Medical Doctors in Malawi need a system that allows them to triage, track, store, and provide metrics on patients they assess in rural areas of Africa. This system must take into consideration the limited power and internet connection of these rural areas.

Mobile Clinic v1.0 provided a solution to this problem and was given to Orant Charities for use on the field. Upon return from their mission, Orant charities provided feedback on features they would like to see introduced or improved.

As a Client User, I want to:
- Close visits from the Client, and remove patients from the Queues.
- Remove all Personal Identifiable Information from the Client at the end of a Clinic visit.
- Close and remove a Patient from the Doctor’s station directly.

As a Local Server Admin, I want to:
- Remove all Personal Identifiable Information from the Local Server at the end of a Clinic visit.

The Client slows down when 100+ patients populate the system.
- Synchronization between between the Client and Local Server can take several minutes.
- Personal Identifiable Information (PII) persists in the system indefinitely.

The System implements Command, Factory, Façade, and Singleton design patterns.
- PatientObject and VisitationObject handles creating and closing patients and.
- ServerCore and GCDAsyncSocket handles the connection and transfer of data between the Client and Local Server.
- MobileClinicFacade handles what data is communicated between the Local Server and the Client.

Verification

- The project was developed according to Scrum principles (iterative design and development)
- Preliminary testing was performed using Xcode unit testing tools in order to verify that only open patients were populating the queue.
- Integration testing was performed using a bottom-up approach to verify that information was removed from the Client and Local Server.
- Testing the System Purge was conducted by manually viewing the contents of the .sqldata.

Object Design

- The System implements Command, Factory, Façade, and Singleton design patterns.
- PatientObject and VisitationObject handles creating and closing patients and.
- ServerCore and GCDAsyncSocket handles the connection and transfer of data between the Client and Local Server.
- MobileClinicFacade handles what data is communicated between the Local Server and the Client.

Implementation

- The Client Purge will not contain closed visits, reducing the size of the array that the Patient Queues query from.
- Once a patient or visit has been closed and sent to the Local Server, the record will be removed from the Client database.
- A new system purge was implemented to remove all patient and user information from both the Local Server and Client databases.
- A Destructive Sync on the Local Server and Client was implemented for the initial deployment of a clinic.

Summary

Mobile Clinic v2.0 will provide Orant Charities, along with other charities:
- An improved patient processing rate.
- Improved Client responsiveness to handle larger patient loads.
- A streamlined way for doctors to close patients directly.
- Improved security to protect Patient and User Information.

Acknowledgement

The material presented in this poster is based upon the work performed by Orant Charities in Malawi, Africa.

I am thankful to the help that I received from my group members: Kevin Diaz, Ernesto Perez, and Humberto Suarez.

I am thankful to the Developers of Mobile Clinic Version 1: Steven Berlanga, Carlos Corvaia, Rigoberto Hernandez, Michael Montaque, Sebastian Zanlongo.