SEM5640 Group Project
Siarad Project
Requirements Specification

Author: Neil Taylor, Nigel Hardy
Config Ref: SEM5640.2019.RS
Date: 15th October 2019
Version: 1.2
Status: Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

CONTENTS

1. INTRODUCTION

1.1 Purpose of this Document

This document describes the requirements for the Integrated Masters Group Project 2019-2020 for SEM5640.

1.2 Scope

This requirements specification describes a distributed computer-based messaging system to provide a discussion facility to support student learning in a university. The application will support the real-time distribution of messages, which are grouped by modules. There will be facilities to search the messages and configure when notifications are sent. These requirements describe new software systems that need to be developed.

1.3 Objectives

The objectives of this document are:

- To describe the background to the SEM5640 group project application (Siarad).
- To provide details of the criteria that the group project's product must meet.
- To describe the types of interaction with the system which must be supported.
- To describe the technologies that must be used to implement the system.

2. GENERAL DESCRIPTION

2.1 Product Perspective

It is proposed to develop Siarad, a new application to enable students and lecturers to discuss topics on modules in a modern interactive way that offers an alternative to using Blackboard (Bb)[1] [1] discussion forums.

Bb discussion forums are linked to each module. They offer a way for everyone on a module to interact by posting messages, known as Threads, and allowing anyone to respond to those messages. However, the interface relies on full-page refreshes and resembles older formats of discussion forums that are slow to use.

The Siarad system is to provide a central hub for conversation on the modules, making it easier to interact. There are some requirements for the format of messages, but there is scope to develop the features supported within a message to make them more engaging for the users of the system.

A system such as Slack[2] [2] is an example of a modern, interactive way to provide messaging in teams. Whilst the Slack system provides some inspiration, the Siarad system should focus on features that would support learning by students in all departments in a Higher Education environment.

The Siarad system will provide the following top-level features:

- Discussion areas for each module that a student is registered for.
- Ability to create extra discussion areas within a module.
- Ability to have one-to-one discussion between staff and students on a module.
- Search facility to find messages with certain search criteria applied.
- Summary statistics available to staff about the usage of the system.
- Notifications sent to users according to some criteria, e.g. when a user is mentioned directly in a post.
- Administrator access to any messages to investigate any claims of inappropriate messages being sent.

2.2 Product Operation

The application will be a set of server-based applications that store and manage messages and related data. These server applications will:

- manage the messages that are written by users and how those messages are linked together, e.g. a set of messages in a module and a set of replies to a specific message;
- provide real-time updates of messages to users who are connected to the system;
- manage the list of modules and users, organised by the academic year;
- search the messages using a set of criteria, e.g. text message or date range;
- manage user preferences for notifications;
- integrate with the department's LDAP service to provide authentication and authorisation

[1] Blackboard and Bb are trademarks or registered trademarks of Blackboard, Inc.
[2] Slack is a trademark or registered trademark of Slack Technologies, Inc.

2.3 Technology

It has been decided that the application will be developed as a set of cooperating micro-services. This is an architecture style that is intended to allow for the set of services provided by Siarad to be upgraded or modified easily with least disruption to the overall service. The services will be written using .NET Core and Java EE.

Microsoft has developed SignalR [3], which provides real-time communication within an ASP.NET application. The group will use SignalR to provide the distribution of real-time messages within Siarad.

A management team within the University is interested in exploring the open source Apache Solr [4] tool to provide a search facility. It would like the project to investigate the tool as an early part of the work. The group should write a technical report on the feasibility of using Solr to provide the search facility. Following the investigation, the group will make a choice to either use Solr or produce an alternative mechanism to provide the search facility.

Figure 1 shows an example decomposition of the main elements in the system, which are described in this document.

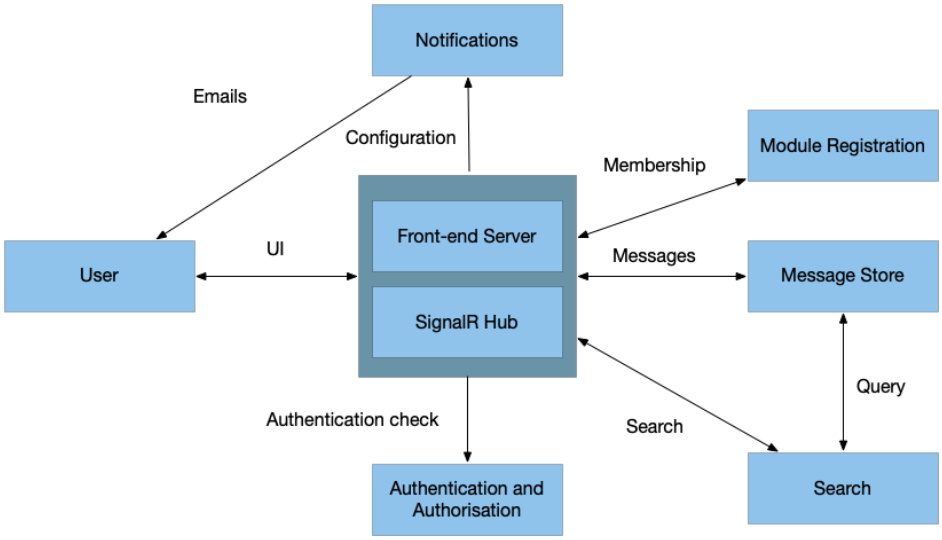

Figure 1: Proposed decomposition of the system

2.4 User Categories

The software will have the following categories of users.

- **Student** – This user can post messages – either creating new messages or responding to other messages.
- **Module Staff** – This user can post messages and has the ability to block a message from being viewed by other users.
- **Administrator** – This user is responsible for overall administration tasks, managing the list of modules and students. The user has the ability to block a message from being viewed by other users. The user also has the ability to see any message in the system, including those in a one-to-one communication.

3. SPECIFIC REQUIREMENTS

3.1 Functional Requirements

This section lists the key functional requirements for the systems.

3.1.1 Messages

The functional requirements for the messages are:

- **M-FR1 – Message Content**
  Messages are text-based content items within the system. Each message has the following information:
  - User – the user id who wrote the message.
  - Message Body – the content of the message, written in Markdown [5] format.
  - A list of emoticons that users have attached to the message. Each emoticon has an associated list of user ids to indicate who added that emoticon.
- **M-FR2 – Referring to other users**
  A message can refer to other users by using @userid within the body of the message. That can generate a notification to alert the other user that they have been mentioned in a message.
- **M-FR3 – Messages Groups**
  Messages are grouped by groups, within modules. A module has at least one group, which is the main group. Messages in a group are ordered by the time that the message was created.
- **M-FR4 – Creating Groups**
  Any user can create new groups. When the group is created, an announcement is made in the main group. The group will be displayed in a list of groups for the module. A group is available to all users within the module.
- **M-FR5 – One-to-one Messages**
  A user can send messages directly to another user.
- **M-FR6 – Replying to messages**
  A user can reply to any message in the group. Replies are shown in the order in which they were created.
- **M-FR7 – Editing Messages**
  A user can edit a message that they wrote. When a message is edited, there is an indication to everyone that it has been edited.
- **M-FR8 – Deleting Messages**
  Any message can be deleted by members of the module staff or by administrators.

  Students can delete their own messages if there are no replies to the messages.

  When a message is deleted, it will be removed from the groups of messages shown to users. A deleted message will not be removed from the Message Store; instead it will be marked as deleted so that it is not be shown in the groups of messages.
- **M-FR9 – New Message Indicator**
  When new messages have been posted, the system will update the user interface to indicate that new messages are available within modules and groups.

3.1.2 Module Registration

The system will manage a list of module information. The information will be used to determine which modules are presented to the different staff and students. This section lists the requirements for that module management.

- **MR-FR1 – Module Information**
  The system will allow administrators to create modules, with the following information:
  - Module Code, e.g. SEM5640.
  - Title, e.g. Developing Advanced Internet-based applications.
  - Academic Year, e.g. 2020 will represent the year from September 2019 to August 2020.
  - Staff Members, which is a list of user ids for the staff working on the module.
- **MR-FR2 – Editing module information**
  The information about modules can be edited by administrators.
- **MR-FR2 – Student List**
  A module will have a list of students who are registered on the module. The University's central student management system contains a list of students taking a module. A file can be exported from that system in a CSV format. This system will provide a facility for an administrator to upload that student list for each module.
- **MR-FR3 – Updating the student list**
  If a CSV list is uploaded to a module where there are existing students for the academic year, the module list is updated.
- **MR-FR3 – Module Membership**
  The Module Registration part of the system will provide a resource to the rest of the system. It will enable other parts of the system to find out which modules a user is a member of for a given academic year.

3.1.3 Search

The system will provide a search facility. This section describes the requirements for that facility.

- **S-FR1 – Search**
  A user can search for messages that match text entered by the user. The search is limited to any modules that the user is associated with.
- **S-FR2 – Search Filters**
  The search can be filtered in the following ways:
  - across all groups in a module,
  - within a specific group,
  - across all modules in a given academic year, or
  - in all modules that a user is associated with.
- **S-FR3 – Search Results**
  The search results will display the messages that match the search text. Where the message is a reply to another message, it should be possible to expand the result to show all

messages: the original message and all replies, while clearly identifying the message found in the search.

3.1.4 Notifications

The system will provide the option for notifications. The notifications will be sent as emails. This section describes the requirements:

- **N-FR1 - Register for Notifications**
  A user can register for notifications from the system. The system will present a set of notification types, as specified in N-FR2. A user can change the registration at any time.
- **N-FR2 – Notification Types**
  The system will provide the following types of notifications:
  - Daily summary – request a daily summary of messages within a specific group or module. The summary can indicate how many messages were created during the day and provide a link to view the messages in the system.
  - Mentions – request a notification when the user is mentioned in a message. The notification will include the text of the message and a link to see the message in the system.
  - Replies to a message – request a notification when a specific message is replied to.
- **N-FR2 – Frequency of Notifications**
  The system will provide a mechanism to send notifications at set time intervals throughout the day. For example, once an hour, the system will check if any conditions have been met for the notifications. If they have, the notifications are sent to the relevant users.

3.1.5 Authentication and Authorisation

The system will use authentication and authorisation.

- **AA-FR1 – Authentication**
  The system should be configured to use an LDAP server within the Computer Science department to authenticate users. In addition to LDAP, the system should have an alternative authentication method for development and test purposes.
- **AA-FR2 – Authorisation**
  Users may be identified in the LDAP server as staff. This information can be used to distinguish between staff and student user categories.

3.2 External Interface Requirements

This section lists general interface requirements for the systems.

- **EIR-1 Appearance**
  The system should be developed as a set of microservices. There should be a suitable User Interface that allows access to the different facilities; there might be one user interface that interacts with the different services. User access to the service is through a web interface that can be accessed through modern web browsers.
- **EIR-2 Internationalised interface**
  The system should be internationalised so that the user interface can be available in different languages. English and Welsh should be supported in this version of the system. For delivery, only English language localisation need be provided.

3.3 Performance Requirements

For this prototype, there are no specified performance or reliability requirements.

3.4 Design Constraints

The following design constraints must be met.

- **DC-1 Use of Java EE**
  At least one of the microservices must be written in Java EE.
- **DC-2 Use of .NET**
  At least one of the microservices must be written in ASP.NET Core.
- **DC-3 Use of web services**
  The system will use RESTful web services for communication between different microservices.
- **DC-4 Reuse of 3rd party software**
  Use of existing 3rd party libraries for parts of the solutions is allowed. Please note that there is possibility of licensing the system under a commercial licence or a licence such as the Apache licence. Your group would need to check the licence terms for the 3rd party software and discuss with your manager before committing to the use of any 3rd party software.
- **DC5 Use of Docker**
  The services are to be deployed to the Docker facilities for testing and demonstration. The group will be provided with access to Docker, hosted on a University machine.

4. References

[1] Blackboard, Inc. (2019) Blackboard Website (Online) https://uk.blackboard.com/ (Accessed 14th October 2019)

[2] Slack Technologies, Inc. (2019) Slack Website (Online) https://slack.com/intl/en-gb/ (Accessed 14th October 2019)

[3] Microsoft (2019) Real-time ASP.NET with SignalR (Online) https://dotnet.microsoft.com/apps/aspnet/signalr (Accessed 14th October 2019)

[4] Apache Software Foundation (2019) Solr Website (Online) https://lucene.apache.org/solr/ (Accessed 14th October 2019)

[5] Gruber, J. (2019) Markdown (Online) https://daringfireball.net/projects/markdown/ (Accessed 14th October 2019)

DOCUMENT HISTORY

Version	Date	Changes made to document	Changed by
1.0	2019-10-14	Initial Draft	NST
1.1	2019-10-15	Additional requirements.	NST, NWH
1.2	2019-10-15	Changes following moderation. Set to Release.	NST