Gynaecological Patient Information management System:

Architectural Requirements

Team Pentec:

Ruth Ojo 12042804
Liz Joseph 10075268
Trevor Austin 11310856
Maria Qumayo 29461775
Lindelo Mapumulo 12002862

Final Version

July 30, 2015

Contents

1 Introduction

2 Architectural requirements

2.1 Architectural Scope

This section discusses the boundaries and extent or range of view, outlook, applications, operations or effectiveness the system software architectur needs to address. More spesificaly we will be looking at the following three topics in depth. Persistence, Reporting and Process execution.

2.1.1 Persistence

Buzz Space needs to be percistant in order for states to be stored and outlive the many processes done. It will do so by making use of the following:

- Databases, to store and retrieve user accounts data and information.
- Java Data Objects (JDO), a specification of Java object persistence. With it's great transparency feature of the persistence services to the domain model.
- System prevalence, a technique that joins system images and transaction journals to achieve percistancey.

2.1.2 Reporting

It is crucial that the Buzz Space gives the user some sort of reporting and feed back after activities such as successful posts, voting(either up or down), deleted or hidden posts, read and unread posts or any other change of state. This will be achieved by sending the user an email to notify him of any changes.

2.1.3 Process execution

Process infrastructure and execution deals with the essential operation components, such as policies, processes, equipment, data and internal operations, for overall effectiveness. In providing a good Process infrastructure for the Buzz Space system, we aim to:

- Reduce duplication of effort
- Ensure adherence to standards (both coding and design)

- Enhance the flow of information throughout an the system
- Promote adaptability necessary for a changeable environment (Including the different human access channels.)
- Ensure interoperability among organizational and external entities
- Maintain effectiveness

2.2 Access and integration requirements

2.2.1 Human access channels

2.2.2 System access channels

2.2.3 Integration channels

2.3 Quality requirements

2.3.1 Security

2.3.2 Usability

2.3.3 Scalability

2.3.4 Reliability and Availability

2.3.5 Integrability

Description:

The Patient Information Management System should be integrated with ease to the Universities website as well as the NHSA web site.

Justification:

The Patient Information Management System is primarily designed for the kalafong medical staff and is required to be accessable on both the University's website and the NHSA website. For security perposes we limit it's intergaration to just these two sites.

Mechanism:

1. Strategy:
   The contracts based decoupling strategy is used to implement integrability and extensibility on the system by providing sections and different functionality between the back end system and the host site.

2.3.6 Maintainability

2.3.7 Monitorability

2.3.8 Testability

2.3.9 Performance

3 Architectural Patterns and Styles

3.1 Model-View-Controller (MVC)

Model-View-Controller is an architectural pattern that divides software applications into three interconnected parts.

The controller is responsible for translating requests in to responses (Nadel, 2012). The controller 'inserts' the response back into the view, which could be html or it's equivalents.

The view translates the response, from the controller, into a visual format for the client (Nadel, 2012). The view also sends requests to the controller.

The model's task is to maintain state and give methods for changing this state. Typically, this layer can be decomposed to:

- Service layer - provides high-level logic for dependent parts of an application.
- Data access layer - services objects invoke this layer, which provides provides access to the persistence layer.
- Value object layer - provides data-oriented representations of terminal nodes in the model hierarchy.

3.2 Technologies

input./Technologies.tex

3.3 Tactics and strategies

4 Architectural responsibilities

5 Architecture constraints