Team 5

Cupola

Project: Tablet PC

Final Report

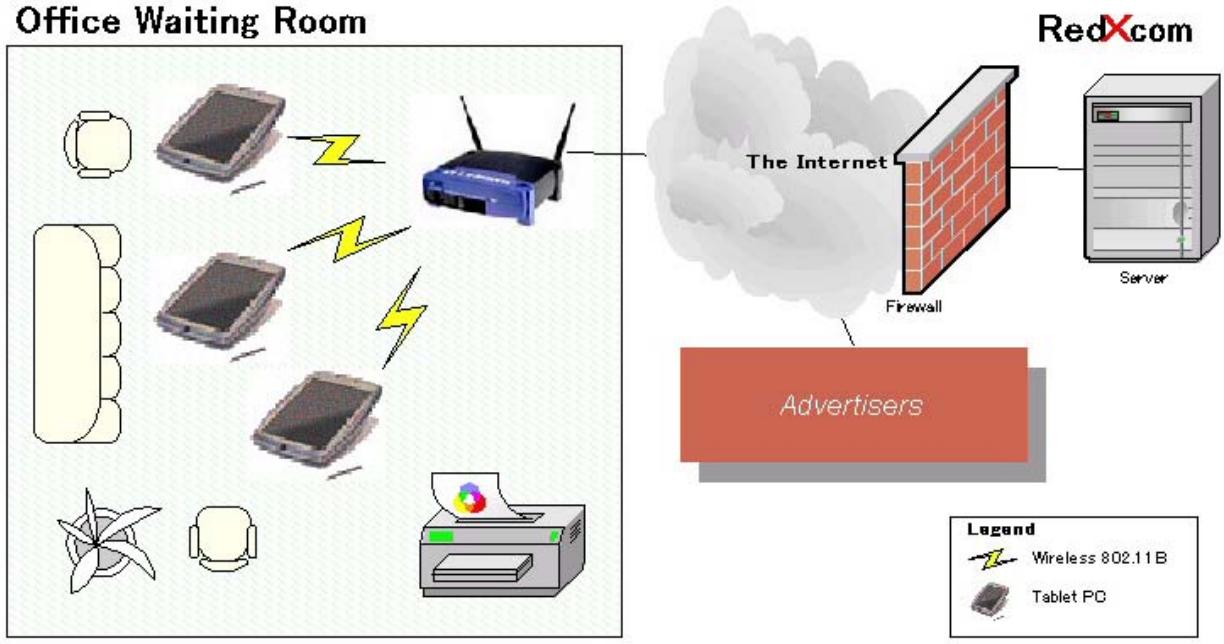
Table of Contents

Project Description

RedX is a new high-tech startup company that promises to enhance the experience of patients waiting in a dentist's waiting room. Currently, patients are being asked to complete lengthy medical forms before an appointment. RedX and FRED a.k.a. Cupola will develop a hand-held (tablet pc) wireless solution that allows patients to complete these forms on-line. The product will be a complete infotainment system that entertains patients while they wait and at the same time serves them with targeted advertisements. The ensuing sections identify the most important features, functions, and requirements for the system.

Product Perspective

The following "cartoon" shows at a high level the general topology of the system. Several (2 to 3 estimated) tablet PCs will be used in each dentist's office. These tablet PCs will communicate with the central RedX server through the office's Internet Gateway where patient forms will be stored and also from where targeted advertisements will be delivered.



Product Functionality/Features

This product has two principal functions:

1) To replace paper insurance forms that need to be filled out or updated each time a patient enters a waiting room of a dentist or a doctor.
2) To attract users to watch Internet advertisements while also providing an infotainment service to the user.

The tablet pc will be the device that is used to interface with the patient in the waiting room. The patient will use this tablet to update personal information. Once the information

is recorded, the user will be able to use the tablet as an 'infotainment' device where the user can browse the internet, check out information about the doctor they are visiting, play games etc. While in use, the tablet will also display advertisements targeted to the particular user, based on the information that the user entered when filling out the medical form. RedX's web site will be the default home page for the tablet and will also serve as the Internet portal for the user.

Quality Attributes

The client is especially concerned with security, since the system will be handling potentially sensitive personal information. Performance is also a major driving requirement: the end user should not experience noticeable delays when using the system. Finally, the system should be able to scale as the number of dentist offices using the system increases.

Constraints

The following constraints have been identified for this project:

Hardware Platform, Operating System, and COTS

All components of the system will run under the Windows 98SE, 2000 or XP operating systems except hand-held computers (tablet PCs), which will either run Windows CE or Windows XP. Commercial off the shelf (COTS) software will be used to construct the system.

Business Constraints

System cost should not exceed $5,000 for each dental office. This includes a package of three hand-held clients, networking hardware, server (if required), software, and installation charges.

Network and Bandwidth Limitation

Network is limited to connect via the Internet. System bandwidth limitations will be restricted by DSL or a DSL comparable Internet connection. The connection shall be capable of download rates of at least 500Kbps. This assumes streaming video takes 200Kbps of the bandwidth, and each office is capable of supporting 2 tablets.

Assumptions and Dependencies

The creation/generation of advertisements is out of the scope of this project. Backup and typical system administrative information is also outside our scope. And it has been decided that the system will not interface with third party dental software, such as Dentrix and SoftDent.

Potential Extensions

Future releases of the system could incorporate a mechanism for even more targeted advertisements. This could be achieved primarily through integration with consumer information databases.

RedX System Architecture

This section describes the functional requirements that drive the RedX architectural design. In some cases, references for detained requirements and constraints are made to [FRED-SRS].

Before presenting a high-level architectural design for the system, the project's driving quality attributes are described, followed by the Utility tree. Descriptions of the specific scenarios addressed by the system are provided in the Appendix.

Driving Functional Requirements

Detailed descriptions of the functional requirements that the RedX system must implement can be found in the Software Requirements Specification document [FRED-SRS].

Future Extensions

The FRED Software Requirements Specification document [FRED-SRS] also contains descriptions of possible future extensions for the system.

Constraints

Both technical and business constraints can be found in the FRED Software Requirements Specification document [FRED-SRS] and are therefore not repeated here.

Driving Quality Attributes

Software quality can be defined as "the degree to which software posses a desired combination of attributes (e.g., reliability, interoperability)." [IEEE 1061] Identifying the system's most important quality attributes is crucial to success. For the RedX system, the following attributes were found to be the primary driving quality attributes. Concepts from the Architectural Tradeoff Analysis Method (ATAM) were used to identify and prioritize these attributes. [K+98]. These Quality attributes are listed in the order of importance from highest priority, too least.

Quality Attribute	Description	Metric

Usability	There are two types of users considered: maintenance staff (including administrators) and patients in the waiting room. Usable maintenance software increases efficiency of maintenance staff. Usable client software for presenting the medical form decreases time to complete the form and thereby increases available time to view advertisements. The system will help users complete the medical form with usability principles, such as error prevention, recovery, and leveraging human knowledge.	Average time to complete medical form, per page. Average time to complete standardized maintenance task(s). Number of user-initiated corrections per medical form submission.
Security	Due to the sensitive nature of the recorded medical information, security is important. User authentication and encrypted data storage and transmission (HTTPS, SSL, etc.) procedures will be used.	Mean time to system break-in. Mean time to unauthorized use. Number of unauthorized accesses per month.
Reliability	The system should perform its functions as intended or not perform them at all. For instance, the system should display ads according to the ad-matching algorithm provided that the remote RedX component is available. Reliable exchange of information is central to the system. Information from the users in the waiting room should be able to reach the central server reliably and visa versa. The system should support atomicity, consistency, isolation and durability for each transaction. The system will operate in a fail-safe mode and contain redundant components where necessary. Error reporting/handling mechanisms will be put in place to allow for traceability and safe recovery.	Number of inconsistent system states per month. Mean time to failure. Mean time to safe recovery.

Modifiability & Extensibility	Modifiability is important to allow for this first prototype to be turned into a production system. This will also allow for the system to be easily extended for use in other waiting rooms (e.g. car dealerships, etc.). And the system may also end up interfacing with commercial off-the-shelf products. It should be modular enough to accommodate such integrations. Our plan is to develop the system using an incremental prototype. Modular, object-oriented design and a tiered architecture will be used to conceal the implementation details of one component from another.	Mean time to add single function to component or modify a component's functionality. Mean time to develop connection with new component. Number of conflicts introduced per new component.
Performance	Performance is important when service content to end-users with limited browsing time and patience. Every component of the system could be a potential performance bottleneck. The system will be designed with performance in mind, optimizing actions such as database accesses, encryption/decryption, use of caching, etc. The system will also be designed with scalability in mind (see Scalability).	Mean time to complete one user-session (single start-to-finish user interaction). Min, Max, Average and Mean time to deliver ads for a single page to a single user.

The following remaining attributes are important to the project, but not as critical as the driving quality attributes.

Quality Attribute	Description	Metric
Scalability	Rapid growth is important for the system to reach the business goals of the client. Bottleneck components will be modifiable to operate in parallel without conflict, thus allowing the system to scale as close to linearly as possible.	Scalability tests. Aggregated development time per user. Number of users in concurrent operation.

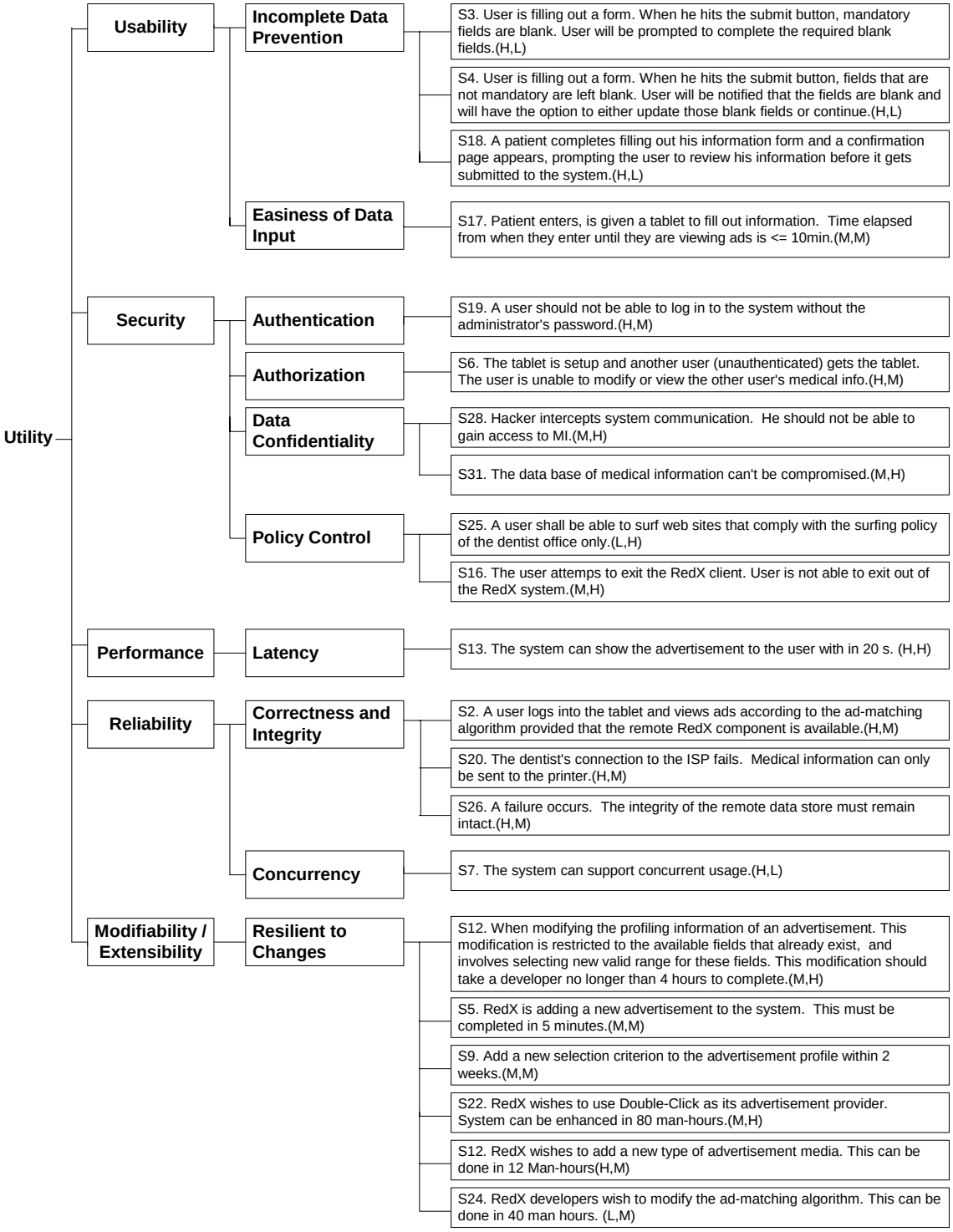
Testability	Testability is important to our process and our approach of using an iterative prototype. Components will be coded to be able to certify with an automated test suite. Component interfaces will allow for testing, while the details of the implementation are concealed in order to modularize the component test suite.	Time to complete test certification per test, per component. Number of defects found during testing / number of total defects found. Mean time to locate defect after insertion.

Utility Tree

Following is the system's utility tree, which is an ATAM output used to prioritize quality attributes based on specific usage scenarios that system might face. Leaf nodes are specific enough to be prioritized relative to each other along two dimensions: importance of each node to the success of the system and the degree of difficulty to implement.

Shown are leafs that are of high importance. Leafs that are of high importance and high difficulty represent risks to the project that should be explored in depth. Detail scenarios are presented in this document's appendix.

The process used by this team to identify the key scenarios was to rank them in order of importance. This ranking activity was accomplished by allowing all 5 team members to cast a vote on a scale of 0-5. 5 identify a scenario that a team member viewed as very important to the success of this project. 0 represented a scenario that could be left out, and not impact the project. Once all team members had an opportunity to vote, the votes were totaled (this is identified as the "Ranking score" for each scenario). It was determined that 42 scenarios would be too many to analyze by the entire team. The team chooses to choose the top 20 scenarios for creating a utility tree, and eventually for an ATAM activity.



Proposed System Architecture

In order to present the system's overall architecture, several views will be presented, starting at the most abstract level. When a component is subject to risks exposed by the scenarios, analysis of alternatives, specific risks, and trade-offs will be presented so the reader can gain an understanding of why particular design decisions were made. Figure

1 represents a view of the system level architecture that identifies the major components that, and their interaction with one another.

Components and Connectors

The system's major components include those within the dentist offices, the RedX.com site, and external web sites that provide marketing content. Within the dentist office are wireless hand-held TabletPCs that gain access to the Internet through a small office firewall/router to a broad band Internet connection. The TabletPC software is a thin-client running a customized web browser (Microsoft Internet Explorer MSIE) which will display streaming video in Windows Media Player (WMP).

The RedX.com application infrastructure, which will be decomposed later, is hosted by an Internet Service Provider who maintains a firewall to restrict entry to named TCP ports. The connectors shown depict pervasive Internet standard protocols, such as HTTP and SMTP. Note that an e-mail server is included in the RedX.com infrastructure, but the tablets do not connect to this server, which is to be used for outbound e-mail. (See Figure 2, RedX Major Components)

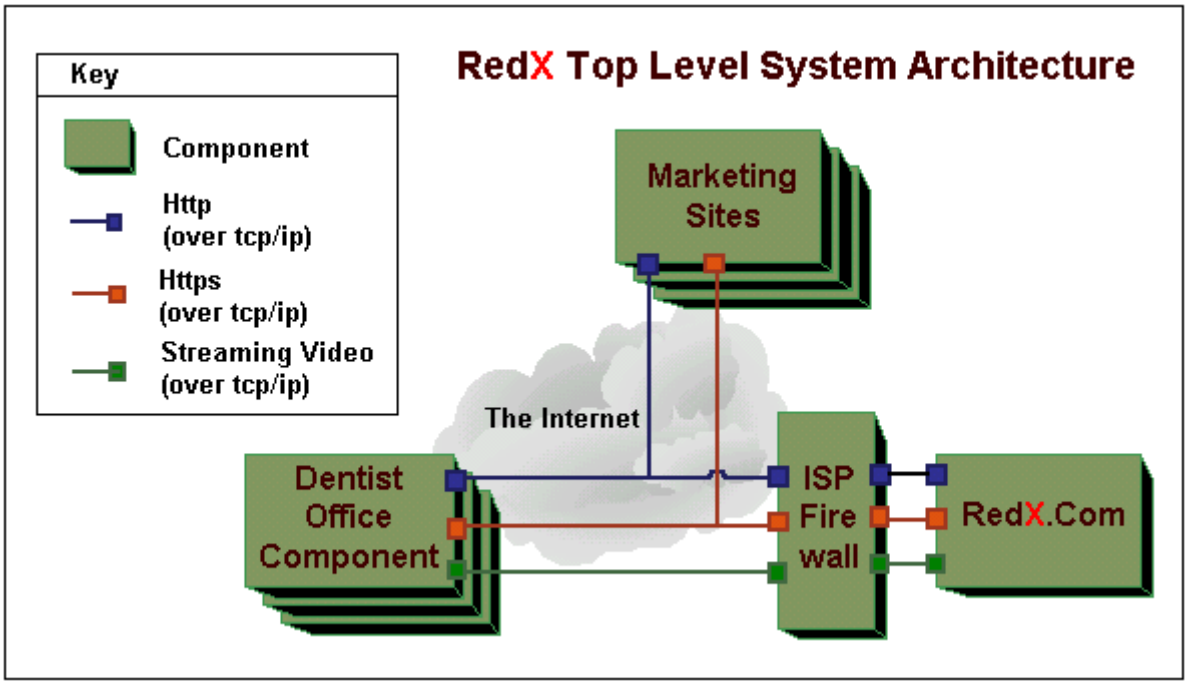


Figure 1 - RedX Major Components

The following components represent a client-server style of computing, not only from the high level view of Internet browser clients communicating with RedX, but also from the view of services provided inside the system.

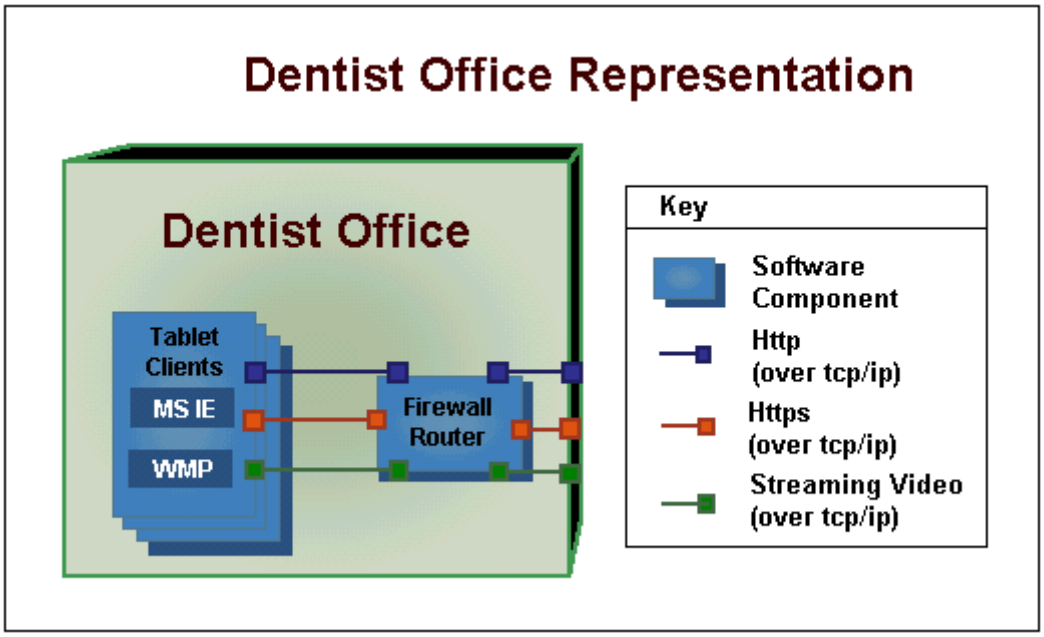


Figure 2 - RedX Major Components

RedX.com Infrastructure Components

The RedX application is composed from the following COTS components. The details of the RedX.com components, and connectors are represented in Figure 3.

Web Server – This component handles all http[s] requests from clients, communicating with the application server through Inter-Process Communication (IPC).

Application Server – This component is the "brains" of the application, handling requests from the web server and generating dynamic web pages based upon data in the RDBMS and business rules.

RDBMS – The Relational Database Management System (Oracle 9i) will store subscriber and medical information.

Authentication Server – Running an open protocol such as LDAP and/or Kerberos, the authentication server will authenticate every transaction, talking to the application server. In addition, this server logs each transaction.

Streaming Video Server – Microsoft Streaming Video Server will be used to buffer streaming video files from the file system (not shown). TabletPCs will connect to the server on TCP ports 7000 and 7001, the default for the Windows Media Player used on the tablet.

SMTP Server– Used for marketing e-mails; mostly outbound.

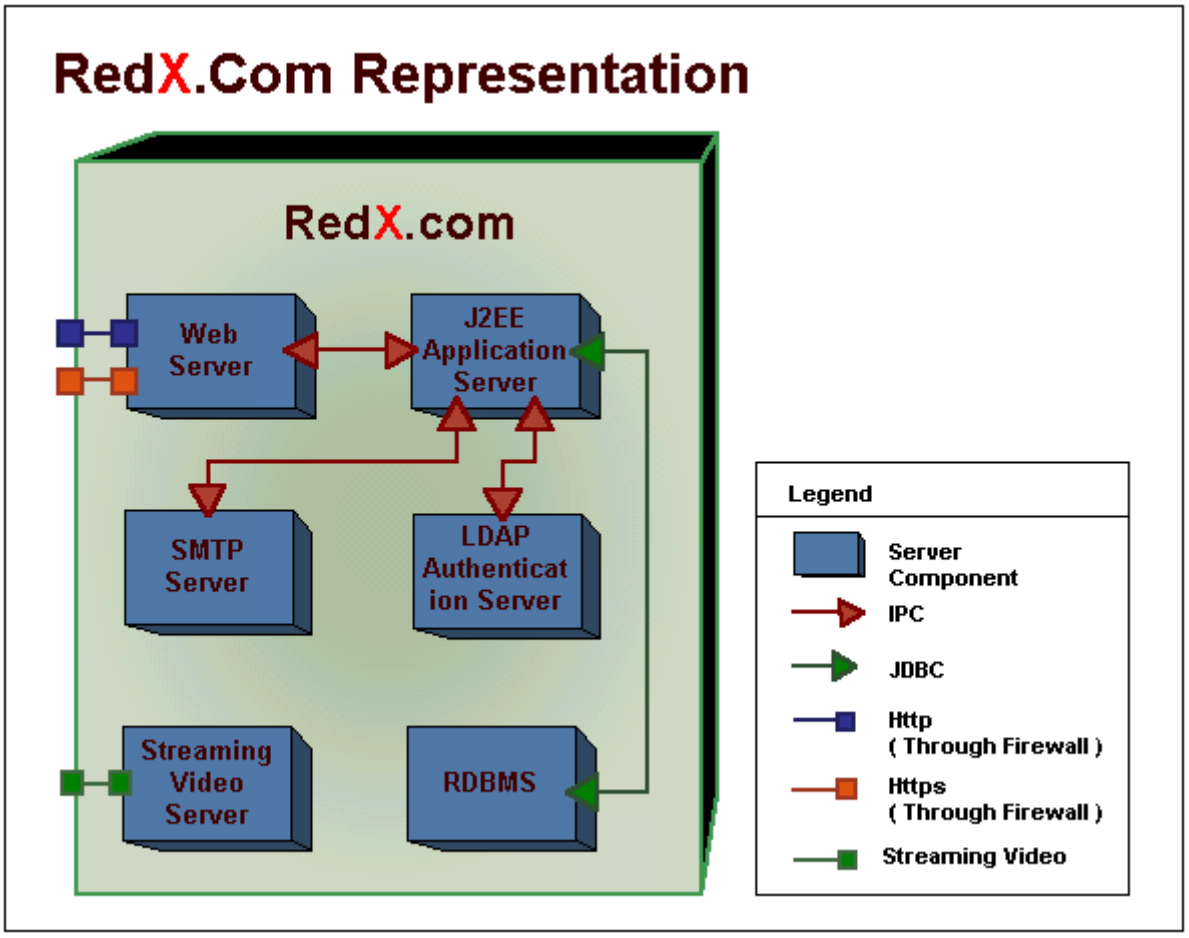


Figure 3, RedX Infrastructure Components

Application Components

Residing inside the J2EE application server are the components responsible for most of the application's functionality. These components are shown in Figure 4, and described below. Generally, components communicate within the application's EJB container via inter-process communication. Some application servers run multiple beans in the same process space with different threads.

The component's purpose is as follows:

- Ad Matching Service – Determine which ad to show to a user.
- Data Services – Retrieve from and save data to RDBMS via JDBC. - Part of EJB container
- Java Server Pages – Render dynamic web pages.
- Authentication Service – Interface with the LDAP server to provide authentication and authorization services for each transaction.

- Request Manager – Remote and/or Local interfaces (beans) that route request to and from the EJBs.

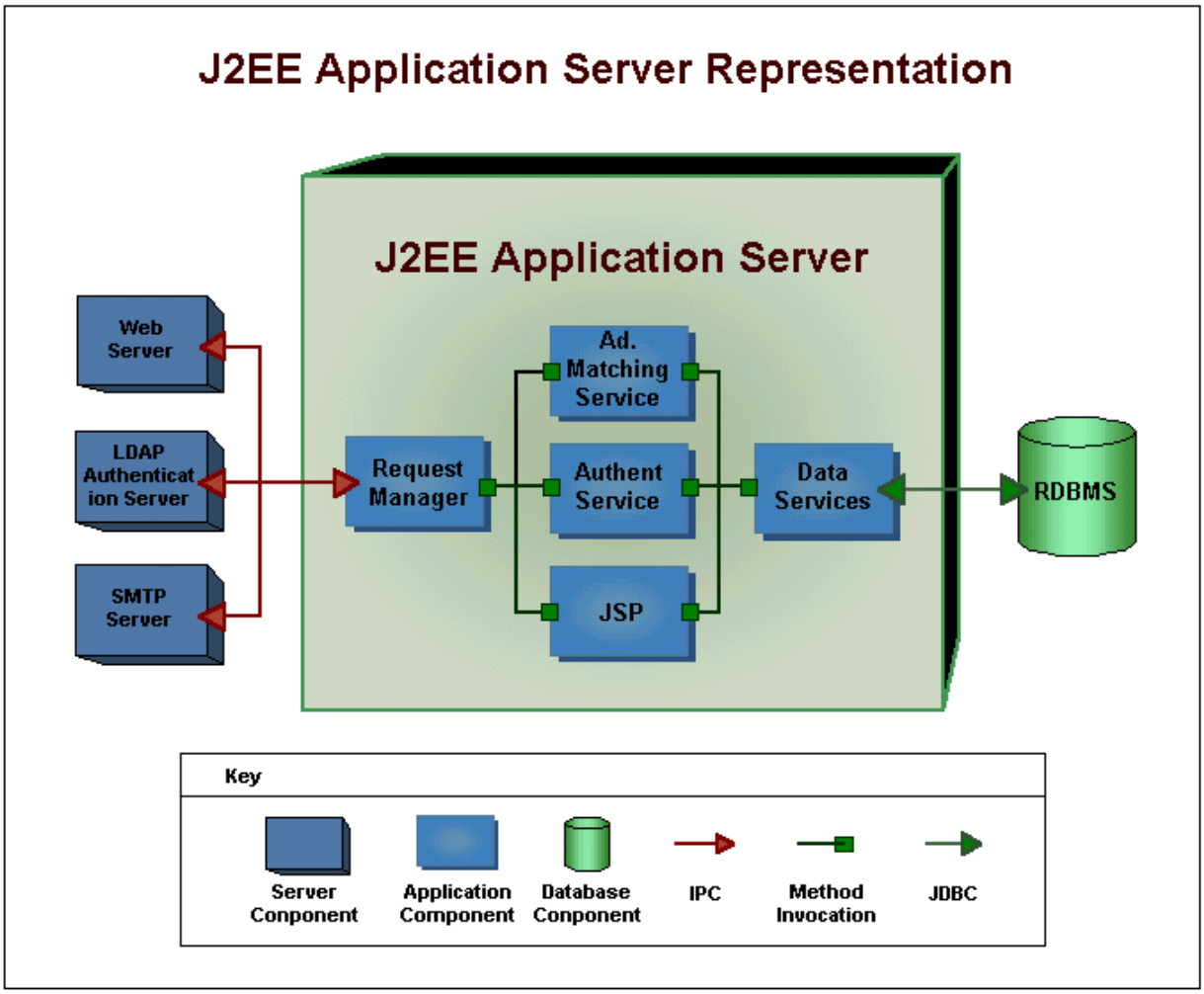


Figure 4, Application Components

ATAM Analysis

The approach of this analysis was to identify a scenario, which supported a specific quality attribute. Then identify the architectural decisions (or possible decisions) that would support this quality attribute. With these architectural decisions, we would then identify the potential risk, the tradeoffs, and sensitivity points. Other possible solutions we also considered, and their impacts we assessed.

Analysis #1 – Modifiability

The first analysis focused on Modifiability as a quality attribute. The following scenario was used to analyze the architectural tradeoffs, risk, and identify sensitivity points.

RedX client could be tablet pc, standard pc, windows CE and even palm. There are many different TabletPC in the future market. Redx software can run on all these platforms.	
Quality Attribute	Modifiability & Extensibility
Source	System User
Stimulus	Support multiple platforms
Context	We do not know which Tablet PC the RedX will choose and Redx may have to accept the existing platform in the dental office.
Artifact	Application and User interface Design
Response	Software can run on multiple platforms.
Response Measure	The number of platform that RedX client support.

The main architectural component that is involved in this analysis is the Redx client. The user interface options were to create own client or use a browser. If we create our own client we would ultimately have can have more control over the user interactions on the tablet. We can cache the video and reduce the bandwidth requirement, we can cache user information when network down, and resend it after network up. The major challenge with this approach was the lack of resources to develop this fat client. This also would affect other quality attributes like maintainability of the client software. On the other hand, if we choose a browser as our client, then any operating system supporting HTTP could be our client. So based on the resource limitations, and the customer's request for low maintenance cost for the system, we opted for the thin client approach.

ATAM Discussion item	1
Quality Attribute	Modifiability
Key factor	Resilient to change
Architectural decisions: • Uses a thin client as opposed to a fat client.	

Risk: • Server loading is a concern • The tablet is required to run specific applications to support streaming video, and other types of media
Tradeoffs: • User interface cannot be as rich. Limited to using HTML for user interface • Performance • Use of beans allows the separation from business logic, and GUI
Sensitivity point: • Number of clients accessing the web site simultaneously

Analysis #2 – Security, Authorization of access to confidential data.

This analysis evaluates the scenario that pertains to making sure the personal information of a user is secure. Unauthorized users cannot have the ability to view or modify any of the confidential user information stored in the RDBMS. Unauthorized users could be in the form of hackers, or just casual web users browsing the RedX.com website. The most likely "casual web user" that would be a threat to security would be someone in the dentist office that picks up a tablet that another patient has set down instead of returning it to the clerk. Because of this distinction of unauthorized users it was discussed weather to discuss them separately or together. We agreed to analyze them together. This decision was based on the thought that decisions made to disallow access of the one hacker, would enhance the ability to disallow access of the casual web user. And the reverse would also be true.

The scenarios for this analysis include the following 2:

Hacker intercepts system communication. He should not be able to gain access to MI.	
Quality Attribute	Security
Source	Hacker.
Stimulus	Hacker manages to intercept transmission of data.
Context	The robustness of the transmission protocol.
Artifact	RedX system components.
Response	The hacker receives encrypted data that is extremely hard to decrypt.
Response Measure	Data is transmitted over secure channels.

The tablet is setup and another user (unauthenticated) gets the tablet. The user is unable to modify or view the other user's medical info.	
Quality Attribute	Security
Source	Unauthenticated user
Stimulus	Acquires tablet
Context	In the waiting room, Person #1 picks up the tablet after Person#2 is done. Person#1 has completed the information sheet, and their information has been

	accepted.
Artifact	User information (general, and medical information)
Response	Person #1 information not available to
Response Measure	Person #2 cannot see the personal/medical information for Person #1

The level of encryption was discussed in great detail. Encryption was identified to have a tradeoff with performance. As the level of encryption was increased, the performance of the system would be negatively impacted. The other architectural decision to support security was to expire web pages after the medical information of the authenticated user has been accepted. This would essentially make the system useable by anyone after the personal information is saved in the RDBMS, and the security of the information of the authenticated user would be greatly enhanced. From this point of view we discussed the various ways to do this, and the final decision was to not just expire a web page, but to expire a session bean that is used to connect an authenticated user to the RDBMS. The idea of expiring this session bean was to mitigate the scenario where an authenticated user places the tablet down prior to submitting their information. If the session bean did not expire, then this connection could theoretically be open for an infinite amount of time. During this time the confidential information for an authenticated user is vulnerable to unauthorized access. Expiring a session bean will set a finite amount of time for which this connection can be open. The session bean timeout become another sensitivity point in this level of security. The discussions and decisions around this topic can be summarized in the following table.

ATAM Discussion item	3
Quality Attribute	Security
Key factor	Authorization:
Architectural decisions: • Use 128 bit encryption • Expire data pages after data is submitted. –Restricts the use of the browsers back button • Can expire a session bean during data entry, if session is too long • Have screen saver lock after 10 minutes of inactivity	
Risk: • Encryption negatively impacts performance	
Tradeoffs: • Level of encryption will impact performance (56,128, or 256 bit encryption available) • Restrict encryption to only those pages where medical and personal information is collected. • Screen lock effects all operations, not just the web pages where security is a concern	
Sensitivity point: • Encryption level • Identify specific web pages that need encryption • Session bean timeout duration	

This analysis also identified an area that would also need a specific security analysis. The discussion was focused around whether to use a customized EJB, or LDAP to

authenticate users. The use of EJB would utilize the user information that exist in the RDBMS, and would eliminate the duplication of user information that would be needed for a LDAP. LDAP is a standard method used to authenticate web site users. At the time of creating this document, the analysis of between these two methods is continuing. The result of this analysis has caused someone on the team to produce a small prototype that uses both methods. No concrete decision has been made at this point in time.

ATAM Discussion item	4
Quality Attribute	Security
Key factor	Authentication
Architectural decisions: • Use LDAP, EJB services, database security, or a custom component to store information and control access of authorized users	
Risk: • The authentication services provided by EJB or the web server could not be strong enough – may require development • A custom component has to be developed, needs to be maintained, and may have security flaws • The database could become a bottleneck, requires creation of users, requires maintenance and is another "hole" into the system. LDAP is standard, EJB needs to be developed	
Tradeoffs: • LDAP is a standard, included in Microsoft Server 2000, provides scalability because it can be used by several web servers, but it cannot be customized and requires an additional DB for authentication • EJB containers may provide the needed authentication services but the EJB needs to be defined and developed, there is an increased development cost, future developers must know hold EJB knowledge, deployment offers more challenges then simply using LDAP. • With the database there is a concern with scalability. Must interface with the RedX serve, and add additional field(s) to the RedX DB. Would add another Database connection to RedX DB that would allow another access point for potential hackers. • A custom component has to be developed, needs to be maintained, and may have security flaws, but it provides more control.	
Sensitivity point: • Desired level of control	

Analysis #3 – More Security. Authentication of access to confidential data.

This analysis also supports the scenario of someone trying that is trying to hack into the system from the Internet. The intention of the hacker would be to steal, corrupt, or somehow disrupt the integrity of the data in the RDBMS. This analysis has been separated from the other security analysis because the tradeoffs of using browser certificates are contained within how the certificates are used. The high-level tradeoff view would be "use browser certificates, or don't use browser certificates". The use of certificates has not yet been decided. This analysis will offer some insight into if certificates are used how they will impact the architecture, and what are ways to use certificates.

ATAM Discussion item	5
Quality Attribute	Security
Key factor	Authentication
Architectural decisions: • Use browser certificates	

Risk:
• Hacking:
  o By not using certificates, RedX.com is available for everyone on the Internet to access. This allows hackers the ability to see the front door of the web site, and possible entice them to try and hack into the system.
  o By using certificates, only those browsers that have valid certificates will be able to see the front door of the web site.	

Tradeoffs: • RedX would have to become a certificate authority. • Certificates expire, and need to be renewed • Certificates associated with user profile when users log into windows. This could annoy the dental office, but add a level of security to the tablet • Common browsing of the RedX site from a 'home pc' would not be allowed • Updating certificate is a manual process, and involves user intervention. This would require a limited amount of PC knowledge by the office administrators. • Improves security, because browsers that do not have certificates will not even be able to see the main web site page.	
Sensitivity point: •	

ATAM Evaluation

The ATAM process offered insight into potential problem areas that would not normally be caught during in normal design inspection. The biggest challenge that confronted our team was identifying how to capture and record the information that was being discussed. The ATAM was broken into “ATAM discussion points”. These discussion points would evaluate a scenario, and the architectural sensitive decision that were necessary to realize this scenario. After we got into the second ATAM discussion point we realized a structured approach to this evaluation would help the flow of the meeting. This structured approach helped in the creation, and understanding of the notes from the meeting. Identifying the sensitivity point was a little confusing in this process. During the ATAM meeting we thought the sensitivity point was the ‘part’ of the product that could be adjusted to achieve various degrees of conformance to the quality attribute. Later we discovered that the sensitivity point should identify the other quality attributes that are sensitive to changes in this ATAM discussion.

Conclusion

This team found it difficult to identify the first scenarios and how to properly describe a scenario. Initially, we would state a scenario and try to justify its validity. Once we realized that scenario identification is basically a brainstorming session, the scenarios sessions were more effective. After identifying a significant number of scenarios we ranked them, and then clarified the top 20 to conform to the general scenario template. We found this activity was best conducted in a group setting rather than individuals creating scenarios and then merging all of the individual work. The group setting fostered learning about architectural sensitive issues and what is important when trying to reason about the architecture.

We found the ATAM analysis to be helpful in identifying parts of the system that may have potential interface problems. It provided a perspective to evaluate the ‘glue’ of a system rather than focusing on the individual components in isolation. This is a major problem with most software systems, so identifying the interface problems early in the life cycle greatly increases the probability of success.

Appendix

Detailed Scenarios

Scenarios are used to determine whether the architecture meets specific quality goals. A scenario is a short statement describing interaction of one stakeholder with the system. They represent specific examples of current and future uses of a system, and are useful to convey an understand run-time qualities. The target audience for scenarios includes users, maintainers, developers, and clients.

Based on the business goals, requirements and the quality needs, a utility tree is derived. Here we are using the Architecture Tradeoff Analysis Method (ATAM) to generate the utility tree. Since we are using the utility tree as an aid to converge to an architectural design, the major driving forces for generation of the utility tree are the business goals and the requirements. Based on the high priority quality attributes extracted from the utility tree, system architecture will be accomplished. Each one of these scenarios has been ranked on a scale of 0-25 on their level of importance. Their score is included for reference.

Security

19	
A user should not be able to log in to the system without the administrator's password.	
Quality Attribute	Security
Source	Customer, secretary, end user.
Stimulus	An end user in waiting room obtains a tablet and tries to log in without the administrator's password.
Context	The robustness of the authorization algorithm and encryption method for administrator password.
Artifact	The profiling database for the administrator password, the encryption method for data in database, and the authorization algorithm such as number failure allowed.
Response	The end user shall receive the warning that he does not have right to log in to the system and can't log in to the system.
Response Measure	Number of unauthorized accesses to the system per month.
Ranking score	16

28	
Hacker intercepts system communication. He should not be able to gain access to MI.	
Quality Attribute	Security
Source	Hacker.

Stimulus	Hacker manages to intercept transmission of data.
Context	The robustness of the transmission protocol.
Artifact	RedX system components.
Response	The hacker receives encrypted data that is extremely hard to decrypt.
Response Measure	Data is transmitted over secure channels.
Ranking score	13

25	
A user shall be able to surf web sites that comply with the surfing policy of the dentist office only.	
Quality Attribute	Security
Source	The customer, dentist.
Stimulus	A user attempts to surf to a web site that does not comply with the surfing policy of the dentist office.
Context	The user is informed about the dentist office policy. The system has monitoring process to monitor usage of the user.
Artifact	The dentist office policy, monitoring process.
Response	The user gets warning message and is not able to visit that web site. The user will be forced to go back to REDX website.
Response Measure	Number of accesses to the web site that is not complies with the dentist office policy and the mean time between surfing that web site.
Ranking score	14

31	
The data base of medical information can't be compromised.	
Quality Attribute	Security
Source	Customer, user, dentist.
Stimulus	User hacks the ISP hosting RedX.com. The user tries to compromise the database of medical information.
Context	The database authenticate the user before allow him to access the data. The Data in database is encrypted.
Artifact	The database authorization process and encryption algorithm.
Response	The user can't access the data in the database.
Response Measure	Number of unauthorized accesses per month. Number of records that are compromised.
Ranking score	11

6

The tablet is setup and another user (unauthenticated) gets the tablet. The user is unable to modify or view the other user's medical info.	
Quality Attribute	Security
Source	Unauthenticated user
Stimulus	Acquires tablet
Context	In the waiting room, Person #1 picks up the tablet after Person#2 is done. Person#1 has completed the information sheet, and their information has been accepted.
Artifact	User information (general, and medical information)
Response	Person #1 information not available to
Response Measure	Person #2 cannot see the personal/medical information for Person #1
Ranking score	20

16	
The user attempts to exit the RedX client. User is not able to exit out of the RedX system.	
Quality Attribute	Security
Source	Users (Patients)
Stimulus	User attempts to exit the RedX front-end and access the tablet's operating system.
Context	
Artifact	Tablet, RedX client.
Response	User should not be able to exit out of the RedX client.
Response Measure	Testing of this scenario should reveal no way to exit out of the RedX client without proper permission/privileges.
Ranking score	18

Usability

4	
User is filling out a form. When he hits the submit button, fields that are not mandatory are left blank. User will be notified that the fields are blank and will have the option to either update those blank fields or continue.	
Quality Attribute	Usability
Source	Users (Patients)
Stimulus	User hits the submit button on the form that is being filled out.
Context	RedX system determines which fields are mandatory and which are not. It also verifies the submitted data.
Artifact	Thin client, RedX system.
Response	User is notified that fields are blank and has option to either fill out blank fields or continue without filling them out.
Response	System checks submitted data again.

Measure	
Ranking score	21

3	
User is filling out a form. When he hits the submit button, mandatory fields are blank. User will be prompted to complete the required blank fields.	
Quality Attribute	Usability
Source	Users (Patients)
Stimulus	User hits the submit button on the form that is being filled out.
Context	RedX system determines which fields are mandatory and which are not. It also verifies the submitted data.
Artifact	Thin client, RedX system.
Response	User is notified that fields are blank and MUST fill out blank fields before continuing.
Response Measure	User will receive feedback within 3 seconds
Ranking score	21

18	
A patient completes filling out his information form and a confirmation page appears, prompting the user to review his information before it gets submitted to the system.	
Quality Attribute	Usability
Source	Authenticated user
Stimulus	User Providing information to the Tablet PC system
Context	Some of the 'mandatory' information is left blank when the user submits their information on the tablet
Artifact	User information repository
Response	Person should not be able to progress further until the mandatory information is provided
Response Measure	No other screens or options available to the user, other than the option to complete the mandatory fields.
Ranking score	16

17	
Patient enters, is given a tablet to fill out information. Time elapsed from when they enter until they are viewing ads is <= 10min.	
Quality Attribute	Usability
Source	Users (Patients)
Stimulus	too many ads will annoy users
Context	Show advertisement
Artifact	Show ads JSP
Response	Calculate the time required by each advertisements
Response	Maximal time required by advertisement

Measure	
Ranking score	17

Performance

13	
The system can show the advertisement to the user within 20 s.	
Quality Attribute	Performance
Source	Customer, user.
Stimulus	A patient completes forms and submits the form. The patient will not wait more than more than 20 sec to see the advertisement.
Context	Network latency between a user and the advertising server. The performance of the database machine.
Artifact	Advertisement server, database server, type of the advertisement.
Response	An advertisement should appear to patient in < 20S.
Response Measure	The mean time before the advertisement appears.
Ranking score	18

Reliability

7	
The system can support concurrent usage.	
Quality Attribute	Performance
Source	Customer, end user.
Stimulus	Three patients are filling MI on separate PCs and posting at the same time.
Context	The ability to receive concurrent transaction of the database server.
Artifact	Database engine, advertisement server.
Response	Correct ads will be displayed on all 3 tablets at the same time.
Response Measure	Number of the concurrent user.
Ranking score	20

2	
A user logs into the tablet and views ads according to the ad-matching algorithm provided that the remote RedX component is available.	
Quality Attribute	Reliability
Source	Users (Patients)
Stimulus	User reaches a page where one or more ads are meant to

	appear.
Context	Availability of the remote RedX component.
Artifact	Thin client, remote RedX component and network.
Response	Proper ad is displayed, according to ad-matching algorithm.
Response Measure	Review of usage logs does not reveal improper system behavior.
Ranking score	21

20	
The dentist's connection to the ISP fails. Medical information can only be sent to the printer.	
Quality Attribute	Reliability
Source	Authenticated user
Stimulus	
Context	User is logged into the system, and is actively entering information in the Tablet. The screen the user is interfacing with is the screen that contains their medical information.
Artifact	Medical information in the RedX repository, and medical information the user sees on the tablet screen
Response	Medical information cannot be sent to the remote RedX server. User can only send them to the printer.
Response Measure	Data can be printed.
Ranking score	16

26	
A failure occurs. The integrity of the remote data store must remain intact.	
Quality Attribute	Reliability
Source	Database, locally cached information.
Stimulus	A failure occurs in the system.
Context	
Artifact	Network, individual system components.
Response	Upon recovering from failure, database is in a consistent state and holds the latest, correct information.
Response Measure	Reviews of the data and log audits should not reveal any inconsistencies. Proper data is always displayed to the user.
Ranking score	13

Modifiability & Extensibility

12

When modifying the profiling information of an advertisement. This modification is restricted to the available fields that already exist, and involves selecting new valid range for these fields. This modification should take a developer no longer than 4 hours to complete.	
Quality Attribute	Modifiability & Extensibility
Source	Developer, customer, REDX system administrator
Stimulus	A request to change existing selection criteria for an advertisement.
Context	A developer familiar with the system, and familiar with oracle will make the modification
Artifact	The profiling database. The database structure will not change, only some of the field information.
Response	The scenario can be implemented and tested.
Response Measure	The time frame to complete the scenario is 4 hours.
Ranking score	18

5	
RedX is adding a new advertisement to the system. This must be completed in 5 minutes.	
Quality Attribute	Modifiability & Extensibility
Source	System administrator
Stimulus	Save the time for administrator
Context	Administrator will add many new advertisements into the system everyday.
Artifact	DB design and User interface Design
Response	It is easy to add new advertisement, new advertisement vendor and the profiles
Response Measure	Average time to add new advertisement based on different scenarios
Ranking score	20

9	
Add a new selection criterion to the advertisement profile within 2 weeks.	
Quality Attribute	Modifiability & Extensibility
Source	System administrator
Stimulus	The customer requests a change to the advertisement profiling schema.
Context	The new selection criterion is similar to the profiling information already being used.
Artifact	The database, and server software will be modified
Response	The scenario can be implemented and tested.
Response Measure	The time frame to complete the scenario is 2 weeks.
Ranking score	19

22	
RedX wishes to use Double-Click as its advertisement provider. System can be enhanced in 80 man-hours.	
Quality Attribute	Modifiability & Extensibility
Source	System administrator
Stimulus	Double-Click is a major player in online advertisement market
Context	Double-Click may provide us some new methods for advertising
Artifact	DB design, middle ware design, Interface design
Response	System should be more loosely coupled
Response Measure	Time we spend to integrate Double Click
Ranking score	15

12	
RedX wishes to add a new type of advertisement media. This can be done in 12 Man-hours	
Quality Attribute	Modifiability & Extensibility
Source	RedX Administrator
Stimulus	A request to add a new type of advertisement (streaming video, banner…)
Context	A developer familiar with the system. Information on the advertisement type can be found at www.iab.net
Artifact	The database that contains the advertisement information.
Response	A new advertisement type can be added
Response Measure	The time frame to complete the scenario is 12 Man-hours.
Ranking score	18

24	
RedX developers wish to modify the ad-matching algorithm. This can be done in 40 man-hours.	
Quality Attribute	Modifiability & Extensibility
Source	Redx owners, or developers
Stimulus	New and improved ad matching algorithm is thought of.
Context	A developer familiar with the system. Advertisement matching algorithm is base lined.
Artifact	Ad-matching algorithm definition, and software that implements the definition
Response	A new ad-matching algorithm exist
Response Measure	The task can be completed in 40 man hours
Ranking score	14

References

[FRED-SRS] Software Requirements Specification, MSE Studio, FRED – 2002 (http://dogbert.mse.cs.cmu.edu/mse2002/projects/TabletPC/docs/requirements/srs.doc)

[IEEE-1061] IEEE STD 1061-1992. *Standard for a Software Quality Metrics Methodology.* New York: Institute of Electrical and Electronics Engineers, 1992.

[GR93] Jim Gray, Andreas Reuter. Transaction Processing: Concepts and Technologies. Morgan Kaufmann Publishers. 1993. pp. 7-18.

[K+98] Kazman, Klein, Barbacci, Longstaff, Lipson, Carriere. The Architecture Tradeoff Analysis Method. Software Engineering Institute Technical Report, CMU/SEI-98-TR-008, July 1998.