Software Requirements Specification

for

Slack Technologies  
Cloud-based Team Collaboration Tool

Version 1.0

Produced by Alexander R Ramsey for University of Maryland, University College

Disclaimer: This document is generated for educational purposes only   
and does not officially represent Slack Technologies.

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 1

1.4 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3. External Interface Requirements 5

3.1 User Interfaces Overview 5

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

4. Non-Functional and Functional Requirements Definitions 6

4.1 Non-functional Requirements Definitions 6

4.2 Functional Requirements Definitions 10

5. Other Non-Functional Requirements: System Execution and System Evolution Qualities 16

5.1 System Execution Qualities 16

5.2 System Evolution Qualities 16

6. Key Terms and Data Model / Dictionary 17

6.1 Key Terms 17

6.2 Data Dictionary 19

6.3 Use Case Diagram 20

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Alexander R Ramsey | 03/24/2017 | Inception of document | 1.0 |
| ^ | 03/25/2017 | Addition of Sections 1 thru 5 | 1.1 |
| ^ | 03/26/2017 | Addition and Adjustments of Content to Sections 4 thru 5. Addition of Section 6. | 1.2 |

# Introduction

## Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the Slack Technologies online chat room. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. All documented requirements are of high priority and will be included in release 1.0 of the Slack Technologies chat room. This document conforms to the IEEE 830 Software Requirements Specification document. It strives to conform to the suggested best practices and standards set forth by the IEEE 830 1998 specification.

## Intended Audience and Reading Suggestions

This document is intended for the software development team at Slack Technologies to continue work on release 1.0 of the Slack chat program. This document contains industry specific terminology and requires a web programming background to understand some aspects of relationship and data identity to fully understand. This document serves as an extension to a former Software Requirements Specification, intended to serve as a product pilot. This document establishes additional functionality in the form of nonfunctional and functional requirements, designed to provide Slack with market differentiation.

## Product Scope

Slack Technologies will provide a team collaboration solution in the form of a feature rich chat room to public account holders. The product will be called Slack. Slack is a messaging suite for teams with data accessible via a back-end cloud web service and cross platform front-end. It will allow for teams to communicate via Team Accounts. Team Accounts are accessible via an easy to remember and custom set URL. A Team Account will provide teams with the ability to create Topic chat areas. Each Topic chat area will have pinned items. Each Topic chat area will allow users to mention others. Topics will be visible upon defined user roles. Topic notifications can be muted. Topics can integrate with other SaaS solutions to start a web meeting, share a gif image, or to share a document.

## References

1. IEEE 830 Software Requirements Specification Recommended Practices. 1998. Accessed via https://learn.umuc.edu/d2l/le/content/202339/viewContent/7944885/View
2. Rational Unified Process Documentation. 1999-2007. Accessed via http://sce.uhcl.edu/helm/rationalunifiedprocess/index.htm
3. Slack Technologies. 2017. Accessed via http://www.slack.com
4. Pilot Software Requirements Specification. 2017. Assignment one of UMUC SWEN 645 Course for Slack Technologies. Accessed via http://alexanderramsey.com/blog/?page\_id=451
5. Standard ECMA-262. (2016). Retrieved March 25, 2017, from http://www.ecma-international.org/publications/standards/Ecma-262.htm
6. Google Material Design Specification. 2017. Accessed via https://material.io/

# Overall Description

## Product Perspective

Slack is an internet based, cross platform accessible chat utility designed to perform similarly to existing IRC chat room systems. It will differentiate itself by providing a Software-as-a-service (SaaS) cloud delivery model to public account holders who form a team specific chat room. The chat rooms will be accessible via a team subdomain.

## Product Features

Nonfunctional and Functional requirements are an extension of a pilot Software Requirements Specification document, which can be accessed via the section 1.4 references section. This document will cover the nonfunctional requirements and some extended functional requirements to enhance the market differentiation of Slack for release 1.0. The following requirements are outlined in Figure 1, Context Diagram for User Roles. This diagram shows all user role relationships with the Slack chat server interacting via asynchronous communication.

Functional requirements established in Pilot Software Requirements Specification accessible via http://alexanderramsey.com/blog/?page\_id=451

* Access via web browser
* Create Account
* Create Team Account
* Join Existing Team Account
* Revisit Team Account Cross Platform
* Revisit Team Account on a Mobile form factor
* Post to a Channel
* Create a Channel/Topic
* Pin an Item to a Channel/Topic
* Mention a Team User
* Mention a Channel/Topic

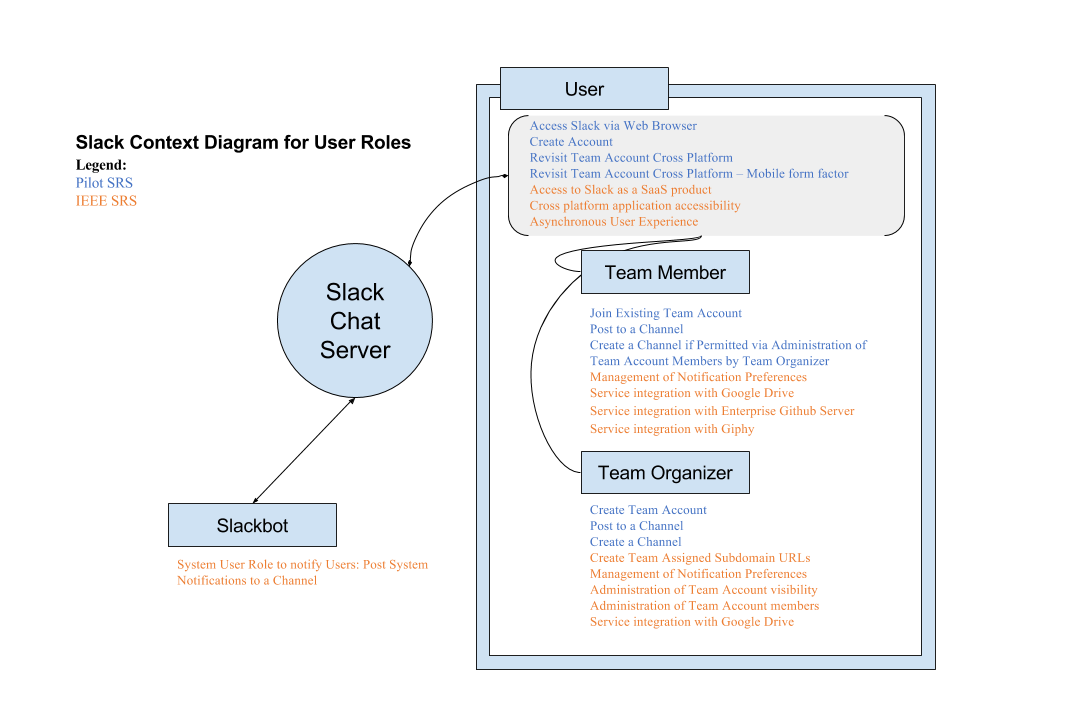
Nonfunctional Requirements were discussed briefly in the Pilot Software Requirements specification. Their specific details will be covered in this document as product features. A broader reaching section on nonfunctional requirements pertaining to execution and evolution qualities will also be addressed in this document. These three (3) system-specific nonfunctional requirements include the following:

* Cloud based Software as a service (SaaS) delivery model
* Cross platform application accessibility via modern web browser, Windows application, Mac application, Linux application, iOS Mobile application, and Android Mobile application
* Asynchronous user experience via a modern API client and server relationship

Functional requirements and features covered in this document will provide Slack with increased market differentiation for release 1.0. These seven (7) include:

* Team assigned subdomain URLs for account holders
* User management of notification preferences
* Administration of Team Account visibility and Administration of Team Account members
* Slackbot System User Role to notify Users
* Service integration with Google Drive
* Service integration with Enterprise Github Server
* Service integration with Giphy

### Figure 1: Context Diagram for Slack User Roles



## User Classes and Characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| **User Class** | **User Role** | **User Characteristics** | |
| **Estimated Population at release 1.0** | **Available Actions at release 1.0 (Pilot SRS followed by IEEE SRS)** |
| User | Any registered user of the system | Greater than 100 thousand | * Access Slack via Web Browser * Create Account * Revisit Team Account Cross Platform * Revisit Team Account Cross Platform – Mobile form factor * Access to Slack as a SaaS product * Cross platform application accessibility * Asynchronous User Experience |
| Team Member | A user which joins an existing Team Account. Inherits User class available actions. | Greater than 80 thousand | * Join Existing Team Account * Post to a Channel * Create a Channel if Permitted via Administration of Team Account Members by Team Organizer * Management of Notification Preferences * Service integration with Google Drive * Service integration with Enterprise Github Server * Service integration with Giphy |
| Team Organizer | A user which initiates a Team Account. Inherits User class available actions. | Greater than 20 thousand (1 Organizer per 4 Members) | * Create Team Account * Post to a Channel * Create a Channel * Create Team Assigned Subdomain URLs * Management of Notification Preferences * Administration of Team Account visibility * Administration of Team Account members * Service integration with Google Drive * Service integration with Enterprise Github Server * Service integration with Giphy |
| Slackbot | A System User which communicates system related messages to the User. | Not applicable. | * System User Role to notify Users: Post System Notifications to a Channel |

## Operating Environment

OE Requirement 1 – Slack shall be accessible via a modern web browser, Windows application, Mac application, Linux application, iOS Mobile application, and Android Mobile application

OE Requirement 2 – Slack shall be hosted within a cloud Platform as a service environment with a Unix based architecture.

OE Requirement 3 – Slack PaaS system should provide scalable system resources to accommodate more than 100 thousand users.

OE Requirement 4 – A messaging service should exist that can accommodate asynchronous data requests via a Web Socket communication pipe.

## Design and Implementation Constraints

CO 1 – The System server shall be hosted in a Platform as a service cloud environment with virtual machine support

CO 2 – The System’s API shall conform to the latest European Computer Manufacturers Association (ECMA) Script 6 development paradigm

CO 3 – The System data shall be stored using MongoLabs Object Oriented database clusters

CO 4 - Cross platform software shall be developed using the Xamarin extension of Microsoft’s .NET Platform to support Android, Windows, and iOS development.

## User Documentation

UD 1 – The system shall provide interactive tooltips throughout the process of user interaction.

UD 2 – The system shall present terms and conditions of usage during the user registration process.

## Assumptions and Dependencies

AS 1 – Users of the system shall have a basic understanding of computer and web technologies.

AS 2 – Users of the system shall have a valid and accessible email account for registration and notifications.

# External Interface Requirements

## User Interfaces Overview

UI 1 – Slack shall conform to the Google Material Design standard of user interface design accessible via material.io/guidelines

Google Material design states the following short goal statements:

“Create a visual language that synthesizes classic principles of good design with the innovation and possibility of technology and science.” And “Develop a single underlying system that allows for a unified experience across platforms and device sizes. Mobile precepts are fundamental, but touch, voice, mouse, and keyboard are all ﬁrst-class input methods” (Google, 2017).

Google Material design is available as a CSS and JS library called Materialize CSS, located via www.materializecss.com

This framework provides similar functionality to Bootstrap via www.getbootstrap.com

UI 2 – Slack shall utilize asynchronous front end form and user response validation to eliminate any page reloading of the application.

UI 3 – Slack shall utilize a responsive layout with a mobile first design concept to ensure compatibility with devices of all sizes.

UI 4 – Slack UI will use pastel colors and bright type colors randomly chosen from a color scheme set. This will engage users with an artful theme.

## Hardware Interfaces

HI 1- Slack will utilize existing end user platform hardware to establish a network connection link via WiFI or Ethernet connection.

## Software Interfaces

SI 1 – Slack will utilize software operating systems as a platform for communication to the hardware interfaces.

# Non-Functional and Functional Requirements Definitions

Aside to professor: The IEEE 830 standard states that, “This recommended practice does not identify any specific method, nomenclature, or tool for preparing an SRS.” Therefore, it should be noted that this format is not a strictly binding one. I am using this interpretation to therefore change the way that the SRS is structured to move the requirements definition section to after the External Interface Requirements section.

The following requirements definition section will be divided into two (2) major parts: the first describes the non-functional requirements and their high-level implementation details. The second section provides descriptions of the functional requirements and their itemized definitions following the IEEE 830 recommended specification.

## Non-functional Requirements Definitions

### SaaS Delivery Model

4.1.1 Description and Priority

A Software as a service (SaaS) delivery model dictates that Slack will host and maintain all aspects of the application for the end user. This allows for Slack to maintain consistency over the user experience. This nonfunctional requirement is a high priority for the application and will help the application reach the market in the most efficient and controlled manner.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User requests team account. | System provisions team account per user request and hosts team account. |
| User posts to a channel. | System records and stores User post to Slack managed storage solution, MongoDB. |
| User requests chat program while offline. | System does not receive response and is unreachable by User. |

4.1.3 Nonfunctional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-1 | Access and Registration  The system will provide registration access via a common URL to the public located at www.slack.com |
| NFR-2 | Team Account Creation  The system will generate a protected Team Account for the Team Owner and maintain Team Account access details in a system provisioned Team Account database. |
| NFR-3 | System Server Resources and Data Persistence  The system server resources shall be protected and accessible only to the development and management team at Slack Technologies. All data stored for the application will be maintained by Slack Technologies and will be accessible only via a closed API. |
| NFR-4 | System Cross Platform Updates  All cross-platform applications will be updated via a common repository. No codebase for native application development will be supplied to the public. |
| NFR-5 | System Integration API  The system shall accept only system known and user accepted incoming integration service API requests per Team Account. This functionality pertains to the usage of third (3rd) party applications like Github, Google Drive, and Giphy. |

### Asynchronous User Experience

4.1.1 Description and Priority

Modern web applications utilize what is referred to as Asynchronous communication between the front-end client browser and the back-end web server or application platform interface (API). A synchronous communication would occur based upon a set of predefined instructions for each page load. Asynchronous communication allows for the client to communicate with a web server without reloading the web page. This creates an impressive, seamless, and instant feeling for the end user. This requirement will allow the chat program to send and receive messages to multiple users in real time with timeliness. This requirement is a high priority for the development of release 1.0.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User posts to a channel. | System receives post without browser refresh. System persists data and sends data update to all other session clients. |
| Multiple users post to a channel while logged into the system at the same time. | System receives each user request and stores the data in real time, while pushing data updates to each connected client. |
| User makes a mistake in a validated input field. | System responds immediately with suggestion as to how to correct the mistake in the specific form field. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-6 | Modern JavaScript and ECMA Script Code Conformance  The system will utilize modern JavaScript frameworks designed to handle asynchronous communication with API endpoints. The API communication will meet the technical requirements set forth by the latest version of ECMA Script. |
| NFR-7 | Common API Endpoint for Cross Platform Communication  The system will utilize a common API endpoint. This common API endpoint will listen for client requests. It will communicate with an object-oriented database. It will utilize user session storage and token based authentication to ensure right of way for client requests. |
| NFR-8 | Client Side Validation  The system will perform client side validation of applicable web forms to enforce security, systems stability, user experience, and user confidence. It is recommended that these validations will be applied to the form elements within inline HTML as an Angular JS directive for web browser based platforms. |
| NFR-9 | Server Side Validation  The system will perform server side validation asynchronously via callback functions. These functions will identify and eliminate any false requests which attempt to circumvent data storage security. |
| NFR-10 | High Performance API Response Time  The system shall respond to client requests within 10ms of receiving the request. This will instill confidence in the end user and will provide for a near instant, fully engaging chat experience. |

### Cross Platform Application Accessibility

4.1.1 Description and Priority

The system shall be accessible via multiple platforms with platform compatibility with the data persisting API endpoint. Each independent platform should communicate with speed and reliability to ensure that the user experience is consistent across each platform. The following platforms should be supported: Windows OS Web Browsers, Mac OS Web Browsers, Linux OS Web Browsers, Windows OS Native, Mac OS Native, Linux OS Native, Android OS Native, Apple iOS Native. The priority of this requirement is high.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User accesses via cross platform application. | System recognizes user and returns appropriate data with performance and reliability. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| NFR-11 | Windows OS Web Browser Support  The system shall be accessible via modern web browsers within the specified OS environment. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-12 | Mac OS Web Browser Support  The system shall be accessible via modern web browsers within the specified OS environment. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-13 | Linux OS Web Browser Support  The system shall be accessible via modern web browsers within the specified OS environment. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-14 | Windows OS Native Application Support  The system shall be accessible via a natively installed desktop form factor application. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-15 | Mac OS Native Application Support  The system shall be accessible via a natively installed desktop form factor application. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-16 | Linux OS Native Application Support  The system shall be accessible via a natively installed desktop form factor application. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-17 | Android OS Native Support  The system shall be accessible via a natively installed mobile form factor designed for Android Mobile OS. This accessibility should provide users with a consistent, performant, and confident user experience. |
| NFR-18 | Apple iOS Native Support  The system shall be accessible via a natively installed mobile form factor designed for Apple Mobile iOS. This accessibility should provide users with a consistent, performant, and confident user experience. |

## Functional Requirements Definitions

### Team Based Subdomain URLs

4.1.1 Description and Priority

The application will be provided to the user as a SaaS solution. Therefore, it will be hosted on Slack servers and linked to the Slack dot com URL. The system will communicate with the web server to supply the user with the ability to use a Slack dot com subdomain. For example, if a Team Organizer were to create an account for their company, the system would ask for a specific Team subdomain, and the user would specify their preference. The server would then verify this preference. An approved subdomain preference would then be used to provision a Team Account and Team Database. The priority of this requirement is high.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User requests Team Account hosted at Slack.com | System provides interface for selecting subdomain URL assigned to Team Account. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-1 | Team Account Generation with valid subdomain supplied  User requests a Team Account. System prompts user with free form choice of one Slack.com subdomain. User enters subdomain. System verifies availability. On availability, System provisions Team Account Chat Room instance to specified subdomain. System stores subdomain information in data management repository. |
| FR-2 | Team Account Generation with invalid subdomain supplied  User requests a Team Account. System prompts user with free form choice of one Slack.com subdomain. User enters subdomain. System verifies availability. On unavailability, System prompts user to enter a valid subdomain choice and provides three suggestions to the user for available subdomains. |
| FR-3 | Team Account Deletion  Team Organizer Requests Team Account to be deleted. System accepts request and removes provisioned Team Account and Subdomain from system, making subdomain available for future use by another User. |

### User Management of Notification Preferences

4.1.1 Description and Priority

The system shall provide notifications for newly create chats via a notification service in the application. The system shall provide the user with the ability to select a preference to hide or show these preferences via a user management menu. The priority of this requirement is high.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User makes a post in a channel. | System issues notification to other Team Members that User posted in a channel. |
| User selects preference to hide notifications. | System does not issue any notifications to the User. |
| User makes a post that targets others in a channel. | System issues notification to mentioned targeted users. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-3 | Notification Event Trigger  A User posts a message in a channel. The System responds by publishing a notification about the unread messages to other Users apart of the Team. |
| FR-4 | Notification Hide Setting  The System shall provide Users with a notification preference setting which hides any further notifications from occurring on their account. |
| FR-5 | Notification Show Setting  The System shall provide Users with a notification preference setting which shows notifications on their account. |

### Administration of Team Account Visibility and Administration of Team Account Members

4.1.1 Description and Priority

The system shall provide the Team Organizer with the ability to selectively make their Team Account publicly or privately accessible via a User management menu. The priority of this requirement is high.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| Team Organizer selects option to make Team Account Publicly visible. | System allows any Slack Registered User to view and participate in the chat room. |
| Team Organizer selects option to make Team Account Private to selected members. | System allows only selected Team Members to view and participate in chat room. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-6 | Team Organizer Preferences Menu  The System shall provide for Team Organizers a menu to display settings which pertain to the Team Account. |
| FR-7 | Team Account Publicly Visible  The System shall provide for the Team Organizer an option in the Team Organizer Preferences Menu to make all Team Account chat rooms publicly visible to registered Slack Users. |
| FR-8 | Team Account Private  The System shall provide for the Team Organizer an option in the Team Organizer Preferences Menu to make all Team Account chat rooms private and visible to only selected registered Slack Users. |
| FR-9 | Team Account Team Member Management  The System shall provide for the Team Organizer an option to remove or add Team Members. In addition, the System shall provide the Team Organizer with the ability to make added Team Members assume the role of a Team Organizer. |

### Slackbot System User Role to Notify Users

4.1.1 Description and Priority

The System shall have its own user account for each Team Organizer and Team Member user to display system notifications regarding the status of System Integrations, Notifications, and other pertinent information about the System itself. The priority of this requirement is high.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| The Slack development team issues an update to the System servers. | The System Slackbot User issues a notification chat to users regarding the Slack development team update. |
| The User encounters an unknown error in the system. | The System Slackbot responds by issuing a notification chat to the User regarding the unknown error with suggestions to remedy the error or to contact the support department. |
| The User disables notifications in the preferences menu. | The System Slackbot does not provide any further notifications to the user. |
| The User enables a System Integration | The System Slackbot notifies the user of the status of the System Integration, whether available or unavailable. |
| The User actively types into the Chat box | The System Slackbot provides real time suggestions as a populated list under the User chat box pertinent to the User input. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-10 | Slackbot Notification Preferences Enabled  If User notifications are enabled, the System will display Slackbot to the User. |
| FR-11 | Slackbot Notification Preferences Disabled  If User notifications are disabled, the System will not display Slackbot to the User. |
| FR-12 | Slackbot Development Team Action  If the Development Team publishes a newly updated version of the System, the System will use Slackbot to notify the user of the update and the release notes. |
| FR-13 | Slackbot Error Handling  If a User encounters an unknown error in the system, the User should be provided with a notification from Slackbot acknowledging the error, directing them to support team or system help documents. |
| FR-14 | Slackbot System Integration Response  If a User initiates a System Integration, Slackbot should respond with status messages indicating the availability of the integration to be used. |
| FR-15 | Slackbot Real Time Chat Box Suggestions  If a User types in the chat box, the System Slackbot should respond with recommendations pertaining to the User input. These may include recommendations regarding using a System integration, supplying a desired emoticon, or spelling related suggestion. |

### System Integration with Google Drive

4.1.1 Description and Priority

The System shall be designed to allow integration of data with the external cloud document service, Google Drive. The priority of this requirement is medium.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| User posts link to Google Drive document. | System detects Google Drive URL and suggests integration features to the User via Slackbot inline to the chat box. |
| User accepts integration suggestion. | System responds by including Google Drive document inline to chat room. |
| User declines integration suggestion. | System allows user to post link to document and does not proceed with integration feature enablement. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-16 | Detection of Google Drive URLs  If a User posts a Google Drive URL to a chat box, the System shall detect this URL and notify Slackbot of the URL. |
| FR-17 | Slackbot Prompt to Enable Google Drive Integration Support  If Slackbot is notified of a Google Drive URL, it shall prompt the user with a suggestion to enable the Google Drive integration features. |
| FR-18 | Integration of Google Drive Native Documents  If the integration is applied by the User, the System should display Google Drive documents inline to the chat window. This will allow the User to display the contents of the document to the chat room. |
| FR-19 | Integration of Google Drive Native Documents Error State  If the integration is applied but an error state is detected, the System will cancel the integration and notify the User via Slackbot of the error state of the integration. |

### System Integration with Enterprise Github Server

4.1.1 Description and Priority

The system shall be designed to allow integration of data with the external repository management software, Enterprise Github Server. The priority of this requirement is medium.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| The Team Organizer enters **/git** into the chat box. | The System Slackbot recognizes the Team Organizer request to initiate a Git integration command. |
| An external Git repository is updated. | If integration is enabled, System captures data from the Git repository and posts it to the chat room for all Users. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-20 | Detection of /git query in chat box  If a User enters /git into the chat box, the System should identify this as a request to setup a Git integration. |
| FR-21 | Slackbot Prompt to Enable Git Integration Support  The System Slackbot should provide the User with an option to enable the Git integration based upon supplied parameters. |
| FR-22 | Post Git Repository Updates to Chat Room  The System shall be able to update the Chat room where Git integration is enabled with the latest Git updates from the Git repository. |
| FR-23 | Git Integration Error State  If an error state occurs for the Git integration, the System Slackbot should notify the user that the integration is not operational. |

### System Integration with Giphy

4.1.1 Description and Priority

The system shall be designed to allow integration of data with the external gif image aggregator, Giphy. The priority of this requirement is medium.

4.1.2 Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus | Response |
| A User enters **/giphy parameter** into the chat box. | The System Slackbot recognizes the User’s entry in the chat box and posts a gif image from the Giphy aggregator. |

4.1.3 Functional Requirements

|  |  |
| --- | --- |
| ID | Title & Description |
| FR-24 | Detection of /giphy parameter in chat box  If a User enters /giphy followed by a parameter, the System should identify this as a request to post a gif animated image based on the supplied parameter. |
| FR-25 | Slackbot Prompt to Enable Giphy Integration Support  The System Slackbot should provide the User with an option to enable the Giphy integration. |
| FR-26 | Post Gif Images to Chat room from Giphy Aggregator based on User supplied parameter  If the User approves the Slackbot request to enable the integration, the System will then post a random Gif image from the Giphy aggregator based upon the supplied parameter. For example, if the User enters /giphy papers, the System will find a funny gif image related to papers and post the image. |
| FR-27 | Giphy Integration Error State  If an error state occurs for the Giphy integration, the System Slackbot should notify the user that the integration is not operational. |

# Other Non-Functional Requirements: System Execution and System Evolution Qualities

## System Execution Qualities

### Usability

The system shall be easy to use by any User with existing foundational computer GUI knowledge.

### Security

The system shall not display information to any public unregistered entity not permitted to access Team Chat rooms.

### Performance

The system shall respond to all API requests with a status, within 10ms of receiving the API request.

## System Evolution Qualities

### Extensibility

The system shall be designed in such a manner as to be extensible; made to be extended upon for future development with ease of addition to existing functionality and infrastructure.

### Interoperability

The system shall be easy to interoperate and share data with external integrations by utilizing interfaces and object oriented design principles.

# Key Terms and Data Model / Dictionary

## Key Terms

* API – Application Platform Interface, a technical definition by which data is exchanged successfully by following a defined schema and access protocol among front end and back end web technologies.
* Asynchronous – term used to describe current web technologies wherein JavaScript calls are used to point to an API without browser page refreshes, thus improving the user experience.
* Back end – the server side components which listen for front end requests and supply an HTTP response
* Cloud based, Cloud – term used to describe how a web application is hosted and accessible to the public across the broader internet.
* Cross Platform – software technologies which operate seamlessly across multiple device form factors and operating systems.
* Front end – the client side scripting used to develop a user interface interpreted by web   
  browsers installed to an operating system, often communicates to the back end via HTTP protocol requests
* Giphy – A popular Gif image format aggregator which pairs funny .gif motion images to keywords
* Git – A popular software repository management system
* Google Drive – A popular in browser cloud document creation, management, and collaboration solution
* HTTP – Hypertext Transfer Protocol, a communications protocol used in the current World Wide Web ecosystem
* JavaScript – a currently popular, loosely typed, web language and technology ecosystem
* SaaS – Software as a Service, a cloud service model in which hosting, development, and support are provided for the end user.
* SaaS Integration – the interoperability of information or data between the chat program Slack, with other SaaS products.
* Slack Technologies – represents an actual organization founded in 2013 with more than 700 employees. This company is head by Stewart Butterfield, a founding member of Flickr and former employee of Yahoo, Inc.
* Team Account – accessible protected area that contains team members
* Topic – Also known as Channel, a chat channel pertaining to a specific subject. These terms are loosely mixed because of the topic driven nature of the channel.

## Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Required | Type | Field Length | Default Value | Notes |
| **User** | | | | | |
| User.ID | Yes | Integer | 10 |  | Auto Increment |
| User.Username | Yes | String | 40 |  |  |
| User.Password | Yes | String | 512 |  | Encrypted |
| User.Email | Yes | String | 40 |  |  |
| User.Role | Yes | Integer | - | Team Member | Team Member = 0 Team Organizer = 1 |
| User.TeamURL | Yes | Array | - |  | Inherit Registration URL or Invite URL. Can contain many. |
| **Post** | | | | | |
| Post.ID | Yes | Integer | 10 |  | Auto Increment |
| Post.Username | Yes | String | 40 |  | Appended on Submit |
| Post.Body | Yes | String | 1024 |  |  |
| Post.Timestamp | Yes | Datetime | 10 | Datetime | Auto Stamp |
| **Setting** | | | | | |
| Setting. Team.Public | Yes | Boolean | - | False | Public = True, Private = False |
| Setting. Team. Members | No | Array | - | Team Organizer User.Username | Contains all User.Usernames of Team Account |
| **Integration** | | | | | |
| Integration. Drive | No | Boolean | - | False | Enabled = True,  Disabled = False |
| Integration. Git | No | Boolean | - | False |
| Integration. Giphy | No | Boolean | - | False |

## C:\Users\Alexander\AppData\Local\Microsoft\Windows\INetCache\Content.Word\UseCaseDiagram2.pngUse Case Diagram