**Software Design Document, Testing, Deployment and Configuration Management, And User Manual of the ASKAMECH**

**115392 Project Approach**



**APPLY PRINCIPLES OF CREATING COMPUTER SOFTWARE BY DEVELOPING A COMPLETE PROGRAMME TO MEET GIVEN BUSINESS SPECIFICATIONS**

**Table of contents**

**Team Members**

|  |  |
| --- | --- |
| **Surname** | **Name** |
| **Mathe** | Khanyisile |
| **Zungu** | Samkelo |
| **Ndaba** | Lindokuhle |

1. **Introduction**

This introduction provides an overview of the System Architecture Document for AskAMech. It includes the purpose, scope, target audience, design approach, main component design and high-level system design considerations of the system.

**Document scope and Purpose**

This document provides a description of the technical design for AskAMech – Q&A Forum. This document’s primary purpose is to describe the technical vision for how business requirements will be realized. This document provides an architectural overview of the system to depict different aspects of the system. This document also functions as a foundational reference point for developers.

Please note that this is a baseline document and may be updated as development progresses.

**Target Audience**

This document is targeted (but not limited) to technical stakeholders:

* Development Team
* IT Management
* Support Staff

It is assumed that the reader has a technical background in software design and development.

**Acronyms/Abbreviations**

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
|  |  |
|  |  |

**Reference Documents**

* System requirement document of AskAMech
* Feasibility Study

**System Environment**

* Development: Visual Studio 2017 + Dotnet 2.2
* Unit Test: NUnit
* Diagrams: Visio 2016 /Draw.IO
* Database Management: SQL Server Management:
* Database: SQL

**Design Approach**

The design approach used here is based on the following:

**Data Flow Design**

The data flow of the ASKAMECH is web-based. Entity Framework technologies will be utilized to retrieve data from SQL database to be displayed by the Web portal user interface and would also allow updating the data where applicable.

**Architecture Design**

The application will follow a Four Layer Architecture so that the objects in the system as a whole can be organized to best separate concerns and prepare for distribution and reuse. A principal advantage to this design is the relative stability of the components as seen by the applications developer. Implementations may change considerably to enhance the performance or in response to changes in the architecture. These changes are less likely to cause major impact to the applications’ programs.

**UI Design**

Wire Frames are used for UI design. Wire frames are an effective tool for collecting and presenting functionality, navigation, and content of an application or web site. Annotations or notes attached to elements or widgets on the wire frame help to communicate specific functions.

Some Screen Shoots

**Design Patterns**

This application is designed as an object-oriented system for a web-based architecture using four-layer architecture by factoring application classes into the following layers:

**The Presentation layer:** This is the layer where the physical window and widgetobjects live. It will also contain Controller classes as in classical MVC. Any new user interface widgets developed for this application are put in this layer.

**The Domain Mode:** Most objects identified in the OO analysis and design will reside.To a great extent, the objects in this layer can be application-independent. Generic objects may be used in this application to reap the benefits of Object-Oriented programming.

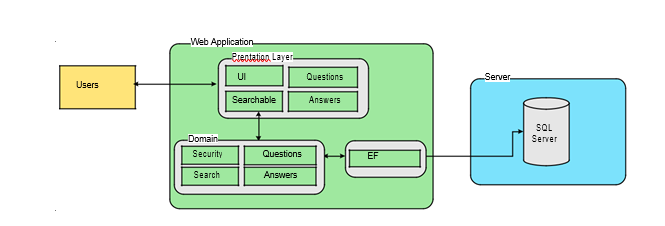
**The Command layer:** This willcontain the application login and make use of domain classes onto a wider range and will be more flexible in the creation of possibly editing of data from the infrastructure layer.

**The Infrastructure layer**: The data is managed by SQL.

In Addition;

**The Command Tests layer**: This is the layer that will consist of all the application tests that will be mainly focused but not limited to commands. To great extent this layer will use and substitute some of the implementations from the command layer and make use of Domain models as well.

**AskAMech High level View**



1. **Modules**
2. **Entity Diagram**