Software Requirements Specification

for the Student Representative Database

Andras Balla

Olga Christensen

Pär Eriksson

Fredrik Olsson

1 Table of content

2 Introduction

3 Purpose

This section introduces the requirements specification document for the *"Student representative database"* system. It provides the purpose and scope of the system. Any definitions and references are listed in this section as well as an overview of the remaining requirements specification document.

3.1 Scope

The scope of this project is to provide a basic prototype of the *"Student representative database"* system which shows the functionality, basic workflow and the advantages that the new system would provide.

The system needs to support the administrator in the daily workflow which includes:

- Keeping content up to date,
- Keeping track of vacant student representative positions in the system,
- Deliver information about different councils,
- Keep the registered information secure,
- The system needs to be able to generate lists of vacant positions and lists of applicants for the election meeting.

3.2 Glossary

Term	Meaning
Linnéstudenterna	the Student Union representing students at LNU
Student	LNU student or PhD (often also a politician) that at some point has shown interest to be a student representative and has an account in the registry.
Applicant Student	Applied for a position in the council, but has not been elected yet
Student Representative	Linnéstudenterna have elected this person to be the official student representative for a specific council
Unelected Student Representative	A student/PhD that is sitting on a council without being officially elected but where Linnéstudenterna has approved and the representative will be elected at next Linnéstudenterna board meeting.
Shadow Student Representative	A student/PhD that is sitting on a counsil without being officially elected and Linnéstudenterna don't have any intention of electing the the student/PhD, this is often due to a misunderstanding on how the SR elections are conducted, often there are misunderstandings among both the counsil staff and the shadow representative.
Staff	LNU employees managing the work of a council. Can have several roles: convener (*sammankallande*), chairman (*ordförande*), secretary. Convener or secretary can also be the council's contact person.
Administrator	Employee of Linnéstudenterna that manages the registry.
Guest / Any	Website user with limited "read-only" permission.
Elections	Elections during the Linnéstudenterna board's monthly meetings. (*Fyllnadsval*)
Admin panel	Website for administrators and council members to manage the registry, requires a login with different access permissions.
Website	Website listing all existing councils and their description, including an application form for becoming a student representative. Doesn't require a login.

4 Environment and system models

For this project we have no real constraints regarding environment or system requirements, since the system is being developed from scratch and the client's main goal is to have a well designed idea that can be developed later in case client's superiors will approve it. So our client has left us a lot of freedom for our prototype when it comes to system environments.

Our task is to design a system that will simplify the current workflow and divide the administrative load among different councils' staff members. We shall in the design phase give our suggestion to the client about what environment the system would benefit from being run in.

5 Scenarios

5.1 A student wants to get in touch with a student representative for a specific council

1. Student opens the website and finds the page with all existing councils
2. Student selects a faculty to see only the councils of this faculty and clicks on the name of a council. A page with this council's profile opens. Here we can see council's description, who is *sammankallande, ordförande*, contact person, the names of student representatives with their email addresses and the amount of open SR positions. (including start and end date of SR term, e.g. 01.2017 - 06.2017)
3. Student uses his/her preferred email client to write an email to the representative.

5.2 A student applies to become a student representative for the first time

1. Student finds a council of interest on the website (3.1). In the bottom of the page there's a link to the application form. Student clicks on the link to open the form, fills it in and submits. The student must agree to create an account in the registry. A new applicant account is created automatically and confirmation email with login info is sent to the student. If the student is already registered (his/her email is in DB), a notification is shown that student must login instead (3.3).
2. Administrator gets a notification via email about it
3. Administrator can view and update the profile

5.3 A student applies to become a student representative not for the first time

1. Student finds a council of interest on the website (3.1). In the bottom of the page there's a link to **login.** Student can also log in directly without looking up a specific council.
2. In the admin panel student can choose to add or remove councils of interest.

5.4 New council

1. LNU decides that a new council at the faculty *X* needs to be created. They assign tasks and staff for the council, as well as number of student representatives, and send info to Linnestudenterna so their administrator would create an account for the new council, or LNU secretary logs into the system and creates the profile.
2. If some of the assigned staff members don't have an account yet in the registry, such an account has to be created first.
3. Now the council is available for viewing (3.1)

5.5 Election

1. Linnéstudenterna at their monthly meetings elect student representatives for different councils. Administrator generates lists of applicants from the admin panel and hands them to linnestudenterna management.
2. During the election members of linnestudenterna elect new student representatives and the management hands the results to admin to enter them into the registry
3. in case some staff members or student representatives do not have an account in the system yet, such accounts must be created first.
4. When staff member or student representative are registered as "elected", they get a "welcome email", and a copy is sent to sammankallande

5.6 New staff member

1. Administrator creates new staff profile and assigns councils and positions they have in respective council
2. Email notification is sent to the new user with login info

5.7 Council staff needs to update some info or upload documentation

1. To be able to edit information of a council, the staff member should be logged in. He/she can edit the info of the councils belonging to their faculty.
2. Council staff should be able to edit description of the council, contact information of staff/representatives, add/remove staff/representatives, upload pdf files (e.g. årsraport)
3. Once the changes are saved other users can see the "last updated" date and the name the council admin on the profile

5.8 Admin creates system users

1. Admin wants to create a user account for LNU secretary and opens a "create admin" form, fills it in, selects which permissions should the user have for council/SR/organisation profiles
2. Once the profile is saved, secretary gets notification

6 Requirements

6.1 Functional requirements

Priority scale:

- Essential (the product is not acceptable unless these requirements are satisfied),
- Conditional (would enhance the product, but the product is acceptable if absent),
- Optional (functions that may or may not be worthwhile)

The client has expressed that clarity in all information is always important. The clearer the information is, and is presented, the less confusion and less errors will happen.

ID: FR-01	Name: Vacant positions	Priority: Essential
Description	The system is required to keep track of and show vacant positions in the councils under ongoing term and for the next term. The system shall clearly show which councils needs what type of representatives, as well as how many.	
Rationale	Linnéstudenterna needs to keep track of vacant positions so that they can fill them during elections. The students need to be able to acquire this information without having to contact Linnéstudenterna.	

ID: FR-02	Name: Handle applications	Priority: Essential
Description	The system is required to be able to accept applications through the web client, save applications, generate a list of applicants and send out confirmation mail to the candidates where they have to confirm their interest before the annual meeting.	
Rationale	Linnéstudenterna wants to have an easy way to keep track of applicants with central database where they can store all the applicants from all sources and then generate lists for elections.	

ID: FR-03	Name: Authentication	Priority: Essential
Description	The system is required to authenticate users and make sure that every user has to proper privileges inside the system. The Admin has full access to the whole system, A staff member can update all fields within its own Faculty, a representative can edit their own information and lastly a guest can only view the information.	
Rationale	Given that the system handles sensitive information it is required to have a authentication system to protect the information.	

ID: FR-04	Name: Student/Phd representative profile	Priority: Essential
Description	An authenticated user shall be able to Create and Update student profiles. The system shall create a representative profile when a user applies for a position in a council. The admin shall be able to manually create, update and remove representative profile if need be. A representative shall be able to Edit their own profile. Any other user shall be able to view any and all Staff Profiles.	
Rationale	This project aims to streamline the workflow therefore the responsibilities should be spread out across multiple users.	

ID: FR-05	Name: Staff Profile	Priority: Essential
Description	A System admin must be able to Create and Update and Remove staff profile. The Staff member must be able to update their own profile information and update the information in their faculty. Any other user must be able to view any and all Staff Profiles.	
Rationale	In order to decrease the workload of the System admin. Staff profiles will be able to edit inside their faculty thus decreasing the daily maintenance that the admin must do.	

ID: FR-06	Name: Council Profile	Priority: Essential
Description	A System admin must be able to Create and Remove Council. Any other user must be able to view any Council Profile. Any admin (System and Staff) must be able to update the council profile.	
Rationale	Decreasing the daily maintenance that the admin must do	

ID: FR-07	Name: Faculty Profile	Priority: Essential
Description	A System admin must be able to Create, Update and Remove Council. Any other user must be able to view any and all Faculty Profiles.	
Rationale	Decreasing the daily maintenance that the admin must do	

ID: FR-08	Name: Number of representatives	Priority: Conditional
Description	An administrator shall be able to set how many representatives are wanted in a council.	
Rationale	Potential student representatives should be able to see how many positions each council has and which have been taken.	

ID: FR-09	**Name:** Create representative candidate	**Priority:** Conditional
Description	When a student has submitted application for SR position (see requirement UI-02), this person should be available for the election at Linnéstudenterna board meetings.	
Rationale	This would enable automation of lists of candidates available for election.	

ID: FR-10	**Name:** Automatic election notice	**Priority:** Conditional
Description	When a SR has been elected, the person shall automatically be informed (by e-mail) about it	
Rationale	To save hours (estimate) of manual work of council and union board staff and to ensure that no shadow representatives exist in the system.	

ID: FR-11	**Name:** Representative reports	**Priority:** Conditional
Description	A student/PhD representative could be able to write a short report periodically about the work they have performed as a representative, and submit it to the system as a form of log.	
Rationale	This would give Linnéstudenterna and council members information about the work the representatives do outside the meetings without the need to manually go through hundreds of emails, and could be saved for meetings.	

ID: FR-12	**Name:** Updates in meetings/appointments	**Priority:** Optional
Description	Changes in meeting appointments should automatically inform the necessary representatives.	
Rationale	It *could* make it easier for conveners to use this system to track meetings.	

ID: FR-13	**Name:** Candidate time period	**Priority:** Conditional
Description	A student should be able to choose what period or periods he/she is interested in being a candidate for a representative position.	
Rationale	This information could make it easier for Linnéstudenterna board to handle elections.	

ID: FR-14	**Name:** Warning on many missions	**Priority:** Conditional
Description	When a candidate has been elected to a specified (by Linnéstudenterna) number of positions/chairs, a warning flag should appear.	
Rationale	This would clearly indicate when a candidate *may* have been appointed an unreasonable number of missions alongside his/her studies.	

ID: FR-15	Name: Student/PhD representative data	Priority: Essential
Description	The data about representatives that shall be stored is first name, last name, personal description, date of birth, city of study/program, email address, phone number, program of education, bank account number, when studies are expected to be finished, member of Linnéstudenterna, languages (SE/EN).	
Rationale	Linnéstudenterna and LNU need access to private information for easy contact and payment processes, but not all information shall be available publicly.	

ID: FR-16	Name: Council data	Priority: Essential
Description	The data about councils that shall be stored is name of council, faculty, city/location, website, language, contact person/staff members, representatives, presentation information (including number of meetings per period).	
Rationale	Students and other people need to be able to have easy access to information about councils, what they do and who they are.	

ID: FR-17	Name: Staff member data	Priority: Essential
Description	Data about staff members stored shall be first name, last name, email address, phone number, position of employment, link to staff page, faculty.	
Rationale	The council's presentation "pages" needs to include information about the contact person.	

ID: FR-18	Name: Election clarity	Priority: Essential
Description	The system must clearly show what students and PhD students are *actually* officially elected as representatives.	
Rationale	To decrease the risk of creating shadow representatives.	

ID: FR-19	Name: Confirm representative interest	Priority: Essential
Description	Before any election, applicants needs to confirm that they are still interested in the position.	
Rationale	To make sure that all elected representatives will be active in their mission.	

ID: FR-20	Name: Election confirmation email	Priority: Conditional
Description	When a representative has been elected, a confirmation mail shall be sent to the student, and a copy shall be sent to the convener.	
Rationale	To make it clear for the student and convener that the student have been officially elected as a representative.	

ID: FR-21	**Name:** Show representatives period interest	**Priority:** Optional
Description	It should be clear if a student has expressed interest in being a representative for more than one term.	
Rationale	This could help Linnéstudenterna to prepare their election processes.	

ID: FR-22	**Name:** Applying for multiple periods	**Priority:** Optional
Description	A student/PhD or representative could be able to apply for positions in future periods.	
Rationale	Will make it clear for Linnéstudenterna and the council staff that students wants to keep working with them.	

ID: FR-23	**Name:** Council suggestions	**Priority:** Optional
Description	When filling out the application form to become a representative, there could be council suggestions based on the program the student currently attends.	
Rationale	This could help the student to find relevant councils if he/she does not know what they want to work with.	

ID: FR-24	**Name:** SR term	**Priority:** Optional
Description	The system should show the term for each SR (format: from - to date).	
Rationale	To bring clarity regarding when the position is taken or open	

ID: FR-25	**Name:** Setting CRUD permissions	**Priority:** Optional
Description	Linnéstudenterna shall decide what permissions each type of user has.	
Rationale	To ensure that only the appropriate system users has the ability to change the information in the system.	

ID: FR-26	**Name:** Permissions to update council information	**Priority:** Conditional
Description	Council staff shall be able to update all councils within their own faculty.	
Rationale	To ease the administrative load.	

ID: FR-27	**Name:** SR previous engagement	**Priority:** Optional
Description	Administrators and council staff should be able to see if a student has had a representative position previously.	
Rationale	To make the election process more effective, since candidates with experience are most welcome.	

ID: FR-28	**Name:** Meeting participants	**Priority:** Conditional
Description	The council staff should be able to log the student representatives meeting attendance.	
Rationale	To make it easy to see how active a SR is, and to help with payment.	

6.2 User interface requirements

We will further develop the user interface as we enter the design phase, as we have mainly focused on the domain and functional requirements in this part of the development process.

ID: UI-01	**Name:** Web based client	**Priority:** Essential
Description	The client shall be a web based application with responsive design for easy access, with forms for data submissions.	
Rationale	A web based system allows for (for most users) a familiar interface and can make it easier to integrate into other already existing systems.	

ID: UI-02	**Name:** Application form	**Priority:** Essential
Description	A student shall through a form be able to submit an application to become a representative in a council.	
Rationale	It should be easy for students to show interest in becoming active in Linnéstudenterna, without any need to first acquire all contact information manually with the council staff through email conversations.	

ID: UI-03	**Name:** Admin Panel	**Priority:** Essential
Description	An authenticated user has access to CRUD functionality across the website based on their permissions.	
Rationale	Authenticated users shall have access to different functionality to decrease the workload of the Linnéstudenterna admin in the daily work flow.	

6.3 Security/privacy requirements

As the client has only expressed an interest in a prototype for demonstration to his employers and colleagues, so not much can be said about this topic. But as the system shall handle social security number and potentially other personal information such as bank account numbers, care must be taken as to where this information is stored and how the information is handled. The final product must take personuppgiftslagen (PuL) into consideration, or the General Data Protection Regulation (GDPR) after May 25th, 2018.

The information must always be encrypted when sent over the internet, and some information must not be freely accessible in the system without some form of authentication.

ID: SEC-01	Name: SSL encryption	Priority: Conditional
Description	The web based application should be SSL secured.	
Rationale	Transmitting sensitive information through form submission and login has to be encrypted to secure the information	

ID: SEC-02	Name: Authentication	Priority: Essential
Description	Only authenticated users should be able to make changes in the system.	
Rationale	We have to make sure that registry is updated only by persons with specific permissions.	

7 Use cases
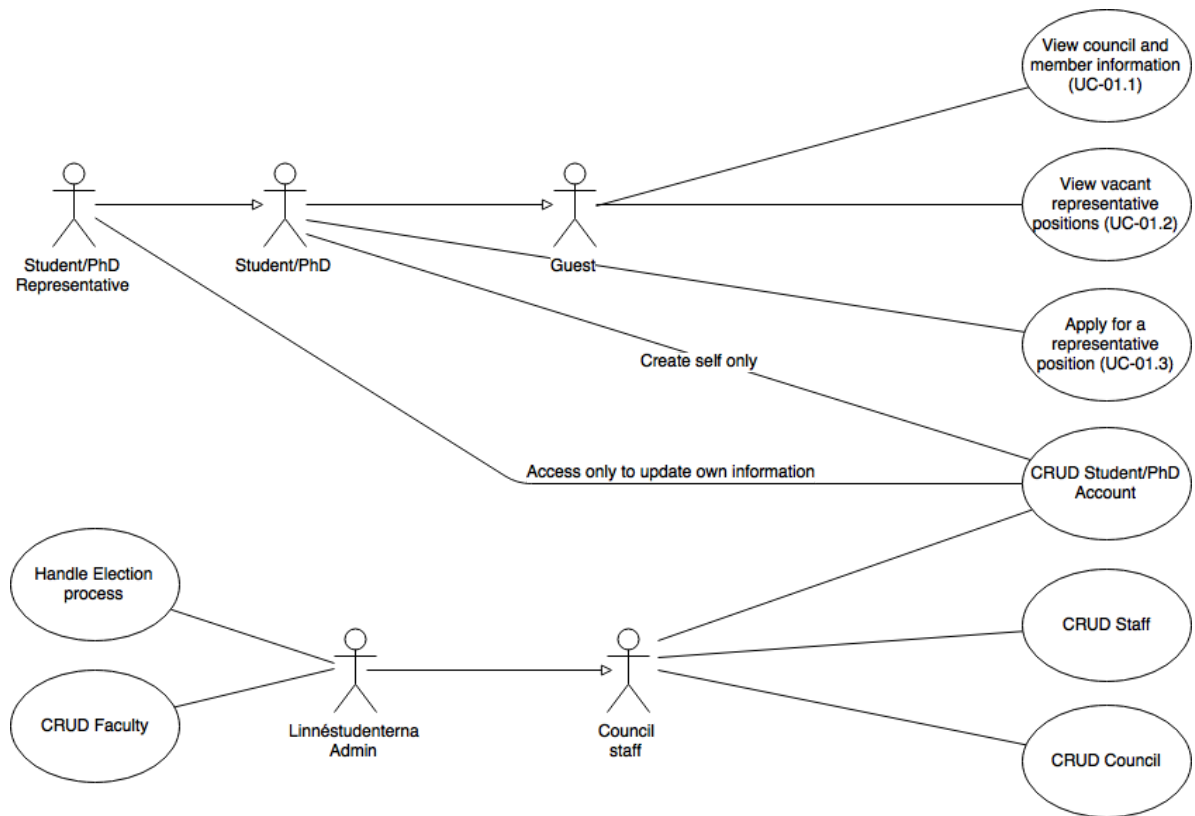
7.1 Viewing website as a guest
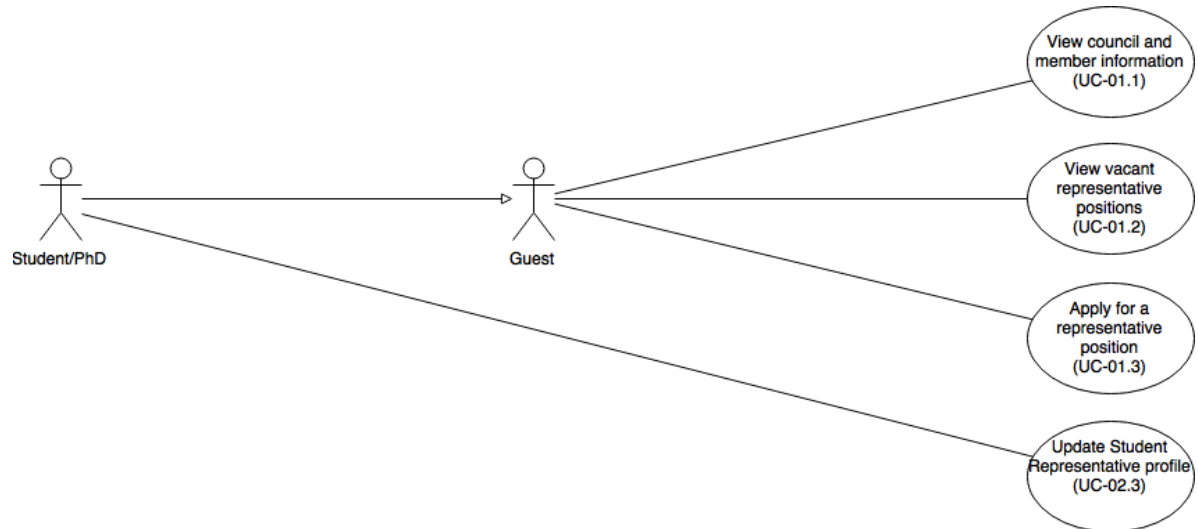
7.1.1 View council and member information (UC-01.1)

Actors: Guest, Student/PhD

Goals: Finding information about different councils and their members (staff, SR)

Preconditions: website is open in the web browser

Success scenario:

Actor actions	System responses
1. Chooses to view all councils	2. Presents a list of councils, divided under faculties
3. Chooses a specific faculty	4. Presents view of that faculty with all councils listed.
5. Chooses a specific council	6. Presents view of that council with its description, SR and staff.

Alternative 1:

3. Chooses to use text search	4. Presents view of matching councils

Postconditions: Actor can view council information with contact information for its staff and SR.

7.1.2 View vacant representative positions (UC-01.2)

Actors: Guest, student/PhD, administrator

Goal: View available vacant student representative (SR) positions.

Related Use Cases: View council and member information (UC-01.1)

Preconditions: website is open in the web browser

Success Scenario:

Actor actions	System responses
1. Chooses to view only councils with vacant positions for a specific period (Student and/or PhD positions)	2. Presents a list of councils with vacant positions
4. Has an option to select a specific council to view more detailed information. (UC-01.1 - step 5)	

7.1.3 Apply for a representative position (UC-01.3)

Actors: Student/PhD

Goal: A student or PhD creates an application to become a SR for a council

Preconditions: User has found a council of interest;

Success Scenario:

Actor actions	System responses
1. Chooses to apply to become a SR for specific council	2. Presents a form with pre-selected Faculty+Council fields
3a. Fills out the form with personal information	
3b. Confirms s/he is not a robot	
3c. Submits the form	4. Success message is shown

Alternative 1: Sending an open application.

1a. Navigates to applicant form directly (outside of council profile)	2. -
1b. Selects Faculty+Council *(optional)*	

Alternative 2: One or more required fields were left empty

3a. Enters incomplete information	4. Error message is presented
5. Gos back to step 3a in success scenario	

Postconditions:

- System creates new student profile in registry and saves personal information
- Respective relation applicant-council is created (if specified)
- Email notification is sent to administrator (Student Union) and the actor with login information

Exception: this email already exists in the system

3a. Enters email that already exists in registry	4. Error message is presented with suggestion to login

7.2 CRUD Student profile
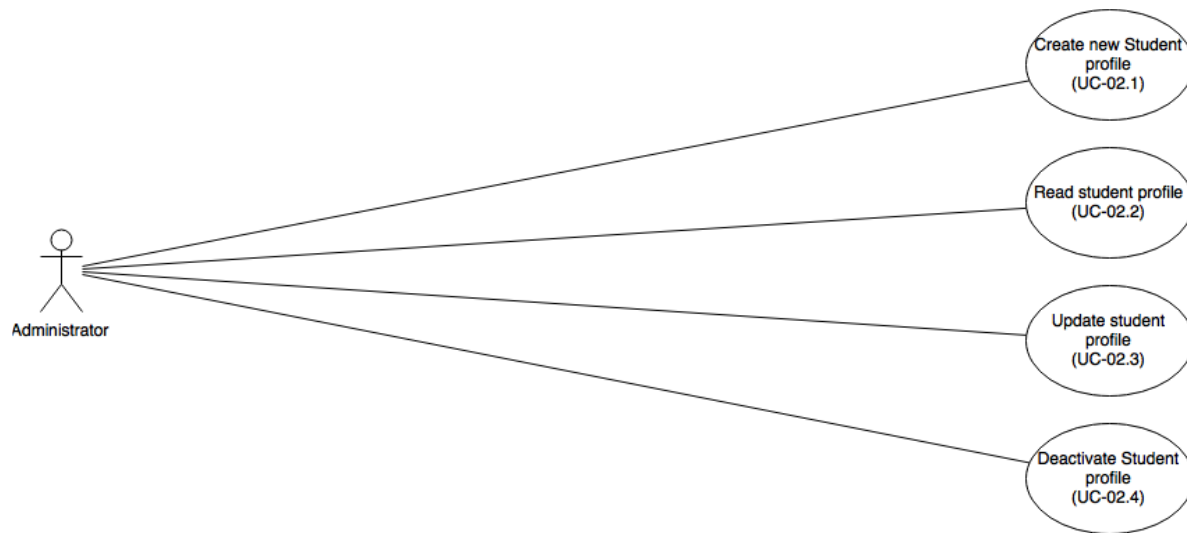
7.2.1 Create new Student profile(UC-02.1)

Actor: Administrator, council staff

Goal: Actor creates a new student profile

Preconditions:

-        User is logged in as admin / council staff
-        Admin can choose among all faculties/councils in step 6
-        Council admin can choose only the councils of her/his faculty
-        Admin panel is shown

Success Scenario:

Actor actions	System responses
1. Chooses to create a new student profile	2. Empty form is presented
3a. Fills in all the required fields	
3b. Chooses to add a council of interest for this SR (*optional*)	4. A form with correlated drop-down lists for faculty+council is presented
5a. Chooses the faculty and council	
5b. Chooses to add this council of interest	6. A form with correlated drop-down lists for faculty+council is presented
7a. Chooses the faculty and council	
7b. Chooses to add this council of interest	8a. Form is closed
	8b. The record is visible in the profile
9a. Repeats the steps 3b-7 0-X times	

9b. chooses to save new record	10. admin panel with success message is shown

Alternative 1: one or more required fields were left empty

3a. enters incomplete information	4. error message is presented
5. gos back to step 3a in success scenario	

Postconditions:

- new student record is saved in registry
- if council(s) of interest were added - corresponding relations are created
- email notification is sent to the student with login information

7.2.2 View Student profile (UC-02.2)

Actor: Administrator, council staff

Goal: Actor views all available information in student account.

Preconditions:

- Actor logged in
- Admin panel is shown

Success Scenario:

Actor actions	System responses
1. Chooses to view student accounts	2. Presents a list of all accounts with filter options for inactive members, faculty and council and search on all account fields
3. Selects account to view.	4. Presents view of account

Postconditions:

- Actor can view student account information.

7.2.3 Update Student profile (UC-02.3)

Actor: Administrator, council staff, student/PhD

Goal: Actor updates information on a student/PhD account.

Related Use Cases: View Student Representative profile (UC-02.2)

Preconditions:

- Actor logged in
- Student account is shown (UC-02.2)

Success Scenario:

Actor actions	System responses
1. Chooses to edit a profile	2. Presents a form

3a. Enters updated information in to profile. 3b. Confirms submission	4. success message is shown

Alternative 1: Some required fields left empty

3a. Enters incomplete information 5. Goes back to step 3a in success scenario	4. Presents error message

Postconditions:

-       Changes made to account profile are saved to DB

7.2.4 Deactivate Student profile (UC-02.4)

Actor: Administrator or council staff

Goal: Student account is set as inactive and is removed from all currently held positions.

Related Use Cases: View Student Representative profile (UC-02.2)

Preconditions:

-       Actor logged in
-       Student account is shown (UC-02.2)

Success Scenario:

Actor actions	*System responses*
1a. Chooses to deactivate the account 1b. Confirms submission	2. Success message is shown

Postconditions:

-       Student profile status is set to inactive, if student/PhD currently holds representative positions, the student is removed from SR lists. No data is deleted.

Exception: Actor cancels deactivation

1b. Cancels submission	2 Presents an appropriate message

7.3 CRUD Staff profile
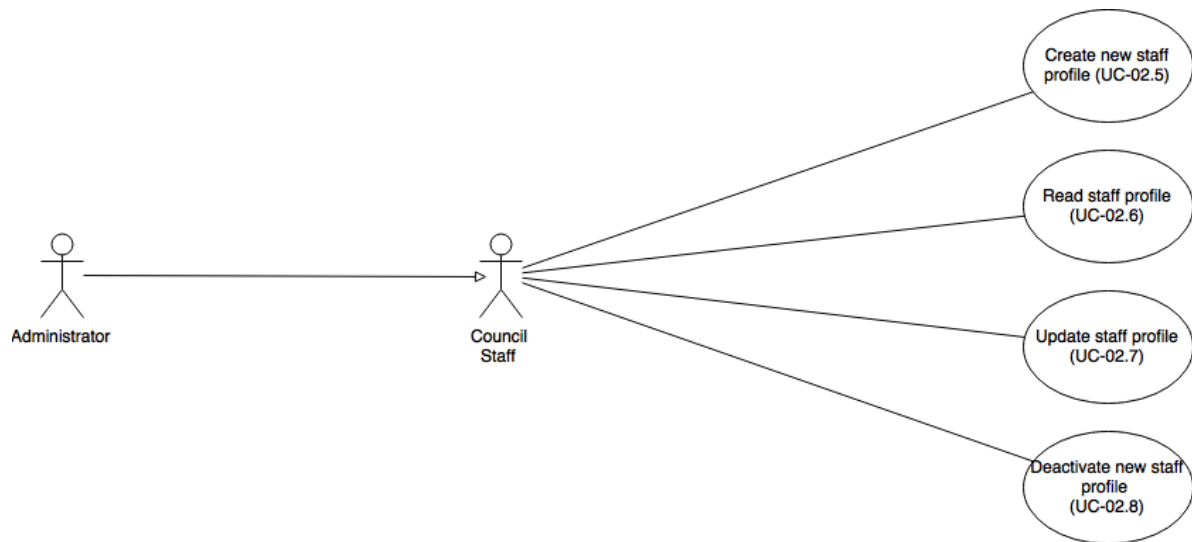
7.3.1 Create new staff profile (UC-02.5)

Actor: Administrator or council staff

Goal: Create new staff profile

Related use cases: None

Preconditions

-      User is logged in as admin / council staff
-      Admin can choose among all faculties/councils in step 6
-      Council staff can choose only the councils of her/his faculty
-      Admin panel is shown

Success Scenario:

Actor actions	System responses
1. Chooses to create a new Staff profile	2. Empty form is presented
3a. Fills in all the required fields	
3b. Chooses to assign a council (*optional*)	4. A form with correlated drop-down lists for faculty+council is presented
5a. Chooses the faculty and council	
5b. Checks all relevant positions, i.e. sammankallande, ordförande, sekreterare, contact	
5c. Chooses to save information	
	6a. The form is closed
7. Repeats the steps 3b-5 0-X times	6b. The record is added/shown
8. Chooses to save new record	

Actor actions	System responses
	9a. Admin panel with success message is shown
	9b. Mail with address to first login is sent to staff person

Postconditions

- New Staff record is saved in DB;
- If council(s) was added - corresponding relations are created;
- Login is created (with auto-generated password)

7.3.2 View staff profile (UC-02.6)

Actor: Administrator, council staff

Goal: Actor views all available information in a staff member account

Related Use Cases: UC-02.1

Preconditions:

- Actor logged in
- Admin panel is shown

Success Scenario:

Actor actions	System responses
1. Chooses to view staff accounts	2. Presents a list of all active members and filtering options for inactive staff, faculty and council, and text search.
3. Selects staff member to view.	4. Presents view of account

Postconditions:

- Actor can view all staff account information.

7.3.3 Update staff profile (UC-02.7)

Actor: Administrator, council staff

Goal: Actor updates information on a staff account.

Related Use Cases: View staff profile (UC-02.6)

Preconditions:

- Actor logged in
- Staff profile is shown

Success Scenario:

Actor actions	System responses
1. Chooses to edit the profile	2. Presents the form
3a. Inputs changes to profile.	

3b. Confirms submission	4. Presents success message

Alternative 1: Some required fields were left empty

	4. Presents error message
5. Goes back to step 3a in success scenario	

Postconditions:

- Changes made to account profile are saved to DB

7.3.4 Deactivate staff profile (UC-02.8)

Actor: Administrator, council staff

Goal: Staff account are set as inactive and are removed from all currently held positions.

Related Use Cases: View staff profile UC-02.6

Preconditions:

- Actor logged in
- Staff profile is shown

Success Scenario:

Actor actions	System responses
1a. Chooses to deactivate the account	
1b. Confirms submission	2. Success message is shown

Postconditions:

- Staff member profile is set to inactive in database, if staff member currently manages some councils - he/she is removed from their staff list. No data is deleted

Exception: Actor cancels deactivation

1b. Cancels submission	2 Presents an appropriate message

7.4 CRUD Council profile
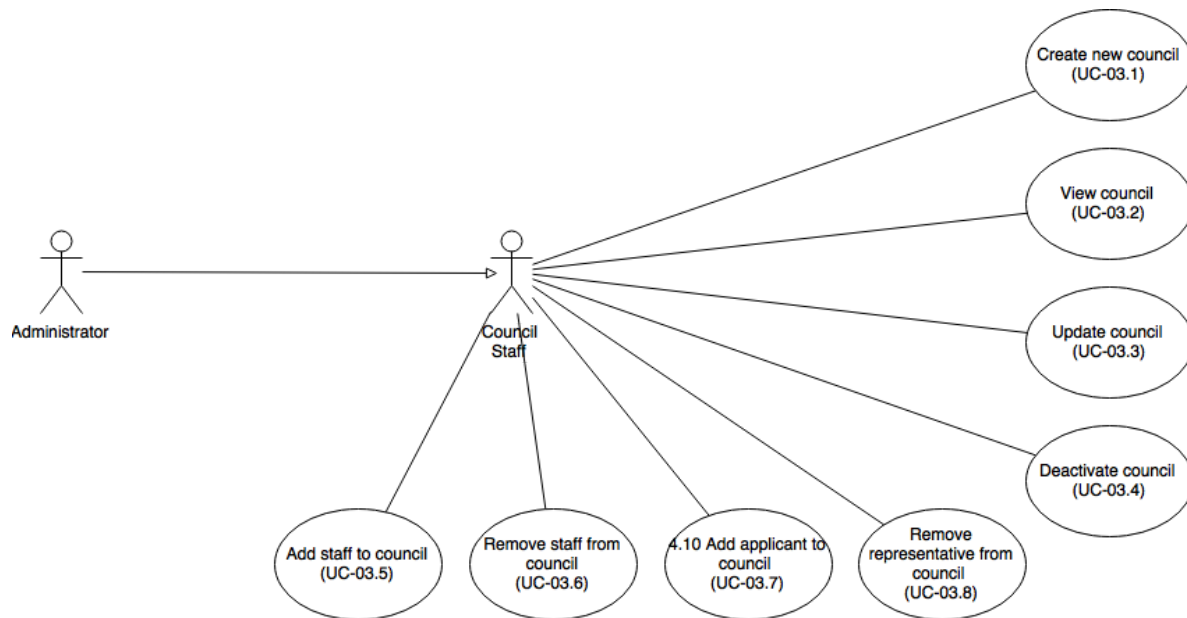
7.4.1 Create new council (UC-03.1)

Actor: Administrator

Goal: Create a new council, with members and information

Related use cases: Add staff to council UC-03.5, Add applicant to council UC-03.7

Preconditions

- User is logged in as admin
- Respective faculty already exists
- Admin panel is shown

Success Scenario

Actor actions	System responses
1. Chooses to create a new council	2. An empty form is presented
3a. Chooses a faculty from drop-down list	
3b. Fills in all the required fields	
3c. Chooses to add minimum one staff person (UC-03.5)	
3d. (*Optional*) Chooses to add applicants (UC-03.7)	
3e. Sets amount of SR positions (students/PhD)	
3f. Chooses to save new record	4. User is redirected to the admin panel and success message is shown

Postconditions

- New council record is saved in DB

7.4.2 View council (UC-03.2)

Actor: Administrator, council staff

Goal : To view a council page containing the its information

Related use cases: Create new council UC-03.1

Preconditions: admin panel is open

Success Scenario:

Actor actions	System responses
1. Chooses to view councils	2. Presents a list of councils with search options by faculty, etc.
3. Selects a council.	4. Present a detailed view of the council.

Postconditions: None

7.4.3 Update council (UC-03.3)

Actor: Administrator, council staff

Goal: Update council information and members

Related use cases: View councilUC-03.2

Preconditions

- User is logged in as admin,
- Respective council already exists;
- Council profile is shown

Success Scenario

Actor actions	System responses
1. Chooses to edit the profile	2. Presents a form with prefilled information about council and members
3a. Edits fields *(optional)*	
3b. Chooses to add new staff person (*optional*)	
3c. Chooses to edit/delete staff (*optional*)	
3d. Chooses to add applicants (*optional*)	
3e. Chooses to add report	
3f. Chooses to save updated record	
	4. User is redirected to the council profile page and success message is shown

Postconditions

-         Council record is updated and saved in DB

7.4.4 Deactivate council (UC-03.4)

Actor: Administrator

Goal: Mark a council as inactive.

Related use cases: View council UC-03.2

Preconditions

-         User is logged in as admin
-         Respective council already exists
-         Council profile is shown

Success Scenario

Actor actions	System responses
1a. Chooses to deactivate the account 1b. Confirms submission	 2. Success message is shown

Postconditions

-         Council record is updated and saved *as inactive* in DB

7.4.5 Add staff to council (UC-03.5)

Actor: Administrator, council staff

Goal : Connect profile of a staff member to a council

Related use cases: View council UC-03.2

Preconditions

-         User is logged in as admin
-         Respective council already exists
-         Respective staff profile already exists
-         Council profile is shown

Success Scenario

Actor actions	System responses
1. Chooses to add staff member 3a. Opens a drop-down list of all staff members with a search field 3b. Fills in first name and 3 first letters of applicant's last name in search field 5a. Selects the name	2. Presents a dialog 4. Shows list of matching records

5b. Checks the role(s) applicable to this profile (ordförande; sammankallande; sekreterare; contact) 5c. User confirms the choice	6. The form is closed; the record is added/shown

Postconditions

-        Respective relation is created in DB

7.4.6 Remove staff from council (UC-03.6)

Actor: Administrator, council staff

Goal: Remove a staff member from a council

Related use cases: View council UC-03.2

Preconditions

-        Actor logged in
-        Respective council already exists
-        Council profile is shown

Success Scenario

Actor actions	System responses
1. Chooses a member to remove 3. Confirms submission	2. Presents confirm deletion view

Postconditions:

-        The DB is updated with the information that there no longer is an active connection between the council and the staff member.

Exceptions: actor cancels the action

3a. Cancels submission	5. Appropriate message is shown.

7.4.7 Add applicant to council (UC-03.7)

Actor: Administrator, council staff

Goal: Add a student representative applicant to council

Related use cases: View council UC 03.2

Preconditions

-        User is logged in as (council) admin
-        Respective student account already exists
-        The profile of respective council is opened

Success Scenario

Actor actions	System responses
1. Chooses to add applicant	2. Presents a dialog
3a. Opens a drop-down list of all student representatives with a search field	
3b. Fills in first name and 3 first letters of applicant's last name in search field	
5a. Selects the name	4. Shows list of matching records
5b. Confirms the choice	
	6a. The form is closed
	6b. The record is added/shown

Postconditions

-        Respective relation is created

7.4.8 Remove representative from council (UC-03.8)

Actor: Administrator

Goal: When a representative period is over, admin removes that representative from the council.

Related use cases: View council UC-03.2

Preconditions

-        User is logged in as admin
-        Respective council already exists
-        Council profile is shown

Success Scenario

Actor actions	System responses
1. Choose a representative to remove	2. Presents confirm deletion view
3. Confirm action	

Postconditions

-        The database is updated and the page is reloaded to show the updated information.

Exceptions

3a. Cancels submission	5. Appropriate message is shown. The representative is not deleted

7.5 CRUD Faculty profile
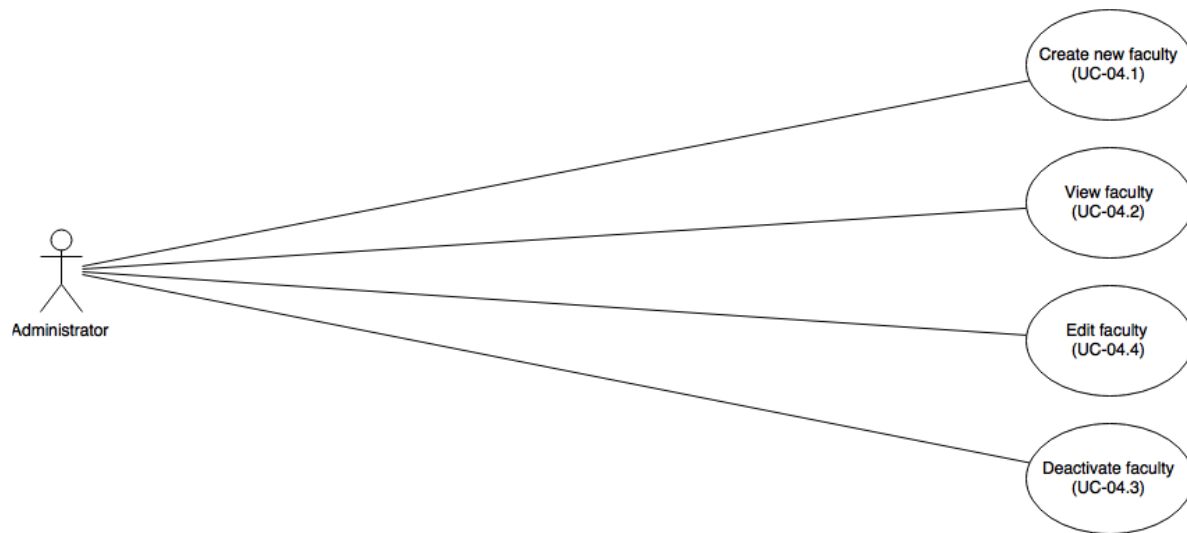
7.5.1 Create new faculty (UC-04.1)

Actor: Administrator

Goal: Create a new faculty.

Preconditions:

- User is logged in as Linnéstudenterna admin
- Admin panel is shown

Success Scenario:

Actor actions	System responses
1. Chooses to view faculties	2. Presents the list of faculties
3. Chooses to create a new faculty	4. Presents an empty form
5a. Fills in the name for the faculty	
5b. Confirms submission	6. Saves faculty to DB redirects to the admin panel presents a success message

Alternative 1: no faculty name / duplicate faculty name

5a. Enters a duplicate name or nothing at all	
5b. Confirms submission	6. Presents error message
7. Goes back to step 5a in success scenario	

Postconditions:

- New faculty record is saved in DB

7.5.2 View faculty(UC-04.2)

Actor: Administrator

Goal: View a faculty.

Preconditions

- User is logged in as Linnéstudenterna admin,
- List of faculties is shown

Success Scenario

Actor actions	System responses
1. Chooses a faculty	2. Presents view of faculty

7.5.3 Deactivate faculty (UC-04.3)

Actor: Administrator

Goal: Deactivate a faculty

Related use cases:

- View Faculty 04.2

Preconditions

- User is logged in as Linnéstudenterna admin
- Faculty view is open
- There should be no active councils in this faculty to be able to delete it

Success Scenario

Actor actions	System responses
1a. Chooses to delete the faculty 1b. Confirms submission	2. Redirects to the admin panel and success message is shown

Postconditions

- Selected faculty is set to inactive in the database.

Exceptions

4a. Chooses to cancel	5. The faculty is not deleted

7.5.4 Edit faculty (UC-04.4)

Actor: Administrator

Goal: Edit faculty's information.

Related use cases

- **Includes:** View Faculty 04.2

Preconditions

- User is logged in as Linnéstudenterna admin,
- Faculty profile is open

Success Scenario

Actor actions	System responses
1. Chooses to edit faculty	2. Presents form
2a. Updates information	
2b. Confirms submission	3. Redirects to the admin panel and display success message.

Postconditions

- The faculty is updated in the database.

Exceptions

2b. Chooses to cancel	3. The faculty is not updated. Appropriate message is shown

7.6 Login

7.6.1 First login to user profile (UC-05.1)

Actor: Student, council staff

Goal: User logs in for the first time into the system and creates password for the account

Related use cases: Create new student profile UC 02.1, Apply for representative position UC 01.3, Create new staff profile UC 02.5

Preconditions:

- When profile is created, a confirmation email with login link is sent to the user (UC 02.1, 01.3, 02.5)

Success Scenario:

Actor actions	System responses
1. Follows the link "Login for the first time"	2. Presents password recovery procedure
2. Enters email	3. Sends email with personalised link to restore the password
4. Follows the link to restore the password	5. Presents view to create new password
6. Enters new password, confirms and submits	7a. Replaces old password with new.
	7b. Login page is shown

Postconditions:

- Student/staff new password saved to DB
- User can login with chosen password and email

Exception A: email doesn't match any in the DB

	3a. Prompts user that email does not exist
4. Goes back to step 2	

Exception B: Invalid password input

	7a. Password doesn't match or doesn't comply with password complexity policy
4. Goes back to step 6	

7.6.2 Login (UC-05.2)

Actors: Student representatives, council staff, administrator

Goal: Actor logs into the system and get correct permissions.

Related use cases: Create new student profile 02.1

Preconditions

- User has an active account.

Success Scenario

Actor actions	System responses
1. Chooses to log in	2. Presents login view
3. Enters email and password and submits	4. Validates email and password, sets the correct permissions.

Postconditions

- Actor is logged in with correct permissions.

Alternative 1: Failure validating actor credentials

3. Enters wrong email / password	4. Present error message
5. Goes back to step 3 in Success Scenario	
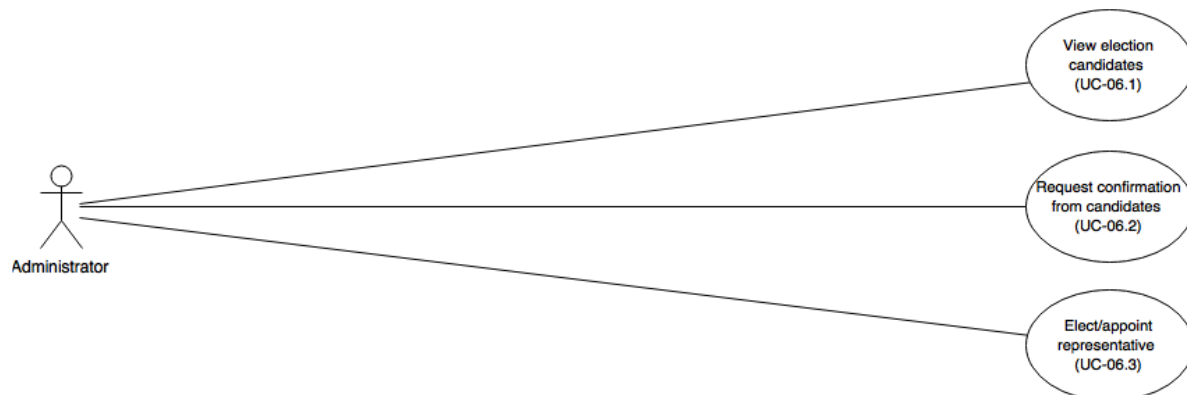
7.7 Elections

7.7.1 View election candidates (UC-06.1)

Actor: Administrator

Goal: Create and list all candidates for elections of student representatives, should include information about how many (and which) positions they already have.

Related use cases: None

Preconditions:

- Actor is logged in

Success Scenario

Actor actions	System responses
1. Requests list of candidates	2a. Compiles a list of all active student/PhD representatives and all students/PhDs who have applied interest to become reps
	3a. Displays list, offers a downloadable PDF
	3b. Shows how many positions each active rep has
	3c. Highlights students that participate in meetings without being elected

Postconditions

- PDF is downloaded

7.7.2 Request confirmation from candidates (UC-06.2)

Actor: Administrator

Goal: Starts an automated process before Linnéstudenterna election meeting to confirm which applicants are still interested in a position they have shown interest in.

Related use cases: View election candidates UC-06.1

Preconditions: None

Success Scenario

Actor actions	System responses
1. Chooses to start confirmation process	2. Collects all applicants information 3. Creates a unique confirmation link for each applicant, adds it to a mail template 4. Sends out the mail to each applicant

Postconditions:

- If applicant has confirmed application - application stays in the council

- If applicant declined application or was inactive in *7 days* - the applicant is not eligible as a representative

7.7.3 Elect/appoint representative (UC-06.3)

Actor: Administrator

Goal: Create a record of an elected student representative for a specific council

Related use cases: View council profile UC 03.2, Add applicant to council UC 02.7

Preconditions

- Actor is logged in
- Respective council already exists
- Applicant profile already exists
- Linnéstudenterna has elected the applicant to be a student representative
- Council profile is shown (UC-03.2)

Success Scenario

Actor actions	System responses
1. Chooses to add a student representative 3. Selects the names of representatives to be in council 4. Confirms selection	2. A form with all applicants is presented (alphabetically ordered) 5a. The form is closed; success message is shown

Postconditions

- Respective relation is created

- Welcome email is sent to all elected SR with a copy to council's convener

Exception 1: Student is not in applicants list

3. Must add a new applicant 4. Proceed with UC-03.7	

7.8 Council Meetings management
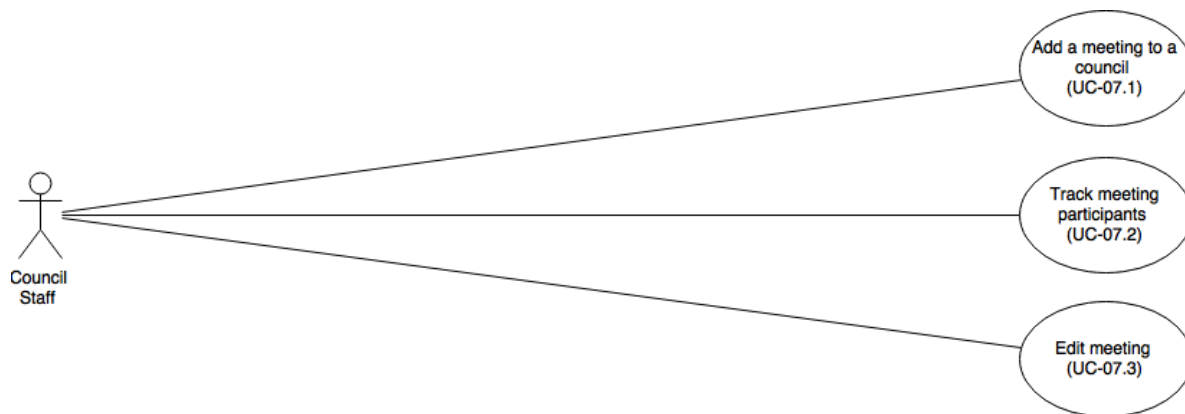
7.8.1 Add a meeting to a council (UC-07.1)

Actor: Council staff, administrator

Goal: Schedule council meetings

Related use cases: Add applicant to council UC-03.7, View council 03.2

Preconditions:

-     User is logged in as admin
-     Council meetings view is shown

Success Scenario

Actor actions	System responses
1. Chooses to add new meeting	2. A form is opened
3. Fills in time, date and place for the meeting.	
4. Confirms submission	5. Success message is shown

Postconditions:

-     Information is saved

-     Sends email notification to all members

7.8.2 Track meeting participants (UC-07.2)

Actor: Council staff, administrator

Goal: Track participants of the meeting that has taken place as well as the length of the meeting

Related use cases: Add a meeting to a council UC 07.1, Create new student profile UC 02.1

Preconditions:

- A meeting has been scheduled (UC-07.1)
- Council meetings view is shown

Success Scenario

Actor actions	System responses
1. Selects specific meeting that needs to be edited 3a. Checks all the SR that attended the meeting 3b. Selects the length of the meeting (Short/Long) 3c. Confirms submission	2. Meeting dialog opens with the names of all SR registered in the council (check-mark list) and option to choose the length of the meeting (drop-down list) 4. Information is saved

Alternative 1:

Actor actions	System responses
3a. Student representative is not in the list - chooses to add "unofficial" representative to the meeting 5. Adds "unofficial" representative to the meeting	4a. Opens a dialog with search option or option to add a new student profile (UC-02.1) 6. E-mail notification is sent to the administrator

Postconditions: None

7.8.3 Edit meeting (UC-07.3)

Actor: Council staff, administrator

Goal: Update meeting information available in the registry

Related use cases: Add a new meeting UC-07.1, Track meeting participants 07.2

Preconditions:

- A meeting has been scheduled (UC-07.1)
- Council meetings view is shown

Success Scenario:

Actor actions	System responses
1. Selects specific meeting that needs to be edited 3a. Edits time / date *(optional)* 3b. Edits the SR that attended the meeting *(optional)* 3c. Selects the length of the meeting (Short/Long) *(optional)* 3d. Confirms submission	2. Meeting dialog opens with the names of all SR registered in the council (check-mark list) and option to choose the length of the meeting (drop-down list) 4a. Information is saved 4b. if 3a -> sends email notification to all members

8 Use case to functional requirement matrix

Requirements → / Use cases ↓	FR-01	FR-02	FR-03	FR-04	FR-05	FR-06	FR-07	FR-08	FR-09	FR-10	FR-11	FR-12	FR-13	FR-14	FR-15	FR-16	FR-17	FR-18	FR-19	FR-20	FR-21	FR-22	FR-23	FR-24	FR-25	FR-26	FR-27	FR-28
UC-01.1				✔																								
UC-01.2	✔																											
UC-01.3		✔							✔				✔	✔														
UC-02.1				✔											✔													
UC-02.2				✔											✔													
UC-02.3				✔											✔													
UC-02.4				✔																								
UC-02.5					✔												✔											
UC-02.6					✔												✔											
UC-02.7					✔												✔											
UC-02.8					✔																							
UC-03.1						✔		✔								✔										✔		
UC-03.2						✔										✔				✔				✔		✔		
UC-03.3						✔										✔										✔		
UC-03.4						✔																				✔		
UC-03.5																										✔		
UC-03.6																										✔		
UC-03.7																										✔		
UC-03.8																										✔		
UC-04.1							✔																					
UC-04.2							✔																					
UC-04.3							✔																					
UC-04.4							✔																					
UC-05.1			✔																									
UC-05.2																												
UC-06.1		✔							✔				✔	✔				✔		✔	✔			✔			✔	
UC-06.2		✔							✔										✔									
UC-06.3										✔				✔				✔										
UC-07.1																												✔
UC-07.2												✔																✔
UC-07.3																												✔