NYU | TANDON SCHOOL OF ENGINEERING

Computer Science and Engineering

Sharit

System Requirements Specification

Version 2.0

Document Number: SRS-002

Team A6

Allen Zheng (az1010)
Hui Huang (hh1128)
Kenneth Liang (kl1792)
Warlon Zeng (wz634)

REVIEW AND APPROVALS

Printed Name and Title	Function (Author, Reviewer, Approval)	Date	Signature
Allen Zheng	Author	March 7, 2016	Allen Zheng
Hui Huang	Author	March 7, 2016	Hui Huang
Kenneth Liang	Author	March 7, 2016	Kenneth Liang
Warlon Zeng	Author	March 7, 2016	Warlon Zeng

REVISION LEVEL

Date	Revision Number	Purpose
March 7, 2016	Version 1.0	Initial Release
March 23, 2016	Version 2.0	Added sections 7.2 & 7.3

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to specify the software requirements for the website Sharit. This document will outline the format, design, functionality, and purpose of Sharit. This document is intended for the client, the designers, and the developers.

2. SCOPE

2.1 Identification

This document is the Software Requirements Specifications for Sharit. This document is revision 2.0, number 2.

2.2 Bounds

The client's idea stems from the need to share files/documents in an organized, coherent manner that is also accessible to many others who you want to share with. There are services out there that do this job but is disorganized and does not provide any kind of feedback system.

Students can use this service to post notes for the underclassmen and receive feedback on their work. Students will be able to collaborate and create notebooks that can be shared with other students in the class. This will benefit those who may not have been able to attend school for that day or those who want to use this platform to study for an upcoming exam. The key to our platform is organization of uploaded files; we want people to be able to look up the file they want and quickly access it or ask the owner of the file to grant permission to access.

The project is a website that facilitates conversations and file sharing between people from a common organization. The layout will be a clean and simple format for easy intuitive understanding. Organizations will have a domain that will be split into smaller pieces, such as the subdomain and topics. Users will will have permissions regarding which organization domain and subsequent subdomains they may access.

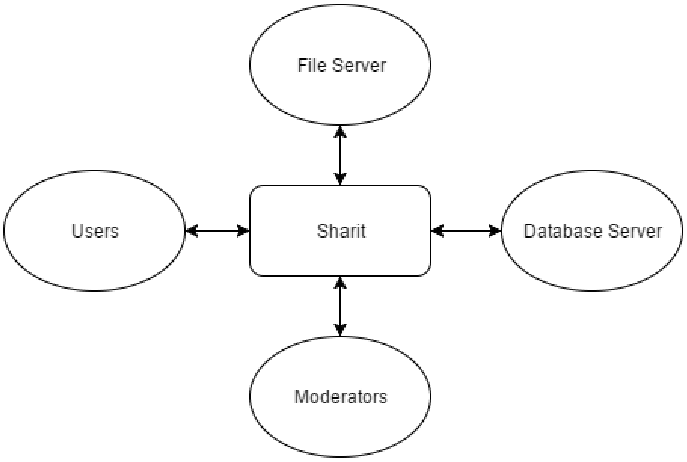
2.3 Objectives

The priority is to release the Software Project Management Plan. The project will be utilizing an incremental life cycle with the deliverables listed below:

Project Proposal	February 17, 2016
Software Requirements Specifications	March 21, 2016
Software Project Management Plan	March 30, 2016
Software Analysis Specification	April 11, 2016
Software Design Document	April 25, 2016

3. OVERALL SYSTEM OVERVIEW

3.1 CONTEXT DIAGRAM



3.2 ADDITIONAL DESCRIPTIVE ITEMS

a. Product Functions:
   i. User login and registration to ensure that maintain permissions and file access for each user.
   ii. Files can be uploaded and edited by users and is maintained by a database.
   iii. Communication systems between users with access to same files.
   iv. Comment system for users to leave behind on files
   v. Database tracks permissions for every file and user.

b. User characteristics: This product is meant to be used by college students as well as staff. It can be used to serve corporations as well.

c. Constraints: Access to the system is maintained by login. Files are also managed by the user who uploaded it. This permission can be changed only by the user and may provide difficulties in sharing to a large group of people.

d. Assumptions: This product will be programmed in C++ and Postgres. Any changes will not impact the product largely.

e. Requirements subset: The product will be broken down into three phases. Website front-end, database management, Website back-end. Database management and the backend are the most important while the design of the website is secondary.

4. DOCUMENT OVERVIEW

The rest of this document contains referenced documents, business requirements, specific requirements of Sharit, system test plan requirements, quality guidelines for this document, and an overview of the changes in this document. A general guideline can be seen in the table of contents.

5. REFERENCE DOCUMENTS

Team A6 Project Proposal, Version 1.0, March 7, 2016.

6. BUSINESS REQUIREMENTS

6.1 TECHNOLOGY

To deliver fast, reliable, and high quality website service, recent dated versions of software in servers, databases, and programming languages should be used. These softwares, in combination with development tools, such as git, google cloud, or any other preference tool, will help build a website to meet users' expectations.

6.2 ECONOMICS

1. Domains will be sold to organizations.
2. Revenue comes from selling disk space to organizations

6.3 REGULATORY AND LEGAL

1. No regulatory or legal requirements at this time.

6.4 MARKET CONSIDERATIONS

Since registration is entirely dependent on having a @nyu.edu email address, the targeted group of consumers will be university students. To the students, the service will be free of charge. The project will proceed to follow a subscription model, in which the school will pay for users' subscription, and will appear free to users.

6.5 RISKS AND ALTERNATIVES

1. Not enough users signing up
    - 1.1. Can potentially expand to more schools outside of NYU
2. Not enough content being produced
    - 2.1. Can offer monetary incentives for students to contribute as a start-up and later retract offers after a self-sufficient number of contributions generated
3. Work that are not acceptable to be shared

3.1. Can be removed by moderators

6.6 HUMAN RESOURCES AND TRAINING

1. Student programmers
2. Full stack training
    2.1. HTML
    2.2. CSS
    2.3. Javascript
    2.4. PHP/Python - general purpose language
    2.5. postgres/oracle - database system
    2.6. C++ - web server
    2.7. Windows - operating system
    2.8. Git - version control

7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS)

7.1 FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS

1.0 The website will provide functions for the user to login and register as well as a profile
    1.1. A user will be able to register using their student or company email
    1.2. Existing users will be able to log in using the credentials they registered with
    1.3. Each user will have a profile with information such as name and school/company
    1.4. Options will be provided to change basic information
2.0 Each organization will be allowed to create a domain on the server

2.1. Domains are only accessible to users in the organization or with permission

2.2. Domains will have administrators or moderators

2.3. Admins can modify aspects of the domain as well as grant permissions

2.4. Each domain can have subdomains that correspond to different parts of the organization

3.0 Users will be able to upload and download files

3.1. A user will be able to create a thread where the file will be uploaded

3.2. Threads will be part of a subdomain

3.3. A user that is part of the subdomain may choose to view or download the file

4.0 Each thread will support feedback/comments

4.1. A thread can be upvoted/downvoted to show popularity within a domain

4.2. Users can leave comments on the thread

4.3. Other users can respond to a previous user's comment

4.4. Users will be able to upvote/downvote other user's comments

5.0 A search will be provided to look up specific information in the system

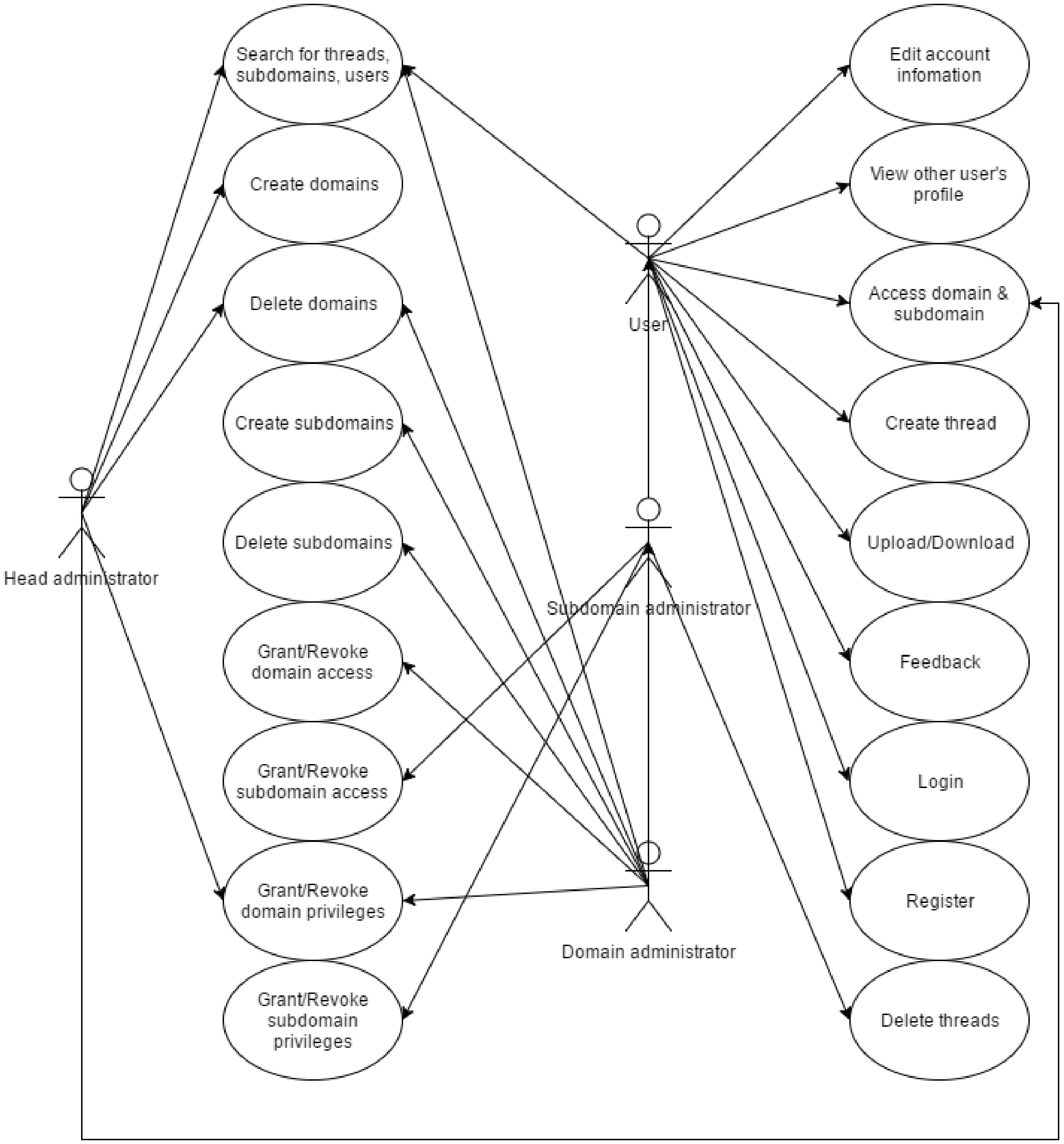
5.1. Users can look up specific files/threads in the domain

5.2. Users can search for other users and view their profile

5.3. Users can search for domains or subdomains to join

7.2 REQUIREMENT USE CASES

7.2.1 USE CASE DIAGRAMS



7.2.2 USE CASE DESCRIPTIONS

Register		
Description	Newcomers could register for an account on the website.	
Pre-Conditions	Registering for an account requires a @nyu.edu email address.	
Flows	**Basic or Normal Flows**	1. Registrants registers for an account using @nyu.edu email address. 2. After registering, the server will send an email confirmation (click link to verify) to the registered email address. 3. Email address will be verified by server and registration complete.
	Alternative Flows	1. Registrant attempts to register with an invalid email address - not @nyu.edu 2. Registrant account creation denied
Post Conditions	Registrant becomes registered, functionality of the website is granted.	
Special Requirements	Email address used to register must be @nyu.edu.	
Extension Points	Invalid email addresses will result in rejection to account creation.	

Login		
Description	A user who has previously registered will be able to log in using the registered username and password.	
Pre-Conditions	The user must have been previously registered.	
Flows	**Basic or Normal Flows**	1. User entered login information. 2. Website validates with database and the information is in the system. 3. The user is redirected to the homepage.
	Alternative Flows	1. User enters invalid credentials. 2. User is prompted to enter again. 3. Process repeats until valid credentials have

		been provided.
Post Conditions	The user is logged in and can use the website and its functionalities.	
Special Requirements	The system will create a session to store the user's information.	
Extension Points	1. User enters invalid credentials - Prompt user to reenter credentials 2. User forgot the login information - Has option to recover via email	

Edit account information		
Description	Users can edit their account information	
Pre-Conditions	Must login to edit account information	
Flows	**Basic or Normal Flows**	1. User specifies which fields they desire to edit. 2. User then edit desired fields. 3. User then saves the information and server will acknowledge saved changes.
	Alternative Flows	1. User specifies which fields they desire to edit. 2. User provides invalid information or information that the server does not know how to process. 3. User edited fields will not be saved and error messages will be displayed.
Post Conditions	User account information is updated with saved changes.	
Special Requirements	All updated changes will be saved into the database.	
Extension Points	If a user tries to enter invalid information, such as wrong formatting, error message(s) pertaining to incorrect fields will be displayed.	

View other users' profile

Description	A user can view the information of other users	
Pre-Conditions	The user must be logged in and within the same domain	
Flows	**Basic or Normal Flows**	1. User is redirected to the other user's profile page 2. User is able to view basic information like name, school, and threads 3. User can see account activity
	Alternative Flows	There are no alternate flows.
Post Conditions	The user is able to view another user's basic information and account activity which includes: comments, threads and upvotes	
Special Requirements	The user must be logged in and within the same domain as the user they are trying to view.	
Extension Points	If the user tries to view the profile of another user who has deleted their account, an error message will be displayed.	

Access domain and subdomain		
Description	Head administrators and users can access domains that expand to further subdomains and threads.	
Pre-Conditions	The groups must be logged in and authorized.	
Flows	**Basic or Normal Flows**	1. User chooses which domain to enter 2. User enters domain 3. User can see subdomains and, if have access to, its threads
	Alternative Flows	1. User does not have access to domain 2. User enters a domain they don't have access to 3. Access denied
Post Conditions	User can view accessed subdomains and threads of entered domain.	
Special Requirements	User must have access to enter desired domain and subdomain	

Extension Points	If user tries to enter a domain they do not have permission to, they will not see its domain contents: subdomains and its threads.

Create thread		
Description	Users can create threads within subdomains.	
Pre-Conditions	The user must be authorized within the subdomain and logged in.	
Flows	**Basic or Normal Flows**	1. User is logged in and is authorized in the subdomain that they want to post to. 2. User browses to the subdomain 3. Use posts a thread
	Alternative Flows	1. User is logged in but is not authorized in the subdomain. 2. User browses to the subdomain. 3. User cannot post a thread.
Post Conditions	There is now an option to upload a file into the subdomain and leave comments.	
Special Requirements	User must be logged in and received permission from the administrator.	
Extension Points	If the user does not have authorization, they will not see the create thread button or be able to view any threads in the subdomain.	

Upload/Download		
Description	Users can upload and/or download file(s) to thread	
Pre-Conditions	The user must be authorized within the subdomain and logged in.	
Flows	**Basic or Normal Flows**	1. User finds which thread to upload/download to/from 2. User uploads/downloads to/from a thread (comment-link) 3. User can post new thread (thread-link)

	Alternative Flows	1. User uploads something inappropriate 2. File(s) uploaded 3. File(s) deleted by moderator
Post Conditions	User will successfully download file from thread or upload file(s) to thread (comment-link) or make a new thread (thread-link).	
Special Requirements	User uploaded file(s) will be saved to database.	
Extension Points	If user tries to upload something inappropriate, the file(s) will be uploaded, but will be quickly brought down by moderators.	

Feedback		
Description	The user can leave a comment or upvote/downvote in a thread.	
Pre-Conditions	The user must be logged in and authorized within the subdomain.	
Flows	**Basic or Normal Flows**	1. User navigates to a subdomain. 2. User finds a thread that they want to leave feedback for. 3. User leaves feedback.
	Alternative Flows	1. User navigates to a subdomain. 2. User does not have authorization within the subdomain. 3. User is unable to see threads or leave feedback.
Post Conditions	The feedback that the user leaves will be visible to other users.	
Special Requirements	User must be authorized within the subdomain	
Extension Points	There will be a character limit on the comments that the user leaves.	

Search for threads, subdomains, users	
Description	Head and domain administrators, as well as users, can search for existing threads, subdomains, and/or users

Pre-Conditions	Must be logged in and have authorization.	
Flows	**Basic or Normal Flows**	1. Specify which thread, subdomain, and/or user to search for 2. Searches for specified item 3. Item found and displayed
	Alternative Flows	1. User chooses which thread/subdomain/user to search for 2. Specified item does not exist 3. Error message displayed
Post Conditions	Request of item will be delivered if it exists and request party has access to.	
Special Requirements	In order to use the search functionality, head and domain administrator, as well as user, will require access beforehand.	
Extension Points	If search functionality is used without proper access rights, error message will be displayed	

Grant/revoke domains, subdomains access/privilege		
Description	Administrators will be able to grant or revoke permission/access to users or other administrators	
Pre-Conditions	The administrator must be registered in the database	
Flows	**Basic or Normal Flows**	1. User requests privilege from administrators. 2. Administrator receives the request. 3. Administrator grants request.
	Alternative Flows	1. User requests privilege from administrators. 2. Administrator receives the request. 3. Administrator denies request.
Post Conditions	The user or other administrators now have privilege/access to a domain or subdomain.	
Special Requirements	Authorization must be granted from a current administrator.	
Extension Points	1. Access to a subdomain grants access to all threads in that sub-domain	

	2. Privilege to to a domain or subdomain means you are the administrator

Create domains, sub-domains		
Description	Administrators can create domains and subdomains.	
Pre-Conditions	Administrators must be logged in.	
Flows	**Basic or Normal Flows**	1. Administrator logs in. 2. Administrator creates a domain. 3. Domain is now visible.
	Alternative Flows	1. Administrator logs in. 2. Administrator creates a subdomain. 3. Subdomain is now visible.
Post Conditions	The domain or subdomain can be accessed by those who have permission,	
Special Requirements	Only head administrator can create domains. Domain administrator can create subdomains.	
Extension Points	Head administrator has the most power, followed by domain administrator, followed by subdomain administrator.	

Delete domains, subdomains, threads		
Description	Administrators can delete domains, subdomains, and threads.	
Pre-Conditions	Administrators must be logged in.	
Flows	**Basic or Normal Flows**	1. Administrator logs in 2. Administrator deletes domain
	Alternative Flows	1. Administrator logs in 2. Administrator deletes subdomain
		1. Administrator logs in 2. Administrator deletes a thread

Post Conditions	The domain/subdomain/thread is deleted.
Special Requirements	Only head administrators and domain administrators can delete domains. Only domain administrators and subdomain administrators can delete subdomains.
Extension Points	Deleting a domain deletes the domain and all subdomains within. Deleting a subdomain deletes the subdomain and all threads within.

7.3 NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS

7.3.1 SYSTEM CAPABILITIES, CONDITIONS, AND CONSTRAINTS

The project requires a browser that supports HTML and CSS. This requires users to have extremely basic computer requirements as well as a browser and a working internet connection. The userbase is currently limited to people within New York University. In order to download files, users must also have sufficient storage space on their device.

7.3.2 PHYSICAL RESOURCE REQUIREMENTS

7.3.2.1 Computer Hardware Requirements

	Minimum requirements
Processor	Intel Pentium® D 2.8 GHz or AMD Athlon™ 64 X2 4400+
Video	Intel Q35 Express or Radeon HD 2400 PCI
Memory	512 MB RAM
Resolution	1280 × 720
Internet	1 Mbps

7.3.2.2 Computer Hardware Resource Requirements

I/O Devices	QWERTY keyboard Mouse Hard drive with sufficient free space

7.3.2.3 Computer Software Requirements

Operating System	Windows® XP/Vista/7/8/8.1/10 OS X Chrome OS Linux
Browser	Google Chrome, Firefox, Safari, Opera, Internet Explorer (Latest version recommended)

7.3.2.4 Computer Communications Requirements

Communication between users will occur through comments on a specific thread. The only communications requirement will therefore be the minimum computer requirements to access the website and a keyboard.

7.3.3 ENVIRONMENTAL CONDITIONS

The product runs in every environment with a mouse, keyboard, and a working internet connection. A hard drive is optional if the user needs to store files offline. The user can access their account assuming they do not forget their login information.

7.3.4 SYSTEM PERFORMANCE CHARACTERISTICS

The system will be able to handle multiple read/write operations simultaneously as well multiple user interactions. The system will also be able to handle load at peak user times. The system allows simultaneous file uploads and downloads and this is checked from time to time to ensure that the database can handle the total file size to be held on the server.

7.3.5 SAFETY REQUIREMENTS

Safety is maintained through a registration and login system. The user will have unique credentials that they can modify that will keep track of all permissions given to them in the system. Upon losing credentials, the user can message the system for resets.

7.3.6 SECURITY AND PRIVACY REQUIREMENTS

The credentials maintained by users is a simple username and password verification system. The system maintains that each new user has a unique username that is not in the system already. Privacy is equivalent to access in our system. Files can only be downloaded and uploaded by a logged in a user. Domains given to groups will be monitored monthly, but contents will not be released. Files uploaded will likewise be checked for legitimacy, but not released. Files deemed to be dangerous will be made inaccessible to users.

7.3.7 SYSTEM HUMAN INTERFACES

The system will give users an interface to view all their uploaded files. The interface will also allow file uploads and downloads for users with access to the domain or subdomain in which the file is located. Files accessed will provide a clean text editor that also have a comment system below that contains comments made by users.

7.3.8 SYSTEM MAINTAINABILITY

The version control of the system is maintained by git. The server requires a system administrator to check every once in a while to ensure that there will be valid space to contain possible future uploads. This amount will be determined after a few months of usage by the system to determine average monthly data upload. This figure will change over time as popularity increases. Any possible faults will be reported by admins as well as users and will promptly fixed.

7.3.9 SYSTEM QUALITY FACTORS

In order to have a high quality system, the goal is have the lowest possible number of faults and defects. To achieve this, programmers will partake in peer code reviews, walkthroughs, and inspections. This will greatly reduce the number of faults found in a system, thus increasing quality.

7.3.10 DESIGN AND CONSTRUCTION CONSTRAINTS

7.3.10.1 Life Cycle Model

The life cycle model used will be the Iteration and Incremental Life Cycle Model. This model is flexible and adaptable. During the design of the website, there will be many changes in its appearance and feel. Increments will be used to handle these changes, while the inherent Waterfall Model will provide a structured basis for development.

7.3.10.2 Policies and standards - Methods, tools, and techniques

Coding will be done primarily in HTML5, CSS, and JavaScript. These three languages will create the front-end, the website. The website will have a header for navigation and a footer for legal terms.

Git will be used for source-control during the development.

7.3.11 PERSONNEL-RELATED REQUIREMENTS

The head administrator will periodically check the content and files uploaded for all the domains and subdomains. This check will be quick and cursory. It is up to the domain administrator and subdomain administrators to conduct more thorough checks on the content in their assigned domain.

7.3.12 TRAINING-RELATED REQUIREMENTS

The accessibility of the website is very simple. Users will register for an account that is aided by a step-by-step form process. Functionality of website will be self-explanatory by a friendly interface.

7.3.13 LOGISTICS-RELATED REQUIREMENTS

All crashes and errors will be logged in the database. Data usage rates will be noted to determine the rate at which space will need to be added to the servers. Git will log code history.

7.3.14 PACKAGING REQUIREMENTS

There are no packaging requirements for Sharit.

7.3.15 PRECEDENCE AND CRITICALITY REQUIREMENTS

The main requirement is to ensure that Sharit does not turn into a site where illegal files (copyrighted material) are shared. File security is also very important, users who do not have access to a domain should not have access to the files inside.

7.3.16 OTHER NONFUNCTIONAL REQUIREMENTS

The Sharit system will be made scalable so that if the user base gets large, the system can adjust accordingly to accommodate.

8. SYSTEM TEST PLAN REQUIREMENTS

This test plan will assess the requirements as specific in section 7.1.
The following test scenario will demonstrate how a user might use Sharit.

Spec. #	Action	Input	Expected Output
7.1.1.1	Sign up for account	smith@nyu.edu	Account created
7.1.1.1	Sign up for account	jane@gmail.com	Fail
7.1.1.2	Login to account	smith@nyu.edu	Login Successful
7.1.1.2	Login to account	foo@nyu.edu	Fail
7.1.1.3	View profile	Smith's profile	Smith's information(name, company/school)
7.1.1.4	Change information	John	Smith's name is now John
7.1.2.1	Access domain	Smith's domain	Success
7.1.2.1	Access domain	Jane's domain	Fail
7.1.2.2	Check administrator	Smith	True
7.1.2.3	Delete thread	Ice skating	Thread deleted
7.1.2.3	Grant permission	Jane	Permission granted
7.1.2.4	Create subdomain	Tests	Subdomain successfully created

7.1.3.1	Create thread	Prof. Bob’s Fall 2015 midterm	Thread successfully created
7.1.3.3	Upload file	fall_2015_midterm .pdf	File upload success
7.1.3.3	Download file	fall_2015_midterm .pdf	Download success
7.1.4.1	Upvote/downvote thread	-1	Vote success
7.1.4.2	Comment on thread	“Useful sample”	Comment posted
7.1.4.3	Comment on comment	“Incorrect info”	Comment replied
7.1.4.4	Upvote/downvote comment	+1	Comment upvoted/downvoted
7.1.5.1	Search file/thread	fall_2015_midterm	File/thread found
7.1.5.2	Search user	John	User profile found
7.1.5.3	Search subdomain	CS2214	CS2214: Computer Architecture course found

9. QUALIFICATION PROVISIONS

This document will be reviewed by each of the four members of team. The qualities that the team will look for are: correctness, clarity, completeness, consistency, stability, verifiability, modifiability, and traceability. Each team member will understand every aspect of the document and will look for faults in the document. Once a fault has been found, it will be noted down and promptly corrected. Once the team feels like all the faults have been corrected, the document will be redistributed to the team for another review. This is done as many times as needed for all the faults to be corrected.

10. REQUIREMENTS TRACEABILITY

The functional and non-functional requirements listed in section 7.1 will be thoroughly traced throughout the life of the software engineering process. Each specification will be corresponded with an artifact in each of the workflows and be traced. Requirements tracing is done to avoid litigation and also mitigates liabilities. Keeping track of requirements will make sure everything is implemented and meets the requirements as described in section 7.1.

11. EVOLUTION OF THE SRS

The first version of this document will cover the scheduled trajectory of the project; all estimates and planning will be ordered and initialized. This version is designed to allow maximum leeway for changes regarding future deficiencies, shortcomings, inaccuracies, or changes in the system environment. As project progression continues, tasks in the schedule will be marked completed and upgrades or deprecates of various parts of the project will be noted.

12. RATIONALE

Finding information relevant to coursework can drastically improve one's grades. NYU-Poly has historically incurred an usually low graduation rate (50~60%). Information, pertaining previous semester student notes, tests, and possibly homeworks, should be upheld as samples for students currently taking the course to help them review. In the requirements, we aim to answer general questions posted by students that not even google.com, NYUClasses, and instructors could not provide. This is most often phrased "what's the format" or "is there a sample midterm?". To uphold integrity of information, we provide and limit information and registration to students with a nyu.edu email address.

We also have in the requirements 7.2.1 and 7.4.4, that inappropriate content posted by students will reviewed by moderators and also by other students. These students can provide anonymous feedback by upvoting and downvoting what they deem to be inappropriate.

13. NOTES

Initial release will be limited to people with a nyu.edu email address. This product will currently only serve only NYU students and staff. This will allow us to track errors without having to modify a huge database. Once we have ascertained the product to be sufficiently good enough for an increase in users, we will allow distribution to other universities as well as possible start-ups.

14. APPENDICES

14.1 SCHEDULE TRACKING

Artifact or Deliverable	Whom	Estimated	Actual	Difference
Initial SRS	Allen Zheng	3 hr	4 hr	1 hr
	Hui Huang	3 hr	5 hr	2 hr
	Kenneth Liang	3 hr	7 hr	4 hr
	Warlon Zeng	3 hr	5 hr	2 hr
	Summary	12	21 hr	9 hr
SRS 2.0	Allen Zheng	2 hr	2 hr	0 hr
	Hui Huang	2 hr	4 hr	2 hr
	Kenneth Liang	2 hr	3 hr	1 hr
	Warlon Zeng	2 hr	4 hr	2 hr
	Summary	8 hr	13 hr	5 hr

Cumulative

Whom	Estimated	Actual	Difference
Allen Zheng	5 hr	6 hr	1 hr
Hui Huang	5 hr	9 hr	4 hr
Kenneth Liang	5 hr	10 hr	5 hr
Warlon Zeng	5 hr	9 hr	4 hr
Summary	20 hr	34 hr	14 hr

14.2 DEFECT TRACKING

Artifact or Deliverable	Whom	Estimated	Actual	Difference
Initial SRS	Allen Zheng	5	3	2
	Hui Huang	5	2	3
	Kenneth Liang	5	4	1
	Warlon Zeng	5	2	3
	Summary	20	11	9
SRS 2.0	Allen Zheng	4	2	2
	Hui Huang	4	5	1
	Kenneth Liang	4	3	1
	Warlon Zeng	4	2	2
	Summary	16	12	6

Cumulative

Whom	Estimated	Actual	Difference
Allen Zheng	9	5	4
Hui Huang	9	7	4
Kenneth Liang	9	7	2
Warlon Zeng	9	4	5
Summary	36	23	15