

Software Requirements Specification Document (CS360)

Back Space



Group Number: 05

Aswad Tariq
Zeerak Babar
Haseeb-Ur-Rehman Faheem
Ibrahim Sanaulah
Nashrah Shaukat

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management
Sciences (LUMS)

Version: 1.0

Date: (15/02/2022)

Number of hours spent on this document: 55

Contents

1	INTRODUCTION	1
1.1	DOCUMENT PURPOSE	1
1.2	PRODUCT SCOPE	1
1.3	INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	3
1.5	REFERENCES AND ACKNOWLEDGMENTS	4
2	OVERALL DESCRIPTION	4
2.1	PRODUCT PERSPECTIVE	4
2.2	PRODUCT FUNCTIONALITY	5
2.3	USERS AND CHARACTERISTICS	6
2.4	ASSUMPTIONS AND DEPENDENCIES	7
1.	SPECIFIC REQUIREMENTS	8
2.5	FUNCTIONAL REQUIREMENTS	8
2.5.1	<i>1- Account management</i>	8
2.6	EXTERNAL INTERFACE REQUIREMENTS	17
2.6.1	<i>User Interfaces</i>	17
2.6.2	<i>Hardware Interfaces</i>	17
2.6.3	<i>Software Interfaces</i>	17
2.7	USE CASE VIEW	18
2.7.1	<i>Use Case Table</i>	18
2.7.2	<i>Use Case Diagram</i>	18
2.7.3	<i>Use Case Description</i>	20
3	OTHER NON-FUNCTIONAL REQUIREMENTS	22
3.1	PERFORMANCE REQUIREMENTS	22
3.2	SAFETY AND SECURITY REQUIREMENTS	23
3.3	SOFTWARE QUALITY ATTRIBUTES	23
3.	APPENDIX A – TOP 10 USER STORIES	24
4.	APPENDIX B – ARCHITECTURAL SPIKE (ONE STORY)	25
5.	APPENDIX C - GROUP LOG	26
6.	APPENDIX D – CONTRIBUTION STATEMENT	28

1 Introduction

1.1 Document Purpose

The document is related to our product 'Back Space', a comprehensive social networking mobile application made to cater for the needs of the lums community. The document highlights the needs and requirements required for the entire process from the product planning to implementation and execution. The document introduces the product scope and the intended audience. Moreover it also consists of the product functionality, functional requirements, non functional requirements and user stories.

The document is a walkthrough of the entire software development process for the first version of the application 'Back Space'. The application itself would be a single functional unit and is not part of any larger system, hence the document encompasses the product design description, product components, features and requirements. The document also enlists the functions that the users of the application, students and the administration would be able to perform in the application.

1.2 Product Scope

Back Space is a much needed application with a goal to provide Luminites with an exclusive platform to connect with each other. The application will allow users to access information regarding courses and instructors, chat with their friends in groups, receive real-time notifications and much more. This will save the time of the users by offering better access to up-to-date information using multiple channels and tabs, all on one platform. Setting up the app will also be very convenient. The app will also be freely available and accessible to everyone with an android device.

It will provide Luminites a platform to easily navigate course and instructor information by making it easily accessible using in-app search functionality. Backspace will also provide subspace for people where people with similar interests can come together and interact. Overall, it will serve the purpose of being an all in one app for Lums students with regards to being a hub of communication as well as a source of information.

1.3 Intended Audience and Document Overview

Intended readers for the document are as follows:

1-Developers:

Names: Nashrah Shaukat, Aswad Tariq, Zeerak Babar, Ibrahim Sanaullah, Haseeb-Ur-Rehman Faheem

The developers are the team members involved in the development process and documentation of the process of the product. The document serves the purpose of a guideline for the developers making it easier for them to highlight the key features to be included in the software as well as the functional and nonfunctional requirements. The document also provides the developers with a clear mind set about what the functionality of the product will be like. The use cases and the user stories also bring about the perspective of the users into the development process making it easier for the developers to design and develop a product *for* the users, keeping in mind their requirements, and making a product

which would be user friendly. The developers should extensively study the Section 3 and Section 4 of the document so they are familiar with the requirements of the project. The developers should also refer to the Section 2 of the document to remind themselves of the product functionality, users and their characteristics when implementing the Project.

- Section 3
- Section 4
- Section 2

2-Teaching Assistants:

Names: Irzum Bin Mansoor Jafri, Maha Sajid, Suleman Khan, Muhammad Mustansar Saeed, Syed Ibrahim Mustafa Shah Bukhari

The teaching assistants for the project are responsible for overlooking the project's progress and the process of the development phase. The document hence provides a framework for the teaching assistant to assess the functionality of the application and the progress of the development team. The document will also help them to provide feedback. The document is also useful for the teaching assistants to assist the developers in the further phases of software development. The contribution statements and the group log will also help the teaching assistants to assess each individual developers' contribution to the project. The teaching assistant can also go through the entire document in the specified order.

- Section 2
- Section 3
- Section 4
- Appendix C
- Appendix D

3-Instructor:

Name: Suleman Shahid

Suleman Shahid is the instructor of the course software engineering offered at LUMS. He would be supervising the development team and the progress made by the team, hence the document contains everything he would need to be introduced to the project, its specifications and functionality as well as the requirements. The instructor can also go through the entire document in the specified order or start from wherever he would prefer/like.

- Section 1
- Section 2
- Section 3
- Section 4
- Appendix A
- Appendix B
- Appendix C
- Appendix D

The document is divided into sections and subsections for clarity. The sections include:

1. *Introduction:*

This includes the document purpose, the scope and purpose of the product and also the definitions of the necessary terms.

2. *Overall Descriptions:*

This includes the product perspective, functionality, users and their characteristics and the assumptions and dependencies.

3. *Specific Requirements:*

This goes into the detail about all the requirements needed for the development of the product including

4. *Other non functional requirements:*

Non functional requirements include safety, security and privacy requirements, performance and also software quality attributes.

5. *Appendices:*

Appendix A - The key features and their descriptions which are deemed important by the users.

Appendix B- Architectural Spike

Appendix C- Group Log, which includes details about the minutes of the group meetings held by the team members

Appendix D- Contribution statement of the team members.

1.4 Definitions, Acronyms and Abbreviations

Back-end	It refers to the program's code that handles database and server-side rendering and cannot be accessed by a user.
Deployment	One of the important steps of creating a website or an app. It means setting up the app or website live so that it could be used by others in their system.
Front-end	It is part of a website or web application through which a user can interact.
LDF	Lums discussion forum a platform where lums students communicate over facebook
LUMS	A university located in Lahore,Pakistan.
Malicious	It refers to the intention of causing any harm.
Modular Code	Program code in which its functionality can be separated into independent modules.
Subspace	Tabs that contain sub forums related to a specific category or genre of posts can only be created or removed by admins and users must join these subspaces to view their posts.
Web Application	An application that runs on a web server through a web browser.
Widget	Building blocks of Flutter app's user interface, it is a framework that

	is displayed on the screen.
--	-----------------------------

1.5 References and Acknowledgments

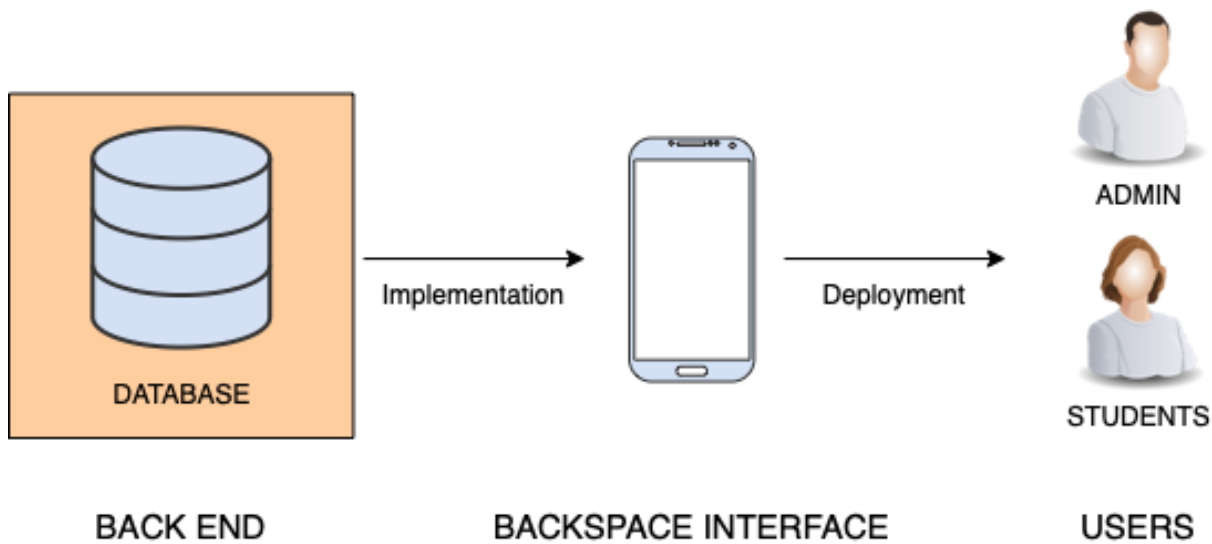
<https://app.diagrams.net/>

2 Overall Description

2.1 Product Perspective

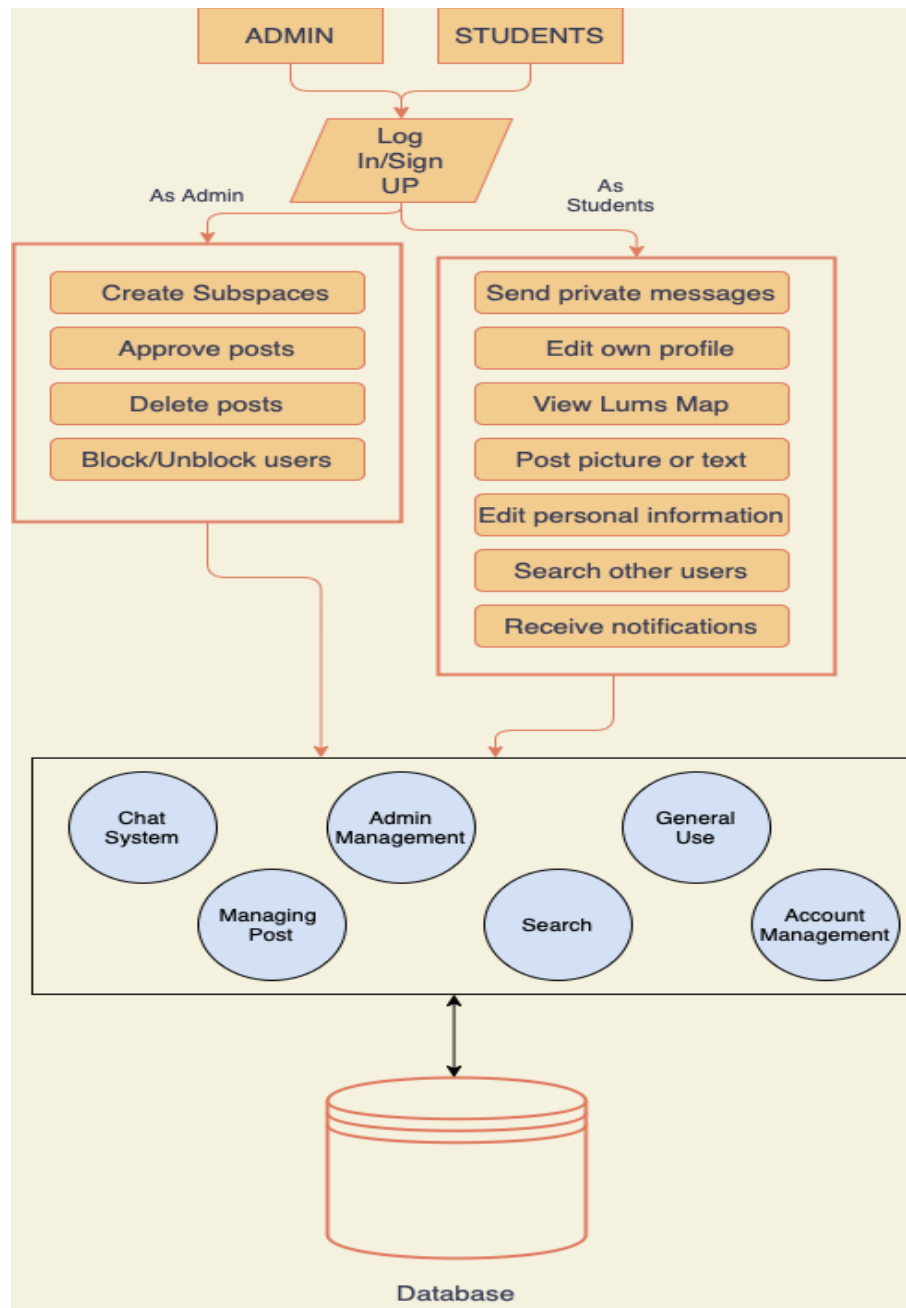
Back Space is an android application created with the purpose of providing students of LUMS with a platform that would allow them to connect with their respective Luminites. Although a similar platform already exists (LUMS Discussion Forum) on facebook, the existing platform has countless limitations including non categorized data, tedious induction process and hard to find data. Back Space will be a new self-contained product, not dependent on any previous versions.

It will provide Luminites a platform to easily navigate course and instructor information by making it easily accessible using in-app search functionality. Backspace will also provide subspace for people where people with similar interests can come together and interact. Overall, it will serve the purpose of being an all in one app for Lums students with regards to being a hub of communication as well as a source of information. The application will support 2 actors: users and admin.



2.2 Product Functionality

- Login/Signup
 - System provides a login and signup option to a user
 - At Login, the user needs to add his/her credentials. Admin will be redirected to the admin page while the user will be redirected to the newsfeed page.
 - At signup, the user provides personal information to create his/her account.
- User 1: Admin
 - Admins will be able to create subspaces
 - Admins will be able to approve posts from users based on the content present within them.
 - Admins can remove already approved posts.
 - Admins will be able to block and unblock users
- User 2: Students
 - The user will be able to send and receive private message from other users
 - The user will be able to edit their own profile and view profile of other users
 - The user will be able to view the LUMS map.
 - The user can post a picture, or a text and can delete it. Moreover, the user can also like and comment on a post.
 - The user can edit their personal information such as email, contact information and address.
 - The user will be able to search other users, and find paid tasks.
 - The user will be able to receive notifications from the app
 - The users will be able to view the 'freshman guide'



2.3 Users and Characteristics

There would be two types of primary users of the product differentiated based on the privileges levels of the two types:

Administrators:

The administrators of the application would have the highest level of privileges as they would be able to approve, disapprove the posts posted on different forums of the application. They would also be

able to view different users, block/restrict their access, create new subspace. They would also be considered as frequent users of the application since they would have to constantly keep checking the subspace in order to approve and moderate the posts and forum space.

Users:

The users would consist of the student body enrolled at LUMS or who are graduates of LUMS. They would be able to add a post on different subspace, like/comment on others' posts, view the LUMS map, view notifications, view chat, search news feed.

The users can further be divided into two categories based on the frequency of their use of the application and their importance as a user.

Students:

Students currently enrolled at lums would be expected to use the application more frequently as they would need to stay updated about the happenings at the campus and university. They would also be considered as a more important user group for the application as compared to the Alumni due to the frequency of their use.

Alumni:

These would include the LUMS alumni. The need not to stay updated about the university and its specifics hence they are expected to use the application less frequently.

2.4 Assumptions and Dependencies

- It is assumed that the intended group is familiar with the functionality and interface of a social media application such as facebook.
- The user must have an active internet connection and mobile phone to access the application.
- The users will be using the groups and channels for their intended purposes and would not engage in any sort of malicious activity.
- Since the intended users are from LUMS, the user must sign-up with a LUMS email id only.
- The users will negotiate the payment and compensation for tasks in the "Find Work" functionality of the app themselves.
- Due to limited time to complete this project, the app would only cater to LUMS students and will be available only on Play Store as our team does not have Mac OS.
- The team is constrained with limited expertise in software development tools.
- The performance of our application will depend on the database management system used and the latency they provide.
- The team will be relying on free APIs due to budget constraints

1. Specific Requirements

2.5 Functional Requirements

i.1- Account management

RQ 1. Login

- **Description:**
The system will allow the user to gain access to their account.
- **Input:**
A user will enter his/her username and password.
- **Processing:**
The system will compare the existing credentials with the ones entered by the user.
- **Output:**
If an incorrect password is entered, an “incorrect password” message would be displayed.
If a user enters credentials that are available in the database, the user will be redirected to the homepage.

RQ 2. Signup

- **Description:**
The system will generate a new account for the user.
- **Input:**
A user will enter his/her username, email, contact information and password.
- **Processing:**
The system will verify the data entered by the user and store it in the database.
- **Output:**
The system will display login screen once the user has completed the signup process

RQ 3. Update Credentials

- **Description:**
The system will allow the user to update his/her credentials such as profile picture and contact information.
- **Input:**
The user will click the option of “Edit profile”.
- **Processing:**
The system will redirect the user to another screen where the user will enter data they wish to change. After adding all the information, the user can click on the “Save” button which will cause the system to validate all the changes and then update it.
- **Output:**
After saving changes, the user will be able to view updated personal information.

2- Search

RQ 4. Search Users

- **Description:**
The system will allow the user to search for other users.
- **Input:**
The user will click the search icon and type the name of the user..
- **Processing:**
System will search the data entered in the database.
- **Output:**
All the users with similar names will be displayed.
If a user does not exist, “No search result” message will be displayed.

RQ 5. Search Instructor Reviews

- **Description:**
The system will allow the user to search within an instructor tab locally to find relevant posts about a specific instructor.
- **Input:**
The user will click the search icon and type the name of the instructor and/or keywords.
- **Processing:**
Based on what the user has entered the database will be queried for all relevant and similar posts that contain the keywords or instructor name.
- **Output:**
All search results will then be returned and shown on the screen if no results match the search criteria then “No search results” will be displayed.

RQ 6. Search find work tab

- **Description:**
The system will allow the user to search in the Find Work tab locally to find specific posts about the work.
- **Input:**
The user will click the search icon and type keywords.
- **Processing:**
Based on what the user has entered the database will be queried for all relevant and similar posts that contain the keywords.
- **Output:**
All search results will then be returned and shown on the screen if no results match the search criteria then “No search results” will be displayed.

RQ 7. Search NewsFeed

- **Description:**
The system will allow the user to search within the news feed tab locally to find specific posts.

- **Input:**
The user will click the search icon and type keywords.
- **Processing:**
Based on what the user has entered the database will be queried for all relevant and similar posts that contain the keywords.
- **Output:**
All search results will then be returned and shown on the screen if no results match the search criteria then a “No search results” message will be displayed.

RQ 8. Search approved posts

- **Description:**
The system will allow the admin to navigate to the approved posts tab in the admin dashboard and search through posts already approved.
- **Input:**
The user will click the search icon and type keywords or post author name.
- **Processing:**
Based on what the user has entered the database will be queried for all relevant and similar posts that contain the keywords or posts that contain the user name or are authored by the user.
- **Output:**
All search results will then be returned and shown on the screen if no results match the search criteria then a “No search results” message will be displayed.

RQ 9. Search joined subspace

- **Description:**
The system will allow users to search through all joined subspace. .
- **Input:**
The user will click the search icon and type name or partial name of subspace.
- **Processing:**
Based on what the user has entered the database will be queried for all relevant subspaces containing or matching the search string.
- **Output:**
All search results will then be returned and shown on the screen if no results match the search criteria then a “No search results” message will be displayed.

3- Chat system

RQ 10. View all chats

- **Description:**
The system will allow the user to view all of his/her chats with other friends.
- **Input:**

The user will click the “Chat” widget.

- **Processing:**
The system will fetch the user’s chat with all other users from the database.
- **Output:**
The user will be shown all the recent chats with the specific friend on a new page on a scrollable screen.

RQ 11. View a specific chat

- **Description:**
The system will allow the user to view his/her chat with a specific friend.
- **Input:**
The user will click the Chat Widget of a specific friend on the chat page.
- **Processing:**
The system will fetch the user’s chat with the specific user from the database.
- **Output:**
The user will be shown all of the recent chat on a new page.

RQ 12. Send chat

- **Description:**
The system will be able to allow users to communicate with friends.
- **Input:**
A user will be able to enter a text or add any picture before clicking the “send message” widget.
- **Processing:**
Once the user has typed a message and clicked the send icon, the message typed will be displayed and a request will be sent to the other user.
- **Output:**
The message be displayed and delivered to a specific user.

4- Managing Post

RQ 13. View Post

- **Description:**
The system will be able to view posts uploaded by other users.
- **Input:**
The news feed will be displayed every time the user will login and can click on the home icon to view it.
- **Processing:**
System will take the user to a new screen where he/she can view posts.
- **Output:**
A complete list of posts made by other users would be displayed.

RQ 14. Add Post

- **Description:**
The system will allow the user to upload a picture or post a status to the newsfeed.
- **Input:**
The user will click the “Post” button. A new screen will open where the user can either write a text or upload any picture before clicking the “post” widget.
- **Processing:**
These changes will be stored in the database after the post has been submitted.
- **Output:**
The user will wait for the approval of post from the admin, after which the post will be visible on the newsfeed

RQ 15. Like a Post

- **Description:**
The system will allow the user to Like a post.
- **Input:**
The user will click the widget of “Like” on a specific post.
- **Processing:**
The system will increment the number of likes on the post.
- **Output:**
After saving changes, the user will be able to view the updated number of likes on the post. The colour of the “Like” widget will also change.

RQ 16 - Comment on a Post

- **Description**
The system will allow the user to add a comment on a specific post.
- **Input**
The user will click the “comment” widget, write a comment on the new page, and then click the submit button.
- **Processing**
Once submitted, the backend will be triggered into updating the comment stored in the database.
- **Output**
The submitted comment will be viewable in the post’s comment section. [SEP]

RQ 17. Delete a Post

- **Description:**
The system will allow the user to remove their post.
- **Input:**
The user will click the “Delete” button.
- **Processing:**

The post will be deleted from the database and will cause the user to delete the post from their profile.

- **Output:**
The selected post will be deleted from the newsfeed and no user will be able to view it.

RQ 18. Remove Post

- **Description:**
Admins will be able to remove posts by users.
- **Input:**
The admin will click on the remove button next to the relevant post to remove them
- **Processing:**
The post will be removed from the database
- **Output:**
The post will be removed from the newsfeed

5- General use

RQ 19. Add Instructor Reviews

- **Description:**
The system will allow the user to add reviews for an instructor on the instructor tab..
- **Input:**
The user will click the “Add” button and will be redirected to a form where the user will fill the form before submitting.
- **Processing:**
Based on what the user has entered the database will be queried for all relevant and similar posts that contain the keywords or instructor name.
- **Output:**
All search results will then be returned and shown on the screen if no results match the search criteria then “No search results” will be displayed.

RQ 20. View Freshmen Guide

- **Description:**
The students will be able to view the freshman guide.
- **Input:**
The student will click on the “View Freshman Guide” button on the sidebar.
- **Processing:**
The system will lead the user to a new screen which contains the relevant information.
- **Output:**
The guide will be displayed to the student.

RQ 21. View LUMS map

- **Description:**
The user will be able to view the map of LUMS.
- **Input:**
The user will click on the “Map” button.
- **Processing:**
The system will fetch the LUMS map from the database.
- **Output:**
The user will be shown a map of LUMS on a new screen.

RQ 22. View All Notifications

- **Description:**
The system will be able to show the recent notifications to a particular user.
- **Input:**
The user will click on the “Notification” widget.
- **Processing:**
The system will fetch the user’s notification from the database.
- **Output:**
The user will be shown all the recent notifications on a new page.

RQ 23. View a specific notification

- **Description:**
The system will be able to show a single notification on another screen.
- **Input:**
The user will click on the specific notification from the list of notifications in the “Notification” widget.
- **Processing:**
The system will fetch the specific user’s notification from the database.
- **Output:**
The user will be shown the notification on a new page for example if a user clicks on a notification related to the post, the user will be redirected to the post.

RQ 24. View FAQs

- **Description:**
The users will be able to view FAQs in the app..
- **Input:**
The admin will click on the “View FAQs” button.
- **Processing:**
The system will lead the user to a new screen which contains the relevant information.
- **Output:**

The FAQs will be displayed on a new screen.

RQ 25. View Terms and Conditions

- **Description:**
The users will be able to view terms defined using this social media application.
- **Input:**
The admin will click on the “View Terms and Condition” button.
- **Processing:**
The system will lead the user to a new screen which contains the relevant information.
- **Output:**
The terms and conditions will be displayed on a new screen.

6- Admin management

RQ 26. Add Events

- **Description:**
The Admin will be able to add new events.
- **Input:**
The admin will click on the “Add Events” button and add all the specific information about the event before clicking the “Add” widget.
- **Processing:**
These changes will be stored in the database after the event has been submitted.
- **Output:**
The event will now be displayed to the students.

RQ 27. Approve posts

- **Description:**
The system will allow the admin to approve posts before they are posted to the news feed.
- **Input:**
The admin will click on the “Approve” button that will allow the post by a user to be seen on the newsfeed by other users
- **Processing:**
The status of the post will go from pending to approved in the database
- **Output:**
Post will be removed from the pending posts screen and will be visible on the news feed

RQ 28. Deny posts

- **Description:**
The system will allow the admin to deny posts before they are posted to the news feed.

- **Input:**
The admin will click on the “Deny” button that will allow the post by a user to be seen on the newsfeed by other users
- **Processing:**
The Post will be removed from the database.
- **Output:**
Post will be removed from the pending posts screen and will not be displayed on news feed

RQ 29. Block User

- **Description:**
Admins will be able to block users
- **Input:**
The admin will click on the block button next to the relevant user to block them
- **Processing:**
The status of the user will go from unblock to blocked in the database
- **Output:**
The user will no longer be able to use the application

RQ 30. Unblock User

- **Description:**
Admins will be able to unblock users
- **Input:**
The admin will click on the unblock button next to the relevant user to block them
- **Processing:**
The status of the user will go from blocked to unblocked in the database
- **Output:**
The user will be able to resume using the application as normal

RQ 31. Create Subspace

- **Description:**
Admins can create subspaces that the users can join.
- **Input:**
The admin will enter relevant details about the subspace and click the create subspace button
- **Processing:**
Based on what the user has entered the database will be queried for all relevant subspaces containing or matching the search string.
- **Output:**

All search results will then be returned and shown on the screen if no results match the search criteria then “No search results” will be displayed.

2.6 External Interface Requirements

2.6.1 User Interfaces

Back Space will be an interactive mobile application which can only be used in android smartphones with an active internet connection. The interface will be easy to use, and all the text will be displayed in English. There will be two main screens:

- 1) Newsfeed Page
- 2) Admin Page

All of these screens will consist of proper icons with texts where necessary so that any user could easily navigate. Newsfeed page will consist of high-definition images with appropriate captions added by other users. Moreover, the interface will be designed using common UI elements such as buttons, pagination and text fields etc.

2.6.2 Hardware Interfaces

The application is a mobile based application hence the user should have access to a mobile phone which would have the specifics to run a mobile application. The supported device should also have a touch screen so the user is able to enter their input to the system. The supported device should be able to connect to a stable network. The mobile application is to be hosted on a server.

2.6.3 Software Interfaces

Back Space will only be compatible in Android phones. The user must have an android version 5 or higher, API level greater than 21 and version code Lollipop. The app will be built using Flutter and the backend will consist of firebase. Our app will be communicating with firebase to fetch data and store details depending on the user operation.

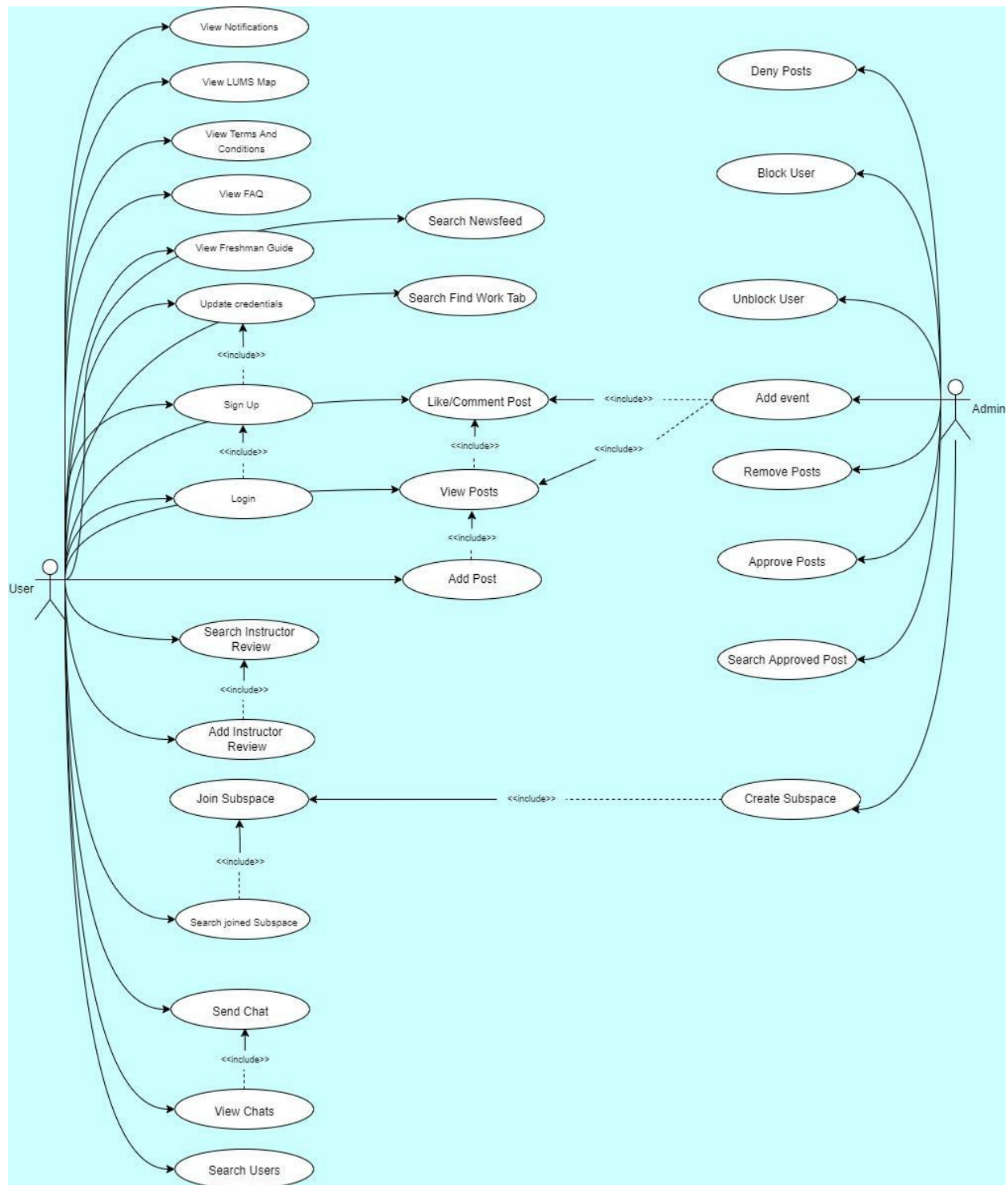
2.7 Use Case View

2.7.1 Use Case Table

2. Use Case table

Primary Actor		Associated Use cases
Both Admin and Students	1	Login
	2	Update Credentials
Students	1	SignUp
	2	Add Post
	3	Delete a Post
	4	Like a Post
	5	View Post
	6	Comment on a Post
	7	View LUMS Map
	8	View all Notifications
	9	View a specific notification
	10	Search Users
	11	Search joined subspace
	12	Search Find work Tabs
	13	Search Newsfeed
	14	Search Instructor Review
	15	Add Instructor Review
	16	View all chats
	17	View a specific chat
	18	Send Chat
	19	View Terms and Conditions
	20	View FAQs
	21	View Freshman guide
Admin	1	Search Approved Posts
	2	Create Subspace
	3	Block User
	4	Unblock User
	5	Approve Post
	6	Deny Post
	7	Remove Post
	8	Add events

2.7.2 Use Case Diagram



2.7.3 Use Case Description

Use Case 1:

Search Instructor Reviews	
ID	UC1
Actors	Students
Pre-conditions	The Actor wants to search a review for a specific instructor.
Flow of Events	<ol style="list-style-type: none">1. The Actor will log in.2. The Actor will go to the instructor tab.3. The Actor will search the name of the instructor in that tab4. All the results similar to the name of the instructor will be displayed.
Post-conditions	The Actor will be able to view instructor reviews

Use Case 2:

Search Find Work Tab	
ID	UC2
Actors	Students
Pre-conditions	The Actor wants to search for relevant posts in the Find Work tab.
Flow of Events	<ol style="list-style-type: none">1. The Actor will log in.2. The Actor will go to the Find Work tab.3. The Actor will search a keyword in the tab4. All the results similar to the keyword will be queried and displayed to the user.
Post-conditions	The Actor will be able to view all the work available.

Use Case 3:

Send Chat	
ID	UC3
Actors	Students
Pre-conditions	The Actor is logged in and wants to chat with another user.
Flow of Events	<ol style="list-style-type: none">1. The Actor will log in.2. The Actor will go to the Chats tab.3. The Actor will select/search a user in the message tab4.The Actor will input the text in the text-box and/or attach a picture5.The Actor will click the “send” button
Post-conditions	The message typed by the Actor will be sent to that particular user.

Use Case 4:

Approve Posts	
ID	UC4
Actors	Admin
Pre-conditions	The Admin is logged in.
Flow of Events	<ol style="list-style-type: none">1. The Actor will log in.2. The Actor will go to the “pending posts” tab.3. The Actor will select a specific post in the pending tab4. The Actor will approve the post by clicking the approve widget.
Post-conditions	Approved posts will be visible to other users.

Use Case 5:

Add Posts	
ID	UC5
Actors	Student
Pre-conditions	The actor is logged in and wants to add a new post.
Flow of Events	<ol style="list-style-type: none">1. The Actor will log in.2. The Actor will click the “Post” button on the homepage.3. The Actor will write a text and/or upload a picture on the new screen.4. The Actor will click the post widget.
Post-conditions	The post(s) will be displayed on the newsfeed for students and will go under the “Approved Posts” tab on the admin side.

The system will be a self-contained product, not dependent on any previous versions.

3 Other Non-functional Requirements

3.1 Performance Requirements

- The system should be easy to use. This can be observed by how quickly the user is able to complete a certain task.
- In case of a system crashing, the service downtime should not be more than a few hours.
- Average response time for every click must be less than 2-3 seconds. Larger response time would result in users losing interest and may end up deleting the application.

- As more users start joining this platform, more storage would be required, hence the system must be able to store data of a large number of users.
- The system should be reliable. All the credentials must be saved in the database.
- Changes such as deleting a post, updating credentials should be immediately visible without any delay.

3.2 Safety and Security Requirements

- A strong password should be implemented at the Signup Screen.
- Accounts and user's personal information should be secured and should not be hackable.
- Database backup of users data will be done one a week.
- Passwords and other important data will be stored in encrypted format in the database.
- SSL encryption will be used for passwords to make data meaningless in case a hacker gets access.

3.3 Software Quality Attributes

3.3.1 Availability

The system should be available at all times so that the customers can avail all the functionality of the application at any time of the day. In case of any errors on the server side, the application will display an error message so that the users are fully aware.

3.3.2 Performance

The system should be designed in a way that it minimises delay and response time. It should be able to handle process interrupts, http requests and query in an efficient manner. In case of a large number of requests, the system should not crash.

3.3.3 Usability

The application should have a user-friendly interface. Users will also be able to ask queries and provide feedback through the application and they will be answered directly through email.

3.3.4 Maintainability

The system will be designed with minimum errors and bugs and these errors will be fixable. A structured code will be written to build this application so that a new and extended functionality could be easily added. As more users will start using this application, more functionality will be needed.

3.3.5 Testability

The system will allow parallel testing of use cases to take place with the development phase. Minor bug and error fixes will then take place accordingly such that all corner cases are covered extensively.

3. Appendix A – Top 10 User Stories

1. As a user, I would like an easy to navigate app that offers a smooth experience.
2. As an admin, I would like to be able to add or remove events for people to join and become aware of.
3. As a user, I would prefer to see where my friends are on the map in real time.
Limitation(s): We may not be able to complete this due to course time limitations.
4. As a user I would like to be able to chat with anyone without needing to send a friend request just search for a name and send a message.
5. As a user, I would like to see some information for students in their freshman year, what to do and what not to do as well as a guide of overall life on campus.
6. As a user, I would like to be able to order food from all on campus restaurants using the app.
Limitation(s): The idea is too distant from a social media app concept and may lead to confusion with regards to what the application is about.
7. As the user, I would like to see a section dedicated to animal rescue to spread awareness of any current animal issues or overall animal society concerns.
8. As a user, I would like to see a special place dedicated to music where people can share their favorite songs as well as have discussions about their favourite artists or instruments.
9. As a user, I would like to have a section for people who want to find new friends and people with similar interests.
10. As an admin, I would like to be able to see statistics about how many users are using the app at a particular time.
Limitation(s): We may not be able to complete this due to lack of experience with google analytics or other similar software

4. Appendix B – Architectural Spike (One Story)

In order to keep the chat section up-to-date and fresh, either the application can send asynchronous requests to the server after a reasonable duration so it does not cause excess load on the network resulting in very high latency, or we can refresh our chats whenever a user sends a message. The latter approach is less optimized as a large number of packets may be exchanged resulting in network load and more data will need to be re-rendered resulting in performance overhead.

5. Appendix C - Group Log

Meeting with TA 01:

Date 07/02/22

Time: 1:30pm - 2:00 pm

Members Present: Nashrah Shaukat, Ibrahim Sanaullah, Zeerak Babar, Haseeb-ur-Rehman Faheem, Aswad Tariq

Discussion:

- Formal Introduction with TA
- Approval of project proposal
- Talk about advantages of using Firebase for storage and backend.
- Overview of document TA instructed us to focus on Functional Requirements the most and first.
- Discussion about Flutter being a good tool for mobile development.

Group meeting 01

Date; 08/02/22

Time: 4:30 pm - 6:30 pm

Discussion: All Group members go over the SRS document and start to decide how to approach it.

Group meeting 02

Date: 09/02/22

Time: 3:30 pm - 5:00 pm

Discussion: Further discussion about document content and points set up about functional requirements decision to conduct interviews the next day for data collection. Discussion over product perspective diagram, Confusion over User characteristics.

Group meeting 03

Date 09/02/22

Time 8:45 pm -10:15 pm

Discussion: Completing assumptions and dependencies, relooking user characteristics 2.4. Admin side functions discussed and added discussion on relationship between users and friends concept to be implemented or not. User profile elements.

Group meeting 04

Date 11/02/22

Time: 4:30 pm - 6:00 pm

Discussion: More discussion about exact functional requirements and how the application is supposed to work. As well as what functionality users should have and what pathways they will use as well as discussion over a global search throughout the application or local searches in each subspace. Local searches were chosen. Analysis of all collected user stories and discussion on which to put in top 10 and which ones are possible and feasible to implement and why certain stories are not possible to implement.

Group meeting 05

Date 12/02/22

Time: 4:00 pm - 6:00 pm

Discussion: Completed external interface requirements Section 2.6 as well as all functional requirements discussion to select use cases top 5 most important use cases. Filling in section 1.4 with all acronyms and abbreviations used throughout the document as well as deciding Group member contributions.

Meeting with TA 02:

Date 14/02/22

Time: 1:30 pm - 2:00 pm

Discussion:

- Minor confusions regarding diagrams for example how a use case diagram should be structured.
- Some confusion about user stories that are not feasible to implement in the application show limitations of all stories that are not being implemented.
- How to categorize functional requirements and user stories and what categories make sense with our requirements.
- Discussion about priorities of use cases,
- General discussion on Software and Hardware interface of the application

Group meeting 06:

Date 14/02/22

Time: 9:00 pm - 10:00 pm

Discussion: Final touches to document. Proofreading for mistakes and checking categories of all functional requirements and the order they are placed. Double checking of group member contributions and Appendix B - Architectural spike discussion based on TAs message on slack.

Note: All group members were present throughout all group meetings. And the above meeting notes are ordered chronologically.

6. Appendix D – Contribution Statement

Name	Contributions in this phase	Approx. Number of hours	Remarks
Aswad Tariq	1.3 Intended audience and document overview 2.5 Seven Use Cases 2.7.3 Use Case 5 Appendix C - Group Log	10	
Haseeb-Ur-Rehman Faheem	1.1 Document Purpose 1.2 Product Scope 2.5 Ten Use Cases 3.2 External Interface Requirement 2.7.3 Use case (2, 3)	10	
Zeerak Babar	2.2 Product functionality 9 2.5 Six use cases 7.1 Performance requirements Data flow diagram Use case diagram	10	
Nashrah	2.5 Six Use Cases 4.2 Safety And security requirements 2.7.3 Use case 4 2.1 Product Perspective & Diagram 1.4 5 terms	10	
Ibrahim Sanaullah	2.3 Users and Characteristics 2.5 Seven Use Cases Categorisation of Use Cases Appendix A - User Stories Formatting of Final Document	10	