

**University of Portsmouth School of Computing**

**Student and Unit Management System – Maintain System  
Requirements Specification**

**Version 1.0**

**Table of Contents**

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Glossary 4

1.4 References 5

1.5 Document Control 5

2 Overall Description 6

2.1 Product Perspective 6

2.2 Product Features 6

2.3 User Classes and Characteristics 6

2.4 Operating Environment 7

2.5 Design and Implementation Constraints 7

2.6 Assumptions and Dependencies 8

2.7 System Features 8

3 External Interface Requirements 8

3.1 User Interfaces 8

3.2 Hardware Interfaces 8

3.3 Software Interfaces 9

3.4 Communications Interfaces 9

4 Non-functional Requirements 9

4.1 Archival 9

4.2 Audit and Control 9

4.3 Authentication 9

4.4 Authorization 9

4.5 Availability 9

4.6 Compatibility 10

4.7 Configurability 10

4.8 Data Integrity 10

4.9 Extensibility 10

4.10 Installation 10

4.11 Integration 10

4.12 Leveragability / Reuse 10

4.13 Licensing 11

4.14 Localization 11

4.15 Maintainability 11

4.16 Multiple Environment Support 11

4.17 Personalization 11

4.18 Portability 11

4.19 Privacy 11

4.20 Reliability 12

4.21 Robustness 12

4.22 Security 12

4.23 Usability / Achievability 12

# Introduction

## Purpose

This document aims to capture the requirements of the SUMS – Maintain System, primarily the software requirements and description of functionality.

This requirements specification will be the basis upon which the system will be designed and implemented.

## Scope

The University of Portsmouth’s School of Computing (SoC) maintains a final year project unit management system called Project Units Management System (PUMS). PUMS is used throughout SoC for tracking the progress of a student’s final year project from initial project choices through project approval, supervisor and moderator allocation, project milestones, and up until the date of submission. PUMS is basically a series of linked web pages written using the HTTP and CGI protocols and runs in a web browser connected to the internet.

The development of SUMS is carried out via smaller modules. Each module will be a web-based enterprise application written in Java. Each module will provide a web application using Apache Struts and Hibernate, two Java Persistence API (JPA) frameworks. The Struts framework allows for the clean separation of the model from the view and controller, while Hibernate maps the model (actual Java classes) to tables in a relational database.

The SUMS – Maintain System is a module of SUMS required to handle the general administrative actions to be carried out on data stored within SUMS.

## Glossary

| **Term** | **Definition** |
| --- | --- |
| API | Application Programming Interface |
| CGI | Common Gateway Interface – Interface between external programs and a web server. |
| Controller | Java Class instance that passes information between the view and model. Usually a Java Servlet. |
| Flat file | Computer file containing data records stored sequentially, with each field instance separated by a delimiter. |
| HTML | Hypertext Markup Language. |
| HTTP | Hypertext Transfer Protocol – Internet standard for data exchange between web server and web browser. |
| HTTPS | A secure version of the HTTP protocol. |
| IDE | Integrated Development Environment |
| IEEE | The Institute of Electrical and Electronics Engineers. |
| Java EE | Java Platform, Enterprise Edition. |
| JPA | Java Persistence API |
| JSP | JavaServer Pages |
| Middleware | Software that connects applications through the network interaction enabling services it provides. |
| Model | Application logic that interacts with a relational database. |
| PUMS | The Project Units Management System. |
| SoC | The University of Portsmouth’s School of Computing. |
| SUMS | The Student and Unit Management System. |
| UoP | The University of Portsmouth. |
| View | HTML pages presented to the client. |

## References

Briggs, J. (2007). *SUMS Documentation*. Retrieved January 7, 2008, from SUMS web site: <http://userweb.port.ac.uk/~briggsj/jimapp/SUMS/index.htm>

Sponsor, S., (1998). *IEEE Std 830-1998,* IEEE Recommended Practice for Software Requirements Specifications. New York: Institute of Electrical and Electronics Engineers.

## Document Control

Version: 1.0

Date: 15th January 2008

Principals: **Dr. Jim Briggs:** Head of Development, Student and Units Management System

Author: **Udeme Ekong:** Lead Designer, SUMS – Maintain System

Organization: University of Portsmouth School of Computing

# Overall Description

## Product Perspective

The SUMS – Maintain System is a web-based application for the performance of general administrative actions required in SUMS. The SUMS – Maintain System will work with data stored in the central SUMS database. The SUMS – Maintain System will be available to staff within SoC having granted administrative rights on the SUMS database.

## Product Features

* ***CRUD operations:*** Users can perform Create, Read, Update and Delete operations on SUMS related data as listed below.
  + *Staff*
  + *Student*
  + *Unit*
  + *Unit Instance*
  + *Cohort*
  + *Course*
  + *Course Instance*
  + *Milestone*
  + *Marking Form*

## User Classes and Characteristics

There is currently only one role, “Administrator.” Users have the ability to perform the tasks listed below.

* Create new SUMS related data
* Query / Read SUMS related data
* Update SUMS related data
* Delete SUMS related data
* Duplicate Milestones based on a Unit Instance and (optionally) adjust the date.

## Operating Environment

***Server***

The development server will be a machine running Microsoft Windows XP Professional SP2; the development database is MySQL Community Server 5.0. They are both hosted locally.

The operating server will be a machine running Linux; the operating database is Oracle. They are both hosted by SoC.

The middleware software will be a Java web application server.

***Client***

Clients are located both on and off UoP campuses. As a minimum requirement the client machine must have a web browser connected to the internet. Minimum recommended versions are: Internet Explorer 6.0 or Mozilla Firefox 1.0. For notifications, the user requires an internet e-mail address and e-mail reader.

***Network***

The SUMS – Maintain System must be run on a Java web or application server within the Soc network. It must be accessible by both users within and outside UoP network who are registered with SoC department.

## Design and Implementation Constraints

The SUMS – Maintain System source code must be written using Sun Microsystems Java programming language, particularly using the Java EE 5 platform. All editing is carried out using the NetBeans 6.0 IDE. The Struts and Hibernate frameworks are integral to the design of the SUMS – Maintain System as they ensure the SUMS – Maintain System adheres to strict principles surrounding the MVC design pattern, while also providing transparent persistence between database and Java objects.

A repository containing pertinent the SUMS – Maintain System and related SUMS source code, web pages and documentation is made available for use and reuse. Version control and access by developers are maintained in NetBeans by means of the Subversion system.

The end-users should not require any special setup to work with the SUMS – Maintain System web application.

## Assumptions and Dependencies

The SUMS – Maintain System uses data stored by SUMS on a regular basis. This data must be continually available for the SUMS – Maintain System to function correctly.

## System Features

The SUMS – Maintain System accepts data from and displays data to clients in the form of HTML and JSP pages. It generates a response either through the HTTP web interface. This data provided in response is acquired through the Hibernate API and is a subset of data available from the SUMS database.

# External Interface Requirements

## User Interfaces

The SUMS – Maintain System will be available as an accessible menu in SUMS to staff members with appropriate administrative privileges. The SUMS – Maintain System HTML and JSP pages will follow a theme consistent with SUMS.

## Hardware Interfaces

There are currently no defined hardware interfaces.

## Software Interfaces

The SUMS – Maintain System uses Java EE and Struts APIs which allows the extension of functionality and web services to users over the internet, so no software interfaces are required.

## Communications Interfaces

The SUMS – Maintain System works with computers running the TCP/IP protocol. Using the HTTP protocol, these computers can receive and send responses. These responses are provided in HTML format for consumption by a web browser.

# Non-functional Requirements

## Archival

Relevant information is generated from the SUMS database. This information is immediately consumed by the client, so no archival efforts are required.

## Audit and Control

Every significant user action gets logged within SUMS.

## Authentication

The SUMS – Maintain System will be available to only SoC staff with administrative rights to the SUMS database. Authentication will be verified on login to SUMS.

## Authorization

All the SUMS – Maintain System development information is available to SUMS development teams.

## Availability

There are currently no specific availability requirements. A server failure may result in the SUMS – Maintain System being temporarily unavailable.

## Compatibility

The SUMS – Maintain System conforms to the standards listed below.

* Apache Struts 2
* Hibernate 3.2
* HTML 4.0
* HTTP/1.1
* Java SE 6 / Java EE 5

## Configurability

There are currently no defined configurability requirements.

## Data Integrity

SoC already has appropriate procedures in place to ensure data integrity.

## Extensibility

Additional feeds for receiving data may be added as required.

## Installation

The SUMS – Maintain System and its necessary components can be installed on any server capable of running a Java web application server.

## Integration

Using the Java EE and Struts APIs, modules of SUMS are free to integrate the functionality of the SUMS – Maintain System as needed.

## Leveragability / Reuse

Many of the components of the SUMS – Maintain System may be reused in other modules of SUMS.

## Licensing

SoC already has appropriate procedures in place to deal with any licensing issues.

## Localization

The SUMS – Maintain System is currently available in the UK/English locale.

Further localization is possible using the Struts API.

## Maintainability

Maintaining the SUMS – Maintain System requires knowledge of Java EE, Struts 2 and JPA with Hibernate.

## Multiple Environment Support

The SUMS – Maintain System is maintained on a local server while in development. All new features are developed and tested before being released to the production server.

## Personalization

There are currently no defined personalization requirements.

## Portability

The SUMS – Maintain System may be ported to any hardware platform supporting all other constraints outlined as non-functional requirements.

## Privacy

Where necessary, the SUMS – Maintain System makes use of the HTTPS protocol, instead of HTTP protocol, to ensure data privacy while in transmission between server and user machines.

## Reliability

The data accessed by the SUMS – Maintain System is generated from the SUMS database. This data is considered the definitive repository.

## Robustness

Users generating erroneous data are immediately notified through their web browsers or through the Struts API. Users are given the opportunity to correct and resubmit data.

## Security

Only authenticated users are enabled access to the SUMS – Maintain System.

## Usability / Achievability

Users familiar with web browsers should be able to use the SUMS – Maintain System with no training.