**Ricky:**

**S1:** Good evening, my name is Ricky Orndorff and this is Ryan Kane and we are here tonight to present to you SEMRS, a secure emergency medical health records management system.

**S2:** Just to give you an idea of the background of our project, this project is intended for MCHC, located in Cape Town, SA. This area in South Africa does not have any health insurance standards mainly because doctors will often require immediate payments for their health services. The average annual income in 2005 was about $10,000.00 and also 1 out of 5 people are diagnosed with HIV/AIDS. A majority of population has a 7th grade education.

**S3:** What is the problem we are trying to solve?

MCHC uses a paper based system to store all patients’ files. Because of the area of the Mitchell’s plain community, patient’s files are often misplaced or damaged because Patients are responsible for keeping their own paperwork. This leads to the delay of the treatment process. Often times the treatment must be restarted because of this. The confidentiality of the patient’s files can be compromised if the files are lost or stolen. There is almost no data integrity with a paper based system because the paper files can be destroyed or lost.

**S4:** This health clinic needs an improved electronic medical records management system that is secure to uphold confidentiality of patient files, as well as ensure data integrity. Because the majority of the population is poor and only has a 7th grade education level, this system must be easy to use. Also, because there are no health insurance standards, this system will adhere to the standards of HIPAA, the health insurance portability and accountability act. Our proposed solution is to develop a lightweight and secure system that addresses the needs of MCHC.

**S5:** Just to give you a background on HIPAA standards, HIPAA requires four technical safeguards that SEMRS has adapted. They are: access control, audit controls, integrity controls, and transmission security.

**Ryan:**

**S6:** There have been many open source electronic medical records management systems that exist and here they are. There have been EMR that enforce patient confidentiality, ensure data integrity, and are HIPAA compliant. The following systems are the most prevalent. ZEPRS, introduced in 2001 has an offline mode and role based access control, but the limitation of the system is that it lacks security. Also introduced in 2001, OpenEMR, the most mature system has many features including insurance and billing support, but the system is very complex and it lacks encryption. A more recent system, THIRRA, is a mobile tele-health emergency health records system for rural and remote areas. It has a bio-surveillance feature that tracks outbreaks in real time. This system also lacks security and you begin to see a trend.

**S7:** Our proposed implementation is based off of OpenEMR, but it will be not as complex and really focus on encryption of patient information. The application will be available through intranet and feature a public/private key pairs through a Secure Socket Layer which uses SHA1 256 bit encryption.

**S8:** Our system is a client-server framework. We have a PHP user interface that is written on the front-end, and a MySQL database on the back-end. The user interface allows users to interact with the database through our various encryption algorithms.

**S9:** The system greets a user with a login screen which upon authentication redirects them to their appropriate user group interface.

**S10:** In summary, we have created a streamlined secure system that uses SSL encryption, and ensures that patients’ treatment will no longer have to be restarted and that the data’s integrity is upheld. To ensure that the system is secure, we will log off idle users. Our access controls keep track of which users modify data and ensure that privileges stay within the user group.

**S11:** Possible future work, for this application, would be to add the ability to add existing multimodal hardcopy files: such as x-rays and lab reports. Lastly, greater domain portability for use in clinics that cater to cancer centers, dentists, and other medical practices.

**Demo:**

1. Make sure that I have a “mug shot” profile photo for the mock patient
2. Login as Receptionist
3. Add a new patient (Make sure you show all information that could be inputted.)
4. Select patient
5. Refresh page
6. Add appointment
7. Log out
8. Log in as Physician
9. View newly create patient
10. Change photo to the one stored from step #1
11. View the appointment
12. View the access log
13. Possibly show what the data looks like in the database\*\*\*