University of California,
Santa Barbara

CS189 Fall'17 Capstone

VR Telemedicine

Product Requirement Documentation

Jinfa Zhu
Kenneth Chan
Shouzhi Wan
Xiaohe He
Yuanqi Li

Supervised by
Ole Eichhorn
Helen Hawkins
Nate Pincus
Marco Pinter
Jazarie Thatch

Contents

1 Introduction

1.1 Problem

Doctors are limited by the way they access patient data. Current interfaces are a step up from excel spreadsheets and file folders with attached scan images, but can be improved.

1.2 Vision

As virtual reality (VR) becomes more mainstream, we recognize the potential for using VR to create a more powerful and efficient user interface, aimed towards enhancing physician experience and making their jobs easier. A VR interface, coupled with hand gesture detection, can make the beautiful cinematic interfaces from science fiction into reality.

1.3 Challenges

Despite the popularity VR has gained, it is still a technology that is rather immature comparing to other fully developed technologies. Intense research with relatively few resources available is needed to get familiar with VR-related software development. Current challenges in this field, such as generally low resolution and video quality, are factors that should be taken into consideration. Human-computer interaction, in particular, is a problem lots of researchers are working on, and is closely related to this project. Another downside of VR in general is its high expense, which underlines the need to make full use of VR-specific features to provide new and useful services that cannot be done in traditional platforms, in order to make this platform worth its price.

1.4 Innovations

Comparing to traditional interfaces, an interface utilizing VR provides much more future probabilities for doctors with remote access. The first phase of this project implements a dashboard displaying and providing easy access to real-time patient information. The advantage this dashboard has is the significant increase of the amount of information a doctor can see in his/her field of vision, and the convenience of navigation through this system. Although this phase itself is only improving the performance of current features, this VR platform we are building will open up further possibilities for brand new features, such as monitoring conditions of a patient using moving cameras this doctor controls, instructing surgeries remotely, or several doctors working in the same virtual space to collaborate. Although we will not be able to explore all these possibilities, this project will serve as the first step to the realization of many features that currently live in people's imaginations.

2 Project Specifics

2.1 Objectives

The main goal of our project is to give doctors more accessibility to real-time patient information and records through an immersive user interface. The VR interface will support hand gestures in allowing doctors to retrieve vital signs and accessing simultaneous data and imagery windows.

2.2 Background

Modern telemedicine solutions allow doctors to see patients remotely over video conferencing, and interact with them on a limited level. However, this user experience is restricted by traditional human-computer interaction methods, such as with a keyboard and a mouse. Touch screen innovations expand the capabilities of the user interface by permitting swipe, pinch, and rotate gestures to navigate a 2D menu, but are not as powerful as a 3D virtual reality interface.

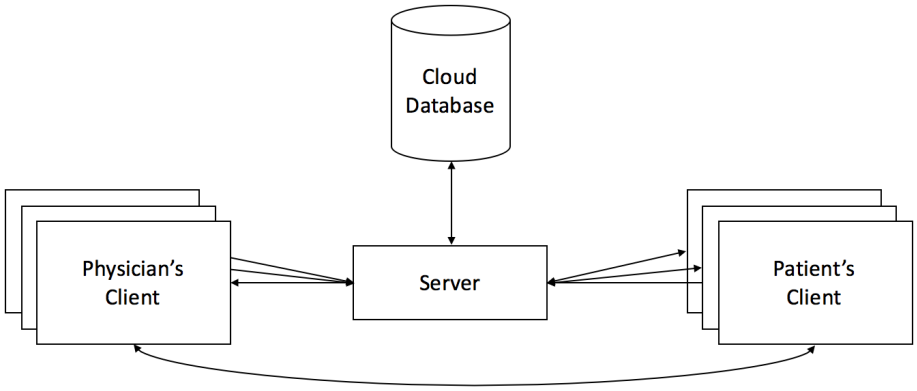
2.3 Assumptions

- Data (vitals data, live video, brain MRI scans, lab results) will be readily available.
- Internet connection between dashboard client and data servers will not be dropped.

3 System Architecture Overview

3.1 High level diagram

Illustrated below is our preliminary system architecture. There are four parts of this architecture: Physician's client, Patient's client, Server, and Cloud database. Arrows illustrate communication between respective entities.



3.1.1 Physician's client

The physician's client consists of a desktop computer and a VR device. The physician can connect to the server by logging in via the webapp. This will retrieve the physician's account data and patient data associated with the physician. Usually, the physician's client will only communicate with the server, which handles all data and messages transmission.

In an emergency, a video/radio call will be established directly between a physician's client and a patient's client to guarantee performance.

3.1.2 Patient's client

The patient's client is an application that connects to the server to retrieve and display their data. In addition, the patient can update personal information and send messages to their physician. This is similar to the physician's client. It will only communicate with the server, except in the case of an emergency, in which the patient's client will initiate a connection with the physician's client.

3.1.3 Server

The server is responsible for responding to all user connections. It connects to the database to retrieve and update the data.

3.1.4 Cloud database

The database stores all user data, including users' account data and messages. It connects to only the server to guarantee security.

4 User Stories

4.1 Physician identity and device registration

As a doctor, I want to access the product website, so that I can register an account for myself and my VR device and update my information.

4.1.1 Acceptance criteria

- A physician cannot be registered without a device serial number
- All updates made by a physician should be synchronized on the database immediately

4.1.2 Non-functional requirements

- Front-end is implemented using Bootstrap
- Back-end is implemented using Ruby on Rails
- All information is stored on a database managed by a server

4.1.3 Prototype and tests

`https://github.com/yuanqili/CS189-F17/tree/master/web/Server`

4.2 Physician Login

As a doctor, I want to see a login page on my computer so that I can log in and view my VR dashboard.

4.2.1 Acceptance criteria

- Login fails if incorrect user credentials are submitted

4.2.2 Non-functional requirements

- Use Unity to build a canvas for log in page
- Use Unity to build login buttons and functions of the buttons
- Use C# to access the database
- Design ER diagram and relational Schema
- Implement the Database using MySQL
- Create sample data for login
- Compile and test the login system

4.2.3 Prototype and tests

https://github.com/yuanqili/CS189-F17/tree/master/src/unity-ui/Assests/_Scene

4.3 Physician main dashboard navigation

As a doctor, I want to see a dashboard on the VR screen, so that I can view a list of my patients profile.

4.3.1 Acceptance criteria

- Patients' record should be listed in a curved UI

4.3.2 Non-functional requirements

- Put in the canvas and the 3D background
- Implement button functions and keyboard interaction
- Implement windows and canvas after click on the button
- Connect the VR and test the dashboard
- Re-modify the Dashboard based on VR testing

4.3.3 Prototype and tests

https://github.com/yuanqili/CS189-F17/tree/master/src/unity-ui/Assests/_Scene

4.4 Patient specific dashboard

As a doctor, I can tap on the profile so that I can see all that patient's data and record.

4.4.1 Acceptance criteria

- On click, the scene should change to patient's electronic record
- Patient's record scene should list all information that has already stored in the database

4.4.2 Non-functional requirements

- Design a patient info dash board
- connect different windows to different database contents
- Put in the fake information about the patients
- profile shows a graph of recent change if necessary
- pop up a small window show detail of certain info (eg. brain scan) if clicked

4.4.3 Prototype and tests

`https://github.com/yuanqili/CS189-F17/tree/master/src/unity-ui/Assests/_Scene`

4.5 Doctor Controlled Windows

As a doctor, I want to move the windows around so that I can fetch and read any documentations

4.5.1 Acceptance criteria

- Able to move, zoom, and select windows use controller

4.5.2 Non-functional requirements

- use C# to implement, select, open and close
- finger gestures zoom in and zoom out windows
- hand gestures move and throw windows
- Test the controller on VR and modify the control system

4.6 Record and Documentation Update

As a doctor, I want to record my prescriptions and update the documentations so that I can have references next time.

4.6.1 Acceptance criteria

- Hold on record button and speak out prescriptions
- A record should be uploaded to the database
- the patient should able to download the record from the server.

4.6.2 Non-functional requirements

- System need to generate a audio record of what doctor says
- store the prescriptions into the database
- add documentation functions to VR system

4.7 Real-time Doctor-Patient Communication

As a patient, I want to talk with doctors in real-time so that I can follow their instruction.

4.7.1 Acceptance criteria

- With good network condition, patient should transmit video data and doctor should transmit voice data.
- videos and voice data should be stored if necessary

4.7.2 Non-functional requirements

- implement video chat function in patient app
- add function to doctor.

4.8 Database Manager

As a service provider, I want to create doctor and patient profiles so that they can get access to the server.

4.8.1 Acceptance criteria

- Other people except service provider should not have access to create these profiles

4.8.2 Non-functional requirements

- implement management application to change user info

4.9 Camera Movements

As a doctor, I want to move around the camera so that I can monitor the patient

4.9.1 Acceptance criteria

- Move the head should see different scenes

4.9.2 Non-functional requirements

- Use controller to move around the camera

4.10 Emergency System

As a patient, I can send an alert to the Doctor so that they can instruct me when there is a emergence.

4.10.1 Acceptance criteria

- Send alert from phone and VR pop up a window

4.10.2 Non-functional requirements

- Implement emergency function in patient application

4.11 Doctor-doctor Communication

As a doctor, I can contact with other doctors in VR system so that we can share the opinions about some patient cases.

4.11.1 Acceptance criteria

- Login and select communication, Should able to see online doctors
- Send request to join the cases
- Two or more doctors in the same scene
- Have access to the same patient case

4.11.2 Non-functional requirements

- Real time chat system for doctor application

5 Appendices

5.1 Technologies employed

5.1.1 Server

- **Bootstrap**. Simple yet pretty front-end web pages can be written using Bootstrap, which saves our a lot of energy.
- **Ruby on Rails**. RoR is our main framework to build website backend. It provides stable and powerful building tools to setup the backend and database connection.

5.1.2 Database

- **MySQL**. The main advantage of MySQL is that our database administrator is very familiar with it and has years of experience using it.

5.1.3 Virtual reality

- **Unity**. The Unity game engine provides a full set of developing tools to implement VR applications. It also has an very active community which provides us solutions and assets that boost our development.
- **Oculus Rift**. Oculus Rift is one of the most popular VR devices. It also has an active community and works very well with Unity.