|  |
| --- |
| CS5200 |
| DBMS Project - MedNet |
| Spring 2017 |
|  |

|  |
| --- |
|  |

Contents

[Abstract 3](#_Toc481111283)

[Introduction 3](#_Toc481111284)

[Requirements 4](#_Toc481111285)

[System Actors 4](#_Toc481111286)

[Healthcare Accreditation Authority 4](#_Toc481111287)

[Doctor 4](#_Toc481111288)

[Hospital 4](#_Toc481111289)

[Civilian 4](#_Toc481111290)

[Government 4](#_Toc481111291)

[Volunteer Organization 4](#_Toc481111292)

[Registered User 4](#_Toc481111293)

[Un-registered User 5](#_Toc481111294)

[Use Cases 5](#_Toc481111295)

[Civilian Registration 5](#_Toc481111296)

[Doctor Registration 5](#_Toc481111297)

[Hospital Registration 5](#_Toc481111298)

[Volunteer Organization Registration 5](#_Toc481111299)

[Request Medical Help 6](#_Toc481111300)

[Book Medical Appointments 6](#_Toc481111301)

[Track Medical Appointments 6](#_Toc481111302)

[Search Profiles 6](#_Toc481111303)

[Upload Certificates 6](#_Toc481111304)

[Become a Donor 6](#_Toc481111305)

[Personalize settings 6](#_Toc481111306)

[Alternate Flows 6](#_Toc481111307)

[Civilian Registration 6](#_Toc481111308)

[Doctor Registration 7](#_Toc481111309)

[Design 8](#_Toc481111310)

[UML Diagram 8](#_Toc481111311)

[Rationales (Issues and Tradeoffs - Design Analysis) 9](#_Toc481111312)

[Issue 1: Custom data types with fixed values 9](#_Toc481111313)

[Issue 2: Different type of MedNet Users 9](#_Toc481111314)

[Issue 3: Medical Profile and Registered User Details 10](#_Toc481111315)

[Issue 4: Medical Requests and Authorization 10](#_Toc481111316)

[Implementation 11](#_Toc481111317)

[Programming Language 11](#_Toc481111318)

[Framework 11](#_Toc481111319)

[Softwares 11](#_Toc481111320)

[Discussion 11](#_Toc481111321)

[Use Case 1: Request Medical Help 11](#_Toc481111322)

[Use Case 2: Medical Profile – Share, Edit, View 11](#_Toc481111323)

[Use Case 3: Organ Donation 11](#_Toc481111324)

[Use Case 4: Fund Donation 12](#_Toc481111325)

[Conclusion 12](#_Toc481111326)

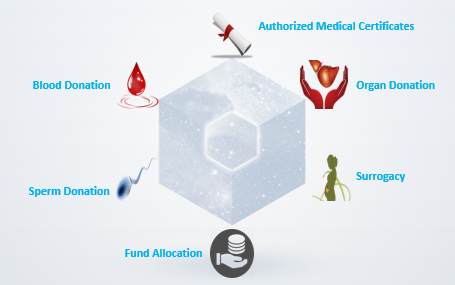
[Future Enhancements 12](#_Toc481111327)

[References 13](#_Toc481111328)

# Abstract

An electronic health record (EHR), or electronic medical record (EMR), refers to the systematized collection of patient and population electronically-stored health information in a digital format. These records can be shared across different health care settings. Despite its importance and confidentiality, there is no secured global platform and the information is usually stored in ad-hoc fashion, with little or no access. This project is an effort to create a social media for medical help – MedNet, for citizens to create medical profiles, form connections, and centralize their medical history. Features such as request for medical guidance, becoming a donor, endorsing medical services etc. are included in MedNet to make it one place for all medical needs. It further helps government and healthcare authorities to monitor health trends and fast access to people who need medical help.

# Introduction



MedNet – A social media for medical help aims to provide a centralized location to connect with people based on your medical interests and needs. You can build authorized medical profile, and upload your medical documents to access them from anywhere.

This native IOS app intends to create a medical network enabling you to build your medical identity online and connect with people of similar medical interests and needs. It provides a mobile enabled web platform with features such as seeking medical help, tracking medical appointments, volunteering for donation, endorsing medical services by hospital and doctors and many more. It organizes data in a sophisticated RDBMS schema providing fast, secure and stable retrieval of information.

One of the motivation for this project was to allow government to monitor the medical condition, analyze disease trends and hospitals to have fast access to their patient’s recent medical history. Furthermore, like any social media platform, it can extend to search users by keywords and view their medical profile, request them for medical help like asking for donation from a donor or funds from a volunteer organization, hospitals can reach out to doctors and vice versa. With the growth of this app and more and more people becoming part of it, it serves a goldmine of health data with endless possibilities for analysis and many more goals can be achieved along with few mentioned above.

# Requirements

## System Actors

### Healthcare Accreditation Authority

The sole authority that manages the registration approval of all the users of this system. One important responsibility is to verify the authenticity of the user in specified time.

### Doctor

An individual who uses the system as a platform for endorsing medical services including but not limited to medical supervision, disease specific guidance and medicine subscription. The doctor may or may not be part of any registered hospital. Doctor must undergo a specialized process for approval into the system. A doctor must abide by the medical guidelines as specified by the accreditation authority.

### Hospital

A community approved by the accreditation authority for main purpose of providing medical services to civilians on request. These services include patient treatment with specialized medical and nursing staff and medical equipment, medicine subscription, blood donations among others. Hospital also houses various registered Doctors.

### Civilian

An individual who uses the system to seek medical help. Civilian is the central entity of this system. A civilian can also volunteer for donations including blood donation, organ donation, and fund donation. In order to become a donor, a civilian must get approval from accreditation authority.

### Government

The governing body of the nation, state or community. The government has authority to use non-personalized data of all other users, for health care analysis. Government empowers the healthcare accreditation authority.

### Volunteer Organization

A community, which offers large, scale medical help such as fund raising for expensive health treatments, organizing health care camps including blood donation camps, monthly health checkup camps. Volunteer organization must follow the guidelines specified by the accreditation authority. It is reachable by other users from within the system.

### Registered User

A single individual or community that registers to take part in seeking or providing medical services through the system. Accreditation Authority must approve a registered user.

### Unregistered User

A single individual or community that empowers other users on the system. These users have access to non-personalized data of the system for doing analysis or detecting hoax information provided by registered users on the system.

## Use Cases

### Civilian Registration

1. A civilian creates a profile

2. The civilian is then asked to upload documents and waits for registration approval

3. The system queues the request and informs the Healthcare Accreditor

4. The Accreditor verifies the documents uploaded by the civilian

5. The accreditor then approves or rejects the registration request based on the information provided by the civilian

6. The system notifies the user that the registration request was accepted or rejected

7. The civilian receives the notification

### Doctor Registration

1. A doctor creates a profile along with medical services he/she want to offer.

2. The doctor uploads the certificates for specified categories and request an approval from accreditation authority

3. Health Accreditation Authority verifies the information and documents within specified time

4. If Health Accreditation Authority find everything authentic, it approves the request

5. The system then notifies the Doctor about the approval or rejection

6. The doctor receives the notification

7. The doctor’s profile becomes public to be found on the system by other users

### Hospital Registration

1. A hospital creates a profile as a community and endorses the medical services offered by them

2. The hospital then waits for approval from accreditation authority

3. The system queues the request and informs the Healthcare Accreditor

4. Healthcare accreditor completes verification of the hospital in specified time

5. If the accreditor finds everything authentic, it approves the request

6. The system then notifies the Hospital about the approval or rejection

7. The Hospital receives the notification

8. Hospital’s profile will become public to be found on the system by other users

### Volunteer Organization Registration

1. Volunteer Organization creates a profile as a community and requests for registration from Healthcare Accreditation Authority

2. The system queues the request and informs the Healthcare Accreditor

3. The healthcare accreditor verifies the organization for authenticity

4. The accreditor then accepts or rejects the registration request based on the data provided by the volunteer organization

5. The system notifies the organization about the acceptance or rejection of its registration request

6. The Volunteer Organization receives the notification

7. The Volunteer Organization’s profile becomes public to be found on the system by other users

### Request Medical Help

1. A registered user selects the category of the medical service

2. Registered particular service like an organ name and other specific filters to service

3. The registered user specify the quantity

4. Registered user timeline

5. Register user clicks for request

6. A list of options gets displayed to the user in the order of most to least relevant

7. Registered user will go to one of the profile and places the request

8. The requested registered user is notified with the request and requester details

### Book Medical Appointments

1. Registered user enters the zip code

2. Registered user select the type of medical service

3. List of hospitals gets displayed to the user

4. User selects one of the hospital

5. User books the appointment on the selected hospital’s page

### Track Medical Appointments

1. A registered user clicks on the available appointments

2. A list of upcoming appointments will be shown

### Search Profiles

1. A registered user types the keyword, name or partial name in the search box and click search

2. A list of Registered Users, Hospitals, Organizations will be displayed

### Upload Certificates

1. Registered User selects the category of document

2. Registered User uploads all the files and click add

### Become a Donor

1. Registered user selects donate

2. Registered select the type of donation

3. Registered user books an appointment with the accreditation authority to get approval as donor

### Personalize settings

1. Registered user selects change account settings

2. Registered user updates the settings and save

## Alternate Flows

### Civilian Registration

**Precondition:** Civilian is not registered already. Approval process is same as defined above.

1. Civilian goes to an accredited medical health center and submit documents

2. Health center initiate the registration for the civilian

3. Civilian receives a notification to authenticate the request

4. Health center uploads the civilian details and approves the profile

### Doctor Registration

**Precondition:** Doctor is not registered already. Approval process is same as defined above.

1. The registered goes to an accredited medical health center and submit documents

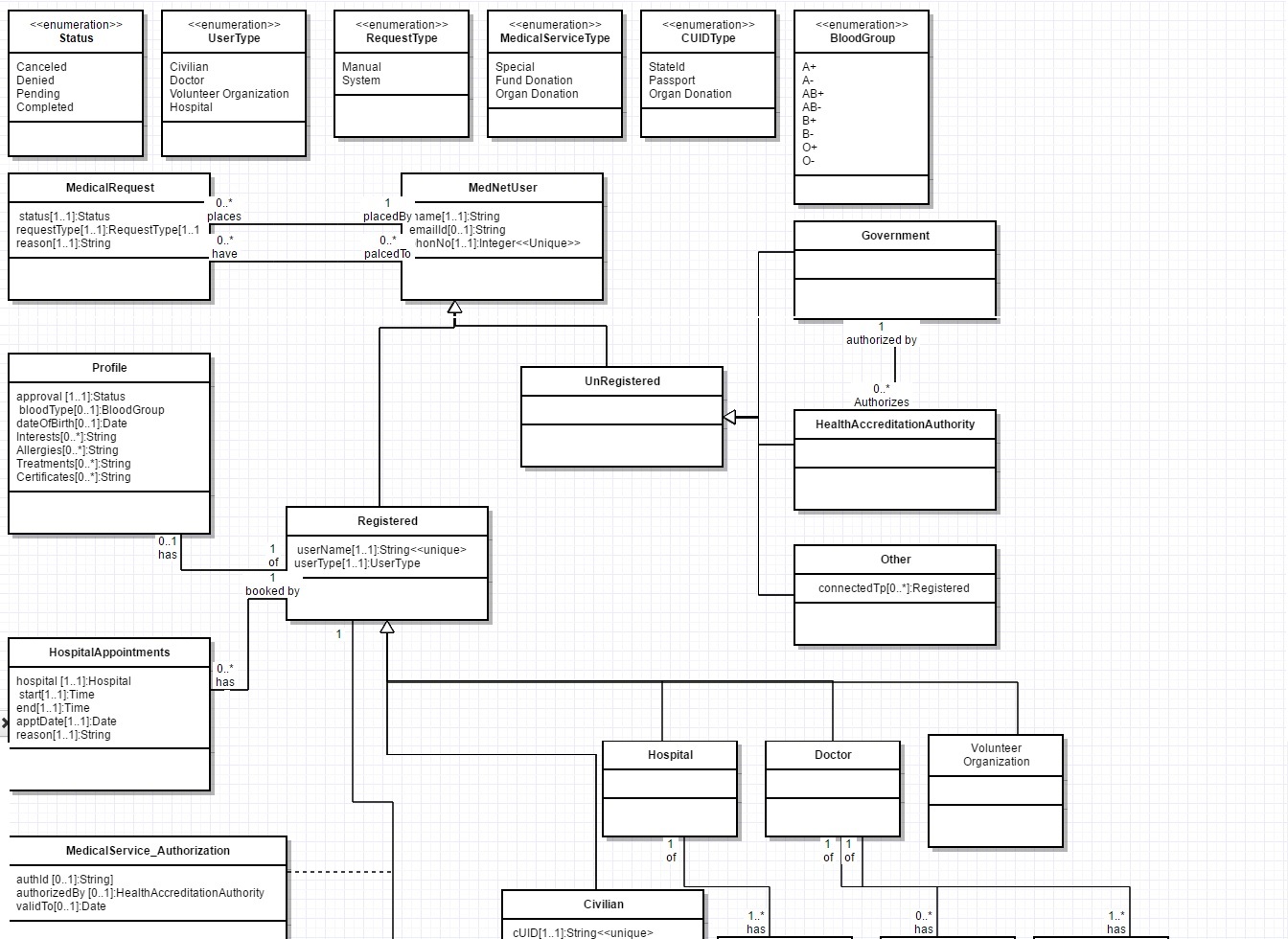
2. Health center initiate the registration for the civilian

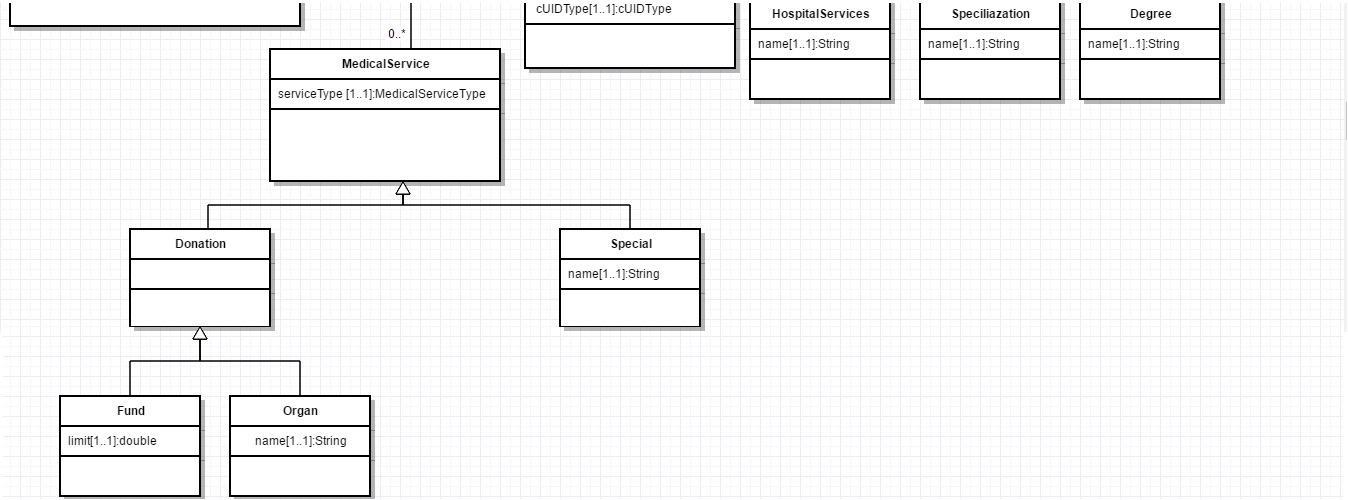
3. Civilian receives a notification to authenticate the request

4. Health center uploads the civilian details and approves the profile

# Design

## UML Diagram





## Rationales (Issues and Tradeoffs - Design Analysis)

### Issue 1: Custom data types with fixed values

For example, blood groups, request status, types of Ids etc.

**Design Choice:** Make each of those types as enumeration and implement them using JOINED Strategy at the database level.

Advantages: Highly organized and customizable.

Disadvantages: Separate table for each enumeration

**Decision:** We chose to make following enumerations as each of them has fixed possible values.

1. Status

2. User Type

3. Request Type

4. Medical Service Type

5. CUID Type

6. Blood Group

We implement them using JOINED strategy as we can add new values for each easily without changing the schema. Also, it is resembles the UML design better and easy to understand.

Issue 2: Different type of MedNet Users

Many type of users registers on MedNet like civilians, doctors, hospitals, volunteer organizations. Also, there are un-registered users who are part of MedNet like government, Health Accreditation Authority and other users who can connect themselves to other registered users.

**Design Choice 1:** Make separate class for each user

**Advantages:** Easy segregation of data associated with each type. Easy to implement in database

**Disadvantages:** Duplication of common attributes

**Design Choice 2:** Make a super class “MedNet User” which holds common user information like name, email etc. Make two subclasses of it called “Registered” and “Unregistered”. Then extend Registered for Doctor, Civilian, Hospital and Volunteer Organization as they are registered user and extend Unregistered for government, Health Accreditation Authority and Other.

**Advantages:** Easy segregation and normalized data (no duplication of data). Object Oriented Design

**Disadvantages:** Somewhat complex to implement. Entities have to be connected to each other through Foreign Keys

**Decision:** We chose to go with above design i.e. Inheritance and we used TABLE\_PER\_CLASS strategy to implement in the database. It is because it organizes the data very well without redundancy and resembles the object-oriented design. Thus, it is easy to map this schema to Object Relational Mapping.

### Issue 3: Medical Profile and Registered User Details

We need to store and fetch different information for users like Medical Profile, Medical Appointments, Medical requests, Medical Services.

**Design Choice:** Since, we need this for all registered users and we chose Design choice 2 in Issue 3, we do not have to connect these with all types of users. It is best to choose each of them as separate classes having one to many relationship with the Registered User resembling that they are part of registered user.

**Advantages:** Updates of mentioned information will be cascaded to the registered user. Also, if registered user is deleted all the related information gets automatically deleted.

**Disadvantages:** Fetching the information requires joining many tables.

**Decision:** We chose to go with the above design as it is easy to understand and serves easy maintenance of the registered user information. For example, one user can have many appointments and medical requests.

**Special Design choice for doctors and hospitals:** For hospitals and doctors, we chose to go with 1..\* association for medical services and degrees respectively. It is to put a constraint that hospitals and doctors must have at least one medical service and degree respectively.

### Issue 4: Medical Requests and Authorization

Different users provide different medical requests like hospitals provides special services, civilians can be donors, volunteer organization. Furthermore, these services must be authorized by Health Accreditation Authority.

**Design Choice:**

**Part1:** We chose to have the super class Medical Service which further extended by Donation and Special. Next, Donation is extended by Fund and Organ.

**Part2:** Next, we created an association class “MedicalService\_Authorization” for the association between Health Accreditation Authority and Medical Service.

**Advantages:** Part1 helps saves us from connecting each medical service with Health Accreditation Authority. Also, it follows inheritance which is easy to translate to Object Relational Mapping

Part2, helps us to store additional information about medical service authorization like validity date, authorized by etc.

**Disadvantages:** Complex implementation involving multiple tables. Fetching information requires joining multiple tables

**Decision:** We chose to go with the above design as it strictly resembles the object oriented design. It avoids redundancy and allows us to relate medical services with minimum connections. We used Table\_Per\_Class strategy for implementing inheritance.

# Implementation

## Programming Language

* Swift – For IOS App

## Framework

* SQLite.swift – Provides methods and classes for SQLite database operations for IOS Apps

## Softwares

* XCode – IDE to build IOS Apps
* SQLite – Database for our IOS App
* DB Browser For SQLite – To view data and execute demo script

# Discussion

We will talk about some of the use cases, which are unique in this application. Other use cases are common and self-explanatory.

## Use Case 1: Request Medical Help

**Scenario:** There is a need of blood for a patient Cyruff. Traditional way is to buy from blood bank or hospital. You can also find a relative or friend with same blood group. However, your reach is limited, especially if Cyruff has a rare blood group.

**MedNet Solution:** This app helps you find people easily of blood group you are looking for from the list of registered donors. You can place a medical request within the app to that person or can contact him/her if they have made their contact public. Furthermore, they are authorized and have a validity period. Only donors having validity remaining will be shown in the list.

## Use Case 2: Medical Profile – Share, Edit, View

**Scenario:** Cathy met an accident. She is new to the place and very few people know her. The local residents carried her to the hospital and one of them was able to find her identity and search her on the app.

**MedNet Solution:** Let’s assume that Cathy has made her medical profile visible. This will help hospital to get her recent medical history and treat accordingly keeping in mind her allergies, treatments and diseases which is known from her profile in the app. Furthermore, without doing a blood test, her blood group is known and a blood donor can be found fast. Important to note that all these information is legitimate as they are made public only after verification by Health Accreditation Authority.

## Use Case 3: Organ Donation

**Scenario:** John wants to donate his kidney. However, none of his acquaintances need a kidney transplant. Michael, who is a complete stranger to John, is in urgent need of a kidney. But Michael doesn’t know anybody who’s willing to donate a kidney for him. How can Michael reach out to John for help?

**MedNet Solution:** John and Michael are registered users of MedNet. Since John wants to become a donor, he will choose to become a donor. On that screen, he will choose the type of donation as Organ and he will specify the organ that he wants to donate, which in this case would be his kidney. Once he is cleared as a certified donor by the accreditation authority, he will be publicly visible as a kidney donor. Now, Michael searches on MedNet for a kidney donor. John’s name will be listed as a certified kidney donor. So, Michael places a medical request to John for his kidney. John receives the request and confirms that he would be willing to donate his kidney. Michael successfully undergoes a kidney transplant thanks to John’s willingness to donate his kidney.

## Use Case 4: Fund Donation

**Scenario:** Deboire Sachtem is the head of LoveCare organization. LoveCare organization is a volunteer organization that helps people in need of funds to cover expensive medical treatment. However, Deboire has come across so many scams that people employ to swindle LoveCare that she’s now wary and suspicious of people requesting for help. She spends a lot of money to hire professionals to do background checks on the people who have placed requests to LoveCare. These additional costs could spell out a reduction in LoveCare’s ability to financially help people in need.

Michael needs money urgently to cover the cost for his kidney transplant. He has placed a request to LoveCare but it’s taking a long time to complete the background checks that LoveCare needs to approve his request. He is extremely worried that he might not have enough funds when the time comes. Also, he has a niggling doubt that LoveCare might be a fraudulent organization.

**MedNet Solution:** LoveCare and Michael are registered users of MedNet. Since LoveCare wants to be a donor, Deboire will choose to become a donor. On that screen, she will choose the type of donation as Funds and she will specify the amount that the organization is willing to spend on a patient. Once LoveCare is cleared as a certified donor by the accreditation authority, it will be publicly visible as a fund donor. Now, Michael searches on MedNet for a donor who could help him pay for the kidney transplant. LoveCare’s name will be listed as a certified fund donor. So, Michael places a medical request to LoveCare for the necessary funds. LoveCare receives the request and confirms that they would be willing to pay for his transplant. LoveCare knows that Michael is genuinely in need of money as his profile indicates that he is scheduled for a transplant. Michael knows that LoveCare is a genuine organization that helps people in need as it is certified by the accreditation authority. Michael successfully undergoes a kidney transplant thanks to LoveCare’s generous donation.

# Conclusion

We were able to achieve all the use cases as targeted. As mentioned in the introduction, there are endless possibilities of future enhancements. Some of them are listed here:

## Future Enhancements

* Get detailed reports of diseases based on age groups, demographic location and lifestyle habits.
* Early detection of diseases based on the year summary medical reports of state or city.
* Monitoring health rate and disease trends among certain group like state or country.
* Authorized and Secure fund transfers for donations.

# References

* Celis, S. (2017, Mar 30). *SQLite.swift*. Retrieved from https://github.com/stephencelis/SQLite.swift
* Chavely, P. and Dixit, P. (2015, Jun 7). *Aadhar MedNet*. Retrieved from https://www.youtube.com/watch?v=2sDQ-P5SEnc
* Wagner, C. (2016, Mar 3). *SQLite Tutorial: Getting Started*. Retrieved from https://www.raywenderlich.com/123579/sqlite-tutorial-swift