.2 Takere - API **51**
4.2.1 Architecture 51
4.2.2 Care plan elements 52
4.2.3 Care plan parser 52
4.2.4 Database 54
4.3 Takere - Patient **54**
4.3.1 Architecture 56
4.3.2 Board 56
4.3.3 Agenda 58
4.3.4 Progress 60
5 VALIDATION METHODOLOGY **61**
6 RESULTS **63**
6.1 Analyzing RQ1 ..... **63**
6.2 Analyzing RQ2 ..... **69**
6.3 Analyzing RQ3 ..... **69**
6.3.1 Care plans planning 73
6.3.2 Building care plans in Takere - HCP 76
6.3.3 Interacting with care plans in Takere - Patient 78
6.3.4 Monitoring patients progress in Takere - HCP 78
7 THREATS TO VALIDITY **81**
8 CONCLUSION AND FUTURE WORKS **82**
REFERENCES **83**
APPENDIX A — INTERVIEW QUESTIONS **90**
APPENDIX B — CASE STUDY - PARAMETER DEFINITION **91**

1 INTRODUCTION

Following a prescribed treatment is following a set of recommendations to control or cure a disease. However, this treatment is frequently interrupted by patients after some time. Burger et al. (BURGER et al., 2018; PRESTON et al., 2018; SHAH et al., 2019) showed that long-term treatment interruption ranges from 19.3% to 94%, and, according to World Health Organization (WHO), in developed countries only 50% of patients with chronic conditions follow the prescribed treatment correctly (BROWN; BUSSELL, 2011). There are several reports about treatment discontinuation: (BROWN; BUSSELL, 2011) shows that the number of patients with systemic arterial hypertension that discontinue the treatment can be as high as 80%; (WU et al., 2008) reveals that medication adherence in patients with heart failure varies widely from 2% and 93%; (DIMATTEO, 2004) analyzed 569 studies of non-psychiatric medication adherence and concluded an average nonadherence rate of 24.8%.

Nonadherence by patients to their treatment can result in fatal consequences. For example, heart failure affects 5.7 million people in United States (MOZAFFARIAN et al., 2016) and may increase the mortality rate if treatment is not correctly followed (GATHRIGHT et al., 2017). In the same way, the survival percentage of patients treating heart attack that discontinue their treatment reduce their survival percentage to 9.2% (HO et al., 2006). On the other hand, older people who have good adherence to their long-term treatment reduce in 21% the risk of death (WALSH et al., 2019). At last, nonadherence to some medications also increases mortality risk (FAUGHT et al., 2008). Therefore, treatment discontinuation negatively impacts patient health, increases the risk of re-hospitalization, and, sometimes, can result in death.

Several approaches have been proposed to increase patient engagement through mobile applications (mHealth). MAO et al. (MAO et al., 2020) shows that using mHealth with diabetes and hypertension patients helps them to keep adherence to their treatments. This is possible because smartphones are accessible to almost everyone (77% of American people (SHEET, 2018)). Also, mHealth helps to improve healthcare in several countries, especially in low- and middle-income countries (MARCOLINO et al., 2018).

1.1 mHealth - current challenges

Although mHealth can increase adherence to long-term treatments (HAMINE et al., 2015), two challenges limit its reachability: generality of current solutions and difficulty for Healthcare professionals (HCP) to generate mHealth applications. In the following subsections, these challenges will be detailed.

1.1.1 Generality of mHealth solutions

First is the generality of current solutions. Current mHealth solutions are static, i.e., they use a care plan as a template to address as many patients as possible. However, as this care plan cannot be changed, some patients may not benefit from this solution. For example, patients who have specific treatment aspects may not have it in current solutions, as they do not allow to change the care plan used as a template for the solution. Also, engagement elements that motivate some patients may not engage others, and a solution that adapts to the needs of each patient is missing. Finally, a patient can have more than one treatment to follow, and its engagement in one is not necessarily the same as the other.

1.1.2 Difficulty for HCP to develop mHealth applications

The second challenge is generating specific mHealth applications that fit the needs of a specific treatment of a patient to keep him/her engaged. To do this, one needs dedicated Information Technology (IT) team for developing and evolving mHealth applications, and, if a new mHealth application needs to be built after developing them, it is necessary to request them again. There are two problems with this: the high cost of developing a mHealth application and the strong dependency between HCP and the IT team. Because of that, when HCP requests an IT team for developing a mHealth application, they ask for a generic solution to subsume as many treatments as possible due to high cost. Consequently, problems mentioned in Section 1.1.1 occur, increasing the probability of treatment discontinuation by patients.

1.2 Goals

The goal of this work is to develop a no-code platform for HCP to develop mHealth applications. Our objective is to instantiate mHealth applications based on care plan information and allow the customization of the application according to the needs of each patient. It is a no-code platform because it is based on Flow-Based Programming (FBP) (explained in Section 2.2.5) which transforms software systems into a processes network, modeling them as a directed graph of predefined processes. These processes are executed in parallel and exchange data through input and output connections (SOUSA, 2012). With that, we expect to offer an abstract tool in the language of the medical team to develop specific mHealth applications for their patients (based on their care plan), improving their engagement.

In order to understand how the care plan works, we contacted around 50 nurses through social networks along with personal contact. From these nurses, 28% returned our contact. So we interviewed 14 nurses from different areas (pediatricians, cardiologists, elderly caregivers, nephrologists, and oncologists). With this research, we learned that the care plan is structured using a set of standards, called NANDA-NIC-NOC (Section 2.1.3). The questions used in this interview are shown in Appendix A. From these interviews, we identified NIC (Section 2.1.3.2) as our target because NANDA (Section 2.1.3.1) is used for diagnoses and NOC (Section 2.1.4) for evaluating an intervention's progress.

In this work, we present Takere [1]: a no-code platform for HCP to instantiate mHealth applications for their patients based on their care plan. In this platform, HCP provides care plan information and a mobile application is instantiated based on it. In this way, HCP can instantiate mHealth applications specifically for each of its patients based on their treatment, offering an additional support to patient engagement along its long-term treatment.

This work presents the following contributions:

- Proposal of a no-code platform for HCP to instantiate mHealth applications;
- Use of patient care plan for generating mobile applications;
- It presents the limitations of the proposed platform and how to extend it to support new types of treatments.

[1] https://github.com/takere

The work is structured as follows: in Section 2 we review the concepts of no-code and mobile development platforms and healthcare terms used in this work. It also discusses the related work. Section 3 presents the overall strategy of the platform. Section 4 details the proposed approach, giving essential knowledge concerning the implementation and use of the platform. Sections 5, 6 and 7 describe, respectively, the validation strategy, the experimental results and the threats to validity. Section 8 discusses the limitations of the proposed solution and possible future works.

2 BACKGROUND

This chapter covers the essential concepts used in this work. We first review healthcare concepts and then present the main computational background.

2.1 Healthcare concepts

In this section, we present the main healthcare concepts used in this work. We start discussing mHealth and gamification concepts. We end by describing the care plan structure (NANDA-NIC-NOC), detailing each classification individually.

2.1.1 mHealth

The set of mobile technologies used for health care is called mHealth (MECHAEL, 2009). It is used as an extension of the doctor's office for helping patients with their treatment. The main criterion of these applications is engagement elements to help patients to stay engaged during their treatment, avoiding abandoning it (ROWLAND et al., 2020). These applications are developed to be generic enough to subsume as many treatments as possible. A study pointed out roughly 50% mHealth users stop using them after some time due to high data entry burden, loss of interest, and hidden costs (KREBS; DUNCAN et al., 2015).

2.1.2 Gamification

Gamification is using game concepts in other contexts beyond games. It is used primarily to increase user adherence to a product, as games are used to keep people entertained and focused (SHERRY et al., 2012; SAILER; HOMNER, 2020). Several studies have shown that gamification provides positive effects depending on how it is implemented along with the context it is used (HAMARI; KOIVISTO; SARSA, 2014; APARICIO et al., 2012; SARDI; IDRI; FERNÁNDEZ-ALEMÁN, 2017; GAALEN et al., 2021).

2.1.3 NANDA-NIC-NOC

In order to improve communication between nurses and to standardize health care, a language standard has been proposed. A nursing classification system began to be discussed in the 1970s, and with it, challenges and issues about the nursing process have appeared (CRUZ, 2008). The main classification systems recognized by American Nurses Association (ANA) are NANDA, NIC, and NOC (ANDERSON; KEENAN; JONES, 2009; BRITO, 2017). Each of these systems will be detailed in the following subsections.

2.1.3.1 NANDA

North American Nursing Diagnosis Association (NANDA) is a global nursing association that standardizes nursing diagnoses (HERDMAN, 2008). A diagnosis is a clinical judgment about individual, family, or community experiences/responses to actual or potential health problems/life processes (PERRY et al., 2013). NANDA is composed of 13 domains, 47 classes, and 201 diagnoses, and its first edition was published in 1982.

Figure 2.1 shows an example of NANDA for a sedentary lifestyle, and it has three components: definite characteristics, related factors, and risk population. The first are signs and symptoms that indicate when the diagnosis should be applied. The second describes possible reasons for the problem, and it is useful for choosing an appropriate nursing intervention. Finally, the risk population refers to people who are more vulnerable to the problem.

2.1.3.2 NIC

Nursing Interventions Classification (NIC) standardizes nursing interventions. Interventions can be independent or collaborative, direct or indirect, and individual or group oriented (WAGNER et al., 2016). NIC is composed of seven domains, 30 classes, more than 500 interventions, and 12000 actions/activities, and it was first published in 1992. The activities are not standardized because it would defeat the purpose of using them to individualize care. In this work, we will refer to activities defined in the NIC reference book (BUTCHER et al., 2018). Finally, we classified actions/activities into two groups: nurse-dependent and independent. The first includes activities that require nurse participation (for example catheter replacement). The second group contains activities that the

Figure 2.1: NANDA - Sedentary lifestyle

Definite characteristics	Related factors	Risk population
• Average daily physical activity is lower than recommended according to sex and age • Choose a daily routine with lack of physical exercise • Do not exercise during free time • Express preference for low physical activity • Perform most activities in reclined position • Perform most of the activities in sedestiation • Loss of physical condition	• Conflicts between cultural beliefs and health practices • Decrease in activity tolerance • Difficulty adapting areas for physical activity • It exceeds the recommended time of use of screens according to age • deterioration of physical mobility • Inappropriate interest in physical activity • Inappropriate knowledge of the consequences of sedentary lifestyle • Inappropriate knowledge about the healthy benefits associated with physical activity • Inadequate motivation for physical activity • Inadequate resources for physical activity • Inappropriate role models • Inadequate social support • Inappropriate skills for time management • Inappropriate training for physical exercise • Low self -efficacy • Low self -esteem • Negative affection towards physical activity • Pain • Breedness practices that inhibit the practice of physical activity by the child • Perception of physical disability • Security risk perception	• Teenagers • People> 60 years of age • People living in urban areas • People living as a couple • People with a high educational level • People with high socioeconomic status • People with significant limitations of time • Married people • Women

Source: https://www.nandadiagnoses.com

patient can do on his/her own (for example drinking water every 2 hours). In this work, we focus on the second group.

Figure 2.2 shows an example of NIC. The definition specifies the goal of the intervention. Next, activities are a set of actions that may be selected for being used in patients to achieve the goal of the intervention. Finally, background reading is a list of recommendations for HCP if they want to get more details about the intervention.

2.1.4 NOC

Nursing Outcomes Classification (NOC) came to standardize nursing expected results from an intervention. Results (or outcomes) are the behavior or perception of a patient in response to nursing interventions (MOORHEAD, 2009). NOC is composed of 31 classes and 385 results, and it was first published in 1991 (MAAS; MOORHEAD, 2000).

Figure 2.3 shows an example of NOC. The definition specifies the goal of the outcome. Next, there are indicators along with a measurement scale. Each indicator is an aspect the patient is being monitored, and the measurement scale is used for HCP to evaluate the patient's progress related to this aspect. Finally, references are a list of

Figure 2.2: NIC - Hypertension management

Hypertension Management 4162

Definition: Preventing and treating blood pressure levels higher than normal

Activities:

- Elicit a detailed patient health history to determine risk level of patient, including medication use
- Identify possible causes of hypertension
- Evaluate for associated risk factors and contributing factors (e.g., diabetes mellitus, dyslipidemia, obesity, metabolic syndrome, age over 60 years, gender, race, smoking, hyperuremia, sedentary lifestyle, family history of hypertension, cardiovascular disease, history of stroke)
- Measure BP to determine presence of hypertension (e.g., normal, less than 120/80; elevated, 120 to 129/80 or less; hypertension stage 1, 130 to 139/80 to 89; hypertension stage 2, equal or greater than 140/90)
- Assure proper assessment of blood pressure (i.e., classification is based on the average of two or more properly measured, seated, BP readings on each of two or more office visits)
- Avoid measurement of blood pressure for classification when contributing factors are present (e.g., consumption of caffeine, migraine headache, insomnia, agitation)
- Implement proper nursing care for patients based on classification of hypertension
- Assist patients with prehypertensive classification to practice lifestyle modification in order to reduce their risk of developing hypertension in the future (e.g., increase exercise, decrease weight, modify diet, obtain adequate sleep)
- Advise patients with prehypertensive classification and comorbid conditions (e.g., heart failure, diabetes, kidney disease) to seek appropriate drug therapy if a trial of lifestyle modification fails to reduce BP to 130/80 mm Hg or less
- Assist patients with hypertensive stage 1 classification and no comorbid conditions (e.g., heart failure, diabetes, kidney disease) to practice lifestyle modifications and to use appropriate drug therapy (e.g., thiazide-type diuretics for most, possibly angiotensinconverting enzyme inhibitor; angiotensin receptor blocker; beta blocker; calcium channel blocker; or combinations of previous)
- Assist patients with hypertensive stage 2 classification and no comorbid conditions (e.g., heart failure, diabetes, kidney disease) to practice lifestyle modifications and to use appropriate drug therapy (e.g., combinations of angiotensin converting enzyme inhibitor, angiotensin receptor blocker, beta blocker, calcium channel blocker)
- Assist patients with hypertensive stage 1 or 2 classification and comorbid conditions (e.g., heart failure, diabetes, kidney disease) to practice lifestyle modifications as able and to follow recommended drug regime protocols for comorbid condition with hypertension
- Monitor at-risk patients for signs and symptoms of hypertension crisis (e.g., severe headache, dizziness, nausea or vomiting, pallor, sweating, cold skin, changes in vision, epistaxis, confusion, nervousness, restlessness, visual disturbances, altered level of consciousness, chest pain, seizures, cardiac arrest)
- Monitor vital signs such as heart rate, respiratory rate, oxygen saturation, temperature, and blood panels for early identification of complications
- Instruct at-risk patients to have regular preventative health screenings, including electrocardiogram, echocardiogram, electrolytes, urinalysis, as indicated
- Monitor patient for signs and symptoms of hypertension or hypotension after administering prescribed hypertension medication
- Instruct related to healthy dietary pattern
- Instruct related to proper physical activity (e.g., exercise 30 to 45 minutes a day)
- Instruct related to contributing lifestyle habits that should be avoided (e.g., use of tobacco in any form and alcohol)
- Instruct the patient on lifestyle modification related to sleep and rest patterns (e.g., 8 hours per night is recommended)
- Provide information on possible changes in lifestyle necessary to avoid future complications and control the disease process
- Provide information related to the purpose and benefit of the lifestyle changes
- Instruct related to self-blood pressure monitoring and to report abnormal findings
- Instruct the patient on possible causes of hypertension
- Instruct the patient and family to take an active role in the management of disease process, (e.g., medication indications and administration, maintaining proper diet, exercise and healthy habits, quitting smoking, reducing stress, reducing weight, reducing sodium intake, reducing alcohol consumption, increasing exercise, as indicated)
- Instruct the patient and family on medication usage and indications
- Encourage the patient and family to maintain a list of current medications and reconcile routinely at wellness checks, hospital visits, or hospital admissions
- Instruct the patient to recognize and avoid situations that can cause increased BP (e.g., stress or sudden discontinuation of drug treatment)

7th edition 2018

Background Readings:

American Association of Critical Care Nurses. (2006). *Core curriculum for critical care nursing* (6th ed.) [J. G. Alspach, Ed.]. Philadelphia, PA: W.B. Saunders.

Chummun, H. (2009). Hypertension: A contemporary approach to nursing care. *British Journal of Nursing*, *18*(13), 784–789.

Hacihasanoğlu, R., & Gozum, S. (2011). The effect of patient education and home monitoring on medication compliance, hypertension management, healthy lifestyle behaviors and BMI in a primary health care setting. *Journal of Clinical Nursing*, *20*(5/6), 692–705.

Margolius, D., & Bodenheimer, T. (2010). Controlling hypertension requires a new primary care model. *The American Journal of Managed Care*, *16*(9), 648–650.

U.S. Department of Health and Human Services, National Institutes of Health, National Heart, Lung and Blood Institute. (2004). *Seventh report of the Joint National Committee on prevention, detection, evaluation and treatment of high blood pressure*. Washington, DC: National Institute of Health.

Whelton, P. K., Carey, R. M., Aronow, W. S., Casey, D. E. Jr., Collins, K. J., Dennison Himmelfarb, C, . . . Wright, J. T. Jr. (2017). ACC/AHA/AAPA/ ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults, *Journal of the American College of Cardiology,* (2017), doi: 10.1016/ j.jacc.2017.11.006

Source: https://nursing.uiowa.edu

recommendations for HCP if they want to get more details about the outcome.

2.2 Computing concepts

In this section, we present the main computing concepts used in this work. We start defining the drag-and-drop technique, followed by graph concepts and web technologies. After that, we describe some software concepts and data modeling. Finally, we present no-code, job scheduler, and non-relational database concepts.

2.2.1 Drag-and-drop

Drag-and-drop in software engineering is when it is possible to drag an element to a different location and drop it. It is a technique for improving usability and removing the need to write text commands (INKPEN, 2001; JALENDER et al., 2011). For example, suppose there are three items, and an user wants to drag-and-drop the first element (Figure 2.4a). He/she needs to drag this item (Figure 2.4b) and drop it where the user wants to put it (Figure 2.4c).

2.2.2 Graph concepts

A graph is a diagram composed of a set of nodes and edges. Nodes are the elements of the graph while edges are lines that can connect them. When a node X is connected with another node Y, then the node X is a parent of the node Y, and the node Y is a child of the node X. If a node does not have any children, it is called a leaf node, while nodes do not have parents are called root nodes.

Graphs can be used for modeling several problems, including flights management (KINCAID, 2003; SOUZA, 2008), allocation (DIAS; FREITAS; MACULAN, 2012), path finding (MENDELZON; WOOD, 1995; MA, 2022), among others. There are several properties for graphs, and we explain some of them in the following subsections. Finally, we end by explaining an algorithm for traversing graphs.

Figure 2.3: NOC - Patient engagement behavior

Patient Engagement Behavior--1638

Definition: Personal actions to actively participate in one's health care through shared decision-making with health professionals

OUTCOME TARGET RATING: Maintain at_____ Increase to_____

		Never demonstrated	Rarely demonstrated	Sometimes demonstrated	Often demonstrated	Consistently demonstrated	
OUTCOME OVERALL RATING		**1**	**2**	**3**	**4**	**5**	
Indicators:							
163801	Obtains reputable health information	1	2	3	4	5	NA
163802	Assesses personal health risk factors	1	2	3	4	5	NA
163803	Identifies causes of illness	1	2	3	4	5	NA
163804	Identifies factors that influence health	1	2	3	4	5	NA
163805	Follows a healthy lifestyle	1	2	3	4	5	NA
163806	Treats minor conditions	1	2	3	4	5	NA
163807	Seeks professional assistance when needed	1	2	3	4	5	NA
163808	Selects appropriate health professional	1	2	3	4	5	NA
163809	Prepares a list of questions to discuss with health professional	1	2	3	4	5	NA
163810	Brings current medication list to discuss with health professional	1	2	3	4	5	NA
163811	Shares medical information with health professional	1	2	3	4	5	NA
163812	Discusses personal health priorities with health professional	1	2	3	4	5	NA
163813	Shares strategies to meet personal health priorities	1	2	3	4	5	NA
163814	Discusses plan of care with health professional	1	2	3	4	5	NA
163815	Seeks second opinion	1	2	3	4	5	NA
163816	Chooses among treatment options	1	2	3	4	5	NA
163817	Monitors treatment effects	1	2	3	4	5	NA
163818	Monitors medication effects	1	2	3	4	5	NA
163819	Shares side effects with health professional	1	2	3	4	5	NA
163820	Follows up with health professional when health status changes	1	2	3	4	5	NA
163821	Obtains test results	1	2	3	4	5	NA
163822	Obtains appropriate health screenings	1	2	3	4	5	NA
163823	Obtains recommended vaccines	1	2	3	4	5	NA
163824	Maintains personal health record	1	2	3	4	5	NA
163825	Maintains insurance coverage	1	2	3	4	5	NA
163826	Maintains advance directives	1	2	3	4	5	NA
163827	Obtains medical power of attorney	1	2	3	4	5	NA
163828	Shares concerns for personal safety	1	2	3	4	5	NA
163829	Uses strategies to cope with the effects of chronic illness	1	2	3	4	5	NA
163830	Manages personal health care	1	2	3	4	5	NA
163831	Uses health care resources consistent with need	1	2	3	4	5	NA

***Domain**-Health Knowledge & Behavior (IV)* ***Class**-Health Behavior (Q)* *6th edition 2018*

OUTCOME CONTENT REFERENCES:

Coulter, A. (2011). *Engaging patients in healthcare.* New York, NY: McGraw-Hill Education.

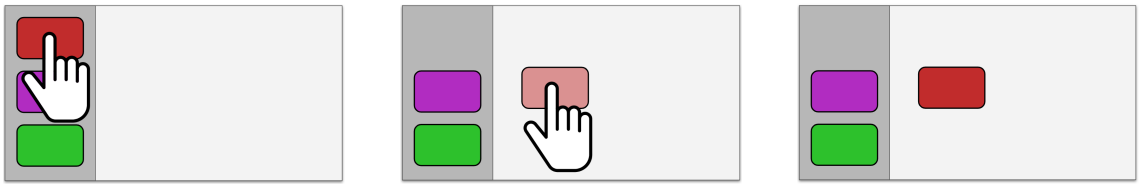
Coulter, A. (2012). Patient engagement -- what works? *Journal of Ambulatory Care Management, 35*(2), 80-89.

Duke, C., Lynch, W., Smith, B., & Winstanley, J. (2015). Validity of a new patient engagement measure: The Altarum consumer engagement (ACE) measure. *Patient, 8*(6), 559-568.

Gruman, J., Holmes-Rovner, M., French, M., Jeffress, D., Sofaer, S., Shaller, D., & Prager, D. (2010). From patient education to patient engagement: Implications for the field of patient education. *Patient Education and Counseling, 78*(3), 350-356.

Source: https://nursing.uiowa.edu

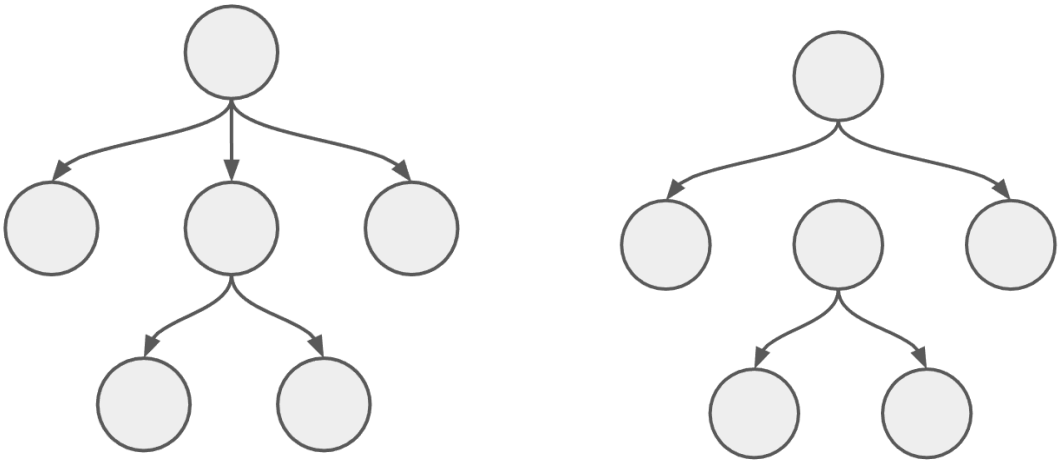
Figure 2.4: User dragging an item and dropping it where he/she wants



(a) Initial state (b) Item being dragged (c) Item after being dropped

Source: The author

Figure 2.5: Connected and disconnected graphs



(a) Connected graph (b) Disconnected graph

Source: The author

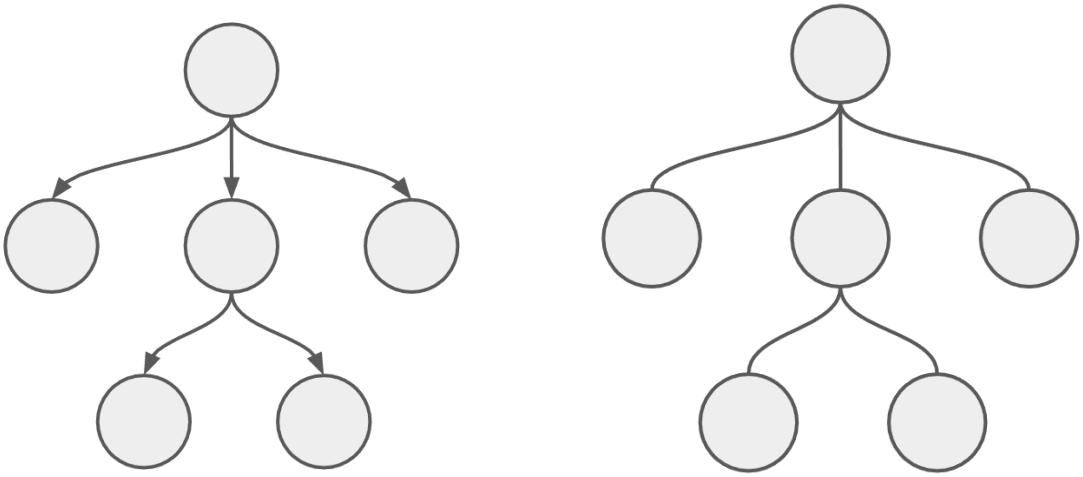
2.2.2.1 Connected graph

A graph is connected if there is a path between every pair of nodes. A path is a sequence of nodes where the first node has an edge connecting it with the next node and so on. On the other hand, a graph that does not have this property is called an disconnected graph (Figure 2.5 shows both graphs).

2.2.2.2 Digraph

A directed graph (also called a digraph) is a graph where each edge has a direction. For example, suppose there is a node X connected with another node Y through an edge. If it is allowed to have a path traversing X to Y and vice-versa, then it is an undirected graph. On the other hand, if it is only possible to traverse X to Y, then it is a directed graph. Figure 2.6 shows better the difference between then.

Figure 2.6: Directed and undirected graphs



(a) Directed graph (b) Undirected graph

Source: The author

2.2.2.3 Breadth-first Search

Breadth-first search is an algorithm used for traversing graphs. It starts from a root node and traverses all nodes at the present depth prior to moving on to the nodes at the next depth level. The algorithm is shown in Figure 2.7.

2.2.3 Web technologies

In 1991, Tim Berners-Lee proposed a standard language for sharing documents, called Hyper-Text Markup Language (HTML). However, style and text were developed in the same language - HTML - reducing maintainability. To solve this problem, a style sheet language was proposed in 1994, called Cascading Style Sheets (CSS). It was designed for separating the presentation from the content. CSS also improves accessibility (ZAKRAOUI; ZAGLER, 2012; KENNEDY; LEÓN, 2011) along with reusability (COLLISON, 2007). With HTML and CSS, it was only possible to create static websites, i.e., sites that do not change after being loaded. In order to allow the creation of dynamic websites (sites that can change after being loaded), a scripting language was proposed in 1995, called JavaScript. Besides being created for websites, these technologies are used in other contexts, such as mobile development (EISENMAN, 2015), and back-end systems (TILKOV; VINOSKI, 2010), among others. In the following subsections, we explain more about these three languages.

Figure 2.7: Breadth-first search algorithm

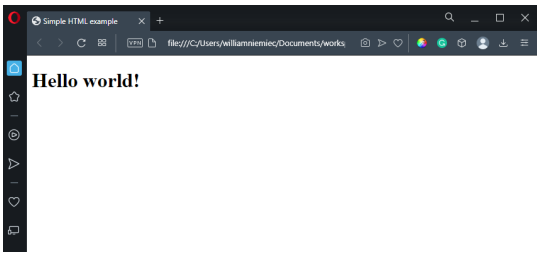
```
procedure BFS(tree):
    root := get root element from tree
    add root in queue
    while queue is not empty do
        node := remove first element from queue
        if node is not marked as traversed then
            mark node as traversed
            for all child of node do
                add child in queue
            end for
        end if
    end while
end procedure
```

Source: The author

Figure 2.8: HTML structure being parsed by a web browser

```
<!DOCTYPE html>
<html>
    <head>
        <title>Simple HTML example</title>
    </head>
    <body>
        <h1>Hello world!</h1>
    </body>
</html>
```

(a) HTML code



(b) Parsed HTML

Source: The author

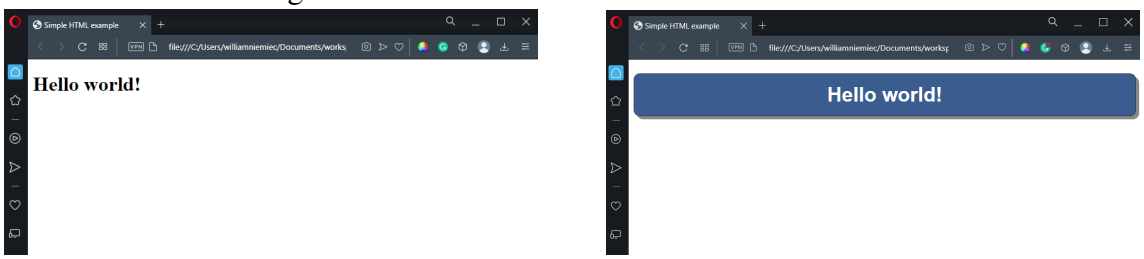
2.2.3.1 HTML

HTML is a markup language used for defining the structure of an application. It uses tags where each tag has semantics. Thus, from these tags, a parser (like web browsers) can read the document and understand the defined structure, as shown in Figure 2.8.

2.2.3.2 Rich text

Rich text is text formatted using HTML. It provides several formatting commands, such as different fonts, colors, size, bold, italic, and underlining, among others. On the other hand, plain text content does not contain formatting, images, colors, or other types of markup. It also includes single line breaks and spacing. Thus, the rich text should be used when one wants to stylize a text or to express some feeling without using words for

Figure 2.9: Parsed HTML with and without CSS



(a) HTML without CSS (b) HTML with CSS

Source: The author

that (ZHANG et al., 2021; GOFFIN et al., 2016).

2.2.3.3 CSS

CSS is a style language responsible for defining the presentation of an application based on a structure. It consists of defining a set of properties for tags from HTML, such as color, size, and border. It also allows several HTML documents to use the same CSS file without having to duplicate content. Finally, Figure 2.9 shows a comparison between an HTML document without CSS and another with the same content but using CSS.

2.2.3.4 JavaScript

JavaScript is a scripting language whose original goal was to allow the creation of interactive websites. With it, it is possible to manipulate page elements along with adding behavior to websites. JavaScript code can run even after a website has been loaded, making it possible to build interactive websites. Besides being created for websites, several libraries have been released using JavaScript as a basis that is not related to the web. For example, React Native [1] is a framework for building mobile applications that uses JavaScript for that.

2.2.4 Visual programming language

Visual programming language (VPL) makes it possible to create programs graphically through element manipulation. It is very useful for allowing the creation of programs without having to write code for that (MORALES; RUSU, 2020). Also, it can be used

[1] https://reactnative.dev

for learning purposes (TSAI, 2019; DASKALOV; PASHEV; GAFTANDZHIEVA, 2021; RAO; BIHANI; NAIR, 2018).

2.2.5 Flow-based programming

Flow-Based Programming (FBP) is a paradigm that uses directed graph (Section 2.2.2.2) of predefined processes for modeling software systems (MORRISON, 1994). It aims to have a natural way of abstracting logic and an easy way of visualizing each of its elements.

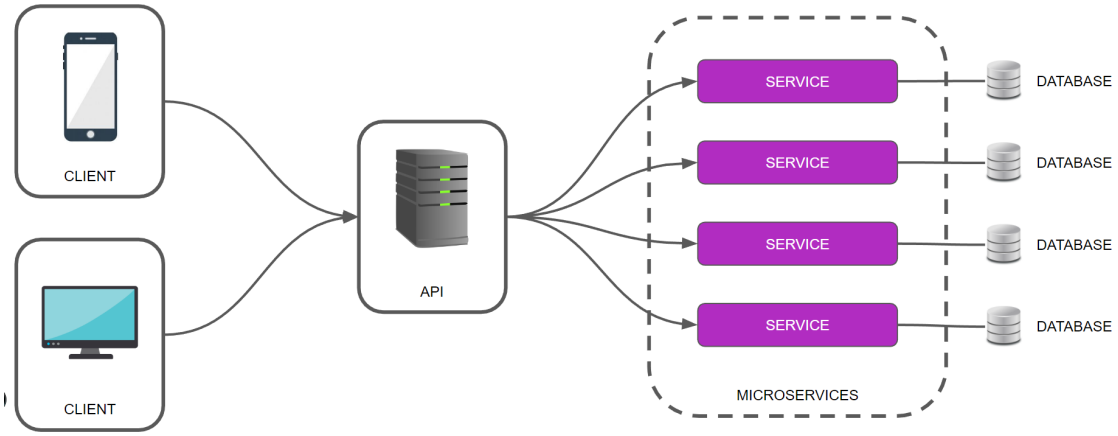
2.2.6 Component-Oriented Programming

Usually, 50% of software requirements are equal in different domains and systems (SOMMERVILLE, 2010). Thus, the probability of developing some piece of code already existent is high. Software reuse improves productivity and, using system pieces already tested, documented, and approved improves software quality (KRUEGER, 1992; FRAKES; TERRY, 1996). Based on that, component-oriented programming proposes to develop software applications by combining components already developed along with new components.

2.2.7 Internationalization in software architectures

Internationalization is the strategy of designing software to be compatible with new languages without having to change the code already produced (KERSTEN; KERSTEN; RAKOWSKI, 2002; LUONG et al., 1995; AYKIN, 2004). We refer to a language as new when the software does not have compatibility with it. When software has been designed to be compatible with internationalization, adding compatibility with a new language does not require changing the source code. All texts present in the software are located in one place, and each language has its dictionary. A dictionary is a set of elements, each with two fields: key and value. The first is an identifier for the second. On the other hand, the value is the text written in the language the dictionary belongs to. Thus, to add compatibility with a new language, the only thing to do is to create a new dictionary with the same keys but with values written in the new language.

Figure 2.10: Microservice architecture - overview



Source: The author

2.2.8 Microservice architecture

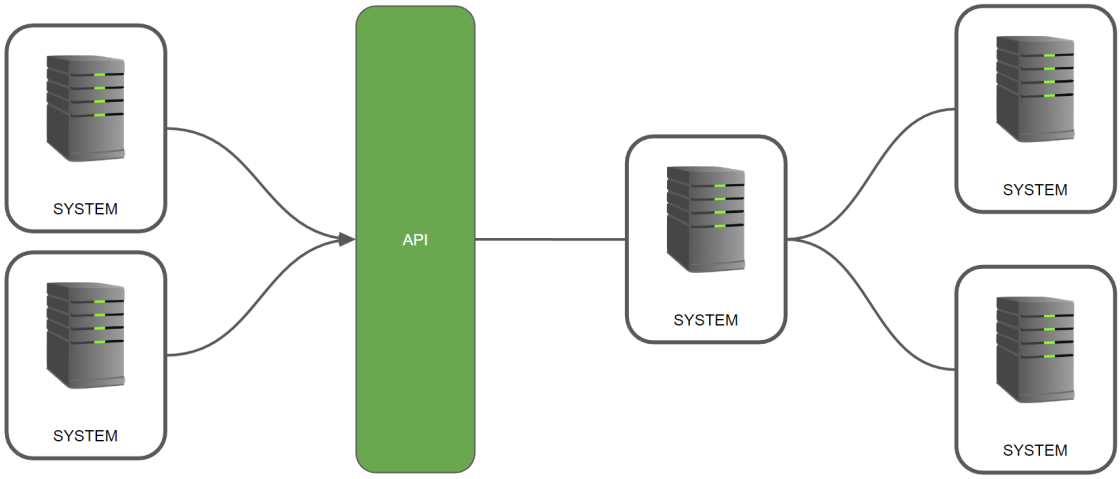
Microservice architecture aims to build systems as a set of small independent processes (Figure 2.10). Each service is developed according to its business logic and is independent of others. Using this architecture, it is possible to provide services for several distinct clients. Also, there are several advantages this architecture produces, including cost reduction (VILLAMIZAR et al., 2016), easy maintainability (DRAGONI et al., 2017), and resiliency (NADAREISHVILI et al., 2016). In the following subsections, we introduce some concepts related to this architecture.

2.2.8.1 Back end

Back-end systems are responsible for managing data for other systems. It consists on providing data on request: an application requests some data, the back-end system parses this request, handles the database to accomplish the request, and, finally, returns to the application the requested data (ADAM; BESARI; BACHTIAR, 2019).

Back-end systems in the microservice architecture context are API servers (SURYOTRISONGKO; JAYANTO; TJAHYANTO, 2017). Each system has a set of endpoints, which are addresses for each available service. They also deal with database connection and database management. Note that back-end systems are usually called servers.

Figure 2.11: API - overview



Source: The author

2.2.8.2 Front end

Front-end systems are responsible for providing a graphical interface to users. They request data from back-end systems and use them for building a layout. These requests can be for getting or writing some data. The front-end system does not need to know how this will be done, only how it builds a request for both options (HARMS; ROGOWSKI; IACONO, 2017; PAVLENKO et al., 2020). Note that front-end systems are usually called clients.

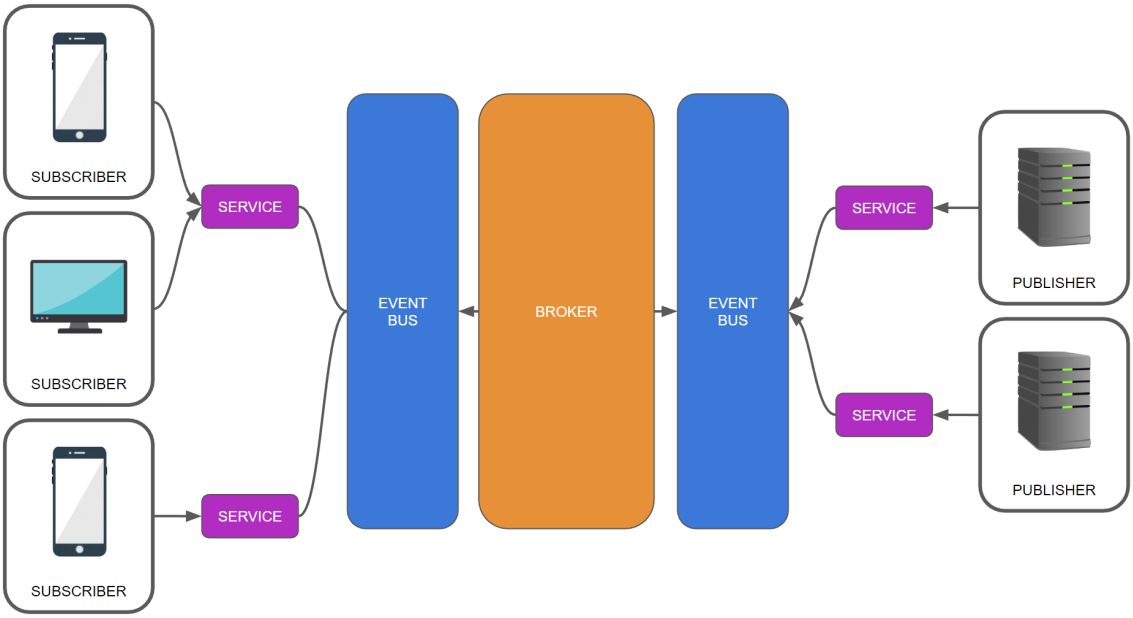
2.2.8.3 Application Programming Interface

Application Programming Interface (API) is a set of definitions used for communication between systems. It is a layer that specifies what actions are possible to do and what information is necessary to provide in order to do this action. It aims to inform a system how how to communicate with another system and what it can do in this system (Figure 2.11).

2.2.9 RESTful API

RESTful API is an API that implements REST, which is a set of specified rules about how an API from a back-end system has to be, including stateless, layered, and code-on-demand. The first rule refers to a communication mode between client and server

Figure 2.12: Event-driven architecture - overview



Source: The author

where each client request is completed independently of the previous requests. On the other hand, the second specifies the system has to use a layered architecture, including a security layer, a business logic layer, and a request handler layer. Finally, the code-on-demand rule says the server is who decides what features a client can use, and it can change them depending on some logic.

2.2.10 Event-driven architecture

Event-driven architecture is a design pattern whose goal is to provide asynchronous communication between services (MICHELSON, 2006; MARÉCHAUX, 2006; CLARK; BARN, 2011; TAYLOR et al., 2009). It uses event streams for notifying all interested applications when the system status changes. This interaction consists of the system sending microservice events to an event bus followed by a broadcast (Figure 2.12). The event bus handles new events while broadcasting and forward them to the subscribed components. Events are published by some components (publishers) and received by others (subscribers). A subscriber needs to subscribe to each component it wants to receive events from. Note that publishers and subscribers do not know each other, as this is the responsibility of the Broker.

2.2.11 Data modeling

Information is a set of data related to some subject. To manipulate it in a computer, it is necessary to define a structure for this information (SIMSION; WITT, 2004). This process is called data modeling, and there are several approaches for representing a piece of information. In the following subsections, we introduce some of these techniques.

2.2.11.1 Models

Models are structured elements based on relevant properties of a real-world concept (FLANDERS; JANNIDIS, 2015). The definition of what properties are relevant or not depends on the context they are used. Also, the real-world concept can be a physical object (like a chair), or an abstract concept (like a law), among others. For instance, if one is building a shopping system, models may include stores, products from this store, and employers, and it is necessary to define which properties each of these models have (for example, employers can have a name, a genre, birth date, wage, among others). The definition of what properties a model will have depends on what information is relevant to keep, and this depends on the context of the system.

2.2.11.2 DAO

The data access object (DAO) is a design pattern that aims to represent data independently of its source (text, database, XML file, among others). For that, it encapsulates data access through an interface, allowing data access mechanisms can be changed independently of the code that uses the data (MATIC; BUTORAC; KEGALJ, 2004; NOCK, 2004). Thus, these objects contain the business logic of the data they represent.

2.2.11.3 DTO

The data transfer object (DTO) is a design pattern used for carrying data between processes. These pattern aims to aggregate data that is expected to be transferred between several processes in order to reduce the number of calls, and, consequently, make this process less costly (MONDAY, 2003). Compared with DAO (Section 2.2.11.2), these objects do not contain any business logic, being more simple than DAO.

2.2.11.4 JSON

JavaScript Object Notation (JSON) is a data interchange format used for exchanging data with other systems in a standard way (PEZOA et al., 2016). It is human-readable data and it is based on a subset of the JavaScript programming language standard. JSON files are composed of JSON objects, and each object is composed of a key and a value. The first is a text used for accessing a value from the object, while the second can be a number, text, boolean, list, empty value, or even another object. Note that, besides being based on JavaScript programming language, JSON is completely language-independent, and the structure of JSON files can be mapped to almost any programming language (CROCKFORD, 2006).

2.2.11.5 BSON

Binary JSON (BSON) is JSON represented in a binary structure. It aims to manage JSON files more efficiently, and also to add new data types, like dates and binary data (VIOTTI; KINDERKHEDIA, 2022). For that, the BSON structure encodes type and length information in JSON files, allowing parsing them more quickly along with adding compatibility with new data types.

2.2.12 No-code platform

No-code software is a software development technique for building applications without writing code (MCLEAN, 2021; YAN, 2021). It consists on using drag-and-drop (Section 2.2.1) concept and connecting components, creating a graph (Section 2.2.2). This approach has some benefits, such as speeding up the development process (PLODER et al., 2019). It also reduces the dependency on programmers, and, consequently, reduces costs (WONG; DRIVER; VINCENT, 2019).

2.2.13 Job scheduler

A job is a program running in a system. A job scheduler is a job manager whose goal is to decide which and when a job should run. Note that operating systems have a similar concept: a job scheduler is a job manager that handles the removal of a running

job and selects another job for run (SILBERSCHATZ; GALVIN; GAGNE, 2018). In our work, we do not want to stop a job to run another; we are only interested in deciding which and when a job should run.

2.2.14 Non-relational databases

Non-relational databases are databases that stores data in a non-tabular form. As data does not have a fixed structure, each document of the database can store data in different formats. A document is a list of records, and a database can contain several documents (BHAT; JADHAV, 2010). Compared with relational databases, each document (called a table) has fixed fields, and, consequently, a fixed structure.

2.3 Related Work

Several approaches have been proposed in the literature for improving patient engagement. DinoApp (SILVEIRA et al., 2021) is a mobile application developed to support children with cancer. It helps HCP to be closer to children with cancer after they are discharged from the hospital. It does not replace HCP interaction, but helps it, having a communication channel between them (frequented asked questions), in addition to other functionalities, such as calendar, glossary, among others. Our approach has a similar goal but expands it for different treatments beyond children with cancer.

Mussi et al. (MUSSI et al., 2013) showed that home visit by HCP improves the healthcare of patients with heart failure. In this research, patients who received HCP visits improved 24.8% their self-care and adhesion to their treatment, in contrast with 9.76% of those who have not had these visits. Continuing the previous work, (SOUZA et al., 2014) focuses on HCP monitoring their patients with heart failure by phone calls. This approach reduced in 27% hospital admissions and deaths by heart failure. Our approach aims to achieve similar results but without the presence of HCP since there may be insufficient professionals to attend to the demand.

(GUZZO, 2017) proposes to automate the pre-hospital attendance system. One of the goals of this automation is to standardize the nursing language - improving their communication - and to persist this information in a database. Our approach proposes to develop a similar system, but including a system for patients, and not only for HCP.

(CARVALHO, 2021) proposes a no-code platform for HCP to instantiate mHealth applications to improve patient engagement. No-code elements are generic enough to be used in different treatments, and he focuses on identifying engagement elements to be used when generating a flow. A flow represents the order that the elements that should appear in the mHealth application along with dependencies (for example if an element should be parsed only after another has finished). Then, this flow is processed, and a mHealth application is instantiated based on it. In our approach, we extend this work to use specific no-code elements based on NIC (Section 2.1.3.2). With that, we expect HCP to instantiate specific mHealth applications that are in accordance with a treatment specified for each patient.

3 PROPOSED APPROACH

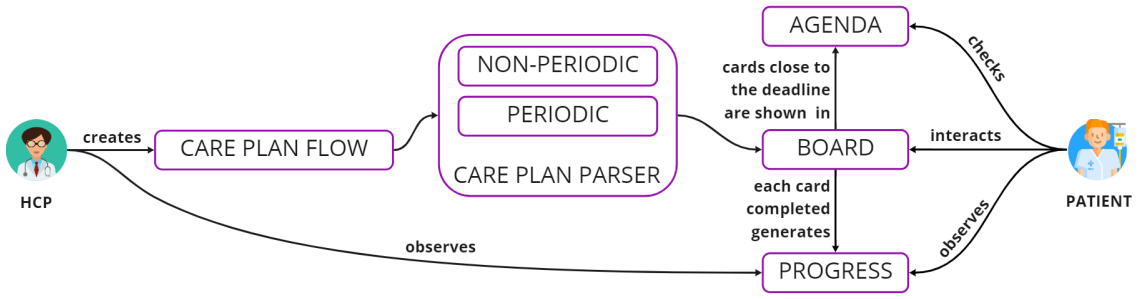
We propose a no-code platform for HCP to instantiate specific mHealth applications based on care plans. As our goal is that non-programmers use our approach, care plans are built using FBP (Section 2.2.5) and VPL (Section 2.2.4). Our approach (Figure 3.1) is composed of five components: care plan flow, care plan parser, agenda, board, and progress. In the following sections, all these components are explained.

For instance, let us assume a patient that has a disease called urolithiasis. Also, assume that HCP evaluated the patient and prescribed a treatment with the following restriction: the patient should intake more than two liters of water per day. In that case, the HCP has to build the flow of this treatment (care plan flow). For that, it is necessary to choose which care plan elements compose this flow along with how they relate to each other. In this example, HCP selected an element for asking the patient daily how many liters of water he/she drinks (Figure 3.8).

After the care plan flow is created, it is necessary to parse this flow. In this analysis, a card is generated for the patient to interact with (Figure 3.3). Also, as this element is periodic, this card will be generated periodically, without having to create another flow for that.

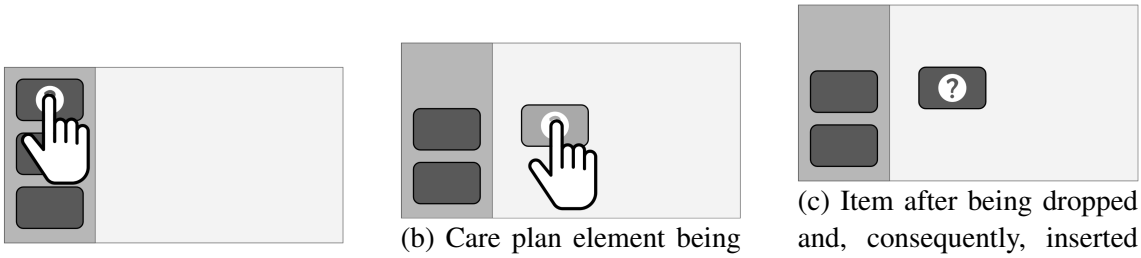
Finally, when the care plan flow is created and analyzed, the patient can interact with it. For that, a board is generated containing one card. A card is a set of information and/or inputs provided by care plan elements. In this example, the card has a question, and the patient should provide some input. After that, the activity is marked as completed. Also, the patient can check which activities have been completed in progress. This progress can also be observed by HCP, checking which activities have been completed along with the inputs provided by the patient. Finally, the patient can see activities

Figure 3.1: Approach general overview



Source: The author

Figure 3.2: User inserting the care plan element in the flow



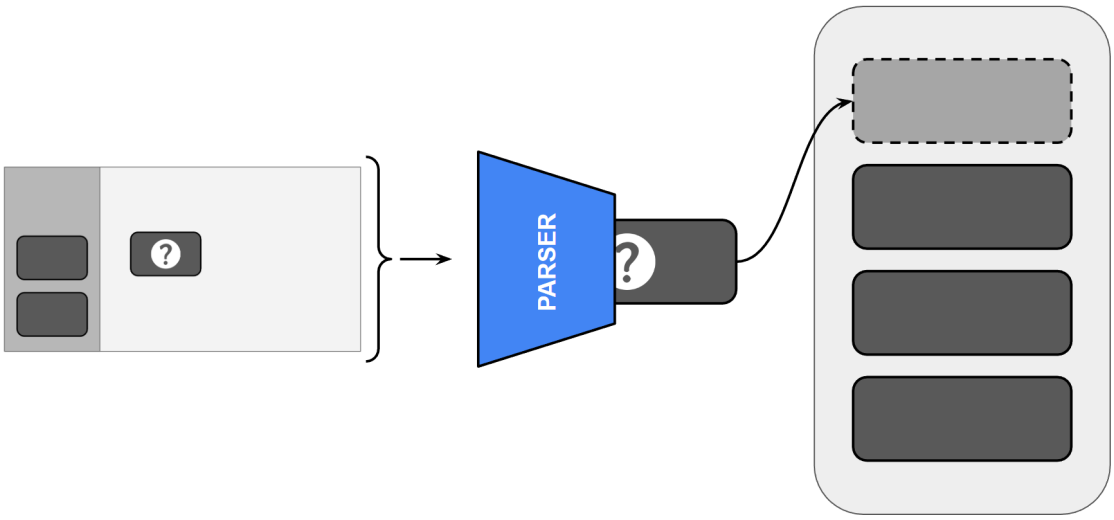
(a) Initial state

(b) Care plan element being dragged

(c) Item after being dropped and, consequently, inserted in the flow

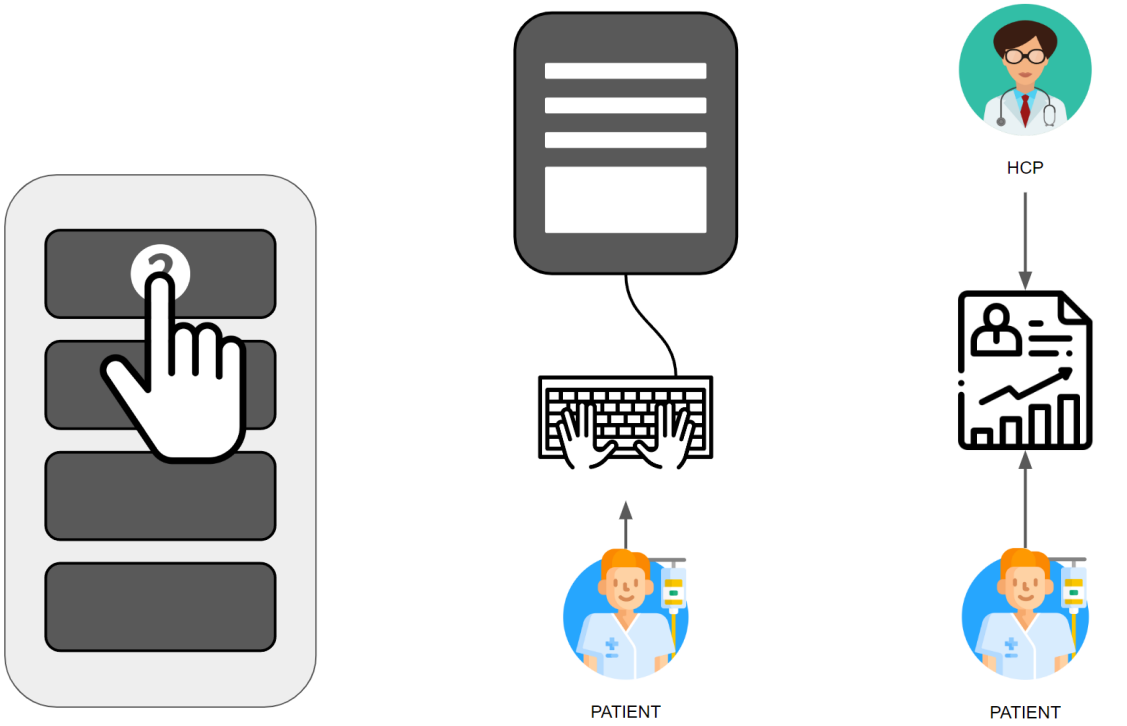
Source: The author

Figure 3.3: Card generation from the care plan flow



Source: The author

Figure 3.4: Patient progress generated from the completed card



(a) Patient selecting the card with the question

(b) Patient answering the question

(c) Item after being dropped

Source: The author

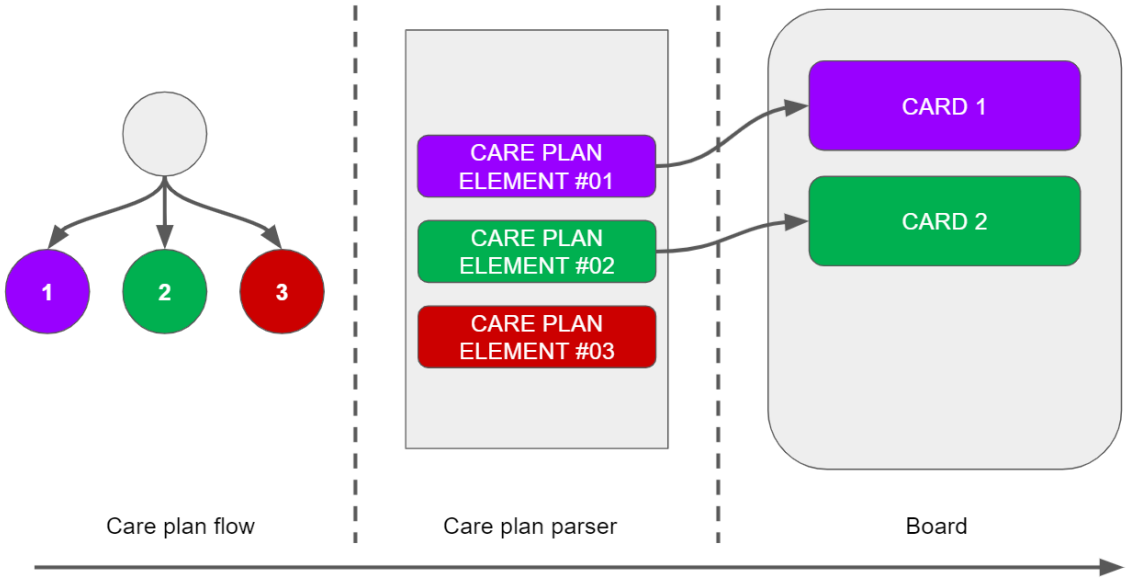
close to their deadline by accessing the agenda (Figure 3.4).

3.1 Board

We defined a board as a set of cards ordered by severity (defined in Table 3.3) and a deadline as a tiebreaker. It is here where patients interact with care plans created by the HCP. In the board, each card is generated according to a flow, and its content depends on which care plan element generated it (as shown in Figure 3.5). It is worth mentioning that the use of a care plan element does not always generate a card, as it depends on its semantics.

The board contains all care plan elements that are available for being completed by the patient. They do not need to be completed at once, and the patient can complete them in his/her time (except for care plan elements with an end date). Consequently, the patient goal is to keep its board empty (once a card is completed, it is removed from the board), meaning that the patient's treatment is up to date.

Figure 3.5: Board generation



Source: The author

3.2 Agenda

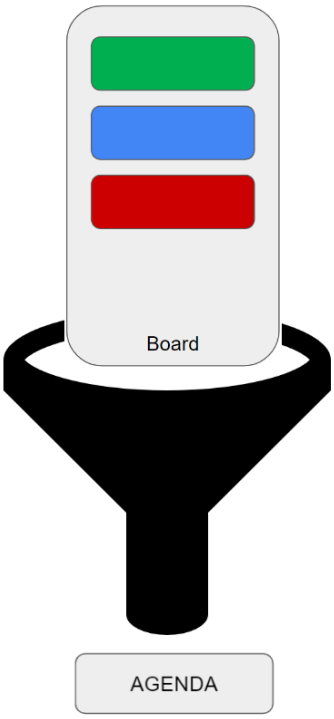
Agenda (Figure 3.6) helps patients to not lose care plan element deadlines. It is a simpler version of the board, where only those elements with very close deadlines are shown. Its purpose is to help patients to prioritize those elements that are more important (high severity and those that are close to deadline).

3.3 Progress

When a patient completes a card on the board, a progress item is generated (Figure 3.7). We defined a progress item as a set of information containing the finishing date and, optionally, patient input data, related to a care plan element. For example, if a question has been made, by HCP, to a patient and this patient answered it, his/her answer would be in a progress item. Another example is when a medication has been taken: in this case, the progress item would contain only when this element was finished by the patient.

Patients can see their progress and check how far they are from accomplishing all possible progress items. Progress is a set of progress items related to some care plan. It has two purposes: to keep a treatment history and to provide a progress measure, display-

Figure 3.6: The agenda idea is to filter the board and show only care plan elements close to their deadline



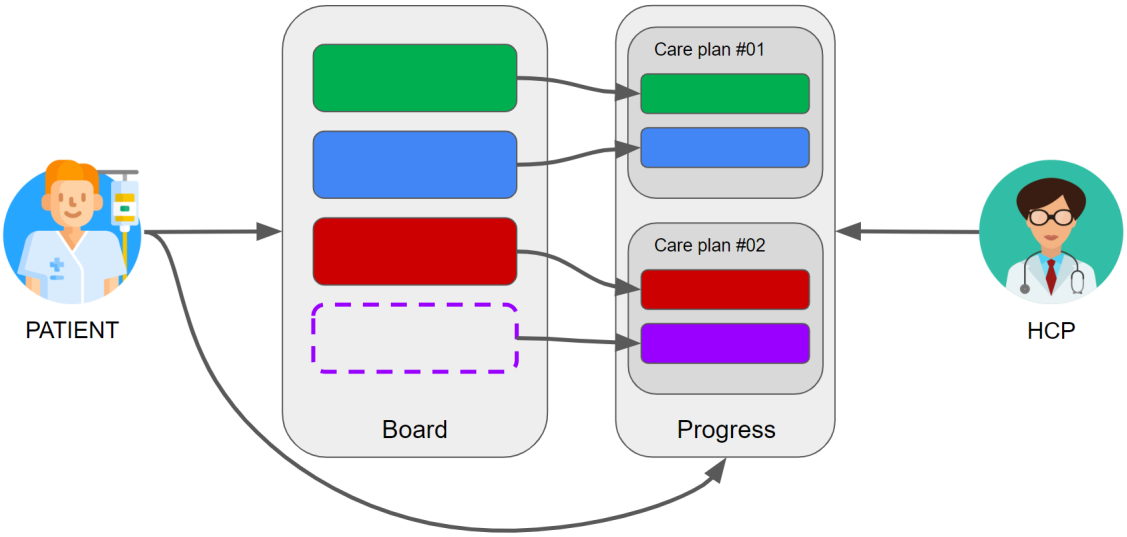
Source: The author

ing how many progress items were finished and comparing this with the total of possible progress items. Finally, HCP also have access to their patient's progress. HCP can see not only completed care plan elements, but also ongoing and late care plan elements.

3.4 Care plan flow

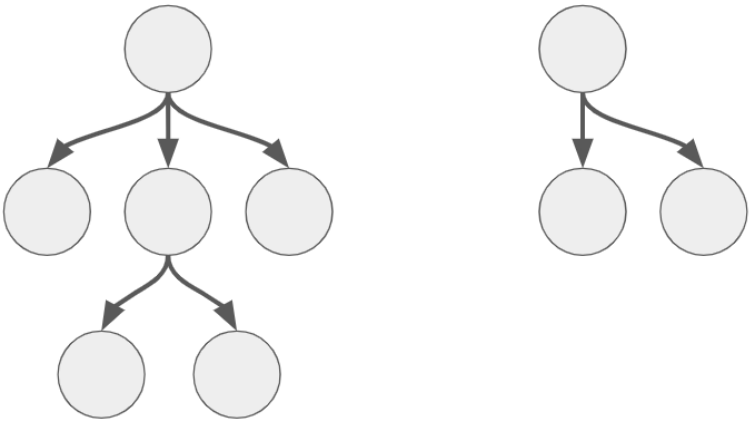
A flow (Figure 3.8) is a connected digraph (Section 2.2.2.1 and Section 2.2.2.2). It is composed of a set of nodes, which can be connected between them depending on their policy. A node policy defines the types of nodes that can be connected to it. This policy is created based on syntax, and its purpose is to avoid creating connections between nodes that do not make sense in a given context.

Figure 3.7: General overview of progress generation. Dashed lines represent a card that is not in the board anymore because it was already finished



Source: The author

Figure 3.8: A flow composed of two connected digraphs



Source: The author

3.4.1 Building a flow

In our approach, nodes represent care plan elements, and edges represent dependency. For example, let us assume there are two nodes: X and Y. If node X has an edge pointing to node Y, it means that node Y cannot be executed before node X. As nodes are care plan elements, an edge represents an ordering of which care plan element must be executed before another can be executed.

3.4.2 Care plan elements

We defined care plan elements as an element representing some item extracted from NIC (Section 2.1.3.2). It is composed of a name, description, type, color, icon, shape, input list, output list, content type and parameters (Tables 3.1 to 3.4). This structure defines the minimum information needed for the approach to work. The type of an element can be defined as: begin, conditional, periodic or non-periodic, and each of them are further explained in the following sub-subsections. Note that there are two parameter type: basic (that are always present), and specific (defined according to the content type).

Parameters can be classified in two groups: basic, that must be in all periodic and non-periodic care plan elements, and specific, that are defined according to the content type field. Basic parameters are shown in Table 3.3, while the specific parameters are defined in Table 3.4.

3.4.2.1 Begin element

The begin element indicates the beginning of the care plan flow. It is the root of the flow and contains only two parameters: start and end date. The first indicates when the care plan starts, and the second when it ends. Note that a care plan element may not have a defined end, being active indefinitely.

3.4.2.2 Conditional element

The conditional element allows choosing between two flows based on some condition (one if it is true and another if it is false). For example, if the patient is feeling pain, the conditional element can choose a flow that recommends taking some medication.

Table 3.1: Structure of care plan elements

Name	Description
Slug	Name that should be used in back-end systems for storing the element.
Name	Name that should be displayed in front-end systems when creating a care plan flow.
Description	Summary of what the element does.
Type	Indicates if is a begin, conditional, periodic or non-periodic element.
Color	Specifies a color that should be used in front-end systems when the element is used.
Icon	Specifies an icon that should be used in front-end systems when the element is used.
Shape	Specifies how the element should be displayed in front-end systems when the element is used (square, diamond, among others).
Input list	Indicates entry points of the element (useful when displaying the element in care plan flow). Should be a list of directions (left, top, among others).
Output list	Indicates output points of the element (useful when displaying the element in care plan flow). Should be a list of directions (left, top, among others).
Content type	As care plan elements can have different parameters besides the basic ones, it is necessary to specify the type of content. The possible types include text, unordered list, ordered list, book and form.
Parameters	Specifies fields that should be provided when a new instance of the element is created (for example when a HCP puts a begin element in a care plan flow). Parameter structure is specified in Table 3.2.

Source: The author

Table 3.2: Structure of each parameter item from parameters item from care plan elements

Name	Description
Slug	Name that should be used in back-end systems for storing the parameter.
Name	Name that should be displayed in front-end systems.
Description	Summary of the parameter.
Required	Specifies if parameter must be provided or not.
Type	Specifies parameter format (number, date, radio, among others).

Source: The author

Table 3.3: Basic parameters of care plan elements. Note that the slug field was omitted due to being used only for internal system use

Name	**Description**	**Type**
Name	Instantiated name.	text
Description	Short explanation about the instantiated care plan element.	text
Severity	How important the care plan element is.	It can be: • Very low; • Low; • Medium; • Critical; • Very critical.

Source: The author

Table 3.4: Specific parameters of care plan elements. Note that the slug field was omitted due to being used only for internal system use

Content type	**Parameters**
text	Content
unordered list	• Icons; • Elements.
ordered list	• Icons; • Elements.
book	Pages
form	Questions

Source: The author

3.4.2.3 Periodic care plan elements

Periodic care plan elements are characterized by the necessity to be generated periodically. These elements contain four additional parameters: frequency field, indicating the frequency type (like daily, spaced intervals, among others) along with its value, begin date and end date whose value can be undefined. For example, a medication that must be taken every 4 hours is a periodic care plan element with frequency type set as spaced interval and frequency value equal to 4.

3.4.2.4 Non-periodic care plan elements

Non-periodic care plan elements are all elements that do not have a frequency. They do not contain a frequency field, and, once completed, they are not generated again. For example, an explanation showing how to follow the treatment does not need to be repeated after the first time the patient read it.

3.5 Care plan parser

After a flow is created, the resulting care plan digraph (Section 2.2.2.2) is parsed. In this parsing, each node is parsed using an algorithm for traversing or searching tree data structures, generating a result according to its semantics. This result may generate a card insertion on the board, depending on its logic. It is the care plan parser's responsibility to know every care plan element that the approach provides support for.

As there are two types of care plan elements (Section 3.4.2), the care plan parser is composed of two modules: periodic and non-periodic. The first is responsible for parsing periodic care plan elements (Section 3.4.2.3), and the other non-periodic care plan elements (Section 3.4.2.4). Both modules are further explained in the following subsections.

3.5.1 Non-periodic module

Non-periodic module parses non-periodic care plan elements (Section 3.4.2.4). It is simpler than the periodic module (Section 3.5.2) because it does not need to keep track of parsed care plan elements due to all elements that this module parses do not repeat after

some time.

3.5.2 Periodic module

Periodic module is responsible for handling periodic care plan elements (Section 3.4.2.3). This module should keep track of each periodic module in order to repeat care plan elements on the board appropriately. It will generate two results: one immediate and another one that will be generated according to the element frequency. To accomplish this, this module should annotate this element for that it can be generated in the future.

3.5.2.1 Periodicity Of The Periodic Element Problem

The periodic module is more complex than the non-periodic module because it has to deal with the Periodicity Of Periodic Element Problem (POPEP). This problem occurs when a periodic element has a periodic element as a child. It occurs because it is necessary to define with which frequency the child element will be generated: after the parent is completed for the first time or only when its parent is completed or in a way that is independent of the parent. Both alternatives are correct, and the decision of which alternative to use is made during the implementation. For example, consider two elements: X and Y, where Y is child of X. X has a frequency set as daily while Y has a frequency set to repeat every 2 hours. POPEP occurs when X is completed for the first time. Y is generated after X completion and is repeated every 2 hours. But, the next day, X is generated again (remember it has daily frequency), and the question is: what to do with Y? Does Y repetition should be canceled, returning to the initial state (where it is generated only after X completion) or it should continue its frequency generation independent of X? Again, there is no correct answer, and this is a point that must be decided in implementation (Section 4).

4 IMPLEMENTATION

In this section, we present the implementation of the approach described in Chapter 3, called Takere: a no-code platform for helping patients in their treatment according to a care plan developed by HCPs. Our approach is based on microservice architecture (explained in Section 2.2.8), where business logic is concentrated in a back-end system and it is responsible for providing logic for the front-end systems. Our approach is composed of one API server [1] and two clients (Figure 4.1): HCP [2] and patients [3]. Thus, Takere is divided into three parts: a front-end (Section 2.2.8.2) system for HCP (Takere-HCP), a front-end system for patients (Takere-Patient) and a back-end system (Section 2.2.8.1) connecting them (Takere-API), as shown in Figure 4.1. This architecture makes our systems more flexible for adding new care plan elements: whenever a new care plan element is added, only the back end needs to be modified and, once the component logic is defined in the back end, it will be propagated to the front-end systems (Figure 4.2). All these parts have been developed using internationalization (Section 2.2.7), and they will be explained in the next three sections.

4.1 Takere - HCP

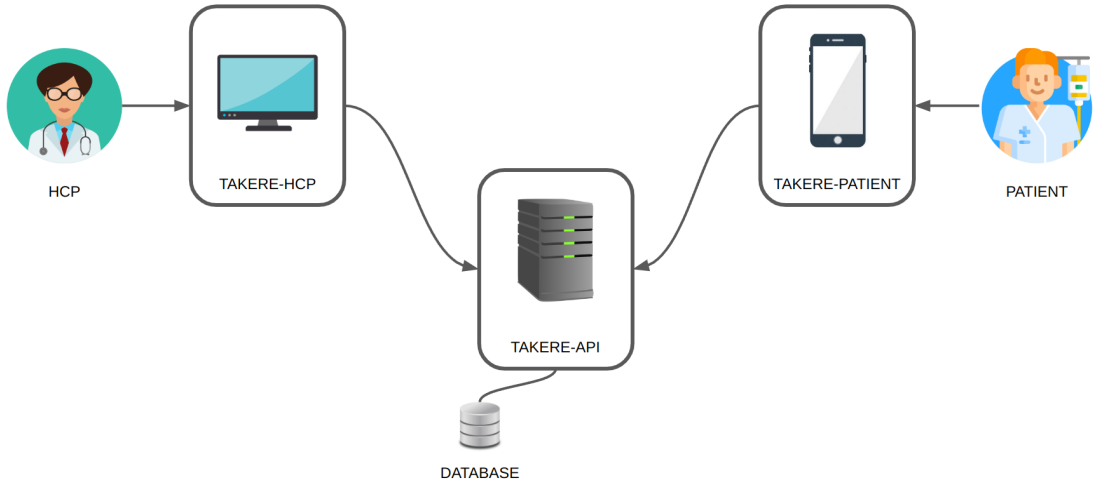
Takere for HCP is a system that allows the definition of a care plan along with monitoring patients' progress. It is developed using React [4], which is a framework for creating websites using component-oriented programming (Section 2.2.6). In the following subsections, we explain the system architecture, how the care plan flow is implemented and how HCP can monitor the progress of his/her patients.

4.1.1 Architecture

This front-end system has no information about the semantics of a care plan element and knows only its structure. The business logic is in Takere - API and it is explained in Section 4.2. This way, we simplify the front-end system - increasing its maintainability

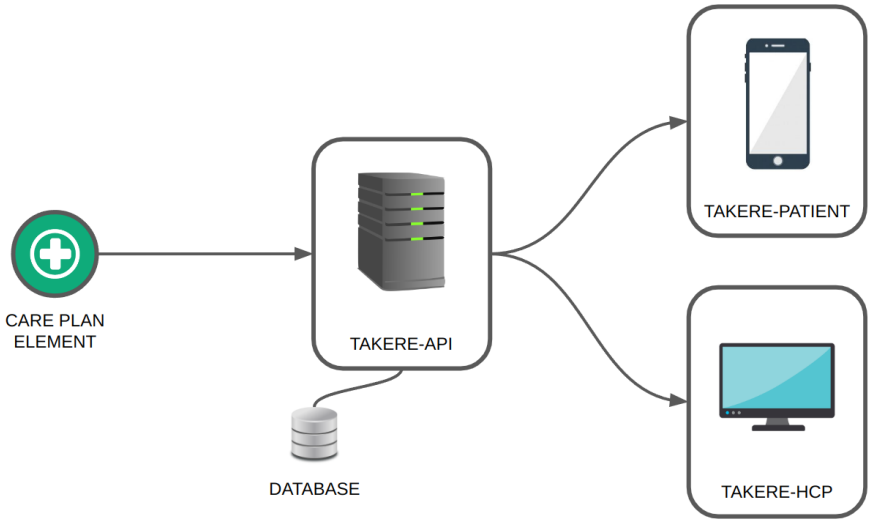
[1] https://github.com/takere/takere-api
[2] https://github.com/takere/takere-hcp
[3] https://github.com/takere/takere-patient
[4] https://reactjs.org

Figure 4.1: Takere systems



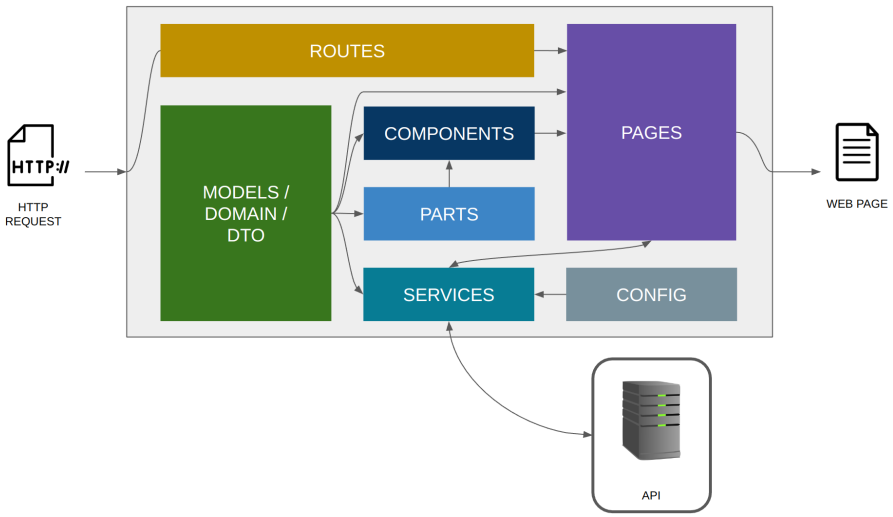
Source: The author

Figure 4.2: Process of adding a new care plan element (structured as described in Section 3.4.2) in the back-end system and its propagation to the front-end systems



Source: The author

Figure 4.3: Takere - HCP: Architecture



Source: The author

while making it more flexible: whenever new care plan elements are added, Takere - API will handle it, and Takere - HCP does not need to be modified. For that, it is necessary that Takere - HCP knows how nodes are structured to handle them. This structure is detailed in Section 4.2.2. Finally, architecture modules (Figure 4.3) are explained in Table 4.1. Note that the "Assets" module was omitted as it is used by all modules.

4.1.2 Care plan flow

We use drag-and-drop (Section 2.2.1) concept for building a care plan flow. The system shows care plan elements available for use in the flow. To include an element in the flow, the user needs to drag it and drop it in the flow (Figure 4.4). After that, it is necessary to connect these elements. Each connection is an arrow, indicating a dependency relation. For example, if an element Y should be generated only after an element X was completed, then the user must connect X with Y (in this order), resulting in the Figure 4.5.

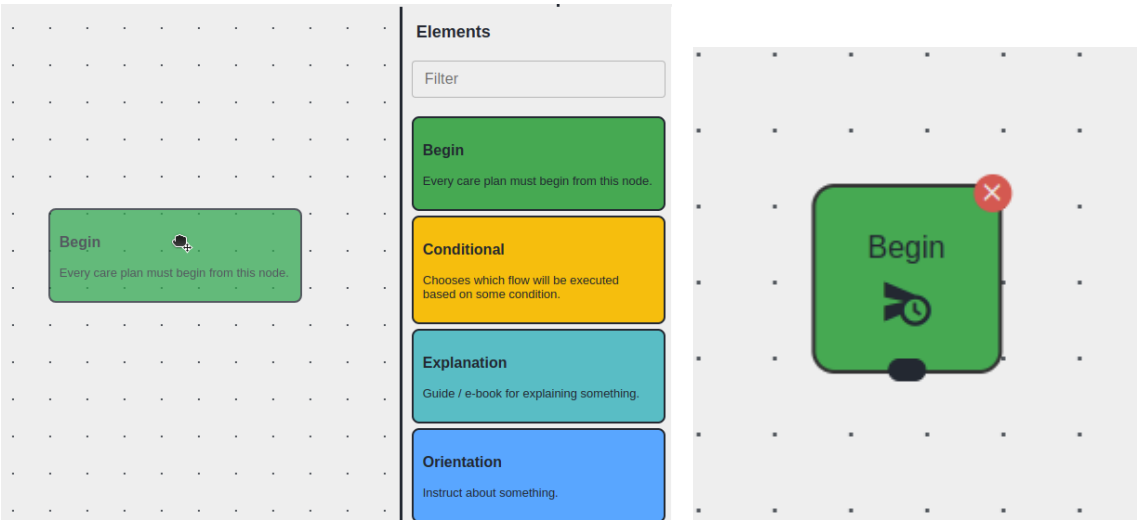
Each flow must begin with the 'Begin' element, which indicates when the care plan begins and when it ends (Figure 4.6). If the end date is unknown, it can be marked as 'undefined end', indicating the flow should stay active without a time to become disabled. Finally, each flow has a name, a description, and the email of the target patient.

Table 4.1: Architecture modules of Takere - HCP

Name	Description
Assets	Application static files (images, dictionaries, among others). It has not been placed in the Figure 4.3 to make the image clearer to understand
Config	Environment variables and configuration related files
Components	Collection of user interface components (like buttons and inputs) that can be used across various files in the project
Models / Domain / DTO	Data and database model files
Pages	Files responsible for showing information to users according to some endpoint
Parts	User interface components used for composing components
Routes	Files responsible for defining application endpoints and handling with them
Services	Files responsible for business logic

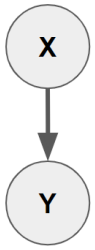
Source: The author

Figure 4.4: Drag-and-drop example using Begin element



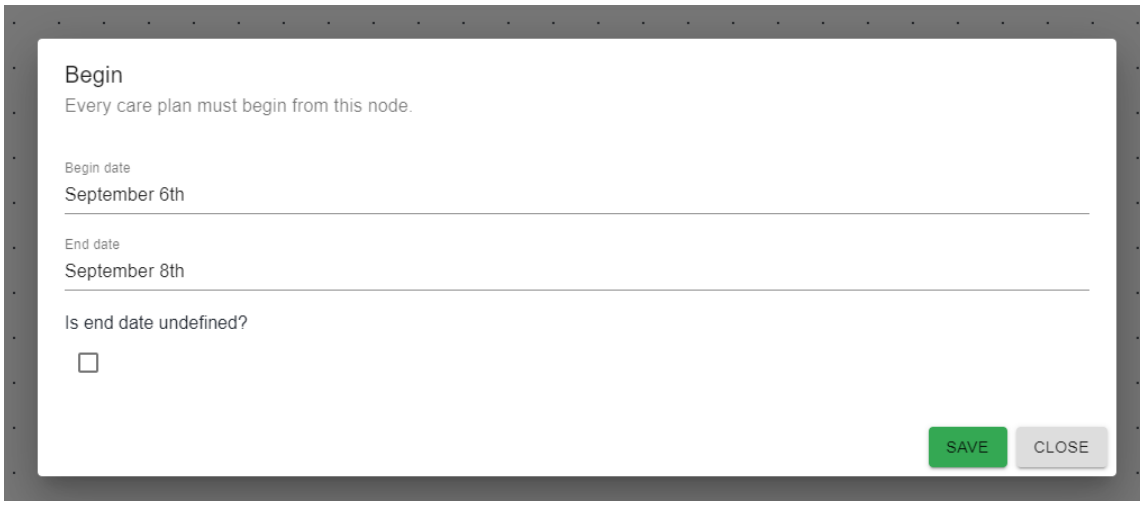
(a) Begin element being dragged (b) Begin element after being dropped

Figure 4.5: Two elements: X and Y, being Y a child of X



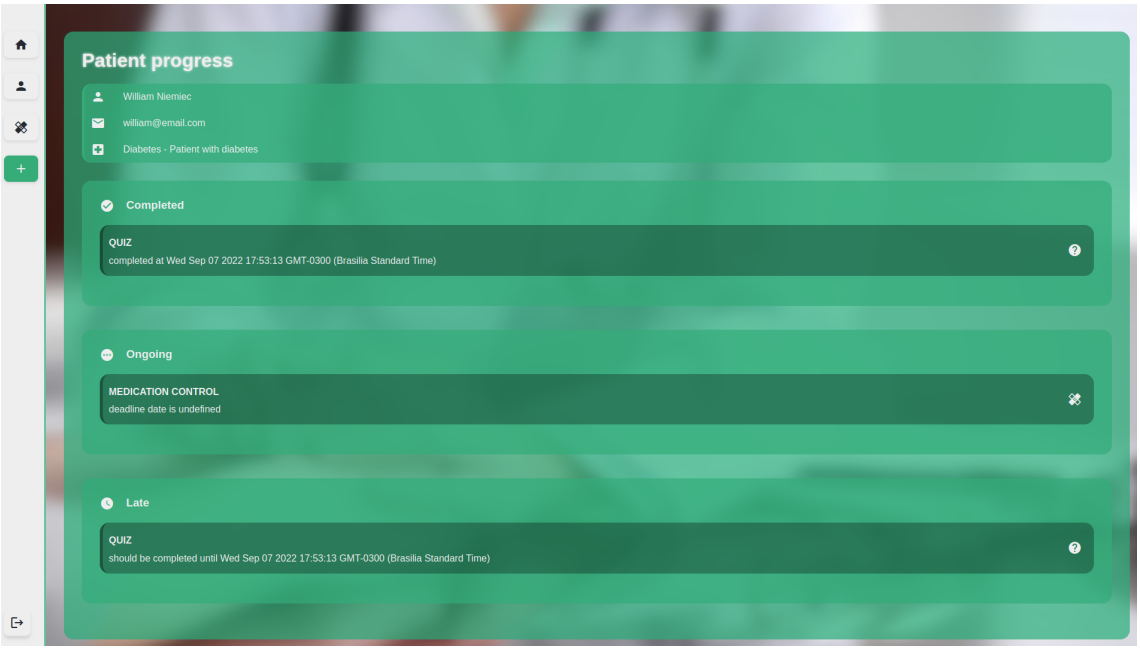
Source: The author

Figure 4.6: Begin element - configuration window



Source: The author

Figure 4.7: Care plan progress based on a created flow - HCP view

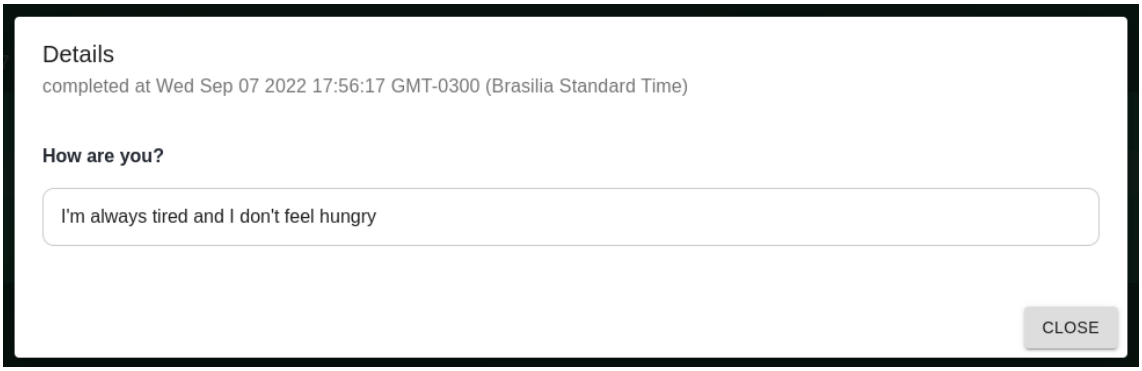


Source: The author

4.1.3 Monitoring patients

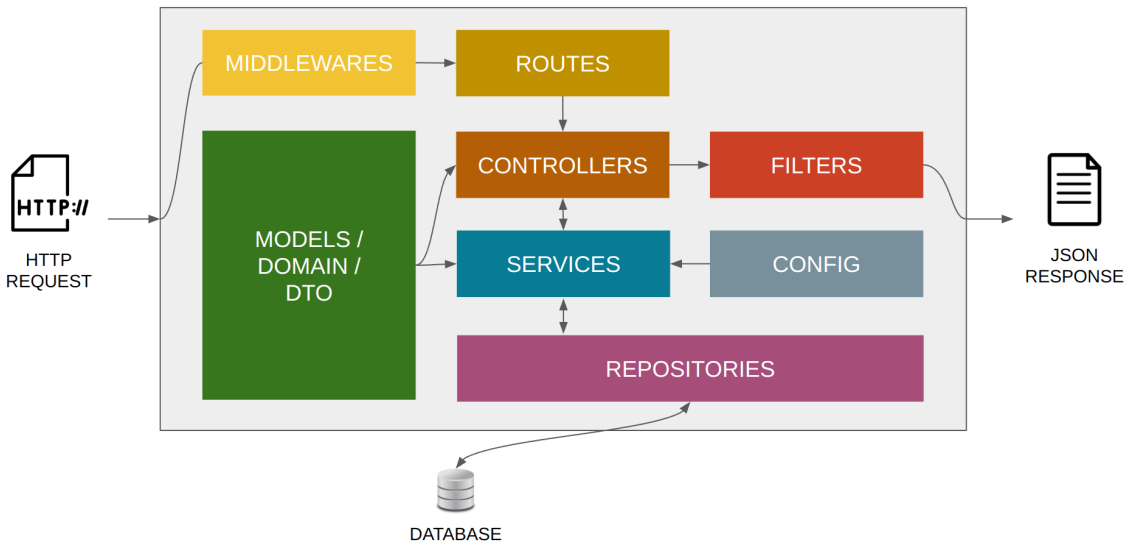
After a care plan flow is created, the target patient of the flow can start his/her treatment. When care plan elements are completed by him/her, results about this patient become available. This information can be monitored by HCP, seeing which elements have been completed, which are ongoing and those that are late (Figure 4.7). It is also possible to see patient input, if the element has inputs (Figure 4.8).

Figure 4.8: HCP view about one care plan element that has input fields



Source: The author

Figure 4.9: Takere - API: Architecture



Source: The author

4.2 Takere - API

This system is a RESTful API (Section 2.2.9). It is responsible for defining care plan elements logic, parsing care plan flows, and generating boards. It handles the database and also provides data for the other two Takere systems: HCP and Patient. Takere - API is built using NodeJS [5] due to its advantages compared to other server frameworks: its architecture is event-driven (Section 2.2.10) and non-blocking I/O (BANGARE et al., 2016; DEMASHOV; GOSUDAREV, 2019). In addition, NodeJS works well with JavaScript, which is the language used in the database. In the following sections, we detail the architecture and how we implement care plan elements (introduced in Section 3.4.2) and the care plan parser (Section 3.5). Finally, we describe the Takere database.

4.2.1 Architecture

This system is composed of eight modules. Seven of these modules have their flow shown in Figure 4.9, and all the eight modules are further explained in Table 4.2. Note that the "Assets" module was omitted as it is used by all modules.

[5] https://nodejs.org

Table 4.2: Architecture modules of Takere - API

Name	Description
Assets	Application static files (images, dictionaries, among others). It has not been placed in the Figure 4.9 to make the image clearer to understand
Config	Environment variables and configuration related files
Controllers	Files responsible for handling with requests and responses
Filters	Files called after the route handler and before a response goes out
Middlewares	Files called only before the route handler is called. It has access to the response object, but it does not have the result of the route handler
Models / Domain / DTO	Data and database model files
Repositories	Files responsible for persisting data
Routes	Files responsible for defining application endpoints and handling with them
Services	Files responsible for business logic

Source: The author

4.2.2 Care plan elements

Care plan elements are stored in JSON (Section 2.2.11.4) format and are structured as defined in Section 3.4.2. We chose JSON because the data structure of our database uses BSON (Section 2.2.11.5). The semantics of specific parameters are defined in Table 4.3. Note that we chose MaterialUI [6] library for providing icons when necessary. Also, icons are part of the care plan structure, and not of parameters, as it is static information. We considered the parameters as the elements of the list.

4.2.3 Care plan parser

Care plan parser is implemented using BFS (Section 2.2.2.3) algorithm. We chose this algorithm because it is more suitable to deal with the POPEP (Section 3.5.2.1), as we chose to continue the frequency generation of a node independently of its children. When a new care plan is generated, the care plan parser traverses the tree from its root and parses each node according to its logic. Besides its logic, it is necessary to configure

[6] mui.com/material-ui/material-icons

Table 4.3: Specific parameters semantics

Parameter name	**Semantics**
Content	Any text, including numbers and symbols
Pages	List of pages, where each page has a structure (HTML code) and a style (CSS code)
Questions	List of questions, where each question has a label (text), a type (defined in Table 4.4) and - optionally, a list of options, where each option has a label (name that is displayed) and a value (name that is used internally). The last should be used when type is radio, checkbox or select.

Source: The author

Table 4.4: Input types

Name	**Description**
Radio	Selects one option from a set.
Select	Selects one option from a list (it is required to provide the options as parameter, where each option has a label - name that is displayed - and a value - name that is used internally).
Checkbox	Selects multiple options from a set.
Single-line text	Short text.
Multi-line text	Long text.
Rich text	HTML text.
Book	List of pages, where each page has a structure (HTML code) and a style (CSS code).
Date	Selects a date from a calendar.

Source: The author

a scheduler if the parsed node is periodic.

Periodic nodes are generated according to some frequency. For that, we use a job scheduler (explained in Section 2.2.13) and create a job for generating each periodic node according to its frequency. Each job is stored in the database, and the job scheduler is responsible for managing these jobs and running them when necessary.

4.2.4 Database

As care plan elements can have different contents (Section 4.2.2), it is more suitable to use a non-relational database (explained in Section 2.2.14). We use MongoDB [7] because it works using JavaScript (as our server framework NodeJS) and its structure is more flexible than relational databases (CHAUHAN; BANSAL, 2017). Also, it has several advantages, such as storing data using BSON (Section 2.2.11.5) - being very efficient if data is managed in JSON - and being more efficient than some relational databases (GYŐRÖDI et al., 2015; PARKER; POE; VRBSKY, 2013). Finally, we created seven collections, which are further explained in Table 4.5.

4.3 Takere - Patient

Takere for patients is a system that allows patients to have access to their care plans. They can see its progress along with elements that they have to complete. It is developed using React Native, which is a mobile development framework for generating mobile applications using component-oriented programming (Section 2.2.6). It generates native applications for Android [8] and iOS [9] operating systems. In the following subsections we explain the system architecture, how the board (Section 3.1), agenda (Section 3.2) and progress components (Section 3.3) are implemented.

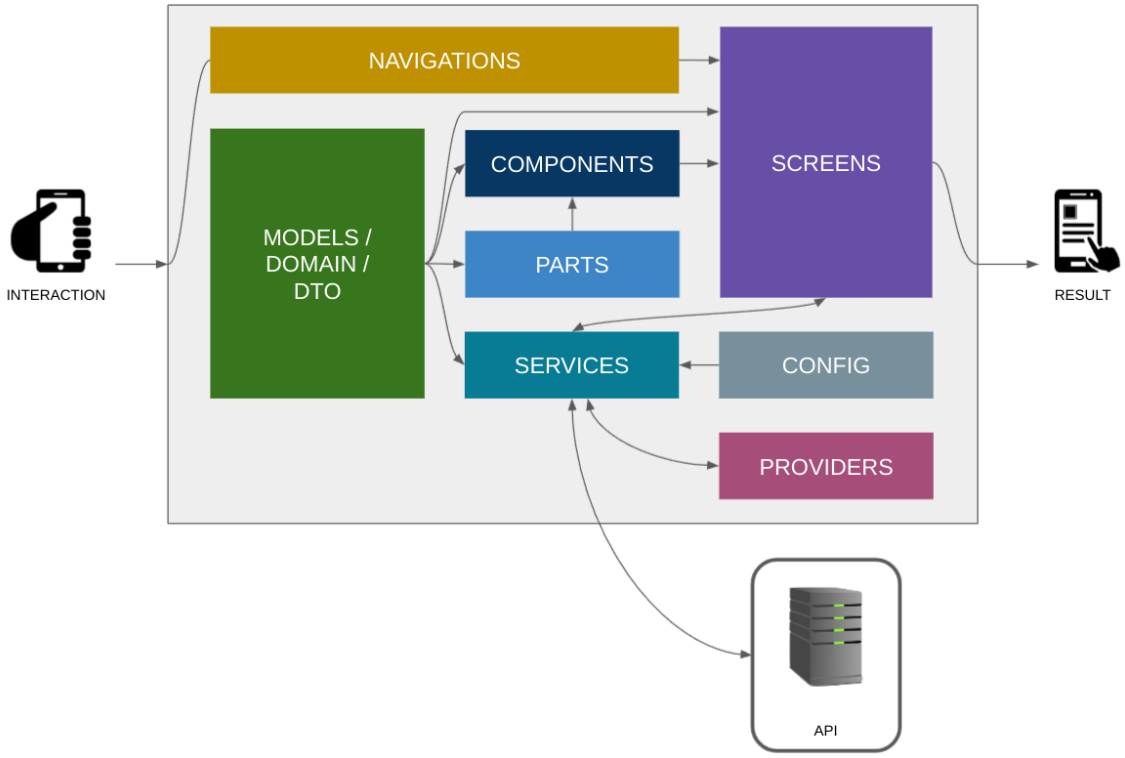
[7] https://www.mongodb.com
[8] www.android.com
[9] www.apple.com/ios

Table 4.5: Takere database. Note that "ObjectId" type is a reference to a record in another table

Collection	Description	Fields	Required
boards	Contains patient boards	id: ObjectId(boards)	No
		name: string	Yes
		description: string	Yes
		userEmail: string	Yes
		flow: ObjectId	Yes
		node: ObjectId(nodes)	Yes
		completed: ObjectId(completed)	No
edges	Contains edges used in each care plan flow	id: ObjectId(edges)	No
		source: ObjectId(nodes)	Yes
		target: ObjectId(nodes)	No
		animated: boolean	No
		flow: ObjectId(flows)	Yes
completed	Contains completed care plan elements of the board	id: ObjectId(completed)	No
		node: ObjectId(nodes)	Yes
		result: Object	No
flows	Contains created care plan flows	id: ObjectId(flows)	No
		author: ObjectId(users)	Yes
		name: string	Yes
		description: string	Yes
		userEmail: string	Yes
jobs	Contains jobs related to periodic care plan elements	id: ObjectId(jobs)	No
		name: string	Yes
		data: Object	No
		type: string	No
		priority: number	No
		nextRunAt: Date	No
		lastModifiedBy: Date	No
		lastRunAt: Date	No
		lastFinishedAt: Date	No
nodes	Contains nodes used in each care plan flow	id: ObjectId(nodes)	No
		type: string	Yes
		data: Object	Yes
		position: Object	Yes
		flow: ObjectId(flows)	Yes
users	Contains users of the system (patients and HCP)	id: ObjectId(users)	No
		firstName: string	Yes
		lastName: string	No
		password: string	Yes
		role: string	Yes
		email: string	Yes
		profileUrl: string	No

Source: The author

Figure 4.10: Takere - Patient: Architecture



Source: The author

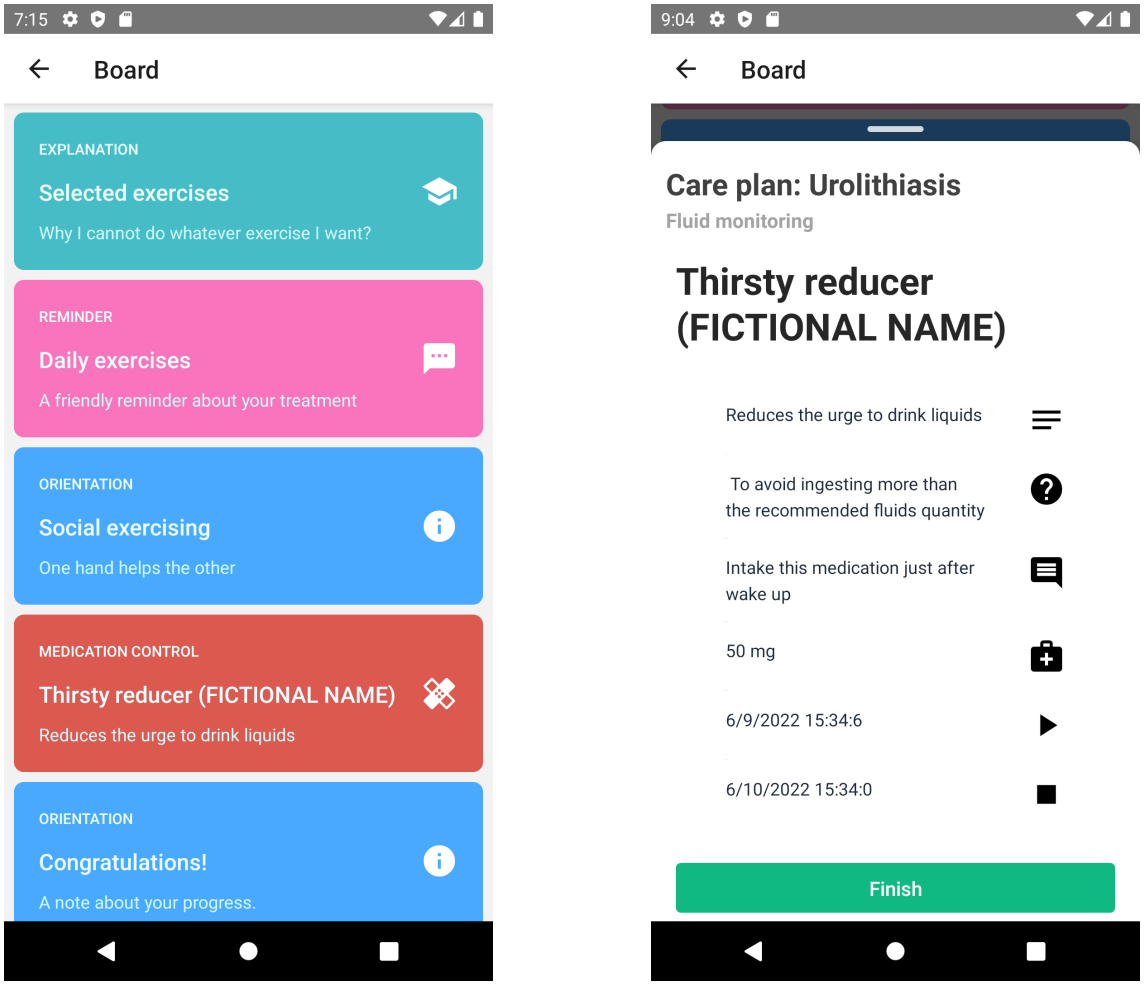
4.3.1 Architecture

Takere for patients system is composed of eight modules. Seven of these modules have their flow shown in Figure 4.10, and all the eight modules are further explained in Table 4.6. Note that the "Assets" module was omitted as it is used by all modules.

4.3.2 Board

All care plan elements the patient should complete are grouped on the board. It is composed of a set of cards, where each card represents a care plan element, and it has two parts: the front and the back, as shown in Figure 4.11. The first is composed of four elements: the care plan element name, a title, its description, and an icon. Also, card color is defined by the care plan element color. The back contains care plan information (name and description), title of care plan element (Figure 4.11a), the content of the care plan element (Figure 4.11b), and a finish button.

Figure 4.11: Takere - Patient: board



(a) Generated cards based on some care plan flow

(b) Card content of some care plan element from board

Source: The author

Table 4.6: Architecture modules of Takere - Patient

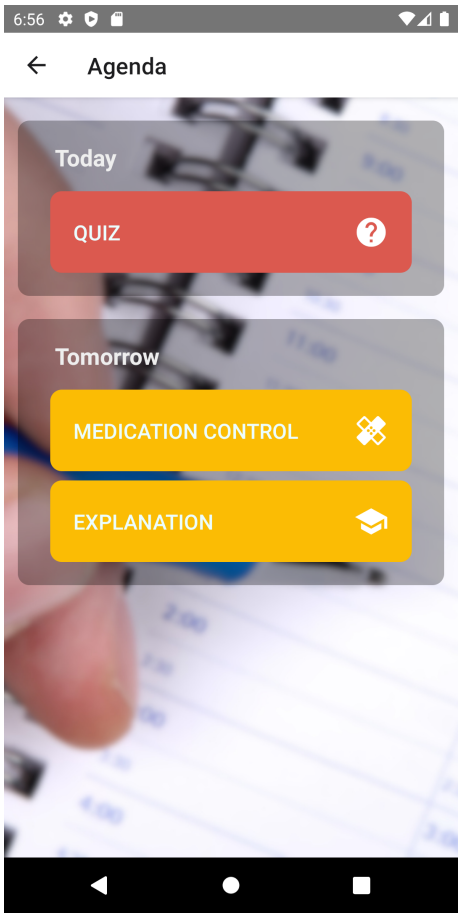
Name	Description
Assets	Application static files (images, dictionaries, among others). It has not been placed in the Figure 4.10 to make the image clearer to understand
Config	Environment variables and configuration related files
Components	Collection of user interface components (like buttons and inputs) that can be used across various files in the project
Models / Domain / DTO	Data and database model files
Navigations	Files responsible for defining application navigation routes
Parts	User interface components used for composing components
Providers	Files responsible for managing local data
Services	Files responsible for business logic
Screens	Files responsible for showing information to users according to some navigation route

Source: The author

4.3.3 Agenda

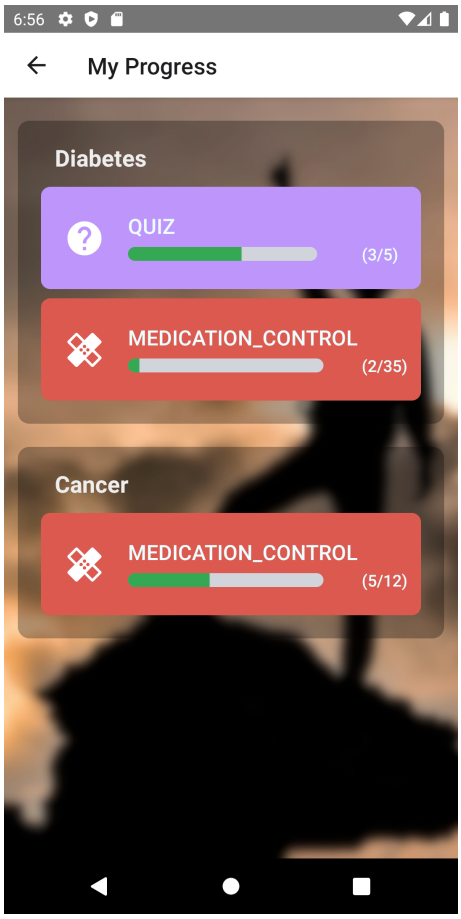
The agenda aims to highlight care plan elements that should be finished as soon as possible. For that, we use two strategies: grouping these elements by deadline day and using colors (Figure 4.12). The first approach uses two groups: "today" and "tomorrow", and each of them contains care plan elements with the deadline for today or tomorrow, respectively. We have chosen to not include other groups because the agenda goal is to display urgent care plan elements that should be finished, and showing more than necessary can reduce its impact and cause unnecessary anxiety to patients. The second approach use colors to highlight elements closest to the deadline: red for today and yellow for tomorrow. This approach is based on studies showing that colors can engage people to do tasks (VALDEZ; MEHRABIAN, 1994; STONE; ENGLISH, 1998; DAREJEH; SINGH, 2013).

Figure 4.12: Takere - Patient: agenda screen



Source: The author

Figure 4.13: Takere - Patient: progress screen



Source: The author

4.3.4 Progress

Progress screen shows patient progress in his/her care plans. For each care plan, it is shown a set of items containing progress information about each care plan element. This information includes the total of care plan elements of each type along with how many of them were completed, as shown in Figure 4.13.

5 VALIDATION METHODOLOGY

Our goal is to provide a no-code platform that makes it possible for HCP to instantiate mHealth applications based on patient care plans. In addition to the typical verification strategies used during implementation, we performed one internal validation experiment to evaluate whether the implemented solution fulfills its main objectives. To this end, we evaluated the following research questions (RQs):

> ***RQ1:*** *Is it possible to generalize actions/activities from NIC?*
>
> ***RQ2:*** *Is it possible to create parameterized computing elements from generalized actions/activities from NIC?*
>
> ***RQ3:*** *Is it possible to generate customizable mobile application from parameterized computing elements obtained from NIC?*

To perform the analysis, we selected two diseases. The chosen diseases were cancer and urolithiasis because more than half of the nurses we interviewed had many patients with these diseases. Consequently, we have had more information about cancer and urolithiasis than other diseases. For each disease, we looked for the most indicated NIC described in the literature. To do this, we performed a literature review on scientific work produced in the last twenty years that address cancer and urolithiasis interventions. Our inclusion criteria for scientific work were the following:

1. the paper addresses cancer or urolithiasis disease;
2. the paper proposes one or more interventions for cancer or urolithiasis disease;
3. the paper has been published in the last twenty years.

On the other hand, our exclusion criteria were the following:

1. it was not possible to read the complete version of the paper;
2. the paper proposes interventions that depend on HCP.

With this search, we obtained 28 papers. After reading their title, abstract, and conclusion, we applied the inclusion and exclusion criteria, leaving 16 works: ten related to cancer and six related to urolithiasis. All selected papers were read in full so that we understand in detail these interventions.

Based on this research, we identified the two most recommended interventions for each disease. Those interventions are indeed listed as NIC (Table 5.1), for the target diseases and were used in our experiments. Thus, we selected four NICs: two for each disease. For each NIC, we considered all activities described in (BUTCHER et al., 2018).

Table 5.1: Selected NICs for each disease

Disease	**NIC**
Cancer	Exercise promotion Self-esteem enhancement
Urolithiasis	Fluid monitoring Teaching: Prescribed diet

Source: The author

6 RESULTS

In this section, we present and analyze the results with respect to the established research questions. The analyses are presented in three parts: 1) the results of the analyses conducted to answer RQ1, 2) the results of the analyses conducted to answer RQ2 and 3) the results of the analyses conducted to answer RQ3.

6.1 Analyzing RQ1

For each selected NIC (Table 5.1), we analyzed each activity and grouped them by similarity. We consider an activity similar to another if they have the same purpose but with different information. For example, consider two activities: one is to monitor the patient weight and another is to check if the patient is dizzy. These activities are similar, due to being activities that the patient should provide some information as input for HCP (in this case, their weight and if they are dizzy or not). From this analysis, we identified five groups that are defined as care plan elements. Tables 6.1 to 6.5 present the activities selected to each selected NIC and the identification of the care plan elements that represent them. Finally, we define the semantics of these elements in Table 6.6. Note that activities that are nurse-dependent (Section 2.1.3.2) have been ignored since our work focuses on nurse-independent activities.

From this analysis, we identified five care plan elements:

- Explanation: aims to instruct the patient about something using rich text (Section 2.2.3.2);
- Medication control: helps patients to handle medications they need to use;
- Orientation: short message for assisting or motivating the patient about something;
- Quiz: aims to receive some input from the patient;
- Reminder: It is similar to the orientation, but it is a periodic element (Section 3.5.2). It also can notify the patient in different ways, and HCP can use the most suitable option according to their patients.

To avoid creating care plans that do not make sense, we defined which element can be connected to others (Table 6.6). Also, although we used a specific set of NIC, our

Table 6.1: Cancer disease - activities from "Exercise promotion" NIC

Activity	**Care plan element**	**Identifier**
Appraise individual's health beliefs about physical exercise	Quiz	EP01
Explore prior exercise experiences	Quiz	EP02
Determine individual's motivation to begin/continue exercise program	Quiz	EP03
Explore barriers to exercise	Quiz	EP04
Encourage verbalization of feelings about exercise or need for exercise	Orientation	EP05
Encourage individual to begin or continue exercise	Orientation	EP06
Assist in identifying a positive role model for maintaining the exercise program	Quiz	EP07
Assist individual to develop an appropriate exercise program to meet needs	Quiz	EP08
Assist individual to set short-term and long-term goals for the exercise program	Quiz	EP09
Assist individual to schedule regular periods for the exercise program into weekly routine	Quiz	EP10
Perform exercise activities with individual, as appropriate	Orientation	EP11
Include family/caregivers in planning and maintaining the exercise program	Ignored	EP12
Inform individual about health benefits and physiological effects of exercise	Orientation	EP13
Instruct individual about appropriate type of exercise for level of health, in collaboration with physician and/or exercise physiologist	Explanation	EP14
Instruct individual about desired frequency, duration, and intensity of the exercise program	Explanation	EP15
Monitor individual's adherence to exercise program/activity	Ignored	EP16
Assist individual to prepare and maintain a progress graph/chart to motivate adherence with the exercise program	Orientation	EP17
Instruct individual about conditions warranting cessation of or alteration in the exercise program	Explanation	EP18
Instruct individual on proper warm up and cool down exercises	Explanation	EP19
Instruct individual in techniques to avoid injury when exercising	Explanation	EP20
Instruct individual in proper breathing techniques to maximize oxygen uptake during physical exercise	Explanation	EP21
Provide reinforcement schedule to enhance individual's motivation (e.g., increased endurance estimation; weekly weigh-in)	Reminder	EP22
Monitor individual's response to exercise program	Ignored	EP23
Provide positive feedback for individual's efforts	Orientation	EP24

Source: The author

Table 6.2: Cancer disease - activities from "Self-esteem enhancement" NIC

Activity	**Care plan element**	**Identifier**
Monitor patient's statements of self-worth	Ignored	SEE01
Determine patient's locus of control	Ignored	SEE02
Determine patient's confidence in own judgment	Ignored	SEE03
Encourage patient to identify strengths	Quiz	SEE04
Assist patient to find self-acceptance	Explanation	SEE05
Encourage eye contact in communicating with others	Explanation	SEE06
Reinforce the personal strengths that patient identifies	Reminder	SEE07
Encourage patient to engage in self-talk and to verbalize positive affirmations daily to self	Orientation	SEE08
Provide experiences that increase patient's autonomy, as appropriate	Ignored	SEE09
Assist patient to identify positive responses from others	Explanation	SEE10
Refrain from negatively criticizing	Explanation	SEE11
Assist the patient to cope with bullying or teasing	Explanation	SEE12
Convey confidence in patient's ability to handle situation	Explanation	SEE13
Assist in setting realistic goals to achieve higher self-esteem	Quiz	SEE14
Assist patient to accept dependence on others, as appropriate	Orientation	SEE15
Assist patient to reexamine negative perceptions of self	Orientation	SEE16
Encourage increased responsibility for self, as appropriate	Quiz	SEE17
Assist patient to identify the effect of peer group on feelings of self-worth	Explanation	SEE18
Explore previous achievements of success	Reminder	SEE19
Explore reasons for self-criticism or guilt	Quiz	SEE20
Encourage the patient to evaluate own behavior	Quiz	SEE21
Encourage patient to accept new challenges	Orientation	SEE22
Reward or praise patient's progress toward reaching goals	Orientation	SEE23
Facilitate an environment and activities that will increase self-esteem	Explanation	SEE24
Assist patient to identify significance of culture, religion, race, gender, and age on self-esteem	Explanation	SEE25
Instruct parents on the importance of their interest and support in their children's development of a positive self-concept	Explanation	SEE26
Instruct parents to set clear expectations and to define limits with their children	Explanation	SEE27
Instruct parents to recognize children's accomplishments	Explanation	SEE28
Monitor frequency of self-negating verbalizations	Ignored	SEE29
Monitor lack of follow-through in goal attainment	Ignored	SEE30
Monitor levels of self-esteem over time, as appropriate	Ignored	SEE31
Make positive statements about patient	Orientation	SEE32

Source: The author

Table 6.3: Urolithiasis disease - activities from "Fluid monitoring" NIC (part A)

Activity	**Care plan element**	**Identifier**
Determine history of amount and type of fluid intake and elimination habits	Quiz	FM01
Determine possible risk factors for fluid imbalance (e.g., albumin loss state, burns, malnutrition, sepsis, nephrotic syndrome, hyperthermia, diuretic therapy, renal pathologies, cardiac failure, diaphoresis, liver dysfunction, strenuous exercise, heat exposure, infection, postoperative state, polyuria, vomiting, and diarrhea)	Quiz	FM02
Determine whether patient is experiencing thirst or symptoms of fluid changes (e.g., dizziness, change of mentation, lightheadedness, apprehension, irritability, nausea, twitching)	Quiz	FM03
Examine capillary refill by holding the patient's hand at the same level as their heart and pressing on the pad of their middle finger for 5 seconds, releasing pressure, and counting time until color returns (i.e., should be less than 2 seconds)	Ignored	FM04
Examine skin turgor by grasping tissue over a bony area such as the hand or shin, pinching the skin gently, holding it for a second and releasing (i.e., skin will fall back quickly if patient is well hydrated)	Ignored	FM05
Monitor weight	Quiz	FM06
Monitor intake and output	Quiz	FM07
Monitor serum and urine electrolyte values, as appropriate	Ignored	FM08
Monitor serum albumin and total protein levels	Ignored	FM09
Monitor serum and urine osmolality levels	Ignored	FM10
Monitor BP, heart rate, and respiratory status	Ignored	FM11
Monitor orthostatic blood pressure and change in cardiac rhythm, as appropriate	Ignored	FM12
Monitor invasive hemodynamic parameters, as appropriate	Ignored	FM13
Keep an accurate record of intake and output (e.g., oral intake, enteral intake, IV intake, antibiotics, fluids given with medications, NG tubes, drains, vomit, rectal tubes, colostomy drainage, and urine)	Ignored	FM14
Insure to measure all intake and output on all patients with intravenous therapy, subcutaneous infusions, enteral feedings, NG tubes, urinary catheters, vomiting, diarrhea, wound drains, chest drains, and medical conditions that affect fluid balance (e.g., heart failure, renal failure, malnutrition, burns, sepsis)	Ignored	FM15

Source: The author

Table 6.4: Urolithiasis disease - activities from "Fluid monitoring" NIC - part B

Activity	**Care plan element**	**Identifier**
Record incontinence episodes in patients requiring accurate intake and output	Quiz	FM16
Correct mechanical problems (e.g., kinked or blocked catheter) in patients experiencing sudden cessation of urine output	Ignored	FM17
Monitor mucous membranes, skin turgor, and thirst	Ignored	FM18
Monitor color, quantity, and specific gravity of urine	Ignored	FM19
Monitor for distended neck veins, crackles in the lungs, peripheral edema, and weight gain	Ignored	FM20
Monitor for signs and symptoms of ascites	Ignored	FM21
Note presence or absence of vertigo on rising	Quiz	FM22
Administer fluids, as appropriate	Medication control	FM23
Assure that all IV and enteral intake devices are operating at the correct rates, especially if not regulated by a pump	Ignored	FM24
Restrict and allocate fluid intake, as appropriate	Orientation	FM25
Consult physician for urine output less than 0.5 mL/kg/hr or adult fluid intake less than 2000 in 24 hours, as appropriate	Reminder	FM26
Administer pharmacological agents to increase urinary output, as appropriate	Medication control	FM27
Administer dialysis noting patient response, as appropriate	Ignored	FM28
Maintain accurate fluid container reference charts to assure standardization of container measurements	Ignored	FM29
Audit intake and output graphs periodically to ensure good practice patterns	Orientation	FM30

Source: The author

Table 6.5: Urolithiasis disease - activities from "Teaching: Prescribed diet" NIC

Activity	**Care plan element**	**Identifier**
Appraise the patient's current level of knowledge about prescribed diet	Quizz	TMD01
Appraise the patient's current and past eating patterns as well as preferred foods and current eating habits	Quizz	TMD02
Determine the patient's and family's perspectives, cultural backgrounds, and other factors that may affect the patient's willingness to follow prescribed diet	Ignored	TMD03
Determine any financial limitations that may affect food purchases	Ignored	TMD04
Instruct the patient on the proper name of the prescribed diet	Explanation	TMD05
Explain the purpose of diet adherence to overall health	Explanation	TMD06
Inform the patient about how long the diet should be followed	Explanation	TMD07
Instruct the patient about how to keep a food diary, as appropriate	Explanation	TMD08
Instruct the patient on allowed and prohibited foods	Explanation	TMD09
Inform the patient of possible drug and food interactions, as appropriate	Explanation	TMD10
Assist the patient to accommodate food preferences into the prescribed diet	Explanation	TMD11
Assist the patient in substituting ingredients to conform favorite recipes to the prescribed diet	Explanation	TMD12
Instruct the patient about how to read labels and select appropriate foods	Explanation	TMD13
Observe the patient's selection of foods appropriate to prescribed diet	Quiz	TMD14
Instruct the patient about how to plan appropriate meals	Explanation	TMD15
Provide written meal plans, as appropriate	Orientation	TMD16
Recommend a cookbook that includes recipes consistent with the diet, as appropriate	Orientation	TMD17
Reinforce information provided by other health care team members, as appropriate	Ignored	TMD18
Reinforce the importance of continued monitoring and changing needs that may require further alteration of dietary plan of care	Orientation	TMD19
Refer patient to dietitian, as appropriate	Ignored	TMD20
Include the family, as appropriate	Orientation	TMD21

Source: The author

analysis can be used for other sets. For that, the first step is to discard those activities whose performance depends on the participation of HCP. After that, it is necessary to group activities by similarity, i.e., those who have the same goal.

6.2 Analyzing RQ2

Once we identified care plan elements, we analyzed if it is possible to specify parameters for them. For that, we analyzed activities mapped to the same care plan element and identified which parameters can be extracted in order to subsume these activities into the care plan element. Using the structure defined in Section 3.4.2, we describe the structure of each care plan element identified in Section 6.1. We start describing explanation (Table 6.7) and medication control elements (Table 6.8). Next we describe orientation element (Table 6.9). After that we show how quiz element is structured (Table 6.10). Finally, we describe the structure of the reminder element (Table 6.11). Note that we omitted the "slug" field because it is only important for the internal handling of the data. We also omitted the following parameters as they are present in all periodic elements: frequency, begin, and end date. It is worth mentioning that we omitted the severity parameter too, as it is present in all elements.

To parameterize a new care plan element, it is necessary to analyze the activities that it refers to (as seen in Section 6.1). Next, we need to identify what is the difference between them, i.e., what is necessary for one to be equal to the other. After that, we have to typify the identified parameters as text, number, or another type. Finally, we need to define the remaining fields that a care plan element has (as defined in Section 3.4.2), including a name, description, type (periodic or non-periodic), icon, color, shape, input list, output list, and content type.

Thus, we parameterized all five care plan elements defined in RQ1. Tables 6.7 to 6.11 show the identified parameters. This information is structured in JSON and stored in Takere - API, following the structure previously described in Section 3.1.

6.3 Analyzing RQ3

In this section, we build a case study using the care plan elements defined in Section 6.1 and Section 6.2. First, we define the personas of our case study and build a care

Table 6.6: All possible connections of elements

Element	**Can be connected to**
Begin	Explanation Medication control Orientation Quiz Reminder
Conditional	Conditional Explanation Medication control Orientation Quiz Reminder
Explanation	Explanation Medication control Orientation Quiz
Medication control	Conditional Explanation Medication control Orientation Quiz Reminder
Orientation	Conditional Explanation Medication control Orientation Quiz Reminder
Quiz	Conditional Explanation Medication control Orientation Quiz Reminder
Reminder	Conditional Explanation Medication control Orientation Quiz Reminder

Source: The author

Table 6.7: Structure of the explanation element

Name	**Value**
Name	Explanation
Description	Guide / e-book for explaining something.
Type	non-periodic
Color	#46bdc6
Icon	school
Shape	square
Input list	top
Output list	bottom
Content type	book
Parameters	

Name	**Description**	**Required**	**Type**
Name	What's the subject?	true	text
Description	This explanation is about...	true	text
Pages	Explanation content.	true	List of: • Structure: page content in HTML; • Style: page style in CSS (it is optional).

Source: The author

Table 6.8: Structure of the medication control element

Name	**Value**
Name	Medication control
Description	Instruct about some medication and how to use it correctly.
Type	periodic
Color	#db594f
Icon	healing
Shape	square
Input list	top
Output list	bottom
Content type	unordered list
Parameters	

Name	**Description**	**Required**	**Type**
Name	Medication name.	true	text
Description	This medication is about...	true	text
Why	This medication is important because...	true	text
Notes	Extra information.	false	text
Dosage	Dosage along with its unit (ml, mg...)	true	text
Icons	Icons displayed for each parameter	false	list of MaterialUI icon names

Source: The author

Table 6.9: Structure of the orientation element

Name	**Value**
Name	Orientation
Description	Instruct about something.
Type	non-periodic
Color	#49a9ff
Icon	info
Shape	square
Input list	top
Output list	bottom
Content type	text
Parameters	

Name	**Description**	**Required**	**Type**
Name	What's the subject?	true	text
Description	This orientation is about..	true	text
Content	Orientation content	true	text

Source: The author

plan for them (Section 6.3.1). Next, we build a care plan flow of these care plans using Takere - HCP (Section 6.3.2). After that, we use Takere - Patient and complete a care plan element that has inputs and the patient provides some input to it (Section 6.3.3). Finally, we use Takere - HCP again and check the patient progress, showing ihis/her input (Section 6.3.4).

6.3.1 Care plans planning

For building care plans for our case study, we first need to describe the target patients (hereafter called personas). We create two personas: one with cancer and another with urolithiasis, and both feel engaged with pop-up messages along with daily reminders. We call P1 the persona with cancer and P2 the persona with urolithiasis. We assume that a nurse selected the intervention "Exercise promotion" for persona P1 and "Fluid monitoring" for persona P2.

Also, this nurse analyzed the P1 profile and selected the following activities: EP01, EP02, EP03, EP04, EP05, EP11, EP14, and EP22. In the same way, another nurse did the same with P2, selecting the following activities: FM06, FM07, FM22, FM23, FM25,

Table 6.10: Structure of the quiz element

<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>Name</td><td>Quiz</td></tr>
<tr><td>Description</td><td>Ask about something.</td></tr>
<tr><td>Type</td><td>periodic</td></tr>
<tr><td>Color</td><td>#be96fb</td></tr>
<tr><td>Icon</td><td>help</td></tr>
<tr><td>Shape</td><td>square</td></tr>
<tr><td>Input list</td><td>top</td></tr>
<tr><td>Output list</td><td>bottom</td></tr>
<tr><td>Content type</td><td>form</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Description</th><th>Required</th><th>Type</th></tr>
<tr><td>Name</td><td>What's the subject?</td><td>true</td><td>text</td></tr>
<tr><td>Description</td><td>This/these questions are about...</td><td>true</td><td>text</td></tr>
<tr><td>Questions</td><td>Questions to be asked</td><td>true</td><td>List of:
Label: question to be asked;
Type: answer type (same of Table 4.4);
Options: it is optional, and each answer option is composed of a label (information that is displayed) and a value (information that is stored).</td></tr>
</table>
</td></tr>
</table>

Source: The author

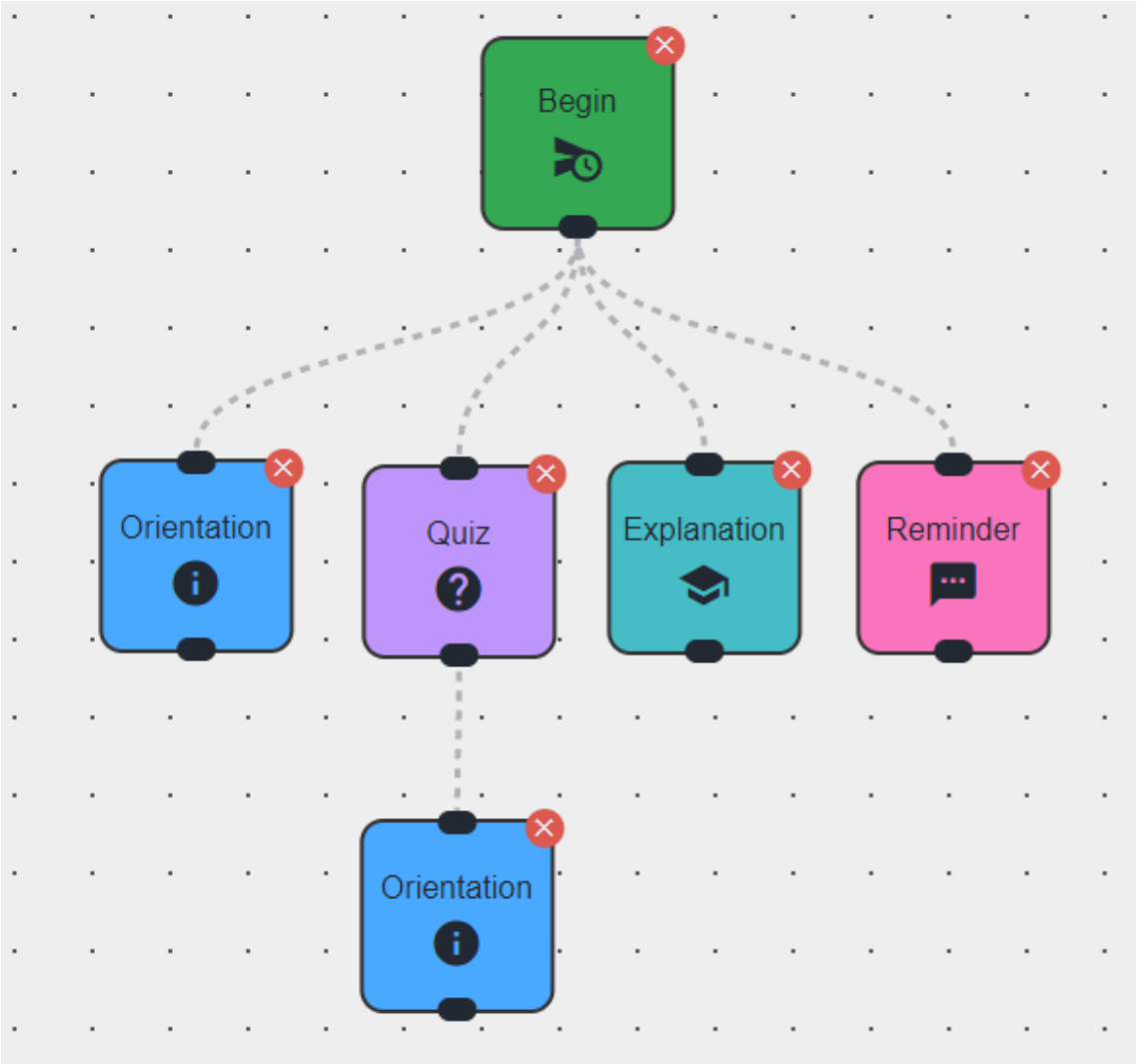
Table 6.11: Structure of the reminder element

Name	**Value**
Name	Reminder
Description	Remember about something.
Type	periodic
Color	#f974bc
Icon	textsms
Shape	square
Input list	top
Output list	bottom
Content type	text
Parameters	

Name	**Description**	**Required**	**Type**
Name	Medication name.	true	text
Description	This reminder is about...	true	text
Content	Reminder content	true	text
Notification type	How this reminder should be displayed?	true	List of options (pop-up message, email, text message or alert)

Source: The author

Figure 6.1: Takere - HCP: Cancer care plan flow



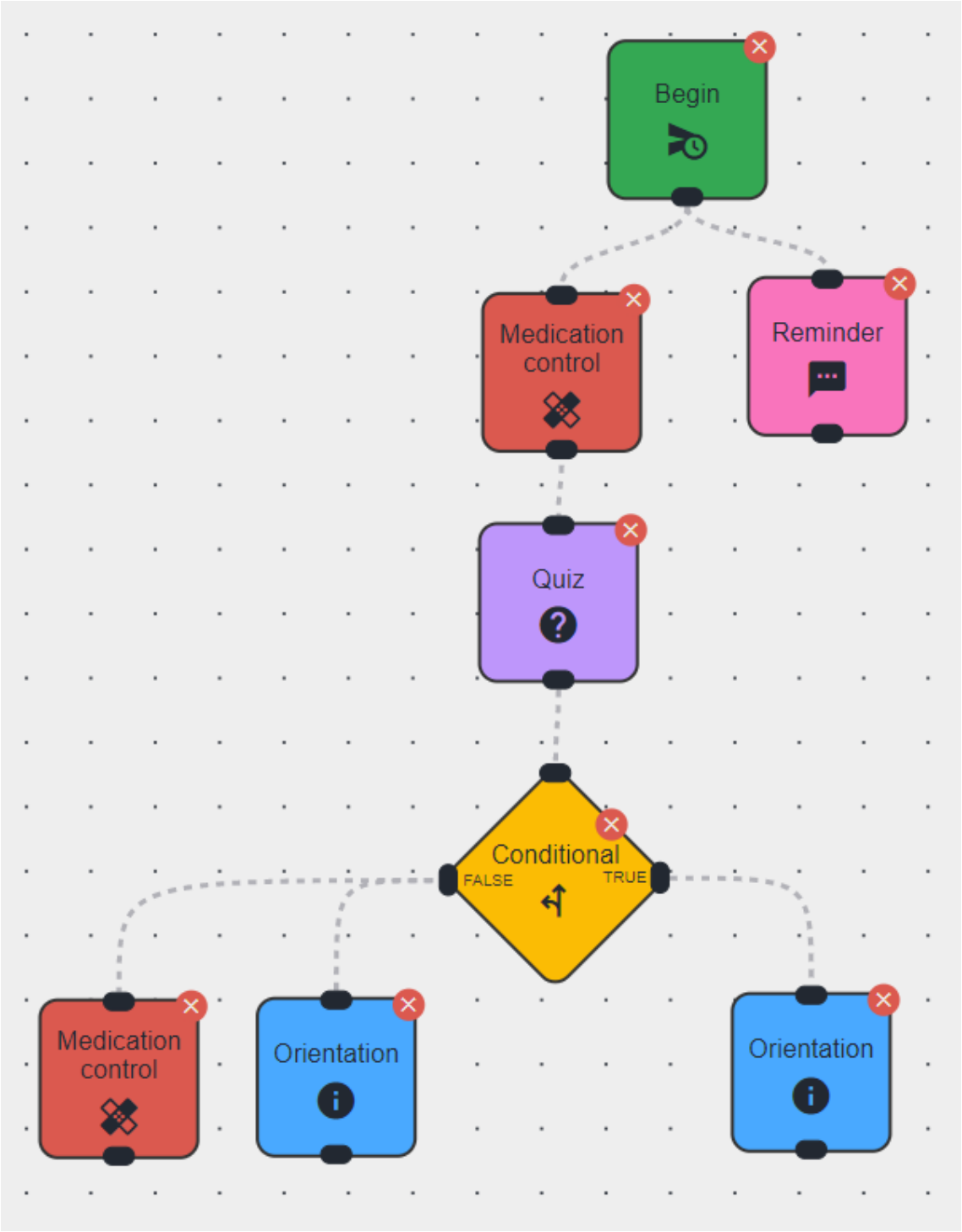
Source: The author

FM26, and FM27. The parameter definition for each of these activities are in Appendix B.

6.3.2 Building care plans in Takere - HCP

We build a flow for each care plan defined in Section 6.3.1. Activities EP01 to EP04 along with FM06, FM07, and FM22 are mapped to the quiz element. On the other hand, EP05, EP11, and FM25 are mapped to the orientation element. Also, the activity EP14 is mapped to the explanation element, FM26 to the reminder element, and the remaining to the medication control element. So, we build a care plan flow for the persona P1 (Figure 6.1) and another flow for the persona P2 (Figure 6.2).

Figure 6.2: Takere - HCP: Urolithiasis care plan flow



Source: The author

6.3.3 Interacting with care plans in Takere - Patient

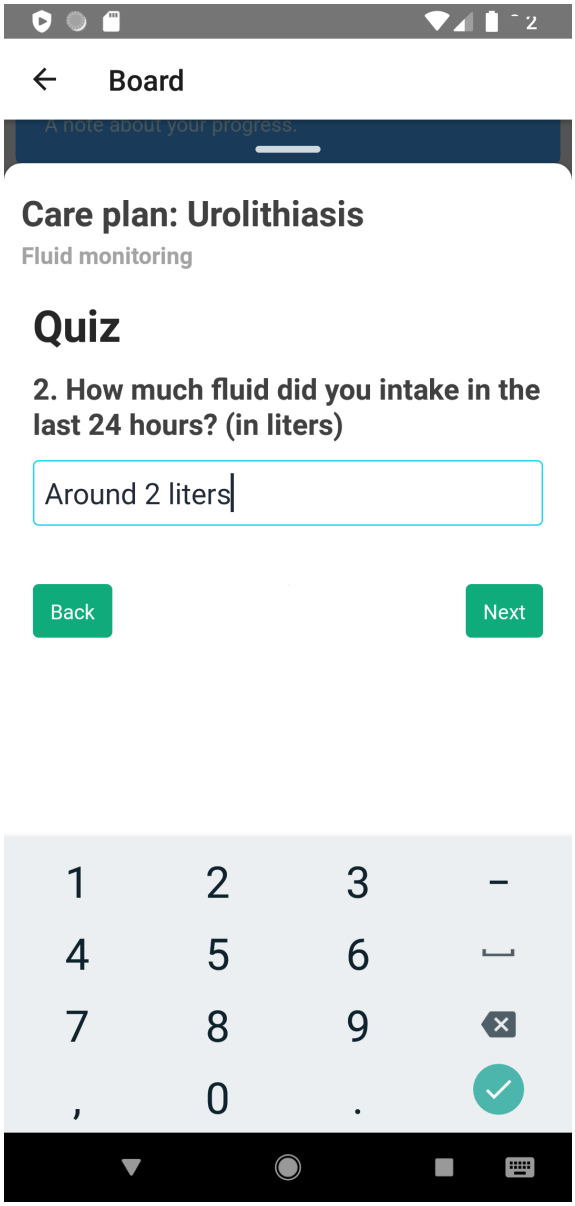
After a care plan flow is created, the target patient of the flow can interact with it through its board (as explained in Section 3.1). Depending on the care plan elements used in the flow, the patient can interact with them in different ways. Figure 6.3 shows the persona P2 answering the quiz defined in Table B.6. After that, the persona can see the care plan progress, checking which elements have been finished and how many remain to be finished.

6.3.4 Monitoring patients progress in Takere - HCP

After a patient finishes the interaction with some element, it is marked as finished, and both the persona and the HCP can see his/her progress. HCP can see the progress of his/her patients per care plan (Figure 6.4). If a care plan element has been finished and it has inputs, it is possible to see the patient's answer, as shown in Figure 6.5. With this information, HCP can change patient care plans according to their progress, adapting the care plan according to each patient.

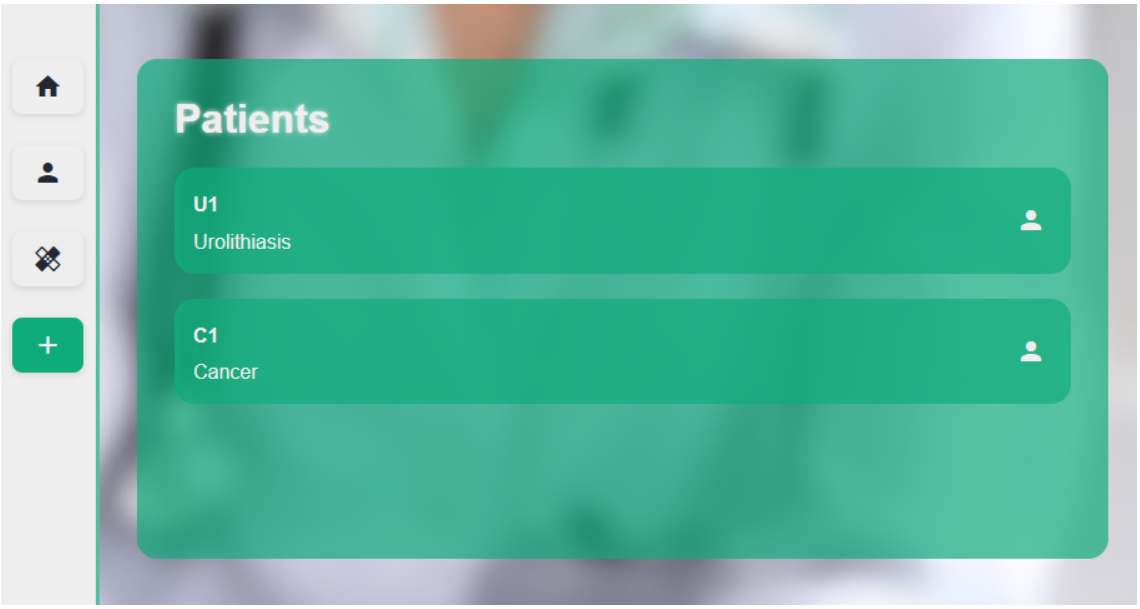
Thus, our approach can instantiate customized mobile applications from care plan elements. For that, care plan elements must be defined according to the structure defined in Section 3.4.2. After that, a care plan flow has to be created, it is necessary to choose which care plan elements will be used along with how they relate to each other. Finally, a flow is generated, and our approach instantiates a mobile application according to selected care plan elements along with how they are connected.

Figure 6.3: Takere - Patient: Patient answering a question of the quiz defined in Table B.6



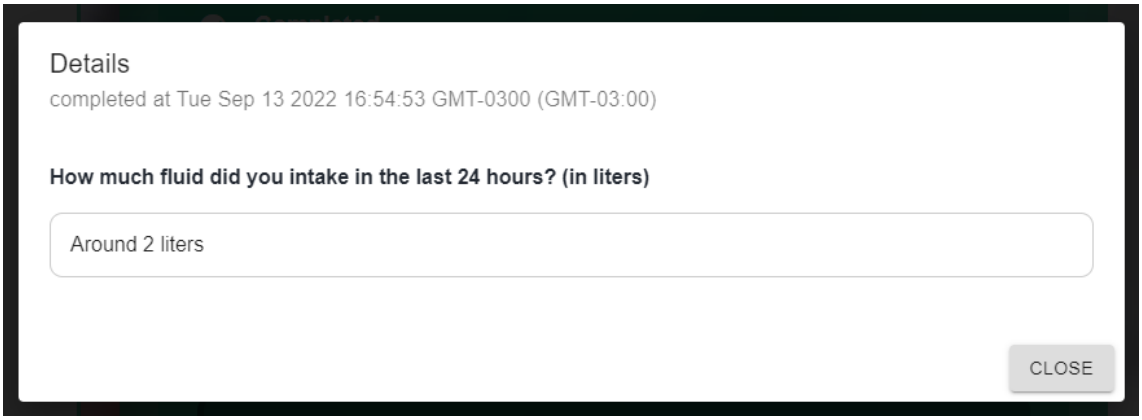
Source: The author

Figure 6.4: Takere - HCP: Patients by care plan



Source: The author

Figure 6.5: Takere - HCP: Patient answer for a question of the quiz defined in Table B.6



Source: The author

7 THREATS TO VALIDITY

In this section, we discuss the threats to validity and describe strategies used to mitigate them. We interviewed 14 nurses from different specializations and identified NANDA-NIC-NOC - a standard for care plan creation - and used it in our approach. However, it may be possible there are other standards or other elements we do not address in this work. To mitigate this risk, we developed front-end and back-end systems independent of NANDA-NIC-NOC. These systems were projected based on the care plan concept, and adding compatibility with other standards will only change care plan elements. This is not a problem, as the business logic of care plan elements is centralized in only one system: Takere - API. Consequently, adding new care plan elements in this system will reflect in all front-end systems, and it is not necessary to do any changes in those systems (Figure 4.2).

Next, we use a reduced set of NICs in our validation experiments. From this set, we derived seven care plan elements, as shown in Section 6.1. To these elements may not be enough to subsume all possible care plans. This risk is mitigated with microservice architecture (Section 2.2.8). Whenever a new care plan element needs to be created, the only system that must concern about it is Takere - API (the back-end system of our approach), while the remaining front-end systems do not need to be changed.

After that, our validation is internal, i.e., we do not validate our approach with users. Our solution is constructed based on interviews with HCP, but we do not know if it is suitable to be used for HCP. To mitigate this risk, we intend to do an external validation with HCP applied in a real case. With that, we expect to obtain feedback from HCP and evaluate our approach from the user perspective.

Finally, care plan element semantics defined in Table 6.6 was not validated. Consequently, the defined semantics may not subsumes some care plans. We mitigate this risk by centralizing semantics in one place: a single file in Takere - API system. So, changing care plan element semantics is a simple task: the only thing to do is to change this file and all front-end systems will be updated with the new semantics.

8 CONCLUSION AND FUTURE WORKS

This work proposes an approach to implement a no-code platform for HCP produce customized mHealth applications. The main goals of the proposed platform is instantiating mHealth applications for their patients according to their specificities. As HCP currently use a standard for building patient care plan (called NANDA-NIC-NOC), our approach uses this concept to define the basic components and to reduce learning time when using it. We expect to reduce non-adherence or partial adherence to treatments by patients, as mHealth applications are built based on their necessities. Also, we want to allow the creation of mHealth applications by HCP, reducing the dependency between HCP and IT teams, and, consequently, costs.

We identified four threats to validity. The first refers to the results obtained from our initial interviews. Next, we use a subset of NICs in our validation experiments, which may not be enough to subsume all possible care plans. After that, our validation is internal, and we do not evaluate our approach from the user perspective. Finally, the definition of how the care plan elements can interact one with another may not reflect reality. We have discussed in Chapter 7 strategies for dealing with each of these threats.

Future work include offline support, allowing the platform to be used when there is no internet connection, and gamification (Section 2.1.2), as studies have shown they increase engagement (MILLER; CAFAZZO; SETO, 2016; EL-HILLY et al., 2016; WANG et al., 2021). Another important aspect to be included is accessibility, as there may be patients with some disabilities, and our approach aims to be as inclusive as possible. Finally, we intend to allow that HCP to update care plans already created and also to include a wizard for assisting HCP to create care plans faster. Finally, we intend to validate our approach with HCP along with patients in order to obtain feedback and improve the solution.

REFERENCES

ADAM, B. M.; BESARI, A. R. A.; BACHTIAR, M. M. Backend server system design based on rest api for cashless payment system on retail community. In: IEEE. **2019 International Electronics Symposium (IES)**. [S.l.], 2019. p. 208–213.

ANDERSON, C. A.; KEENAN, G.; JONES, J. Using bibliometrics to support your selection of a nursing terminology set. **CIN: Computers, Informatics, Nursing**, LWW, v. 27, n. 2, p. 82–90, 2009.

APARICIO, A. F. et al. Analysis and application of gamification. In: **Proceedings of the 13th International Conference on Interacción Persona-Ordenador**. [S.l.: s.n.], 2012. p. 1–2.

AYKIN, N. **Usability and internationalization of information technology**. [S.l.]: CRC press, 2004.

BANGARE, S. et al. Using node. js to build high speed and scalable backend database server. In: **Proc. NCPCI. Conf**. [S.l.: s.n.], 2016. v. 2016, p. 19.

BHAT, U.; JADHAV, S. Moving towards non-relational databases. **International Journal of Computer Applications**, Citeseer, v. 1, n. 13, p. 40–47, 2010.

BRITO, N. M. R. Conjunto de dados mínimos de enfermagem para unidade de internação clínica. Universidade Estadual Paulista (UNESP), 2017.

BROWN, M. T.; BUSSELL, J. K. Medication adherence: Who cares? In: ELSEVIER. **Mayo clinic proceedings**. [S.l.], 2011. v. 86, n. 4, p. 304–314.

BURGER, C. D. et al. Treatment patterns and associated health care costs before and after treatment initiation among pulmonary arterial hypertension patients in the united states. **Journal of Managed Care & Specialty Pharmacy**, Academy of Managed Care Pharmacy, v. 24, n. 8, p. 834–842, 2018.

BUTCHER, H. K. et al. **Nursing interventions classification (NIC)-E-Book**. [S.l.]: Elsevier Health Sciences, 2018.

CARVALHO, R. V. Proposta de uma plataforma codeless para implementação de apps de promoção da saúde. 2021.

CHAUHAN, D.; BANSAL, K. Using the advantages of nosql: a case study on mongodb. **International Journal on Recent and Innovation Trends in Computing and Communication**, v. 5, n. 2, p. 90–93, 2017.

CLARK, T.; BARN, B. S. Event driven architecture modelling and simulation. In: IEEE. **Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE)**. [S.l.], 2011. p. 43–54.

COLLISON, S. **Beginning CSS web development: from novice to professional**. [S.l.]: Apress, 2007.

CROCKFORD, D. **The application/json media type for javascript object notation (json)**. [S.l.], 2006.

CRUZ, D. Processo de enfermagem e classificações. **Gaidzinski RR, Soares AVN, Lima AFC, Gutierrez BAO, Cruz DALM, Rogenski NMB, organizadores. Diagnóstico de Enfermagem na prática clínica. Porto Alegre: Artmed**, p. p25–37, 2008.

DAREJEH, A.; SINGH, D. A review on user interface design principles to increase software usability for users with less computer literacy. **Journal of computer science**, Science Publications, v. 9, n. 11, p. 1443, 2013.

DASKALOV, R.; PASHEV, G.; GAFTANDZHIEVA, S. Hybrid visual programming language environment for programming training. **TEM Journal**, v. 10, n. 2, p. 981–986, 2021.

DEMASHOV, D.; GOSUDAREV, I. Efficiency evaluation of node. js web-server frameworks. In: **MICSECS**. [S.l.: s.n.], 2019.

DIAS, B.; FREITAS, R.; MACULAN, N. Alocação de canais em redes celulares sem fio: algoritmos e modelos teóricos em grafos e escalonamento. **Anais do XLIV Simpósio Brasileiro de Pesquisa Operacional. Sociedade Brasileira de Pesquisa Operacional (SOBRAPO)**, 2012.

DIMATTEO, M. R. Variations in patients' adherence to medical recommendations: a quantitative review of 50 years of research. **Medical care**, JSTOR, p. 200–209, 2004.

DRAGONI, N. et al. Microservices: yesterday, today, and tomorrow. **Present and ulterior software engineering**, Springer, p. 195–216, 2017.

EISENMAN, B. **Learning react native: Building native mobile apps with JavaScript**. [S.l.]: " O'Reilly Media, Inc.", 2015.

EL-HILLY, A. A. et al. Game on? smoking cessation through the gamification of mhealth: A longitudinal qualitative study. **JMIR serious games**, JMIR Publications Inc., Toronto, Canada, v. 4, n. 2, p. e5678, 2016.

FAUGHT, E. et al. Nonadherence to antiepileptic drugs and increased mortality. **Neurology**, Wolters Kluwer Health, Inc. on behalf of the American Academy of Neurology, v. 71, n. 20, p. 1572–1578, 2008. ISSN 0028-3878. Available from Internet: <https://n.neurology.org/content/71/20/1572>.

FLANDERS, J.; JANNIDIS, F. Data modeling. **A new companion to digital humanities**, Wiley Online Library, p. 229–237, 2015.

FRAKES, W.; TERRY, C. Software reuse: Metrics and models. Association for Computing Machinery, New York, NY, USA, v. 28, n. 2, p. 415–435, jun 1996. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/234528.234531>.

GAALEN, A. E. van et al. Gamification of health professions education: a systematic review. **Advances in Health Sciences Education**, Springer, v. 26, n. 2, p. 683–711, 2021.

GATHRIGHT, E. C. et al. The impact of medication nonadherence on the relationship between mortality risk and depression in heart failure. **Health Psychology**, American Psychological Association, v. 36, n. 9, p. 839, 2017.

GOFFIN, P. et al. An exploratory study of word-scale graphics in data-rich text documents. **IEEE transactions on visualization and computer graphics**, IEEE, v. 23, n. 10, p. 2275–2287, 2016.

GUZZO, D. A. Desenvolvimento de solução informatizada para o registro de atendimento pré-hospitalar. 2017.

GYŐRÖDI, C. et al. A comparative study: Mongodb vs. mysql. In: IEEE. **2015 13th International Conference on Engineering of Modern Electric Systems (EMES)**. [S.l.], 2015. p. 1–6.

HAMARI, J.; KOIVISTO, J.; SARSA, H. Does gamification work?–a literature review of empirical studies on gamification. In: IEEE. **2014 47th Hawaii international conference on system sciences**. [S.l.], 2014. p. 3025–3034.

HAMINE, S. et al. Impact of mhealth chronic disease management on treatment adherence and patient outcomes: a systematic review. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 17, n. 2, p. e3951, 2015.

HARMS, H.; ROGOWSKI, C.; IACONO, L. L. Guidelines for adopting frontend architectures and patterns in microservices-based systems. In: **Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering**. [S.l.: s.n.], 2017. p. 902–907.

HERDMAN, T. H. **North American Nursing Diagnosis Association. Nursing Diagnoses: definitions & classification, 2009-2011**. [S.l.]: Oxford: Wiley-Blackwell, 2008.

HO, P. M. et al. Impact of Medication Therapy Discontinuation on Mortality After Myocardial Infarction. **Archives of Internal Medicine**, v. 166, n. 17, p. 1842–1847, 09 2006. ISSN 0003-9926. Available from Internet: <https://doi.org/10.1001/archinte.166.17.1842>.

INKPEN, K. M. Drag-and-drop versus point-and-click mouse interaction styles for children. **ACM Transactions on Computer-Human Interaction (TOCHI)**, ACM New York, NY, USA, v. 8, n. 1, p. 1–33, 2001.

JALENDER, B. et al. Drag and drop: influences on the design of reusable software components. **arXiv preprint arXiv:1103.1497**, 2011.

KENNEDY, A.; LEÓN, I. d. Css and accessibility. In: **Pro CSS for High Traffic Websites**. [S.l.]: Springer, 2011. p. 165–190.

KERSTEN, G. E.; KERSTEN, M. A.; RAKOWSKI, W. M. Software and culture: Beyond the internationalization of the interface. **Journal of Global Information Management (JGIM)**, IGI Global, v. 10, n. 4, p. 86–101, 2002.

KINCAID, R. K. Scale-free graphs for general aviation flight schedules. Citeseer, 2003.

KREBS, P.; DUNCAN, D. T. et al. Health app use among us mobile phone owners: a national survey. **JMIR mHealth and uHealth**, JMIR Publications Inc., Toronto, Canada, v. 3, n. 4, p. e4924, 2015.

KRUEGER, C. W. Software reuse. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 24, n. 2, p. 131–183, jun 1992. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/130844.130856>.

LUONG, T. V. et al. **Internationalization: Developing software for global markets**. [S.l.]: John Wiley & Sons, Inc., 1995.

MA, H. Graph-based multi-robot path finding and planning. **Current Robotics Reports**, Springer, p. 1–8, 2022.

MAAS, J.; MOORHEAD, S. **Nursing Outcomes Classification (NOC)**. [S.l.]: St Louis: CV Mosby, 2000.

MAO, Y. et al. Impact and efficacy of mobile health intervention in the management of diabetes and hypertension: a systematic review and meta-analysis. **BMJ Open Diabetes Research and Care**, BMJ Specialist Journals, v. 8, n. 1, p. e001225, 2020.

MARCOLINO, M. S. et al. The impact of mhealth interventions: systematic review of systematic reviews. **JMIR mHealth and uHealth**, JMIR Publications Inc., Toronto, Canada, v. 6, n. 1, p. e8873, 2018.

MARÉCHAUX, J.-L. Combining service-oriented architecture and event-driven architecture using an enterprise service bus. **IBM developer works**, v. 12691275, 2006.

MATIC, D.; BUTORAC, D.; KEGALJ, H. Data access architecture in object oriented applications using design patterns. In: IEEE. **Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No. 04CH37521)**. [S.l.], 2004. v. 2, p. 595–598.

MCLEAN, A. Software development trends 2021. **Canadian Journal of Nursing Informatics**, Canadian Journal of Nursing Informatics, Editor in Chief June Kaminski, v. 16, n. 1, 2021.

MECHAEL, P. N. The case for mhealth in developing countries. **Innovations: Technology, Governance, Globalization**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 4, n. 1, p. 103–118, 2009.

MENDELZON, A. O.; WOOD, P. T. Finding regular simple paths in graph databases. **SIAM Journal on Computing**, SIAM, v. 24, n. 6, p. 1235–1258, 1995.

MICHELSON, B. M. Event-driven architecture overview. **Patricia Seybold Group**, v. 2, n. 12, p. 10–1571, 2006.

MILLER, A. S.; CAFAZZO, J. A.; SETO, E. A game plan: Gamification design principles in mhealth applications for chronic disease management. **Health informatics journal**, SAGE Publications Sage UK: London, England, v. 22, n. 2, p. 184–193, 2016.

MONDAY, P. B. Implementing the data transfer object pattern. In: **Web Services Patterns: Java™ Platform Edition**. [S.l.]: Springer, 2003. p. 279–295.

MOORHEAD, S. A. The nursing outcomes classification. **Acta Paulista de Enfermagem**, SciELO Brasil, v. 22, p. 868–871, 2009.

MORALES, J.; RUSU, C. Usability perception of visual programming language: A case study. In: **CEUR Workshop Proceedings**. [S.l.: s.n.], 2020. v. 2747, p. 83–88.

MORRISON, J. P. Flow-based programming. In: **Proc. 1st International Workshop on Software Engineering for Parallel and Distributed Systems**. [S.l.: s.n.], 1994. p. 25–29.

MOZAFFARIAN, D. et al. Heart disease and stroke statistics—2016 update: a report from the american heart association. **circulation**, Am Heart Assoc, v. 133, n. 4, p. e38–e360, 2016.

MUSSI, C. M. et al. Visita domiciliar melhora conhecimento, autocuidado e adesão na insuficiência cardíaca: ensaio clínico randomizado helen-i. **Revista Latino-Americana de Enfermagem**, SciELO Brasil, v. 21, p. 20–28, 2013.

NADAREISHVILI, I. et al. **Microservice architecture: aligning principles, practices, and culture**. [S.l.]: " O'Reilly Media, Inc.", 2016.

NOCK, C. **Data access patterns: database interactions in object-oriented applications**. [S.l.]: Addison-Wesley Boston, 2004.

PARKER, Z.; POE, S.; VRBSKY, S. V. Comparing nosql mongodb to an sql db. In: **Proceedings of the 51st ACM Southeast Conference**. New York, NY, USA: Association for Computing Machinery, 2013. (ACMSE '13). ISBN 9781450319010. Available from Internet: <https://doi.org/10.1145/2498328.2500047>.

PAVLENKO, A. et al. Micro-frontends: application of microservices to web front-ends. **J. Internet Serv. Inf. Secur.**, v. 10, n. 2, p. 49–66, 2020.

PERRY, A. G. et al. **Canadian fundamentals of nursing**. [S.l.]: W. Ross MacDonald School Resource Services Library, 2013.

PEZOA, F. et al. Foundations of json schema. In: **Proceedings of the 25th International Conference on World Wide Web**. [S.l.: s.n.], 2016. p. 263–273.

PLODER, C. et al. The future use of lowcode/nocode platforms by knowledge workers–an acceptance study. In: SPRINGER. **International Conference on Knowledge Management in Organizations**. [S.l.], 2019. p. 445–454.

PRESTON, I. R. et al. Temporary treatment interruptions with oral selexipag in pulmonary arterial hypertension: Insights from the prostacyclin (pgi2) receptor agonist in pulmonary arterial hypertension (griphon) study. **The Journal of Heart and Lung Transplantation**, Elsevier, v. 37, n. 3, p. 401–408, 2018.

RAO, A.; BIHANI, A.; NAIR, M. Milo: A visual programming environment for data science education. In: IEEE. **2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)**. [S.l.], 2018. p. 211–215.

ROWLAND, S. P. et al. What is the clinical value of mhealth for patients? **NPJ digital medicine**, Nature Publishing Group, v. 3, n. 1, p. 1–6, 2020.

SAILER, M.; HOMNER, L. The gamification of learning: A meta-analysis. **Educational Psychology Review**, Springer, v. 32, n. 1, p. 77–112, 2020.

SARDI, L.; IDRI, A.; FERNÁNDEZ-ALEMÁN, J. L. A systematic review of gamification in e-health. **Journal of biomedical informatics**, Elsevier, v. 71, p. 31–48, 2017.

SHAH, N. B. et al. High rates of medication adherence in patients with pulmonary arterial hypertension: an integrated specialty pharmacy approach. **PLoS One**, Public Library of Science San Francisco, CA USA, v. 14, n. 6, p. e0217798, 2019.

SHEET, M. F. Pew research center: Internet. **Science & Tech**, v. 5, 2018.

SHERRY, J. L. et al. Video game uses and gratifications as predictors of use and game preference. In: **Playing video games**. [S.l.]: Routledge, 2012. p. 248–262.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Operating System Concepts, 10e Abridged Print Companion**. [S.l.]: John Wiley & Sons, 2018.

SILVEIRA, J. P. et al. Dinoapp: aplicativo para apoio ao tratamento de crianças com câncer. In: SBC. **Anais da VIII Escola Regional de Computação Aplicada à Saúde**. [S.l.], 2021. p. 22–25.

SIMSION, G.; WITT, G. **Data modeling essentials**. [S.l.]: Elsevier, 2004.

SOMMERVILLE, I. **Software Engineering**. 9. ed. Harlow, England: Addison-Wesley, 2010. ISBN 978-0-13-703515-1.

SOUSA, T. B. Dataflow programming concept, languages and applications. In: **Doctoral Symposium on Informatics Engineering**. [S.l.: s.n.], 2012. v. 130.

SOUZA, B. B. d. Modelo de balanceamento com multi-fluxos para aplicaçao em gerenciamento de tráfego aéreo. 2008.

SOUZA, E. N. de et al. A nurse-based strategy reduces heart failure morbidity in patients admitted for acute decompensated heart failure in brazil: the helen-ii clinical trial. **European Journal of Heart Failure**, Wiley Online Library, v. 16, n. 9, p. 1002–1008, 2014.

STONE, N. J.; ENGLISH, A. J. Task type, posters, and workspace color on mood, satisfaction, and performance. **Journal of Environmental Psychology**, Elsevier, v. 18, n. 2, p. 175–185, 1998.

SURYOTRISONGKO, H.; JAYANTO, D. P.; TJAHYANTO, A. Design and development of backend application for public complaint systems using microservice spring boot. **Procedia Computer Science**, Elsevier, v. 124, p. 736–743, 2017.

TAYLOR, H. et al. **Event-driven architecture: how SOA enables the real-time enterprise**. [S.l.]: Pearson Education, 2009.

TILKOV, S.; VINOSKI, S. Node. js: Using javascript to build high-performance network programs. **IEEE Internet Computing**, IEEE, v. 14, n. 6, p. 80–83, 2010.

TSAI, C.-Y. Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. **Computers in Human Behavior**, Elsevier, v. 95, p. 224–232, 2019.

VALDEZ, P.; MEHRABIAN, A. Effects of color on emotions. **Journal of experimental psychology: General**, American Psychological Association, v. 123, n. 4, p. 394, 1994.

VILLAMIZAR, M. et al. Infrastructure cost comparison of running web applications in the cloud using aws lambda and monolithic and microservice architectures. In: IEEE. **2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)**. [S.l.], 2016. p. 179–182.

VIOTTI, J. C.; KINDERKHEDIA, M. A survey of json-compatible binary serialization specifications. **arXiv preprint arXiv:2201.02089**, 2022.

WAGNER, C. et al. Classificação das intervenções de enfermagem (nic). In: **Classificação das intervenções de enfermagem (nic)**. [S.l.: s.n.], 2016. p. 610–610.

WALSH, C. A. et al. The association between medication non-adherence and adverse health outcomes in ageing populations: A systematic review and meta-analysis. **British Journal of Clinical Pharmacology**, v. 85, n. 11, p. 2464–2478, 2019. Available from Internet: <https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/bcp.14075>.

WANG, T. et al. The impact of gamification-induced users' feelings on the continued use of mhealth apps: A structural equation model with the self-determination theory approach. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 23, n. 8, p. e24546, 2021.

WONG, J.; DRIVER, M.; VINCENT, P. **Low-code development technologies evaluation guide**. [S.l.]: Gartner, Inc, 2019.

WU, J.-R. et al. Medication adherence in patients who have heart failure: a review of the literature. **Nursing Clinics of North America**, Elsevier, v. 43, n. 1, p. 133–153, 2008.

YAN, Z. The impacts of low/no-code development on digital transformation and software development. **arXiv preprint arXiv:2112.14073**, 2021.

ZAKRAOUI, J.; ZAGLER, W. A method for generating css to improve web accessibility for old users. In: SPRINGER. **International Conference on Computers for Handicapped Persons**. [S.l.], 2012. p. 329–336.

ZHANG, X. et al. Minimally-supervised structure-rich text categorization via learning on text-rich networks. In: **Proceedings of the Web Conference 2021**. [S.l.: s.n.], 2021. p. 3258–3268.

APPENDIX A — INTERVIEW QUESTIONS

We made 12 questions when interviewing HCP in our initial research. The questions are listed bellow.

1. What is your name?
2. What do you do for a living? (nurse, psychologist...)
3. What kind of patients do you treat? (related to ongoing treatment)
4. What do patients usually ask about treatment?
5. How do you keep track of patient care? Token? App? Hospital system? online?
6. What kind of information is essential to have from the patient?
7. What kind of information would it be good to have about the patient but not have it or have it but it is difficult to access?
8. How is patient care set up? What is its structure (in general terms)?
9. How does the patient see the treatment and know what to do? Do you use an app, prescription, table. . . ?
10. How do you deal with patient engagement? How do you keep them engaged to continue treatment?
11. What are the main challenges in treating patients?
12. Do you know other nurses who deal with treatments in continuous care? If so, could you provide their contact details? It will be of great help

APPENDIX B — CASE STUDY - PARAMETER DEFINITION

We start defining the parameters of the activities related to P1: EP01, EP02, EP03 and EP04 (Table B.1) followed by the activities EP05 (Table B.2), EP11 (Table B.3), EP14 (Table B.4) and EP22 (Table B.5). After that, we define the activities of the persona P2: FM06, FM07 and FM22 (Table B.6) followed by FM23 (Table B.10), FM25 (Table B.8), FM26 (Table B.8) and end with the definition of the activity FM27 (Table B.9).

Table B.1: Care plan for "Exercise promotion" NIC - parameters of EP01, EP02, EP03 and EP04

Parameter	**Value**
Name	Tell about you
Description	We want to know more about you
Questions	**Label** / **Type** / **Options** Have you ever exercised? Please, tell about your previous experience doing exercise. — Multi-line text Cite barriers to you exercise (if there are some) — Multi-line text What makes you feel motivate? — Multi-line text What do you think about doing exercises? — Multi-line text
Severity	Medium
Frequency	Only once
Begin date	September 10th, 2022
End date	October 10th, 2022

Source: The author

Table B.2: Care plan for "Exercise promotion" NIC - parameters of EP05

Parameter	**Value**
Name	Expressing feelings about exercising
Description	How to increase engagement.
Content	Try to tell someone about your exercises. Expressing your feelings increases engagement.
Severity	High

Source: The author

Table B.3: Care plan for "Exercise promotion" NIC - parameters of EP11

Parameter	**Value**
Name	Social exercising
Description	One hand helps the other
Content	Doing exercises with someone is amazing! Why don't you try?
Severity	Low

Source: The author

Table B.4: Care plan for "Exercise promotion" NIC - parameters of EP14

Parameter	**Value**
Name	Social exercising
Description	One hand helps the other
Pages	**Structure** / **Style** Why I cannot do whatever exercise I want? - Your exercises has selected according to your health; - Some exercises has not been selected because they can hurt or to harm you. /
Severity	Medium

Source: The author

Table B.5: Care plan for "Exercise promotion" NIC - parameters of EP22

Parameter	**Value**
Name	Daily exercises
Description	A friendly reminder about your treatment
Content	Hi! Sorry to disturb you, but I just want to remember you about your agenda: don't forget your daily exercises :)
Notification type	Pop-up message
Begin date	September 7th, 2022
End date	Undefined
Severity	Medium
Frequency	Daily

Source: The author

Table B.6: Care plan for "Fluid monitoring" NIC - parameters of FM06, FM07 and FM22

Parameter	**Value**
Name	Daily review
Description	Give us a feedback about you
Questions	**Label** / **Type** / **Options** Please, measure your weight and inform it. / numeric / How much fluid did you intake in the last 24 hours? (in liters) / numeric / How much urine did you output in the last 24 hours? (in liters) / numeric / Did you feel vertigo on rising? / radio / • Yes; • No.
Severity	Medium
Frequency	Daily
Begin date	September 7th, 2022
End date	Undefined

Source: The author

Table B.7: Care plan for "Fluid monitoring" NIC - parameters of FM23

Parameter	**Value**
Name	Thirsty reducer (FICTIONAL NAME)
Description	Reduces the urge to drink liquids
Why	To avoid ingesting more than the recommended fluids quantity
Notes	Intake this medication just after wake up
Dosage	50 mg
Severity	Low
Frequency	Daily
Begin date	September 7th, 2022
End date	Undefined

Source: The author

Table B.8: Care plan for "Fluid monitoring" NIC - parameters of FM25

Parameter	**Value**
Name	Fluids restriction
Description	You have to avoid some foods
Content	Please, avoid the following foods: all drinks and foods that are liquid at room temperature (like ice cubes gelatin Ice cream, yogurt...)
Severity	High

Source: The author

Table B.9: Care plan for "Fluid monitoring" NIC - parameters of FM26

Parameter	**Value**
Name	Fluids reminder
Description	Warning related to fluids intake limit
Content	Hi! Do not forget that you cannot intake more than 2L of fluids per day ;)
Notification type	Pop-up message
Begin date	September 7th, 2022
End date	Undefined
Severity	Low
Frequency	Daily

Source: The author

Table B.10: Care plan for "Fluid monitoring" NIC - parameters of FM23

Parameter	**Value**
Name	Urine increaser (FICTIONAL NAME)
Description	Increase urine output
Why	You need to use this medication because your fluids intake are higher than the permitted.
Notes	Intake this medication after you wake up.
Dosage	50 mg
Severity	Very critical
Frequency	Daily
Begin date	September 7th, 2022
End date	September 20th, 2022

Source: The author