

CliniConnect

UserGuide

A Project By: Luke McDuff, Shweta Parihar, Sri Chaitanya Patluri and Benjamin Chrysler

Table of Contents:

Introduction.....	3
Administration Application.....	4
Home Page.....	4
Search Bar.....	5
New User Registration.....	6
Edit Clinic Info.....	7
Patient Info.....	8
Details.....	8
Vitals Log.....	9
Post-Visit Survey.....	10
Pre-Visit Forms.....	12
Appointment Reminders.....	13
Send Lab Paperwork.....	14
Patient Application.....	15
Navigation Bar.....	15
Home Page.....	16
Pre-Visit Forms.....	17
Post-Visit Survey.....	18
Vitals Log.....	19
Clinic Info.....	20
Lab Forms.....	21

Table of Contents (Continued):

Directions, Email and Call Clinic.....	22
Errors, Bugs and Deficiencies.....	23
Index.....	24

Introduction:

Thank you for choosing CliniConnect as your medical application resource. This document will serve as a guide to the use of our application. CliniConnect, as a service, offers separate modes of use. These modes are for clinic administration use and patient use, Furthermore, the application is intended for use by clinics and the patients that frequent the clinics that utilize CliniConnect. This means that use of this application will be limited based on accessibility from clinic to clinic.

The reason this application has been created is for the purpose of creating a better dialogue between patient and clinic. Many patients often do not have available times to sit and wait at a clinic to fill out paperwork, as most people do not have jobs that have paid time off. Also, clinics can stay updated with daily vital information as there are many people out there who need to keep information updated on a daily or weekly basis. Through our system, patients can leave feedback about their experience, This information can be used to help clinics better serve patients in the future. The system also includes a lab QR code generator, which allows patients to have their lab paperwork available through the application, instead of having to keep several sheets of loose papers that could become lost or damaged from clinic to lab. Some, basic features have been implemented as well, such as e-mail, phone and google maps connections.

As a result of trying to improve communications between patient and clinic, several options have been implemented in the administration application. Clinics can use our administration to register patients, track appointments, send reminders, send lab work details, as well as view patients edited information. Ease of use was heavily emphasized throughout the entire system and is quite intuitive as a result.

Administration Application:

In CliniConnect, the goal is to have each instance of the system isolated, so there is no login required for the administration application, or admin site, at this time. Since this is the first version, this may change in future versions. Following each description here, there will be an image highlighting points of interest in the user interface. Also, the admin site is available across multiple platforms including iOS, Android and Windows.

Home page:

The Home page is the the first page that you see when entering the admin site. On the Home page, the list of all the upcoming appointments are automatically populated, as you can see. There are four types of clickable components on the Home page. The search bar (in blue), register patient (in orange), clinic info (in purple) and each patient (each is clickable, but only the first is green for effective area demonstration per patient).

CliniConnect: Upcoming Appointments

Upcoming Appointments		
<input type="text"/>	Search	Logout
Patient: Myrtle Racheal Appointment Date: 2016-03-23 12:45 Physician: Dr. Leontine		
Patient: Renda Debbie Appointment Date: 2016-05-13 3:00 Physician: Dr. Krysten		
Patient: Clara Roselia Appointment Date: 2016-06-18 2:30 Physician: Dr. Candance		
Patient: Valentina Jani Appointment Date: 2016-07-07 1:45 Physician: Dr. Raymon		
Patient: Chu Gerry Appointment Date: 2016-08-16 10:30 Physician: Dr. Tobi		
Patient: Lucienne Joey Appointment Date: 2016-09-03 6:30 Physician: Dr. Marx		
Patient: Kia Marlys Appointment Date: 2016-10-03 7:45 Physician: Dr. Leontine		
Home	Register Patient	Clinic Info

Search Bar:

While staying on the admin site Home page, we can utilize the search bar without leaving the page. When the search bar is clicked, the page is cleared and as names are typed into the search bar, it is populated with matching results. These results are populated in real time, so there is no need to confirm your search parameter. At the upper-rightmost corner of the page, there is an “X” and a “Cancel.” The “X” will clear the search bar and the search results. The “Cancel” button, will cancel the search and reverts the view of the page back to the default and will again display all the upcoming appointments.



New User Registration:

Each clinic will have new patients on a regular basis. CliniConnect has a built in feature to accommodate for these patients in the “Patient Registration” tab initially available at the bottom of the main page. Each of the other pages of the admin site are still available, the Home page (in yellow) and the Clinic Info page (in purple), on the bottom of the page as they were before on the Home page. The information needed from the patient is pretty self explanatory here, first name, last name, birth date, username (must be patient’s email), password (can be provided by admin if desired). Physician should be selected from the dropdown menu provided (at the location of the orange box). Then just press the big blue button to register the patient. The patient will be notified by e-mail, this e-mail was in the username field upon clicking the “Register Patient” button.

Register Patient	
First Name	First Name
Last Name	Last Name
Date of Birth	01/01/1901
Physician	<input type="text"/>
Username	user@clinic.com
Password	Password
Password	Confirm Password
<input type="button" value="Register Patient"/>	

+ Register Patient i

Home Register Patient Clinic Info

Editing Clinic Information:

The Clinic Information page is the other available page immediately from the Home page. The Clinic Info page is quite simple. It allows the clinic to update any information about the clinic at any time. The Home page (in green) and patient registration page (in yellow) are still available at the bottom of the page. If anything changes about the clinic, the information can be updated on this page. The fields are name, address, phone, e-mail, and hours of operation. All information must be updated at the same time, so if only the hours change, the rest of the options must also be filled out. While this may seem excessive at the moment, updating the information here, will update information that the patients are able to see on their devices.

Clinic Information

Current Information

Name: Kansas City Clinic
Address: 123 Happy Street
Phone: 123-123-1234
Email: smiley@smiley.com
Hours: M-F 8am-5pm

Clinic Name	Kansas City Clinic
Clinic Address	123 Happy Street
Clinic Phone	123-123-1234
Clinic Email	smiley@smiley.com
Clinic Hours	M-F 8am-5pm

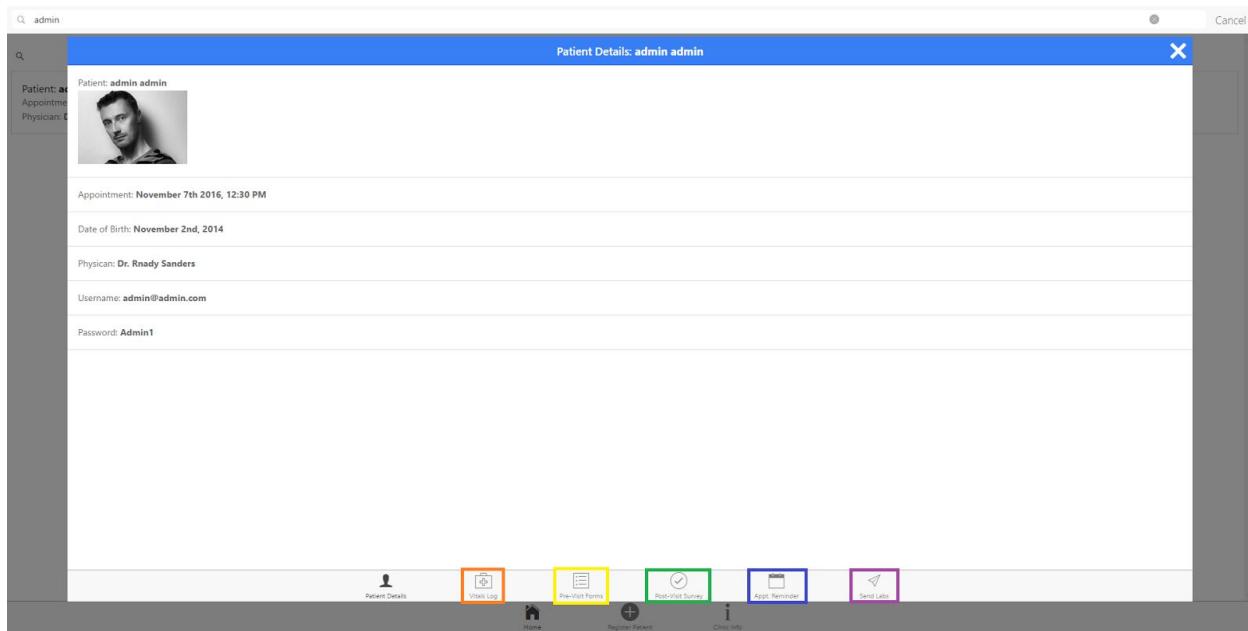
Update Clinic Information

 Home  Register Patient  Clinic Info

Patient Information:

Details:

This is the information model that arises after clicking on a name from the appointment list from the Home page. This details page is a simple page that shows the information of the patient that was filled out upon registration. This page links to other useful pages within the model. These pages are vitals log (in orange), Pre-Visit Forms (in yellow), Post-Visit Survey (in green), appointment reminders (in blue) and sending lab forms (in purple). To exit just click on the “X” in the top right, or any shaded area.



Vitals Log:

This is the Clinic's view of the daily vital log that each particular patient provides in the patient application. The admin site does not add new information to this page but can view it to assess the conditions of the patient. This page contains links to details (in red), Pre-Visit Forms (in yellow), Post-Visit Survey (in green), appointment reminders (in blue) and sending lab forms (in purple). By clicking on the print icon on the upper left (in orange), the Clinic can print to any connected printer.

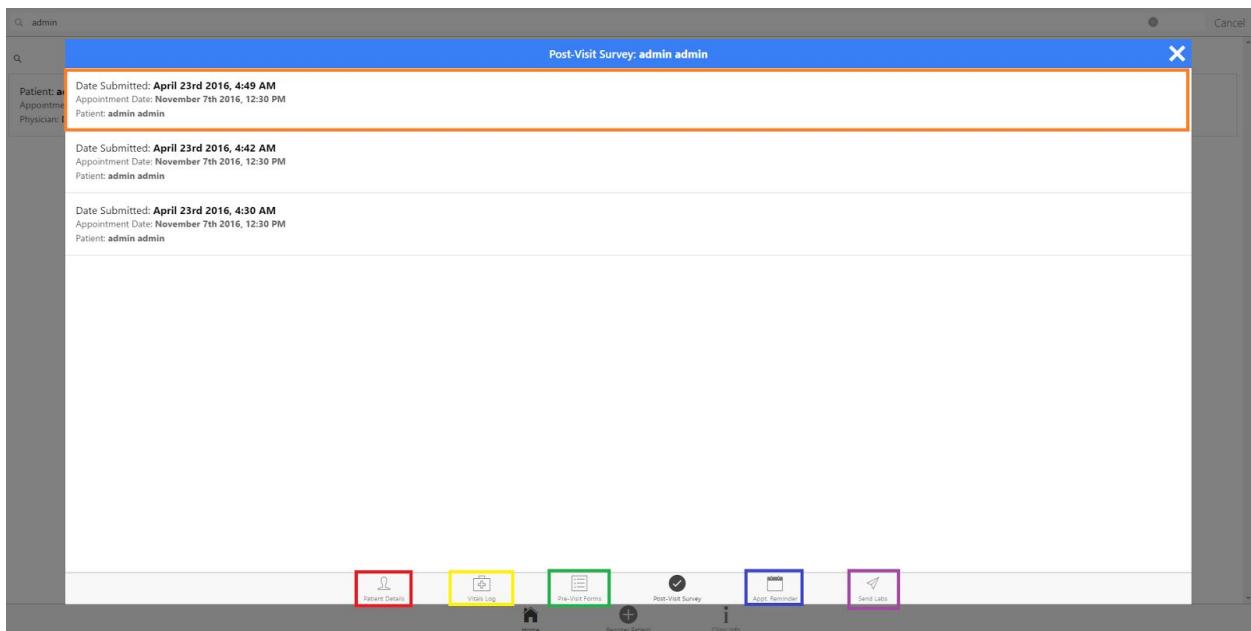
The screenshot shows a computer window titled "Vitals Log: admin admin". At the top left, there is a search bar with placeholder text "Patient: ac" and a date range selector. On the far right are "Cancel" and "Print" buttons. The main content area displays a table of vital data:

Date	Blood Sugar	Blood Pressure	Status	Temperature	Well Being
April 24th 2016, 12:20 AM	90	120/80	SAFE	98 °F	I feel super freaking awesome!
April 24th 2016, 12:09 AM	12	12/12	DANGER	12 °F	good
April 24th 2016, 12:00 AM	90	120/80	SAFE	98 °F	Good

Below the table, there is a row of functional icons with labels: "Patient Details" (red box), "Vitals Log" (grey), "Pre-Visit Forms" (yellow box), "Post-Visit Survey" (green box), "Appt. Reminder" (blue box), and "Send Lab" (purple box). The "Pre-Visit Forms" icon is highlighted with a yellow box.

Post-Visit Survey:

The Post-Visit Survey page is comprised of a list of surveys that the patient submits from their application. Each list item(the first of which is surrounded in orange) links to that particular survey. This page contains links to details (in red), Vitals Log (in yellow), Pre-Visit Forms (in green), appointment reminders (in blue) and sending lab forms (in purple). By Clicking the “X” or anywhere in the shaded area, you can close this sub window.



After clicking on a specific survey, the Clinic can view all the responses filled in by the patient. By clicking on the print icon on the upper left (in orange), the Clinic can print to any connected printer. By Clicking the “X” (in Orange) or anywhere in the shaded area, you can close this sub window.

Post-Visit Survey: admin admin

Date Submitted: November 7th 2016, 12:30 PM

Patient: admin admin

Appointment Date: November 7th 2016, 12:30 PM

1. How likely is it that you would recommend your Doctor to a friend or family member?
10 - Extremely Likely

2. Overall, how satisfied or dissatisfied were you with your last visit to our office?
Very satisfied

3. How easy or difficult was it to schedule your appointment at a time that was convenient for you?
Very easy

4. How convenient was the appointment time you were able to get?
Extremely convenient

5. In your opinion, how convenient is the location of our office?
Extremely convenient

6. Overall, how would you rate the service you received from the staff at our office?
Excellent

7. Overall, how would you rate the care you received from your Doctor?
Excellent

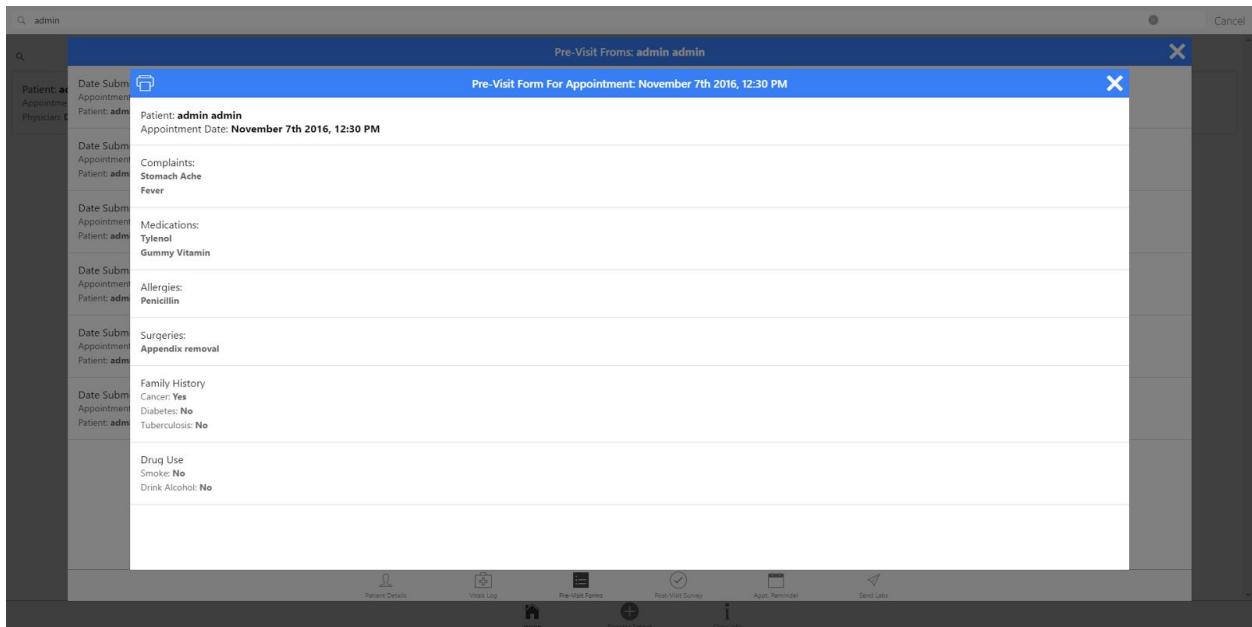
8. How well did your Doctor explain your treatment option?
Extremely well

9. Is there anything we could have done to improve your last visit?
Excellent!

Patient Details Visit Log Pre-Visit Forms Post-Visit Survey App. Reminder Send Lab

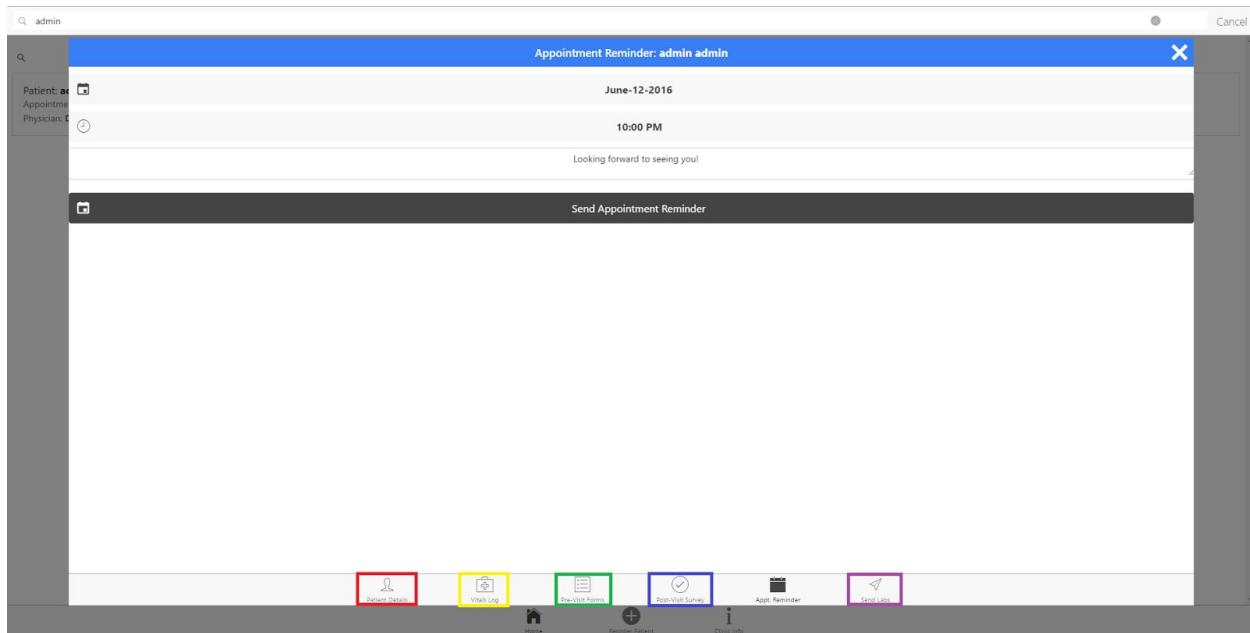
Pre-Visit Forms:

The Pre-Visit Forms page is very similar to the Post-Visit Survey. The Menu on the bottom is that same and links to Patient Details, Vitals Log, Post-Visit Survey, Appointment Reminders and Send Labs. In the Upper-Left corner there is a print option, much like in the Post-Visit Survey. By Clicking the “X” in the upper right corner or anywhere in the shaded area, you can close the sub window.



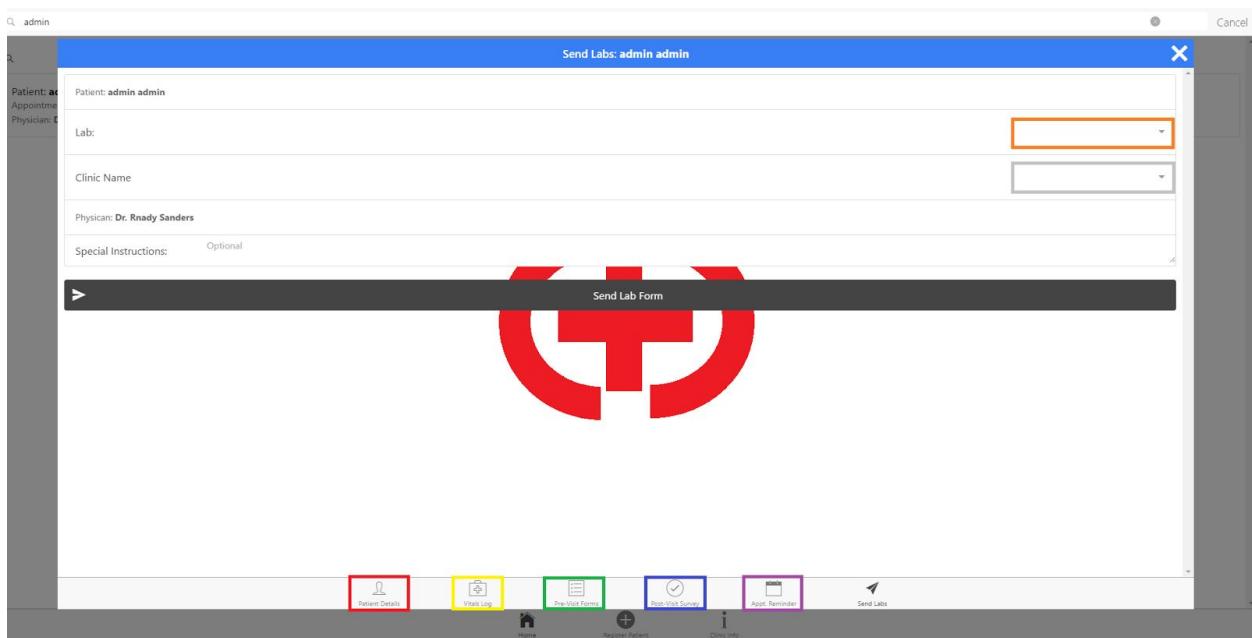
Appointment Reminders:

The Appointment Reminder page is available for the Clinic to send appointment reminder push notifications to the patient's device. The first option is a datepicker, which should be the date of the patient's next appointment. The second option is a time menu, which should be the time of the patient's next appointment. The third option is a message that can be used to ask the patient to remember to do something. Then the message can be sent by clicking the "Send Appointment Reminder" button. This page contains links to details (in red), Vitals Log (in yellow), Pre-Visit Forms (in green), appointment reminders (in blue) and sending lab forms (in purple). By Clicking the "X" or anywhere in the shaded area, you can close this sub window.



Send Lab Paperwork:

The Send Labs page requires information about the lab work that needs to be done for the patient. The page has several fields that need to be filled in. The patient is automatically populated. The Lab portion has a drop down menu (in orange) which is a list of several different lab work types that can be requested from the clinic. The clinic name is another drop down menu (in grey) that lists the available clinics where the lab work can be completed. There is also an option to send a message with the work to be done. Then, by clicking the “Send Lab Form” button, this information is translated into a QR code and is made available on the patient’s device. This page contains links to details (in red), Vitals Log (in yellow), Pre-Visit Forms (in green), Post-Visit Survey (in blue) and appointment reminders (in purple). Again, to exit, click on the “X” in the upper-right, or in the shaded area.

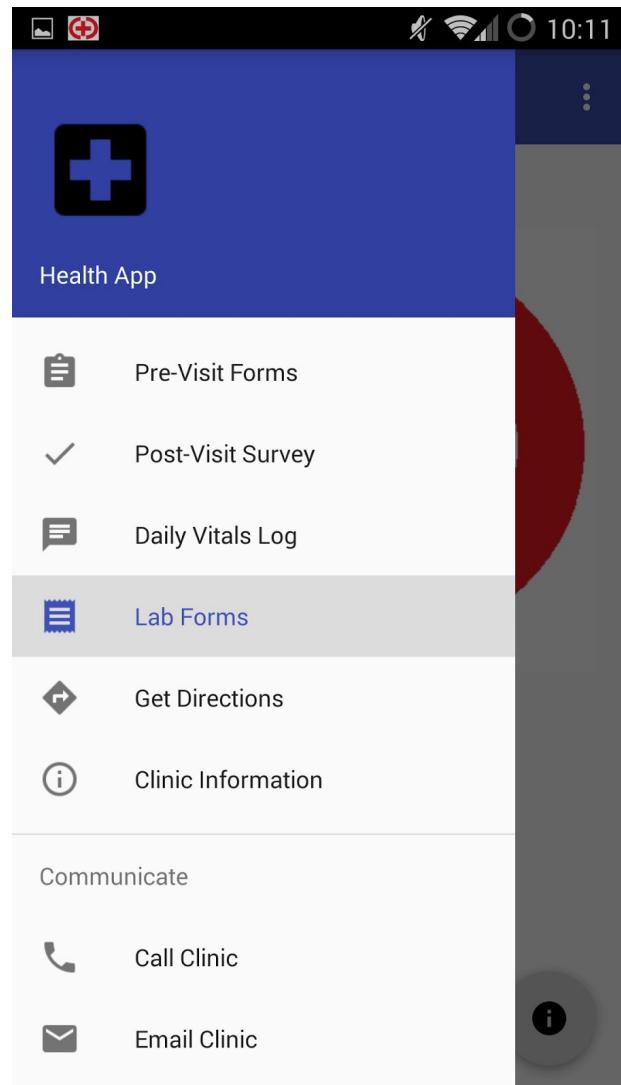


Patient Application

The patient application side of CliniConnect is implemented as a mobile application upon opening the application there is a basic login screen that requires an email and password that has been loaded into the administration application, before the patient can have any access to the mobile application. Also, the patient application is currently only available on Android platforms.

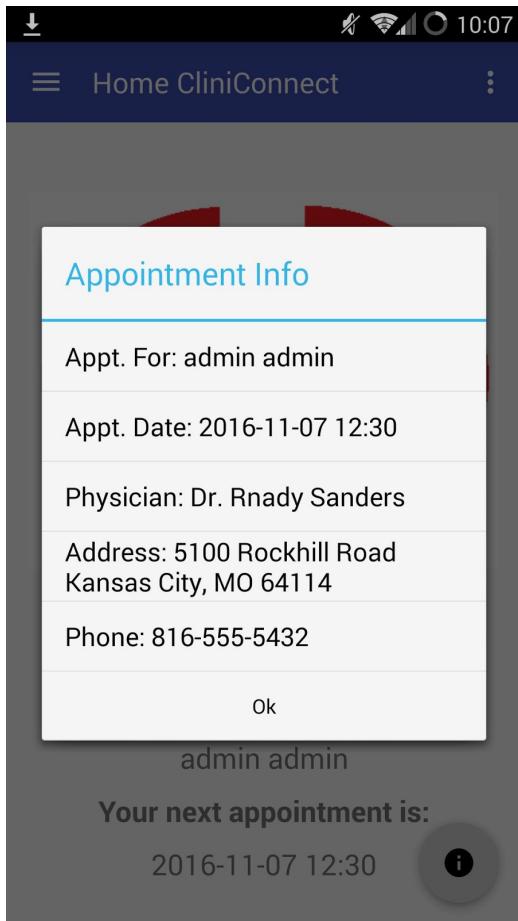
Navigation Bar:

The Navigation Bar is the hub for all interactions in the patient application. The navigation bar is a drawer style menu that is available from the upper left corner of the home page in the patient application. Accessible items from the menu are visible in the image to the right.



Home Page:

The home page is the basic page that is loaded upon login. This page shows the patient name and the patient's next appointment date. Tapping the icon on the bottom right will open up a subscreen that shows more information about the patient's upcoming appointment. After using the navigation bar to leave the home screen, press the back button to return to the home screen.



Pre-Visit Forms:

The pre visit forms page is populated with all the questions that the Clinic will want you to answer before your visit. Caution: Filling out the last question on this form will submit the form to the Clinic, so be sure all other questions are filled out before the last one, and do not make a mistake as it will submit upon selection.

TODAY'S COMPLAINT

First _____

Second _____

Third _____

MEDICATION ALLERGIES

Medication 1 Select _____

Medication 2 Select _____

Medication 3 Select _____

Food Allergies _____

MEDICATIONS I TAKE

First _____

Post-Visit Survey:

The post visit survey page is populated with all the questions that the Clinic will want you to answer after your visit. Caution: Filling out the last question on this form will submit the form to the Clinic, so be sure all other questions are filled out before the last one, and do not make a mistake as it will submit upon selection.

1. HOW LIKELY IS IT THAT YOU WOULD RECOMMEND YOUR DOCTOR TO A FRIEND OR FAMILY MEMBER?

Please select an option Select

2. OVERALL, HOW SATISFIED OR DISSATISFIED WERE YOU WITH YOUR LAST VISIT TO OUR OFFICE?

Please select an option Select

3. HOW EASY OR DIFFICULT WAS IT TO SCHEDULE YOUR APPOINTMENT AT A TIME THAT WAS CONVENIENT FOR YOU?

Please select an option Select

4. HOW CONVENIENT WAS THE APPOINTMENT TIME YOU WERE ABLE TO GET?

Please select an option Select

5. IN YOUR OPINION, HOW CONVENIENT IS THE LOCATION OF OUR OFFICE?

Please select an option Select

6. OVERALL, HOW WOULD YOU RATE THE SERVICE YOU RECEIVED FROM THE STAFF AT OUR OFFICE?

Please select an option Select

7. OVERALL, HOW WOULD YOU RATE THE CARE YOU RECEIVED FROM YOUR DOCTOR?

Please select an option Select

8. HOW WELL DID YOUR DOCTOR EXPLAIN YOUR TREATMENT OPTION?

Please select an option Select

9. IS THERE ANYTHING WE COULD HAVE DONE TO IMPROVE YOUR LAST VISIT?

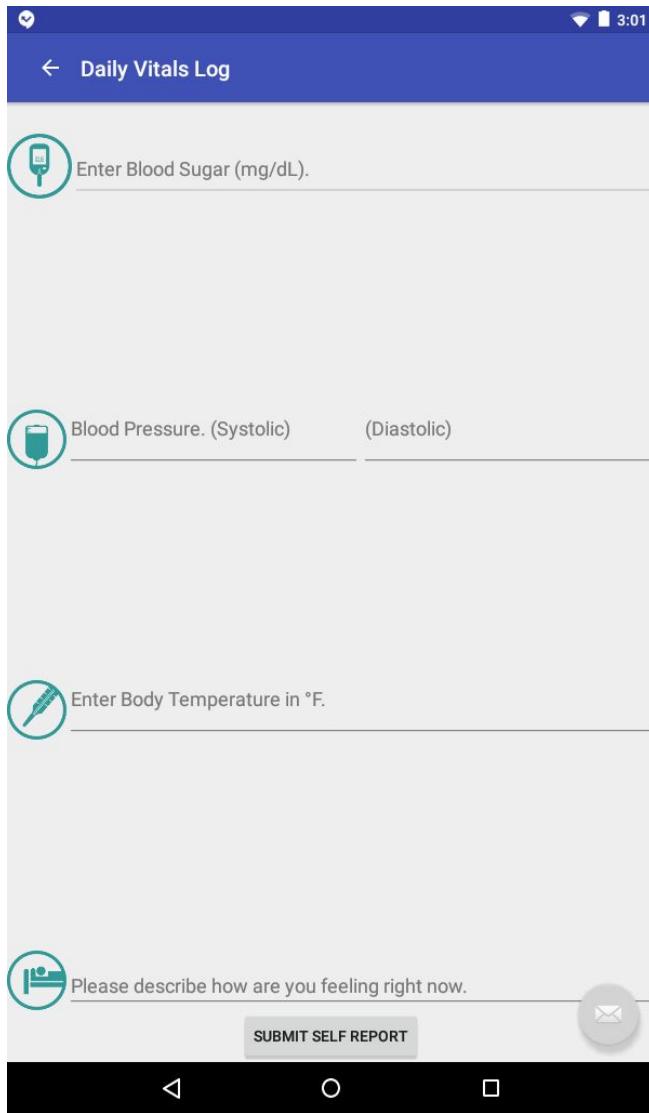
Please Enter Here

10. HOW WELL DID YOUR DOCTOR EXPLAIN YOUR FOLLOW-UP CARE?

Please select an option Select

Vitals Log:

The Vitals Log page is available for all users to update information on a daily, hourly, weekly, or “however often it needs to be” basis. These options are self explanatory and will be available for the admin to see as soon as the log is submitted. Which can be submitted simply by clicking the button on the bottom.



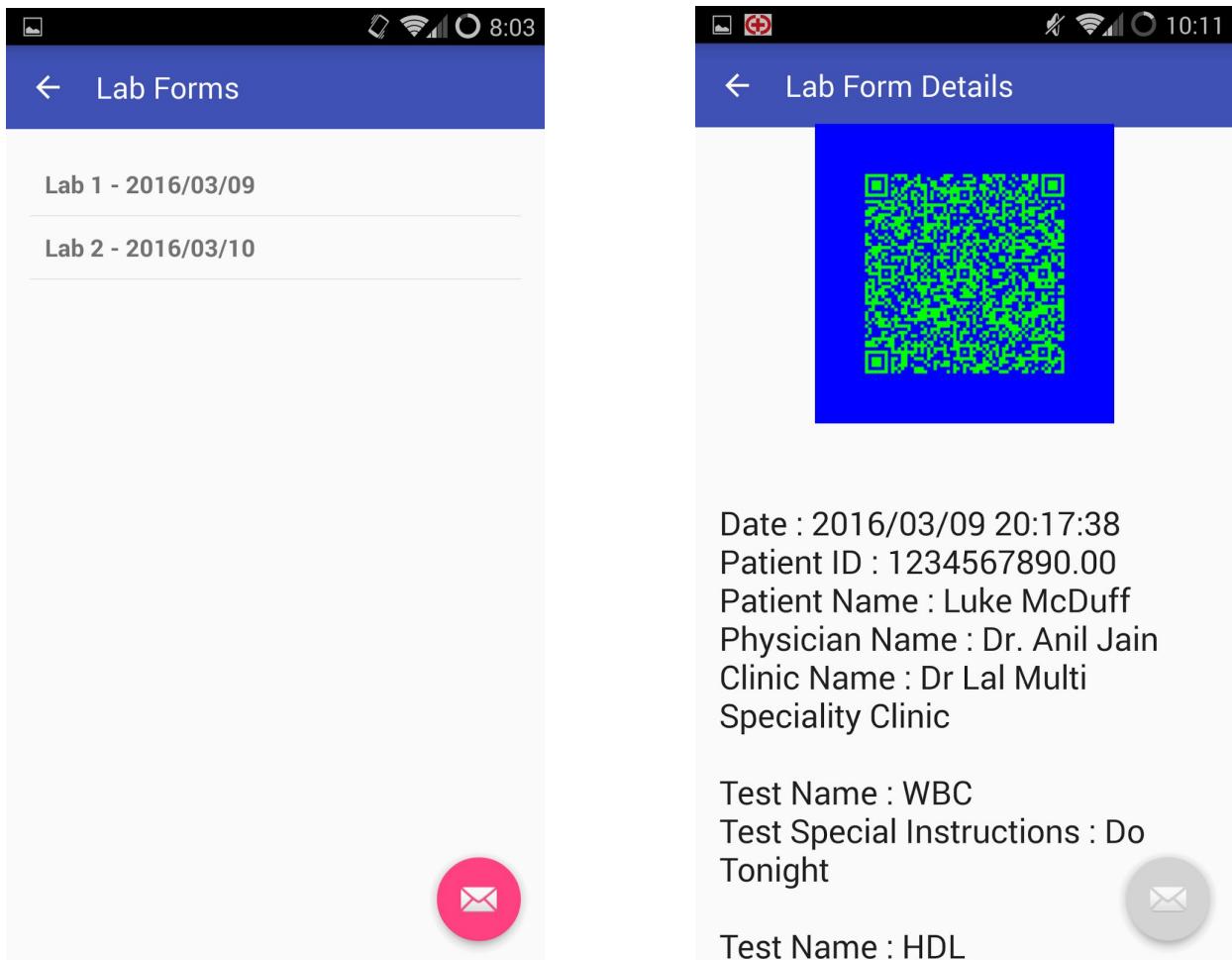
Clinic Info:

The clinic info page shows the patient's current clinic information that is provided by the clinic from their editable clinic info page from the administration application.



Lab Forms:

The Lab Forms pages has a main page with a list of all the labs that have been requested by the administration application and each item in that list links to a page with a QR code that is unique to each particular lab request. The pages are shown below, the list page on the left and the QR code page is on the right.



Directions, Email and Call Clinic:

All three of these functions are available from the navigation bar. The directions link loads google maps and populates the current location of the device as the starting point and the address from the clinic's info as the destination. The email link, opens the default email application and populated the recipient's address with the clinic's email from the clinic info. The call clinic link brings up the default phone dialer and populates the number, all that's left is to press the call button on the dialer.

Errors, Bugs and Deficiencies

There is only one error that a user may encounter while using our patient application, that error is, the fact that some of the back buttons on the top left corner of the pages using forms may not function. When this case occurs, just use the back button on the device and you will be directed back to the home page. The administration application should encounter no natural errors at this point.

The patient application, at this point, should not encounter any bugs. However the administration application does have one severe bug when restarting the patient database server. This bug can cause major issues, though the workaround to this is to delete the extra blank patient from the database. After deleting the blank patient the system runs just fine.

There are a few minor deficiencies in this project. In the patient application, there could be some UI modifications and improvements to clean up the interface. In both applications there could be better security in http requests, as there is no current encryption being used which would be a problem when dealing with sensitive medical data. However, encryption is beyond the current scope of this project. Lastly, In the administration application there is no way for uploading photos when also registering a new patient. Also, when looking at a patient a random photo from a small photo database is inserted. This could be remedied in the future by including a photo uploading option.

Index

Administration Application, 4-14

Appointments,

Clinic,

Reminders, 14

Schedule, 4

Patient, 16

Clinic,

Contact and Directions, 22

Editable Information, 7

Information, 19

Errors and Bugs, 23

Lab,

QR Code, 20

Sending, 14

Navigation,

Clinic, 4, 6-11, 13-14

Patient, 15

Patient,

Application, 15-21

Registration, 6

Search, 5

Index (Continued)

Post-Visit Survey

Clinic, 10

Patient, 18

Pre-Visit Forms

Clinic, 12

Patient, 17

Vitals Log

Clinic, 9

Patient, 18

Project Management

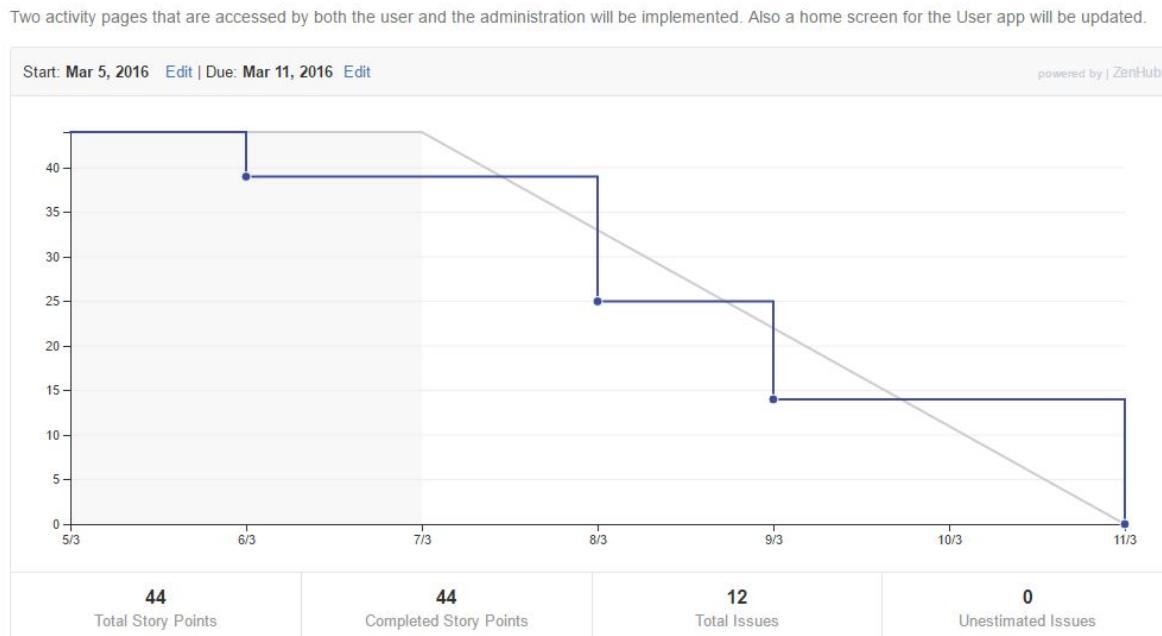
Report:

This project was managed close to the Agile style that was discussed in class. To do this, we used Zenhub to track stories, adjust current stories and add new stories as we gained feedback on our project after meeting with the TAs. As examples, we can look at the progress our group has made through these Zenhub burndown charts of Second and Fourth Increments. We tried to use the boards as often and efficiently as possible. We double checked each part of the project that was “Done” before allowing them to be “Closed”. In some Increments, this made the burndown charts less accurate to the true timeline of completion but we continued to use Zenhub as best we could. Much more time was spent outside of class, making sure that tasks were distributed evenly among teammates as to ensure the load of the project did not fall on anyone. The boards for Zenhub can be found at:

https://github.com/SCE-UMKC/ASESP16_CliniConnect_1#boards?repos=50469210

Second Increment and Fourth Increment Burndown charts are pictured below.

Second Increment



Fourth Increment

 Edit Milestone  Change Milestone ▾

The final increment will add one activity to Users and one activity to Administration. All aspects will be linked and undergo final testing as a cohesive final product.



Final Evaluation:

All in all, our group is very satisfied with the results of our project. Although we had intended to do an entirely different project in the beginning, we had a project too similar to another group. So we had to work on our back-up project idea, in which we found satisfaction in completing this back-up option. In terms of our initial vision for this project, we have accomplished everything we had planned. We were even able to merge few ideas together to improve the functionality of the app. An example of this is taking the idea of separate blood sugar and blood pressure level logs and combining them into the Vitals Log and adding a few extra components like temperature and a wellbeing question. The Agile process was a great tool in being able to complete this project. Initially, we did not quite understand what the point of the Agile process was, but we realized that the Agile process allowed us to make decisions

about the order of completing the components of our project. We were able to change what was worked on in each increment to better serve us for the upcoming deadlines. In the future, the Agile process could be expanded into the subcomponents of each component. Actually, we deviated from our initial schedule a bit. However, since the Agile process allows for this, we were able to switch out one component for another. Basically allowing us to adjust our schedule and work on components that were similar, simultaneously, where they had previously been scheduled in different increments. The management structure within our group was a variation of a 2-tiered system, where Benjamin handled meetings, paperwork, group morale and scheduling as the group leader, and Luke was the technical leader, where if any questions arose about connectivity between parts or about the technologies we were using that were new, he would answer them. This was very close to the planned structure, though the idea of having a technical lead was not initially planned for, it was certainly helpful to have someone within the group answer technical questions, while having another member lead meetings, plans, ideas and discussions. We had no problems getting anyone in the group to participate. Everyone did their best to contribute as much as possible both inside and outside meetings. Breaking down total contributions on this project into percentages, Benjamin contributed 24%, Shweta contributed 24%, Sri contributed 24% and Luke contributed 28%. This made the process of completing this project very smooth, every member was actively involved in the participation and completion of this project. The only way this could have been handled better, is if we knew the lab information before we decided what to do as our project is concerned, this could have flushed out more ideas at the beginning of the project, though in spite of this, we had several great ideas in our initial meetings. If this were a real project, we probably would have allowed for more time for thorough testing and debugging of our system and database, even though our final bug and error tally was extremely low. We also would address our deficiencies. In addition,

we would have kept better track of our contributions through Zenhub, but it was still a new technology we were all still getting used to, so contributions look overly skewed. Next year, allow students to get a better understanding of what they will learn over the semester in the first few days of class.

Project Proposal

Group Information:

- o Group Number: 1
- o Project Title: "CliniConnect"
- o Group Members
 - Benjamin Chrysler
 - Luke McDuff
 - Shweta Parihar
 - Sri Chaitanya Patluri

Project Goals and Objectives:

Overall goal: This application is to improve contact between patient (in particular, low income demographic) and their clinic which they attend.

Objectives:

We are planning to create a mobile application that will do the following:

- Patient can fill the form prior to clinic appointment, which in turn reduce the patient and doctor waiting time and will also improve the wait time in busy inner city clinics.
- Patient can fill the exit satisfaction form as per his comfort.

- Patient will be able to get his lab order for bloodwork etc.... electronically, which can be scanned.
- Patient will get remainder notification of his appointment.
- Patient who are required to monitor their blood pressure and blood sugar daily or weekly can keep track of their readings using this application.
- A map that will show from whatever location opened, the directions to the clinic, this would benefit patients who may require rides from friends or family members unfamiliar with the location of the clinic.
- Administrator will be able to administer the application using admin page.

Features:

- New patient can register in the registration page of the application.
- Already registered patient can login to the application using the login page.
- Prior clinic visit form page with the exact replica of the actual physical form.
- Exit satisfaction form page.
- Bloodwork details page with the facility to be scanned by pathology labs.
- Reminder icon on the status bar for the upcoming appointment.
- Blood pressure page for updating and keeping track of daily or weekly blood pressure readings.
- Blood sugar page for updating and keeping track for blood sugar readings.
- A map that will show from whatever location opened, the directions to the clinic, this would benefit patients who may require rides from friends or family members unfamiliar with the location of the clinic.
- Administrator can add new patient to the system, send blood work notification to the patient.

Significance: While some patient-to-clinic apps exist, they are typically patient portals which can contain an overwhelmingly large amount of data and features. By streamlining a few key important services we can help increase patient health.

Project Plan

Stories (Issues): This first increment lays the foundation of the user application side of our service. We have a User Login Page in which users can input their credentials to access the App. Also, we have a drawer style navigation bar that will connect to the other activities that we will implement in future increments. Both of these are on the user side of the software, that will be accessed through android devices. We have introduced push notifications in which both administrators and users participate. Users will receive a notification through the app on their device whenever the administrator sends a notification through the admin web site. Also, we have implemented a base page layout for all web pages. Additionally, the user application can receive QR codes that contain information about the needed lab work. Also, the database has been created to store user name, password and identification certification. We have incorporated an ionic framework to our administration side, making it more streamlined than before. Also, forms were added to have the Users fill out information about their visit and reason for visiting. The User Application now includes a blood log utility that allows users to input new blood logs that are stored on the server and can be retrieved on the administration side as well. Additionally, the information about the clinic is posted from the administration page to the database and is retrieved on the user app from the database. Forms were added to have the Users fill out a survey about their experience at the clinic. The Administration side can also view these surveys. The User Application's vitals log, previously called blood log utility, has been updated to take in new and more information. When the Admin registers a User, that User will get an mail to the User provided e-mail address. Additionally, both user and admin sides have undergone thorough testing and debugging, making sure that everything is running smoothly from end to end.

Service Design: The service design at this stage, is to create a base application that a User can log into after visiting the clinic in the first place. Users will only be able to login once they have visited the clinic. After, the service is meant to make interactions between the User and the clinic to be more immediate and more transparent. The Unit testing progressed by individually isolating the login from the navigation bar. Within the navigation bar, the widgets for the phone and the email connections were also tested by themselves before being connected with the navigation bar. Admin will be able to create new users and send notifications directly to the android application. This code displays the information users need for their lab tests, instead of having to carry along extra paperwork from place to place. Additionally, we included push notifications that send appointment reminders from the administrators. These notifications may also include short messages from the administrators. The service design includes an update to the ionic framework for the admin site. Admin will be able to update information about the clinic on the web site. Also, Users and Admin can now send and receive information in forms about the User's reason for visiting. The User is now able to update their blood logs regularly, which the administrators can check on. Users can fill out a brief survey about the experience they had at the clinic. Admin can view and print exit surveys to use for their internal improvement. The User receives an e-mail when the Admin registers a new patient. The User has more options to update in the new vitals log area, allowing the administration to have a better understanding of the User's conditions.

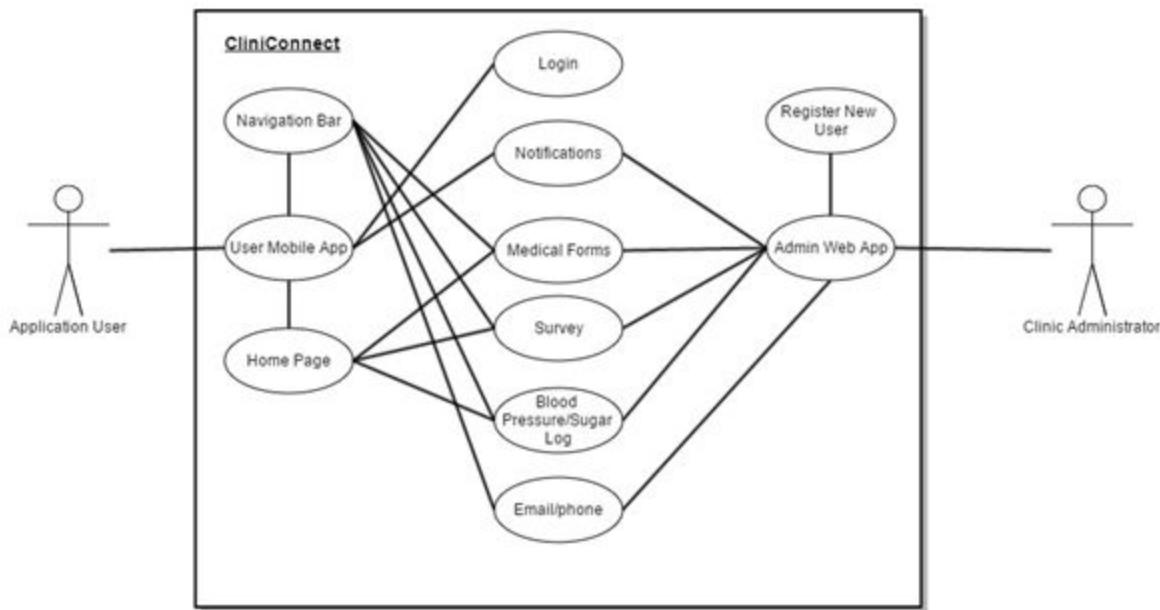
Service Implementation: The service at this point is only being implemented in xml and java. Since our service is currently concentrated on the User side, we will only be using xml and java for this. The service is currently tested on a variety of devices and device emulators. The service

now includes the implementation of the administration side which is comprised of html, css and javascript. The QR code creation and display on the user app side, the push notification sending and receiving, as well as the home page and shell of the user site have all been implemented. Both the administration page and user application sides have been implemented. Our database information has been implemented on mlab. The service implementation of the administration side has been updated to the ionic framework and is now more user friendly. Also, forms have been added to the administration side about the patient's visit. On the user application side, the blood log service has been implemented, which has also been updated on the admin site. A new version of the vitals log has been implemented in this increment. Also, exit survey forms have been added to the User application. Using MailJS, an e-mail service has been implemented in the administration side. Also, a connection to these surveys has been implemented on the administration side.

User Stories: When the User wants to initially access the application the user will enter their email and predetermined password to access the application. When the User wants to Navigate to any other page the User can use the navigation bar to redirect to another page. When the user wants to contact the clinic the User can use the navigation bar to access communication links, calling and emailing that will put the user in touch with the clinic. When the User wants to present their lab code they can access it through the web application through the navigation bar. When the User has an upcoming appointment, the Administration can send an appointment push notification to the User's app. When the Admin wants to Navigate to any other page the User can use the navigation bar to redirect to another page. When the User visits the Clinic, either the user can fill out a reason for visit form on the app. Or, for initial visits the admin site can fill out the form. When the User takes blood pressure or blood sugar, the user can input the

information into the blood log portion of the user app. When the Admin wants to navigate to any other page the User can use the navigation bar to redirect to another page. When the User, in addition to the existing blood log information there are new elements that the User can update and can input the information into the vitals log portion of the user app. When the User wants to leave a review of the experience that they had at the clinic they can fill out an exit survey which can be seen by the Administration page as soon as the information is submitted.

Use Case



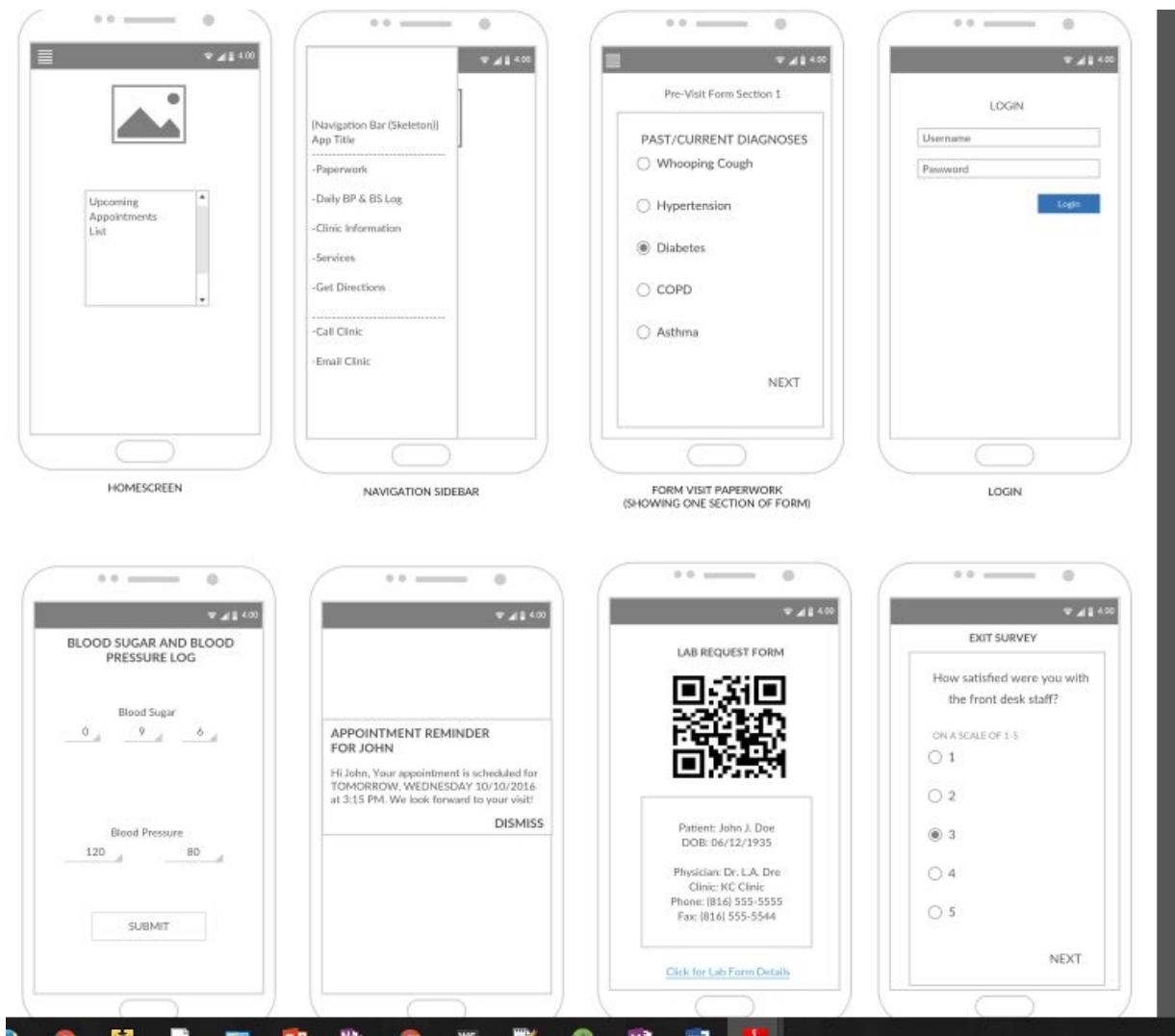
Service description: CliniConnect is an application that helps patients handle their paperwork in an organized and timely fashion. The application is intended for those who either do not have the time to come early to an appointment to fill out paperwork, have a tendency to miss/forget appointments and even those who need more constant interaction with their medical clinic. The application is intended for those who either do not have the time to come early to an

appointment to fill out paperwork, have a tendency to miss/forget appointments and even those who need more constant interaction with their medical clinic. The service includes an administration portal that serves as to connect the administration to the user. This service allows administrators to keep their patients properly up to date.

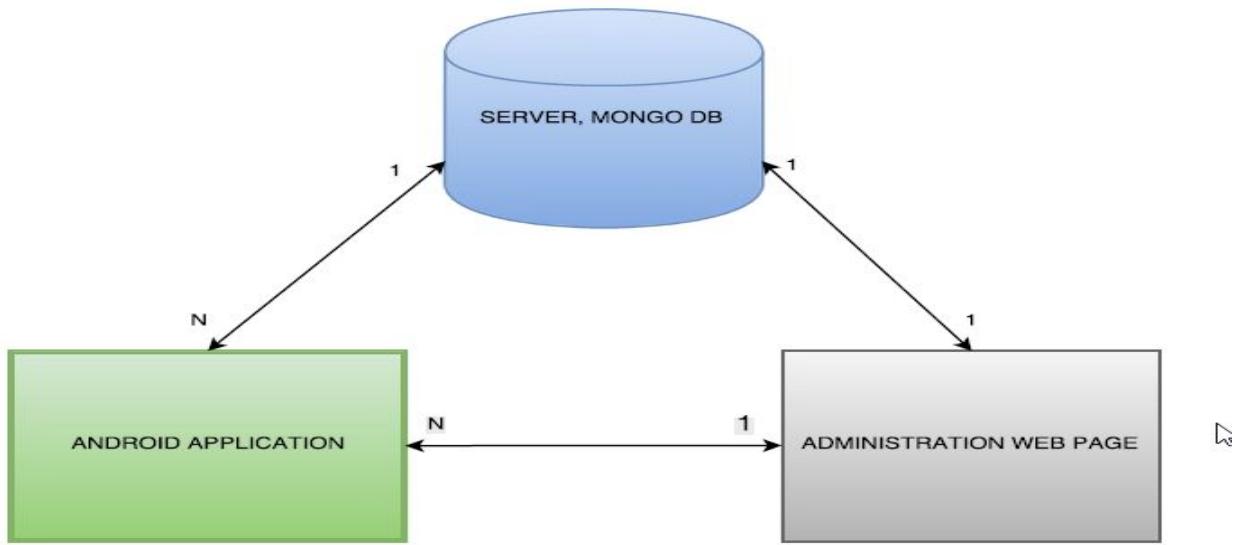
First Increment Report

In this first increment of “**CliniConnect**” we have designed the overall structure and flow of the application using wireframes and UML diagrams. In the initial face we have begun with implementing the login activity and navigation sidebar.

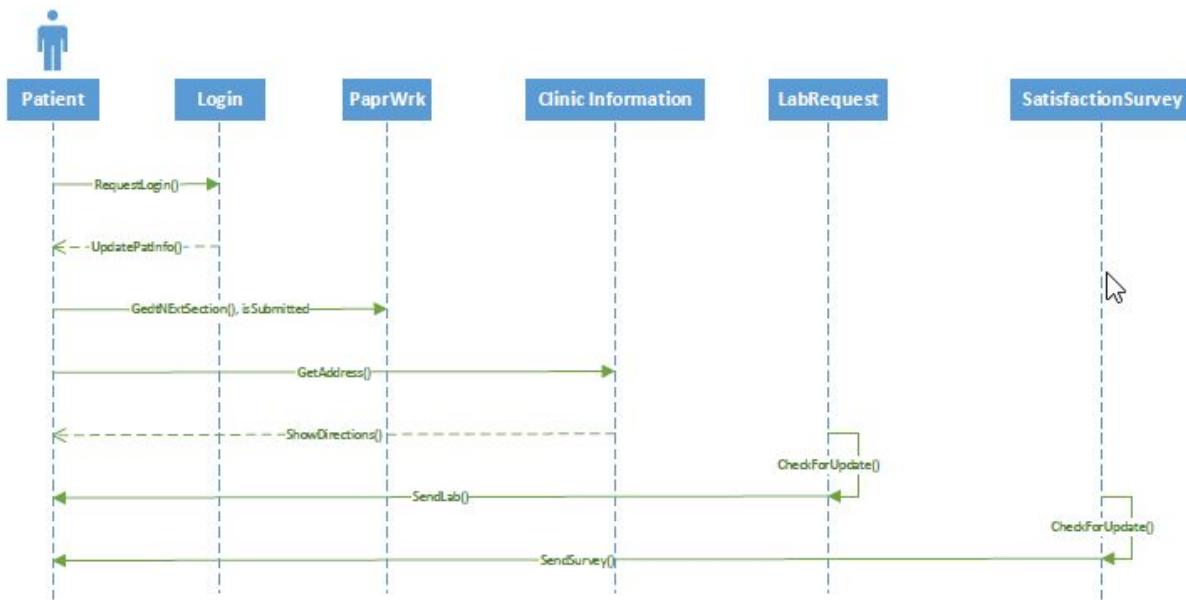
Detail Design: Wireframes



ARCHITECTURE STACK

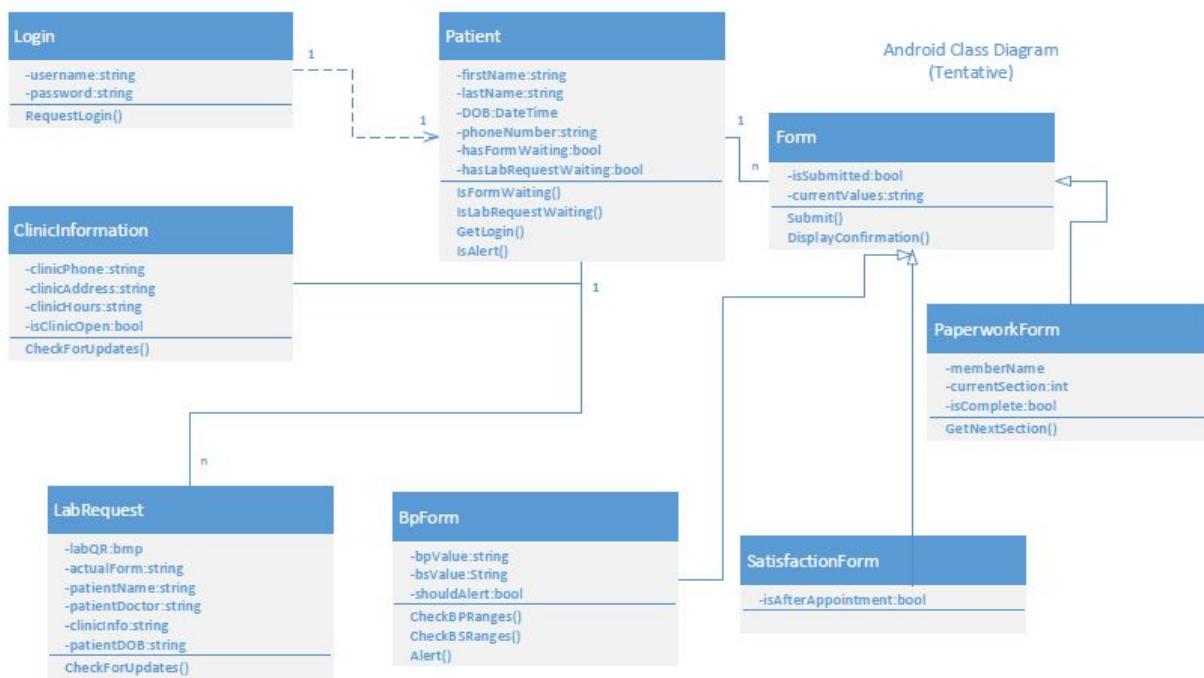


Sequence diagram



ANDROID SEQUENCE DIAGRAM (TENTATIVE)

Class diagram



Used existing Services/API

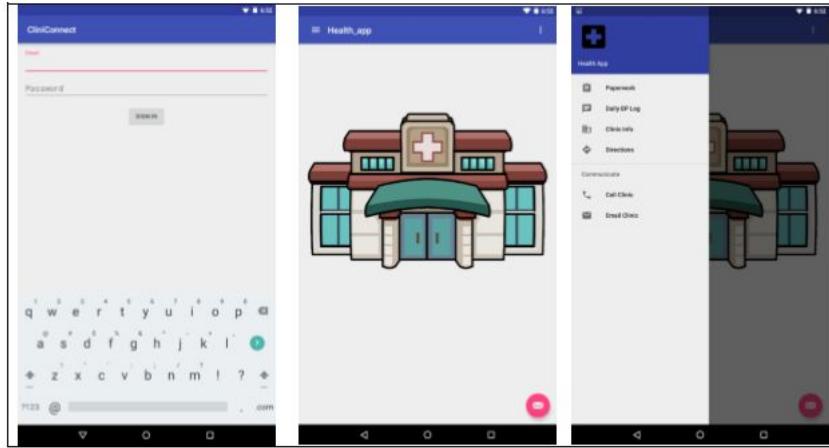
Google Maps API: Used to get directions from the user's current location to the clinic in Google Maps.

Widgets:

- **Phone Widget:** When selected, the user can call the clinic using existing phone services on the phone like Google Hangouts Dialer or the Phone Dialer.
- **Email Widget:** When Selected, the user can email the clinic using existing email applications on the phone like E-mail, GM

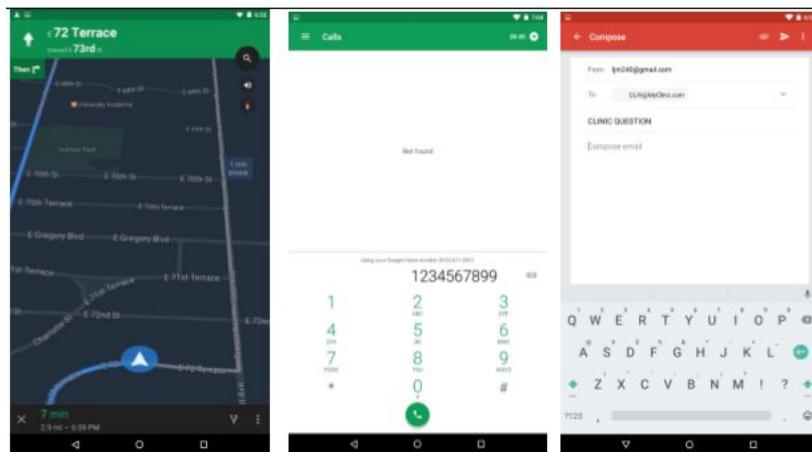
Implementation and Deployment:

We have implemented Login screen, home screen and Navigation drawer below are the screenshots.



When the user select “Direction” tab on navigation drawer, google map is opened providing direction from user’s current location to a hard coded address (clinic address)

When user select “Call Clinic” or “Email Clinic” phone/email client is opened with hard coded (Clinic phone number/email address)



Test Cases using Junit:

```

@RunWith(AndroidJUnit4.class)
@LargeTest
public class LoginActivityTest {
    private String email;
    private String badEmail;
    private String password;
    private String badPassword;
    @Rule
    public ActivityTestRule<LoginActivity> rule =
        new ActivityTestRule<LoginActivity>(LoginActivity.class);
    @Before
    public void initValidString(){
        email = "admin@admin.com";
        badEmail = "cat";
        password = "admin";
        badPassword = "a";
    }
    @Test
    public void testSuccessLogin() throws Exception{
        Intents.init();
        onView(withId(R.id.email))
            .perform(typeText(email));
        onView(withId(R.id.password))
            .perform(typeText(password), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(1));
        Intents.release();
    }
    @Test
    public void testFailedLoginBadPassword() throws Exception{
        Intents.init();
        onView(withId(R.id.email))
            .perform(typeText(email));
        onView(withId(R.id.password))
            .perform(typeText(badPassword), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(0));
        Intents.release();
    }
    @Test
    public void testFailedLoginBadEmail() throws Exception{
        Intents.init();
        onView(withId(R.id.email))
            .perform(typeText(badEmail));
        onView(withId(R.id.password))
            .perform(typeText(password), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(0));
        Intents.release();
    }
    @Test
    public void testFailedLoginBadEmailAndBadPassword() throws Exception{
        Intents.init();
    }
}

```

```

        onView(withId(R.id.email))
            .perform(typeText(badEmail));
        onView(withId(R.id.password))
            .perform(typeText(badPassword), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(0));
        Intents.release();
    }
}

```

Implementation status report

Work completed:

Login screen, home screen and Navigation drawer with direction service

Responsibility (Task, Person)

The timeline for this increment spanned for only two weeks, since our original plan was rejected based on similarity to another project. Every member of the group was involved in the brainstorming sessions that led to the format for the service, ClinConnect, we have mapped out. As we found the deadline fast approaching we divided up the tasks. On Implementation and testing, Sri and Luke put in several hours of coding. Luke also contributed the class UML design and sequence design. Shweta is on the task of the Report and connecting all of the pieces contributed by our team members. Ben is the team lead, and worked on team management, increment scheduling and Project Planning.

Second Increment Report

In this second increment of “**CliniConnect**” we have designed the overall structure and flow of the application and administration using wireframes and UML diagrams. In this phase we have implemented the shell of the administration portal. Included in this increment, are the ability to update the lab forms from the administration portal to the user app. Also, we have created the ability to have administrators send personalized notifications to the user's device. When clicking on the notification the User is sent back to the application.

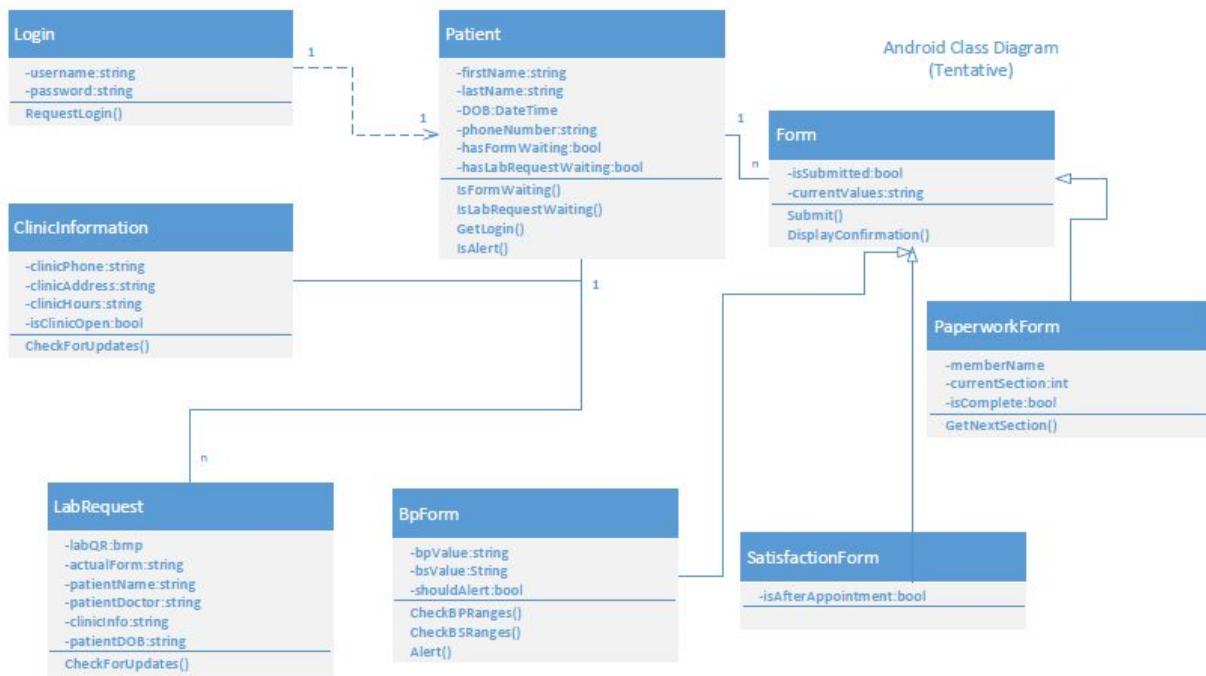
Detail Design: Wireframes

The image displays four wireframe screenshots of the CliniConnect application interface, arranged in a 2x2 grid:

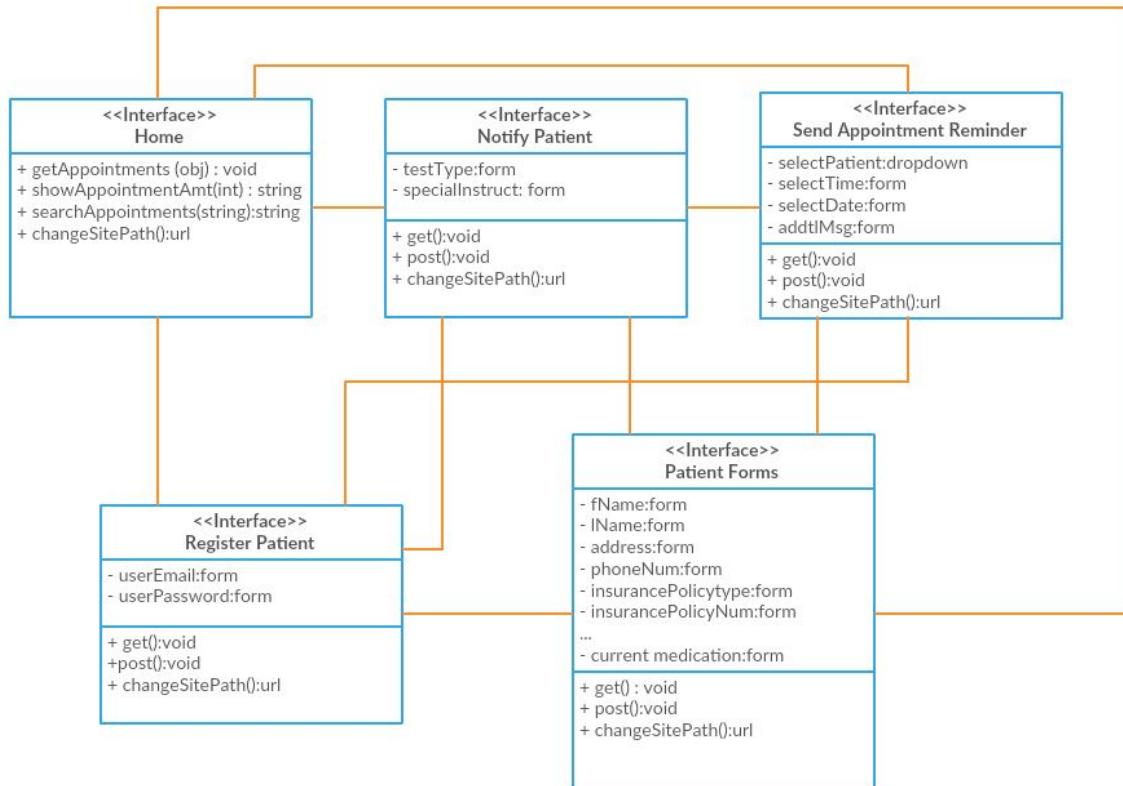
- Top Left:** Home screen showing a list of appointments. The table has columns for Appointments, Times, and Physicians. Data includes: Mark Davis (12:40, Dr. Klotz), William Thomas (12:50, Dr. Frankenfurter), Jeremy Wong (1:30, Dr. Bleh), Clyde Frosch (1:45, Dr. Klotz), Ingrid Hasslebeck (2:35, Dr. Bleh), Michael Wolfe (2:50, Dr. Frankenfurter), and Shenae Wallace (3:20, Dr. Bleh). A date input field shows 05/06/2015.
- Top Right:** Patient registration screen. It features fields for First Name, Last Name, Address, Phone number, User Email, and Password, followed by a "Register New Patient" button.
- Bottom Left:** Notify Patient screen. It includes a "Patient Lookup" button, a "Search" button, and a "Print Blood Log" button. Below these are buttons for "Remind", "Send Digital Paperwork", and "Send Lab Request". To the left is a sidebar with patient details: Patient Name, Patient Address, Patient Phone, Patient Insurance Policy.
- Bottom Right:** Another patient registration screen, similar to the top right, but with a vertical scroll bar on the right side.

Class diagram

Android



Web



Used existing Services/API

APIs:

- **Google Maps:** Used to get directions from the user's current location to the clinic in Google Maps.
- **ZXing (Zebra Crossing):** Used to convert lab request information into QRcode.

Widgets:

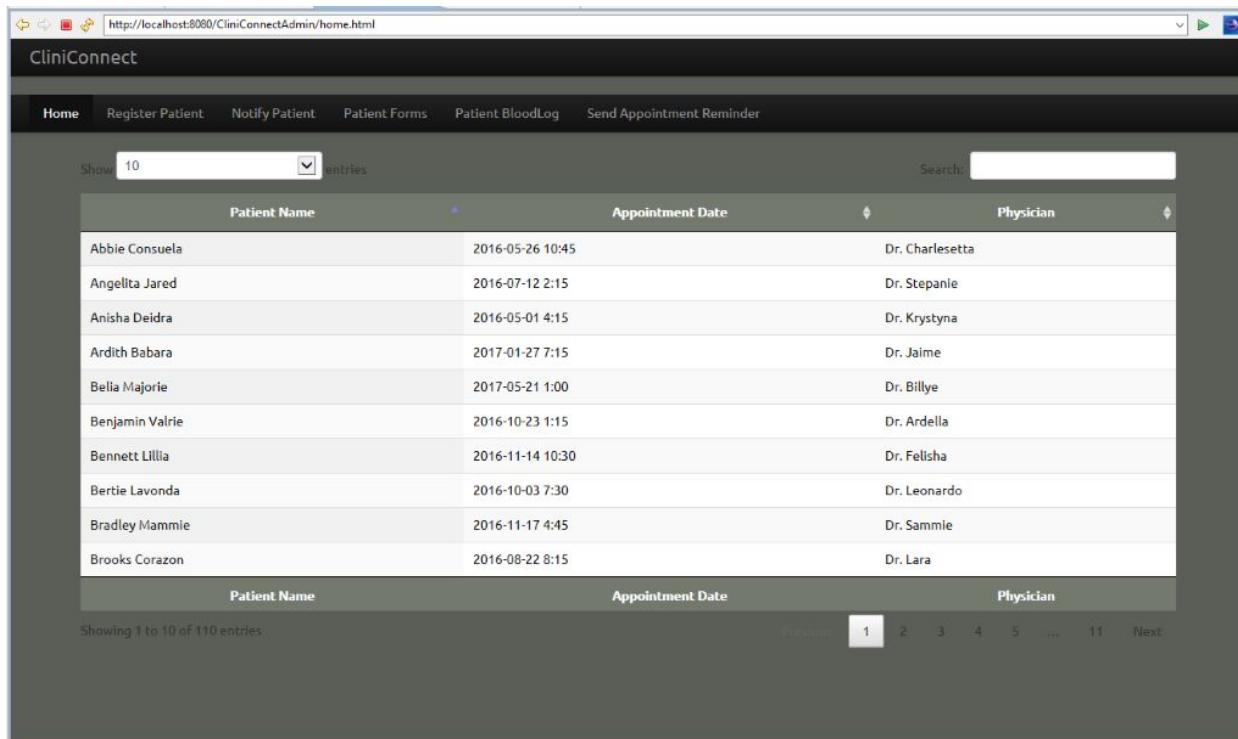
- **Phone Widget:** When selected, the user can call the clinic using existing phone services on the phone like Google Hangouts Dialer or the Phone Dialer.

- **Email Widget:** When Selected, the user can email the clinic using existing email applications on the phone like E-mail, GM
- **ImageView Widget:** Used to display the QR code after it had been processed by ZXing.
- **TextView Widget:** Used to display information about the information embedded in the QR code.

Implementation and Deployment:

We have implemented Home screen, Navigation bar, Notification sending and receiving, QRcode sending and receiving, Labs form pages and also created a generic logo for the application. Below are the screenshots.

Admin Web Page Home:



The screenshot shows a web browser window with the URL <http://localhost:8080/CliniConnectAdmin/home.html>. The page title is "CliniConnect". The navigation menu includes "Home", "Register Patient", "Notify Patient", "Patient Forms", "Patient BloodLog", and "Send Appointment Reminder". Below the menu is a search bar with "Show: 10 entries" and a "Search:" input field. A table displays patient appointment data with columns: "Patient Name", "Appointment Date", and "Physician". The table contains 10 entries. At the bottom, a footer indicates "Showing 1 to 10 of 110 entries" and a navigation bar with links "Previous", "1", "2", "3", "4", "5", "...", "11", and "Next".

Patient Name	Appointment Date	Physician
Abbie Consuela	2016-05-26 10:45	Dr. Charlesetta
Angelita Jared	2016-07-12 2:15	Dr. Stepanie
Anisha Deidra	2016-05-01 4:15	Dr. Krystyna
Ardith Babara	2017-01-27 7:15	Dr. Jaime
Belia Majorie	2017-05-21 1:00	Dr. Billye
Benjamin Valrie	2016-10-23 1:15	Dr. Ardella
Bennett Lillia	2016-11-14 10:30	Dr. Felisha
Bertie Lavonda	2016-10-03 7:30	Dr. Leonardo
Bradley Mammie	2016-11-17 4:45	Dr. Sammie
Brooks Corazon	2016-08-22 8:15	Dr. Lara

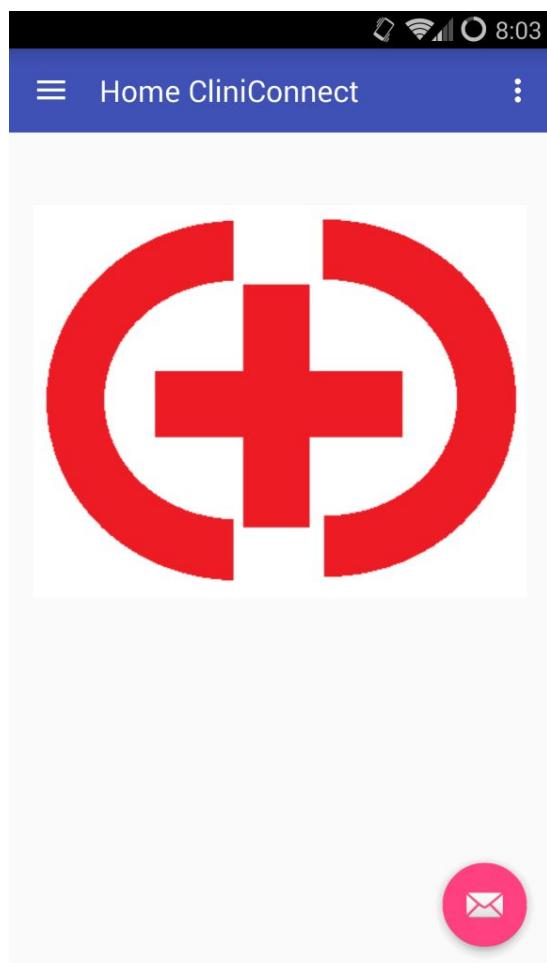
Admin Web Page QR requester

The screenshot shows a web browser window titled "CliniConnect" with the URL <http://localhost:8080/CliniConnectAdmin/patientnotifier.html>. The page has a dark grey header with the CliniConnect logo and a navigation bar containing links for Home, Register Patient, Notify Patient, Patient Forms, Patient BloodLog, and Send Appointment Reminder. The main content area features a large title "Rx Clinical Test for Patient". Below it, a message says "Please select a test from the dropdown." followed by a dropdown menu with "CBC" selected and a checked checkbox. A text input field labeled "Special Instructions(optional):" is empty. At the bottom is a "Rx Test" button.

Admin Web Page Appointment Sending

The screenshot shows a web browser window titled "CliniConnect" with the URL <http://localhost:8080/CliniConnectAdmin/pushNotification.html>. The page has a dark grey header with the CliniConnect logo and a navigation bar containing links for Home, Register Patient, Notify Patient, Patient Forms, Patient BloodLog, and Send Appointment Reminder. The main content area features a large title "Send Appointment Notification". Below it are four input fields: "Select Patient" (dropdown), "Select Time" (text input), "Select Date" (text input), and "Message (optional)" (text input). At the bottom is a "Send Appointment Reminder" button.

Mobile Home page with logo



Mobile Lab Pages (1st image with QR code)

The image displays two side-by-side screenshots of a mobile application interface, likely for managing laboratory forms.

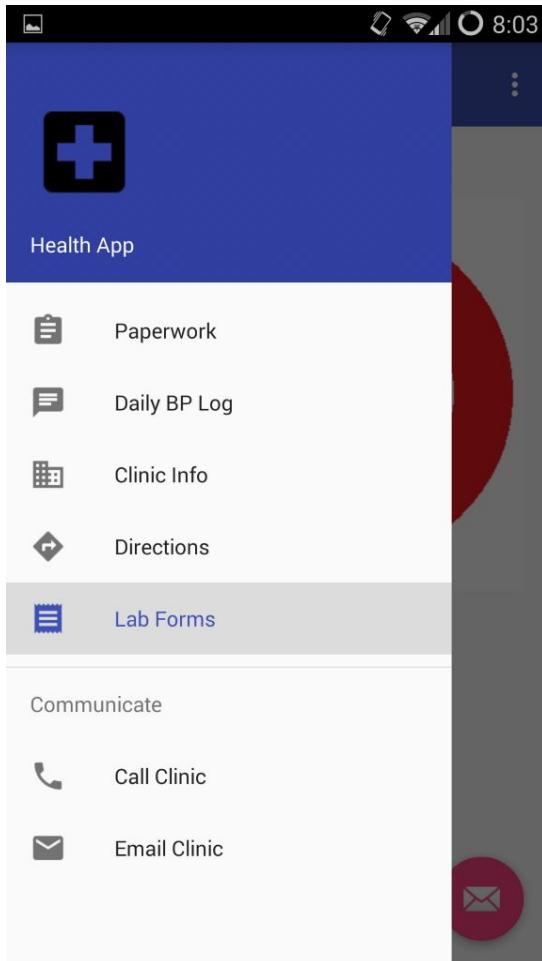
Screenshot 1: Lab Form Details

- At the top, a blue header bar shows the title "Lab Form Details" and a back arrow icon.
- The main content area features a large QR code centered on a blue background.
- Below the QR code, there is a section of patient and physician information:
 - Date : 2016/03/09 20:17:38
 - Patient ID : 1234567890.00
 - Patient Name : Shweta Parihar
 - Physician Name : Dr. Anil Jain
 - Clinic Name : Dr Lal Multi Speciality Clinic
- Below this information, there is a section for test details:
 - Test Name : WBC
 - Test Special Instructions : Do Tonight
- At the bottom left, the text "Test Name : WBC" is repeated.
- On the right side of the screen, there is a pink circular button with a white envelope icon, likely for sharing or emailing the form.

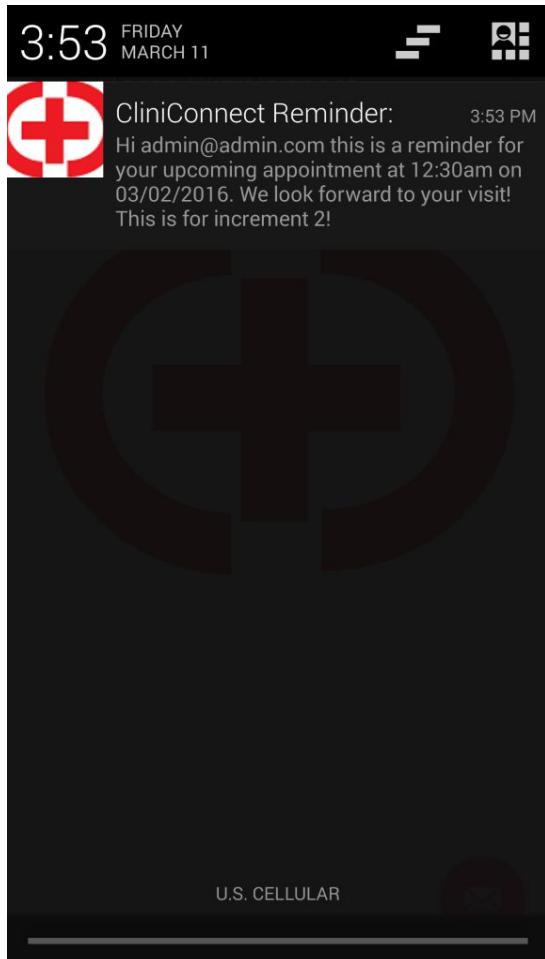
Screenshot 2: Lab Forms

- At the top, a blue header bar shows the title "Lab Forms" and a back arrow icon.
- The main content area lists two lab entries:
 - Lab 1 - 2016/03/09
 - Lab 2 - 2016/03/10
- On the right side of the screen, there is a pink circular button with a white envelope icon, likely for sharing or emailing the list of forms.

Mobile Application Navigation Bar Updated with Lab Forms



Mobile Application Push Notification



Unit Test Cases:

```
@RunWith(AndroidJUnit4.class)
@LargeTest
public class QRLabTest {

    @Rule
    public ActivityTestRule<ClinicalTest> rule =
        new ActivityTestRule<>(ClinicalTest.class);

    @Test
    public void DateOpensQRTTest() throws Exception{
        Intents.init();
        SystemClock.sleep(5000);
        onView(withId(R.id.clinical_test_date_list)).perform(click());

        rule.launchActivity(new Intent());
        SystemClock.sleep(5000);
        intended(hasComponent(ClinicalTest.class.getName()));
        intended(hasComponent(QRCodeAndDetailsActivity.class.getName()), times(1));
        Intents.release();
    }
}
```

```
@RunWith(AndroidJUnit4.class)
@LargeTest
public class LoginActivityTest {
    private String email;
    private String badEmail;
    private String password;
    private String badPassword;
    @Rule
    public ActivityTestRule<LoginActivity> rule =
        new ActivityTestRule<LoginActivity>(LoginActivity.class);
    @Before
    public void initValidString(){
        email = "admin@admin.com";
        badEmail = "cat";
        password = "admin";
        badPassword = "a";
    }
    @Test
    public void testSuccessLogin() throws Exception{
        Intents.init();
        onView(withId(R.id.email))
            .perform(typeText(email));
        onView(withId(R.id.password))
            .perform(typeText(password), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(1));
        Intents.release();
    }
    @Test
    public void testFailedLoginBadPassword() throws Exception{
        Intents.init();
        onView(withId(R.id.email))
            .perform(typeText(email));
        onView(withId(R.id.password))
            .perform(typeText(badPassword), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(0));
        Intents.release();
    }
    @Test
    public void testFailedLoginBadEmail() throws Exception{
        Intents.init();
        onView(withId(R.id.email))
            .perform(typeText(badEmail));
        onView(withId(R.id.password))
            .perform(typeText(password), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(0));
        Intents.release();
    }
    @Test
    public void testFailedLoginBadEmailAndBadPassword() throws Exception{
        Intents.init();
```

```
        onView(withId(R.id.email))
            .perform(typeText(badEmail));
        onView(withId(R.id.password))
            .perform(typeText(badPassword), closeSoftKeyboard());
        onView(withId(R.id.email_sign_in_button)).perform(click());
        rule.launchActivity(new Intent());
        intended(hasComponent(LoginActivity.class.getName()));
        intended(hasComponent(MainActivity.class.getName()), times(0));
        Intents.release();
    }
}
```

Implementation status report

Work completed:

The work completed in this increment was, web page design, QR code (both user and administration sides), Push notifications (both user and administration sides), home page for web, and database setup were completed.

Responsibility (Task, Person)

In this increment, work was divided into tasks to use time efficiently. The task of the QR codes went to Shweta, implementing the user end result as well as the administrator's back end request. Sri handled the Implementation of the home web pages and the general shared layout for each web. Ben, designed the web pages, created the logo for the application and wrote the report. Luke implemented the database setup, as well as implemented the push notifications on both administrator and user ends.

Third Increment Report

In this third increment of “**CliniConnect**” we have implemented versions that follow the overall structure and flow of the application and administration wireframes and UML diagrams. In this phase we have updated the implementation of the administration portal. Included in this increment, is the ability to fill out forms related to reasons for visiting. Also, we have implemented the blood log option in our user app. The blood log also comes with some basic text feedback based on the results of the blood.

Detail Design: Wireframes

Web Wireframes:

Appointments	Times	Physicians
Mark Davis	12:40	Dr. Klotz
William Thomas	12:50	Dr. Frankenfurter
Jeremy Wong	1:30	Dr. Bleh
Clyde Frosch	1:45	Dr. Klotz
Ingrid Hasslebeck	2:35	Dr. Bleh
Michael Wolfe	2:50	Dr. Frankenfurter
Shenae Wallace	3:20	Dr. Bleh

CliniConnect



Host Clinic Information

Home Register Patient Notify Patient Patient Forms

First Name

Last Name

Address

Phone Number

Insurance Policy

Insurance Policy Number

CliniConnect



Host Clinic Information

Home Register Patient Notify Patient Patient Forms

Patient Lookup Print Blood Log

Patient Name
Patient Address
Patient Phone
Patient Insurance
Policy

Next Appointment: @Time

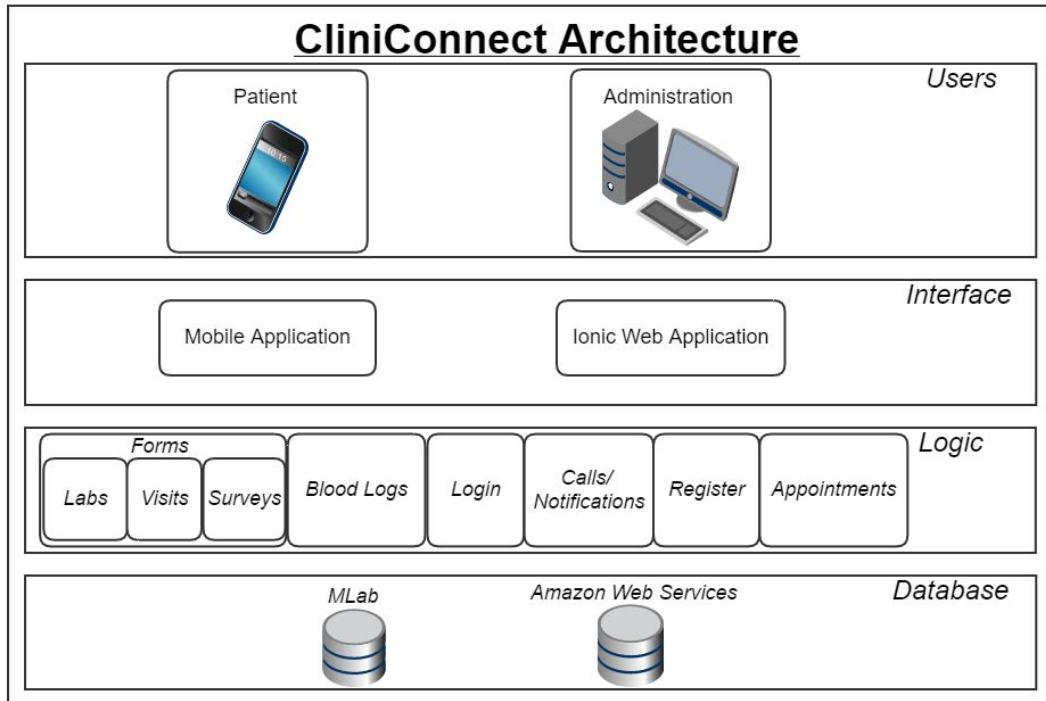
Paperwork up to date?
Yes/No

Lab work needed

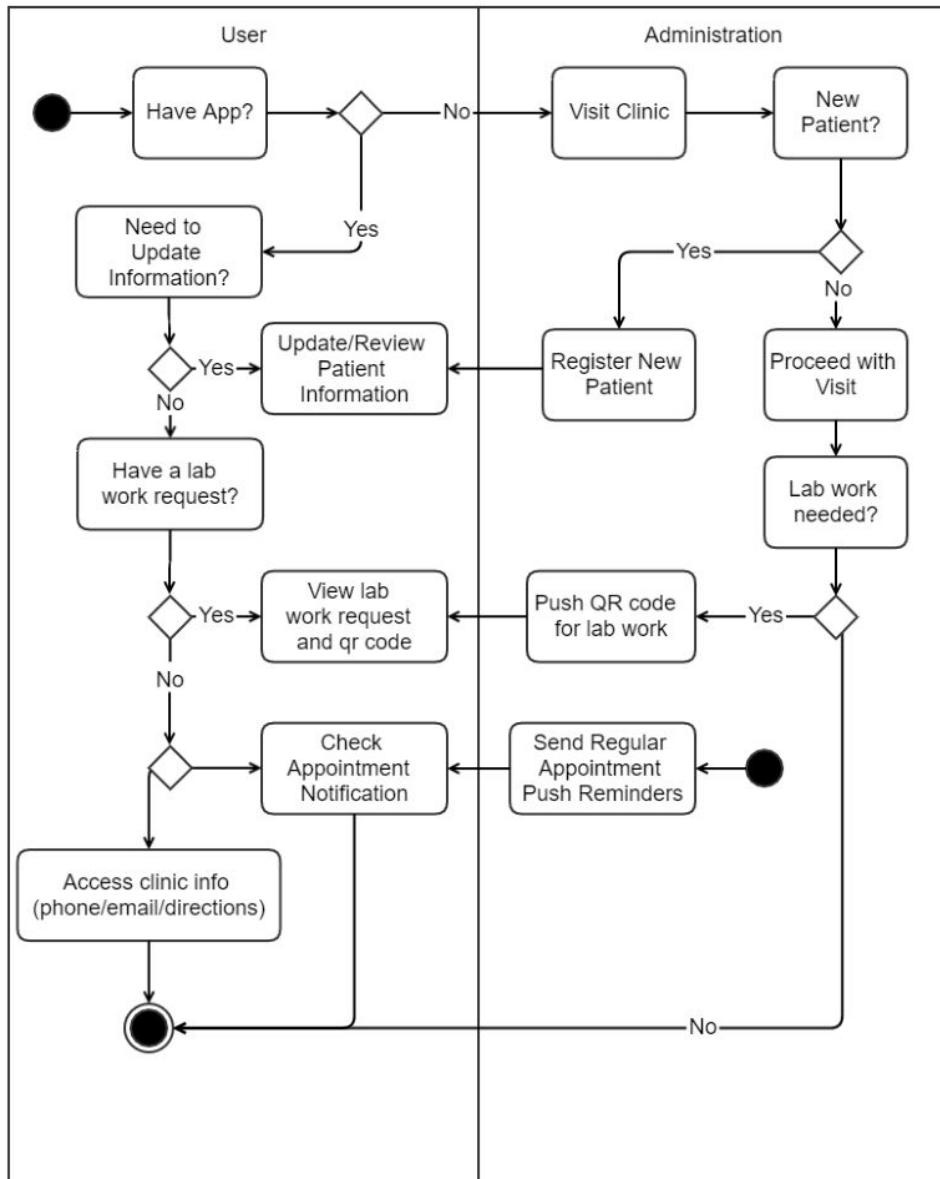
The screenshot shows a web browser window titled "CliniConnect". In the top right corner, there is a placeholder image of a landscape. To the right of the image, the text "Host Clinic Information" is displayed. Below the title bar, there is a navigation menu with four items: "Home", "Register Patient" (which is highlighted in grey), "Notify Patient", and "Patient Forms".

The main content area contains a form for registering a new patient. The form consists of six input fields stacked vertically: "First Name", "Last Name", "Address", "Phone number", "User Email", and "Password". Below the input fields is a large blue button labeled "Register New Patient".

Architecture Diagram:



Activity Diagram:



Used existing Services/API

APIs:

- **Google Maps:** Used to get directions from the user's current location to the clinic in Google Maps.
- **ZXing (Zebra Crossing):** Used to convert lab request information into QRcode.

Widgets:

- **Phone Widget:** When selected, the user can call the clinic using existing phone services on the phone like Google Hangouts Dialer or the Phone Dialer.
- **Email Widget:** When Selected, the user can email the clinic using existing email applications on the phone like E-mail, GM
- **ImageView Widget:** Used to display the QR code after it had been processed by ZXing.
- **TextView Widget:** Used to display information about the information embedded in the QR code.

Services:

- **Amazon Web Services – EC2:** Used to host admin website.

Implementation and Deployment:

We have implemented an ionic framework for the admin web site, the blood log activity for user and admin sides, and the visit forms for admin and users. The admin website has been implemented with Apache http server on Amazon Web Services. [A link to the admin site page.](#)
Below are the screenshots.

Web Page Ionic Conversion:

Appointment List:

Patient: **Myrtle Racheal**
Appointment Date: 2016-03-23 12:45
Physician: **Dr. Leontine**

Patient: **Renda Debbie**
Appointment Date: 2016-05-13 3:00
Physician: **Dr. Krysten**

Patient: **Clara Roselia**
Appointment Date: 2016-06-18 2:30
Physician: **Dr. Candance**

Patient: **Valentina Jani**
Appointment Date: 2016-07-07 1:45
Physician: **Dr. Raymon**

Patient: **Chu Gerry**
Appointment Date: 2016-08-16 10:30
Physician: **Dr. Tobi**

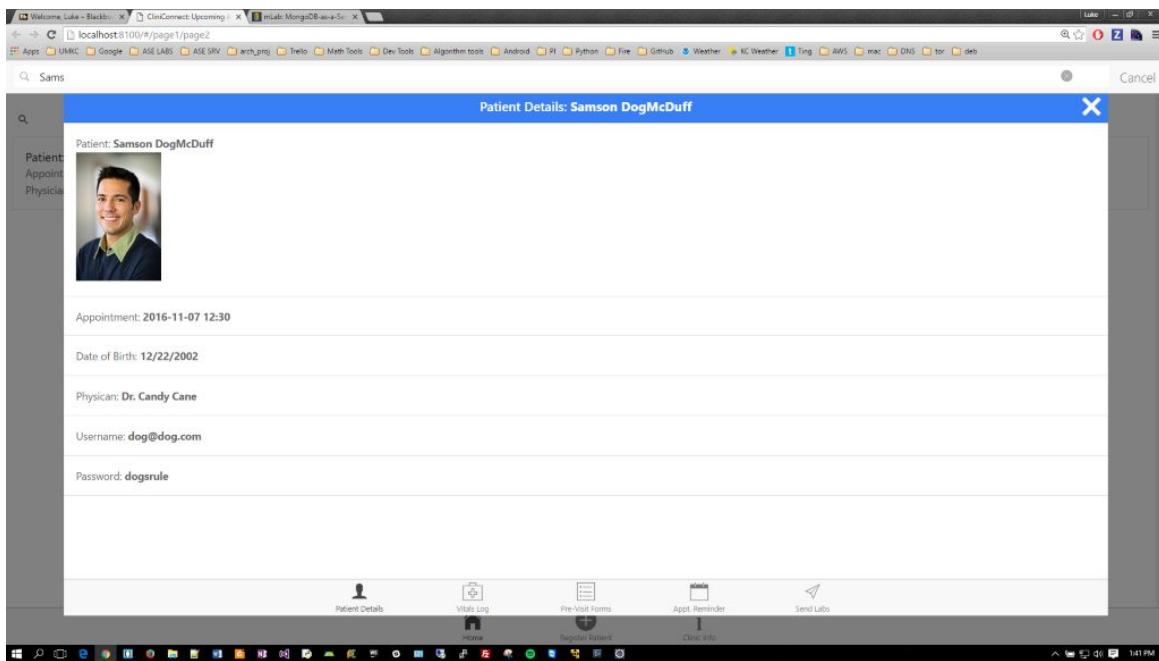
Patient: **Lucienne Joey**
Appointment Date: 2016-09-03 6:30
Physician: **Dr. Marx**

Patient: **Kia Marlys**

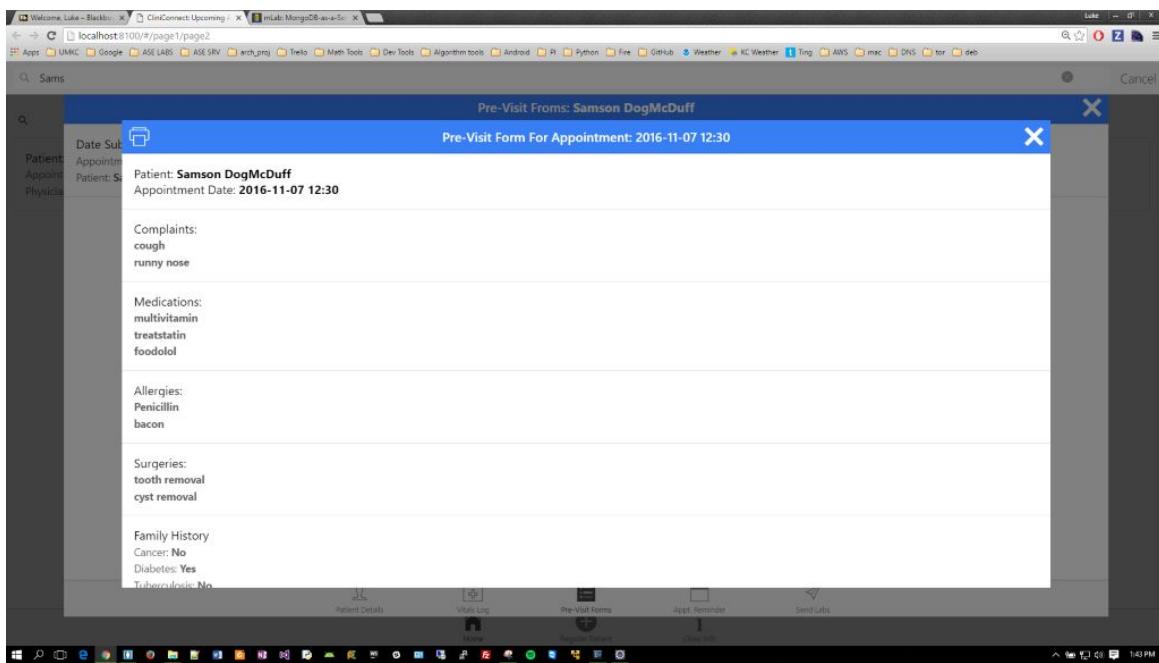
Home search page:

Patient: **Samson DogMcDuff**
Appointment Date: 2016-11-07 12:30
Physician: **Dr. Candy Cane**

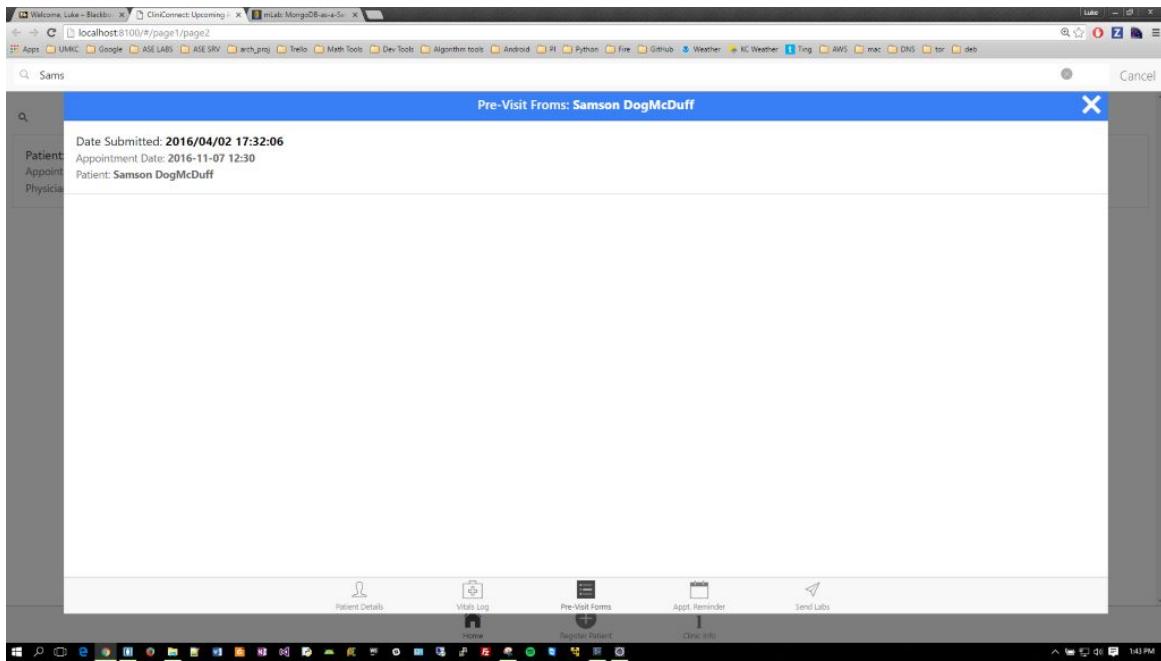
Patient Details:



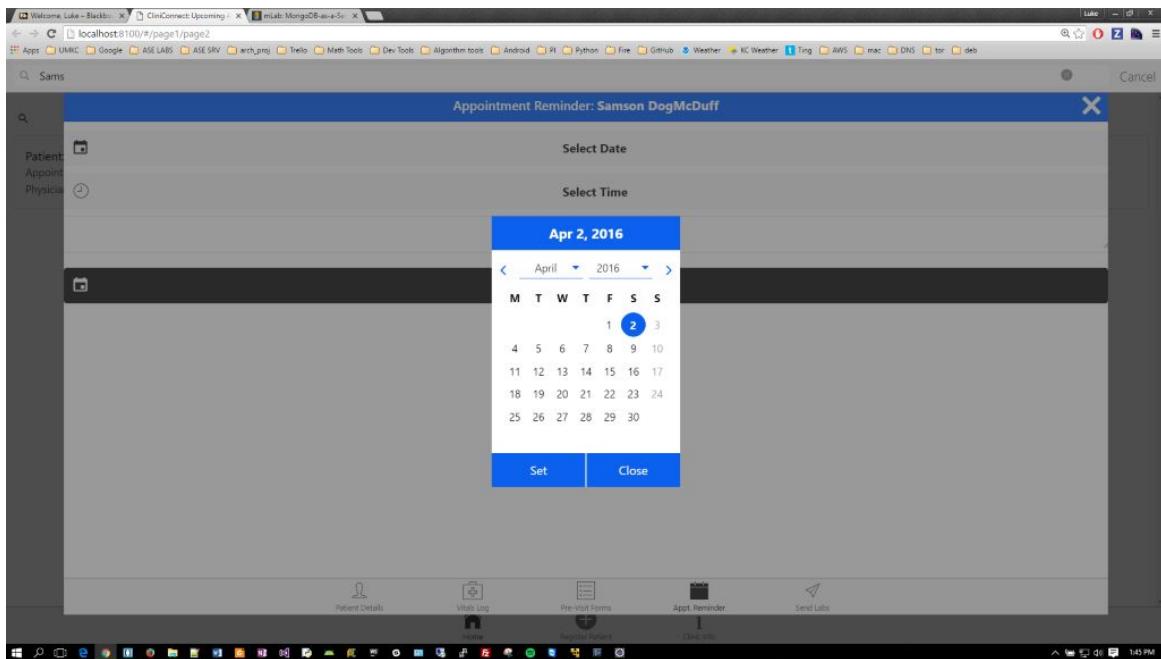
Patient visit form details:



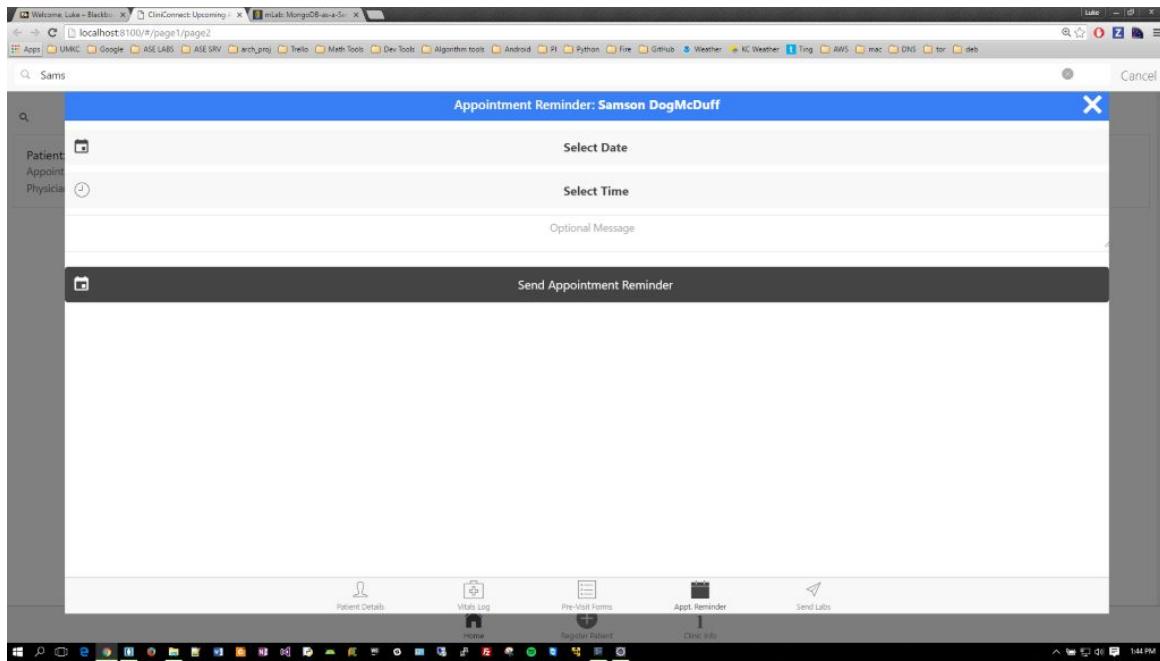
Patient visit form Log:



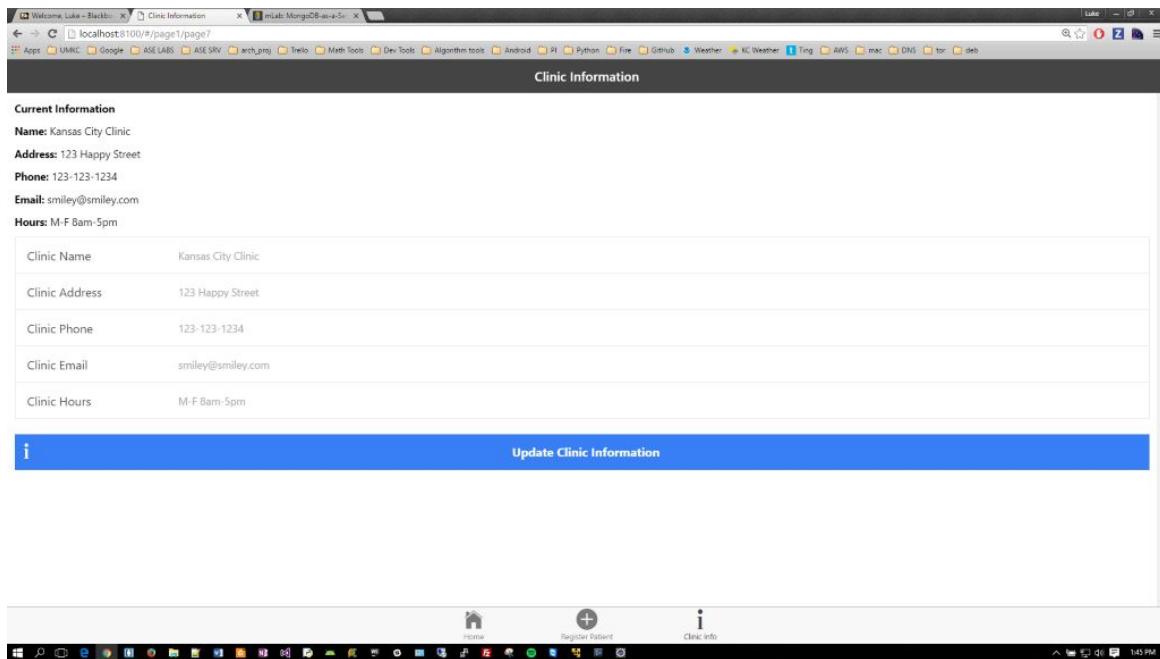
Reminder Date Picker:



Appointment Reminder:



Admin Clinic Information (editable):



Patient Registration:

Lab Sending:

Welcome, Luke - Blackberry X CliniConnect: Upcoming | mLab: MongoDB-as-a-Service

localhost:8100/#/page1/page2

Apps UUNC Google ASE LABS ASE SRV arch_pmc Trello Math Tools Dev Tools Algorithm tools Android Python Fire Github Weather NC Weather Ting AWS mac DNS tor deb

Sams

Send Labs: Samson DogMcDuff

Patient Appoint Physician

Patient: Samson DogMcDuff

Lab:

Clinic Name

Physician: Dr. Candy Cane

Special Instructions: Optional

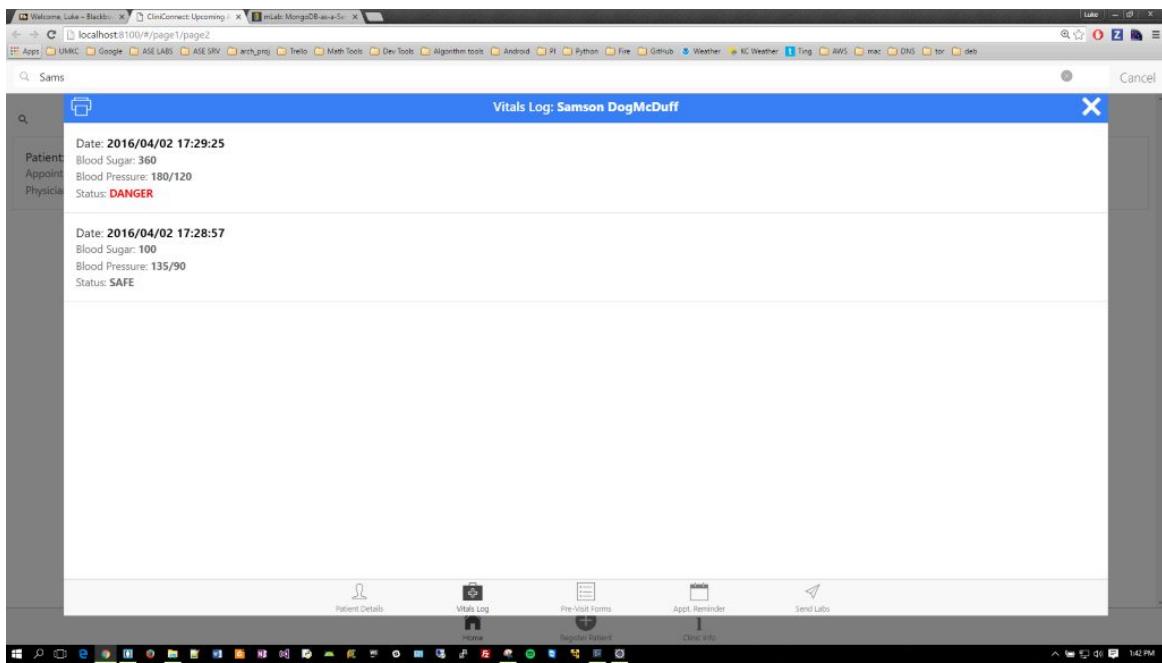
Send Lab Form

>

Patient Details Vitals Log Pre-Visit Forms appt. Reminder Send Labs

Home Register Patient Clinic Info

Blood log/Vitals log:



Patient App:

Visit forms pages:

The image displays two side-by-side screenshots of a mobile application interface, likely from an Android device, showing various patient history and medication forms.

Screenshot 1 (Left):

- TODAY'S COMPLAINT**: Fields for First, Second, and Third complaints.
- MEDICATION ALLERGIES**: Fields for Medication 1, Medication 2, and Medication 3, each with a "Select" dropdown.
- MEDICATIONS I TAKE**: Field for First medication, with a "Select" dropdown.

Screenshot 2 (Right):

- SURGERIES I HAVE HAD**: Fields for First, Second, and Third surgeries, each with a "Select" dropdown.
- DRUGS**: Fields for Do You Smoke? (with a "Select" dropdown), Drink Alcohol? (with a "Select" dropdown), Cancer? (with a "Select" dropdown), Tuberculosis? (with a "Select" dropdown), and Diabetes? (with a "Select" dropdown).
- HAS ANY RELATIVE HAD?**: Fields for Cancer?, Tuberculosis?, and Diabetes?, each with a "Select" dropdown.

Updated Info page:



Unit Test Cases:

JUnits:

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;
public class LevelMonitorTest {
    @Test
    public void isBloodPressureSafeTrueTest() {
        String BP = "120/80";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(true, lm.isBloodPressureSafe(BP));
    }

    @Test
    public void isBloodPressureSafeFalseHighTest() {
        String BP = "145/95";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodPressureSafe(BP));
    }

    @Test
    public void isBloodPressureSafeFalseLowTest() {
        String BP = "85/55";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodPressureSafe(BP));
    }

    @Test
    public void isBlooddSugarSafeTrueTest(){
        String BS = "150";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(true, lm.isBloodSugarSafe(BS));
    }

    @Test
    public void isBlooddSugarSafeTrueTest(){
        String BS = "150";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(true, lm.isBloodSugarSafe(BS));
    }

    @Test
    public void isBlooddSugarSafeFalseHighTest(){
        String BS = "301";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodSugarSafe(BS));
    }

    @Test
    public void isBlooddSugarSafeFalseLowTest(){
        String BS = "69";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodSugarSafe(BS));
    }
}
```

```

import static org.junit.Assert.assertEquals;
import org.json.*;
import org.junit.Test;

public class GetClinicInfoTest {

    @Test
    public void buildClinicInfoTest() throws JSONException {
        GetClinicInfo gci = new GetClinicInfo();
        JSONObject info = gci.buildClinicInfo("loc1", "umkc", "555-5555", "a@a.com",
"7-8");

        assertEquals(info.get("clinicName"), "loc1");
        assertEquals(info.get("clinicAddress"), "umkc");
        assertEquals(info.get("clinicPhone"), "555-5555");
        assertEquals(info.get("clinicEmail"), "a@a.com");
        assertEquals(info.get("clinicHours"), "7-8");
    }
}

```

Yslow:

The screenshot shows the YSlow extension interface for a web page titled "CliniConnect: Upcoming Appointments". The page displays a patient appointment for "Myrtle Racheal" on "2016-03-23 12:45" with the physician "Dr. Lauchana". The YSlow report card is visible, showing a grade of "Grade D" with an overall performance score of 80. A rule set applied is "YSlow(V2)". The URL is "http://cliniconnectadmin.tk/#/page1/page2". The report highlights several issues under the "D: Make fewer HTTP requests" category, such as combining multiple CSS files into one style sheet and using CSS Sprites and image maps. It also suggests combining multiple JavaScript files into one script. The report includes a "Read More" link for further details and a copyright notice from 2016 Iahub Inc.

Implementation status report

Work completed:

The work completed in this increment was, Ionic conversion for administration, visit forms(both user and admin sides), blood log(both user and admin sides), architecture and activity diagrams, and web hosting were completed.

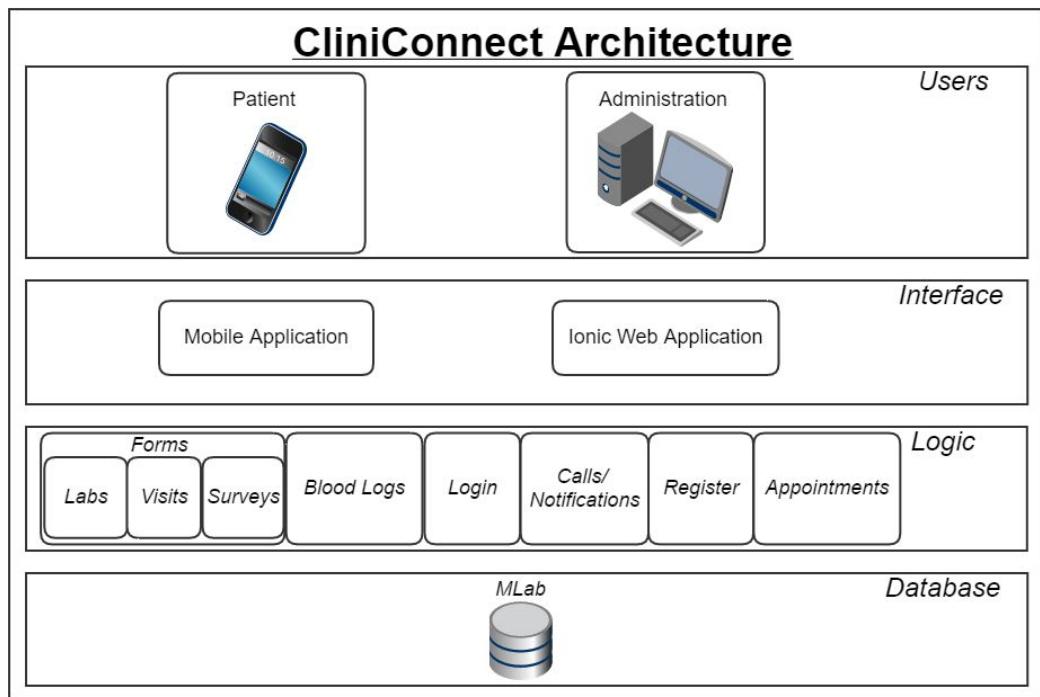
Responsibility (Task, Person)

In this increment, work was divided into tasks to use time efficiently. The task of the blood log went to Shweta, implementing the blood log for both user and admin. Sri handled the forms for both user and admin. Ben, designed the architecture and activity diagrams and wrote the report. Luke implemented the ionic framework conversion and set up the web hosting, as well as performing various tests.

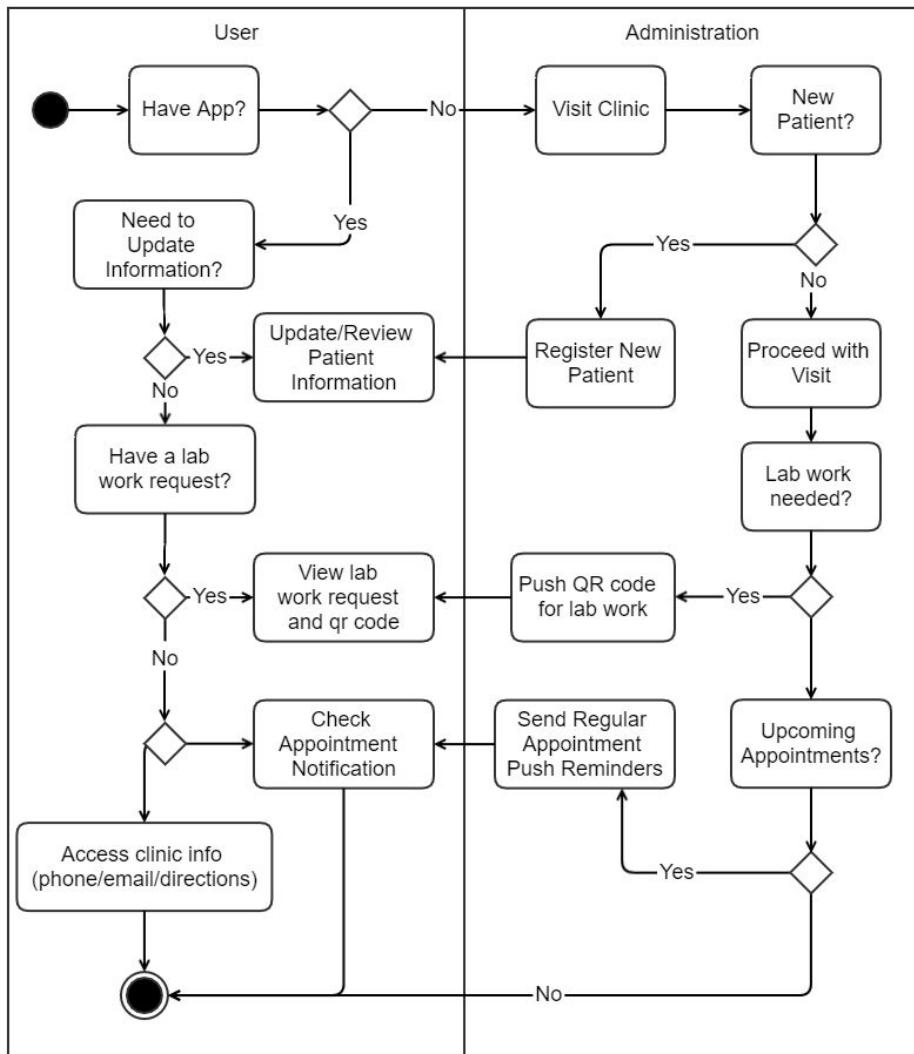
Fourth Increment Report

In this fourth increment of “**CliniConnect**” we have implemented versions that follow the overall structure and flow of the application and administration wireframes and UML diagrams. In this phase we have updated the options available to the user in the vitals log. Included in this increment, is the ability to fill out forms for the exit survey. Also, we have made various changes to make the interface more user friendly.

Architecture Diagram:



Activity Diagram:



Used existing Services/API

Frameworks:

- **Ionic** – AngularJS framework used for Admin web page.
- **Android SDK** – Java framework for Android Apps.
- **GCM** – Google Cloud Messaging used for sending push notifications to users.

APIs:

- **Google Maps**: Used to get directions from the user's current location to the clinic in Google Maps.
- **ZXing (Zebra Crossing)**: Used to convert lab request information into QRcode.

Widgets:

- **Phone Widget**: When selected, the user can call the clinic using existing phone services on the phone like Google Hangouts Dialer or the Phone Dialer.
- **Email Widget**: When Selected, the user can email the clinic using existing email applications on the phone like E-mail, GM
- **ImageView Widget**: Used to display the QR code after it had been processed by ZXing.
- **TextView Widget**: Used to display information about the information embedded in the QR code.

Services:

- **Amazon Web Services – EC2**: Used to host admin website as well as tomcat8 server.
- **MLab** - Mongo as a service.
- **MailJS** - Javascript email service, sends mail to a newly registered user.

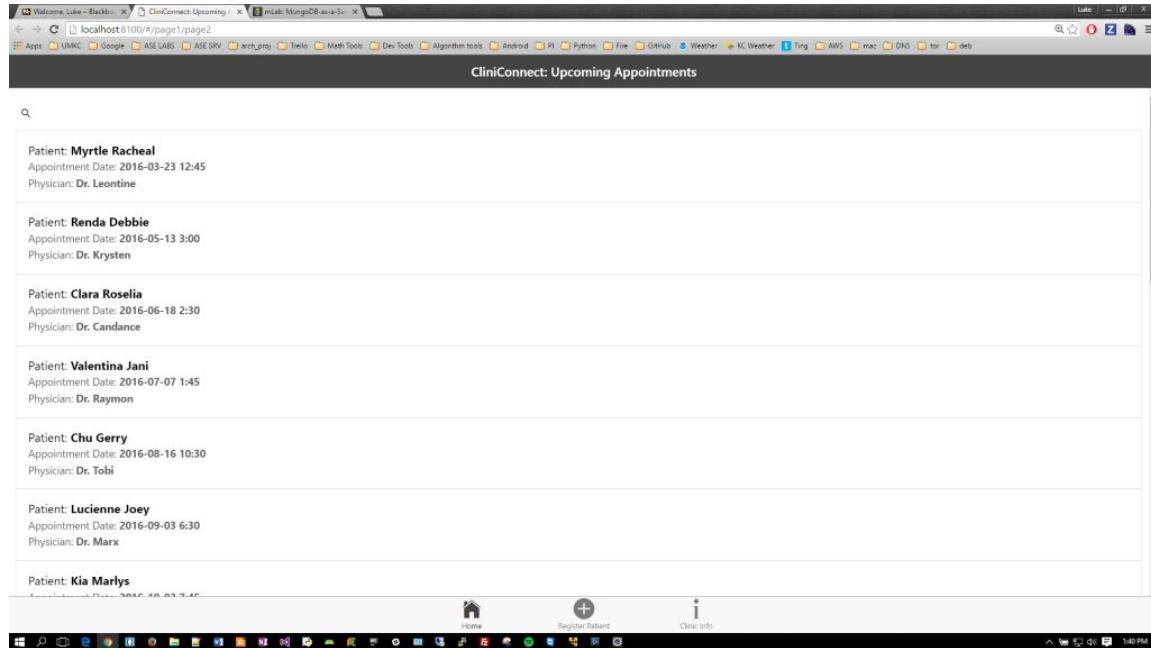
Implementation and Deployment:

We have implemented an ionic framework for the admin web site, the blood log activity for user and admin sides, and the visit forms for admin and users. The admin website has been implemented with Apache http server on Amazon Web Services. [A link to the admin site page.](#)

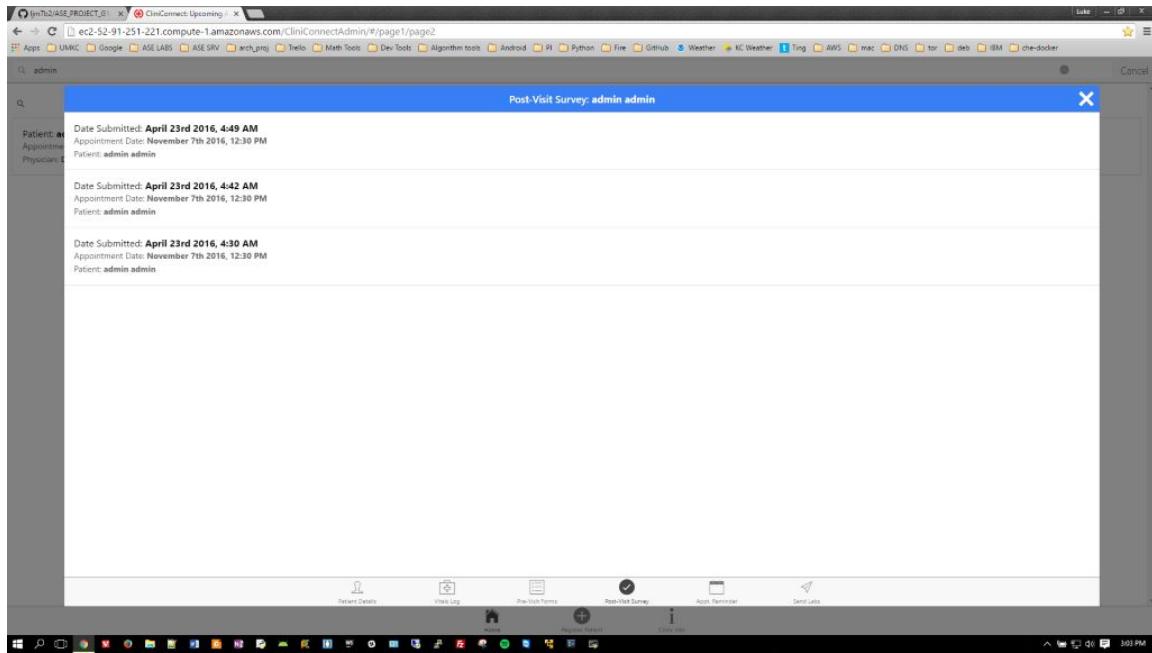
Below are the screenshots.

Web Page:

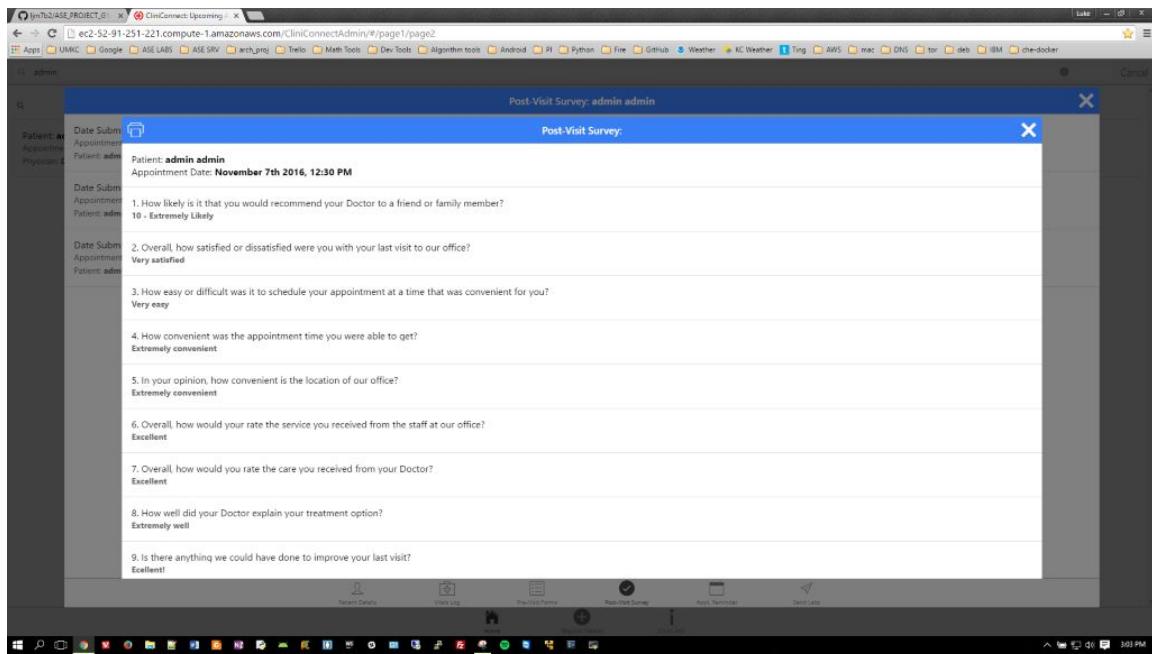
Appointment List (New Format):



Exit Survey Results Pg1:



Exit Survey Results Pg2:



Vitals Log (New format):

The screenshot shows a web browser window titled "Vitals Log: admin admin". The main content is a table with the following data:

Date	Blood Sugar	Blood Pressure	Status	Temperature	Well Being
April 24th 2016, 12:20 AM	90	120/80	SAFE	98.9°	I feel super freaking awesome!
April 24th 2016, 12:09 AM	12	12/12	DANGER	12.9°	good
April 24th 2016, 12:00 AM	90	120/80	SAFE	98.9°	Good

Below the table, there is a toolbar with icons for Patient Details, Visit Log, Visit Forms, Visit Summary, Visit Reminder, and Demographics.

Patient App:

Vitals Log (Updated):



Exit Survey:

1. HOW LIKELY IS IT THAT YOU WOULD RECOMMEND YOUR DOCTOR TO A FRIEND OR FAMILY MEMBER?

Please select an option Select

2. OVERALL, HOW SATISFIED OR DISSATISFIED WERE YOU WITH YOUR LAST VISIT TO OUR OFFICE?

Please select an option Select

3. HOW EASY OR DIFFICULT WAS IT TO SCHEDULE YOUR APPOINTMENT AT A TIME THAT WAS CONVENIENT FOR YOU?

Please select an option Select

4. HOW CONVENIENT WAS THE APPOINTMENT TIME YOU WERE ABLE TO GET?

Please select an option Select

5. IN YOUR OPINION, HOW CONVENIENT IS THE LOCATION OF OUR OFFICE?

Please select an option Select

6. OVERALL, HOW WOULD YOUR RATE THE SERVICE YOU RECEIVED FROM THE STAFF AT OUR OFFICE?

Please select an option Select

7. OVERALL, HOW WOULD YOU RATE THE CARE YOU RECEIVED FROM YOUR DOCTOR?

Please select an option Select

8. HOW WELL DID YOUR DOCTOR EXPLAIN YOUR TREATMENT OPTION?

Please select an option Select

9. IS THERE ANYTHING WE COULD HAVE DONE TO IMPROVE YOUR LAST VISIT?

Please Enter Here

10. HOW WELL DID YOUR DOCTOR EXPLAIN YOUR FOLLOW-UP CARE?

Please select an option Select

Unit Test Cases:

JUnits:

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;
public class LevelMonitorTest {
    @Test
    public void isBloodPressureSafeTrueTest() {
        String BP = "120/80";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(true, lm.isBloodPressureSafe(BP));
    }

    @Test
    public void isBloodPressureSafeFalseHighTest() {
        String BP = "145/95";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodPressureSafe(BP));
    }

    @Test
    public void isBloodPressureSafeFalseLowTest() {
        String BP = "85/55";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodPressureSafe(BP));
    }

    @Test
    public void isBlooddSugarSafeTrueTest(){
        String BS = "150";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(true, lm.isBloodSugarSafe(BS));
    }

    @Test
    public void isBlooddSugarSafeTrueTest(){
        String BS = "150";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(true, lm.isBloodSugarSafe(BS));
    }

    @Test
    public void isBlooddSugarSafeFalseHighTest(){
        String BS = "301";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodSugarSafe(BS));
    }

    @Test
    public void isBlooddSugarSafeFalseLowTest(){
        String BS = "69";
        LevelMonitor lm = new LevelMonitor();
        assertEquals(false, lm.isBloodSugarSafe(BS));
    }
}
```

```

import static org.junit.Assert.assertEquals;
import org.json.*;
import org.junit.Test;

public class GetClinicInfoTest {

    @Test
    public void buildClinicInfoTest() throws JSONException {
        GetClinicInfo gci = new GetClinicInfo();
        JSONObject info = gci.buildClinicInfo("loc1", "umkc", "555-5555", "a@a.com",
"7-8");

        assertEquals(info.get("clinicName"), "loc1");
        assertEquals(info.get("clinicAddress"), "umkc");
        assertEquals(info.get("clinicPhone"), "555-5555");
        assertEquals(info.get("clinicEmail"), "a@a.com");
        assertEquals(info.get("clinicHours"), "7-8");
    }
}

```

Yslow:

The screenshot shows the YSlow extension interface running on a Chrome browser. The URL is <http://cliniconnectadmin.tk/#/page1/page2>. The overall grade is D, with an overall performance score of 80. The report highlights several issues under the 'Grade D on Make fewer HTTP requests' section:

- D: Make fewer HTTP requests**
- F: Use a Content Delivery Network (CDN)**
- A: Avoid empty src or href**
- F: Add Expires headers**
- A: Compress components with gzip**
- A: Put CSS at top**
- F: Put JavaScript at bottom**
- A: Avoid CSS expressions**
- A: Minify JavaScript and CSS external**
- A: Reduce DNS lookups**
- F: Minify JavaScript and CSS**
- A: Avoid URL redirects**
- A: Remove duplicate JavaScript and CSS**
- A: Configure entity tags (ETags)**
- A: Make AJAX cacheable**
- A: Use GET for AJAX requests**
- A: Reduce the number of DOM elements**
- A: Avoid HTTP 404 (Not Found) error**
- A: Reduce cookie size**
- A: Use cookie-free domains**
- A: Avoid AbnormalImageLoader filter**
- A: Do not scale images in HTML**
- A: Make favicon small and cacheable**

The report also notes that the page has 11 external JavaScript scripts and 4 external style sheets, suggesting they should be combined into one.

Implementation status report

Work completed:

The work completed in this increment was, the implementation of the exit survey, the vitals log update, slight changes to the interface of the admin site, sending an e-mail to the user to confirm registration, final comprehensive testing, the presentation PowerPoint and the presentation video.

Responsibility (Task, Person)

In this increment, the main focus was to complete the project and be able to present a cohesive system. Shweta updated the vitals log and worked on the PowerPoint. Sri handled the exit survey forms for both user and admin. Ben, wrote the report, made the video and assisted with final testing and bug reporting. Luke updated changes to the admin site to make it user friendly, such as adjusting display formatting and adding an email sent for user registration confirmation, as well as performing testing and debugging.

Powerpoint Slides



CliniConnect

Advanced Software Engineering(CS5551)
Guided By-Professor Lee
Project Team -1
Shweta Parihar(42)
Luke McDuff (27)
Benjamin Chrysler (8)
Sri Chaitanya Patluri (44)



ClinicConnect

Motivation:

There are many patient to clinic application exist, which are typically patient portal containing overwhelming large amount of data and features. A simple application with few key important features which can help patient to monitor regular health and share the details with doctors easily and timely manner.

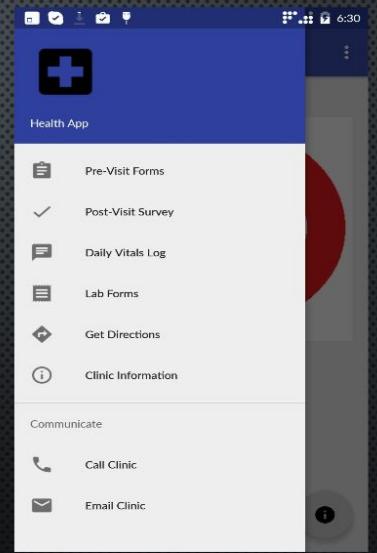
Significance:

- CliniConnect is a communication application to help Patients connect easier with their clinic and doctors connect easier with there patient.
- The application targets at-risk patients typically in low income areas who face challenges connecting with their Clinic.

CliniConnect Android application

Patients can use the CliniConnect Android application to:

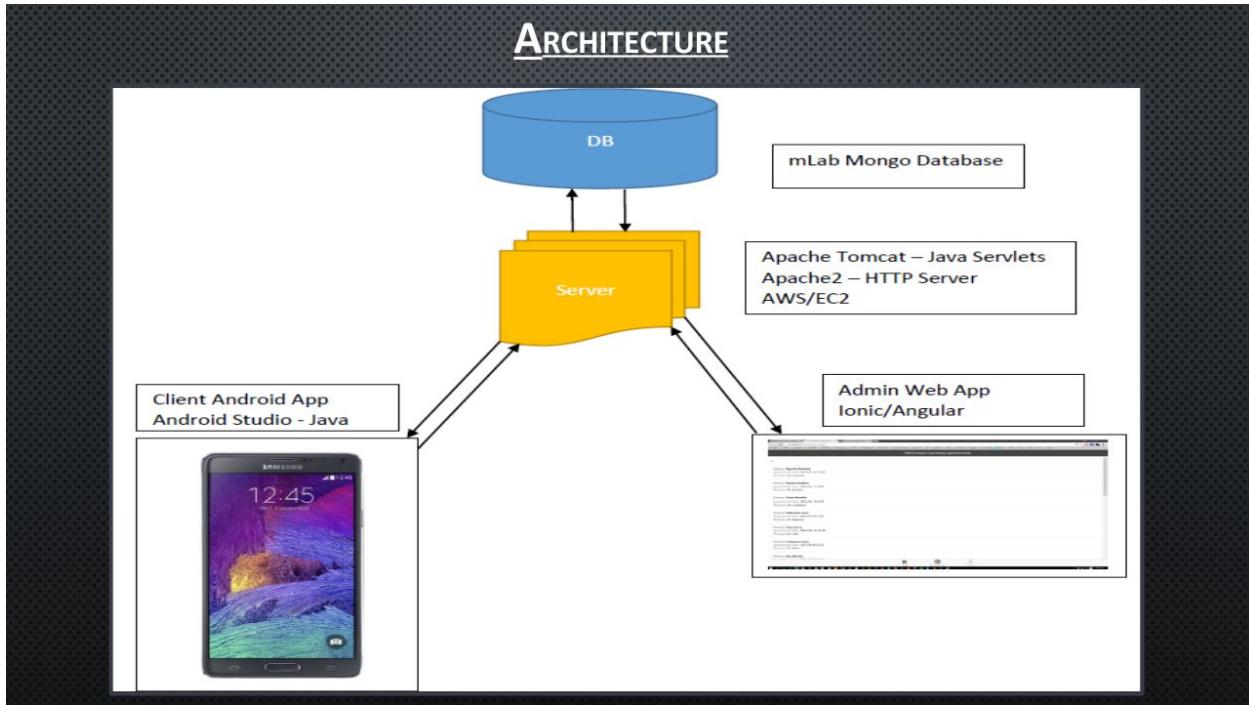
- Fill out Pre-Visit paperwork before clinic visit and save waiting time.
- Complete Post-Visit surveys.
- Send vital information
 - Blood pressure
 - Blood sugar
 - Temperature
 - Description of current state of wellness
- Receive lab work orders with QR code
- Receive turn-by turn directions to the clinic in a single click.
- Call/email the clinic without having to enter an email or dial a number.
- Receive appointment reminder push notifications on phone.



CliniConnect web application

Clinic administrators(Doctor/Nurses) can use the **CliniConnect** web application to :

- Register a new patient into the system
- View all registered patients using android application
- View all daily vitals log entered by any patient.
- View the submitted Pre-Visit and Post-Visit Paperwork for any patient.
- Send a patient a push notification appointment reminder which will pop up on patient smart phone.
- Send a patient a new lab order with scanable QR code
- Update new information about the clinic that controls android application features like: navigation & call/email.



Live Demo

 **CliniConnect**

Github URL: https://github.com/SCE-UMKC/ASESP16_CliniConnect_1

Youtube Project Video URL: <https://www.youtube.com/watch?v=5t4klitnIH0>