Product Limitations

As all products do, StackBook has limitations, and understanding these limitations is the only way to overcome them and move forward with a product that safe and useable to the public.   
Having researched subjects specific to the time-scale, this document will contain a review of which implementations are possible and which may need to be scrapped, or tweaked in order to meet a deadline.

# The Overall Product

StackBook is a peer-peer support site, it is intended to support various features including;

. Live Chat  
. Support Forums  
. Rating/Ranking System  
. User Accounts  
. Validation and Authentication

These features are the key-significance of the site, it is also intended to convert the site into an Application for Android users and possibly IOS in the future.

## Forums

The forums are the key-zone for help on the website. Forums allow users to open a thread about specific issues, allowing other users comment responses. In order to ensure “quality-control” within the system a system is required to ensure users are moderated. It is intended to use the user-rating system in order to give users a trust-score (See: Research – Live Chat Function, Prevention of Misleading Information) for the ScrapBook website.   
  
Although it could take a while, preserving server space should be considered as the first limitation, as running out of space to store data is an important issue to consider.  
As users post more and more content, more memory will be required to store data, the implementation of a system to monitor posts and remove duplicate posts should be considered.

A system as such does itself does, however, have issues. If you remove a user’s post and they did not get a response, nor know where to find the duplicate question, it can be frustrating to the user seeking help, this means the site Is not “user-friendly” as the problem which the site is intending to solve is instead transformed into a new one.

A clever system could remove a duplicate post, inform the user why their post has been removed, and, while doing so, provide a link to the thread they need, advising them of who to contact if additional support is needed.

A secondary solution which could be beneficial would be to add an additional “Similar Problem” tab could be considered. This could function as an additional comments tab, separate from the “Answers” tab in which users who have a similar issue, but they think is slightly different, or are still struggling to understand the issue can post, although this solution still creates more data to store, it is still a more efficient solution than sporting duplicate threads.

## Live Chat

Live Chat implementation has issues in itself. Although many users and developers believe that it is useful to have on a website a live chat needs to be following;

* **Safe** – A Live Chat needs to ensure it keeps users as safe as possible, and informs them of how to be safe online. Users not only need to be safe but feel safe using the service on the website.
* **Reliable** – Live Chat Support needs to be a reliable service which users can access easily at any given time and have messages send and appear successfully in a reasonable length of time.
* **Moderated** – Live Chat Support should be moderated – this can be done by a person or programme running to try and keep users on track and prevent “prank chats” or people wasting others time. Conversations can be saved into a log file, so long as the user agrees to this on joining.

### Moderation

Moderating a chat can be tricky, too much security can put users off using the software, keeping chats too formal may hinder the process of solving an issue or making a contact who can help more in the future, this would work against the overall goal of ScrapBook, however too little and the chat is opened to people looking to waste others time, troll those in need of help or even bully others using the online service. A good middle ground is to Moderate public chats heavily, users agree to have their chats “recorded”, every so often the chat is saved to a log file for that date, however in private chats have a laxer approach in which users simply have the ability to report users, and certain phrases or words are prohibited.

#### Repercussions

Every online service should involve a “terms of service”, whether a user agrees to this formally (clicking a button) or there is simply a page stating that by using the service the user is agreeing to abide by the terms and rules stated.

Templates for a terms of service contract can be found online and have items added or removed to fit the requirements of the website, beforehand there should be rules and limits set to prevent intentional misuse of the service, or issues arising where the developers are held responsible for an event.

The repercussions should a user choose to ignore this agreement should match to what the offense is, or if it is a repeated violation, for instance;

A user reported for harassment could be muted for a time period, increasing in time to an arbitrary amount depending on how many times the user has been reported and found to have committed the act. Repeated violations could result in a temporary or permanent ban from the site.

In order to incorporate this however there needs to be a system for false reports.  
This should involve an automated system to check over the recent chat and check for keywords depending on the violation reported, automatically muting or banning a user, however if the report comes back as false, an admin should be contacted to check over the log file manually, if the report is found to be a false report, the user who submitted the report should be issued a warning or a muting period.

#### Limited Admins

Due to the size of the team, admins to monitor data are limited, depending on the number of users the task of moderation could become unmanageable, an automated system would be ideal, however implementation may not be possible during the time scale given, at least not for an overall “clever”, robust, working system, instead alternatives should be considered.

Reports should work as stated earlier, a simple system to check over the recent logs, with the option of the system emailing an admin, this could be improved over time in order to meet security requirements.

## User Accounts

A limit should be ensured, as one account per user in order to share space, this can be done with email addresses, as there is no need for more accounts, although users could effectively bypass this simply by creating multiple email addresses, using a unimail account for authentication could prevent this issue arising.

### Authentication System

Limited knowledge within the team of authentication systems could result in security flaws. Many authentication systems are compromised through interception during the transition of data from the user to the server. Hash coding is seen as relatively unbreakable, the strongest encryption as of writing this is AES, however implementing this is limited to the amount of knowledge or experience the team has working with these systems, if something goes wrong with authentication it can render the entire service unusable to a user, or even compromise user’s personal data. During research it has been brought to attention that Google offers a service which works between the web application and a google service (0Auth 2.0) there are also various other authentication services online including safeNet.