SOEN 6471

ADVANCED SOFTWARE ARCHITECTURES
SUMMER 2018

Deliverable 1
iCare

Declaration

We, the members of the team, have read and understood the Fairness Protocol and the Communal Work Protocol, and agree to abide by the policies therein, without any exception, under any circumstances, whatsoever.

Team A

Maria Ahmed

Wasim Alayoubi

Marwah Alsadun

Shahram Aryanpour

Ramit Basra

Bikramjit Singh

Table of Contents 2

List of Tables

List of Figures

Part 1: Vision

1.1 Introduction

Recently mHealth tools have been more and more used to improve health care or medical services. In this project, we are employing mHealth informatics to provide a better care of children's health in Montreal. Therefore, we are developing a software architecture for a mobile application iCare that improves services of Montreal Children's Hospital for both patients and professionals.

iCare aims to provide effective and more accessible services related to Parents and nurses, as they are the immediate users of the App. Improving the quality of these services for those users will certainly better meet children health needs, as it's our ultimate goal.

Through our iCare App, parents can remotely access the booking appointment system of Montreal Children's Hospital and nursing assistance and feedback. The App will make parents more involved in the hospital's contributions toward children healthcare awareness. Nurses can efficiently manage home visits to take care of newborns and new moms. ICare also exploits mHealth resources such as Electronic Health Record for more effective and efficient care.

"Congrats! You are a new mommy in Montreal and your baby was born in our hospital, Probably You have so many questions about your baby, you want to know when are his vaccine schedules, you are seeking information about the right ways of breastfeeding or just simply want to make a doctor appointment. Have you ever thought about using your phone to assist you with your mommy duties? If you own a smartphone, iCare "New Mom" is the application you need. It will also help you to keep in touch with Nurses for Home based follow-ups and many other services"

This is how we will market our new healthcare mobile application "iCare for new Mom", which we are presenting in this document. The main objective of this project is to design, describe and evaluate an efficient, cost-effective, state of the art and platform independent healthcare mobile application. iCare aims to provide a mobile-operated system to allow remote patients (new parents) the access to immediate medical advice, digital record keeping, appointments booking and data sharing.

1.2 Motivation

Maternity is an umbrella term encompassing a continuum of care provided to the mother and child before, during and after birth. This includes prenatal/antenatal care (care during pregnancy), intranatal care (care during labour and delivery) and postnatal/postpartum care (care for the mother and newborn following birth). Since both mothers and infants undergo major changes throughout the perinatal period, maternity care entails a diverse range of health monitoring and care.

We focused our iCare app on postpartum care, as this is the time when both mothers and infants need a lot of care for their well-being. "According to a study by a team of doctors and researchers at the University of North Carolina Chapel Hill[18], it is found that new moms very often aren't aware of possible complications, are too uncomfortable to discuss their symptoms, and have no idea there are treatments that could help them. They get just one medical visit six weeks after birth, and that's often sadly not enough for the issues they're dealing with."

Postpartum Office Visits:
According to a survey[19], "among the 6% of mothers who did not have a postpartum office visit between 3 and 8 weeks after birth, the leading reasons were that "I felt fine and didn't need to go," (35%), followed by "too hard to get to office" (14%) and "didn't have insurance (10%). Mothers reported traveling an average of twelve miles each way for their maternity care office visits".
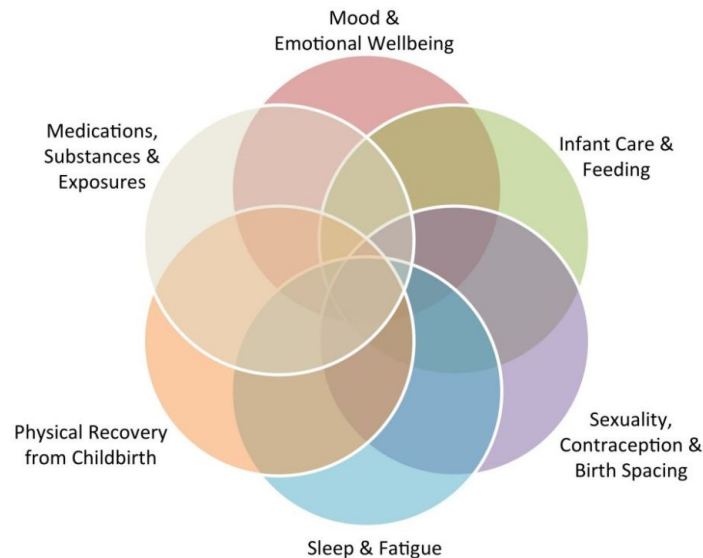
Figure 1. Basic Postpartum issues[20]

Typical postpartum symptoms can be severe if not properly dealt with, like: heavy bleeding, abdominal cramping, constipation, hemorrhoids, chills, night sweats, difficulty going to the bathroom, engorged breasts, back pain, headaches. And it goes on: pain in the perineum (the diamond-shaped sling of muscles in the pelvis), incision pain (if the woman has had a C-section), pain and difficulty walking (after an episiotomy or tear), depression, anxiety, and exhaustion.

1.3 Our Vision

Our vision is to become one of the most used healthcare mobile applications in Montreal at connecting parents to healthcare services, We are striving to become Canada's leader in providing mobile healthcare services focusing on enjoyable and reliable patient experience.

1.4 Business Goals

Every large system contains multiple objectives. Decomposing the whole software architecture problem into smaller pieces is an appropriate way to be able to describe, analyze and evaluate them in detail.

Aspect of concern	Goal	Questions
Appointment system	**Enhance customer experience and satisfaction with the efficiency of booking doctors appointments in Montreal Children's Hospital.**	1. What are the complaints against current hospital appointment booking system for doctors and nurses? 2. Is behavior of hospital staff an issue while booking or cancelling a doctor's appointment or a nurse's home visit ? 3. How flexible is the time window for cancelling a doctor's appointment or a nurse's home visit?

	Increase efficiency for nurses to check home visits appointments and cancellation	1. How current allocation system for nurses home visit works? 2. How much time hospital needs to notify a nurse in case of any change in a home visit appointment? 3. How the nurses notify the hospital in case they want to change a house visit appointment?
	Make it easier for parents to view available appointments and to book appointments that suit them better.	1. How difficult it is to book an appointment for parents who live far from Montreal children's hospital? 2. Is it easy for parents to explore available appointments? 3. Is the appointment system accessible for 24/7?
	More efficient appointments confirmations and notifications	1. What's the current way of confirming appointments with patients? 2. What's the current way of reminding or notifying appointments with patients?
Information and knowledge support	**Contribute to raise new mother's healthcare awareness.**	1. What percentage of current patients are aware of postpartum depression? 2. How many patients take advantage of the services provided by the hospital about postpartum depression? 3. How hospital notify them about the services now? 4. How single mothers cope up with the stress related to childbirth? 5. Will mothers be more motivated to take advantage of the service if they access more information about postpartum depression?
	Contribute to raise children healthcare awareness.	1. How does the Montreal Children's Hospital raise children healthcare awareness? 2. How to make hospital's current contributions in raising children healthcare awareness more available and accessible for each parent? 3. How does the hospital currently inform parents

		about event and workshops ?
		4. How to improve raising children healthcare awareness through our App?
	Easy access to suggestions from professionals for new parents	1. Is there any trusted resource available for new parents to help them to deal with their general doubts and concerns?
		2. How much parents can access professionals and get needed information?
		3. What kinds of mishaps new parents report to the hospital caused by general doubts?
Effective use of existing tools and resources	**Make efficient use of resources and tools such as EHR**	1. How often do nurses need to update the EHR after home visit?
		2. How often nurses need to view EHR while visiting patient's home?
Saving time and reducing cost	**Reduce time and cost for end users**	1. What is the average waiting time to access the appointment service from hospital?
		2. What is the cost associated with appointment booking and cancelling for hospital nurses?
Natural environment concern	**Reduce the use of paper (Go green)**	1. How much paper is being used currently for appointment services?
		2. How does hospital preserve the paper files?

Table 1. The business goals of the system

Part 2: Problem

2.1 Functional Requirements

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component [22]. Functional requirements drive the application architecture of a system.

Aspect of Requirement	Requirement item
General end user-related requirements	1. System shall allow the end user to register in the iCare App. 2. System shall allow the end user to login in the iCare App. 3. System shall allow the end user to view his\her profile. 4. System shall allow the end user to update his\her profile. 5. System shall allow the end user to retrieve username and password in case user forget any of them. 6. System shall allow the end user to view upcoming booked appointments.
Parent\ legal guardian\ mother end user-related requirements	1. System shall allow the parent end user to book an appointments with pediatricians. 2. System shall allow the parent end user to cancel any booked appointment within allowed cancellation period. 3. System shall allow the parent end user to view health-related FAQs about new born, new mom, and children. 4. System shall allow the parent end user to contact with a nurse for further information. 5. System shall allow the parent end user to post questions asking for after birth related concerning questions.

	6. System shall allow the parent end user to review and rate answered questions. 7. System shall notify parent end user when his question is answered. 8. System shall notify parent end user for upcoming appointments.
Professionals (nurse) related requirements	1. System shall allow the nurse user to view the home visit details such as time, home address, and the purpose of the visit. 2. System shall allow the nurse user to use a live map directions to the homes. 3. System shall allow the nurse user to update Electronic Health Record after every visit. 4. System shall notify nurse user for upcoming home visits. 5. System shall notify nurse user for cancellation of home visits. 6. System shall allow the nurse user to answer questions posted by parents.

Table 2. Functional requirements of the system

2.2 Non-functional requirements

In systems engineering a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors [16]. These requirements are as important as the functional requirements.

Software Quality Requirement	Decomposed Quality	Requirement Description
Security	**Authentication**	1. The app should support user authentication with a unique ID and a password only known by the user.
	Integrity	1. The data should be encrypted using WPA2 and 128-bit key encryption. 2. It should have adjustable security settings for trusted and untrusted locations. 3. Remote wiping should be enabled, as it allows a user to permanently delete all data stored on a mobile device remotely, such as when the device is stolen. 4. The app should support automatic log out, after a specific time of inactivity. 5. The app should warn the users if they are connected to a less secure Wi-Fi network, such as to a public wireless Internet. [17]
	Non-repudiability	1. The app should keep track of login/logout time of every user.
	Accountability	1. The app should have provisioning of logging interactions.

Privacy	Compliance	1. The user's consent for the handling of personal data must be free, specific and informed. If the user withdraws his consent it has to result in the deletion of his personal data.
		2. The data must be kept only the necessary time for the initial purpose and must be deleted once the purpose is achieved.
		3. The purposes for which the data is being processed should be precise and legitimate. Only data that are strictly essential for the functionality of the icare may be processed.
		4. The user needs to be informed before the disclosure of data to a third party and the app developer needs to enter into a binding legal agreement with the concerned third party.
	Awareness	1. The user must be educated what data are being collected and what can be inferred from such data.
	Confidentiality	1. It should be ensured that the icare app should not access SMS, camera, contacts, etc. without permission.
Maintainability	Testability	1. The software life cycle must follow a test first driven approach.
		2. The source code authored by the project team must be covered with unit testing. the system unit testing coverage should be more than 80% at all times.
		3. The application feature must be covered by automated testing, the system.
		4. automation testing coverage should be more than 70% at all times.

	Modifiability	1. All unit testing should pass correctly before commiting code to the source code repository from the integration machine. 2. No failed unit tests should be present on the source code repository at any time. 3. Follow the project coding, indentation and naming conventions standards. 4. Follow Code review and Code refactoring practices during SDLC process.
	Analyzability	1. Builds must contain no more and no less modules than are listed within the design document. 2. Eliminate the existence of duplicate codes by using Code Analysis tools.
	Modularity	1. The architecture should as much as possible utilize non monolithic approach and allow for loose coupling.
	Reusability	1. Follow and use different code metrics suites "MOOD and QMOOD" to monitor code maintainability. Metrics should be calculated at the end of every iteration or release.
Usability	**User Interface Aesthetics**	1. There should be at most 5-7 items showing on each page of the application so as to not confuse the user. 2. User interface should only show the relevant elements specific to each type of user (i.e parent, professional). 3. App should support both portrait and landscape orientation.

	Learnability	1. Application must have clear navigation, logical structure and easy-to-follow hierarchy and user must know what section they're currently in at any point.
		2. Provide a unobstructive and concise tutorial/tooltip overlay when the user lands on a page for the first time. The user should be able to easily bring the helpful overlay back as a refresher if they choose so as part of the help menu.
		3. In places where applicable, it should conform to existing "standard" design flows common in apps that are in similar vein or domain. Example are location of back button, menu item style, gesture used for modifying entries, etc.
	User Error Protection	1. Ask for confirmation when the parent wants to both book or cancel an appointment, making the system more error tolerant. However, application should not ask for confirmations indiscriminately or in too many places.
	Operability	1. Wherever it seems fit, textual options should be accompanied with appropriate graphics to aid in understanding what that option does, especially useful for those where their mother language is not in the app.
		2. A typical parent should be able to book an appointment in less than or equal to 5 minutes.
		3. App should have a "Remember me" option and keep the user from having to type their username/password for login every time.
		4. Provide the option for the parent of being notified/reminded by the application a certain number of days before their appointment.

	Accessibility	1. Option for different languages common to the region should be made available to the user.
		2. The use of colours should be such that it does not affect colour blind parents, or at least a colour blind setting should be made available to those who need it.
Portability	**Adaptability**	1. The application must provide a system of supporting variety of phones and tablets consisting of different resolutions and OS systems. 2. No more than 10% of the system implementation must be specific to an operating system
	Replaceability	1. The mean time to replace the database with another (of same type) should be less than or equal to 12 hours.
	Installability	1. Application must be available on the standard application market depending on the current OS.
Reliability	**Maturity**	1. No more than 1 per 1000 system actions (booking, cancellation, searching, etc) should result in a failure such that the application needs a restart. 2. The system defect rate should be less than or equal to 1 failure per 1500 hours of operation.
	Availability	1. System should have equal or more than 95% uptime.
	Recoverability	1. In a case of failure, system should need less than or equal to 1 min of restart time.
Performability	**Time Behaviour**	1. Search results (for available doctors, booking times, etc) should take less than or equal to 10 seconds on average.

	Resource Utilization	1. Software development shall utilize available related artifacts.
	Capacity	1. The app should be able to handle concurrent use from no less than 40% of the number of registered users.
Compatibility	Interoperability	1. The platform shall provide a method to connect the iCare platform to those backend legacy systems eliminating the need for any point to point integration and achieving interoperability between those systems and iCare services

Table 3. Non-functional requirements of the system

Part 3: Stakeholders and Concerns

3.1 Stakeholder Mind Map

Stakeholder is a person or an organization that has an interest in a company and can either affect or be affected by the project [7]. It is very important to identify and engage the stakeholders from the very beginning of the project to make sure that deliverables meet customer's need. Early identification of stakeholders and their concerns also help to reduce and uncover risks which paves the way for a successful and accepted project.

To identify the stakeholders of iCare at first we conducted brainstorming sessions and tried to answer following questions [8]:

- Who is directly involved with the project?

- Who is indirectly involved with the project?

- Who may be affected by the project?

- Who may be affected by the project's outcome?

- Who gains or loses from the project's success?

- Who wants to complete the project successfully and who doesn't?

- Who are the suppliers?

- Who is the user of the end result of the project?

- Who are the competitors?

- Who are the shareholders?

- Is any local community impacted by the project or its outcome?

- Who has the authority to influence the project or its outcome?

- Who has the authority to make the project succeed?

- Who can make your project fail?

We also studied about similar projects to analyse and group the stakeholders. Finally we came up with following stakeholders for iCare depicted by the mind map that is shown in figure 1 and further described in table 4.

Figure 2. Stakeholders Mind Map

Stakeholder Category	Stakeholder	Description
Competitor	Similar Applications	Other health care applications like iCare.
Investor	Sponsors	A person or an organisation, funding the project.
	Hospital	Owner of application.
Maintenance/ Customer Service	System Integrators	A person or a company integrating different project components
	Maintainers	A person or a company taking care of the technical issues of the project.
	Call Center	Department taking care of feedbacks from the users about the project.
Misuser	Hackers	Unauthorised person affecting the application security.
Regulatory authorities	RAMQ	The Régie de l'assurance maladie du Québec (RAMQ) is the government health insurance board in the province of Quebec, Canada.[2]
	Insurance Companies	Company or organisation providing financial protection or reimbursement to a policyholder.
Project Development Team	Project Manager	The person in overall charge of the planning and execution of a particular project.[3]
	Developers	Person concerned with facets of the software development process, including the research, design, programming, and testing of computer software.[4]
	Testers	Part of a software development team and perform functional and non-functional testing of software using manual and automated software testing techniques.[5]

|		
	Software Analyst	A software analyst is the person who studies the software application domain, prepares software requirements, and specification (Software Requirements Specification) documents.[6]
Users	**Parents/ Legal Guardians**	A father, a mother or a legal guardian to someone, according to Canadian Law.
	Professionals	Registered nurses of the hospital.
	Other Apps	Third party usable apps.

Table 4. Stakeholder Descriptions

3.2 Stakeholder-Concern mapping

To ensure a project's success it is very important to communicate with the stakeholders and to determine the level of communication we need to understand each stakeholder's interest and influence on the project and prioritise stakeholders according to that .For iCare we have prioritised the stakeholders based on Mendelow's matrix[1].

We have mapped iCare's stakeholders into four groups as described below:

1. Low interest, low influence – Stakeholders we need to keep informed.

2. High interest, low influence – Stakeholders we need to involve and consult with.

3. Low interest, high influence – Influential stakeholders we need to engage.

4. High interest, high influence – Partners we need to collaborate with.

Prioritisation of stakeholders is illustrated in figure 2. The arrow represents the direction of influence. The stakeholders at the tail of the arrow has an influence over the stakeholder at that arrowhead. For example Regulatory Authorities can influence Investors but Investors can not influence the Regulatory Authorities, so there is one arrow going from Regulatory Authorities to Investors.
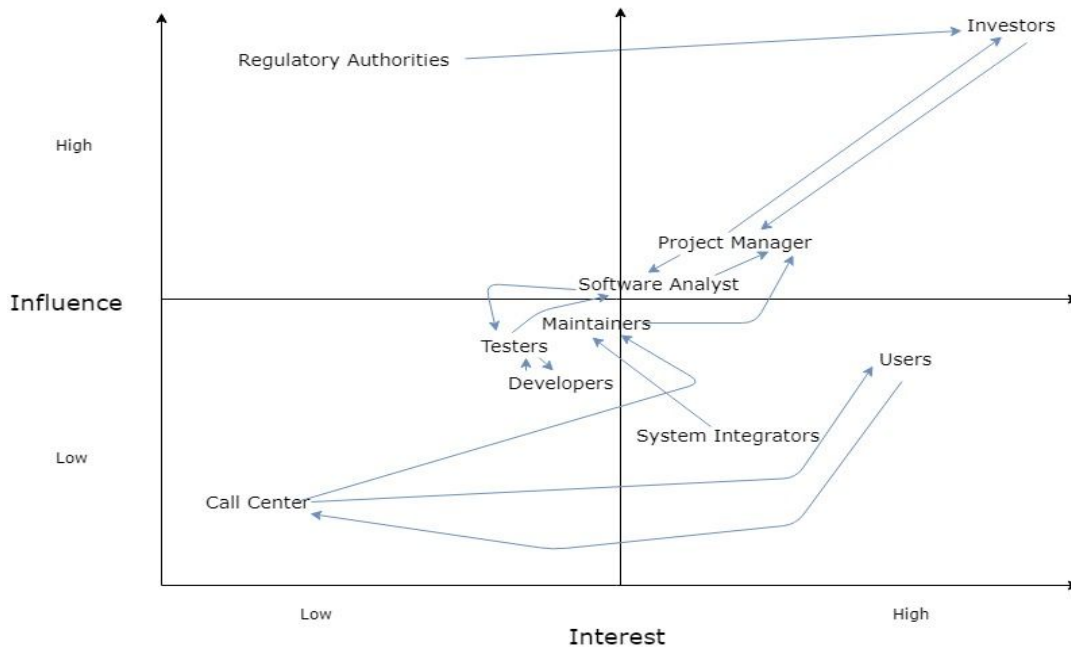
Figure 3. Prioritization of stakeholders based on interest and influence (adapted from Mendelow's matrix)

The following table describes the concerns and associated attributes:

Stakeholder	Concerns	Quality Attributes
Sponsors	1. High quality of final system to satisfy market users 2. Accessible, anywhere and anytime	● Maturity ● Availability
	1. Low cost of the system development 2. Proper use of resources	● Resource Utilization
Hospital	1. Simple user interface 2. Easy product use for end users 3. Accessible for people with color blindness 4. Accessible in multiple languages	● Appropriateness Recognizability ● Learnability ● Operability ● User Error Protection

		● User Interface Aesthetics ● Accessibility
	1. Compatible with other systems present in the hospital	● Interoperability
	1. Easy system maintenance 2. Easy update of the system	● Analyzability ● Modifiability ● Modularity ● Installability
	6. Protection from intrusion and data corruption	● Authenticity ● Integrity ● Confidentiality ● Compliance
	7. Accessible, anywhere and anytime	● Availability ● Time Behavior
	8. Low cost of the system development	● Resource Utilization
	9. High quality of final system to satisfy market users	● Maturity ● Fault Tolerance ● Recoverability ● Capacity
	10. On time delivery of final product	

Project Development team	1. Simple user interface	● Appropriateness Recognizability
	2. Effective error detection and easy recovery	● Learnability
	3. Accessible for people with disabilities, or special needs.	● Operability
	4. Accessible for people with color blindness	● User Error Protection
	5. Accessible in multiple languages	● Accessibility
	6. Effective resource allocation	
	7. Low cost of system development	● Resource Utilization
	8. Easy system maintenance	● Analyzability ● Modifiability ● Modularity ● Testability ● Reusability
	9. Accessible, anywhere and anytime	● Availability ● Time Behavior ● Capacity
	7. Easy exchange of shared information	● Interoperability
	8. Protection from unauthorized access and data corruption	● Authenticity ● Integrity ● Confidentiality ● Compliance
	9. High quality of final system to satisfy market users	● Maturity ● Fault Tolerance ● Adaptability

| | | • Installability |
		• Recoverability
	10. On time delivery of final product	
	11. Realistic and achievable requirements	
	12. Proper Documentation of the system	
Regulatory Authorities	1. Easy verification of legality and authenticity of doctors and professionals	• Accountability
	2. Easy verification of all the intense actions and services offered	• Authenticity
		• Non repudiability
	3. Authenticity and security of the information being public by the iCARE	• Compliance
Maintenance / Customer service	1. Easy system integration	• Analyzability
	2. Easy system maintenance	• Modularity
		• Installability
Parents / Legal Guardians	1. Simple user interface	• Appropriateness Recognizability
	2. Easy to learn user interface	• Accessibility
	3. Effective error detection and easy recovery	• Learnability
	4. Accessible for people with color blindness	• User Error Protection
	5. Accessible in multiple language	• User Interface Aesthetics
		• Operability
	6. Accuracy and reliability of information	• Accountability
	7. Protection of personal information	• Authenticity
		• Confidentiality
		• Compliance
		• Awareness
		• Integrity

	8. Compatible with devices	● Adaptability ● Installability
	9. Fast response time 10. Accessible, anywhere and anytime	● Time Behavior ● Availability ● Capacity
	11. Useful notifications 12. Usage cost should be adequate	
Professionals	1. Simple user interface 2. Includes essential and useful information 3. Easy to learn user interface 4. Effective error detection and easy recovery	● Appropriateness Recognizability ● Learnability ● Operability ● User Error Protection
	5. Accessible from anywhere	● Availability
	6. Protect private information	● Compliance ● Confidentiality
	7. Compatible with device	● Interoperability ● Adaptability
Other Applications	1. Easy and fast access	● Time behavior
	2. Protection from unauthorized access and data corruption	● Non repudiability ● Integrity
	3. Easy exchange of shared information	● Interoperability

Table 5. Stakeholder Concern Mapping

Part 4: Viewpoints and Views

We are using the 4+1 view model designed by Philippe Kruchten[9] for describing the architecture of iCare mobile application system. The views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers.
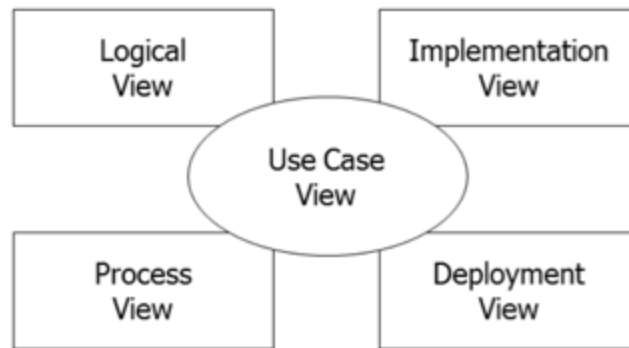
Figure 4. The 4+1 View Model os Software Architecture: Version 2 [10]

4.1 Use Case View

This view presents architecturally significant use cases that help highlight main architectural decisions and choices (see section 5 of this document) . The following use case view shows the most important use cases for iCare.

1. **Use Case: Account services** is for user to be able to register in the iCare app, login later on, view and update her profile, and in case she forgets her usernames or passwords she can retrieve them.

2. **Use Case: View upcoming appointments** allows user to check her upcoming appointments with pediatricians or nurses visits. This is for both the child appointments and the mom workshops\sessions.

3. **Use Case: Create appointment and Cancel appointment** are for the parent to book or cancel appointments with pediatricians or nurses.

4. **Use Case: View FAQs** shows the health-related FAQs about newborns, new moms, and children. It has an extend use case **Ask a nurse** which allows parents to send message to a nurse in case of the FAQs are not useful enough for a specific case.

5. **Use Case: View upcoming parents workshops** is for parents to keep them informed about any event or workshop offered by the hospital for parents or new moms.

6. **Use Case: View home visit details** shows the home visit details to the nurse such as time, home address, and the purpose of the visit.

7. **Use Case: Answer a Question** allows nurses to answer doubts and questions posted by parents.

8. **Use Case: Update EHR** enables the nurse to retrieve the patient(child or mom) EHR and update it according to the information obtained by home visit check up.

9. **Use Case: Notify for Appointments** is a reminder for the parent's appointments and nurses home visits or the cancellation of a visit.

Figure 5. UML Use Case Diagram for iCare Application

4.2 Logical View

The Logical view focuses on functionality inspired by abstractions in the problem domain. It is responsible for the conceptual organization of layers and high-level functionality of components in each layer. The iCare platform contains the following layers:

1. iCare Microservices: This is the component where iCare Business logic will reside, each use case will use one or more than one microservice. These microservices will communication with the backend end systems

2. The Customer Identity Access Management (CIAM): This is the component where the end users information will be stored, it will also be responsible to provide authentication and authorization services.

3. The API Gateway: This component will be the interface between the mobile application and the backend microservices, it will be responsible about the lifecycle management of the microservices different APIs.

4. The Enterprise Service Bus: This component will be responsible to provide a solution to one to one integration design anti pattern and provide mediation and transformation services to communicate with the backend legacy systems.

5. The Message Broker: The message broker (Queue) will be responsible to provide and support all necessary Asynchronous communication between Microservices and backend systems.

6. The iCare mobile Application: This is the component that will be responsible for interacting with the end user collecting his input and commands and showing results to him.
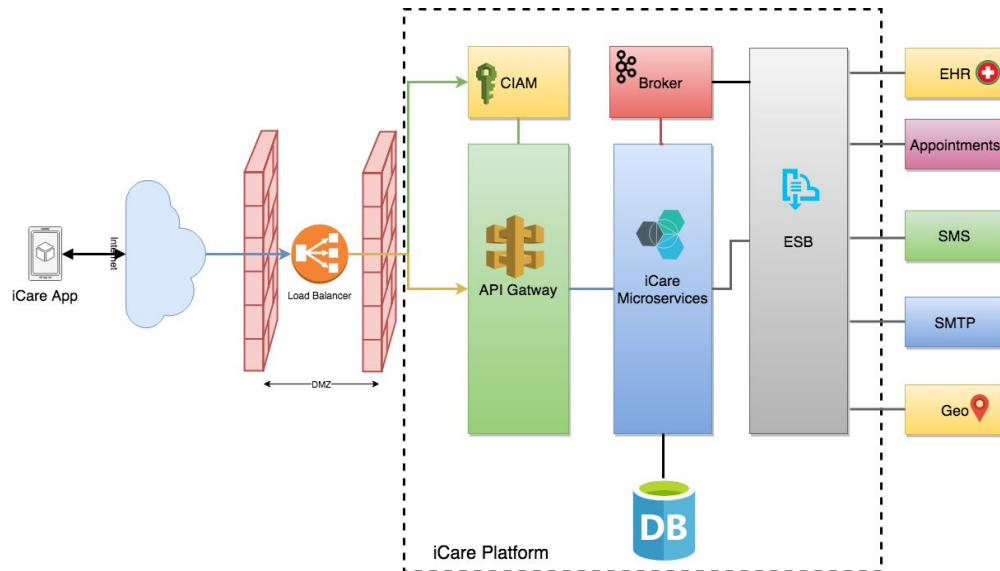
Figure 6. Boxes and lines diagrams showing components used in iCare

4.3 Process View

This view focuses on (concurrent) tasks performed by software at run-time. It is responsible for processes, threads, and control. We are presenting a sample process of how Model-View-Controller will be used in implementing the Appointment Handler Microservice.

User will send a request to book an appointment through iCare mobile application. iCare will send that request to controller, controller will direct the task to Appointment Handler Microservice. After that Appointment Handler will instruct Hospital Appointment System to register the appointment. If every condition associated with booking an appointment is satisfied correctly then Appointment System will book the appointment and send a message to Appointment Handler indicating the appointment is booked. Appointment Handler will send a confirmation message to controller and controller will direct that message to View. Finally View will format and send the confirmation message to display the end result for user to iCare indicating that the appointment is booked.
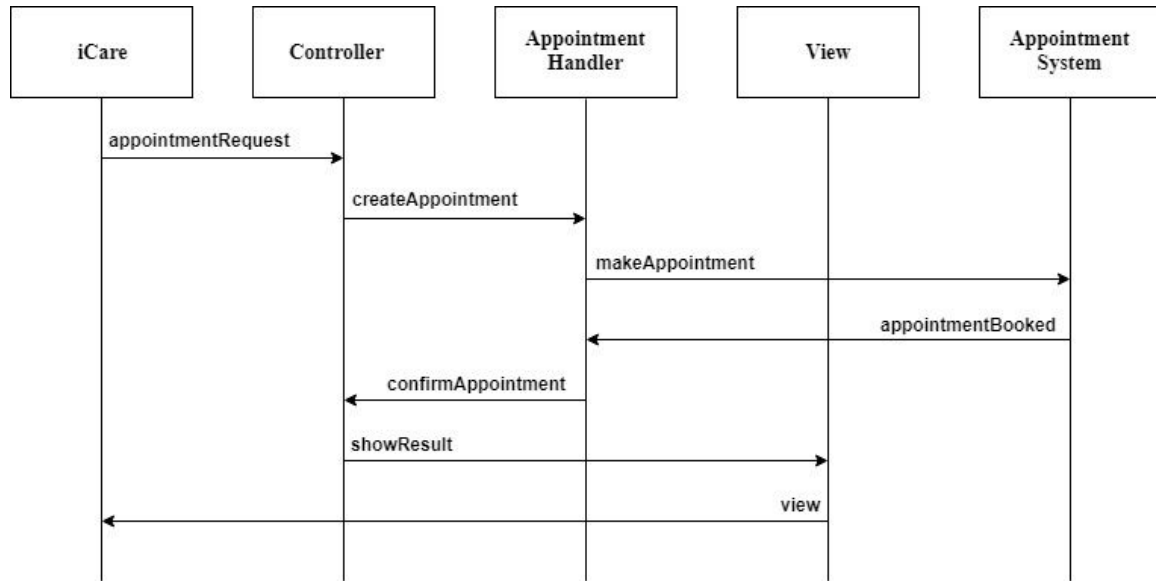
Figure 7. UML sequence diagram showing MVC process in a sample microservice

4.4 Implementation View

This view focuses on the actual source code, data files, and executables. Figure 4.3 below shows the UML class and package diagram for implementing iCare Appointment Microservice. The implementation will follow the MVC design Pattern as explained in previous sections. The model package will include the Session Facade and Data Access Object classes. The Data Access object design pattern is used to design the classes responsible to store, retrieve and delete information from persistence systems like databases and backend legacy system, the session facade is an interface between the DAO classes and different types of clients to support loose coupling. The Facade client in this implementation would be the Microservice, which is according to this architecture is the controller, The Appointment Microservice will receive requests from the mobile application and will use Model tier classes to persist or retrieve the information. Finally the Appointment Microservice will use one view implemations from the view package to format the final results. The mobile application at the end will use the formatted results to show result to the end user.
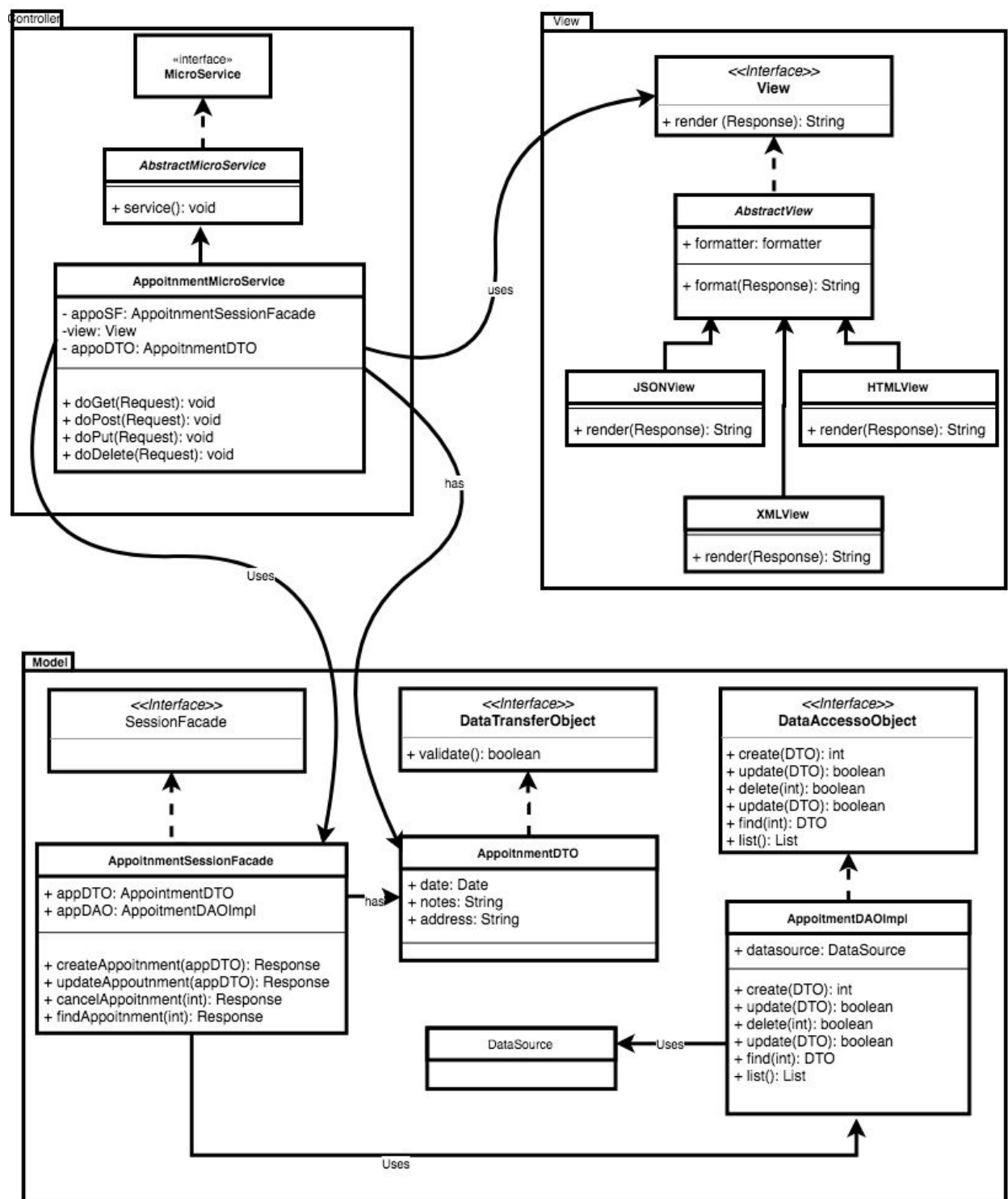
Figure 8. UML Class diagram showing MVC implementation in a Appointment MicroService

4.5 Deployment View

This view describes the iCare environment into which the system will be deployed, including capturing the dependencies the system has on its run-time environment. According to the components listed in the logical view section the following figure shows how different components will be deployed physically on different servers. For the iCare environment we will have two different vLans, the DMZ and and the Platform vLan where different components are installed, the DMZ is used to reduce security risk by not exposing the platform vLan directly to the internet. Linux CentOS servers will be used to host the different components, different components have different size of memory. All components are made high available in active-active mode.
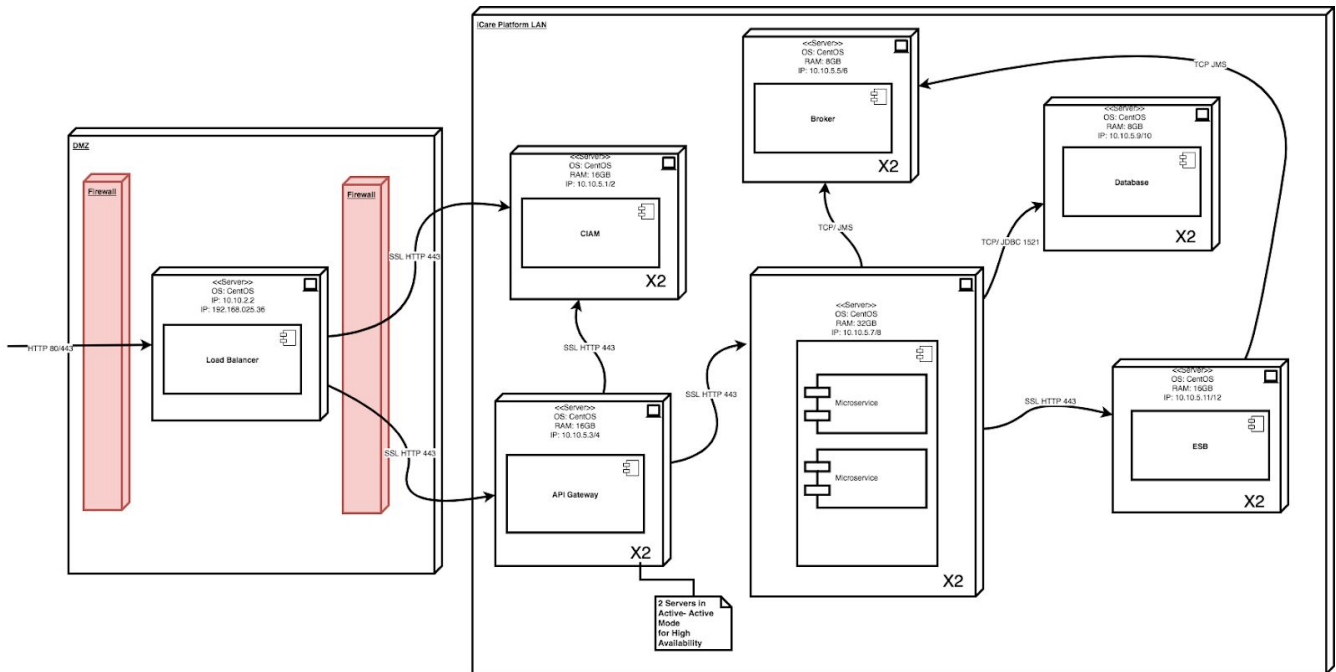
Figure 9. UML deployment showing different components installation

Part 5: Architectural Decisions [21]

5.1 Using Microservices Architecture

iCare platform which is the backend end system that will support the iCare "New Mom" mobile application will follow a Microservices architecture, which is an architectural style that structures an application as a collection of loosely coupled services [11]. It comes to break the classical monolithic web based applications into smaller granular, autonomous and loosely coupled business components, it enables the continuous delivery and continuous deployment of large, complex applications which makes it possible to release new features and bug fixes more frequently and more efficiently.

5.2 Using Service Oriented Architecture and Enterprise Service Bus

The iCare Platform is designed to come compatible and interoperable with existing serive providers "hospitals" infrastructure and legacy system (ex. SMTP, SMS, EHR ..etc), in other words the platform will be compatible with the existing hospital backend systems and will not require any changes on those systems. Doctors and nurses will keep using the same patients' EHR system, receptionist will keep using the hospital's appointment system as usual, and Emails will be sent using the same mail exchange service. The platform will use an Enterprise Service Bus "ESB" to connect the iCare platform to those backend legacy systems eliminating the need for any point to point integration and achieving interoperability between those systems and iCare services. We will pick an open source ESB to be used in our project, good candidates are WSO2 Enterprise Integrator from WSO2 or Mule from MuleSoft, both support various types of protocols including the HL7 [12] [13] protocol.

5.3 Using Customer Identity Access Management System

iCare mobile application users digital identity attributes will be saved on the platform's customer identity and access management system CIAM, the CIAM system will connect to the hospital's legacy user stores system (example Active Directory) to allow professionals "Nurses in iCare for Moms app" to login to the application for example. The CIAM system will also be responsible for generating the access tokens that will be used by the mobile application to access the platform services and APIs, the CIAM system will support OAuth and OpenID Connect protocols which are the de facto

standard protocols for securing APIs and microservices. There are so many available open source CIAM that we can use like WSO2 IDM, Gluu and ForgeRock.

5.4 Using API Manager (API Gateway)

Access to the platform microservices will go through an API Gateway which will be used to publish and manage the lifecycle of iCare microservice APIs, enforce usage policies, throttling, collecting and analyzing usage statistics, reporting on performance and controlling access by integrating with the platform's CIAM component. A very popular open source API manager to use is WSO2 API Manager.

5.5 Using Microservices, Java and Spring Boot

iCare Microservices layer is the core component of our platform, a microservice is the basic building block of the core business logic layer. Functional requirements will be translated in source code in their final format, business logic then will be broken into granular and autonomous small pieces of codes and will be deployed independently as microservices, for example, "Appointment" will be one of the Microservices built in the platform, it will handle all requests coming from the mobile application related to create, view, update or cancel a doctor's appointment, a use case like taking an appointment with a doctor will compose using multiple microservices including the "Appointment" "Email Sender" "Health Card checker" and "SMS Notifier", this is just a simple example on how a use case can be broken into smaller independent functionalities. There are so many open source frameworks and packages for building microservices, Spring Boot, NodeJS and MS4J are just few of available SDKs and frameworks. We will use the Spring Boot framework to build our Microservices because it is Java based which makes it benefit from the portable nature of Java based frameworks, it is has a quick and easy learning curve and rich content on the internet [14]. Model-View-Controller "MVC" design pattern will be used when authoring different microservices, in a Microservices the view will be the JSON messages exchanged with the mobile app, the controller will be the used framework's proxy and the model will contain the java classes that persist and retrieve information from the database or the backend legacy systems.

5.6 Using Message Broker (Kafka)

Microservices will need to exchange messages with each other and also exchange messages with legacy systems to fulfill functional requirements, to support different types of message exchange patterns including asynchronous communication, the platform will use a message broker, not all communication between platform components should use simultaneous synchronous communication, it is commonly known that is always

expensive using synchronous communication especially when it comes to non functional requirements like reliability, availability and fault tolerance. We will use one of the so many used open source message brokers like Kafka, RabbitMQ and ActiveMQ. Kafka offers a high guarantee that the service will be available and non-blocking under any circumstances. In addition, messages can easily be replicated for higher data availability. Kafka performance is just great and resource usage modest.[15]

5.7 Using Relational Database

Relational Database will be used by the microservices to persist data, we are going use a relational database instead of NoSQL database. We will just use the existing database engine available part of the service provider infrastructure and we want to save the complexity associated with managing a new type of components. Also it should be easier for future platform extensions using relational databases creating dashboards. To avoid information leakage between microservices, each microservice will have its own database schema with no access to other schemas on the database. All communication between microservices should happen via the front channels of their APIs.

5.8 Using Native SDK to develop the mobile application

The iCare Mobile application is the human interface component of the platform, it will be created as a native iOS and Android application, all communication between the application and the backend iCare platform will be thought APIs and using JSON messages. Before accessing the any backend APIs the application will authenticate the end user through the CIAM system using the OAuth protocol, communication between the application and the APIs has to go over the https protocol and passing the correct access tokens.

5.9 Using Load Balancer to support high availability

All platform components will be used in clustering mode to guarantee high availability and to avoid single point of failures in the design, at least two instances of each component will running on Active-Active or Active-Passive modes. The Load Balance will take the responsibility to distribute the traffic between each the same instances of each component. The LB will be placed in a DMZ zone to achieve high network security measures and allow extra time to detect and address breaches before they would further penetrate into the internal networks of the platform.

Glossary

Term	Description	Source
MOM (Minutes of Meeting)	Minutes, also known as minutes of meeting (abbreviation MoM), protocols or, informally, notes, are the instant written record of a meeting or hearing. They typically describe the events of the meeting and may include a list of attendees, a statement of the issues considered by the participants, and related responses or decisions for the issues.	Wikipedia contributors. Minutes. Wikipedia, The Free Encyclopedia. May 11, 2018, 08:01 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Minutes&oldid=840653946. Accessed May 15, 2018.
Health App/mHealth App	Programs (applications or apps) that deliver health-related services using smartphones and tablets. Some apps offer advice and tracking functionality for healthy living. Some are designed to transmit information between doctors and patients (e.g. glucose readings for diabetes management). Some are meant for doctors to keep accurate, accessible records.	http://mhealthregulatorycoalition.org/resources/mhealth-glossary/
Electronic Health Record (EHR)	A real-time electronic record of patient health information and medical history, generated by one or by multiple visits in any care delivery setting. EHRs provide access to evidence-based decision support tools to aid clinicians in decision making. They can also support the collection of data for uses other than clinical care, such as billing, quality management, outcome reporting, and public health disease surveillance and reporting.	http://mhealthregulatorycoalition.org/resources/mhealth-glossary/

Demilitarized zone (DMZ)	In computer security, a DMZ or demilitarized zone (sometimes referred to as a perimeter network) is a physical or logical subnetwork that contains and exposes an organization's external-facing services to an untrusted network, usually a larger network such as the Internet.	Wikipedia contributors. (2018, April 24). DMZ (computing). In Wikipedia, The Free Encyclopedia. Retrieved 18:15, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=DMZ_(computing)&oldid=838062513
Enterprise service bus (ESB)	An enterprise service bus (ESB) implements a communication system between mutually interacting software applications in a service-oriented architecture (SOA).	Wikipedia contributors. (2018, June 1). Enterprise service bus. In Wikipedia, The Free Encyclopedia. Retrieved 18:17, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Enterprise_service_bus&oldid=843954527
Customer Identity Access Management (CIAM)	Customer (or Consumer) Identity Access Management is a subset of the larger concept of Identity Access Management (IAM) and is focused specifically on managing the identity, access and security of software applications.	Wikipedia contributors. (2018, May 22). Customer Identity Access Management. In Wikipedia, The Free Encyclopedia. Retrieved 18:19, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Customer_Identity_Access_Management&oldid=842494768
Microservices	Microservices is a software development technique—a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services.	Wikipedia contributors. (2018, May 27). Microservices. In Wikipedia, The Free Encyclopedia. Retrieved 18:20, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Microservices&oldid=843216915

API management	It is the process of creating and publishing web APIs, enforcing their usage policies, controlling access, nurturing the subscriber community, collecting and analyzing usage statistics, and reporting on performance	Wikipedia contributors. (2018, May 26). API management. In Wikipedia, The Free Encyclopedia. Retrieved 18:21, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=API_management&oldid=843029546
Message broker	A message broker is an intermediary computer program module that translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver.	Wikipedia contributors. (2018, May 17). Message broker. In Wikipedia, The Free Encyclopedia. Retrieved 18:23, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Message_broker&oldid=841721420
Short message service (SMS)	SMS (short message service) is a text messaging service component of most telephone, internet, and mobile-device systems. It uses standardized communication protocols to enable mobile devices to exchange short text messages.	Wikipedia contributors. (2018, June 2). SMS. In Wikipedia, The Free Encyclopedia. Retrieved 18:25, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=SMS&oldid=844103917
Simple Mail Transfer Protocol (SMTP)	Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (email) transmission. First defined by RFC 821 in 1982, it was last updated in 2008 with Extended SMTP additions by RFC 5321, which is the protocol in widespread use today.	Wikipedia contributors. (2018, May 23). Simple Mail Transfer Protocol. In Wikipedia, The Free Encyclopedia. Retrieved 18:27, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Simple_Mail_Transfer_Protocol&oldid=842662591

Geolocation software	In computing, geolocation software is software that is capable of deducing the geolocation of a device connected to the Internet.	Wikipedia contributors. (2018, May 28). Geolocation software. In Wikipedia, The Free Encyclopedia. Retrieved 18:28, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Geolocation_software&oldid=843393093
Service oriented architecture (SOA)	It is a style of software design where services are provided to the other components by application components, through a communication protocol over a network.	Wikipedia contributors. (2018, May 12). Service-oriented architecture. In Wikipedia, The Free Encyclopedia. Retrieved 18:30, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Service-oriented_architecture&oldid=840782929
OAuth	This is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords	Wikipedia contributors. (2018, April 1). OAuth. In Wikipedia, The Free Encyclopedia. Retrieved 19:55, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=OAuth&oldid=833649422
Monolithic application	In software engineering, a monolithic application describes a single-tiered software application in which the user interface and data access code are combined into a single program from a single platform.	Wikipedia contributors. (2018, February 4). Monolithic application. In Wikipedia, The Free Encyclopedia. Retrieved 19:15, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Monolithic_application&oldid=823996769

Interoperability	It is the characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, at present or in the future, in either implementation or access, without any restrictions.	Wikipedia contributors. (2018, May 22). Interoperability. In Wikipedia, The Free Encyclopedia. Retrieved 19:32, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Interoperability&oldid=842391066
OpenID Connect	OpenID Connect (OIDC) is an authentication layer on top of OAuth 2.0, an authorization framework. The standard is controlled by the OpenID Foundation.	Wikipedia contributors. (2018, April 26). OpenID Connect. In Wikipedia, The Free Encyclopedia. Retrieved 19:56, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=OpenID_Connect&oldid=838355012
Health Level-7 (HL7)	It is a set of international standards for transfer of clinical and administrative data between software applications used by various healthcare providers. These standards focus on the application layer, which is "layer 7" in the OSI model.	Wikipedia contributors. (2018, May 16). Health Level 7. In Wikipedia, The Free Encyclopedia. Retrieved 20:03, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Health_Level_7&oldid=841576673
Throttling	In software, a throttling process, or a throttling controller as it is sometimes called, is a process responsible for regulating the rate at which application processing is conducted, either statically or dynamically.	Wikipedia contributors. (2018, March 29). Throttling process (computing). In Wikipedia, The Free Encyclopedia. Retrieved 20:21, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Throttling_process_(computing)&oldid=833143627

Model–view–controller	Model–view–controller is commonly used for developing software that divides an application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.	Wikipedia contributors. (2018, May 21). Model–view–controller. In Wikipedia, The Free Encyclopedia. Retrieved 20:53, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=Model%E2%80%93view%E2%80%93controller&oldid=842259450
JSON	It is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value).	Wikipedia contributors. (2018, May 30). JSON. In Wikipedia, The Free Encyclopedia. Retrieved 20:55, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=JSON&oldid=843693815
NoSQL	A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.	Wikipedia contributors. (2018, May 27). NoSQL. In Wikipedia, The Free Encyclopedia. Retrieved 21:30, June 2, 2018, from https://en.wikipedia.org/w/index.php?title=NoSQL&oldid=843169983
Stakeholder	A person or an organization that has an interest in a company and can either affect or be affected by the business.	https://www.investopedia.com/terms/s/stakeholder.asp
Negative stakeholder	A negative stakeholder is, intentionally or incidentally, a threat to the software project.	[Alexander, Beus-Dukic, 2009] Discovering Requirements: How to Specify Products and Services. By I. Alexander, L. Beus-Dukic. John Wiley and Sons. 2009.

Mind Map	A diagram used to visually organize information.	https://en.wikipedia.org/wiki/Mind_map
Mendelow's Matrix	A matrix to help analyse stakeholders	http://kfknowledgebank.kaplan.co.uk/KFKB/Wiki%20Pages/Mendelow%27s%20matrix.aspx
Stakeholder mapping	Mapping that helps to deal with stakeholders' conflicting demands. It identifies stakeholder expectations and power and helps in establishing political priorities.	http://kfknowledgebank.kaplan.co.uk/KFKB/Wiki%20Pages/Mendelow%27s%20matrix.aspx
Security	Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization	ISO/IEC, 2011 https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en
Privacy	The ability of an individual, group, or an organization to seclude themselves, or information about themselves, and thereby express themselves selectively.	Wikipedia https://en.wikipedia.org/wiki/Privacy
Maintainability	The degree of effectiveness and efficiency with which a software product can be modified by the intended maintainers	ISO/IEC, 2011 https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en
Usability	The degree to which a software product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction, in a specified context of use.	ISO/IEC, 2011 https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en

Portability	The degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.	ISO/IEC, 2011 https://www.iso.org/obp/ui/#iso :std:iso-iec:25010:ed-1:v1:en
Reliability	The ability of a system or component to perform its required functions under stated conditions for a specified period of time.	ISO/IEC, 2011 https://www.iso.org/obp/ui/#iso :std:iso-iec:25010:ed-1:v1:en
Postpartum depression	This is a type of mood disorder associated with childbirth, which can affect both sexes	https://en.wikipedia.org/wiki/ Postpartum_depression

Table 6. Glossary Descriptions

References

[1] Kaplancouk. (2018). Kaplancouk. Retrieved 10 June, 2018, from http://kfknowledgebank.kaplan.co.uk/KFKB/Wiki Pages/Mendelow's matrix.aspx

[2] Wikipedia contributors. (2018, May 18). Régie de l'assurance maladie du Québec. In Wikipedia, The Free Encyclopedia. Retrieved 00:06, June 10, 2018, from https://en.wikipedia.org/w/index.php?title=R%C3%A9gie_de_l%27assurance_maladie_du_Qu%C3%A9bec&oldid=841889388

[3] Wikipedia contributors. (2018, May 10). Project manager. In Wikipedia, The Free Encyclopedia. Retrieved 00:08, June 10, 2018, from https://en.wikipedia.org/w/index.php?title=Project_manager&oldid=840497379

[4] Wikipedia contributors. (2018, June 5). Software developer. In Wikipedia, The Free Encyclopedia. Retrieved 19:41, June 10, 2018, from https://en.wikipedia.org/w/index.php?title=Software_developer&oldid=844599977

[5] Techopedia.com. (2018). What is a Software Tester? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/29845/software-tester [Accessed 10 Jun. 2018].

[6] Wikipedia contributors. (2017, December 20). Software analyst. In Wikipedia, The Free Encyclopedia. Retrieved 00:11, June 10, 2018, from https://en.wikipedia.org/w/index.php?title=Software_analyst&oldid=816347569

[7] Investopedia. (2018). Stakeholder. [online] Available at: https://www.investopedia.com/terms/s/stakeholder.asp [Accessed 9 Jun. 2018].

[8] Usmani, F. (n.d.). How to Identify Project Stakeholders? Retrieved June 9, 2018, from https://pmstudycircle.com/2012/06/identify-stakeholders-project-management/

[9] Wikipedia contributors. (2017, March 25). 4+1 architectural view model. In Wikipedia, The Free Encyclopedia. Retrieved 21:13, June 8, 2018, from https://en.wikipedia.org/w/index.php?title=4%2B1_architectural_view_model&oldid=772138375

[10] The 4+1 View Model of Architecture. By P. B. Kruchten. IEEE Software. Volume 12. Issue 6. 1995. Pages 42-50.

[11] Chris Richardson. What are microservices? Retrieved 19:15, June 2, 2018, from http://microservices.io/

[12] WSO2 Enterprise Integrator Documentation. HL7 Transport. Retrieved 19:15, June 2, 2018, from https://docs.wso2.com/display/EI600/HL7+Transport

[13] Mule Documentation. HL7 Connector. Retrieved 19:15, June 2, 2018, from https://www.mulesoft.com/exchange/org.mule.modules/mule-module-hl7/

[14] Spring Boot website. Spring Boot Guides. Retrieved 19:15, June 2, 2018, from https://spring.io/guides

[15] Dzone, Integration Zone. Exploring Message Brokers: RabbitMQ, Kafka, ActiveMQ, and Kestre Retrieved 19:15, June 2, 2018, from https://dzone.com/articles/exploring-message-brokers

[16] Wikipedia contributors. (2018, March 6). Non-functional requirement. In Wikipedia, The Free Encyclopedia. Retrieved 20:53, June 10, 2018, from https://en.wikipedia.org/w/index.php?title=Non-functional_requirement&oldid=829143575

[17] Shifali Arora, Jennifer Yttri, and Wendy Nilsen: "Privacy and Security in Mobile Health (mHealth) Research" 2014 , Retrieved 21:15, May 29, 2018, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4432854/

[18] Allison Yarrow: What no one tells new moms about what childbirth can do to their bodies, updated May 4, 2018, 9:51 am EDT, Retrieved 18:15 June 10, 2018, from https://www.vox.com/science-and-health/2017/6/26/15872734/postnatal-care-america

[19] Eugene R. Declercq, Carol Sakala, Maureen P. Corry and Sandra Applebaum: New Mothers Speak Out, National Survey Results Highlight Women's Postpartum Experiences, Retrieved 18:30 June 10, 2018, from http://www.nationalpartnership.org/research-library/maternal-health/listening-to-mothers-ii-new-mothers-speak-out-2008.pdf

[20] Voxcom. (2017). Vox. Retrieved 11 June, 2018, from https://www.vox.com/science-and-health/2017/6/26/15872734/postnatal-care-america

[21] Ieeeorg. (2018). Ieeeorg. Retrieved 11 June, 2018, from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=

[22] Wikipedia contributors. (2018, January 21). Functional requirement. In Wikipedia, The Free Encyclopedia. Retrieved 00:52, June 11, 2018, from https://en.wikipedia.org/w/index.php?title=Functional_requirement&oldid=821649846

Tool Support and Team Communication Channel:

Team used different online tools and a common communication channel to during the project.

1. Project Work space and communication channel :
   https://drive.google.com/drive/folders/1od_XV7LjZ1aipC0IJN-fBXq0swwUbaJN

2. Mind mapping tool : https://coggle.it/

3. Diagramming tool: draw.io

Contribution Table

Our team utilized a somewhat adapted agile methodology, with each user stories assigned to a number of people. We had meeting after every class and on (most) Fridays where all members reviewed, modified and approved each item.

Group/Team/Unit Member	Contributions
Maria Ahmed	- Part 1: Vision and Business Goals - Part 3: Stakeholder Mind Map and Concern Mapping - Part 4: Process View - Authoring Final Document
Wasim Alayoubi	- Part 1: Vision - Part 2: Non Functional Requirements (Usability, Maintainability) - Part 4: Improved / Guided in creation of all the views - Part 5: Architectural Decisions - Authoring Final Document
Marwah Alsadun	- Part 1: Vision and Business Goals - Part 2: Functional Requirements - Part 4: Use Case View - Authoring Final Document
Shahram Aryanpour	- Part 1: Motivation - Part 2: Non Functional Requirements - Part 4: Implementation view - Part 5: Architectural decisions

	● Authoring Final Document
Ramit Basra	● Part 1: Business Goals ● Part 2: Functional Requirements ● Part 3: Stakeholder Mind Map and Concern Mapping ● Part 4: Logical View ● Authoring Final Document
Bikramjit Singh	● Part 1: Motivation ● Part 2: Non Functional Requirements ● Part 4: Deployment View ● Part 5: Architectural Decisions ● Authoring Final Document

Table 7. Team Contributions