LI PINGRUI

GE ZHUOYAN

TAN BOON PING

TANG YUTING

WANG BINLI

WANG YING

Team ACE

SCSE, NTU

System Requirement Specifications

Of StudyHub@NTU

# Table of Contents

[System Requirement Specifications 0](file:///C:\Users\tanbp\Documents\NTU\y3\cz3002%20ase\System%20Requirement%20Specifications.docx#_Toc82630721)

[1 Table of Contents 1](#_Toc82630722)

[2 Statement of Problem 2](#_Toc82630723)

[3 Overview 2](#_Toc82630724)

[3.1 Background 2](#_Toc82630725)

[3.2 Overall Description 2](#_Toc82630726)

[4 Investigation & Analysis Methodology 2](#_Toc82630727)

[4.1 System Investigation 2](#_Toc82630728)

[4.2 Analysis Methodology 3](#_Toc82630729)

[4.2.1 Feasibility Study 3](#_Toc82630730)

[4.2.2 Requirement Elicitation 3](#_Toc82630731)

[4.2.3 System analysis 4](#_Toc82630732)

[4.2.4 Scope and Limitations 4](#_Toc82630733)

[5 System Design 5](#_Toc82630734)

[5.1.1 Prototype 5](#_Toc82630735)

[6 Constraints 8](#_Toc82630736)

[6.1 Scalability 8](#_Toc82630737)

[6.2 Proprietary hardware and software 8](#_Toc82630738)

[6.3 Data Consistency 8](#_Toc82630739)

[6.4 Data Validation and Security 8](#_Toc82630740)

[7 Operational Requirement 9](#_Toc82630741)

[7.1 Website Admin 9](#_Toc82630742)

[7.2 Maintenance 9](#_Toc82630743)

[8 Functional Requirement 9](#_Toc82630744)

[8.1 Account 9](#_Toc82630745)

[8.2 Forum 9](#_Toc82630746)

[8.3 Swap index 9](#_Toc82630747)

[8.4 Form a team 10](#_Toc82630748)

[9 Input Requirement 10](#_Toc82630749)

[9.1 Course 10](#_Toc82630750)

[9.2 Index Swap 10](#_Toc82630751)

[9.3 Team Formation 10](#_Toc82630752)

[9.4 Forum 10](#_Toc82630753)

[10 Process Requirement 11](#_Toc82630754)

[10.1 Data Validation 11](#_Toc82630755)

[11 Output Requirement 11](#_Toc82630756)

[11.1 Email Notification 11](#_Toc82630757)

[11.2 Website Notification 11](#_Toc82630758)

[11.3 Exception Reports 11](#_Toc82630759)

[12 Software Requirement 11](#_Toc82630760)

[12.1 Requirement on Client side 11](#_Toc82630761)

[12.2 Server Requirement 11](#_Toc82630762)

[12.3 Network System 11](#_Toc82630763)

[12.4 Licences 11](#_Toc82630764)

[13 Deployment Requirement 12](#_Toc82630765)

# Statement of Problem

Due to COVID-19 pandemic NTU has adopted a hybrid teaching mode with many virtual lectures and tutorial sessions. This reduces the chances for NTU students to interact with their fellow classmates. However, this social interaction with their peers is beneficial for students to exchange information on course registration, group projects and course content.

Therefore, a study-based social platform, StudyHub@NTU, is needed for NTU students to increase communication and collaboration with each other throughout their student life at NTU.

# Overview

## Background

There are various existing platforms such as Telegram and Reddit, where students at NTU can communicate with each other. However, these platforms are not primarily focused on academics at NTU, topics in these platforms tend disorganized and contain broad general information. When students search for course information, index swapping intentions and potential teammates, these general platforms would be inefficient. There are also some existing systems to exchange course information, for instance NUSMOD. NUSMOD has the functionalities of course information, course planning, course commenting, but it does not emphasize the interaction among peers.

Therefore, a study-based platform is needed to improve the study environment and make students information searching easier.

## Overall Description

NTU’s current STARS Planner has several features such as displaying course information, plan schedule and register courses. StudyHub@NTU aim to complement and fill in several quality-of-life features such as course information sharing which includes commenting and rating. Study-based interaction including study forum, index swapping section and teammate searching section to assist students in their course registration.

# Investigation & Analysis Methodology

## System Investigation

StudyHub@NTU interacts with an external database, the browser used in the client side, an external email server as well as the user. When User launches the system, the system will send a series of queries to the database and render the web page with the data received to the browser. When certain important events happen (e.g., new index swapping request), the system will invoke the email server to notify the user.

Graphical user interface, application

Description automatically generated

Figure 1 System Context Diagram

## Analysis Methodology

## Feasibility Study

There are many mature and well-developed web frameworks (e.g., Vue.js) and tools (e.g., Firebase database). Since the Firebase database provides sufficient API on query, the major effort will be on the front-end, such as query sending, content update, web page rendering and animation. If all the development team focus on the front-end development and utilize open-source template on common components such as user-authentication, thread posting, the project should be feasible.

## Requirement Elicitation

The requirements of the system are elicited in the following steps,

1. Identify the major use cases, as shown in Figure 2.
2. Survey the target users for the expected requirements.
3. Go through the user journey when using the system and identify the important tasks, as shown in Figure 2.
4. Decompose the tasks into tangible and verifiable functional requirements.

Table, website, timeline

Description automatically generated

Figure 2 User Journey

### System analysis

As shown in the figure below, the major features include course comments from other students, look for other students for index swapping and forming a team for course project, as well as a student-led Q&A forum.

Diagram

Description automatically generated

Figure 3 The major use case diagram

### Scope and Limitations

Analysis methodology will involve business analysis, requirement analysis, data analysis, process analysis, (web) and application architecture:

* Business analysis
  + Follow the regulations of NTU
  + State the sponsorship and associated project budget requirement
* Requirement analysis
  + System I/O description
  + user requirement definition
  + System Constraint requirement
* Data analysis – Involve data collection process, data validation, data storage, manipulation and retrieval
* Process analysis – Data/process flow analysis, process decomposition and system interfaces
* Application architecture – Analyse application information structure, usability, user interface design, interaction, and application implementation.

# System Design

The system is a web-based application. It follows the client-server architecture pattern. The client system, e.g., Web Client, will query the cloud server for data.

Diagram

Description automatically generated

Figure 4 System Component Diagram

Table 1 The description of the components in the system architecture.

|  |  |
| --- | --- |
| Component | Description |
| Account | Basic functions including registration, log-in, log-out and user profile |
| Forum | A system that allows users to post threads, reply. It requires users to log-in. Before the thread is post, it must pass the content check. |
| ContentChecker | A system that forbids illegal and inappropriate content to be published. |
| CourseInformation | A system displaying the course information. |
| Index Swap | Subsystem for users to post and display index swap. It requires users to be logged-in |
| Teammate Search | Subsystem for users to post and display teammate search information, requiring users to be logged in. |
| Feedback & Rate | Subsystem to record and display users’ comments and ratings on courses. Posting information requires users to be logged in. |
| ServerQuery | API to do CRUD operations on database. |
| Server | Deployed server storing data for the system. |

### Prototype

Graphical user interface

Description automatically generated with medium confidence

Table

Description automatically generated with medium confidence

Graphical user interface, application, table

Description automatically generated

Graphical user interface, table

Description automatically generated

Text

Description automatically generated

# Constraints

## Scalability

With the increasing users and the accumulation of content posted in the system, the scalability is an important constraint to the system. Hence, it requires the system to [1]

* Have good modularity
* Conduct the load set
* Select 3rd party services wisely

## Proprietary hardware and software

The system needs Firebase database to store the data for the system and an email server to notify the users of important events.

## Data Consistency

The web page may display “dirty” data, which occurs when data is not updated on the current webpage as data updates will only be fetched from the database when a manual refresh is triggered. Therefore, to ensure the data consistency between the client browser and the remote database, Firebase Realtime Database should be used. Such a Realtime database can generate a trigger to reload the web page whenever the data is modified.

## Data Validation and Security

Because of personal data protection policy, the system should not collect unnecessary personal information. For example, real name, gender, or age is not mandatory, and the system will not make it a requirement for users to submit such data. All data stored in the database will be encrypted.

# Operational Requirement

## Website Admin

An admin is needed to monitor the content on the website to avoid study-irrelevant or abusive content. Because course information is updated each semester, the admin is also responsible to verify the information update proposed by users.

## Maintenance

Due to uncertainty and sometimes sudden changes to course information especially at the start of new semesters, it is necessary for administrators to quickly update outdated information based on user generated reports. To facilitate the maintenance, the system should be designed neatly to achieve high cohesion and low coupling.

# Functional Requirement

## Account

1. System shall ask the user for the school email (i.e., ending with @e.ntu.edu.sg) and verify it during registration.
2. System shall allow the logged-in user to log out or switch to another account using school email.
3. System shall not allow the use to log in to two different accounts simultaneously.
4. System shall allow the logged-in user to edit his profile.
5. System shall distinguish between the admin and normal student account and give them different permissions.
6. System shall allow the logged-in user to establish connections with other logged-in users.

## Forum

1. System shall allow the logged-in user to post, edit and delete threads that belongs to him in the forum.
2. System shall allow the user who posted the thread to set the visibility of the thread.
3. System shall allow the logged-in user to reply to other users’ threads in the forum and send a notification to the user whose thread is be replied.
4. System shall allow the logged-in user to search the keyword of threads or search based on the tag of threads.
5. System shall allow the logged-in user to top a thread.
6. System shall allow the logged-in user to like and forward other users’ threads.
7. System shall allow the logged-in user to put a limited number of threads into his collection.
8. System shall allow the logged-in user to send a private message to other users.
9. System shall allow the logged-in user to check his collection and his private message.

## Swap index

1. System shall allow the logged-in user to post, edit, close, and delete one index swap request per course at the same time.
2. System shall open the index swap functionality during the Add/Drop period.
3. System shall allow the admin to set the Add/Drop period.
4. System shall only allow the logged-in user to view the index swap request.
5. System shall allow a user to respond to only 1 request per course if the user does not post a request.
6. System shall not allow a user who has already post the index swap record under the same course to respond to any other request.
7. System shall send an email notification to the user who post the request if another user respond to it.
8. System shall delete the request after the user closes the request or the Add/Drop period finishes.

## Form a team

1. System shall allow the logged-in user to create one team per course, if the user does not join in any other teams.
   1. System shall ask the user to specify the team’s name as a Unicode string with the length ranging from 1 to 30.
   2. System shall ask the user to input the team description as a Unicode string with the length ranging from 5 to 100.
   3. System must set the maximum team size as a positive integer ranging between 2 to 30, which set by the admin user.
2. System shall indicate the user who create the team as the team creator.
3. System must restrict the number of teams the user joins in for one course is one at maximum.
4. System shall allow other logged-in users to send requests to join in multiple teams.
5. System shall send the email notification to all team members if other users request to join.
6. System shall make all pending requests visible to the team members.
7. System shall allow the team creator to accept a request to add a new user into the team if the current team size does not exceed the limit.
8. System shall delete all requests made by the user for the same course if the user is accepted by a team.
9. System shall delete the team information at the specified date given by the admin.

# Input Requirement

## Course

The course code is unique, but a course can have multiple course codes. Users can comment and rate a course for other students to read about. User is free to edit the comment, but user can only give 1 comment for each course to avoid duplication.

## Index Swap

The index-swapping functionality is open during Add/Drop period each year. Users can input the index they registered with multiple preferred indexes. Once other Users indicate their interests, the request record will be in status of “Temporarily Closed” to avoid over-whelming enquiries on the same index swap index. User who posts the record can delete the record after the successful match, and then the record will be deleted from the server. All index swapping index will be deleted after the Add/Drop period.

## Team Formation

Users can post one team formation record per course and then automatically become the team leader. Other users can request to join. An email will be sent automatically to the team leader for confirmation. The formation record will be closed after the team is full or the team leader closed the record, and the record will be deleted afterwards.

## Forum

There is a section in forum for each course. In a section, there are multiple threads with the similar topic, e.g., about the course. A thread includes a title with the length of 10-100 characters and the content with the length of 50-1000 characters. The content is formatted with markdown format. All users can post threads and reply to other threads.

# Process Requirement

## Data Validation

Before User submits the form, e.g., registration form, the system should validate the input and prompt clear error messages if there are errors. After the information is changed from the server, the system should inform the user of the change and ask the user to fresh the page.

# Output Requirement

## Email Notification

The system will send an email notification to User’s school email when someone replies to s/he index swapping requests or team formation requests.

## Website Notification

The system will notify the users of new events in the user centre after the user launches the web page. The events including the new response in the forum as well as the response to the request.

## Exception Reports

The system should provide the user with clear error and exception messages during query and input stages.

# Software Requirement

## Requirement on Client side

Web-browser (Firefox, Chrome, etc.) with Internet connection.

## Server Requirement

The deployed server should have 1GB storage space with 1 core CPU.

The storage server should have the storage memory of 10GB with 1 core CPU. As the system may be queried frequently daily, the server should also support high I/O throughput.

## Network System

Network software and protocols in order for systems to communicate:

* TCP/IP
* HTTP
* HTTPS
* FTP

## Licences

The system uses Firebase provided by Google as the storage and a Linux Server provided by Digital Ocean. Currently, the system can function with free licences.

# Deployment Requirement

A picture containing graphical user interface

Description automatically generated

Figure 5 Deployment Diagram of System