

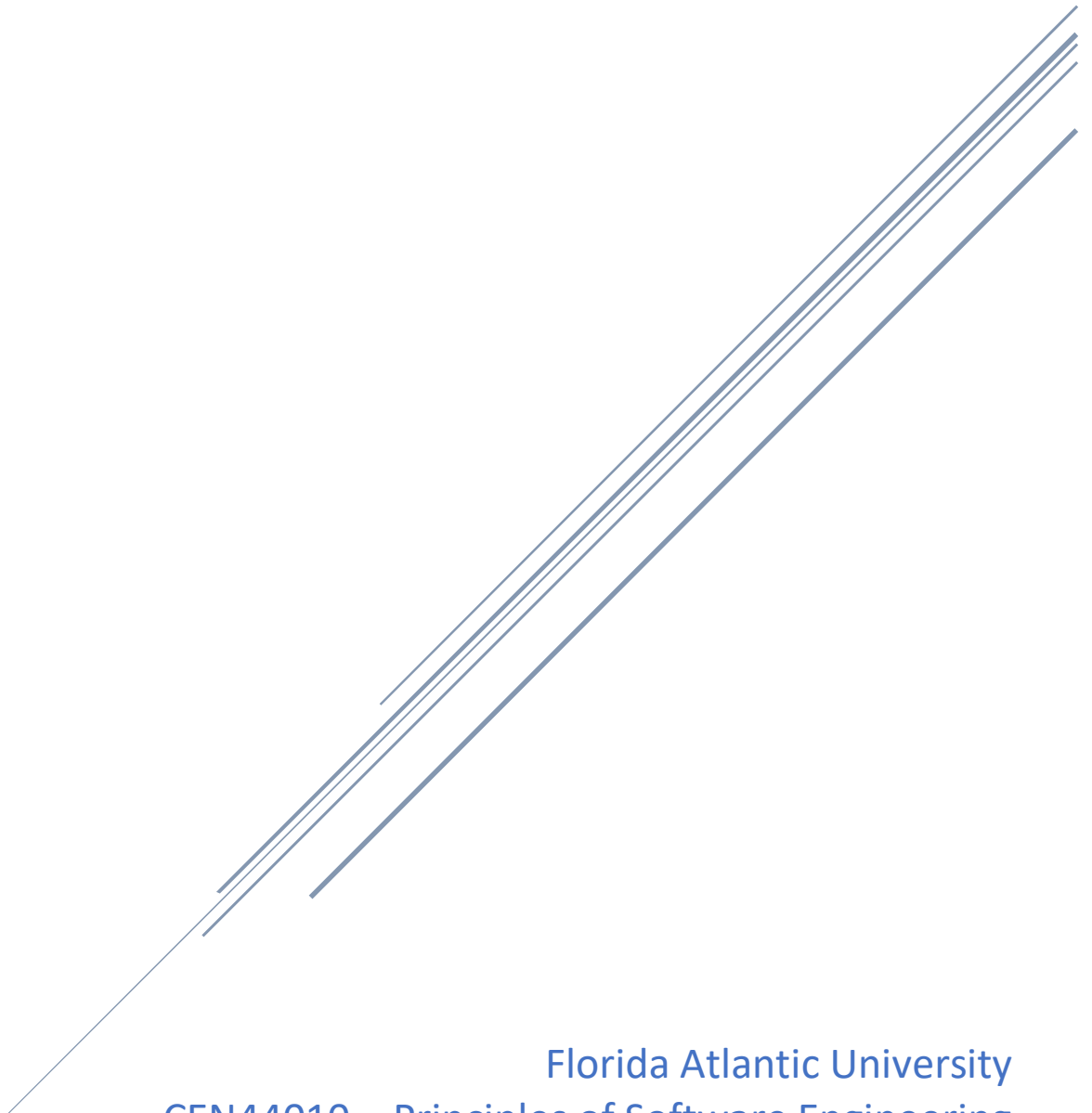
CYBERHEALTH – MILESTONE 3

CEN4010_GROUP22

Nelly Delgado Planche(ndelgadoplan2020@fau.edu)

Nha Tran (ntran2020@fau.edu)

Huy Nguyen (huynguyen2020@fau.edu)



Florida Atlantic University
CEN44010 – Principles of Software Engineering
Date: 10/25/2021

Table of Contents

I. Executive Summary.....3

II. Competitive analysis3

III. Data definition5

IV. Overview, scenarios, and use cases.....5

V. High-level functional requirements7

VI. List of non-functional requirements.....9

VII. High-level system architecture.....10

VIII. High-Level UML diagrams.....11

IX. Identify actual key risks for the project at this time:12

X. Team and checklist.....12

XI. History table13

XII. References13

TESTING VERTICAL PROTOTYPE

1. Username and Password

For Admin, **admin1** – **abc**

For Doctor, **doctor1** – **abc**

For Patient, **patient1** – **abc**

2. Since anyone can sign up to be a patient, we decide to implement Sign up feature for patient. For Admin and Doctor, these accounts will be provided since not everyone can be admin or doctor.

Depending on your role, it will direct you to a different page.

For database testing purpose, we will use 2 simple information which are username and password. We will add other information such as first name, last name, date of birth, etc. later.

Validation of the form is implemented in this prototype. We also make sure password of users secured by hashing the password before inserting into the database.

Please find our prototype at **lamp.cse.fau.edu/~cen4010_fa21_g22/test1**

I. Executive Summary

As of now, the COVID-19 pandemic has impacted every area of our lives. Therefore, we cannot have the same routine as we were used to when we must distance ourselves and limit the exposure to others as much as possible. The number of COVID-19 patients is increasing dramatically to the point that some hospitals are struggling to keep up with the rising demand for beds and shortage of staff. Illness severity can range from mild to critical. All deaths usually occurred among patients with critical illness while patients with mild symptoms can be treated at home with the doctor's instructions. Speaking of it, we would like to introduce our application to ensure everyone with COVID-19 symptoms can get safe home care and prevent the spread of coronavirus through close contact by going to a hospital while they can receive certified doctor's instructions via a website application called CyberHealth.

CyberHealth is an online health monitoring platform where patients with COVID-19 symptoms can upload their medical history as well as vital signs in order to receive treatment from certified doctors. Licensed doctors will determine if you should be tested or not and provide timely care based on the illness severity. Our virtual care application commits to deliver a seamless user experience, responsive web design, and a user-friendly interface to all users regardless of patients or doctors.

II. Competitive analysis

	CyberHealth	Teladoc	MDLIVE	MyMDnow	Private Offices
Online visit	5	5	5	5	0
Fast & simple process	5	4	4	3	2
User interface	5	5	3	5	0
Fast access to covid's doctors	5	4	4	2	2
Fast communication	5	4	4	2	2
Process tracking	5	4	4	1	1
Mean	5	4.3	4	3	1.2

Private Offices (1.2):

- Private offices are the local and offline doctor's offices. Patients usually follow one doctor for a long time, so the doctor knows the patients as well as their medical records. However, patients sometimes need to go through a long process from phone calls, making an appointment, be in the office, filing paperwork before they can see the doctor. In addition, since patients have to physically be in the office, the covid exposure is much higher. Furthermore, there are not many specific Covid doctor offices available locally.

Teladoc (4.3):

- Teladoc is a platform to connect doctors and patients by phone. They provide a good user interface and process. Since they are an online service, they also eliminate the covid exposure. On the other hand, Teladoc focuses on general healthcare, so there is a longer and more complicated process for patients to access Covid care.

MDLIVE (4):

- MDLIVE is a virtual doctor's platform, which allows patients to see doctors by phone call, on a computer, or in an app. It provides many healthcare services. However, since they provide many services, not focus on Covid care, the process is longer and more complicated to access Covid care. Furthermore, their user interface is not eye-catching.

Mymdnw (3):

- Mymdnw is the information and booking website for MDnow urgent care. The website is clean and informative. It provides good information for patients and allows them to make appointments. However, patients still need to physically be in the office, this can increase the covid exposure. In addition, MDnow is also a general healthcare provider, so it is a long and complicated process to access Covid care. Furthermore, it's a physical office, so patients will need to go through a long process from driving, filling paperwork, waiting for their turn.

CyberHeath(5):

- CyberHealth is a platform connecting patients and doctors, focusing on Covid care services. Even though this platform doesn't provide a wide range of healthcare services, it laser focuses on Covid care to create a great experience for users. CyberHealth has a great clear, clean, and eye-catching user interface. It also provides a fast, simple, and effective process for both Covid patients and doctors. In addition, it allows patients to quickly access Covid treatment and medication. Furthermore, CyberHealth lets patients and doctors effectively communicate and get updated on the treatment process.

Planned advantages:

- At CyberHealth, since we are laser focus on Covid care, it gives us any advantages compare to our competitor. First, we are online service, so we can eliminate the Covid exposure to others and help patient access to doctors anywhere and anytime, even if they are sick. Second, we are focus on Covid care only, so we can make the process simpler, faster, and more effective for both patients and doctors. It also allows us to reduce the complexity of the system and focus on user experience and interface. Third, we provide many great and unique features such as process tracking, communication box, and saved

treatments and prescription depend on condition. Those allow users to communicate effectively, get clear updated on their process, save time, and have a much better experience.

III. Data definition

Name	Meaning	Usage
Home	User interface	The introductory page of a website
Service	User interface	List what we offer for users
Sign in	User service	Existing patients can upload their health information and see their treatment while doctors can see their patients and provide treatment
Register	User service	Allow new patients to create an account
User	Actors	Refer to both patients and doctors
Member	Actors	An user who has an account with the system
Non-member	Actors	An user who does not have an account with the system
Admin	Actors	An user can see doctor's information as well as patient's information
Profile	Use case scenarios	Store user's information
Status	Use case scenarios	Display the illness severity of patients
Comments	Use case scenarios	Leave a note

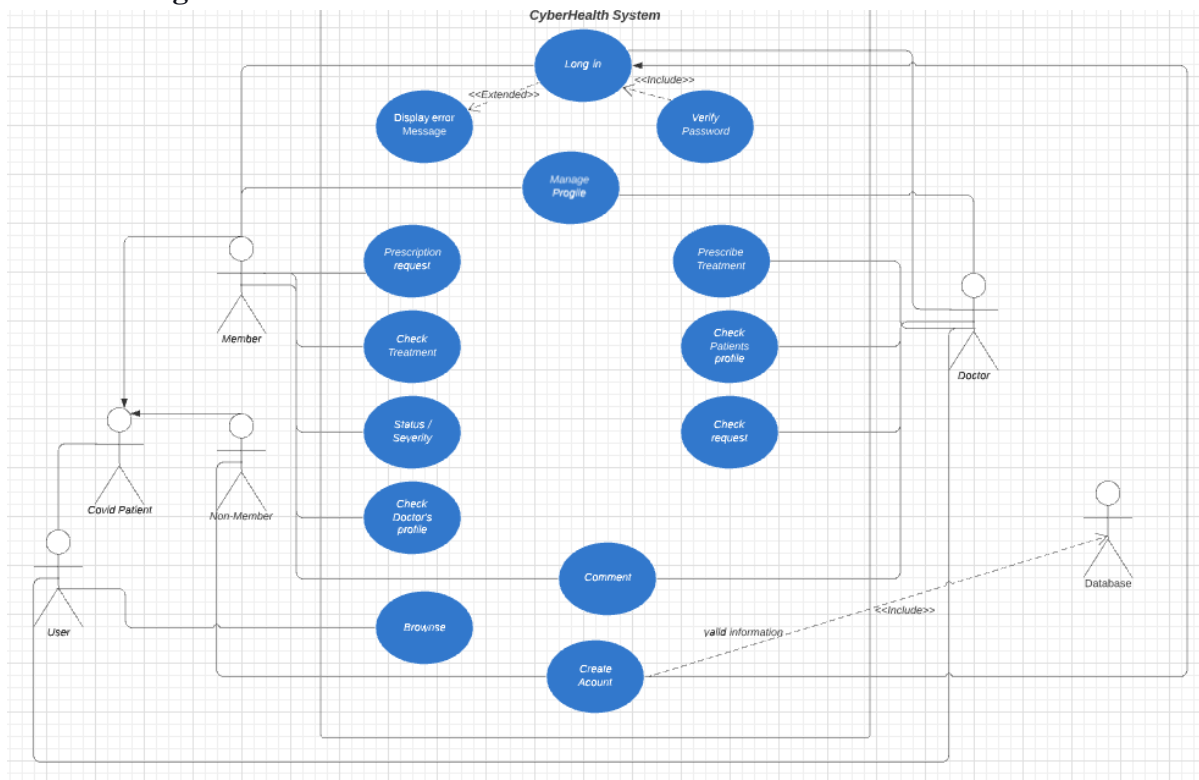
IV. Overview, scenarios, and use cases

User Stories:

1. As a user, I want to browse the homepage and services to learn what cyberhealth provides.

2. As a member, I want to create or update my information so that my profile can be up to date.
3. As a member, I want to ask for prescription requests.
4. As a member, I want to check the treatment my doctor has prescribed.
5. As a member, I want to view my doctor's information.
6. As a member, I want to comment on my treatment, doctor, and progress.
7. As a member, I want to state my health status.
8. As a doctor, I want to create or update my information so that my profile can be up to date.
9. As a doctor, I want to check my patient's information to know what to prescribe them.
10. As a doctor, I want to send the patient a prescription so that it can be sent to a pharmacy.
11. As a doctor, I want to check treatment requests so that I may prescribe treatment to these patients.
12. As a doctor, I want to check my patient's status so that I can follow their progress.
13. As a doctor, I want to comment on my patient's condition and treatment to help the patient.
14. As an Admin, I want to see all patients and doctors.
15. As an Admin, I want to see and edit patient and doctor's information
16. As an Admin, I want to add doctor into the system
17. As a non-member, I want to create an account so that I may receive cyberhealth services.

Use case diagram:



V. High-level functional requirements

Priority: 1-must have, 2-desier, 3-opportunistic

Non-member:

1. Browse the main homepage (1)
 - User can use the navigation bar and interact with the Home page and Service page buttons
 - The system shall let this user see the contents of the main Home page and Service page
2. Create an account: (1)
 - Users can create an account by entering their information such as username, email address, password, first name, last name, date of birth, phone number, and location and their medical records.
 - The system shall validate if the username is registered or not
 - The system shall validate all the information that users entered in the correct format
 - The system shall make sure there is no empty field when users create their account
 - The system shall display a message if an account is successfully created and redirect the user to the Sign in page.
 - The system shall store user's information provided above
 - The system shall provide unique ID for each user

Member:

1. Sign in (1)
 - Users can sign in if their username and password are registered with the system
 - Doctors can log in with their registered account
 - Patients can log in with their registered account
 - Admin can log in with account provided by developers
 - The system shall let users with registered account to log in
 - The system shall not let users with unregistered account to log in
 - The system shall display the error message when users log in with incorrect credentials
 - The system shall offer a register button in the sign in page
2. Edit Profile (1)
 - Users can see their interface depending on which role they are logging in
 - Doctors can see their own interface and edit their information such as first name, last name, date of birth, password, phone number, location, and email address

- Patients can see their own interface and edit their information such as first name, last name, date of birth, password, phone number, location, and email address
 - The system shall let users see and edit their information depending on their roles
 - The system shall let doctors see their information and edit it
 - The system shall let patients see their information and edit it
 - The system shall validate all the information that users entered in the correct format
 - The system shall make sure there is no empty field when users edit their profile
 - The system shall let users save information that they edited and store it
3. View main page (1)
- Doctors can see their information and patient's information
 - Patients can see their information and doctor's information
 - Admin can see both doctor and patient's information
 - The system shall let users browse their interface depending on their roles
 - The system shall restrict what each role should see
4. Manage as an Admin (1)
- Admin can see all the patients and their information
 - Admin can see, delete, and edit patient's information such as first name, last name, birthdate, phone number, location, gender, insurance, email, and doctor
 - Admin can see patient's medical history
 - Admin can see all the doctors and their information
 - Admin can see, delete, and edit doctor's information such as first name, last name, birthdate, phone number, location, gender, insurance, email, and license number
 - Admin can add doctor
 - The system shall have buttons for Admin to click on to see patient and doctor's information
 - The system shall have buttons for Admin to click on to edit patient and doctor's profile
 - The system shall have a list of all patients and doctors
 - The system shall let Admin browse and interact with buttons, list, etc
5. Manage as a doctor (1)
- Doctor can prescribe treatment to patients
 - Doctor can check patient's information
 - Doctor can update status of patients
 - Doctor can see a list of patients
 - Doctor can check request from patients
 - Doctor can comment and leave a note for patients to see
 - The system shall have a navigation bar

- The system shall have buttons for doctors to click on to prescribe treatment to patients
 - The system shall have buttons for doctors to click on to check patient's information
 - The system shall have drop-down menu for doctors to click on to update status of patients
 - The system shall have a list for doctors to see patients
 - The system shall have a list for doctors to see the requests from patients
 - The system shall have a comment section for doctors
 - The system shall store doctor's comment and display to patient
 - The system shall let doctors browse and interact with buttons, list, etc.
6. Manage as a patient (1)
- Patient can see their information and update it
 - Patient can request prescription
 - Patient can see doctor's profile
 - Patient can see their status of severity
 - Patient can check their treatment
 - Patient can comment and leave a note about their condition for doctor to see
 - The system shall have a navigation bar
 - The system shall have buttons for patients to click on to request treatment
 - The system shall have buttons for patients to click on to check their prescription
 - The system shall display a status of patient
 - The system shall display doctor's profile so that patients can contact
 - The system shall have a comment section for patients
 - The system shall store patient's comment and display to doctor
 - The system shall let patients browse and interact with buttons, list, etc.

VI. List of non-functional requirements

1. Performance: loading time should not exceed 1 second for users
2. Reliability: users can access the website 98% of the time without failure
3. Recoverability: if problems happen to the website, it should be recovered no more than three days for major ones and 8 hours for minor ones
4. Storage: The storage of our system will use lamp.cse.fau.edu server holding our mySQL databases within an unknown capacity
5. Expected Load: Our system is expected to make allowances for up to 50 users at the same time
6. Security ***: Only admin can view doctors and patients' information. Only doctors can view a patient's medical record

7. Compatibility: the website must work on multiple browsers (chrome, safari, etc.) and devices (tablet, phone, laptop, etc.)
8. Usability***: the website must be user-friendly and prioritize user experience
9. Data integrity***: the system must keep all doctors and patient data secure and fully back-up for every record
10. Easy to use: the website should be easy to use to any users, even in their first visiting.

***: prioritize requirement.

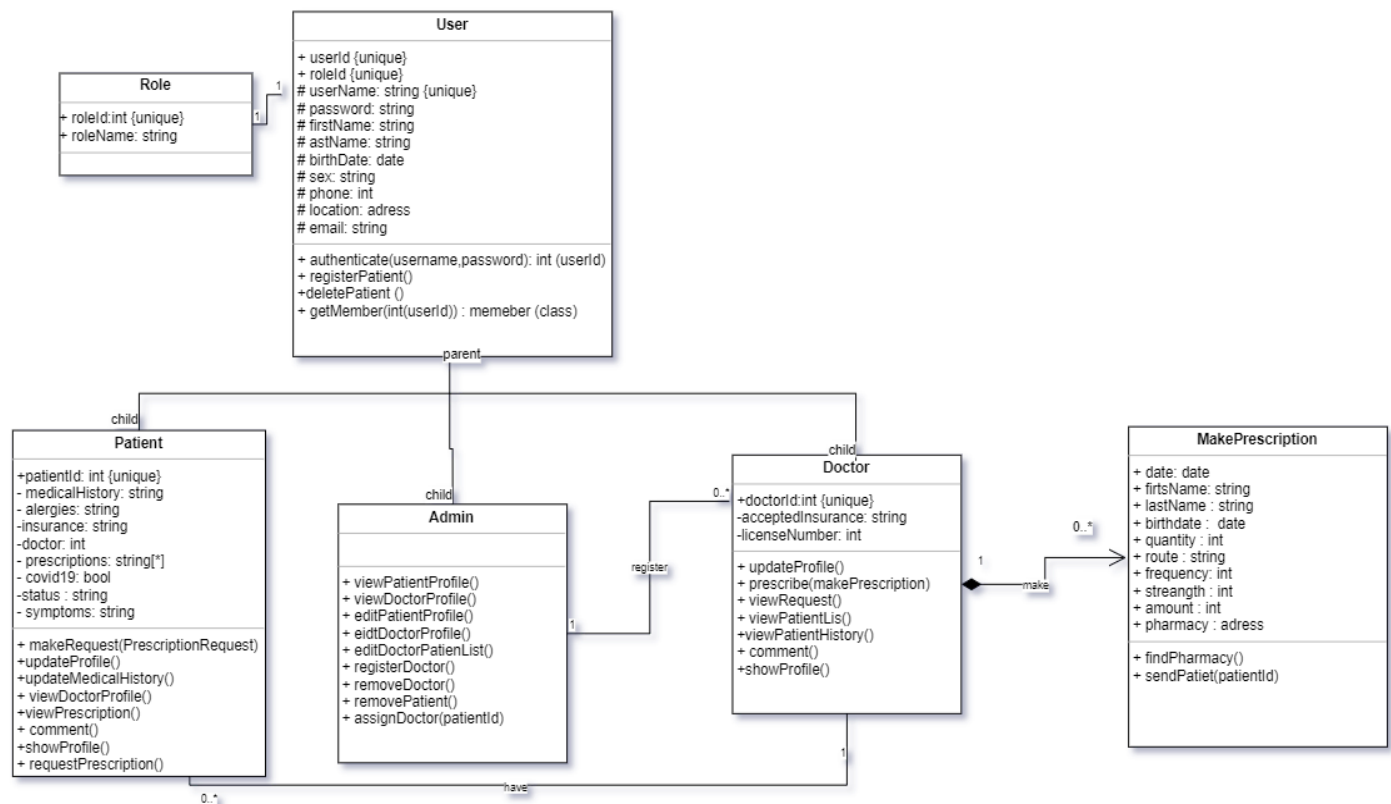
VII. High-level system architecture

1. Lamp server: we will be hosting our Fall 2021 project on https://lamp.cse.fau.edu/~cen4010_fa21_g22/
2. WhatsApp: is the main communication tool for us.
3. Jira: we will use Jira to track our project process and tasks.
4. GitHub: will be used to building and editing our project together.
5. PhpAdmin database: will be the main database for the project.
6. Languages:
 - a. HTML: will be used to display all the documents on the website.
 - b. CSS: will be used to decorate the website and pages.
 - c. PHP: will be used to manage data and contents.
 - d. JavaScript: will be used for most back-end development of the project.
 - e. Bootstrap: will be used for the initial construction for the website.
7. DB organization: Describe the main database schema/organization (high level), e.g. list main DB tables and items in each DB table
 - a. Patients: patient_id, user_name, password, first_name, last_name, gender, location, phone_number, doctor_assigned, insurance, email, current_medication_bool, current_medication_note, allergies_bool, allergies_note, diagnose, diagnose_status, feelling_note, syptoms_note
 - b. Doctors: doctor_id, user_name, password, first_name, last_name, birth_date, gender, phone_number, location, insurance_accepted, license_number, email.
 - c. Prescription: doctor_id, patient_id, medication, quantity, route, frequency, strength, amount, pharmacy_name, pharmacy_address, warning.
8. Media storage: Decide if images and video/audio will be kept in file systems or in DB. Describe any other special data format requirements like for video/audio/GPS etc.
 - Since we are a small-scale product, we will save all media file in the lamp sever folder, not in the database.
9. Search/filter architecture and implementation: what will be the algorithm for search; what DB terms will be searched, how it will be coded and organized in the DB. Similarly, say what DB items will be filtered/sorted.

- As of now, we do not see if we have a chance to use any specific search/filter implementation in our project.
 - However, for security purpose, we use hash function to hash our password and then insert to our database.
10. Your own APIs: Describe and define at high level any major APIs that you will create
- We do not use any API in our project.
11. Describe any significant non-trivial algorithm or process (like rating, ranking, automatic prioritizing of items etc.
- We use hash function to hash password of users for security purpose.
 - We might think about implementing algorithms to automatically prioritize of items or filter items if we get a chance and time is allowed. It could be one of our desired features.

VIII. High-Level UML diagrams

Class Diagram



Deployment Diagram



IX. Identify actual key risks for the project at this time:

- Schedule risk: At the beginning, we intended to do a full-scale product that can connect patients, doctors, and offices nationally. However, due to a three members team and our intensive schedule, we may not be able to delivery the product as initially plan.
- Action: We reduce the scale to patients and small offices to fit our ability. In addition, we believe, if we can create a great and scalable product, we can expand it in the future.
- Technical risks: None
- Teamwork risks: Ineffective communication. As a three-member team, we must keep each other updated to avoid project delays.
- Legal/content risks: We do not expect to encounter any legal or content risks.

X. Team and checklist

Team:

Group name: 22
 Scrum master: Huy Nguyen
 Product owner: Nha Tran
 Front End Developer: Nelly Delgado Planche
 Back End Developer: Huy Nguyen

Checklist:

- Team decided on basic means of communications: DONE
- Team found a time slot to meet outside of the class: DONE
- Front and back-end team leads chosen: DONE
- GitHub master chosen: DONE
- Team ready and able to use the chosen back and front-end frameworks: DONE
- Skills of each team member defined and known to all: DONE
- Team lead ensured that all team members read the final M1 and agree/understand it before submission: DONE

XI. History table

Document	Date	Note	Professor's feedback
M1 Proposal	09/28/2021	First M1 submission	
M3 Updated	10/25/2021	Revise M1	

XII. References

Jafari, L. (2020, December 26). *What Are Non-Functional Requirements? Types and Examples.*

WINaTALENT | Blog. <https://winatalent.com/blog/2020/05/what-are-non-functional-requirements-types-and-examples/>

Funke, D. U. T. (2021, August 9). *Fact check: Hospitals staff near max capacity, but COVID-19 isn't business as usual.* USA TODAY.

<https://eu.usatoday.com/story/news/factcheck/2021/08/09/fact-check-covid-19-surge-overwhelms-hospitals-beyond-bed-capacity/5472960001/>

Healthcare Workers. (2020, February 11). Centers for Disease Control and Prevention.

<https://www.cdc.gov/coronavirus/2019-ncov/hcp/clinical-guidance-management-patients.html>