

Boots IT Dashboard

Technical documentation

Ajeesh Sudhakaran (UST, IND)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer |
| 22-04-2016 | 0.1 | Initial Draft | Ajeesh Sudhakaran |  |
| 27-04-2016 | 0.2 | Incorporated review comments | Ajeesh Sudhakaran | Jaison Ittoop |
| 18-05-2016 | 0.3 | Updated Architecture | Ajeesh Sudhakaran | Jaison Ittoop, Pawan Sharma |

Contents

[1. EXECUTIVE SUMMARY 4](#_Toc451460035)

[2. TERMINOLOGIES USED 4](#_Toc451460036)

[3. SCOPE 5](#_Toc451460037)

[3.1. PHASE 1 5](#_Toc451460038)

[3.1.1. WILL BE INCLUDED 6](#_Toc451460039)

[3.1.2. WILL NOT BE INCLUDED 6](#_Toc451460040)

[3.2. PHASE 2 6](#_Toc451460041)

[3.2.1. WILL BE INCLUDED 6](#_Toc451460042)

[3.2.2. WILL NOT BE INCLUDED 6](#_Toc451460043)

[3.3. PHASE 3 7](#_Toc451460044)

[3.3.1. WILL BE INCLUDED 7](#_Toc451460045)

[3.3.2. MAY BE INCLUDED 7](#_Toc451460046)

[4. APPLICATION AUDIENCE 7](#_Toc451460047)

[4.1. TARGET AUDIENCE 7](#_Toc451460048)

[4.2. TARGET REQUIREMENTS 7](#_Toc451460049)

[5. ARCHITECTURE DESIGN 7](#_Toc451460050)

[6. EQUIPMENT AND FACILITIES 8](#_Toc451460051)

[6.1. DEVELOPMENT ENVIRONMENT 8](#_Toc451460052)

[6.1.1. DEVELOPMENT MACHINES 8](#_Toc451460053)

[6.1.2. INTEGRATED DEVELOPMENT ENVIRONMENT 9](#_Toc451460054)

[6.1.3. DATABASE 9](#_Toc451460055)

[6.1.4. COLLABORATION 9](#_Toc451460056)

[6.1.5. OTHER TOOLS & TECHNOLOGIES 9](#_Toc451460057)

[6.2. HOSTING ENVIRONMENT 9](#_Toc451460058)

[6.3. COSTS AND ESTIMATES 10](#_Toc451460059)

[6.3.1. TIME ESTIMATE 10](#_Toc451460060)

[6.3.2. COST ESTIMATE 10](#_Toc451460061)

[6.4. SKILLSET REQUIREMENTS 10](#_Toc451460062)

[7. WORK ITEMS 10](#_Toc451460063)

[7.1. MONITORING LAYER 11](#_Toc451460064)

[7.1.1. SCHEDULER 11](#_Toc451460065)

[7.1.2. PLUGINS FRAMEWORK 11](#_Toc451460066)

[7.1.3. PLUGIN HELPERS 11](#_Toc451460067)

[7.1.4. BASIC PLUGINS FOR ABACUS 11](#_Toc451460068)

[7.2. SERVICE LAYER 11](#_Toc451460069)

[7.2.1. IP WHITELISTING 11](#_Toc451460070)

[7.2.2. WEB SERVICES 11](#_Toc451460071)

[7.3. DATA LAYER 11](#_Toc451460072)

[7.3.1. DATABASE DESIGN 12](#_Toc451460073)

[7.3.2. ORM AND DATABASE ENTITIES 12](#_Toc451460074)

[7.4. BUSINESS LOGIC 12](#_Toc451460075)

[7.4.1. APPLICATION AND COMPONENTS 12](#_Toc451460076)

[7.4.2. RULES 12](#_Toc451460077)

[7.4.3. CREDENTIALS 12](#_Toc451460078)

[7.4.4. NOTIFICATION & SUBSCRIPTION 12](#_Toc451460079)

[7.4.5. REPORTS 12](#_Toc451460080)

[7.4.6. LOGGING 12](#_Toc451460081)

[7.5. PRESENTATION LAYER 12](#_Toc451460082)

[7.5.1. LANDING PAGE 13](#_Toc451460083)

[7.5.2. APPLICATION DETAILS PAGE 13](#_Toc451460084)

# EXECUTIVE SUMMARY

Boots uses different IT solutions for providing better experience and satisfaction to customers. Boots Loyalty team plays a key role in this by organizing and conducting customer loyalty programs and campaigns. This is achieved through Boots Analytical Data Warehouse – ABACUS and campaigning application UNICA. These two systems are highly critical to Boots day to day business and availability of this applications is at top priority of application management team.

To ensure better IT service to business and as a first step to proactive application management, a dashboard application has been proposed for monitoring critical applications in real-time and to provide efficient notifications on critical events.

This solution is proposed to address below problems,

* Database hangs, file system issues and table space issues are not identified unless manually checked. The situation gets worse on weekends
* Delay in receiving source files causes job failures
* Business has to reach apps management team frequently to know system status on failures
* IBM Tivoli monitoring team fails to create incidents on some instances and apps management team miss the notification
* Difficult to monitor system status on/before execution of complex jobs
* Current daily monitoring is one time activity and cost more than an hour of effort

This solution is expected to bring the following benefits to Boots.

* Up to 9% reduction in Sev-1 incidents in ABACUS, Delphi & UNICA per year
* Get real-time statistics applications and services
* Business doesn't need to reach IT Support to know system status
* Efficient prediction and notification before failures
* Reusable between multiple teams and applications
* Reduce manual activities - more productivity, eliminates chances of manual errors

This document summarizes software architecture, application modules, skillsets required and estimated timeline for development of Boots IT dashboard application. Current software design and estimates covers only requirement from Boots Loyalty & Analytics team only.

# TERMINOLOGIES USED

|  |  |
| --- | --- |
| Term | Description |
| Dashboard Application | The application which monitor other systems and/or shows status information of other systems to end user |
| Client Application | The application whose status information is shown in dashboard application |
| Application Component | Components of a client application being monitored or whose information is shown in dashboard |
| Reporting Component |  |

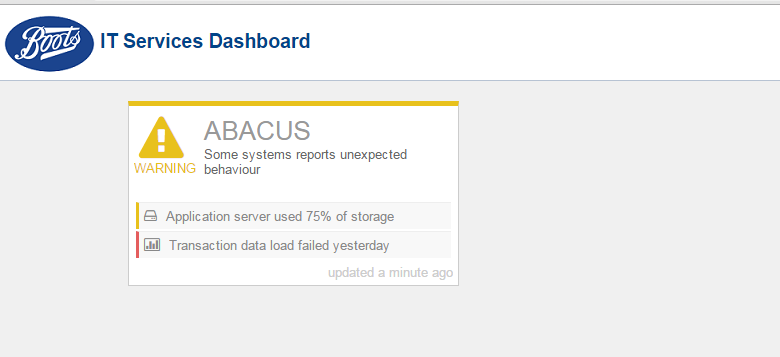
# SCOPE

Current scope and estimates of this software design covers only critical applications ABACUSB & UNICA from Boots Loyalty & Analytics team. Application modules and estimates may vary if requirements are extended beyond current scope. UAT support and implementation support is not estimated in current strategy.

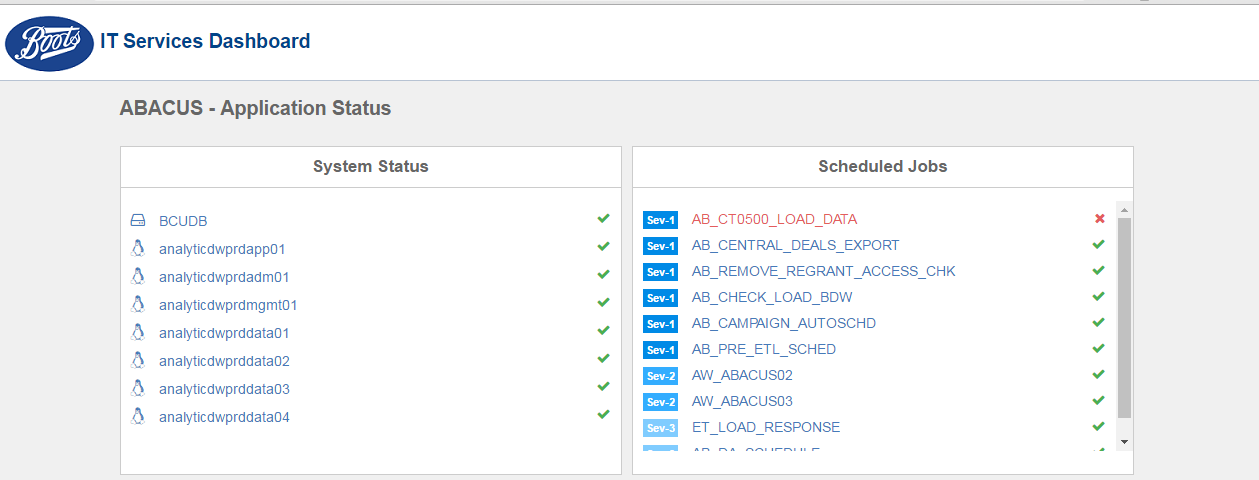
Software modules to be developed has been categorized based on priority and development activity will be carried out based on priority in three phases. Suggestions and changes will be accommodated to appropriate phases and estimates may vary according to effort required for change.

## PHASE 1

Phase 1 will deliver basic monitoring capabilities for application ABACUS. It will include monitoring of Servers, Database and critical batch jobs. ABACUS application page developed in Phase 1 will look similar to below images (user interface design is subject to change).



(Landing Page)



(ABACUS Application Detail Page)

### WILL BE INCLUDED

* Landing page listing ABACUS application status
* Application Details page showing status of Servers, databases and critical jobs
* Monitoring of Servers
* Monitoring of Database
* Monitoring of Critical Jobs

### WILL NOT BE INCLUDED

* Multiple Application Monitoring
* Email notifications
* Reports (Transaction Report, SVOC missing customers)
* DB2 Tablespace Monitoring
* UNIX Storage Space Monitoring

## PHASE 2

This phase will cover development of email notification & subscription and reports. Reports includes ABACUS transaction reports, DB2 Tablespace and UNIX Server space.

### WILL BE INCLUDED

* Multiple Application Monitoring (Unica, Delphi)
* Email Notifications & Subscription
* EPOS Transaction Report
* DB2 Tablespace (ABACUS, Delphi)
* UNIX Server space (ABACUS, Unica, Delphi)

### WILL NOT BE INCLUDED

* UNIX CPU Utilization Monitoring
* UNIX Memory Utilization Monitoring
* Windows Monitoring

## PHASE 3

This phase will cover development of features CPU and Memory monitoring for UNIX machines and customization of charts/graphs in UI. Additional features like monitoring in Windows environment can be accommodated in this phase and estimates needs to be amended accordingly.

### WILL BE INCLUDED

* Accept external feeds through Web Service
* UNIX CPU Utilization Monitoring
* UNIX Memory Utilization Monitoring
* Multiple views for charts/graphs

### MAY BE INCLUDED

* Windows Monitoring
* Changes or Modifications to existing system

# APPLICATION AUDIENCE

## TARGET AUDIENCE

Audience of this solution will be anyone working with Boots IT systems and applications. This includes business users, application managers and application management team. The dashboard might be showcased across TV screens at Boots Head office at Nottingham.

## TARGET REQUIREMENTS

Dashboard web application will be hosted in Boots Intranet and thus access to Boots Intranet will be minimum requirement for accessing the application. End user can access dashboard through a web browser. And the dashboard web application will have browser support starting from Internet Explorer 7.

# ARCHITECTURE DESIGN

Dashboard application development relies on Microsoft Technologies. Solution will contain 5 software layers – Monitoring Layer, Service Layer, Data Layer, Business Layer and Presentation Layer.

A Business Layer will provide core functionalities which can be used across other layers and has a Data Layer underneath which contains a database to store monitoring/configuration information and functionality to interact with database.

Monitoring Layer can have a generic monitoring service implemented using Windows Service and an In-House Monitoring developed and maintained at client application to be monitored. Both In-House monitoring and Generic monitoring feeds information to dashboard via a Service Layer implemented using RESTful Web Service.

In-House monitoring gives flexibility to develop monitoring logic using any available technology and to implement complex monitoring logic. Where client application is responsible to perform monitoring and to feed data to dashboard in periodic intervals.

Generic monitoring using Windows Service provides scalability by giving capability to run multiple instances shared across multiple machines. And it is flexible to accommodate changes by developing and installing Plugins which contains monitoring logic and which is decoupled from Windows Service. Windows Service will be running a scheduler to perform monitoring in periodic intervals by using configuration information from database and functionalities provided by plugin.

Presentation Layer will be implemented using ASP.Net MVC Web Application and is responsible to present information as meaningful data to end user.



# EQUIPMENT AND FACILITIES

## DEVELOPMENT ENVIRONMENT

Development of solution requires necessary systems, tools and technologies to be made available. And number of systems and tools required will be proportional to number of developers involved in development process. Requirements are listed below.

|  |  |
| --- | --- |
| Requirement | Specification |
| Development Machine | Windows 7 SP1 or later |
| IDE | Microsoft Visual Studio 2015 Professional |
| Database | SQL Server 2012 Express |
| Collaboration Tool | Git/SVN/TFS |
| Other tools & Technologies | * ASP.Net MVC 4 * Entity Framework 6 * [JQPlot](http://www.jqplot.com) * [FontAwesome 3.2.1](http://fortawesome.github.io/Font-Awesome/3.2.1/) * [jQuery 1.9](https://jquery.com) |

### DEVELOPMENT MACHINES

Each developer involved in development process must be equipped with a computer/virtual machine hosted in Boots network. These development machines should have access to test/pre-prod environments of Boots systems, which comes under scope of dashboard application. This is required for the purpose of testing and debugging the code while development.

The computer/virtual machine should meet below requirements.

|  |  |
| --- | --- |
|  | Specification |
| Operating System | Windows 7 Service Pack 1 or later |
| CPU | 1.6 GHz or faster |
| Memory | 4 GB |
| Storage | 15 GB free hard disk space after all software installations |
| Network | Access to Boots Network |

### INTEGRATED DEVELOPMENT ENVIRONMENT

Microsoft Visual studio will be used as the IDE for software development. Visual studio might require additional tools/plugins installed separately to use specific technologies like ASP.Net MVC and Entity Framework in development.

Recommended version and edition of Visual Studio is 2015 Professional. Below features/tools should be enabled in Visual Studio.

* ASP.Net MVC 4
* Entity Framework 6

### DATABASE

Development requires a common database system accessible to all developers. This ensures seamless development experience. Recommended database technology is Microsoft SQL Server 2012 Express. One or more user accounts with necessary permission in the database should also be made available.

### COLLABORATION

A software versioning and revision control system similar to SVN/Git/TFS is required and each developer requires individual user account with read/write permission to repository.

### OTHER TOOLS & TECHNOLOGIES

* ASP.Net MVC: Web application framework - will be used for developing web component of the solution
* Entity Framework: Object-relational mapper - will be used for developing data access layer of the solution.
* jQuery: JavaScript framework – will be used for developing frond end components.
* jQPlot: Charting library – will be used for rendering charts, graphs in frond end.
* FontAwesome : Font and icon toolkit – will be used in frond end application

## HOSTING ENVIRONMENT

This includes server and infrastructure requirements to host and run application’s production environment.

|  |  |
| --- | --- |
| Requirement | Specification |
| Server | Windows 2008 R2 |
| .Net Framework | .Net 4.0 |
| Database | SQL Server 2012 Express |
| Web Server | IIS 7.5 |

## COSTS AND ESTIMATES

### TIME ESTIMATE

TODO: (Attach excel file)

### COST ESTIMATE

|  |  |  |
| --- | --- | --- |
| Item | Units required | Cost per unit |
| Development | | |
| Windows Virtual Machine | 1 per developer |  |
| Visual Studio 2015 Professional | 1 license per developer |  |
| SVN/Git/TFS | 1 user account per developer |  |
| Hosting | | |
| Windows Server 2008 R2 | 1 license |  |

## SKILLSET REQUIREMENTS

Human resources are required for application development, frond end design & development and testing with skills in below technologies.

|  |  |  |
| --- | --- | --- |
| Skill | Proficiency | Resources required |
| Application Development | | |
| .Net 4.0 with C# | Intermediate | 5 |
| ASP.Net MVC 4 | Beginner – Intermediate | 4 |
| Entity Framework 6 | Beginner – Intermediate | 4 |
| SQL Server 2012 | Intermediate | 2 |
| UNIX | Intermediate | 1 |
| Frond End Design & Development | | |
| JavaScript | Intermediate | 2 |
| HTML & CSS | Intermediate | 2 |
| Testing | | |
| Manual Testing | Intermediate | 2 |

# WORK ITEMS

The software architecture contains 5 layers and they are,

1. Monitoring Layer
2. Service Layer
3. Data Layer
4. Business Layer
5. Presentation Layer

## MONITORING LAYER

Monitoring Layer is responsible to perform checks and will be a Generic Monitoring implemented using Windows Service which is configurable to monitor different systems. Logic to perform checks will be implemented as plugins to provide flexibility to extend in future.

### SCHEDULER

Module provides functionality of scheduling tasks. Tasks includes checking servers, databases and scheduled jobs. Scheduler will be configurable so that tasks could be released specific point of time. Work item includes development of logic for configuring scheduler and releasing tasks.

### PLUGINS FRAMEWORK

Module responsible for searching and invoking plugins. The library search for installed plugins and identifies ‘Tasks’ provided by plugins from Windows service on its startup.

### PLUGIN HELPERS

The module contains functionalities to be shared among Windows Service and Plugins. This helps de coupling plugins from Windows Service.

### BASIC PLUGINS FOR ABACUS

This work item includes development of basic plugin to provide ‘Tasks’ to monitor ABACUS. Tasks currently identified are,

1. Monitor DB2 Database
2. Ping Servers
3. Identify status of scheduled jobs by checking logs
4. Identify status of scheduled jobs by checking database
5. Identify status of scheduled jobs by executing series of UNIX commands

## SERVICE LAYER

The module acts as gateway of information flow into dashboard database. Service layer will be implemented as a RESTful Web service or File watcher service. A RESTful web service will have below work items,

### IP WHITELISTING

This module provides security for information flow into dashboard. Information will only be accepted from whitelisted IP Addresses. Work item includes development of IP whitelisting and validation modules.

### WEB SERVICES

A web services will be accessed by clients to push information into dashboard. Work item includes development of web resource and logic for storing the information into database.

## DATA LAYER

The layer provides functionality for accessing information from database. And the work items are,

### DATABASE DESIGN

This work item includes designing databases, views used by dashboard and development of stored procedures and triggers.

### ORM AND DATABASE ENTITIES

The work item includes development of Object Relational Mapping and generating entities representing database objects.

## BUSINESS LOGIC

The module defines logic for managing applications and its components that are under monitoring. An application can have multiple components like servers, databases, scheduled jobs, their credentials and they are treated as sub modules. These sub modules will have different priority in development timeline. The sub modules are,

### APPLICATION AND COMPONENTS

The module is responsible for identifying the components to be monitored in an application and how a components has to be monitored. Components could be servers, databases and scheduled jobs. Work item includes development of application management framework.

### RULES

The module defines logic for determining an application’s overall status based on a set of rules. Work item includes implementing logic for creating/managing rules and processing information based on rules.

### CREDENTIALS

The modules defines common entry point for managing credentials used for different systems. Work item includes implementing logic for storing credentials encrypted and retrieving them decrypted.

### NOTIFICATION & SUBSCRIPTION

Module implements email notification functionality and subscription to events. Work item includes development of notification & subscription framework.

### REPORTS

Module provides functionality for collecting information from sources and storing the information in a unified format. Any kind of information shown in dashboard other that status information will fall under category of report. Work item includes development configuring, generating and storing reports data.

### LOGGING

Work item includes development of set of APIs for logging actions and events into database. This module has low priority in development cycle and will be implemented in final stages of development.

## PRESENTATION LAYER

Contains presentation logic, how data is presented to the user. Additional access control may be implemented at presentation logic. It also consumes libraries containing business logic. Modules are listed below.

### LANDING PAGE

Landing page will be listing all application under monitoring with corresponding status. Additionally, landing page will include summary of errors occurred within system. This work item includes UI development using HTML, CSS, JavaScript and consumption of business logic to present data.

### APPLICATION DETAILS PAGE

Every application will have an application specific detail page. Detail page will show in depth statistics of components monitored in application. This work item includes UI development using HTML, CSS, JavaScript and consumption of business logic to present data.