

# **Software requirements specification**

**MIB** FACHSCHAFT MATHEMATIK INFORMATIK  
UNIVERSITÄT BERN

Team 6

2018

Elena Kunitsyna, Elio Fritschi  
Pascal Andre, Roland Widmer

## Content

Introduction.....	3
Use cases .....	4
1. Name: Log into the system. ....	4
2. Name: Log out from the system.....	5
3. Name: Registration on the website .....	5
4. Name: Change password .....	6
5. Name: Change user data .....	7
6. Name: Create job posting .....	8
7. Name: Edit job posting .....	8
8. Name: Delete job posting by User .....	9
9. Name: Correct.....	9
10. Name: Approval Job Postings which is In Review.....	10
11. Name: Approval Job Postings which is already denied.....	11
12. Name: Deny.....	11
13. Name: Delete by admin .....	12
14. Name: Feature.....	12
15. Name: Unfeature .....	12
16. Name: Approval Company which is In Verification .....	13
17. Name: Approval Company which is already unverified.....	14
18. Name: Unverify .....	14
19. Name: Delete Company.....	15
20. Name: Feature Company .....	15
21. Name: Unfeature Company .....	15
22. Name: Companies .....	16
23. Name: About Us .....	16
24. Name: Search .....	16
25. Name: Cancel Search .....	17
26. Name: Filter.....	17
Actor characteristics: .....	18
Functional and non-functional requirements .....	19

# Introduction

## Purpose

The goal of this software is to provide a website where companies can post their job offers and students can find these jobs.

This Software Requirements Specification document serves as a guideline for the project and gives a detailed overview of the functionalities of the software. Its aim is to capture the needs of the company RCP and establish a common understanding of the functional and non-functional requirements of the project.

## Stakeholders

- University students
- External companies
- MIB Fachschaft Mathematik und Informatik der Universität Bern

## Definition

User: company

Guest: Student

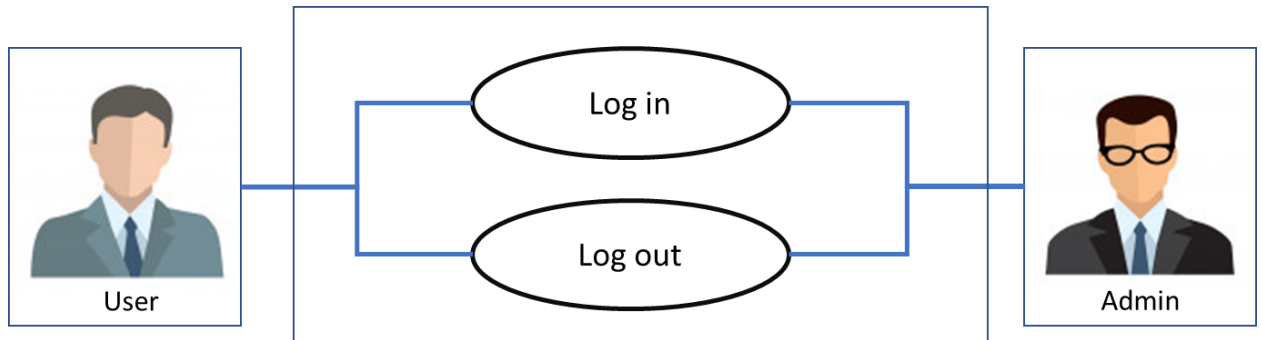
Admin: someone from MIB who's moderating the website.

## System overview

The system should:

- Show all the accepted job postings on the main page
- Show the company page with list of the companies
- Provide an opportunity to CRUD (Create Read Update Delete) operations for job postings
- Be able to open every job postings page with details
- Be able to provide a registration for companies
- Be able to proceed login/logout for companies
- Provide an admin page with additional rights (approve/disapprove new jobs/companies or changes in existing job postings and companies)
- Search and filter the job postings
- Save companies and job postings information in the database
- Be able to make featured/unfeatured Job Postings and Companies

## Use cases



### 1. Name: Log into the system.

**Description:** Company which is already registered visits the website and logs in with his own data.

**Preconditions:** Company has an account on the website, knows his e-mail address and password

Event Flow:

1. User opens a website and presses the button "Sign in"
2. User fills in e-mail address field
3. User fills in password address field
4. User clicks the button "Sign in".
5. System checks whether the credentials are correct
6. User is led to its homepage. The toastr "Sign In successful" appeared.

Alternate scenario  
"Incorrect Password":

1. User opens a website and presses the button "Sign in"
2. User fills in e-mail address field
3. User fills in password address field
4. User clicks the button "Sign in".
5. System checks whether the credentials are correct
6. In case the e-mail address is correct, the password is incorrect, the toastr "Sign In failed – wrong password" appeared.

Alternate scenario 2  
"Incorrect E-mail":

1. User opens a website and presses the button "Sign in"
2. User fills in e-mail address field
3. User fills in password address field
4. User presses the button "Sign in".
5. System checks whether the credentials are correct
6. In case the e-mail address is incorrect - the toastr "Sign In failed – user not found" appeared.

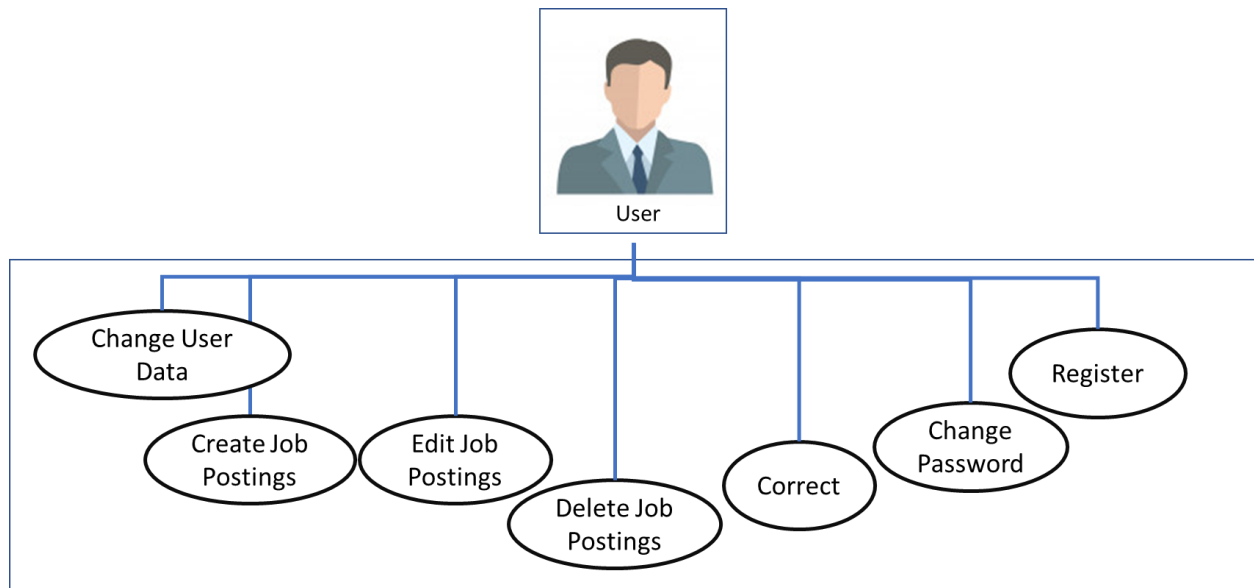
## 2. Name: Log out from the system.

**Description:** User which is already registered and signed in logs out from the website.

**Preconditions:** User has an account on the website, and already logged in

Event Flow:

1. User is on the website
2. User presses the button "Sign out" and is led to the main page. The toastr "You now are signed out" appeared



## 3. Name: Registration on the website .

**Description:** User registers the website (user fills all the data about itself) .

**Preconditions:** User has an internet connection.

Event Flow:

1. User opens a website and presses the button "Register"
2. User fills in all fields (E-mail, password, Confirm password)
3. User choose the user type: company.
4. Company details, Address and Contact Person fields become available
5. User fills in all fields (Company details: Name, Description, Website, Logo; Address: Street/ House No., ZIP code/ place; Contact Person: Name, E-mail, Phone)
6. User clicks the button "Register"
7. All the data are sent to database. The toastr "Registration is successful" appeared
8. User is led to the main page (where he can sign in)

**Postcondition:** Data of the new user is saved in the database

Alternate scenario 1 “Different “Password” and “Confirm Password””	<ol style="list-style-type: none"> <li>1. User opens a website and clicks the button “Register”</li> <li>2. User fills in all fields (E-mail, password, Confirm password)</li> <li>3. User choose the user type: company.</li> <li>4. Company details, Address and Contact Person fields become available</li> <li>5. User fills in all fields (Company details: Name, Description, Website, Logo; Address: Street/ House No., ZIP code, Place; Contact Person: Name, E-mail, Phone)</li> <li>6. User clicks the button “Register”</li> <li>7. In case data in fields “password’ and “confirm password” are not the same the toastr “Registration failed. Invalid Input” appeared. Under the field “confirm password” the message “Password and confirm Password do not match” appeared. User should rewrite information</li> </ol>
Alternate scenario 2 “Student User Type”	<ol style="list-style-type: none"> <li>1. User opens a website and clicks the button “Register”</li> <li>2. User fills in fields (E-mail, password, Confirm password)</li> <li>3. User choose the user type: student</li> <li>4. The message “This option is currently not available.” appeared</li> </ol>
Alternate scenario 3 “Data missing in mandatory fields”	<ol style="list-style-type: none"> <li>1. User opens a website and presses the button “Register”</li> <li>2. User fills in fields (E-mail, password, Confirm password)</li> <li>3. User choose the user type: company.</li> <li>4. Company details, Address and Contact Person fields become available</li> <li>5. User fills in some of the fields (Company details: Name, Description, Website, Logo; Address: Street/ House No., ZIP code/ place; Contact Person: Name, E-mail, Phone)</li> <li>6. User presses the button “Register”</li> <li>7. In case there is no data in any of fields “E-mail”, “Password”, “Confirm Password”, “Name”, “Description”, “Street/ House No.”, “ZIP code”, “Place”, “Contact Person” the toastr “Registration failed. Invalid Input” appeared. User should fill in information</li> </ol>

#### 4. Name: Change password

**Description:** User wants to change his password

**Precondition:** User is logged in.

Event Flow:

1. User put the button “My Account”
2. User chooses “Account Settings”
3. User is led to his account page
4. User fills in fields “Password” and “Confirm Password”
5. User presses the button “Save changes”
6. Data is sent to database and toastr “User update successful” appeared.

**Postcondition:** New User’s password is saved in the database

Alternate Flow  
“Different “Password”  
and “Confirm  
Password””:

1. User put the button “My Account”
2. User chooses “Account Settings”
3. User is led to his account page
4. User fills in fields “Password” and “Confirm Password”
5. User presses the button “Save changes”
6. In case data in fields “password” and “confirm password” are not the same the toastr “User Update failed. Invalid Input” appeared.

## 5. Name: Change user data

**Description:** User wants to change some data about itself

**Precondition:** User has an account on the website is logged in.

Event Flow

1. User put the button “My Account”
2. User chooses “Account Settings”
3. User is led to his account page
4. User changes data in any fields
5. User presses the button “Save changes”
6. Confirm box “Are you sure that you want to save changes?” will appears
7. User clicks “yes” button
8. Data is sent to database and toastr “User update successful” appeared.

**Postcondition:** New User’s data is saved in the database. Job attribute accepted changed to NULL

Alternate Flow “Data  
missing in  
mandatory fields”

1. User put the button “My Account”
2. User chooses “Account Settings”
3. User is led to his account page
4. User changes data in any fields
5. User presses the button “Save changes”
6. In case there is no data in any of fields “E-mail”, “Password”, “Confirm Password”, “Name”, “Description”, “Street/ House No.”, “ZIP code”, “Place”, “Contact Person” the toastr “User update successful Failed” appeared. User should fill in information.

## 6. Name: Create job posting

**Description:** User wants to create a job posting

**Precondition:** User is verified by admin and logged in.

Event Flow

1. User put the button "My Account"
2. User chooses "My Job Postings"
3. User is led to his Job Postings page (../my-account/job-postings)
4. User presses the button with plus
5. User is led to the page of new job posting creation
7. User fills in all the needed fields.
8. User presses the button "Submit"
7. If all the required fields (fields with sign \*) are fulfilled, then data is sent to admin for verification and toastr "Job creation successful. Now in review of our admin" appeared.

**Postcondition:** Data about new Job Posting is saved in the database

Alternate flow  
"Wrong or missing  
data"

1. User put the button "My Account"
2. User chooses "My Job Postings"
3. User is led to his Job Postings page (../my-account/job-postings)
4. User presses the button with plus
5. User is led to the page of new job posting creation
6. User fills in all the needed fields.
7. User presses the button "Submit"
8. If some of the required fields (fields with sign \*) are not fulfilled or written with wrong data type or amount, then the toastr "Job creation failed. Invalid Input" appeared. User should fill in information.

## 7. Name: Edit job posting

**Description:** User wants to edit an existing job posting

**Precondition:** User is verified by admin and logged in, job posting exists.

Event Flow:

1. User put the button "My Account"
2. User chooses "My Job Postings"
3. User is led to his Job Postings page (../my-account/job-postings)
4. User chooses the job posting which he is going to edit and presses the edit button (button with pencil)
5. User is led to the page of job posting edition
6. User fills in/changes all the needed fields.
7. User presses the button "Save changes"
8. Data is sent to admin for verification and toastr "Job updated successfully" appeared.

**Postcondition:** New data about Job Posting is saved in the database. Job attribute accepted changed to NULL



Alternate flow  
“Wrong or missing  
data”

1. User put the button “My Account”
2. User chooses “My Job Postings”
3. User is led to his Job Postings page (../my-account/job-postings)
4. User chooses the job posting which he is going to edit and presses the edit button (button with pencil)
5. User is led to the page of job posting edition
6. User fills in/changes all the needed fields.
7. User presses the button “Save changes”
8. If some of the required fields (fields with sign \*) are not fulfilled or written with wrong data type or amount, then the toastr “Job update failed. Invalid Input” appeared. User should fill in information.

## 8. Name: Delete job posting by User

**Description:** User wants to delete an existing job posting

**Precondition:** User is verified by admin and logged in, job posting exists.

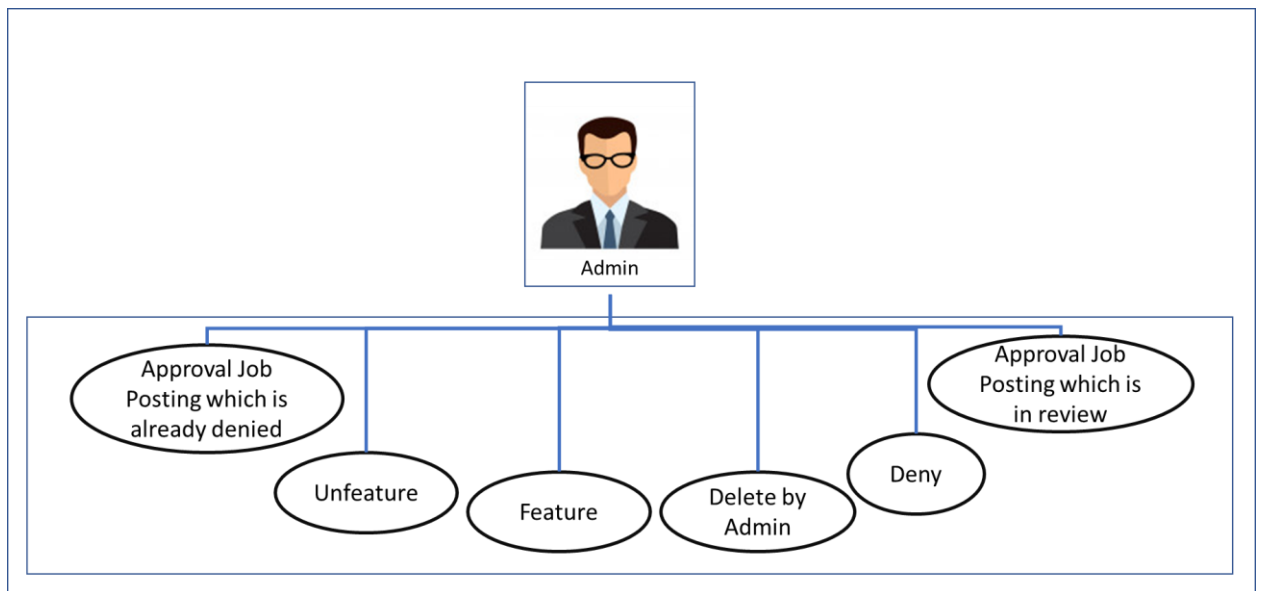
Event Flow	<ol style="list-style-type: none"> <li>1. User put the button “My Account”</li> <li>2. User chooses “My Job Postings”</li> <li>3. User is led to his Job Postings page (../my-account/job-postings)</li> <li>4. User chooses the job posting which he is going to delete and presses the delete button (button with trash can)</li> <li>5. The verification message “You are about to delete your job posting. Are you sure?” is appeared.</li> <li>6. User presses the button “Yes”</li> <li>7. The job posting is deleted</li> </ol> <p><b>Postcondition:</b> Data about Job Posting is deleted from the database</p>
------------	---

## 9. Name: Correct

**Description:** User wants to make all the job postings work

**Precondition:** User is verified by admin and logged in, user has different job postings

Event Flow:	<ol style="list-style-type: none"> <li>1. User put the button “My Account”</li> <li>2. User chooses “My Job Postings”</li> <li>3. User is led to his Job Postings page (../my-account/job-postings)</li> <li>4. User clicks the group/ungroup button</li> <li>5. User sees all the job postings, which are sorted for Approves, In Review, and Denied</li> <li>6. User edit or delete denied job postings</li> </ol> <p><b>Postcondition:</b> New data about Job Posting is saved or deleted from the database</p>
-------------	--



## 10.Name: Approval Job Postings which is In Review

**Description:** Admin approves job posting

**Precondition:** Admin is logged in, job posting exists in review

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin clicks "Job Postings"
4. Admin is led to Unaccepted Jobs page (../admin/unaccepted/jobs)
5. Admin reads Jobs information
6. Admin clicks the green button
7. The Job Posting disappears from the list. The toastr "Job accepted successfully" appeared. In all Jobs list the mark box changes from "In Review" to "Approved"

**Postcondition:** Job attribute accepted changed to 1 - accepted

Alternate flow  
"No in review Jobs"

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin clicks "Job Postings"
4. Admin is led to Unaccepted Jobs page (../admin/unaccepted/jobs)
5. If there is no Job Posting to verify Admin sees the message "There are currently no open job submissions to approve"

## 11.Name: Approval Job Postings which is already denied

**Description:** Admin approves job posting

**Precondition:** Admin is logged in, job posting is already denied

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Job Postings"
4. Admin is led to All Jobs page (../admin/all/jobs)
5. If a Job Posting has red mark box "Denied" admin clicks the green button near it
6. The toastr "Job accepted successfully" appeared. The mark box changes from "Denied" to "Approved"

**Postcondition:** Job attribute accepted changed to 1 - accepted

## 12.Name: Deny

**Description:** Admin denies job posting

**Precondition:** Admin is logged in, job posting exists

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin presses "Job Postings"
4. Admin is led to Unaccepted Jobs page (../admin/unaccepted/jobs)
5. Admin reads Jobs information
6. Admin presses the red button
7. The message "Reasons why the job submission is denied:" appears
8. Admin fills in the field and presses the button "Ok"
9. The Job Posting disappears from the list. The toastr "Job denied successfully" appeared. In all Jobs list the mark box changes from "In Review" to "Denied".

**Postcondition:** Job attribute accepted changed to 0 - denied

Alternate flow 1  
"No Job Postings in review"

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin presses "Job Postings"
4. Admin is led to Unaccepted Jobs page (../admin/unaccepted/jobs)
5. If there is no Job Posting to verify Admin sees the message "There are currently no open job submissions to approve"

Alternate flow 2  
"Different path"

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin presses "Job Postings"
4. Admin is led to All Jobs page (../admin/all/jobs)
5. Admin presses the red button near the Job Posting which should be denied
6. The message "Reasons why the job submission is denied:" appears
7. Admin fills in the field and presses the button "Ok"
8. The toastr "Job denied successfully" appeared. The mark box changes from "In Review" to "Denied".

**Postcondition:** Job attribute accepted changed to 0 - denied

### 13.Name: Delete by admin

**Description:** Admin deletes job posting

**Precondition:** Admin is logged in, job posting exists

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin presses "Job Postings"
4. Admin is led to All Jobs page (../admin/all/jobs)
5. Admin presses the delete button near the Job Posting which should be deleted
6. The message "Are you sure you want to delete this job item?:" appears
7. Admin presses the button "Ok"
8. The Job Posting disappears from the list. The toastr "Job deleted successfully" appeared.

**Postcondition:** Data about Job Posting is deleted from the database

### 14.Name: Feature

**Description:** Admin features job posting

**Precondition:** Admin is logged in, job posting exists and is not featured

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Job Postings"
4. Admin is led to All Jobs page (../admin/all/jobs)
5. Admin clicks the feature button near the Job Posting which should be featured
6. The feature button becomes blue. The toastr "Job Status updated successfully" appeared.

**Postcondition:** New feature status of the Job Posting is saved in the database

### 15.Name: Unfeature

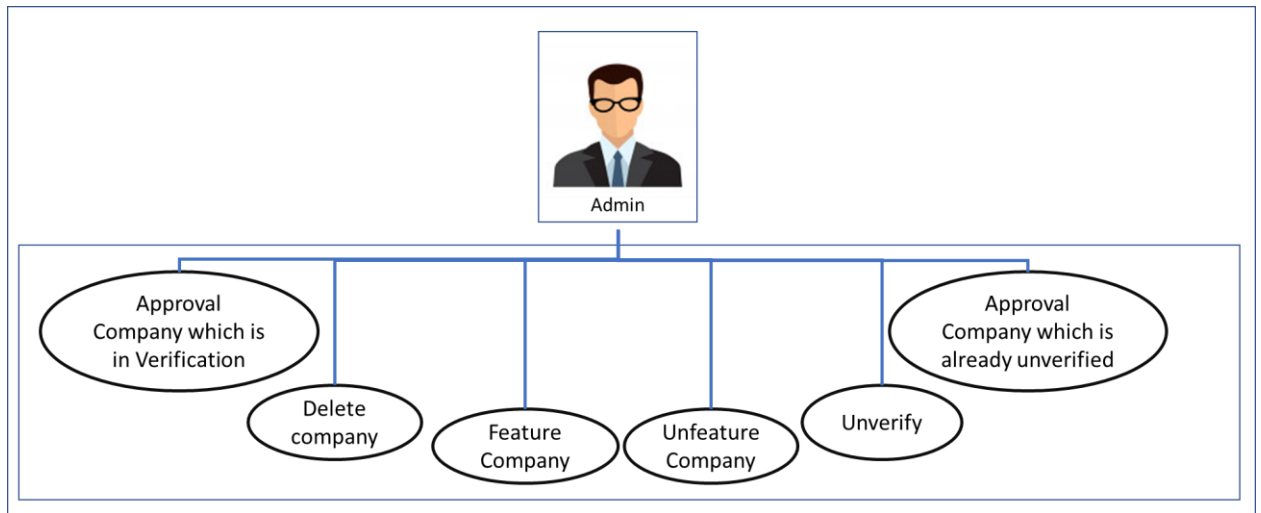
**Description:** Admin unfeatures job posting

**Precondition:** Admin is logged in, job posting exists and is featured

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Job Postings"
4. Admin is led to All Jobs page (../admin/all/jobs)
5. Admin clicks the feature button near the Job Posting which should be unfeatured
6. The feature button becomes white. The toastr "Job Status updated successfully" appeared.

**Postcondition:** New feature status of the Job Posting is saved in the database



## 16.Name: Approval Company which is In Verification

**Description:** Admin approves company

**Precondition:** Admin is logged in, company exists in review

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin clicks "Companies"
4. Admin is led to Unverified Companies page  
(../admin/unverified/companies)
5. Admin reads Company information
6. Admin clicks the green button
7. The Company disappears from the list. The toastr "Company approved successfully" appeared. In all Companies list the mark box near this company changes from "In Review" to "Verified"

**Postcondition:** Company attribute verified changed to 1 - Verified

Alternate flow "No company in verification"

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin clicks "Companies"
4. Admin is led to Unverified Companies page  
(../admin/unverified/companies)
5. If there is no Company to verify Admin sees the message "There are currently no open company applications to verify."

## 17.Name: Approval Company which is already unverified

**Description:** Admin approves company

**Precondition:** Admin is logged in, company is already unverified

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Companies"
4. Admin is led to All Companies page (../admin/all/companies)
5. If a Company has red mark box "Unverified" admin clicks the green button near it
6. The toastr "Company approved successfully" appeared. The mark box changes from "Unverified" to "Verified"

**Postcondition:** Company attribute verified changed to 1 - Verified

## 18.Name: Unverify

**Description:** Admin un verifies company

**Precondition:** Admin is logged in, company exists

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin clicks "Companies"
4. Admin is led to Unverified Companies page (../admin/unverified/companies)
5. Admin reads Company information
6. Admin clicks the red button
7. The message "Reasons why the user application is denied:" appears
8. Admin fills in the field and clicks the button "Ok"
9. The Company disappears from the list. The toastr "Company denied successfully" appeared. In all companies list the mark box changes from "In Verification" to "Denied".

**Postcondition:** Company attribute verified changed to 0

Alternate flow 1  
"No company to  
verify"

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW IN REVIEW" Admin clicks "Companies"
4. Admin is led to Unverified Companies page (../admin/unverified/companies)
5. If there is no Company to verify Admin sees the message "There are currently no open company applications to verify."

Alternate flow 2  
'Different path"

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Companies"
4. Admin is led to All Companies page (../admin/all/companies)
5. Admin clicks the red button near the Company which should be unverified
6. The message "Reasons why the user application is denied:" appears
7. Admin fills in the field and clicks the button "Ok"
8. The toastr "Company denied successfully" appeared. The mark box changes from "In Verification" to "Verified".

**Postcondition:** Company attribute verified changed to 0

## 19.Name: Delete Company

**Description:** Admin deletes

**Precondition:** Admin is logged in, company exists

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Companies"
4. Admin is led to All Companies page (../admin/all/companies)
5. Admin clicks the delete button near the Company which should be deleted
6. The message "Are you sure you want to delete this Company?" appears
7. Admin clicks the button "Ok"
8. The Company disappears from the list. The toastr "Company deleted successfully" appeared.

**Postcondition:** Data of the Company is deleted from the database

## 20.Name: Feature Company

**Description:** Admin features company

**Precondition:** Admin is logged in, company exists and is not featured

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Companies"
4. Admin is led to All Companies page (../admin/all/companies)
5. Admin clicks the feature button near the Company which should be featured
6. The feature button becomes blue. The toastr "Company Status updated successfully" appeared.

**Postcondition:** New feature status of the Company is saved in the database

## 21.Name: Unfeature Company

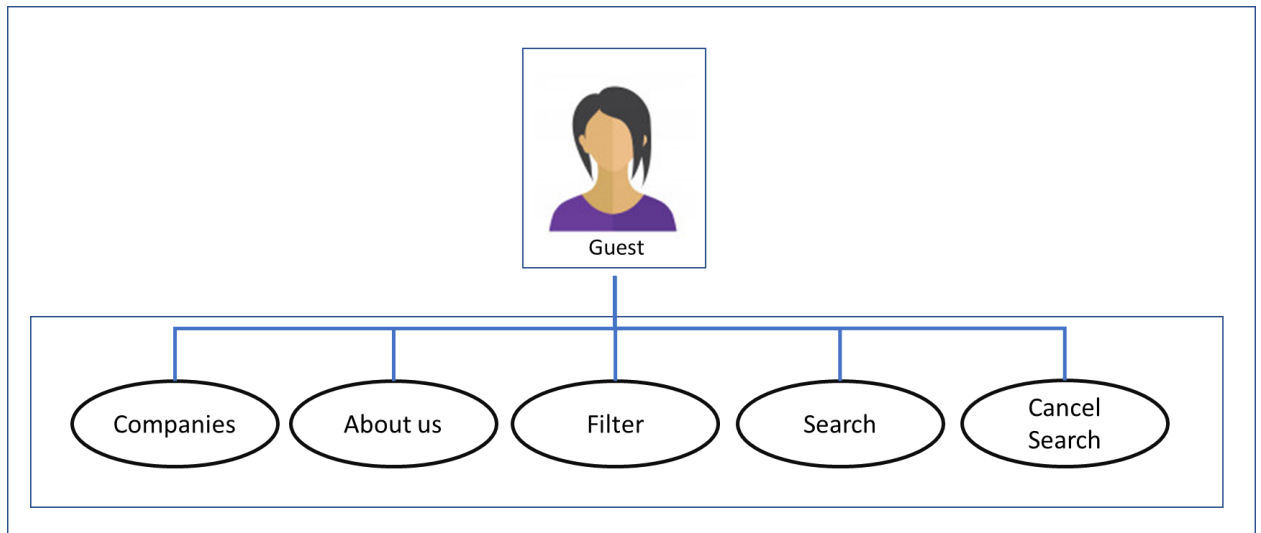
**Description:** Admin unfeatures company

**Precondition:** Admin is logged in, company exists and is featured

Event Flow

1. Admin put the button "Admin"
2. Admin sees the list with two parts "SHOW ALL" and "SHOW IN REVIEW"
3. In "SHOW ALL" Admin clicks "Companies"
4. Admin is led to All Companies page (../admin/all/companies)
5. Admin clicks the feature button near the Company which should be featured
6. The feature button becomes white. The toastr "Company Status updated successfully" appeared.

**Postcondition:** New feature status of the Company is saved in the database



## 22.Name: Companies

**Description:** Guest want to see all the companies

**Precondition:** Guest has an internet connection

Event Flow

1. Guest opens the website
2. Guest put the button “Companies” on the header
3. Guest is led to the Companies page

## 23.Name: About Us

**Description:** Guest want to see information about application

**Precondition:** Guest has an internet connection

Event Flow

1. Guest opens the website
2. Guest put the button “About us” on the header
3. Guest is led to the Companies page

## 24.Name: Search

**Description:** Guest want to search the Job Posting

**Precondition:** Guest has an internet connection

Event Flow

1. Guest opens the website on the main page
2. Guest clicks the search button
3. The search field appears
4. Guest fills in any information
5. Guest clicks the green button or presses enter
6. If such information exists Job Postings searched, all unneeded disappear



Alternate Flow  
“Invalid Input”

1. Guest opens the website on the main page
2. Guest clicks the search button
3. The search field appears
4. Guest fills in any information
5. Guest clicks the green button or presses enter
6. If such information does not exist, the message “No Job postings matches ‘data which user filled in’” appears

## 25.Name: Cancel Search

**Description:** Guest want to cancel the Job Posting search

**Precondition:** Guest has an internet connection and already searched Job Postings

Event Flow

1. Guest clicks the red button
2. All the Job Postings appear

## 26.Name: Filter

**Description:** Guest want to filter the Job Posting

**Precondition:** Guest has an internet connection

Event Flow

1. Guest opens the website on the main page
2. Guest clicks the filter button
3. The search fields appear
4. Guest fills in any information
  - 4.1 In field “Show postings posted” guest can choose “all”, “this week”, “this month”
  - 4.2 In fields “Show postings between..and” guest can choose all the different months
  - 4.3 In “Languages” field guest can fill in some language and click the “add” button
  - 4.4 In “Postcode” field guest can fill in some postcodes and click the “add” button
  - 4.5 In “workload” bar guest can choose the amount of workload moving the beginning and ending dots
  - 4.6 In field “Salary type” guest can choose “all”, “monthly”, “hourly”, “one time payment”
  - 4.7 In “Salary amount” bar guest can choose the amount of payment moving the beginning and ending dots
5. Guest clicks the “Filter” button
6. If such information exists Job Postings are filtered, all unneeded disappear. The message “Found ‘amount of job postings which are appropriate’ job postings matching your request.” Appears

Alternate Flow  
“Invalid Input”

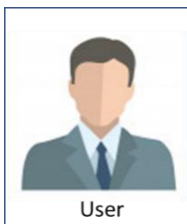
1. Guest opens the website on the main page
2. Guest clicks the filter button
3. The search fields appear
4. Guest fills in any information
5. Guest clicks the “Filter” button
6. If such information does not exist all Job Postings disappear. The message “No job posting matches your filter criteria” appears

## Actor characteristics:



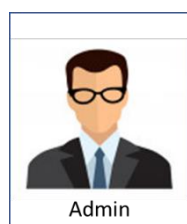
**Guests:** the system was developed for any kind of students with undergraduate, graduate, master, etc degree. An approximate age of the target audience is from 20 to 30.

Students use this application for looking for the most appropriate job offers.



**Users:** We expect to have both local and worldwide companies.

Companies use the application to allocate the job offers.



**Admin:** Admin is a any person from MIB Fachschaft Mathematik und Informatik der Universität Bern who is moderating the website. This person is expected to have some appropriate knowledge in programming field.

# Functional and non-functional requirements

## **Functional**

- User requirements: Every user should be able register.  
Every user should be able to log-in and log-out of the system.  
Every user should have a name, password, and some additional mandatory information.  
Every user should be able to add, modify and delete job postings.  
Every user should be able to add and modify information about itself.
- Guest requirements: Every guest should be able to search and filter job postings without registration
- Admin requirements: Admin should be able to log-in and log-out of the system.  
Admin should be able to approve, deny, and delete any job postings  
Admin should be able to approve or deny new companies and changes in companies data, and delete any company  
Surprise feature: Admin should be able to feature and unfeature any company or job postings
- Overall requirements: The application should be able to show all the Job Postings, all the company-partners' information and the small "about us" page publicly without any registration

## **Non-functional**

Security: all the users data should be saved in the database confidently.

Platforms: The system should work on all common browsers

The system should work on mobile-platform

User friendly: any system feature should be reachable in 3 clicks