6-14-2018

Diagnostic Color Strip Reader for World Health Partners Clinics

Steven Hu
Santa Clara University, shu@scu.edu

Jisoo Park
Santa Clara University, jpark2@scu.edu

Geminiano Romano Yabut
Santa Clara University, gyabut@scu.edu

Santa Clara University
DEPARTMENT of COMPUTER ENGINEERING

Date: June 13, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Geminiano Yabut, Jisoo Park, and Steven Hu

ENTITLED

Diagnostic Color Strip Reader for World Health Parters Clinics

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THESIS ADVISOR

DEPARTMENT CHAIR

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
SOFTWARE ENGINEERING

Diagnostic Color Strip Reader for World Health Partners Clinics

by

Steven Hu
Jisoo Park
Geminiano Romano Yabut

Santa Clara, California
June 14, 2018

Diagnostic Color Strip Reader for World Health Partners Clinics

Steven Hu
Jisoo Park
Geminiano Romano Yabut

Department of Computer Engineering
Santa Clara University
June 14, 2018

ABSTRACT

Despite the advancement of medical technology, many people in developing countries like India and Kenya still suffer from treatable diseases. In many of the health clinics in these areas, color strips are used for checkups and diagnosis of diseases. However, a big problem with these color strips is that the diagnosis of color strips take a long time because they have to be manually checked. Currently, World Health Partners (WHP) works with doctors and hospitals in India and Kenya to provide more accessible healthcare through telehealth networks to get consultations from rural clinics to specialists at hospitals. We are working with WHP to streamline the process of color strip diagnosis, by creating an application that goes through the process of reading a color strip in a single step. Our application analyzes an image of a color strip and returns the concentration of the different factors being tested on the color strip. By doing so, we provide a precise analysis of color strips, instead of having to wait for a specialist.

Acknowledgements

We would like to acknowledge the following people who contributed and assisted us throughout this design process.

Thank you to our senior design advisor Dr. Silvia Figueira for guiding us throughout this whole process.
Thank you to Allan A. Bez Morales at Frugal Innovation Hub for all your help and being a great liason for us and World Health Partners.
Thank you to Dana, Jeff, Lilian, and Ryan in the bioengineering group we worked in collaboration with.
Thank you to Kamal Talreja from World Health Partners for working with us on this project.

Table of Contents

List of Figures

List of Tables

1 Introduction

Here in the 21st century, medical technology has advanced to the point of curing many of the worlds ailments. Unfortunately, many people in the developing countries still suffer from entirely treatable diseases. People all over India, Kenya and similar countries have communities where people are unable to see a doctor, or receive treatment and medicine. Doctors stationed at these locations tend to see upwards of hundreds of patients a day while also dealing with remote telehealth requests. This makes it difficult for many to get necessary healthcare services.

Currently, the World Health Partners (WHP) are working with doctors and hospitals all over India and Kenya to provide more accessible healthcare to rural and underserved communities. WHP focuses on creating technology-focused networks to help healthcare providers, entrepreneurs and communities get the tools they need to provide consultations and medication. WHP makes use of telehealth networks to get consultations from rural clinics to specialists at hospitals. WHP faces time limitation issues in certain kinds of consulting, such as giving diagnoses through reading instruments. One kind of routinely used instrument at these clinics is color strips. Color strips are used for different analyses ranging from malaria to contamination in urine. After doing the test, nurses or doctors at the clinic have to send the information out. However, specialists receiving the samples are often not able to send the results out in a timely manner.

At Santa Clara, our team has designed a solution to make processing color strips more efficient. Our proposed solution is a mobile application designed to help read color strips. The design is aimed at providing a precise analysis of color strips, instead of having to wait for a specialist, or having someone unqualified read the data. Given the color strip and the disease or contaminant being tested for, the application should outline whether or not there is a problem based on the color. We also used color analysis API to aid in the development of the application.

The application is designed to be compatible with tablets in order to take advantage of higher quality cameras. The application is focused around allowing the tablet to be inserted into a housing unit to ensure a clear picture of the color strip. The mobile-friendly design allows the product to be easily incorporated into medical procedures used by the doctors and nurses in the rural clinics.

2 Requirements

Our project meets the following requirements in order to fulfill the usability standards of our system. Many requirements directly indicate what our system must provide to the users of the system while others specify how our system operates.

2.1 Functional

- The system is able to take pictures.
- The system allows for the analysis of different types of test strips.
- The system correctly analyzes different color gradients in the strips in accordance with the lab results.
- The system will allow for the searching of patients from the database.
- The system will allow inputting new patients into the database.

2.2 Non-Functional

- The system must have a friendly and intuitive UI to allow patients and health workers to easily navigate the system.
- The system must be reliable in order to allow quick processing of data.

2.3 Design Constraints

- The system must be web-based.
- The system must be mobile compatible.
- The system must be compatible with the client's database and existing technologies.

3 Use Cases

Figure 1 and the descriptions following explain how a health worker will use our application.

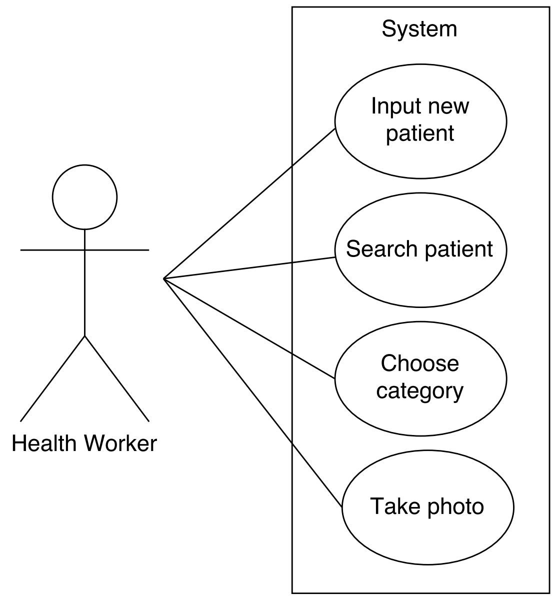

Figure 1: Use Case Diagram

1. **Input new patient**
   Goal: To input new patients to the database
   Actors involved: Health Worker
   Pre-conditions: None
   Steps: Input in all necessary information of new patient as needed
   Post-conditions: New patient is added to the database
   Exceptions: None

2. **Search patient**
   Goal: To search for the correct patient to add new data
   Actors involved: Health Worker
   Pre-condition: None
   Steps: Search for the patient needed
   Post-conditions: After choosing patient, system will prompt you to choose a category
   Exceptions: None

3. **Choose category**
   Goal: To choose the correct test to analyze
   Actors involved: Health Worker
   Pre-condition: Patient has been chosen
   Steps: Choose the correct test that needs to be analyzed
   Post-conditions: Displays the results of the analysis
   Exceptions: None

4. **Take photo**
   Goal: To successfully take a photo to be analyzed
   Actors involved: Health Worker
   Pre-condition: Photo must be taken in a housing unit
   Steps: Take a uniform picture within the housing unit
   Post-conditions: Results will be analyzed and stored into the database
   Exceptions: None

4 Activity Diagram

This section will show and explain the associated activity diagram for our solution.

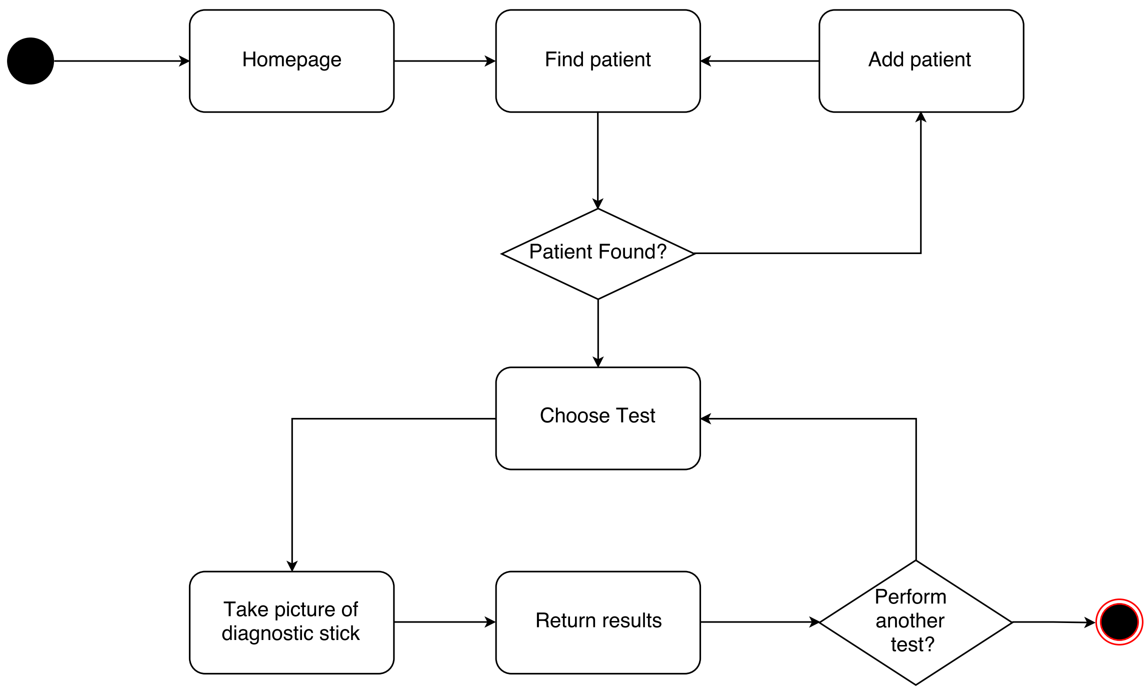

Figure 2: Activity Diagram

5 Architectural Diagram

The architecture we used was a client-server architecture. We wanted one centralized entity to take care of running most of the backend, while allowing anyone to connect and interact with the system. Because the system will be used by multiple clinics and health workers simultaneously, it is important that it be easy to add clients and have multiple clients. Since clients in this architecture are independent of each other, this allows for simple addition of new users.

Our server will take care of handling connections, and working with the database. The server will handle the test analysis calculations and the queries to the database.

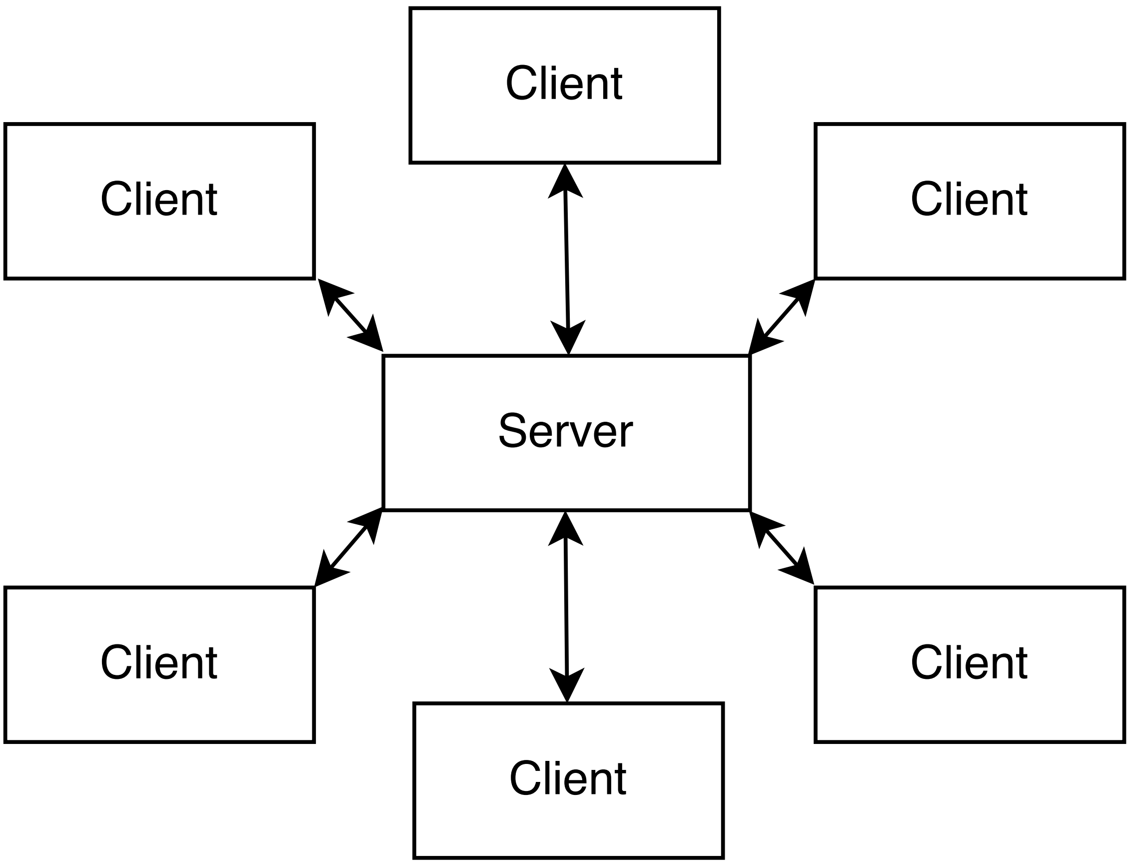

Figure 3: Architectural Diagram

6 Technologies Used

6.1 Backend

Python Programming language used for server side scripting and computer vision computations.

Flask Python-based web application framework to allow communication between the server and the front end.

OpenCV Open source computer vision library with python support.

6.2 Frontend

HTML The standard markup language used to create basic web pages.

CSS Used to create styling for web pages.

JavaScript Scripting language used to create reactive web pages.

Bootstrap CSS library used to create mobile-friendly web pages.

jQuery Fast, small, Javascript library. Version 3.2.1

7 Design Proper

7.1 Front End

Our front end is written largely in HTML, CSS and Javascript. Our front end is built as a mobile friendly web page. We make the web page mobile friendly by using libraries like bootstrap for CSS and jQuery for Javascript. The following figures show our front end design.

This sections shows the conceptual models of our system. The following figures give an idea of how the different parts of our system will look. Note that the results page only currently displays the results for the compounds we have tables for. The others are just the RGB values.

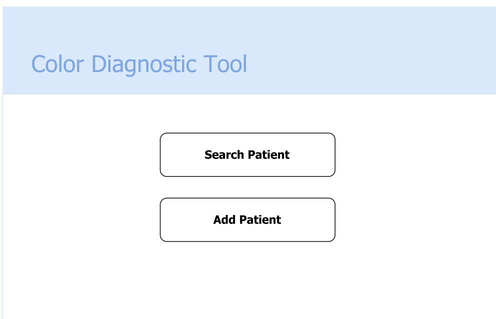

Figure 4: Homepage

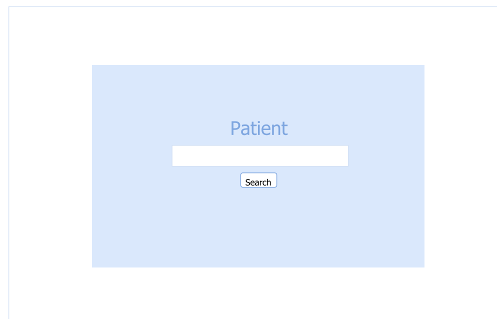

Figure 5: Search Patient

Figure 6: Add New Patient

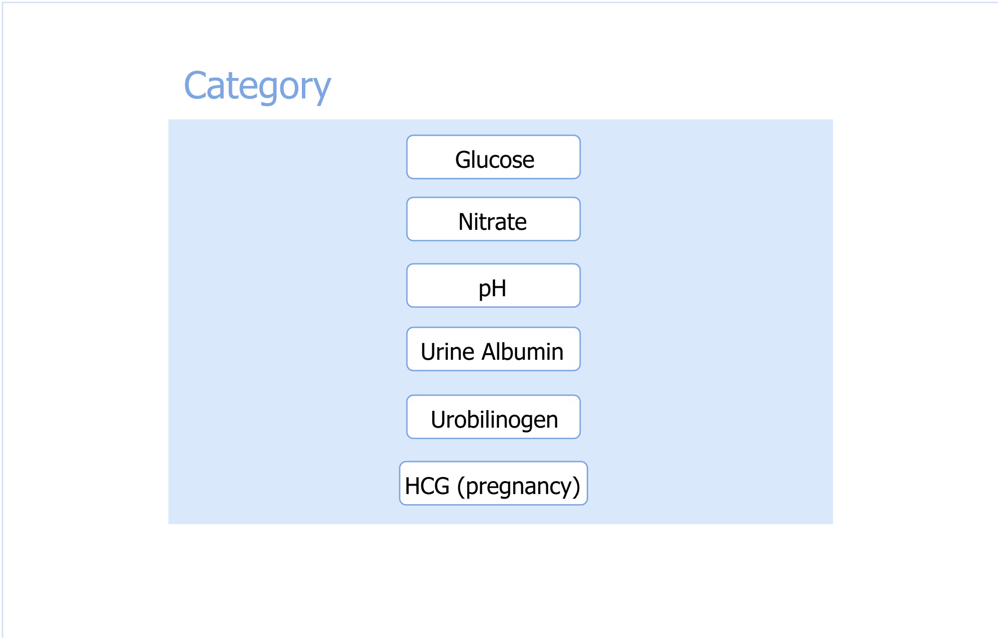

Figure 7: Choosing Type of Analysis

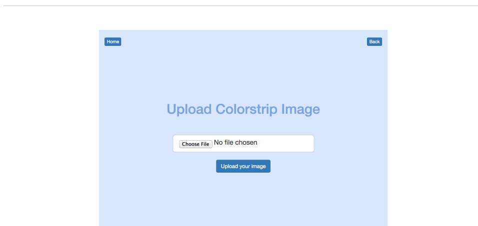

Figure 8: Take photo

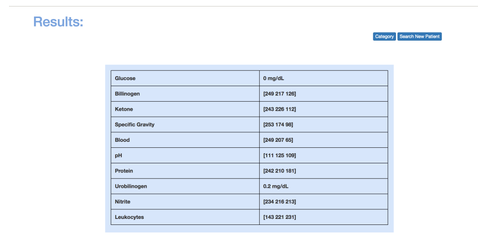

Figure 9: Displayed Results

7.2 Back End

Our back end is written in Python with Flask as a web application framework. All of the front end is loaded using Flask 'templates'. The image uploading page calls the imaging algorithm on the upload of a file. The image uploading page also handles bad uploads, which will return a prompt to the front end for the user to re take the photo. The Results page presents the results returned by the imaging algorithm. The Back End also gets rid of all the image files created by the imaging algorithm that do not need to be stored, in order to save space.

7.3 Imaging Algorithm

The imaging algorithm runs within the back end component of the web application. The algorithm serves to correlate various colors on the image of the test strip with the specific concentrations of each compound within the test strip. The whole algorithm is written in python, using supporting functions from OpenCV, a python-supported computer vision library.

The algorithm will receive a photo of the color strip in the housing unit taken by the web application. The algorithm begins by manually cropping a specified section of the image to capture just the color strip (Figure 10).

After that, a bitwise not is performed on all the white in the image to turn them black (Figure 11). This is used as a preprocessing effect to help with square detection later on.

Figure 10: Manually cropped color strip

Figure 11: Color strip with white turned black

The algorithm then uses OpenCV's various function to perform square detection. It then cuts and saves each color sequentially (Figure 12).

Figure 12: Colors individually cropped

Finally, for each cut image, the algorithm then analyzes the dominant color using K-Means clustering and returns an RGB value. The RGB value is then compared against a lookup table containing concentration values, calculated from equations given to us by our partner bioengineering group (Appendix B). The lookup table is implemented by mapping RGB values as points and on a euclidean plane and connecting the data point with its closest neighbor on the lookup table, by euclidean distance. Invalid colors are marked as errors. Invalid colors appear when the color strip is fresh out of the box, and the squares do not have urine on them.

8 Design Rationale

8.1 Frontend

We chose to make a simple user friendly UI because our users will be health care workers who have limited to basic understanding of diagnostic tools. It will be an intuitive design that allows even first time users to be able to easily use and maneuver through the system. All pages of our system will be easy to use as it will be straight forward and only contain the necessary functionalities of our system.

8.2 Backend

We chose to make use of Python and Flask to simplify our design and code base. Python was used to be compatible with the Imaging Algorithm, which we wanted in Python. Flask as a web application framework allows for easy and quick communication between the front end and the back end. Flask also allows us to easily work with the front end. Flask's support of 'templates' allows us to easily integrate our front end design with our back end without having to deal with compatibility.

8.3 Imaging Algorithm

We decided to make the algorithm only partially involve manually cropping. We did this because we hoped the use of image recognition would decrease the amount of times a user might have to retake the image because they positioned the color strip incorrectly. The Image Algorithm makes use of Python because it is powerful and convenient to use with scientific calculations, and computer vision is calculation heavy. K-Means clustering was used over a simpler averaging algorithm because sometimes a crop would leave some black spots in the image (As in Figure 12). K-means clustering would return a dominant mean color without taking the black spots into account.

9 Risk Analysis

The following are some risks associated with our project. Table 1, shows various risks as well as their consequences as a result. Each risk has a mitigation strategy as shown in the right-hand column. Probability, severity, and impact (probability * severity) are noted as well.

Table 1: Risk Analysis Table

Risk	Consequences	Probability	Severity	Impact	Mitigation
Lack of experience with technology	Slow development	0.8	5	4	Study documentation Constant Communication
Time	System may not be completely finished on time	0.5	7	3.5	Set deadlines and meetings Prioritize core features
Bugs	Delays in development	0.99	2	1.98	Clean coding style Well commented code
Group member getting sick	Loss of productive time More work for other members Development may take longer	0.25	7	1.75	Constant communication Eat healthy, sleep well
Loss of Code	Delays in development	0.1	8	0.8	Using a version control system Saving local backups

10 Development Timeline

As detailed in the figure below, our development timeline for completing and testing our product spanned entirety of the winter quarter. Our development timeline was split into 4 phases: Requirements, Design, Testing, and Documentation. For the color coding, although a member was assigned a task, all members worked together on all issues - the color only denotes who is primarily in charge of that aspect's completion.

For the Requirement phase, we finalized the needs of this project. To do so we need to sync the needs of our partners, the World Health Partners, as well as the Bioengineering team working on the color strips here at SCU. After the design requirements were collected and unified, we began on designing the solution.

For the Design phase, the process was been split in 4 categories. The first and most important aspect of the design is the color detection algorithm, and was the most difficult task to complete. Implementing a database to hold patient info was equally time-consuming. The other tasks, such as UI and web scripting, were less difficult to implement but just as important.

Testing began after around 2 weeks of design. Main stress points included testing the algorithm against verified data, and making sure that it returns correct data. Another important aspect was the integration of the database and web scripts, and making sure the UI protects against edge cases and improper input.

Finally, documentation began halfway through the winter quarter. The Design review took place early in Winter, and writing the Design Report took place afterwards. Elements of the report were used to complete Final Presentation, which took place in Spring.

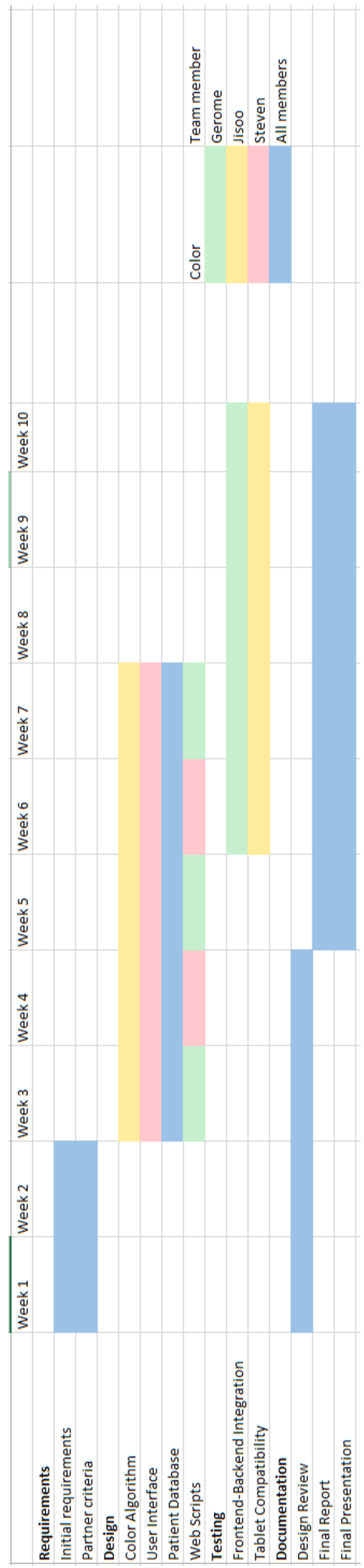

Figure 13: Project Timeline Gantt Chart

11 Societal Issues

11.1 Ethical

Our application deals with medical information of patients, so an ethical concern would be the confidentiality of all this information. It is important that all information is secure and only used appropriately with confidentiality. It is also important that our application gives precise and accurate diagnosis.

11.2 Social

The implementation of our system will hopefully allow for more people in rural communities access to basic health care. The fast and accurate diagnosis of color strips will allow for doctors to see more patients.

11.3 Political

Our project does not really have any political implications. However, it is important to note that politics plays a critical role in health affairs.

11.4 Economic

The main economic concern of our project is the availability of electricity. As our application is designed for use with a tablet, making sure that having a power source to charge it can be a a problem for some communities that do not have access to it. Having access to an Internet connection is a future consideration. If the health workers cannot access the WHP database, saving the results of patient diagnoses would not be possible with the application.

11.5 Health and Safety

Health and safety issues are a big part of our project since our project deals with medical information of patients and gives diagnosis of color strips. We have to make sure that our project is accurately diagnosing patients because inaccurate data can be detrimental to a patient's life. Our application also works in conjunction with a special housing unit for the color strips and since the color strips currently used are urinalysis strips we also have to consider the hygiene of the process. The hygienics of this were handled by the bioengineering group, who designed the housing unit to work with the tablets we designed our application for.

11.6 Manufacturability

Since our project is a computer/mobile implementation there are no physical manufacturing costs. Instead, most of the cost will come from developing the application's future additions. Developing costs would include programmer wages, as well as potential software licenses to augment the application's functionality.

11.7 Sustainability

Our product will be sustainable – viable and useful – as long as color strips are used for health diagnosis in rural communities. The purpose of our product is to analyze color on urinalysis strips so as long as these medical strips continue to be implemented, our application will be able to scan them. Additional development to make future color strips compatible with our application would have to be taken into consideration.

11.8 Environmental Impact

Our project does not really have environmental impacts because it is a mobile application.

11.9 Usability

Usability was a very important factor we considered when developing our project. Since our application will be mainly used by health care workers and we do not know how comfortable they are with technology we made sure our application was easy to use even for first time users. We made sure there were only the necessary functionalities on each page so that it is easily navigable.

11.10 Lifelong learning

This project helped prepare us for the time when we will have to learn on our own. We had to research and learn things on our own when developing this project, so we were able to self-learn and grow as learners.

The project also served as a great introduction to Frugal Engineering. Learning to develop a solution that is designed for areas that cannot sustain newer technologies, like rural villages or less developed nations. Our project required us to develop an application with this in mind - for example, making sure the application can run on many older styles of tablets commonly used by WHP health workers. Integrating frugal engineering processes into this project helped us learn a skill that can be applied to countless other projects we may work on in the future as engineers.

11.11 Compassion

Basic health care is something we believe that all human beings should have access to. Therefore, we decided to work on this project that we believe will allow people in rural communities faster and easier access to health care. By speeding up the process of color strip analysis doctors are able to see more patients in rural communities.

12 Conclusion

12.1 Summary

Overall, we managed to complete most of the objectives set out for the project. Unfortunately, due to time constraints and difficulty synchronizing work across both teams, we were unable to integrate the patient database into the application.

Developing a Frugal application helped us as a team to develop a project in a different manner to traditional software development taught at Santa Clara University. As a team, we prioritized efficiency, usability and accessibility over cutting-edge advancements.

12.2 Lessons Learned

Working with multiple teams allows work to be completed faster, but not necessarily at the same pace: having the bioengineering team work independently of us allowed development on both ends to progress at a faster rate. However, this simultaneous work also led to branching paths that could not easily be reconnected.

One of the hardest parts of the application development was using the color recognition for different color strip sections to directly diagnose patients. Although we wanted to integrate this function into the application, the equations for the diagnostics were delegated to the bioengineering team. Ultimately, we were unable to coordinate with them to integrate the feature in time, and it was unable to be added to the demo we developed.

12.3 Advantages and Disadvantages

The main advantage of this application we designed is the efficiency and quality of life improvements our application has over traditional methods of reading a color strip. Having the application read the strip for our users helps health care workers increase productivity and prevent user error. In addition, future improvements to the app adds even more functionalities that make providing health care to rural communities easier to manage.

Of course, using the application requires more technology than simply using color strips manually. A reliance on technology means that communities without Internet or electricity would not be able to use our application. Another disadvantage is needing some technology experience to use our application. Although designed to be as intuitive as possible, a small amount of tablet experience is still needed to utilize the application.

12.4 Future Work

- WHP Database Integration

  We still need to integrate our application with WHP's database. Given the proper schema and the ability to access WHP's secure patient database server, the application would be able to store the results recorded by the application's image scan.

- WHP Application Integration

  We still need to integrate the application's functions with the companion Urinalysis Screening project that our bioengineering partners worked on throughout the year. This includes integrating the color finding algorithm we used, with the color evaluation algorithm their team developed to convert the color values into the corresponding chemical values. Currently, only a few of the compounds are supported.

13 References

Koeneman, Blair; Miller, Amy; Neumeyer, Joe; and Prince, Jake, ”MUMS: Mobile Urinalysis for Maternal Screening” (2016). Interdisciplinary Design Senior Theses. 22.

Telemedicine Solution. World Health Partners, worldhealthpartners.org/technology/telemedicine-solution/.

Appendices

A User Manual

The users of our system will be healthcare workers.

- Users will first need to login to continue to use the application.
- After logging in successfully users will be able to search for or add new patients.
- After finding the patient the user wants they will be prompted to upload or take a picture of the colorstrip they want to analyze. Users can either look through their camera roll to upload an image or be taken to the camera application to take a photo on the spot. If photo is not taken correctly users will be prompted to upload or take a new photo.
- After succesfully uploading an image to the application the application will do its analysis of colors.
- When analysis is done it will lead the user to a results page with all the results from the analysis.
- After seeing results, users will be able to choose another test to do or search for another patient.

B Lookup tables

Below are the lookup tables used when mapping rgb values to the concentrations of compounds tested by the Bioengineering Group.

Table 2: Glucose Concentrations

Concentration	**Red**	**Green**	**Blue**
0	144	211	197
60	150	200	162
100	153	193	142
130	155	188	128
160	157	182	116
250	160	165	86
500	150	120	62
1000	56	29	255

Table 3: Albumin Table

Concentration	**Red**	**Green**	**Blue**
0	118	119	121
60	120	122	124
100	121	122	126
130	121	123	127
160	121	123	128
250	120	122	130
500	109	113	126
1000	51	55	81

Table 4: Urobilinogen Table

Concentration	**Red**	**Green**	**Blue**
0	238	203	127
0.2	240	203	129
1	242	201	137
2	244	198	145
4	247	190	157
8	244	173	161