Requirements Document - Group 5

Malik Hannan Ahmed - gusmalikah@student.gu.se
Julia Ekeblad - juliaeke@chalmers.se
Dzhan Hasan - gushasandz@student.gu.se
Johan Lindén - johedvi@chalmers.se
Ingrid Stake - ingridst@chalmers.se
Xiaoran Zhang - xiaoran@chalmers.se
Muzhe Zhou - muzhe@chalmers.se

November 20, 2023

Contents

1 High-Level Description **3**
1.1 Goal & Scope . . . 3
1.2 Business Case and Stakeholder Map . . . 4
1.2.1 Business Case . . . 4
1.2.2 Goal Domain Tracing . . . 5
1.2.3 Stakeholder Map (Analysis) . . . 5
1.3 Core Functionality . . . 6
1.4 Non-functional Requirements . . . 7
1.4.1 Quality Grid . . . 7
1.4.2 Performance Requirements . . . 8
1.4.3 Specific Quality Requirements . . . 8
1.4.4 Constraints . . . 9

2 Operational Description **10**
2.1 Functional requirements . . . 10
2.2 Data requirements . . . 18
2.2.1 Class diagram . . . 18
2.2.2 Data dictionary . . . 18
2.3 Proposed Priorities . . . 21

3 System/Product Requirements **22**
3.1 System Requirements . . . 22
3.2 UI Prototype . . . 24
3.3 Detailed data Requirements . . . 24

4 Cross-cutting **27**
4.1 Traceability . . . 27
4.2 Rationale and appropriate detail . . . 27

5 Conclusion **29**

6 References **30**

Versions and Authors

Date	Version	Comment	Author
2023-9-18	1.0	First release, version submitted as R1. Focusing on High-Level description.	M. Hannan Ahmed, J. Ekeblad, D. Hasan, J. Lindén, I. Stake, X. Zhang and M. Zhou
2023-9-29	2.0	Remade the context diagram to better reflect our domain, added goal domain tracing, added a use-case diagram, updated our quality grid, refined our quality requirements, refined our functional requirements to a higher level, refined class diagram to a higher level, added proposed priorities, added our UI prototype, added detailed data requirements, added a traceability matrix, added rationale and appropriate detail section.	M. Hannan Ahmed, J. Ekeblad, D. Hasan, J. Lindén, I. Stake, X. Zhang and M. Zhou
2023-10-20	3.0	Linked FR1 to UC1, added Priority-ordered proposed priorities, fixed minor mistakes on User diagram, added a concluding paragraph of Requirement document, added cross-cutting introduction, added numbers from 4.2 to performance requirements, paraphrased sentences and corrected errors in expression.	M. Hannan Ahmed, J. Ekeblad, D. Hasan, J. Lindén, I. Stake, X. Zhang and M. Zhou

Note: More detailed information about the authors can be found on the first page of this document.

1 High-Level Description

This document provides an overview of the requirements for the *CampusHub* (Campus Hub) platform. The primary purpose of the platform is to provide university students from Chalmers University of Technology (Chalmers) and University of Gothenburg (GU) with an opportunity to connect, socialise and communicate with each other. The platform can also serve as a means to organise and gather information about events and activities. As well as find help solving any basic questions when they are new to one of the universities. This will be enabled through the use of categories and posts in an online forum - CampusHub. The platform will be limited to only students and will utilise the university identification to ensure a successful login. To avoid harassment and toxicity, a team of students will monitor and moderate the platform. Companies will also be able to interact with students and post useful information about e.g. job opportunities or lunch lectures.

1.1 Goal & Scope

The goal of the platform is to enable students to connect with each other by creating a simple and easy-to-use user interface. The diagram in Figure 1 outlines the domain of the system and how it engages with different entities. The system serves university students as users and establishes two external connections and one internal. One of these connections involves a monitoring system, depicted as *Moderation*. The first external connection involves interactions with companies via their representatives. These relationships enable the platform to feature advertisements, such as job postings for students and other pertinent information. The other external connection enables the platform to ensure that the users are valid students.

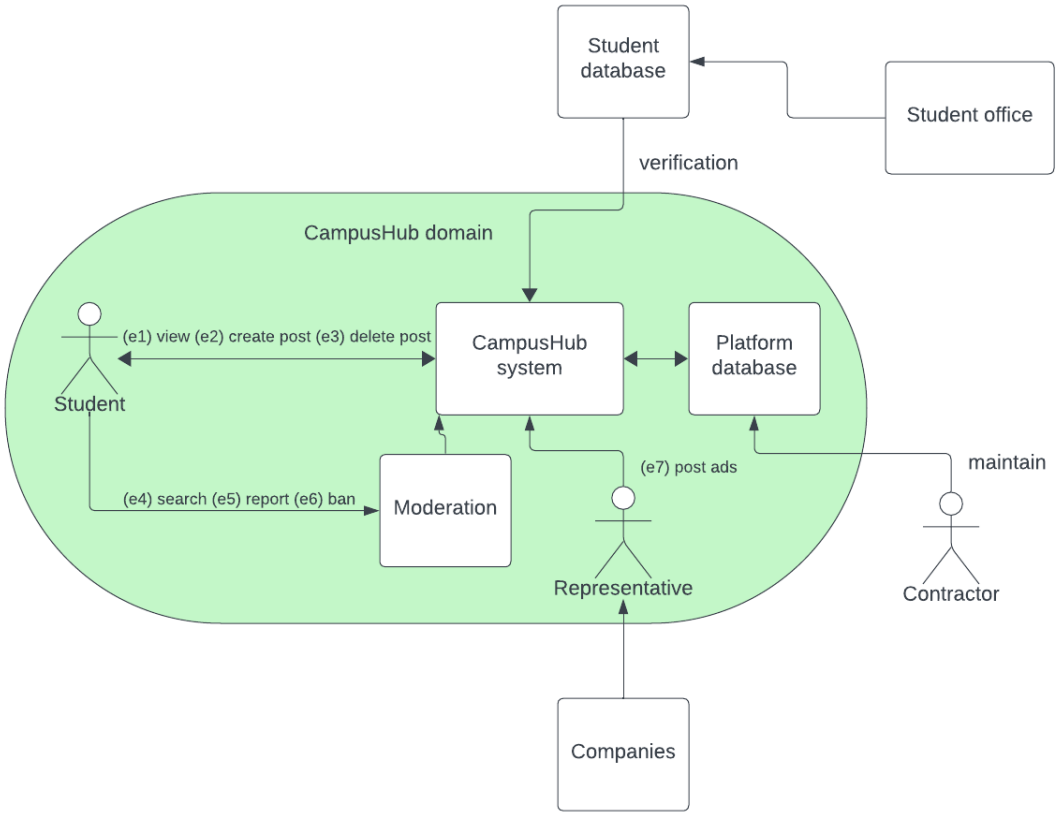

Figure 1: A context diagram for CampusHub showing the main relationships between different entities.

1.2 Business Case and Stakeholder Map

This section of the report explains what audience the software appeals to, the stakeholders identified for the project, and in which ways the software is going to be worth the resources spent.

1.2.1 Business Case

Below is a list of the business goals identified for the platform, specified in no particular order.

Business Goal 1: Increase both company reputation among students, and students' employment rates. By advertising on the platform, companies have the chance to promote their products and provide job opportunities for students.
Business Goal 2: Increase student well-being by providing a platform to make friends, socialise and share their interests. This is done by posting their ideas, opinions and invitations to student-organised events on the platform.
Business Goal 3: Enhance the quality of education by offering the universities the platform, which will be financed by student unions and companies. The platform aims to facilitate academic assistance and answer-seeking among students.

1.2.2 Goal Domain Tracing

This section presents how the different events identified in Figure 1 can be traced to one or more of the business goals. The result is shown in Table 1.

Goal	(e1) view	(e2) create post	(e3) delete post	(e4) search	(e5) report	(e6) ban	(e7) post ad
Goal 1							x
Goal 2	x	x	x	x	x	x	
Goal 3	x	x	x	x	x		

Table 1: A Goal Domain Tracing Table relating events to the different business goals.

1.2.3 Stakeholder Map (Analysis)

Table 2 demonstrates the stakeholder map for *CampusHub*. In the table, major stakeholders are highlighted along with their relationship with the platform, how they influence the platform and their level of interest in the platform.

Name	Relationship	Influence	Interest
Students	Users & Managers	High	High
Chalmers & GU	Customers	Moderate	Moderate
Student Office	Collaborators	Moderate	Moderate
Student Unions	Collaborators	Low/Moderate	High
Companies	Customers	Low	High
Uni Buddy	Competitors	Low	High

Table 2: The stakeholder map shows the name, relationship, influence and interest of all stakeholders

Students: Students represent the core user base of the platform, using it to connect with peers, share information, and engage in discussions. They influence the platform's content and dynamics through their interactions, making their active engagement essential to its success. Feedback and preferences from students should be carefully considered during development to ensure that the platform meets their needs and expectations.

Companies: Companies are external stakeholders interested in the platform primarily for recruitment and advertising purposes. They can utilise the platform to post job opportunities, internships, and advertisements targeted at the student population. While they can be considered customers to some extent due to their usage of the platform's advertising features, their influence on the platform's overall direction and features is relatively low. However, their presence can add value to the platform by providing students with relevant career opportunities. This is an important stakeholder for the project since this group will be one of the platform's main sources of income.

Student Office: The Student Office represents the administrative arm of the university responsible for managing student records and data. They require access to student information for various administrative purposes, such as enrolment, academic advising, and student support services. Additionally, they may have a role in shaping the platform's policies, ensuring compliance with data privacy regulations, and negotiating data-sharing agreements. Their involvement is crucial for maintaining data integrity and security on the platform.

University: The university is the overarching institution that hosts the students and the platform. While they may not be direct users of the platform, they have a vested interest in its success. The university can potentially become a customer by financially supporting the platform's development and operation if it aligns with its goals for enhancing student well-being, engagement, and campus life. Their support can go beyond financial backing and may include providing input into the platform's

strategic direction, ensuring it aligns with the university's broader mission and objectives.

Competitors: Competitors are other organisations or platforms that offer services similar to what the *CampusHub* platform aims to provide to students. While they are not customers or collaborators, they are significant stakeholders because they represent a competitive force in the market. Understanding their strategies, offerings, and user base is essential for your platform's competitiveness. Monitoring competitor activities and differentiating your platform's value proposition can help you stay ahead in the market.

Student Unions: Student Unions are student-run organisations within the university that organise events, activities, and services to enhance the student experience. They serve as advocates for students and often play a vital role in building a sense of community on campus. Student Unions may find the platform valuable for reaching students with information about events, meetings, and other activities. Collaborating with Student Unions can facilitate effective engagement with the student body and enhance the overall platform experience.

1.3 Core Functionality

This section demonstrates the core functionality of the system, which mainly focuses on Posting and Replying, Direct Messaging, Forum Categories, Follow Features, Platform Accessibility, Content Sorting, Anonymous Posting, System Monitors, and company advertisements. This section also includes a use case diagram, which can be seen in Figure 2. The diagram provides an overview of key actors and key functions.

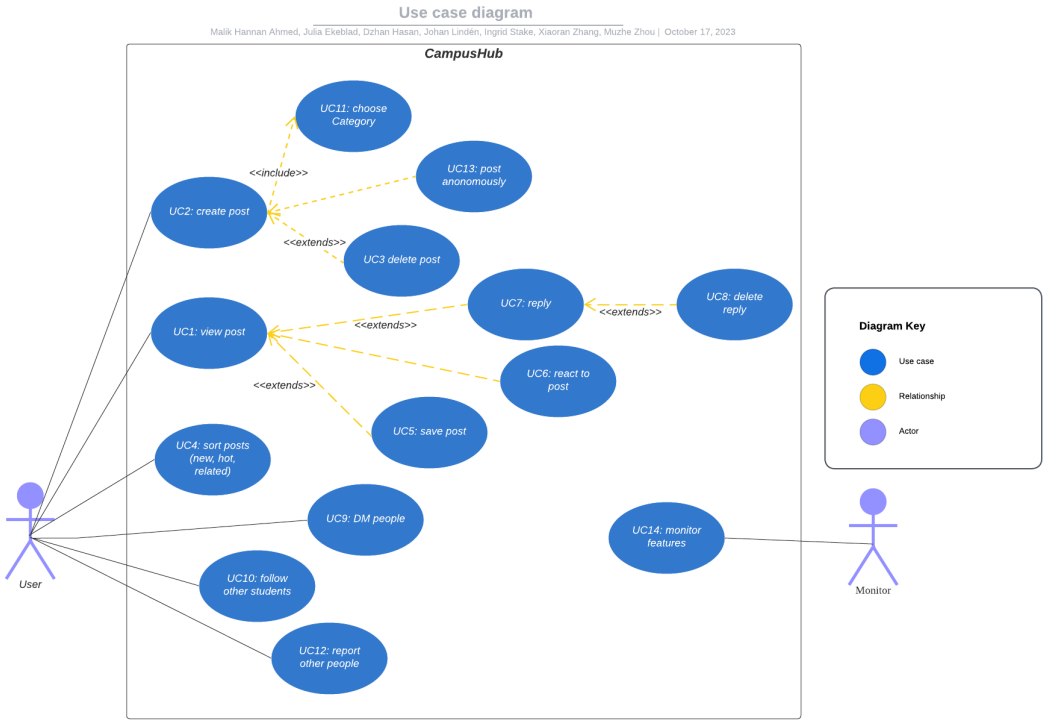

Figure 2: User case diagram

UC1: As a user, I want to view posts so that I can get useful information.

UC2: As a user, I want to create posts so that I can express my ideas and seek others' help.

UC3: As a user, I want to delete my own posts so that I can undo them.

UC4: As a user, I want to sort posts so that I can find what I'm searching for faster.

UC5: As a user, I want to save posts so that I can save them for future use.

UC6: As a user, I want to be able to add reactions to others' posts so that I can interact with them.

UC7: As a user, I want to comment so that I can interact with others' posts.

UC8: As a user, I want to delete my comments to undo any mistakes in them.

UC9: As a user, I want to send direct messages so that I can communicate with others.

UC10: As a user, I want to follow students so that I can receive notifications whenever they post or engage in discussion.

UC11: As a user, I want to choose a category when creating a post so that I can find content I am interested in.

UC12: As a user, I want to be able to report any inappropriate behaviour in order to promote a pleasant experience.

UC13: As a user, I want to be able to post anonymously so that I can share opinions without revealing my identity

UC14: As a monitor, I want monitor features so that I can resolve any inappropriate activities or problems on the platform.

1.4 Non-functional Requirements

This section aims to highlight the non-functional requirements and attributes related to the project goal. More specifically, the *Constraints*, *Performance-* and *Quality Requirements* are relevant for providing a platform offering new ways for students at Chalmers and GU to communicate and connect with each other.

1.4.1 Quality Grid

Table 3 provides an overview and importance assessment of qualities based on quality standards ISO 9126 and McCall and Matsumoto.

	Critical	Important	As usual	Unimportant	Ignore
Operation			x		
Integrity/Security			x		
Correctness			x		
Reliability/Availability		QR1			
Usability	QR4	QR3			
Efficiency/Performance		QR2			
Revision					
Maintainability			x		
Testability			x		
Flexibility			x		
Transition					
Portability		QR5			
Interoperability			x		
Reusability			x		
Installability			x		

Table 3: A quality grid for the platform. Sections 1.4.2 and 1.4.3 will explain the reasoning behind requirements that possess an unconventional order of prioritization.

1.4.2 Performance Requirements

For the student's experience on the platform to be as positive as possible, there are a few performance-related aspects that need to be in place. This section outlines these critical performance requirements and their significance in achieving the platform's goals. The numbers specified are a result of section 4.2.

QR1: The platform shall be able to host 40,000 user accounts.

As of 2022, approximately 11,000 students attended Chalmers [1], the latest corresponding number for GU is from 2021 and sums to approximately 29,000 students [2]. Even though each of the total 40,000 students might not sign up and use the platform, it still is reasonable that all of them should be able to do so. This results in a performance requirement for the platform to support concurrent access by a minimum of 40,000 users, ensuring that it can accommodate the entire student population of both Chalmers and GU.

QR2: The platform shall load new pages and content within 4 seconds.

For students to use the platform, they need to feel that it is a sufficiently effective tool for communication. Only after that point will the platform be able to fulfil its purpose. One thing that can throw users off when using a system is slow loading speeds. Thus, the platform shall load new pages and content within x seconds.

1.4.3 Specific Quality Requirements

To continue on the same track as *QR2*, usability is another factor to improve the users' experiences on the platform and thus, increase the likelihood of them continuing to use it. One important thing to consider is that among the 40,000 students, the experience and skills when it comes to using online platforms will vary. Especially for the sake of students new to the Universities (and potentially also to the country), it is important to keep usability in mind when designing the platform. Doing so means that less time and effort will be spent understanding the platform and its functionality, allowing it to be used for its purpose sooner and with more ease. With a platform that is easy to use, fewer students will refrain from using the platform due to a lack of skills needed to use it, and thus make it more inclusive. *QR3* and *QR4* specify quality requirements aimed to ensure the usability of the platform. Again, the numbers specified are a result of section 4.2

QR3: 95 of 100 users shall find relevant categories within 2 minutes after signing up

to the platform.

This quality requirement ensures that the concept of the platform and its categories have been conveyed well to the users while signing up and entering the platform. Further, it also establishes that the process of finding and selecting categories of interest has a decently intuitive flow. If this requirement is met, the first-time users have the opportunity to start engaging with other students sooner.

QR4: 98 of 100 users shall be able to create a post within 1 minute, excluding the time taken for composing the post content.

While *QR3* focuses on the users who sign in to the platform for the first time, *QR4* includes all users. Its cause is to ensure that the flow of creating or interacting with posts and threads is well-designed and productive. This has significance for the platform and its purpose since it is through these posts the students will interact with each other.

QR5: The platform shall be available as an app and on the web

Another matter of inclusivity is that all students should have easy access to the platform on their devices. Thus motivating *QR5*, an app version should be available on all phones, making the platform easily accessible when the students are on the move. Further, a web page will make the platform available from any computer, regardless of its operating system.

1.4.4 Constraints

As for all organisations processing personal data, the platform must comply with data protection laws and regulations according to the *Swedish Authority for Privacy Protection* (*IMY*)[3]. This serves as a development constraint for the platform.

QR6: The system must adhere to GDPR data protection requirements

2 Operational Description

This section provides a comprehensive overview of the operational aspects associated with the platform. That is specifications of the functional requirements and data requirements. Additionally, the section also includes a proposed priority of the functional requirements.

2.1 Functional requirements

FR1: The system shall allow the user to view posts by supporting Task 1.

Task 1	View Post	
Purpose:	Support viewing posts from the mobile app, browser-enabled device.	
Trigger:	User clicks the 'posts' button.	
Pre-condition:		
Frequency:	User might view posts 5-10 times every day.	
	Sub-Tasks:	**Example Solution:**
1.	Select or search for post	Jump to the selected post page.
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	

FR2: The system shall allow the creation of posts by supporting Task 2.

Task 2	Create Post	
Purpose:	Support creating post. Restrict access to authorised users.	
Trigger:	User clicks the 'New post' button	
Pre-condition:		
Frequency:	User might create 3-6 posts every day.	
	Sub-Tasks:	**Example Solution:**
1.	Show post detail	Show the post's comments, likes, favourites and related posts.
2.	Select if post should be anonymous or not.	
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	

FR3: The system shall allow deleting posts by supporting Task 3.

Task 3	Delete Post	
Purpose:	Support deleting posts which users have created. Restrict access to authorised users.	
Trigger:	User click the 'Delete post' button	
Pre-condition:	The user has already created at least one post.	
Frequency:	User might delete 1-2 posts every day.	
	Sub-Tasks:	**Example Solution:**
1.	Select post to delete	Show a list of posts that the user had created.
2.	Press yes in the additional confirmation window.	
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	

FR4: The system shall allow content sorting by supporting Task 4.

Task 4	Sort Posts	
Purpose:	Sorting content based on user preferences, including most liked posts and newly posted activities/questions.	
Trigger:	User click the 'sort' button	
Pre-condition:		
Frequency:	Multiple times a day	
	Sub-Tasks:	**Example Solution:**
1.	Select how user wants to sort the content	Show a list of ways to sort
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	

FR5: The system shall allow to save posts by supporting Task 5.

Task 5	Save Posts	
Purpose:	Support Saving posts under the post. Restrict access to authorised users.	
Trigger:	User click the 'Save' button	
Pre-condition:		
Frequency:	User might save 1-3 posts every day.	
	Sub-Tasks:	**Example Solution:**
1.	Find post to save	
2.	Save post to users profile	Visual confirmation
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	
2a.	If user already saved the post and presses the "Save" button it will instead unsave it.	

FR6: The system shall allow reacting to posts by supporting Task 6.

Task 6	React to posts	
Purpose:	Support add reactions under the post. Restrict access to authorised users.	
Trigger:	User click the 'reaction' button	
Pre-condition:	The user is on the post page.	
Frequency:	User might react to posts 5-10 times every day.	
	Sub-Tasks:	**Example Solution:**
1.	Find post or comment to react on.	
2.	Show previous reactions	
3.	Select from pre-defined emojis to react with.	
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	
2a.	If user presses same emoji, on same comment or post, the emoji will instead be deleted.	

FR7: The system shall allow replying to posts by supporting Task 7.

Task 7	Reply	
Purpose:	Support replying to the post and comments. Restrict access to authorised users.	
Trigger:	User click the 'Reply' button	
Pre-condition:	The user enters the post detail page.	
Frequency:	User might make 5-10 comments every day.	
	Sub-Tasks:	**Example Solution:**
1.	Find comment or post to reply on.	Show all the comments under the post.
2.	Write reply	
3.	Select if reply should be anonymous or not.	
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
1b.	If the user is not registered, ask to go to the register page.	

FR8: The system shall allow deleting comments on posts by supporting Task 8.

Task 8	Delete Comment	
Purpose:	Support deleting comments under the post. Restrict access to authorised users.	
Trigger:	User click the 'Delete comment' button	
Pre-condition:	The user left a comment.	
Frequency:	User might delete 5-10 comments every day.	
	Sub-Tasks:	**Example Solution:**
1.	Show post details	Show the post's comments, likes, favourites and related posts.
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page.	
2a.	If the user is not registered, ask to go to the register page.	

FR9: The system shall allow direct messaging by supporting Task 9.

Task 9	Direct Messaging	
Purpose:	Support sending private messages to each other for direct communication. Restrict access to authorised users.	
Trigger:	User click the 'Private message' button on other users' main page.	
Pre-condition:		
Frequency:	User might send 0-100 private messages every day.	
	Sub-Tasks:	**Example Solution:**
1.	delete direct message	User can press message that sent to other users and choose to delete within 5 minutes.
2.	block other user	User can press 'block' button next to other users' avatar.
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page	
2a.	If the user is not registered, ask to go to the register page.	

FR10: The system shall allow a follow feature by supporting Task 10.

Task 10	Follow Features	
Purpose:	Support following other students to receive notifications whenever they post or engage in discussions.	
Trigger:	User click the 'Follow' button on the bottom.	
Pre-condition:		
Frequency:	User might follow 0-10 students every day.	
	Sub-Tasks:	**Example Solution:**
1.	unfollow other students	the 'follow' button will change to 'unfollow' when this student is followed by user.User can press 'unfollow' button.
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page	
2a.	If the user is not registered, ask to go to the register page.	

FR11: The system shall allow sub-forums or categories by supporting Task 11.

Task 11	Forum Categories	
Purpose:	Support different sub-forums or categories, such as selling course literature, sports activities, pubs, and more. Users can choose the category that matches their interests.	
Trigger:	User click the 'Categories' button on the bottom.	
Pre-condition:		
Frequency:	The platform might have some built-in categories and new categories might be created once a month.	
	Sub-Tasks:	**Example Solution:**
1.	Show the most popular categories	sort the categories by the popularity and put them in ascending order.
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page	
2a.	If the user is not registered, ask to go to the register page.	

FR12: The system shall allow reporting of other people by supporting Task 12.

Task 12	Report Other People	
Purpose:	Support reporting features and make the platform a peaceful and comfort place.	
Trigger:	User click the 'Report' button on the bottom.	
Pre-condition:		
Frequency:	The user might report 0-1 per week.	
	Sub-Tasks:	**Example Solution:**
1.	Block the reported user	jump to the user's page and click the 'block' button.
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page	
2a.	If the user is not registered, ask to go to the register page.	

FR13: The system shall allow anonymous posting by supporting Task 13.

Task 13	Anonymous Posting	
Purpose:	Support users have the option to post or reply anonymously, allowing them to seek help or share opinions without revealing their identity.	
Trigger:	User toggle the 'anonymous' option	
Pre-condition:	User enter the creating post page	
Frequency:	3-6 times a day	
	Sub-Tasks:	**Example Solution:**
1.	Exit anonymous mode	user click the 'stop anonymous' button when they are posting or replying
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page	
2a.	If the user is not registered, ask to go to the register page.	

FR14: The system shall allow monitor features by supporting Task 14.

Task 14	Monitor Features	
Purpose:	Support monitor features to make the platform more user-friendly.	
Trigger:	User reports a comment or a post with the report button	
Pre-condition:	The user is logged in on a moderator account	
Frequency:	User might ban 0-10 accounts every day.	
	Sub-Tasks:	**Example Solution:**
1.	Ban user	Show list of reported comments or posts
2.	Language Control	Automatically language check
	Variants:	
1a.	If the user is not logged in, ask to go to the logging page	
2a.	If the user is not registered, ask to go to the register page.	

2.2 Data requirements

2.2.1 Class diagram

DR1: As seen in the Figure 3, the implementation of the core functionalities of the system requires data from various entities such as users and messages. The system will encapsulate and process these data as depicted in the class diagram.

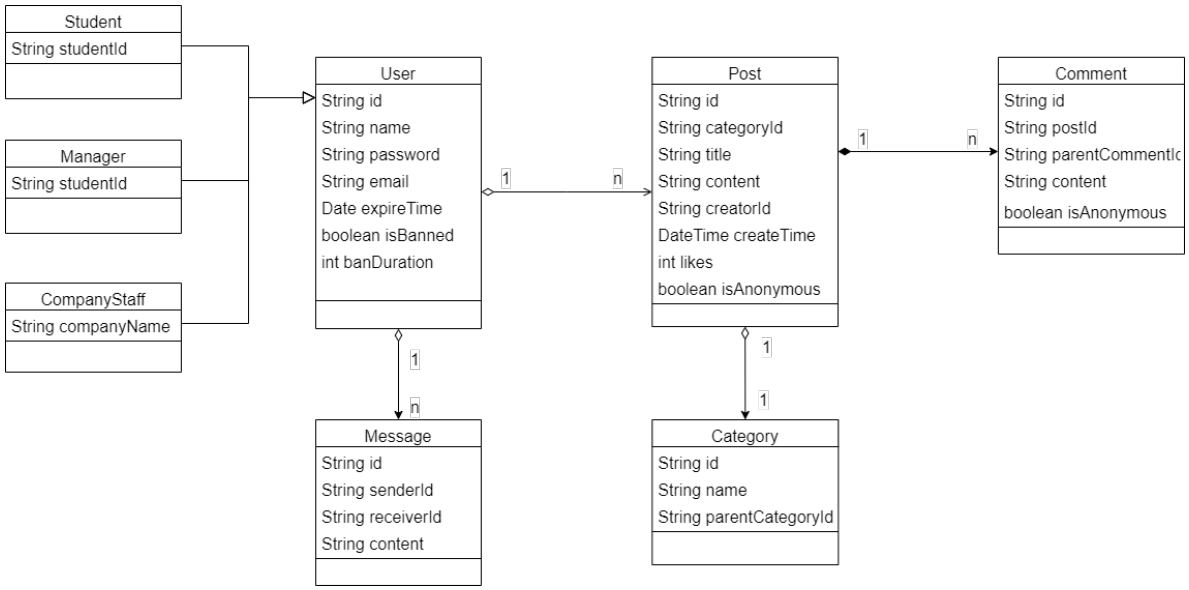

Figure 3: The Class diagram shows the different entities, their properties and how they are related.

2.2.2 Data dictionary

Class: User
This class encapsulates the personal information of all users, providing the necessary data for *FR10*. Since the method for verifying student identity is provided by the university, this system does not mandate the acquisition of any personal information beyond the username. For managers or external company staff, there are alternative methods of logging in besides student identification.

Attributes:
id Identifies a user uniquely.
email The email address provided by the user. It's used as the login account for managers or external company staff.
password It's only used as the login password for administrators or external company employees.
expireTime Indicates the time at which the account expires. This system primarily targets current students, and therefore, access privileges are revoked for a certain period after graduation.
isBanned Represents whether the account is banned. As described in *FR12* and *FR14*, the system incorporates reporting and monitoring functionality. If any violations of usage policies are detected, the account will be banned, prohibiting the use of system functionalities.
banDuration Indicates the duration of the ban.

Class: Student
This class is an extension of the User class and represents student users, storing information related to student identity verification.

Attributes:
studentId A unique identifier used for verifying the student's identity.

Class: Manager
Represents administrator users, who have privileges for actions such as user registration and user banning.

Attributes:
studentId A unique identifier used for verifying the student's identity. When an administrator is also a student user, this identifier needs to be associated with the student identity verification.

Class: CompanyStaff
Represents external company employees, who are manually registered user accounts by administrators and have permission to post advertisements.

Attributes:
companyName Indicates the name of the company where the staff is employed.

Class: Post
This class encapsulates all the information about each post, providing the necessary data for *FR3*, *FR4*, and *FR5*. When a user creates a post as part of *FR2*, the provided information is stored in this class. It does not contain detailed data about internal replies; it serves as an overall representation.

Attributes:
id A unique identifier for distinguishing different posts in the system.
categoryId Indicates the category to which the post belongs.
title Describes the title of the post.
content Describes the content of the post, with some content potentially displayed in the post list.
creatorId Represents the user ID of the post's creator.
createTime Represents the date of post creation.
likes Indicates the current number of likes on a post.
isAnonymous Indicates whether the post was published anonymously. If it is anonymous, the creator's information will not be displayed. It's related to *FR13*.

Class: Comment
This class encapsulates all information about individual replies to a specific post or comment, providing the necessary data for *FR6*, *FR7*, and *FR8*.

Attributes:
id A unique identifier for distinguishing different comments in the system.
postId Indicates the ID of the post to which the comment is directed.
parentCommentId Besides replying to a post, users can also reply to specific comments. This attribute indicates the ID of the comment to which the comment is directed.
content Represents the specific content of the comment.
creatorId Represents the user ID of the comment's creator.
isAnonymous Indicates whether the comment was published anonymously. If it is anonymous, the creator's information will not be displayed.

Class: Category
This class encapsulates information about the categories existing in the system to support *FR11*.

Attributes:

id A unique identifier for distinguishing different categories in the system.
name Represents the name of a specific category.
parentCategoryId Categories can exist under larger categories, indicating the category to which the current category belongs.

Class: Message
This class encapsulates the data of messages sent and received by users when using direct messaging functionality. It stores data for completion of *FR9*.

Attributes:
id A unique identifier for distinguishing different messages in the system.
senderId Represents the sender's ID of the message.
receiverId Represents the receiver's ID of the message.
content Represents the specific content of the message.

2.3 Proposed Priorities

The prioritization found in Table 4 is the result of a *priority grouping* process. During the first iteration, each of the requirements was assigned one of the levels of importance *H*, *M* and *L* (High, Middle, Low). For this first iteration, combinations of importance levels such as *H/M* were allowed. The evaluation of importance levels was conducted with some criteria in mind:

- Maximize customer value
- Minimize development cost
- Minimize Development lead-time
- Maximize fitness to the brand
- Maximize usage among students

Req No.	Requirement description	Priority
FR1	The platform shall allow the user to view posts created by others.	H
FR2	The platform shall allow users to create their own posts	H
FR7	A function for users that allows them to reply to other people's posts and replies	H
FR11	A function for users that allows them to choose a specific category for their post	H
FR13	The platform shall provide the option for the users to post anonymously on the platform	H
FR14	The platform shall have a moderation and content monitoring system to ensure its compliance with community guidelines, policies, and legal regulations.	H
FR3	A function for users that allows them to delete any post they have previously created	M
FR4	The platform shall allow the users to sort content through predefined orders	M
FR6	A function for users that allows them to "react" with emojis to other people's posts and replies	M
FR8	A function for users that allows them to delete their previously written replies	M
FR9	A function for users that allows them to Direct Message (DM) other people	M
FR12	A feature that allows users to report bad behaviour on the platform.	M
FR5	A function for users that allows them to save posts to a personal collection	L
FR10	A function for users that allows them to follow other people	L

Table 4: The table gives an overview of the functional requirements and how they were prioritized using the priority grouping technique.

During the first prioritization iteration, importance was estimated with only customer value, fitness to brand and usage among students in mind. While stakeholders were not actively engaged in this step, we diligently applied these criteria, reflecting the implicit preferences and expectations of these stakeholders. For the second iteration, we conducted a thorough comparison and inspection of requirements to determine if trade-offs could be made. While stakeholders were not actively engaged, we recognized that interdependencies might exist between requirements, especially concerning customer value, fitness to brand, and usage among students. Thus, the aim here was to reach a balance between these while also minimizing development costs and lead time. This prioritization approach made it possible to reflect the stakeholders' implicit influence and preferences.

With the aim to further verify this prioritization, the $100-method was implemented with the modification that stakeholders again were not directly involved. Each of the group members was instead asked to distribute the $100 among the 14 functional requirements. The average distribution is found in Table 5.

Req No.	Requirement description	Priority	Score
FR2	The platform shall allow users to create their own posts	H	$11.95
FR1	The platform shall allow the user to view posts created by others.	H	$11.81
FR7	A function for users that allows them to reply to other people's posts and replies	H	$9.48
FR14	The platform shall have a moderation and content monitoring system to ensure its compliance with community guidelines, policies, and legal regulations.	H	$8.84
FR13	The platform shall provide the option for the users to post anonymously on the platform	H	$7.50
FR12	A feature that allows users to report bad behaviour on the platform.	M	$7.07
FR9	A function for users that allows them to Direct Message (DM) other people	M	$6.65
FR8	A function for users that allows them to delete their previously written replies	M	$6.01
FR3	A function for users that allows them to delete any post they have previously created	M	$5.94
FR11	A function for users that allows them to choose a specific category for their post	H	$5.94
FR5	A function for users that allows them to save posts to a personal collection	L	$5.09
FR6	A function for users that allows them to "react" with emojis to other people's posts and replies	M	$4.52
FR10	A function for users that allows them to follow other people	L	$5.09
FR4	The platform shall allow the users to sort content through predefined orders	M	$4.10

Table 5: The table gives an overview of the functional requirements and how they were prioritized using the priority $100-method. The prior priority score from the priority grouping is also included in the table.

From the table above, it is evident that the two prioritisations overall agree on the most essential aspects of the platform.

With the priority grouping method, we thoughtfully assign importance levels to requirements, considering factors such as customer value, brand alignment, and user relevance. This initial prioritization reflects our insights gained from our own experiences and understanding of the diverse stakeholder landscape. Furthermore, the integration of the $100 model introduces a unique dimension by allowing the group members to 'spend' their imaginary $100 to indicate the relative importance of requirements the way they have understood them through interactions with the stakeholders.

3 System/Product Requirements

3.1 System Requirements

This section will include the system requirements for which the motivation has been taken from [4]. These system requirements represent the minimum specifications the system is required in order to meet the minimum functionality.

Ubiquitous Requirements:

1. The CampusHub platform shall provide university students with the ability to connect, socialise, and communicate with each other.
2. The platform shall allow students to organise and gather information about events and activities.
3. The platform shall enable students to find help for basic questions when they are new to the university.
4. The Campus Hub platform shall be accessible only to Chalmers and GU students.
5. The platform shall utilise university identification for successful user login.
6. The platform shall implement a team of student moderators to monitor and moderate reports to prevent any misbehaviour on the platform.
7. Companies shall have the ability to interact with students and post information about future opportunities.

State Driven Requirements:

1. While a user is not logged in, the Campus Hub platform shall display a "Please log in" message.
2. While a user is browsing the platform, the system shall display relevant posts based on the user's tags and interests.
3. While a student is a new user, the platform shall guide them through the onboarding process.

Event Driven Requirements:

1. When a user clicks on the "Submit Post" button, the Campus Hub platform shall prompt the user to enter post details.
2. When a user sends a direct message, the platform shall notify the recipient of the message.
3. When a new event is created, the platform shall send event notifications to interested students.
4. When a student reports inappropriate content, the platform shall initiate a review by the moderation team.

Optional Feature Requirements:

1. Where the Campus Hub platform includes a mobile application, it shall support all features available on the web version.
2. Where the platform supports user profiles, it shall allow users to upload profile pictures.

Unwanted Behaviour Requirements:

1. If a user attempts to log in with invalid credentials, then the Campus Hub platform shall display an error message.
2. If a user violates the platform's code of conduct, then the platform shall issue a report to the monitors for review.
3. If the platform experiences a security breach, then it shall immediately notify affected users and take steps to mitigate the breach.

3.2 UI Prototype

The prototype of the CampusHub platform is shown in the Figure 4 below. Multiple screens were created to visualise how the user can interact with the platform.

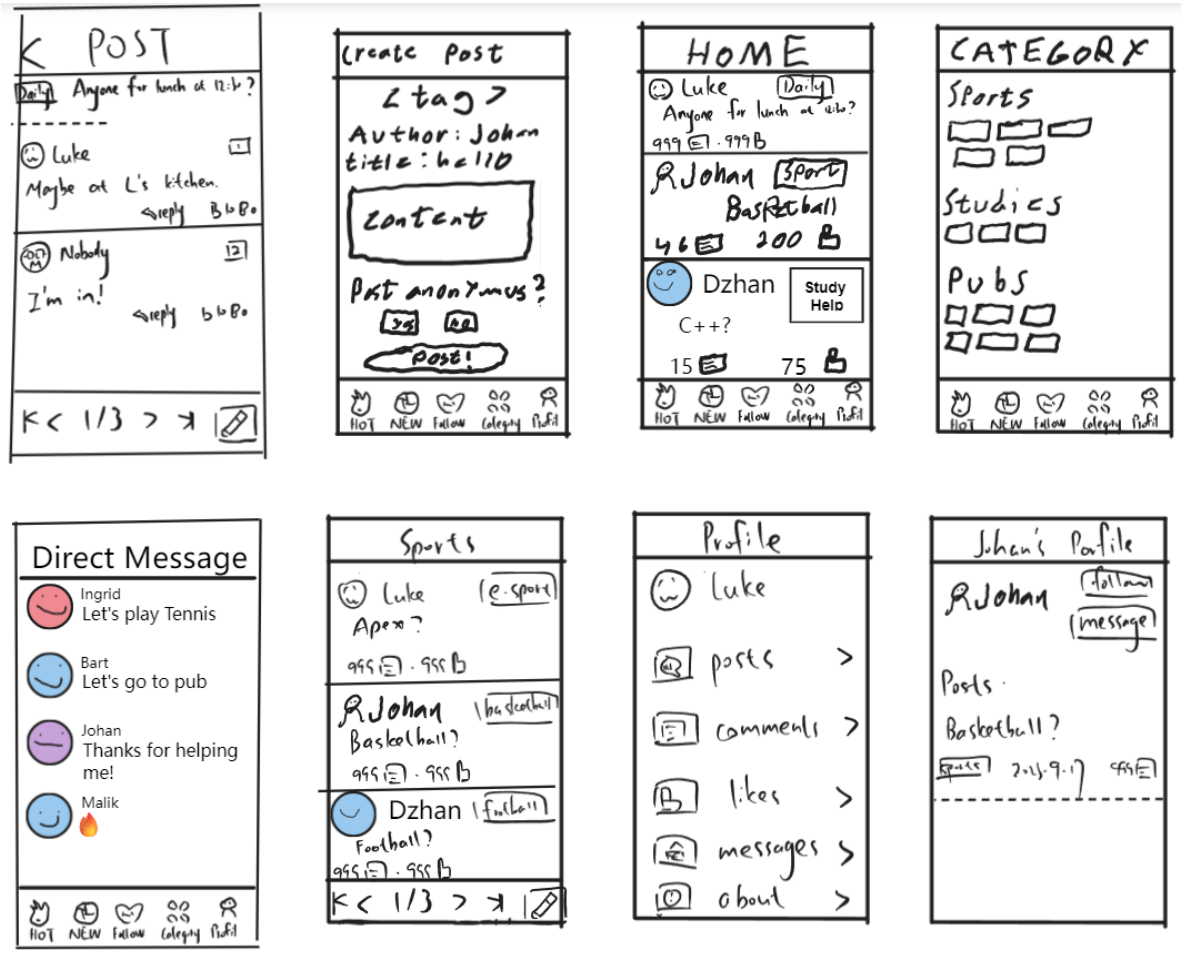

Figure 4: Prototype

3.3 Detailed data Requirements

This section specifies the constraints and detailed properties of data. It is essential for the data stored within the system to conform to these rules to ensure the proper execution of the core functionalities.

For each user, table 6 presents specific data constraints. Regarding the extension of the user to include student and manager, both of them need to maintain a 'studentId' attribute of string type. It must be the id of an existing student within the university database. On the other hand, companyStaff need to maintain a 'companyName' attribute of string type.

Attribute	Type	Length	Constraint
id	string	64	Must be a unique identifier, which can be generated using various distributed global ID generation strategies, such as UUID or Snowflake.
name	string	up to 64 max	
password	string	up to 64 max	Must consist of a combination of uppercase and lowercase letters, numbers, and special characters (,.-_), with a minimum length of 8 characters.
email	string	up to 256 max	Must conform to the common email format, such as username@domain
expireTime	date		Must be greater than the current time to ensure the account remains active
isBanned	boolean		Defaults to false
banDuration	int		Cannot be a negative value. It is associated with the 'isBanned' property. When 'banDuration' is 0, 'isBanned' is false; otherwise, 'isBanned' is true

Table 6: Detailed property of User

For each message, table 7 presents specific data constraints:

Attribute	Type	Length	Constraint
id	string	64	Must be a unique identifier, which can be generated using various distributed global ID generation strategies, such as UUID or Snowflake.
senderId	string	64	Must be the id of an existing user within the system
receiverId	string	64	Must be the id of an existing user within the system and cannot be the same as senderId
content	string	up to 1024 max	

Table 7: Detailed property of Message

For each category, table 8 presents specific data constraints:

Attribute	Type	Length	Constraint
id	string	64	Must be a unique identifier, which can be generated using various distributed global ID generation strategies, such as UUID or Snowflake.
name	string	up to 64 max	
parentCategoryId	string	64	Can be empty or the ID of an existing category with the system

Table 8: Detailed property of Category

For each post, table 9 presents specific data constraints:

Attribute	Type	Length	Constraint
id	string	64	Must be a unique identifier, which can be generated using various distributed global ID generation strategies, such as UUID or Snowflake.
categoryId	string	64	Must be the id of an existing category within the system
title	string	up to 512 max	
content	string	up to 1024 max	
creatorId	string	64	Must be the id of an existing user within the system
createTime	date		
likes	int		
isAnonymous	boolean		Defaults to false

Table 9: Detailed property of Post

For each comment, table 10 presents specific data constraints:

Attribute	Type	Length	Constraint
id	string	64	Must be a unique identifier, which can be generated using various distributed global ID generation strategies, such as UUID or Snowflake.
postId	string	64	Must be the id of an existing post within the system
parentCommentId	string	64	Can be empty or the ID of an existing comment with the system
content	string	up to 1024 max	
isAnonymous	boolean		Defaults to false

Table 10: Detailed property of Comment

4 Cross-cutting

This section aims to explicitly link various parts, such as stakeholders, functional requirements, and non-functional requirements, to ensure they are closely connected and share a consistent target, which is to ultimately achieve the business goals. A traceability matrix is included in this section to ensure the consistency and coherence of the requirements. Additionally, this section also provides rationale and details describing how quality requirements contribute to the business goals.

4.1 Traceability

This subsection describes the interdependencies between the artefacts generated throughout the system creation process, facilitating the analysis of potential impacts of future modifications with ease. Figure 5 illustrates pre-requirements traceability, highlighting a strong correlation between the system's business goals and various types of requirements. Simultaneously, it also demonstrates post-requirements traceability, indicating dependencies and associations among different requirements.

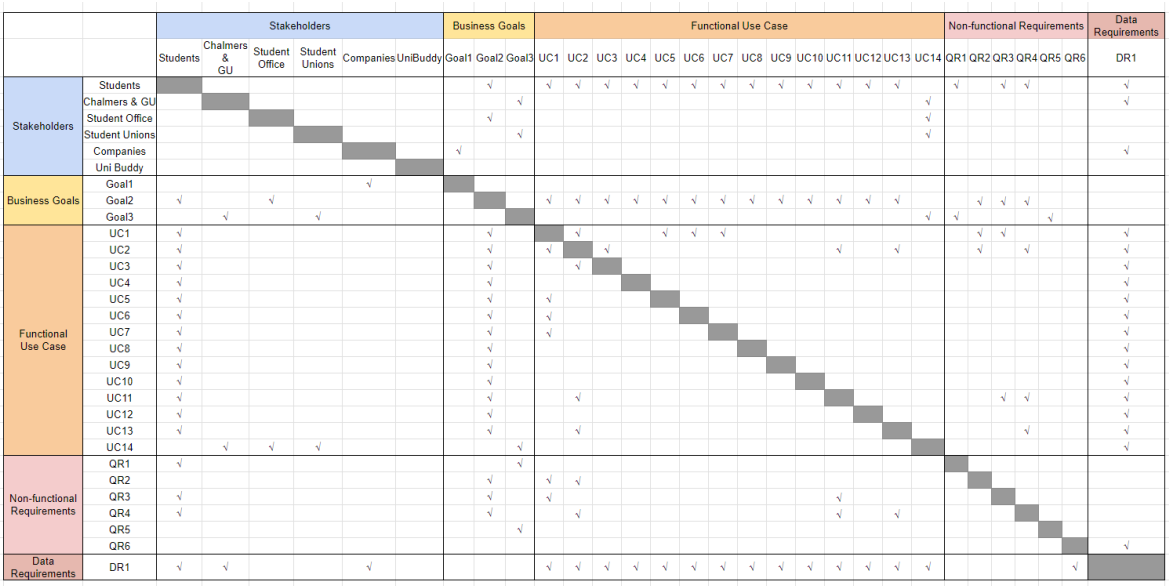

Figure 5: Traceability Matrix displays all the dependencies and relationships between the artefacts identified for the platform.

4.2 Rationale and appropriate detail

QR1: The platform shall be able to host 40,000 user accounts
Rationale:

- Issue: How many users are going to use the platform?
- Position 1: 40,000
- Position 2: 25,000
- Argument: We do not want slow servers
- Assumption: Slow servers will cause users to leave the platform
- Decision 1: 40,000
- Decision 2: It is mandatory and non-negotiable

QR2: The platform shall load new pages and content within 4 seconds.
Rationale:

- Issue: How many seconds can the users wait before they will swap to a new app?
- Position 1: 6 seconds
- Position 2: 4 seconds
- Argument: We don't want users to be bored
- Assumption: Bored users will swap the app
- Decision 1: 4 seconds
- Decision 2: it is negotiable

Requirement: 95 of 100 users shall find relevant categories within 2 minutes after signing up to the platform
QR3:

- Issue: How long does it take users to find relevant pages before feeling lost?
- Position 1: 3 minutes
- Position 2: 2 minutes
- Argument: We don't want users to feel lost
- Assumption: Users who feel lost might want to not use the app again
- Decision 1: 2 minutes
- Decision 2: Negotiable

QR4: 98 of 100 users shall be able to write a post within 1 minute after entering the platform
Rationale:

- Issue: How long does it take users to post before losing their interest?
- Position 1: 1.5 minutes
- Position 2: 1 minute
- Argument: We don't want users to lose interest
- Assumption: No interaction between users will cause the app to go out of business
- Decision 1: 1 minute
- Decision 2: Non-negotiable

QR5: The platform shall be available as an app and on the web
Rationale:

- Issue: Do users need a variety of platforms to use?
- Position 1: Yes
- Position 2: No
- Argument: We don't want an inaccessible app
- Assumption: An inaccessible app causes less interaction between users
- Decision 1: Yes
- Decision 2: Optional and non-negotiable

QR6: The system must adhere to GDPR data protection requirements
Rationale:

- Issue: Do users need privacy about their data?
- Position 1: Yes
- Position 2: No
- Argument: We don't want user data shared with 3rd parties
- Assumption: Companies that do not care about privacy are less trustworthy and go out of business
- Decision 1: Yes
- Decision 2: Mandatory and non-negotiable

5 Conclusion

This document lays the foundation for the *CampusHub* platform, which we hope one day might become a reality, thanks to this project. The next step in this project would be the development process, where these guidelines would be translated into actionable tasks. These requirements that we have set up may also evolve as the development progresses and as the platform changes. It is our hope that we have laid out the groundwork for a sturdy and easy way forward for those who will take over after us, to make their journey as straightforward as possible.

6 References

References

[1] Chalmers in numbers (2022), “Chalmers.se.” [Online]. Available: https://www.chalmers.se/en/

[2] Facts & Figures, “University of gothenburg.” [Online]. Available: https://www.gu.se/en/about-the-university/facts-and-figures

[3] Data protection, “Imy.se.” [Online]. Available: https://www.imy.se/en/organisations/data-protection/

[4] A. Mavin, “Ears.” [Online]. Available: https://alistairmavin.com/ears/