

MedCapture

CS 4000 - Senior Capstone: Design Document

Tanner Jones, Ryan Springborn, Ben Shepherd, Phillip Hidayat

1. Executive Summary

MedCapture is an innovative software system that aims to empower medical professionals by facilitating accurate documentation and effective communication during patient examinations. This document provides an overview of MedCapture, its purpose, intended users, key functionalities, and what sets it apart from existing solutions. The goal of MedCapture is to create an intuitive interface for medical professionals, specifically ophthalmologists, to use when conducting/documenting eye examinations. This interface will incorporate an image of the eye (left or right) in which the user would be able to interact with various areas of the eye and document what they are seeing in the patient to streamline the examination process.

I. Purpose and Scope:

The primary purpose of MedCapture is to enhance the accuracy and efficiency of medical record-keeping, ultimately leading to better patient care. Ophthalmologists, in particular, can greatly benefit from MedCapture's intuitive touch interface, which simplifies the process of documenting and communicating their findings.

The result is not only enhanced precision in documentation, but also the potential to reduce the margin of error in communication between medical professionals and in patient records. MedCapture's intuitive design is a powerful tool that seeks to transform the way ophthalmologists work, providing them with a sophisticated yet user-friendly platform to improve the quality of care they deliver to their patients.

II. Intended Users:

MedCapture is designed for medical professionals, with a primary focus on ophthalmologists. These specialists demand a high degree of precision in their work, and MedCapture caters to their specific requirements. However, the system's intuitive design and functionalities are versatile enough to benefit a broader range of medical specialists who need to document and communicate their findings accurately. This versatility extends the potential reach and impact of MedCapture, offering value to various medical specialties beyond Ophthalmology.

2. Background and Technical Requirements

In this section, we explore the landscape of similar software, emphasizing how MedCapture distinguishes itself from existing solutions. We also delve into the required technologies and assess our team's current skill set.

I. Existing Software Landscape:

One of the largest Electronic Medical Record (EMR) systems is Epic's MyChart System, a well-established healthcare software. This software offers a variety of helpful tools for both providers and patients pertaining to storing and editing medical information over the internet. However, MedCapture stands out with its unique features and advantages, including an intuitive touch interface and seamless integration of advanced imaging technology. The most significant advantage MedCapture holds over MyChart is its innovative approach to documentation in eye exams, achieved through the addition of intuitive images and the removal of complex lists and checkboxes. This streamlines the documentation process, making it significantly more user-friendly. Our goal is to strike a balance by offering a system that is simpler to learn and use, while still providing advanced features and options that meet the specific needs of medical professionals, including Ophthalmologists. MedCapture aims to provide a user-friendly, efficient, and advanced solution, differentiating itself from existing software in the market like MyChart.

II. Technology Requirements:

We have thought about a variety of different technologies that we are planning to use for MedCapture. Firstly, high-definition images generated by AI, possibly through Dall-E, are integral to enhancing the project's visual appeal. These images will play a significant role in creating an engaging and immersive user experience. For robust and scalable data management, we plan to use AWS DynamoDB, a NoSQL database. This choice ensures efficient data handling while offering the flexibility required for our application's dynamic requirements. When it comes to hosting our web server, we've chosen AWS CloudFront and S3, facilitated through Amplify Hosting. This combination guarantees reliability and fast content delivery. For the frontend of our application, we will be using React. React's component-based architecture will enable us to create a seamless and interactive user experience.

III. Skill Assessment:

- Tanner Jones
 - Strong understanding of React's front-end framework.
 - Extensive knowledge of the AWS cloud with multiple certifications achieved.
 - Experience building full end-to-end solutions that accomplish business requirements.
 - Strong understanding of serverless technologies and their advantages in designing systems with low maintenance.
- Phillip Hidayat
 - Robust background in backend development.
 - Proficient in frontend, database management, and security.
- Ryan Springborn
 - Robust background in developing full-stack applications.
 - Strong proficiency in front-end, backend, and database technologies.
- Ben Shepherd
 - Versatile software engineer with proficiency in multiple programming languages, databases, algorithms, and networking.
 - Excels in adapting to project-specific needs and promoting Agile practices for efficient project management.

IV. Software/Hardware Requirements:

The plan for MedCapture is for it to be a fully static web application so the hardware requirements would be to have a device that can connect to and access the internet. Similarly, the software requirements would be to have access to an internet browser to use our application from.

Everything below here is not due yet

Requirements Analysis (2-3 pages):

This section defines the major components of MedCapture, outlines individual responsibilities within the team, and lists the system's features.

System Architecture:

- High-level description of software components
- Explanation of module integration
- System design diagram

Personnel:

- Assignment of responsibilities for each component
- Justification for team members' qualifications

System Features:

List of key features categorized by priority:

Rank 1: Infrastructure and Core Functionality:

- Core database tables/interactions
- Basic UI/UX for main functionality
- Local host implementation
- Main algorithm or plugin for future development
- Expected completion by the end of Alpha phase

Rank 2: Planned Features:

- User/roles authentication and authorization
- Database/data integration
- Core algorithms
- Core UI/UX enhancements
- Software engineering practices (e.g., cloud deployment, API documentation)
- Expected completion by the end of Beta phase

Rank 3: Advanced Individual Features (Bells and Whistles):

- Examples of advanced, project-specific features
- Individual ownership for development during the Final Phase

Software Engineering Tools and Techniques (1 page):

Discussion of the software engineering tools and techniques the team plans to employ, including versioning, bug tracking, video conferencing, agile methodologies, scrum, stand-up meetings, task management, and CI/CD.

Timeline (1-2 pages):

A schedule matrix outlining the three 4-week phases (Alpha, Beta, Final) of CS 4500. Each cell contains specific tasks and objectives for team members, with possible major milestones included. The timeline serves as a flexible guide for project planning and monitoring progress.

Appendix A: UI Sketches:

A collection of annotated sketches depicting the projected user interface, illustrating how users will interact with MedCapture.

Appendix B: Use Cases:

A set of use cases from both user and system perspectives, detailing how MedCapture will be used in various scenarios, with references to UI sketches when applicable.