Project Development Phase

Debugging Traceability

| Team Id	NM2023TMID04415
Project Name	Block Chain Technology For Electronic Health Records

Introduction:

Electronic health record (EHR) systems present a formidable "trustworthiness" challenge because people's health records, which are transmitted and protected by these systems, are just as valuable to a myriad of attackers as they are to health care practitioners. Major initiatives in EHR adoption and increased sharing of health information raise significant challenges for protecting the privacy of patients' health information.

Patient-Centered Functionality:

The primary way of tracking care for a given patient is through office visits. An office visit represents a specific consultation with an LHCP on a specific date in a specific location. Various standardized health care codes are linked to office visits, including diagnoses, immunizations, procedures, prescriptions, and general demographics

such as height and weight. The LHCP logs the information for a given office visit, and the patient can view the records for of his or her previous office visits. Patients can also take a satisfaction survey on the LHCP, which is aggregated for other patients in search for an LHCP.

Figure 1. High-level Overview of the iTrust Use Cases

iTrust Architecture:

The iTrust source code is designed around the Model-View-Controller design pattern [4]. The

goal of this organization is to separate the logic associated with the user interface (i.e. the "view") from the logic of the persistent storage (i.e. the "model"), while organizing most of the complex business logic in one place (i.e. the "controller"). In iTrust, the view is implemented in JavaServer Pages (JSPs), the controller is implemented in Java, and the model is implemented in SQL and Java. An overview of the iTrust architecture can be found in Figure 2.
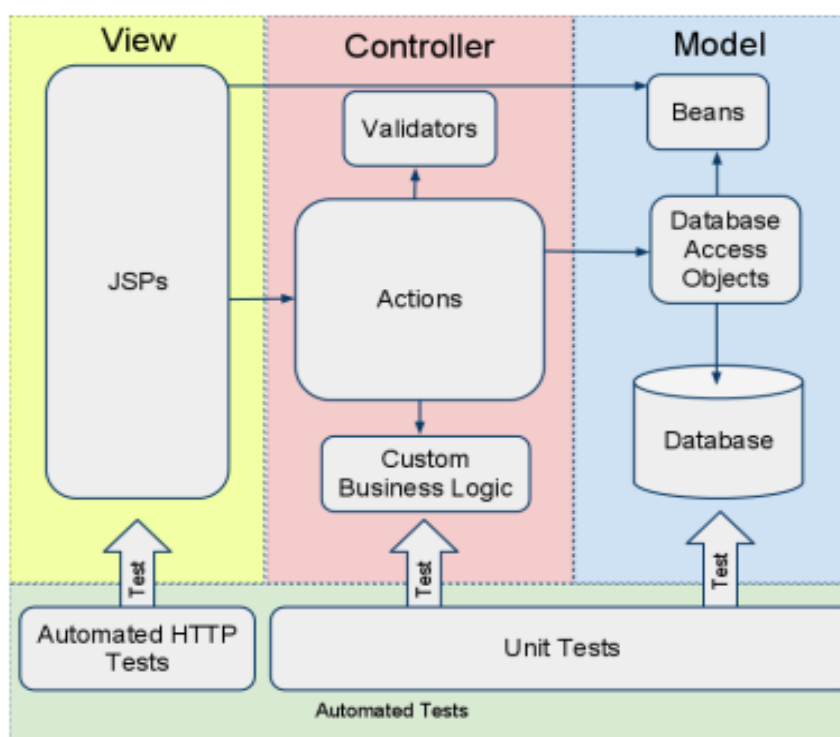
Traceability in iTrust:

Figure 3 shows an overview of how the test plan, requirements, and system archetypes are traced to each

other. The requirements document contains sub-sections for each use case (UC) that trace to the implementing JSP. Students can use this traceability analysis to find the place in the code that implements a given requirement for comprehending the code as well as improved testing. Additionally, the whole traceability matrix is available for students on the wiki for posterity.

Figure 3. Traceability Overview