

# Part I

## System and Software Design Description (SSDD): Incorporating Architectural Views and Detailed Design Criteria for Groups in a University Setting

# Contents

<b>I System and Software Design Description (SSDD): Incorporating Architectural Views and Detailed Design Criteria for Groups in a Univeristy Setting</b>	<b>1</b>
<b>1 Introduction</b>	<b>7</b>
1.1 Identification . . . . .	7
1.2 Purpose . . . . .	7
1.3 Scope . . . . .	7
1.4 Definitions, Acronyms, and Abbreviations . . . . .	8
1.5 References . . . . .	8
1.6 Overview and Restrictions . . . . .	8
<b>2 Constraints and Stakeholder Concerns</b>	<b>10</b>
2.1 Constraints . . . . .	10
2.1.1 Environmental Constraints . . . . .	10
2.1.2 System Requirement Constraints . . . . .	10
2.1.3 User Characteristic Constraints . . . . .	10
2.2 Stakeholder Concerns . . . . .	10
<b>3 System and Software Architechture</b>	<b>11</b>
3.1 Users Architectural View . . . . .	12
3.1.1 User's View Identification . . . . .	12
3.1.2 User's View Representation and Description . . . . .	12
3.2 Developer's Architectural View . . . . .	12
3.2.1 Developer's View Identification . . . . .	12
3.2.2 Developer's View Representation and Description . . . . .	12
3.2.3 Developer's Architectural Rationale . . . . .	12
3.3 Consistency of Architecutural Views . . . . .	12

3.3.1	Developer's Viewpoint Detailed Software Design . . . . .	12
3.3.2	Component Dictionary . . . . .	12
3.3.3	Component Detailed Design . . . . .	12
3.4	Data Dictionary . . . . .	12
<b>4</b>	<b>Requirements Traceability</b>	<b>13</b>
<b>II</b>	<b>Systems and Software Requirements Specification (SSRS) for Groups in a University Setting</b>	<b>14</b>
<b>5</b>	<b>Overall Description</b>	<b>15</b>
5.1	Product Perspective . . . . .	15
5.2	Product Functions . . . . .	15
5.3	User Characteristics . . . . .	16
5.4	Constraints . . . . .	17
5.5	Assumptions and Dependencies . . . . .	17
5.6	System Level (Non-Functional) Requirements . . . . .	17
5.6.1	Site dependencies . . . . .	17
5.6.2	Safety, security and privacy requirements . . . . .	17
5.6.3	Performance requirements . . . . .	17
5.6.4	System and software quality . . . . .	18
5.6.5	Packaging and delivery requirements . . . . .	18
5.6.6	Personnel-related requirements . . . . .	18
5.6.7	Training-related requirements . . . . .	18
5.6.8	Logistics-related requirements . . . . .	18
5.6.9	Precedence and criticality of requirements . . . . .	18
<b>6</b>	<b>Specific Requirements</b>	<b>19</b>
6.1	External Interface Requirements . . . . .	19
6.1.1	Hardware Interfaces . . . . .	19
6.1.2	Software Interfaces . . . . .	19
6.1.3	User Interfaces . . . . .	19
6.1.4	Other Communication Interfaces . . . . .	19
6.2	System Features . . . . .	21
6.2.1	Use Case Diagrams . . . . .	21
6.2.2	Use Case Diagrams . . . . .	21

6.2.3	System feature 1: Officer Management (Administration Module) . . . .	21
6.2.4	System feature 2: Group Management (Officer Module) . . . . .	23
6.2.5	System feature 3: User Menu . . . . .	27
6.2.6	System feature 4: Forum Module) . . . . .	33
<b>7</b>	<b>Requirements Traceability</b>	<b>35</b>
7.1	... . . . .	36
7.1.1	... . . . .	36
7.1.2	... . . . .	36
7.1.3	... . . . .	36
7.1.4	... . . . .	36
7.1.5	... . . . .	36
7.1.6	... . . . .	36
7.2	... . . . .	36
7.2.1	... . . . .	36
7.2.2	... . . . .	36
7.2.3	... . . . .	36
7.2.4	... . . . .	36
7.2.5	... . . . .	36
7.2.6	... . . . .	36
7.3	... . . . .	36
7.3.1	... . . . .	36
7.3.2	... . . . .	36
7.3.3	... . . . .	36
7.3.4	... . . . .	36
7.3.5	... . . . .	36
7.3.6	... . . . .	36
<b>III</b>	<b>...</b>	<b>37</b>
<b>A</b>	<b>...</b>	<b>38</b>
<b>B</b>	<b>...</b>	<b>39</b>

## List of Figures

# List of Tables

6.1	Hardware Interfaces . . . . .	20
6.2	Software Interfaces . . . . .	20
6.3	User Interfaces . . . . .	20
6.4	Other Communication Interfaces . . . . .	20

# Chapter 1

## Introduction

### 1.1 Identification

The software system being considered for development is referred to as Groups in a University Setting or gus. The customer providing specifications for the ethnic and religion team is the Lutheran Campus Ministry. The ultimate customer, or end-user, of the system will be student groups at the University of Idaho. This is a new project effort, so the version under development is version 0.05

### 1.2 Purpose

The purpose of the system under development is to provide a tool for the easy administration and tracking of university-style groups including but not limited to clubs and sports teams, the system will also try to increase student involvement by connecting and recognizing the involvement of users. While the system will be used by university personnel, this document is intended to be read and understood by UICS software designers and coders. This document will also be approved by Dr. Clinton Jeffery.

### 1.3 Scope

GUS is to social-networking as an intranet is to the Internet. Where other social networks distract users with non-university non-local non-face-to-face non-involvement, GUS will focus these types of functionalities to the university and local community setting to best meet the administrative and service needs of groups in a university setting. In addition to being a

group-centered student-involvement web-application.

## 1.4 Definitions, Acronyms, and Abbreviations

Term or Acronym	Definition
Alpha test	Limited release(s) to selected, outside testers
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
SRS	Software Requirements Specification
SSRS	System and Software Requirements Specification
GUS	Groups in a University Setting

## 1.5 References

1. [www.churchteams.com](http://www.churchteams.com)
2. [www.groupmeister.com](http://www.groupmeister.com)
3. [www.teamr.com](http://www.teamr.com)
4. [www.salesboom.com](http://www.salesboom.com)
5. [www.wikipedia.org](http://www.wikipedia.org)

## 1.6 Overview and Restrictions

This document is for limited release only to UI CS personnel working on the project.

Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.

Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree



sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.

Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number and linked to its SDD and test references.

Sections 5 and up are appendices including original information and communications used to create this document.

# Chapter 2

## Constraints and Stakeholder Concerns

### 2.1 Constraints

#### 2.1.1 Environmental Constraints

At this time, there are no reasonable Environmental Constraints native to GUS other than a working computer, electricity, and a bit of privacy.

#### 2.1.2 System Requirement Constraints

Since G.U.S. is a web-based product, all significant system requirements will be shifted away from the client-side and onto the server. All the client will need to properly utilize GUS is a modern web browser. So far, GUS proven to be compatible with:

- FireFox 3.0 (or higher)
- Internet Explorer 7.0 (or higher)
- Google Chrome (any version)
- Safari 5 (or higher)

itemize In short, any user who's system can adequately run any of the above applications will be able to utilize GUS.

### **2.1.3 User Characteristic Constraints**

GUS carries with it the inherent danger of private data loss; just like every other online application that deals with private information. As such, any user who uses the system must be aware of the security risks associated with doing so. To ensure that our clients understand this, GUS will notify new users of these risks during registration.

## **2.2 Stakeholder Concerns**

### **2.2.1 StakeHolder Concerns**



# Chapter 3

## System and Software Architecture

### 3.1 Users Architectural View

#### 3.1.1 User's View Identification

#### 3.1.2 User's View Representation and Description

### 3.2 Developer's Architectural View

#### 3.2.1 Developer's View Identification

#### 3.2.2 Developer's View Representation and Description

Object Model

Dynamic Model

#### 3.2.3 Developer's Architectural Rationale

### 3.3 Consistency of Architectural Views

#### 3.3.1 Developer's Viewpoint Detailed Software Design

#### 3.3.2 Component Dictionary

#### 3.3.3 Component Detailed Design

Detailed Design for Component/Entity: Name of Component

Detailed Design for Component/Entity: Name of Component

Detailed Design for Component/Entity: Name of Component

# Chapter 4

## Requirements Traceability

### 4.0.1 Requirements Traceability

In the case of GUSPHP, all requirements originated from the customer, Dr. Jeffery. Though extensive discussion both with Dr. Jeffery and conversations as a team, a general set of requirements were eventually agreed upon. These requirements formed the core goals that GUS strives for.

Once a set of requirements had been agreed upon, actual implementation details were then conferred upon as a team. These details included use-case creation and division of labor. Since GUS is a web-based project, each requirement formed a different segment of functionality of the overall project. With this being the case, each team member is assigned a specific requirement. Their responsibility is to ensure that their part was completed by the end of the semester.

Development can be traced by tickets present on the GUS Trac server. Trac is an online project management tool that easily shows team progress through a project roadmap. When properly updated, the roadmap shows a detailed view of the development cycle by utilizing a ticket system. The GUSPHP team leader will add tickets - or small segments of work - for each team member. These tickets will then provide a work log for each employee, as well as reflecting progress towards a complete implementation.

**Part II**

**Systems and Software Requirements  
Specification (SSRS)  
for  
Groups in a University Setting**

# Chapter 5

## Overall Description

### 5.1 Product Perspective

Gus is an independent software system, as it does not directly integrate with a larger system. However, GUS does draw data from external sources, such as personal information databases, and needs to be integrated with a web server in order to be readily accessible.

### 5.2 Product Functions

1. Simplifying tasks to leaders of groups, such as:
  - (a) Sending notifications to group members, prospective members, former members, and interested community members (email)
  - (b) Sending information (files) to group members via email or download link
  - (c) Managing a group-wide calendar of events
    - i. track volunteers, attendees, and contributors
    - ii. suggest potentially beneficial services other groups could provide related to the event
    - iii. provide a calendar of events that includes events from other groups that members would want to attend (like marching band if half of LCM are in the marching band.)
    - iv. keeping track of who is responsible for bringing / doing what at an event



- (d) Automatically generating:
  - i. Contact information (contact sheets, phone directories)
  - ii. Website with updated contact, group, event, and customized information
  - iii. Organization charts
  - iv. Graphical relationships between groups
  - v. Fees, dues, and expenses notifications
  - vi. Event reminders
- 2. Consolidating information for members, former members, potential members (and parents) of groups:
  - (a) Common location of group information
  - (b) Searching existing groups
  - (c) Tying together existing groups (even suggesting similar groups)
  - (d) Personalized emails regarding changes/updates
  - (e) Outstanding expenses or reimbursements
  - (f) Reliable (i.e., automatically updated):
    - i. Group contact information
    - ii. Group event information
  - (g) Transcript of verified group activity (for use with service-learning classes, and proof of volunteerism for potential employers)
  - (h) Supplementing Vandal Friday with emails to prospective high school Seniors
- 3. Getting member input through: forums, project managers, surveys, and polls
- 4. Payment processing and sponsorship collection
- 5. 4. Recruitment and advertising for groups, volunteer / paid opportunities, services provided, possibly a bartering tool

## 5.3 User Characteristics

Gus should be easy for any user to understand with a brief explanation and intuitive enough for an uninitiated user to figure out by looking through the options. Basic computer use skills and a simple conceptual explanation should be enough for every day usage.

## **5.4 Constraints**

GUS must meet privacy policies as they apply to both the University of Idaho and social networking sites. GUS must be able to interface with outside database servers (such as the Center for Volunteerism's database, UI's career seeker site, common social networks, and parent groups of university groups). Member's activities and group's activities must be audited for accuracy and safety. The languages used to program GUS will be primarily, HTML, CSS, and PHP for the user interface, C++ for the interface between the user interface (which will implement security and complex business rules), and the database, and SQL for the database. The networking protocols will be TCP/IP and Open MP / MPI will be used to enhance parallel operation. The system will have personal information for over 5,000 students, so confidentiality is of the utmost importance.

## **5.5 Assumptions and Dependencies**

The software system should run like a web-app and need not be downloaded by users. It is assumed that users will be running Internet Explorer, Fire Fox, or another popular web browser. The server for the system is expected to run a UNIX operating system.

## **5.6 System Level (Non-Functional) Requirements**

### **5.6.1 Site dependencies**

GUS will require a server that can support 1,000 concurrent users. The database must store the information, interests, and activities of approximately 5,000 external users, 5,000 students and 200 groups.

### **5.6.2 Safety, security and privacy requirements**

GUS contains the personal information of over 5,000 users security should be integrated into every facet of this program. The privacy criteria for this system must reflect privacy policies that apply to the University of Idaho, and the security criteria for this system must reflect the need to secure over 5,000 users from identity theft and potential defamation of character.

### **5.6.3 Performance requirements**

1. The number of simultaneous users to be supported are: 1,000.
2. Supported information ranges from text to files to streaming video.
3. 95% of the transactions shall be processed in less than half a second.

### **5.6.4 System and software quality**

Gus must perform all required functions, behave consistently and correctly, be easily corrected, running between 5:30 am all day to 1:30 am be easily adaptable, test-driven, and easy to use.

### **5.6.5 Packaging and delivery requirements**

The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer via Internet download. The final, edited version of the above documents will accompany the final, accepted version of the executable system.

### **5.6.6 Personnel-related requirements**

The system under development will require a graduate student system-level administrator to maintain the system.

### **5.6.7 Training-related requirements**

No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.

### **5.6.8 Logistics-related requirements**

A server will be required to maintain the software system. The user will be required to have an Internet connection.

### **5.6.9 Precedence and criticality of requirements**

1. Maintaining confidentiality and privacy of PII
2. This system must be reliable enough for users to not give up on it
3. All other features are less important than the first two and equally important

# **Chapter 6**

## **Specific Requirements**

### **6.1 External Interface Requirements**

#### **6.1.1 Hardware Interfaces**

The system will require a server and secure networking abilities.

#### **6.1.2 Software Interfaces**

The system will require an interface to interact with emailing systems, databases, and authentication servers.

#### **6.1.3 User Interfaces**

The system will require user interfaces for non-university users (prospective students, community members, alumni, parent groups, etc.), students, officers, and staff/faculty.

#### **6.1.4 Other Communication Interfaces**

GUS will interface with the university career seeking site and social networking sites.

Table 6.1: Hardware Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats
HTTP Server	Dedicated Server or VPS / Client	This device is responsible for serving HTML content (and other content) to clients. Preferably Apache2.	All	Requires a server-capable machine	N/A
VPS or Dedicated Server	NA	A VPS or a Dedicated Server, preferably running a pre-configured Linux distribution such as Fedora or Ubuntu.	All	Electricity, high-speed Internet connection	N/A

Table 6.2: Software Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats
SQL Server	Dedicated Server or VPS / Client	Works in conjunction with HTTP server to provide data.	All	Requires a server-capable machine	N/A
PHP5	HTTP Server / Client	Provides computational power so tasks that serve HTML content via apache can be completed.	Requires a server capable of running PHP5.	Electricity, high-speed Internet connection	N/A

Table 6.3: User Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats
Website	HTTP Server/Client	Allows user to interact with the service	All	HTTP Server	Web
Cell phone	Cellphone	Receive text-messages	All	HTTP Server	Text

Table 6.4: Other Communication Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats

## 6.2 System Features

### 6.2.1 Use Case Diagrams

### 6.2.2 Use Case Diagrams

insert 1+ use case diagrams here

### 6.2.3 System feature 1: Officer Management (Administration Module)

#### Add Officer

- **Actors:** Group Administrator
- **Goals:** Administrator will add an officer to the group
- **Preconditions:** Administrator is logged onto to the group's administration page
- **Summary:** Related use cases: modify officer access
- **Steps:**
  1. Administrator clicks add officer button
  2. Administrator types in officer's username
  3. Administrator selects the officer's title/role
  4. Administrator modifies officer's access
  5. Administrator selects save and exit
- **Alternatives:** Administrator cancels out of use case
- **Postconditions:** the selected user is given officer status in the group

#### Modify Officer Access

- **Actors:** Group Administrator
- **Goals:** Administrator will change the access given to an officer
- **Preconditions:** Administrator is logged onto to the group's administration page
- **Summary:** Related use cases:

- **Steps:**
  1. Administrator clicks modify officer access button
  2. Administrator selects officer / role to be modified
  3. Administrator selects type of access to be modified
  4. Administrator selects save and exit
- **Alternatives:** Administrator cancels out of use case
- **Postconditions:** The selected officer(s) access is modified

### **Remove Officer**

- **Actors:** Group Administrator
- **Goals:** Administrator will remove an officer from the group
- **Preconditions:** Administrator is logged onto to the group's administration page
- **Summary:** Related use cases:
- **Steps:**
  1. Administrator clicks remove officer button
  2. Administrator selects officer
  3. Administrator selects save and exit
- **Alternatives:** Administrator cancels out of use case
- **Postconditions:** the selected user is removed as a group officer

### **Change Group Password**

- **Actors:** Group Administrator
- **Goals:** Administrator will change the password to access the group page
- **Preconditions:** Administrator is logged onto to the group's administration page
- **Summary:** Related use cases:
- **Steps:**
  1. Administrator clicks change group password button
  2. Administrator types in the new password twice
  3. Administrator selects save and exit
- **Alternatives:** Administrator cancels out of use case
- **Postconditions:** the group password is changed



### Edit Forum Priveleges

- **Actors:** Group Administrator
- **Goals:** Administrator will select who can make changes to forums
- **Preconditions:** Administrator is logged onto to the group's administration page
- **Summary:** Related use cases:
- **Steps:**
  1. Administrator clicks forum priveleges button
  2. Administrator selects who's priveleges will be changed
  3. Administrator selects the the priveleges
  4. Administrator saves and exits
- **Alternatives:** Administrator cancels out of use case
- **Postconditions:** the user's priveledges are changed

## 6.2.4 System feature 2: Group Management (Officer Module)

### Add Member

- **Actors:** Officer
- **Goals:** Officer will add a member to the group
- **Preconditions:** Officer is logged onto to the group's officer page
- **Summary:** Related use cases:
- **Steps:**
  1. Officer clicks add member button
  2. Officer types in the member's username
  3. Officer selects save and exit
- **Alternatives:** Officer cancels out of use case
- **Postconditions:** the selected user is given membership status in the group

## Remove Member

- **Actors:** Officer
- **Goals:** Officer will remove member from the group
- **Preconditions:** Officer is logged onto to the group's officer page
- **Summary:** Related use cases:
- **Steps:**
  1. Officer clicks remove member button
  2. Officer selects member
  3. Officer selects save and exit
- **Alternatives:** Officer cancels out of use case
- **Postconditions:** the selected user is removed from the group

## Add Event

- **Actors:** Officer
- **Goals:** Add an event to fit officer specifications
- **Preconditions:** Officer is logged onto to the group's officer page
- **Summary:** Related use cases: Modify Event
- **Steps:**
  1. Officer clicks Events button
  2. Officer clicks Add New Event
  3. Officer adds a title
  4. Officer chooses a date, time, and recurrence
  5. Officer checks the post to site option
  6. Officer checks the email notifications to members option
  7. Officer modifies the message to be sent
  8. Officer selects save and exit
- **Alternatives:** Officer cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** Event has been added to fit the officer's specifications

## Modify Event

- **Actors:** Officer
- **Goals:** modify (add, edit, remove) an event on the calendar
- **Preconditions:** Officer is logged onto to the group's officer page
- **Summary:** Related use cases: Add Event, Remove Event
- **Steps:**
  1. Officer clicks Events button
  2. Officer clicks Modify Event button
  3. Officer modifies event information
  4. Officer sends notifications and has the site updated as needed
  5. Officer selects save and exit
- **Alternatives:** Officer cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** Event has been modified to fit the officer's specifications

## Remove Event

- **Actors:** Officer
- **Goals:** remove an event on the calendar
- **Preconditions:** Officer is logged onto to the group's officer page
- **Summary:** Related use cases: Add Event, modify
- **Steps:**
  1. Officer clicks Events button
  2. Officer clicks Remove Event button
  3. Officer sends notifications and has the site updated as needed
  4. Officer selects save and exit
- **Alternatives:** Officer cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** Event has been removed from calendar and site

## Send Group Email

- **Actors:** Officer
- **Goals:** the officer will be able to send email to the group
- **Preconditions:** Officer is logged onto to the group's email page
- **Summary:** Related use cases: Add Event, modify event
- **Steps:**
  1. Officer clicks the compose email button
  2. Officer composes an email
  3. Officer selects which subgroup the email goes to
  4. Officer clicks the send button
- **Alternatives:** Officer cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** email has been spellchecked, and sent. Officer is returned to the group's email page

## Modify Website

- **Actors:** Officer
- **Goals:** to generate a web site of information about a group
- **Preconditions:** Officer is logged onto to the group's officer page, website is already posted and linked to GUS
- **Summary:** Related use cases:
- **Steps:**
  1. Click Web Site button
  2. Click Web site options
  3. Modify information on website
  4. Click Preview button
  5. Select save and exit
- **Alternatives:** Officer cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** the website is published on the website, and automatically linked to from its supergroup

## 6.2.5 System feature 3: User Menu

### Edit Email

- **Actors:** User
- **Goals:** user will be able to send emails to other GUS users
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks Email button
  2. User clicks on the Drafts folder
  3. User clicks on the email to be edited
  4. User edits email
  5. User sends email, and is returned to the drafts page
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** Email is sent and user is returned to drafts page

### Edit Major

- **Actors:** User
- **Goals:** user will be able to change personal attributes
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks MyProfile button
  2. User makes changes to personal fields
  3. User selects save and exit
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** User's profile and personal page is updated, and user is returned to their home page

## Edit Personal Page

- **Actors:** User
- **Goals:** user will be able to change appearance and content of personal page
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases: Change Major
- **Steps:**
  1. User clicks MyPage button
  2. User edits personal information
  3. User selects privacy settings; to include which user's and groups they want seeing their information
  4. User selects what color theme / appearance they want for their page
  5. User selects what items and information they want on their page
  6. User previews their page and clicks save
- **Alternatives:** User can move through page changes non-sequentially or cancel out
- **Postconditions:** User's personal page is updated

## Join Group

- **Actors:** User
- **Goals:** user will be able to request to join group
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User enters group name
  2. User clicks on the join button
  3. User identifies which subgroups they want to join
  4. User selects save and exit
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** user's request is auto approved, or request sent to group, user is added to group (if approved), and user is returned to the group's page

## Leave Group

- **Actors:** User
- **Goals:** user will be able to leave group
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks MyGroups button
  2. User selects the group to drop
  3. User clicks the drop button
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** User is dropped from the group, the group is removed from the user page and the user is returned to the MyGroups page

## Send Email

- **Actors:** User
- **Goals:** user will be able to send emails to other GUS users
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks Email button
  2. User clicks on the Drafts folder
  3. User clicks on the email to be edited
  4. User edits email
  5. User sends email, and is returned to the drafts page
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** Email is sent and user is returned to drafts page

## Send Text

- **Actors:** User
- **Goals:** user will be able to send texts to other GUS users
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks text button
  2. User types in the recipient
  3. User types the message
  4. User clicks send
  5. User makes any required changes, recipient, spelling (if required) go to step 4
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** text is sent and user is returned to previous page

## Join Event

- **Actors:** User
- **Goals:** user will be able to sign up for event
- **Preconditions:** user is logged onto GUS, and at the group's site
- **Summary:** Related use cases:
- **Steps:**
  1. User hovers over the calendar, the system displays the user's calendar next to the group calendar
  2. User hovers over the event to join, the system displays event details
  3. User right-clicks on the event, and selects will-be-there, or may-be-there
  4. User may enter a message
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** User is added to the event's guest list, the event is added to the user's calendar, and the user is returned to the group page



### Leave Event

- **Actors:** User
- **Goals:** user will be able to drop an event
- **Preconditions:** user is logged onto GUS
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks on MyCalendar
  2. User right-clicks on the event
  3. User selects drop
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** User is removed from the event's guest list, the event is removed from the user's calendar, and the user is returned to the calendar page

### Sign Up

- **Actors:** User
- **Goals:** user will be able to sign up for GUS
- **Preconditions:** user is at GUS page
- **Summary:** Related use cases: Modify personal information, modify personal page
- **Steps:**
  1. User clicks on Sign Up
  2. User completes steps for modifying their page
  3. User selects groups to join
- **Alternatives:** User cancels out of use case by clicking on a different navigational area or clicking Cancel
- **Postconditions:** User is added to GUS, and relevant users/groups are notified

## 6.2.6 System feature 4: Forum Module)

### Add / Post Topic

- **Actors:** User

- **Goals:** User will add a post or start a topic
- **Preconditions:** User is logged in, and at the group's forum page
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks on topic to post to, or clicks AddTopic button
  2. User completes a post (the default title is the first line of the post)
  3. User clicks post button The system spell-checks the post, and truncates it if it's too long, it also validates the post, and returns an error message (if needed)
  4. If needed user makes corrections and repeats step 3
- **Alternatives:** user cancels out of use case
- **Postconditions:** the post is sent to the group for validation, and the user is returned to the topic page

### **Remove / Post Topic**

- **Actors:** User
- **Goals:** User will remove a post or topic
- **Preconditions:** User is logged in, and at the group's forum page
- **Summary:** Related use cases:
- **Steps:**
  1. User clicks on topic or the post to remove
  2. User clicks remove button
- **Alternatives:** User cancels out of use case
- **Postconditions:** If user has the correct permissions, the post or topic is removed, otherwise it is not

### **Edit / Post Topic**

- **Actors:** User
- **Goals:** User will edit a post or a topic
- **Preconditions:** User is logged in, and at the group's forum page
- **Summary:** Related use cases: add post/topic

- **Steps:**
  1. User right-clicks on topic or post to be edited
  2. User selects edit
  3. User modifies the post (the default title is the first line of the post)
  4. User clicks post button, the system spell-checks the post, and truncates it if it's too long, it also validates the post, and returns an error message (if needed)
  5. If needed user makes corrections and repeats step 3
- **Alternatives:** user cancels out of use case
- **Postconditions:** the post is sent to the group for validation, and the user is returned to the topic page

## 6.2.7 System feature: Group Files

### 6.2.8 Upload File

- **Actors:** Logged in Users
- **Goals:** User will upload a file to a group from their computer
- **Preconditions:** User is a member of the group they want to upload a file to
- **Summary:** The file uploading functionality allows users to share files with groups they are a member of. Uploaded files are stored on the server and can be viewed and downloaded by other users with sufficient permission levels.
- **Steps:**
  1. User navigates to the Upload page
  2. User clicks the browse button to bring up the file browser
  3. User selects a file from their computer to upload and clicks the Open button in the file browser
  4. User selects a group from the list of groups they are a member of using the drop-down menu on the Upload page
  5. User clicks the Upload button to upload their file
- **Alternatives:**
  - \* User can cancel out of process by hitting their browser's back button at any step before 5

- \* User can do steps 2-4 in any order, as long as step 3 happens immediately after step 2
- \* If the user tries to upload a filetype that isn't allowed, or a file that is too large, or if the user does not specify a file, they will get an error
- **Postconditions:** The file chosen by the user is uploaded to the server, and inserted into the database
- **Related use cases:** Download File, Delete File, Change File Permissions

### 6.2.9 Download File

- **Actors:** Logged in Users
- **Goals:** User will download a file from a group to their computer
- **Preconditions:** User is a member of the group they want to download a file from or user is the file's original uploader
- **Summary:** The downloading functionality allows users to download a copy of an uploaded file from the server. Users are only able to download files they have permission to view.
- **Steps:**
  1. User navigates to the Docs page
  2. User clicks the thumbnail of the file they want to download
  3. User clicks the download button
  4. User's browser downloads the file
- **Alternatives:**
  - \* User can cancel out of process by hitting their browser's back button at any step before 3
  - \* If the user tries to download a file they don't have permission to view, they will receive an error
- **Postconditions:** The file is downloaded from the server via the user's browser
- **Related use cases:** Upload File, View File Online, Delete File, Change File Permissions

### 6.2.10 View File Online

- **Actors:** Logged in Users
- **Goals:** User will view a file inside their browser rather than download it to their computer to view
- **Preconditions:** User is a member of the group that the file was uploaded to or user is the file's original uploader
- **Summary:** Users are able to view some file types online. Images and pdfs can be viewed within the user's browser rather than downloading them to the user's local machine first. A user needs the same permission to view files as they do to download files.
- **Steps:**
  1. User navigates to the Docs page
  2. User clicks the thumbnail of the file they want to view online
  3. User clicks the view button
  4. User's browser displays the file
- **Alternatives:**
  - \* User can cancel out of process by hitting their browser's back button at any step before 3
  - \* If the user tries to view a file they don't have permission to view, they will receive an error
- **Postconditions:** The file is displayed within the user's browser
- **Related use cases:** Upload File, Download File, Delete File, Change File Permissions

### 6.2.11 Delete File

- **Actors:** Group administrators, Users that have uploaded files, group officers
- **Goals:** User will delete an uploaded file from the database and server
- **Preconditions:** User has permission to manage files of the group that the file was uploaded to or user is the file's original uploader
- **Summary:** Users are able to delete files they have uploaded from the server. Group officers and administrators are able to delete any file uploaded to their group.

- **Steps:**
  1. User navigates to the Docs page
  2. User clicks the thumbnail of the file they want to view online
  3. User clicks the delete button
  4. Server deletes the file and the user is redirected to the Docs page
- **Alternatives:**
  - \* User can cancel out of process by hitting their browser's back button at any step before 3
  - \* If the user tries to delete a file and they don't have sufficient permissions within the group or as the file uploader, they will receive an error
- **Postconditions:** The file is deleted from the server
- **Related use cases:** Upload File, Download File, View File Online, Change File Permissions

### 6.2.12 Change File Permissions

- **Actors:** Group administrators, Users that have uploaded files, group officers
- **Goals:** User will change permissions for an uploaded file
- **Preconditions:** User has permission to manage files of the group that the file was uploaded to or user is the file's original uploader
- **Summary:** Each file has a permission level required to view the file. If a user has insufficient permissions, they won't be able to view the file. By changing a file's permission level, you can control who is able to view the file. Users can change the permissions for any file they've uploaded, and administrators and group officers can change the permissions for any file uploaded to their group.
- **Steps:**
  1. User navigates to the Docs page
  2. User clicks the thumbnail of the file they want to view online
  3. User clicks the modify button
  4. User chooses a new permission level for the file
  5. User clicks the submit button
- **Alternatives:**

- \* User can cancel out of process by hitting their browser's back button at any step before 5
- \* If the user tries to change permissions for a file and they don't have sufficient permissions within the group or as the file uploader, they will receive an error
- **Postconditions:** The file's permissions are changed in the database
- **Related use cases:** Upload File, Download File, View File Online, View File Online





# Chapter 7

## Requirements Traceability

7.1 ...

7.1.1 ...

7.1.2 ...

7.1.3 ...

7.1.4 ...

7.1.5 ...

7.1.6 ...

7.2 ...

7.2.1 ...

7.2.2 ...

7.2.3 ...

7.2.4 ...

7.2.5 ...

7.2.6 ...

7.3 ...

## Part III

...

# Appendix A

...

# Appendix B

...