TEAM 2

Sigma

Review package

Contents:

TEAM 2

Sigma

Requirements document

Version 4.2

Project Manager:	Imtiaz Ahmed 0444588
Requirement Analyst:	Kuchimanchi Lakshmi Prasanna 0433913
Softaware Designer:	JoonasMaksimainen 0372184
Developer:	VitezslavKriz 0457494
Designer and Developer:	Eduard Telezhnikov 0460339
Software Tester:	JuhoJuvani 044472

Revision History

Date	Version	Description	Author
02.10.15	1.0	Draft Version(vision artifact)	Imtiaz Ahmed
02.10.15	1.1	Draft Version(Functional requirement)	Joonas, Vita, Eduard
02.10.15	1.2	Draft Version(use-case details) For recruiter , organization	VitezslavKriz
03.10.15	1.3	First baseline	Juho Juvani
5.10.15	1.4		Kuchimanchi Lakshmi Prasanna
11.10.15	2.0,2.1		Prasanna, Joonas
12.10.15	2.2		Joonas, Juho
19.10.15	3.0		Kuchimanchi Lakshmi Prasanna
25.10.15	4.0	New edited version, first baseline draft	Juho, Joonas
26.10.15	4.1	Use cases redone completely, content structure redone, content table redone, typos and errors fixed	Juho, Joonas
3.11.15	4.2	Requirements documents modified according to defects	Kuchimanchi Lakshmi Prasanna
3.11.15	4.2	Use cases and use case model modified by defects.	Joonas

CONTENTS

1. INTRODUCTION

The project Sigma is a meeting point for the students and the job recruiters. This is a website which is being developed with the integration of the university to help student to get placed in a desired company.

1.1Purpose

The purpose of this document is to define the high-level requirements of the systemSigma in terms of the needs of the end users. This Vision Document applies to the system Sigma, which will be developed by a team of IT students of LUT.

The main purpose is to change the current situation of 'Job Hunting' into 'Student Hunting'. Finding a job for a student or finding a student for a job will become a trouble-free task for both concerned parties.

1.2Problem Statement

Problem	Finding an appropriate student for a job position by company and finding a desired job in company by student. Matching the right work with right talent has become difficult for individual companies.
Stakeholders	Students (Job seekers), Companies (Job provider) and Recruiters(Third party)
Impact of project	Ideally the project would produce a website where student, with right set of skills, can meet a company with a need of those set of required skills.

1.3Project Scope

The student portal application Sigma is being developed to implement an easy and reliable application for students to find dream job of their interest. This portal provides great features like sharing the student information about all their skill sets and genuine project /subject experience during their study period. The portal is also intended to provide an effective communication between the student and the organizational recruiters. It is also subjected as knowledge sharing platform retaining the present functionality available in the present systems.

1.4 Existing system

The existing system portals like Monster and LinkedIn also help student to find a job, the communication cannot be reliable there is no one to one dependency. There arise situations where the student has to wait long for being placed in the company. There are also the similar problems faced by the recruiters to get the candidates selected for the candidature.
Examples: Naukri, Monster, CAREER Experteer

Problems ofexisting system:

- Difficulty in maintaining the security for the documents
- Difficulty in performing the various tasks like browsing, navigation and searching for the required job which matches the interest and skill set of the students and vice versa
- Difficulty in selecting the skill set required for the job and vice versa
- High processing time.
- Difficulty in managing a multi number of portals for different companies.
- Student should approach a company.

1.5 Proposed system

The proposed system will be a website managed by the universities, which will open more possibilities over the existing system. The proposed system allows the student to create a portal with their interest; skills set which in-turn are connecting directly to the dream companies they wish to work.
It provides facilities to both the users like recruiters and company to directly exchange interests. This provides an efficient, effective and reliable communication between both the parties. This will also give expert opinion for each other.
Benefits of the proposed system:

- It facilitates provide knowledge documents across the company
- It allows the students to upload the documents from their systems
- This is easy to use and helps students to browse, navigate and search for required jobs and also helps recruiters to find students.
- Provides a facility to restrict the permission levels to access the personal profile and access documents.
- The information can be retrieved from anywhere and anytime by saving time and providing the users with up to date information
- The centralized data base system provides the necessary functionalities avoiding any discrepancies.
- This helps in publishing the entire work of the student. It has excellent user interface experience and has access to information on removing any ambiguity.

1.6 Objective

The main objective is to use the project sigma in an effective way to get high numbers of employment rates before the student graduates from the university and at the same time to provide a better job website for a recruiter to get skilled and knowledge oriented student heading projects.

2. FEASIBILITY STUDY

The software development is plagued by the scarcity of the available resources, security and delivery the services. To avoid these problems and the feasibility study with the three main key aspects are considered at the early stage to avoid major confusions and misconceptions.
The three key aspects are:

- Economic feasibility
- Technical feasibility
- Operational feasibility
- User Environment feasibility

2.1 Economic Feasibility

Sigma is undertaking student-recruiter specialist ventures. This is exceptionally helpful for the organization enrolment specialist to keep up a specialized connection between student and the recruiters. The framework of the improvement is the university itself which gives students understudy alternative advantages before graduating.

The selection representative's get joins the university with their terms and conditions. Subsequently it is monetarily practical for this task sigma to be actualized

2.2 Technical Feasibility

Sigma is a web based project. It uses the following tools and frameworks to handle different tasks:

- CKAN
- Backend: Python3.X
- Django
- Database search engine
- Apache operating in Linux

The website is mainly a meeting point, which allows creating a student portal which is linked to the university portal, where the recruiter can see the student profile and their projects with their skill set and area of interest. There are few add-ons available, like flexible search engine.

2.3 Operational Feasibility

The website is mainly for the student welfare. This website meets the requirements of the student to get acquainted with their dream Company and job and recruiter can easily select one among the applicants based on skill set.

The student and recruiters are well trained to use the technology, which reduces the detailing and training sessions making the website more operationally reliable.

2.4 User Environment Feasibility

The University Fresh Graduates is a large sophisticated community that demands the flexibility and response time for job finding and besides companies also look for graduates in an effective manner that an on-line interest exchange platform can provide.

The users are educated, computer literate, and in most cases they have their personal/official computers. The ability to register in the system via computers and to review their updates and information on-line would greatly streamline job hunting or job giving process.
The initial release of Sigma will be limited to LUT, Computer Science students.

3. REQUIREMENT ANALYSIS

This is the critical phase for the software to deploy its project with high amount of success rate in implementing the project. The requirement analysis is to identify the problem and propose the evaluation and synthesis to the generated problem by analyzing the driven conclusions. The main problem identified is to provide ease of access to student to find their desired company by providing the means to communicate and interact with the company. The same is applicable for the company to select the appropriate talent to perform the particular task in the job.

3.1 Stakeholder Identification

The main stakeholders of this website are

- Students
- Companies
- Recruiters
- Administrators

3.2 Stakeholder needs

1. The stakeholder needs are different to each other

Stakeholder	Need
Student	Job
Company	Skilled Students
Recruiter	Reduce time in selection process and finding student with right set of skills
Administrators	Ability to validate and control user accounts, postings and data on the site

2. Stakeholder Summary

Name	Description	Responsibilities
Students	Technically inclined university student	- ensures that the system will meet the needs of students - they will maintain their profile - will receive notifications according to their choices
Company	Different companies who offer jobs	Represents the interest of companies
Recruiters	Third party who works for companies	Ensures that meets the needs of companies
Administrators	Technically inclined personnel that upholds and controls the site	Validate users, postings and any new information presented to the site. Also moderates any unwanted or

		faulty content.

3. User Summary

Name	Description	Responsibilities	Stakeholder
Students	Provide their interest, follow desired posts	Maintaining an updated profile and responsive to the system	Self represented
Company	Tracking students(searching students)	They contact with chosen students selected from a pool provided by their requirements matching	Self-represented or by third party
Recruiters	Will search students on behalf of company	They can post jobs, they can contact students and hire them	Represented by company

4. User and stakeholder needs with proposed solution

Need	Priority	Concerns	Current Solution	Proposed Solutions
Finding perfectly suitable student for an open job	High	The process slow and inefficient.	Company post jobs via different medium and wait for the applications from students, then make a pool for taking interview.	Company can find a pool of students according to their requirements and can call them from the pool immediately.
Finding a desired job	High	The process is time consuming and needs much effort.	They apply for jobs by seeing job advertisements or leave an open application.	Company will look for students and contact them according to their qualifications. Students also can follow many jobs according to their matched requirements.

3.3 Actors and their Events

Actors	Events and their Motive
Students	**Login** : Student Login portal is mainly to login the page **Signup** : signup as a new student to create a new portal ID **Register**: can register in the portal if needed! **View :** view the existing job notifications or new job opportunities **Update:** To update their profile with new skill set/projects/recommendations/ achievements **Delete:** To delete unwanted scrap. **Add:** Add skill set/interest **Major:** Domain of their course **Minor:** Sub domain of their course **Description:** 1. All the above events are achievable if the website is functioning well and can support the user with necessary actions 2. Poor functioning can be addressed to admin immediately 3. The student should be approachable by an email or phone call 4. All the information stated should be reliable.

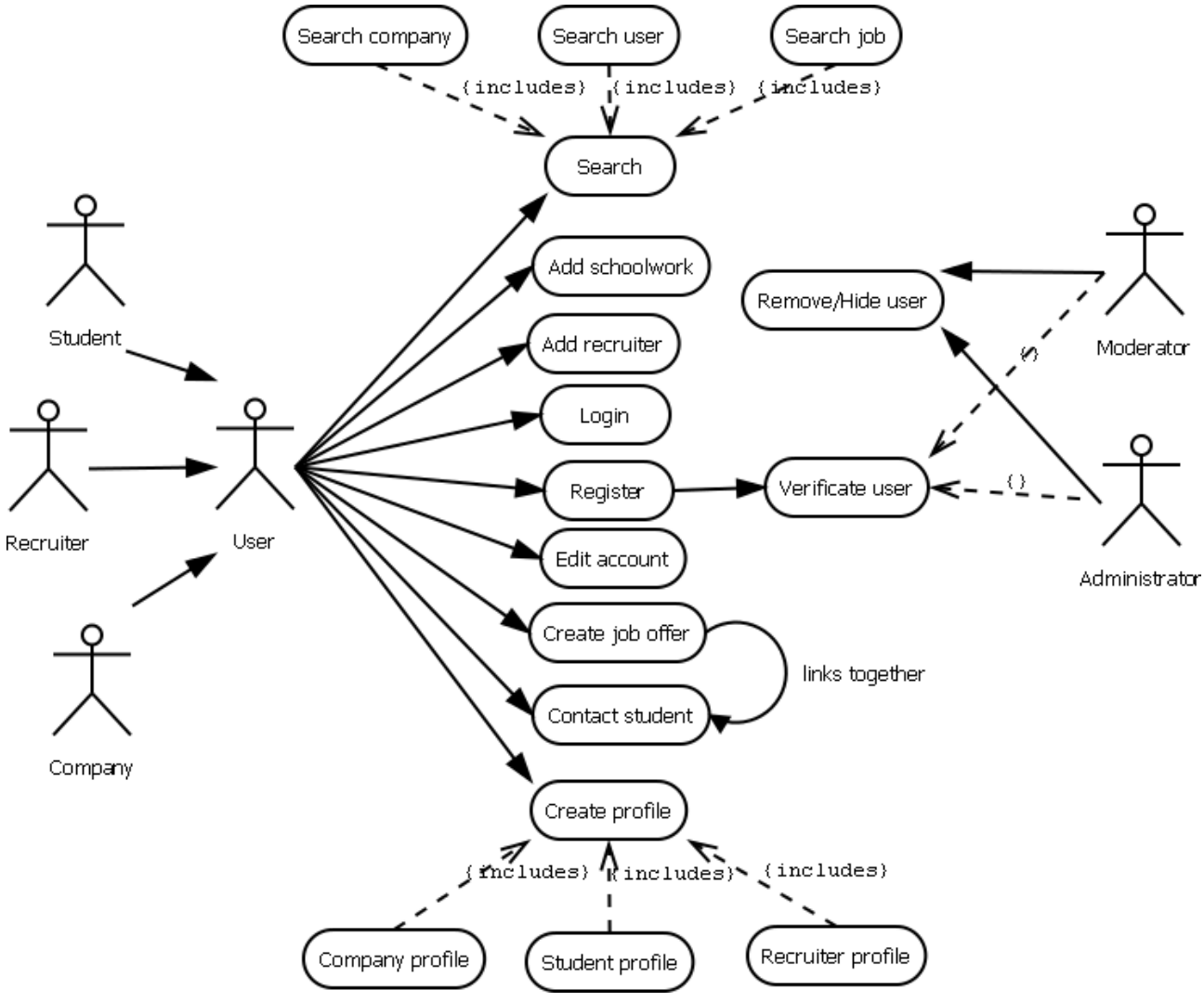
Student	**View**: Can view the profile **Edit:** Can edit if needed **Create portfolio**: Create an account(portfolio) to list down the details needed for job portal **Update:** To update the portfolio **Accept:** To accept the job invitations and proceed further **Decline:** To decline if, not interested **Negotiations:** To deal with the company and recruiters **Search:** Search for jobs **Verify :** To verify the company **Request:** Sending request for the interest in a company **Share**: Can share any document or information need to the recruiter
Company/recruiter	**Create**: An account for publishing jobs **Login:** To login **View**: This function can be used in two types 1. To view job announcements(to edit modify) the open application details 2. To view students skill set and their portfolio **Update:** This also has 2 types 1. To updatejob announcement 2. To update their requirement portfolio **Delete**: This also has 2 types 1. To delete the old job announcements 2. To delete the students who decline the offer **Search:** Search for required skill set and students **Request :** Sending request for the skill set matching students **Accept:** To accept the student interest if it matches the requirement **Verify:** To verify the student **Decline**: Decline if when the skill set is not matching **Negotiate:** Deal with students **Description:** All the above events are achievable if the website is functioning well and can support the user with necessary actions.

Administrator	**Add:** Add account of the newly joined members to the website **Change:** Change the details if needed (mostly in security issues). **Approve:** Can approve the student/recruiter as valid users **Verify:** To verify the users **Remove/delete:** Remove, hide or delete account incase student has already graduated **Dealing negotiations:** Deal the negotiations if needed if there are issues like 1. Rationaldiscrimination 2. Irrelevantdiscussion 3. Crisis 4. Demand for theskill set 5. To approachstudentmorepersonally 6. Discussions and arguments

4. USE CASES

This chapter presents an overall use case diagram for the Sigma project and it's use case descriptions.

4.1 Use case diagram



4.2 Use case descriptions

Use Case: Search function
ID: UC_01
Actors: Student Recruiter Company
Preconditions: User has created an account and has logged in.
Basic Flow of Events: 1. User will select the “Search”-option from the website. 2. User selects the certain type of search for three options. - Company - User - Job 1. User types the input data for the search depending from the search type. 2. User marks the special sections as negative or positive depending from the search type. 3. User selects “Start search”. 4. Search engine starts to find similar targets from server depending the search input and selections. 5. Search results page is opened and search results are arranged and shown by the most relevant targets according to inputs and selections.
Alternative Flows: The search input and sections to be marked depends from the search type the user chooses.
Post-Conditions: The search gives search results depending from the user’s inputs and selections. Results also show the amount of found targets. If there is none relevant choice from the server database, the search gives message “No targets were found”.

Use Case: Add coursework
ID: UC_02
Actors: Student
Preconditions: User has created an account and has logged in.
Basic Flow of Events: 1. User will select the “My profile”-option from the website, which opens section containing information from account and user. 2. User selects the “Add coursework”-option, to add his/her works, documents and programs to be shown in his/her public profile page in the website. 3. User selects skill tags to be linked to the file. This will help to prove his/her certain skills used in to create this certain work. 4. The list uploaded files are shown with “Delete file”-option. 5. User select “Add file” to select the file to be transferred in the server database and to be shown in the portfolio section in the profile. 6. User selects the file from his/her computer to be uploaded in the system.

7. The selected file is uploaded to the system. 8. The uploaded file is shown in the user profile's portfolio.
Alternative Flows: The user selects "Delete file"-option behind the target file to delete it from the account. After selecting the option, confirmation window is shown to ensure, if the user is sure about the deletion.
Post-Conditions: The uploaded file is shown in the user profile's portfolio. The files can be selected to download them and inspect them by other users.

Use Case: Contact student
ID: UC_03
Actors: Recruiter Company Student
Preconditions: User has created an account and has logged in.
Basic Flow of Events: 1. User will select the "Search"-option from the website. 2. User selects the "Student" type of search from three options. 3. User types the input data for the search. 4. User marks the special sections as negative or positive. 5. User selects "Start search". 6. Search engine starts to find similar users from server 7. Search results page is opened and search results are arranged and shown by the most relevant users according to inputs and selections. 8. The user selects the "Invite" to contact the student he/she likes. This opens message part. 9. User selects the job offer his/her company likes to give the student. 10. User writes message to message box to give additional information about the job offer. 11. User selects "Send message" to send the message to target user. 12. Student is informed by email about the contact. 13. Student can later respond to the invitation by selecting "I'm interested" or "No" and add message for explanation.
Alternative Flows: Student will not answer the invitation or students account is removed. This makes the message to be removed.
Post-Conditions: The contacting user is informed by message "Message has been sent" and target user sees message in his/her email and account.

Use Case: Login
ID: UC_04
Actors: Student Recruiter Company
Preconditions: User has created an account and is not logged in.

Basic Flow of Events:

1. User will open the website and selects the “Login”-option from the website.
2. User types his/her nickname and password.
3. User selects “Log in the account”
4. The nickname and password are checked from the system database.
5. If the nickname and password are correct the user logs in the system.
6. The “Login” option changes into “Log out”, which logged in user, can use to log out from the system website.

Alternative Flows:

If the password and/or nickname are incorrect, the user is informed by error message and is offered a way to recover the password by email.

Post-Conditions:

The user is logged in the system.

Use Case: Register
ID: UC_05
Actors: Student Recruiter Company
Preconditions: User doesn’t have an account.
Basic Flow of Events: 1. User will open the website and selects the “Register”-option from the website. 2. User writes all compulsory and optional information depending from his/her user type. 3. User types email account to be contacted by system moderator and administrator for the account verification. 4. User selects “Register account” to send the typed information and account to be verified. 5. Depending from the user type, the moderator or administrator verifies the new user account. 6. When the verification of the account is complete, message with activation code is sent to user’s email address. 7. User checks the email for the message with account activation code and message about verification of the account completed. 8. The user selects the “Login”-option from the website. 9. User types his/her nickname and password first time. 10. User types the account activation code. 11. User gets message that the registration of the account is fully completed. 12. The profile creation page opens to the user.
Alternative Flows: If the account doesn’t get verified by the verification process due to error, the message is sent to user’s email address to inform about the problem.
Post-Conditions: The user gets his/her account verified and registered in the system.

Use Case: Edit account
ID: UC_06
Actors:

Student Recruiter Company
Preconditions: User has created an account and is logged in the system.
Basic Flow of Events: 1. User selects “Settings”-option from the website to open the settings of the user’s account. 2. User will make changes to the sections in the to the settings page. These are for example to change the password or the email address for the user’s account. 3. User will select “Apply changes” to apply the changes he/she did in the sections. Password-section has its own apply-button. 4. The changes in the account settings are saved in the system. 5. User gets message that changes have been accomplished.
Alternative Flows: If the user tries to leave the settings page before applying the changes, user gets message that is he/she applying the changes he/she just made in the settings. If user chooses not to apply changes or closes the website instead of selecting, the changes are not saved.
Post-Conditions: The changes in the settings are saved.

Use Case: Create job offer
ID: UC_07
Actors: Recruiter Company
Preconditions: User has created an account and is logged in the system.
Basic Flow of Events: 1. User selects “Create job offer”-option from the front page of the website, which then opens page to make announcement of the company. Only recruiter- and company-type users have this option. 2. User selects the company, for which he/she makes the job announcement. Only those companies, which company user owns or recruiter user works for can be selected and only one company can be selected. 3. User types all the compulsory and optional description information about the job. For example name, job title and short description. 4. User selects the specific requirements for the job. For example certain skills and coding languages. 5. User will select “Post job offer” to send the job offer in the systems database. The job offer is shown in the job announcement list in the company’s profile page. Company user or recruiter can delete the job offer by selecting “Delete job offer”. The students have option to apply the job offer. 6. The job offer is sent and saved in the system and can be applied by students.
Alternative Flows: If the user tries to post the job offer, which is missing compulsory information, he/she gets error message about missing information and that the job offer cannot be saved. Closing the website before posting causes the job offer posting to be cancelled.
Post-Conditions: The job offer is saved to the system and it’s shown in the job announcement list in the company’s profile page.

Use Case: Add recruiter
ID: UC_08
Actors: Company
Preconditions: Company user has created an account and is logged in the system. Target recruiter has an account in the system.
Basic Flow of Events: 1. User opens the company's settings page by selecting "Settings"-option from the front page. 2. User opens list of company's recruiters by selecting the list called "Recruiters" from the "Settings"-page. 3. User selects "Add recruiter"-option from the "Recruiters"-list, which opens the search function in smaller window with settings to search recruiters already selected. 4. User will fill in the missing information sections of the search. 5. User starts the search by selecting "Start search". 6. Search engine starts to find similar targets from server depending the search input and selections. 7. Search results page is opened and search results are arranged and shown by the most relevant targets according to inputs and selections. 8. User highlights the most suitable recruiter(s) by selecting target recruiter(s) and "Hire selected"-option becomes available. 9. User selects "Hire selected" and hiring messages are sent to recruiters to be accepted. 10. Target recruiters accept the hiring messages and company user gets the messages of accepted offers. 11. Company user finalizes hiring by confirming the message sent from recruiters. 12. Target recruiter is saved in the company's recruiter list.
Alternative Flows: The company user can delete the recruiters from the list and deleted recruiters are informed by messages. The recruiters already working for the company are disabled to be selected and hired second time.
Post-Conditions: The information about recruiter working for certain company is saved in the system and company's list of its recruiters.

Use Case: Create profile
ID: UC_09
Actors: Student Recruiter Company
Preconditions: User has created an account and is logged in the system. The profile for the user hasn't been created.
Basic Flow of Events: 1. The system opens profile page of the user automatically, when the user logs in for the first time. Depending from the account type, every user has different type of profile page. - Company

- Student
- Recruiter

1. Introduction of making the profile page starts. The introduction shows key sections of the profile page and gives short instructions to modify different parts. The type introduction depends from the user type and the instructions can be skipped.
2. User fills the sections in the profile page using instructions of the introduction.
3. User applies the changes made to the profile page by selecting “Apply changes” and selecting positive option in confirmation window after it.
4. The modifications to the users profile page are saved in the system.

Alternative Flows:
The profile page can be left empty and be modified later. The recruiters working for certain companies can modify company’s profile page. The profile page can be completely deleted or completely new page can be done by selecting “Create profile page” from “Settings”.

Post-Conditions:
The profile page for the user is created and saved in the system.

Use Case: Verification of user
ID: UC_10
Actors: Moderator Administrator
Preconditions: Moderator or administrator has logged into system. New account has been sent to administrator or moderator be verified.
Basic Flow of Events: 1. User selects “Manage accounts”-option from the webpage. Object also shows how many account situations are waiting to be solved. 2. User selects the “Verify accounts” option, which brings up list of all user accounts which need to be verified. 3. User selects the target account, which brings up window of all the data written and selected options in the selected account’s data. 4. User checks the information in the account and decides whether to accept or deny the verification of the new account. 5. User selects the “Verify account”-option and account activation code is sent to email address of the new user. 6. The new user uses his/her activation code and finalizes the registration process.
Alternative Flows: If user denies the verification of the account, it brings up message window, where user can explain the reason for denied verification. Denied account will be deleted.
Post-Conditions: The new account is verified and ready for the finalization of the registration process.

Use Case: Remove/Hide user
ID: UC_11
Actors: Moderator Administrator
Preconditions: Moderator or administrator has logged into system.
Basic Flow of Events: 1. User selects “Manage accounts”-option from the webpage. Object also shows how many account situations are waiting to be solved. 2. User selects the “Remove/Hide accounts” option, which brings up window for searching users. 3. User selects the parameters for the search and target account that must be removed or hidden. 4. User selects “Start search” to begin the search. 5. Search engine starts to find similar targets from server depending from the search input and selections. 6. Search results page is opened and search results are arranged and shown by the most relevant targets according to inputs and selections. 7. User selects target users to be removed or hidden. User can select several accounts to be removed of hidden. After selecting the accounts, the user must confirm his/her decision. 8. User confirms his/her decision and selected accounts are hidden and removed from the system.
Alternative Flows: If user doesn’t confirm his/her actions, nothing happens.
Post-Conditions: Selected accounts are removed or hidden.

5. REQUIREMENT SPECIFICATION

There are mainly two groups

- Functional requirements
- Non Functional requirements

5.1 Functional Requirements

ID	Functionality	Description	Type
FR01	Login	User can login.	Essential
FR02	Logout	Users can logout.	Essential
FR03	Register	User can sign up.	Essential
FR04	Delete account	Users can delete own account.	Essential
FR05	Edit account info	User can edit information of its account. (name, email, password)	Essential
FR06	Own organization	User is owner of organization, recruiter, student, moderator or admin.	Essential
FR07	Edit other accounts	Moderator can edit student-, recruiter- and organization –accounts.	Essential
FR08	Edit user rights	Administrator can add and remove other users and moderators.	Essential
FR09	Administrator role extends moderator	Administrator role includes moderator capabilities.	Essential
FR10	Owner creates accounts	Owner is the creator of the organization in the system and can add recruiters to its organization.	Essential
FR11	Approve organization	Moderator can approve the creating of organization account.	Essential
FR12	Employment	Owners of organizations and recruiters can belong to multiple organizations.	Essential
FR13	Post job listing	Owners of organizations and recruiters can post job listings.	Essential
FR14	Invite to interview	Owners of organizations and recruiters can invite students to interview.	Essential
FR15	Student search	Owners of organizations and recruiters can search for students by skills etc.	Essential
FR16	Search parameter selection	User can make different searches by different parameters.	Essential
FR17	Reply to invitation	Student can reply to the invite positively and negatively.	Essential
FR18	Document sharing	Student can add and show actual assignment thesis and reports.	Essential

FR19	Verify users	Moderator can verify other users.	Desirable
FR20	Comment	Student can add explanation to invite reply.	Desirable
FR21	Edit profile	User can edit other information of its account. (photo, bio, content information)	Desirable
FR22	Job feed	Student can view latest job listings.	Desirable

5.2 Non-Functional Requirements

ID	Requirement	Description
NFR1	User interface should be straightforward and simple to use	Minimal amount of clicks
NFR2	User interface should be intuitive	Logic of using the site should be clear in short time after looking over the first page
NFR3	Site should have clear and simple instructions on how to use it (Both search and registration)	Easy instructions on how to register as a user and how to search offers with minimal confusion
NFR4	Reliability	There should not be empty pages or errors appearing on the site.
NRF5	Predictability	Search functions should provide predictable results: - Specific job offers according to keywords - Student search should return all fitting skills and talent from students according to search
NFR6	Accuracy of information	Both job offers and user information should be up to date. Job offers should be timed to be visible or have their status changed at the end of recruitment dates. If student graduates, information should be hidden or status changed within reasonable time.
NRF7	Performance	Response delay 1-3 seconds, pending on traffic and users connection.
NRF8	Availability	Site uptime should be 24/7

NRF9	Supportability	Site should be easily managed by administrators and moderators with minimal effort. - Post edit - Post removal - Profile control (Hide / Show profile) - Edit Profile / Company site and information
NRF10	Database	Data must be easily controlled and privacy of user information guaranteed.
NRF11	Language (English)	Site should be available in English
NRF12	Server (Linux)	Site must be hosted and held on a Linux server provided by the LUT
NRF13	Security	SIGMA is concerned for the individual privacy issues, it has special features to allow and restrict information when needed, it also have many protocols like web security tools and standard security technology like SSL to overcome security issues related with passwords and file uploads. There will be a check on following issues 1. Duplication of files 2. Authorization 3. Authenticity 4. Validity of data
NRF14	Capabilities	The will be error free edition of website, where the typos and grammatical errors will be identified and corrected automatically, allowing the users a error free copy of the contents. In addition to this SIGMA also provides a user friendly system which can be easily accessed by a naive user.
NRF15	Usability	The main aspect of usability is availability and accessibility of the website. If the website is not reachable by the students and the recruiters than that website is totally useless. **Server time:** The SIGMA portal is able to easily access website without any errors in loading the page. The contents will be loaded all at once.

		Broken Links: The SIGMA will not have any broken links **Clarity:** The information in the SIGMA will be consistent with easy navigational architecture. **Credibility:** The SIGMA will be trustworthy to acquire a dream job. **Relevant information:** The contents in the SIGMA will be relevant without any ambiguity. **View and Appearance:** The website gives a professional environment with easy navigation and explicit display.
NRF16	Design Constraints	The design content will have the first view with simple view of a window with all necessary functionalities and with design flexibility. The contents can be viewed on personal computers, laptops and tablets.

6. PRODUCT OVERVIEW AND FEATURES

6.1 Product Perspective

6.1.1 System Interface

ID	Name	Description	Type
S01	Web	System will have web interface at phase 1	Essential
S02	Server	The server component of the system shall operate on Linux OS	Essential
S03	Client	The client component of the system shall be OS independent	Essential

6.1.2 User Interface

Assumptions and Dependencies
The system Sigma should use open data integrated with CKAN. It is assumed that it will be OS independent.

ID	Name	Description
UI_01	Home page	This will be the main and first view of the portal. It will simply display a window to log in or to register.
UI_02	Log in view	It will receive user name and password and will start session accordingly
UI_03	Registration view	It will provide a registration form
UI_04	Search view and Result view	Search by key words and it will display result in list
UI_05	Settings	Users can change or update information
UI_06	Profile view	Users can change their information
UI_07	portfolio	It is mostly an organized presentation of users It relates to information regarding skills set, projects

		and so on
UI_08	Administation view	Mainly responsible for maintaining the website without any failures and errors

6.2 Product Features

This section defines and describes the features of the system Sigma. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users.

Details	**Description**
Login	Students, Companies and Recruiters will be provided a valid ID and password for entry to Sigma. The system shall enable a user to change their temporary password.
Register	The system will provide distinct registration forms to fill up by students, companies or recruiters. Recruiters should be approved by company.
Post Job	The system will allow companies or approved recruiters to post job directly.
Search Student	The system will allow companies or approved recruiters to search students according to requirements matching.
Enter, Update, and View Information	The system will allow all users to maintain and to update their own profile.
Contact Student	The system will allow companies and approved recruiters to ask contact information from students.
View Job Posts	The system will allow students to see job post according to their preset preference which is matched with his/her qualifications.
Apply for Job	The system will allow students to apply for any open position by any company. .
Other information	The system will support multiple simultaneous users

6.3 UI mockup


Sigma
search
latest
iaml
Start searching for a job right now!
Just select your skills:
Javascript
Python
PHP
Ruby
Frontend
DB

Sigma search latest iaml

@iaml
Eduard Telezhnikov

Studies at:
[edit] LUT (Computer Science)

Skills:
[edit] C#/XAML, Python, Django, HTML5/CSS3, client-side JS

Bio:
[edit] Something something I dunno

Sigma search latest iaml▾

Search results for language:Python[x] university:LUT[x]

@iaml Eduard Telezhnikov

@vita Vitezslav Kriz

1-2 *of* 2

Parameters:

language:

Python x

PHP + Perl +

Ruby +

university:

LUT x

GLOSSARY

Sigma	Project Name
API	Application Program Interface
Monster	Job portal
LinkedIn	Job Portal
portal	An internet site which is used to access links
CKAN	Comprehensive Knowledge Archive Network
Django	Web Framework
Client	Access the services available to server
python	Programming language
JavaScript	High level programming language
Linux	Computer operating system
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
My SQL	My Sequel
MSDN	Microsoft Development Network
PDF	Portable document Format
REST	Representational State Transfer
SOA	Service Oriented Architecture

Project Sigma

Software architecture document

version: 1.0

Version control:

Version	Date	Status	Authors	Mentions
0.1	14.10.2015	Template	Juho, Joonas	Template for document
0.2	24.10.2015		Juho, Joonas	Draft
0.3	26.10.2015		Juho, Joonas	Draft, fixes to all diagrams, package descriptions require input from designers for last changes before finalizing Use case realization unfinished
0.9	29.10.2015	Almost complete version of the architecture document.	Joonas	Use case realization added. Some package descriptions missing.
1.0	3.11.2015	First complete version of the document.	Joonas	Use case realizations modified. Other modifications made by defects.

Project Manager: Imtiaz Ahmed 0444588
Requirement Analyst: Kuchimanchi Lakshmi Prasanna 0433913
Software Designer: Joonas Maksimainen 0372184
Developer: Vitezslav Kriz 0457494
Designer and Developer: Edard Telezhnikov 0460339
Software Tester: Juho Juvani 044472

Table of content

1. INTRODUCTION

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant decisions which have been made on the system. Purpose is to make the customers and group members to understand overall description of the system functions and functionality.

1.2 Scope

This software architecture document provides an architectural overview of the project Sigma's system. The system is being developed by students of Lappeenranta University of Technology (LUT) from course Software Quality, Processes and Organizations. Students belong into Team 2.

The purpose of the system is to provide communication between companies and the students of the LUT. This document has been generated by using learning material and templates from website called Unified Process for EDUcation, or UPEDU. The system will use CKAN in its functionality.

1.3 Definitions, Acronyms and Abbreviations

These are found in the glossary of supplementary documents.

1.4 References

UPEDU Software Architecture Document guides and instructions
http://www.upedu.org/process/artifact/ar_sadoc.htm
UPEDU Use case realization instructions
http://www.upedu.org/process/artifact/ar_ucrea.htm

2. ARCHITECTURAL REPRESENTATION

This document presents the architecture as series of views, which are use case view and logical view. These are views on an underlying Unified modeling Language (UML) model created with program called Dia Diagram Editor. Only the most significant use-cases, packages, actors and descriptions are presented in this document, because for example many of the use-cases are similar and they differ only by their actors.

3. ARCHITECTURAL GOALS AND CONSTRAINTS

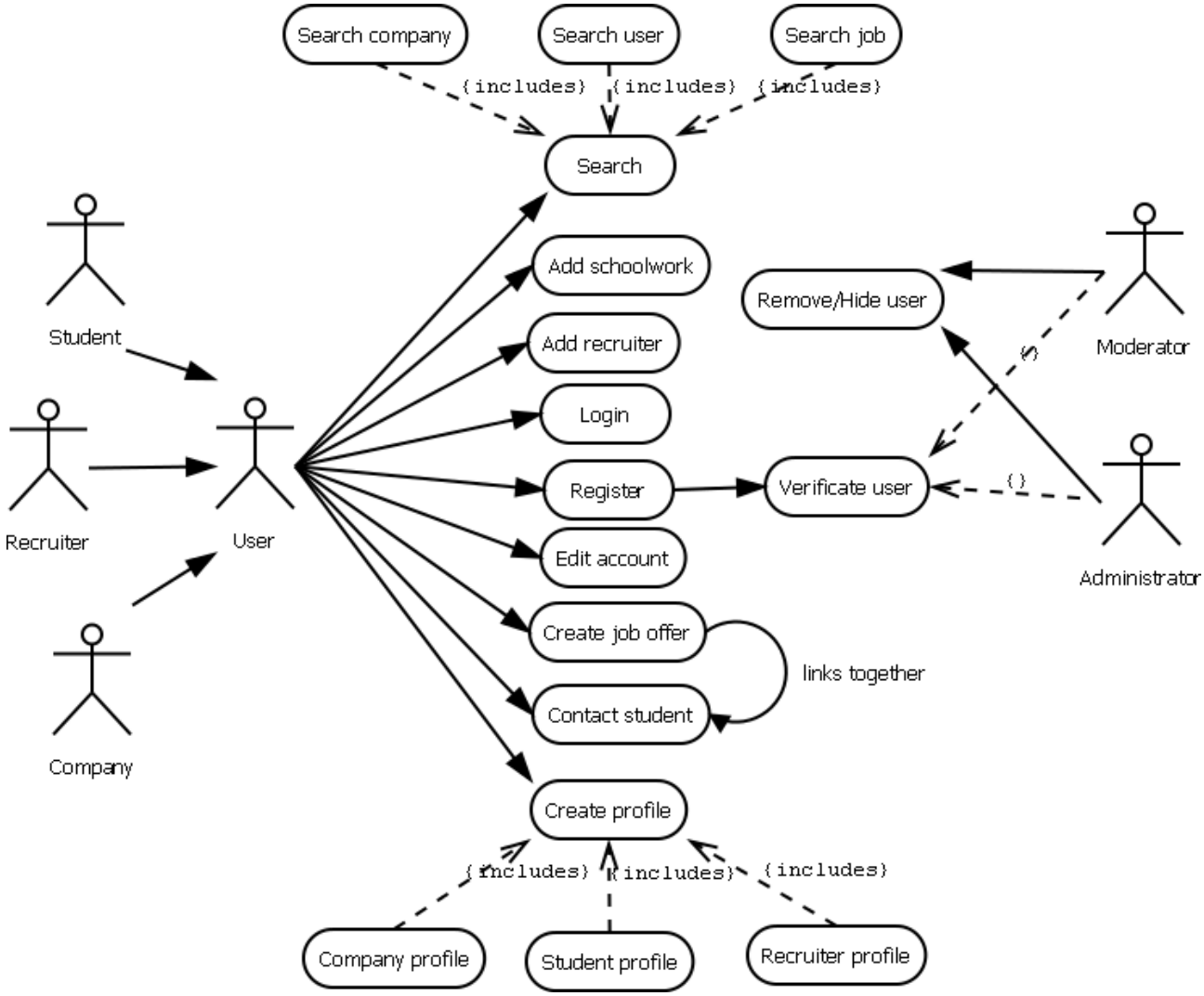
Here are some key requirements and system constraints that have significant bearing on the architecture. They are:

- Although working on smaller scale, the servers of the system must be capable to handle large amount of users and data simultaneously.
- Using CKAN brings restrictions and changes to coding structure.
- The system needs decent group of maintenance personnel, like administrators and moderators, to keep the system up and running. Without proper maintenance the usage of system may suffer and it can be huge constrain.
- The users of the system must follow the international and national rules and laws set to protect the employees and good working conditions.
- The personal and other confidential information put in the system must be protected from outsiders and used only by selected users. The system must ensure complete protection of the data from unauthorized access. All accesses are subject to user identification and password control.
- The system will be implemented as a client-server system. The client portion resides on personal computers and smart phones and the server portion must operate on LUT's UNIX server.
- The system must be available to use from personal computers and mobile devices with internet connections.

4. USE CASE VIEW

Use case view is based on the models and use case descriptions from the requirements document.

4.1 Use case model

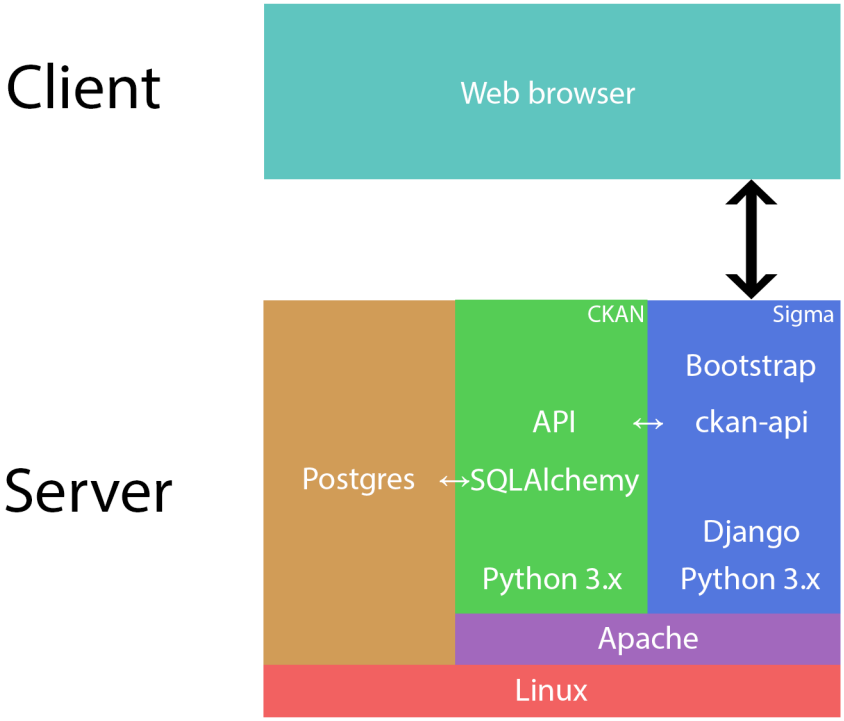


4.2 Use case realizations

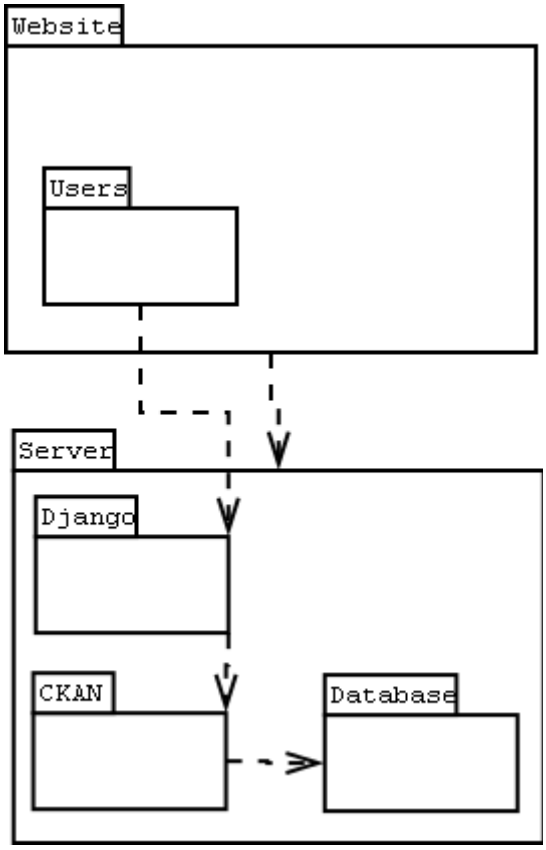
Use case realizations are in the appendix of the document.

5. LOGICAL VIEW

5.1 Overview



5.2 Package diagram



5.2.1 Users

Contains the users: Student, Company, Recruiter and Administrators (Administrator and Moderator). The users of the system.

5.2.2 CKAN

Contains the CKAN API.

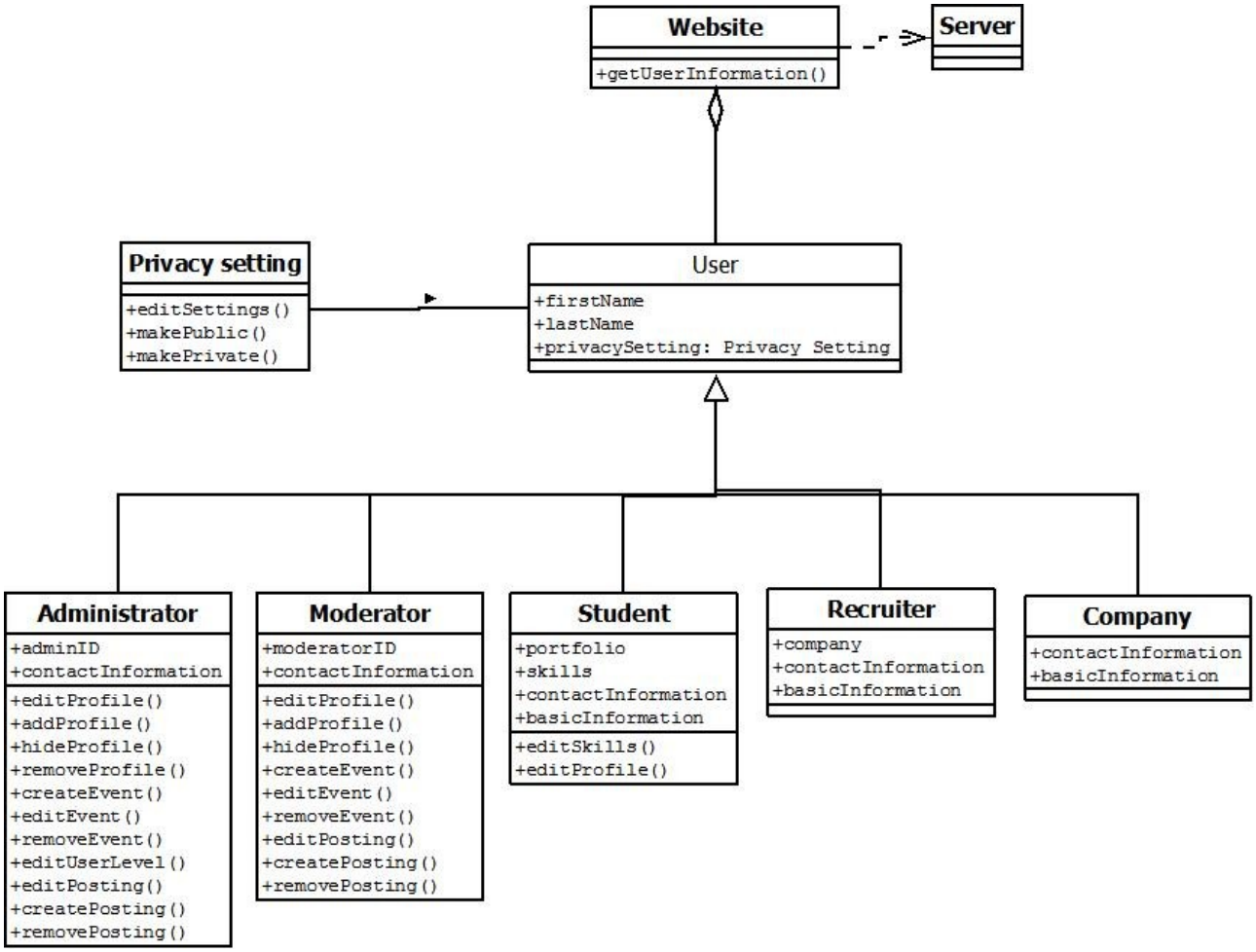
5.2.3 Django

Works as a link between website and CKAN.

5.2.4 Database

Contains the data from users and the system.

5.3 Class diagram



6. SIZE AND PERFORMANCE

The architecture of the site is based on client-server model, where the functions and data are fully controllable in a single location, set to serve many clients. It allows security and control over the data, while maintaining easy and direct connection for developing and making changes to the systems core structures.

7. QUALITY

The software architecture supports the requirements presented in the requirements document.

APPENDIXES

Use case realizations

1. **Use case <Login>**

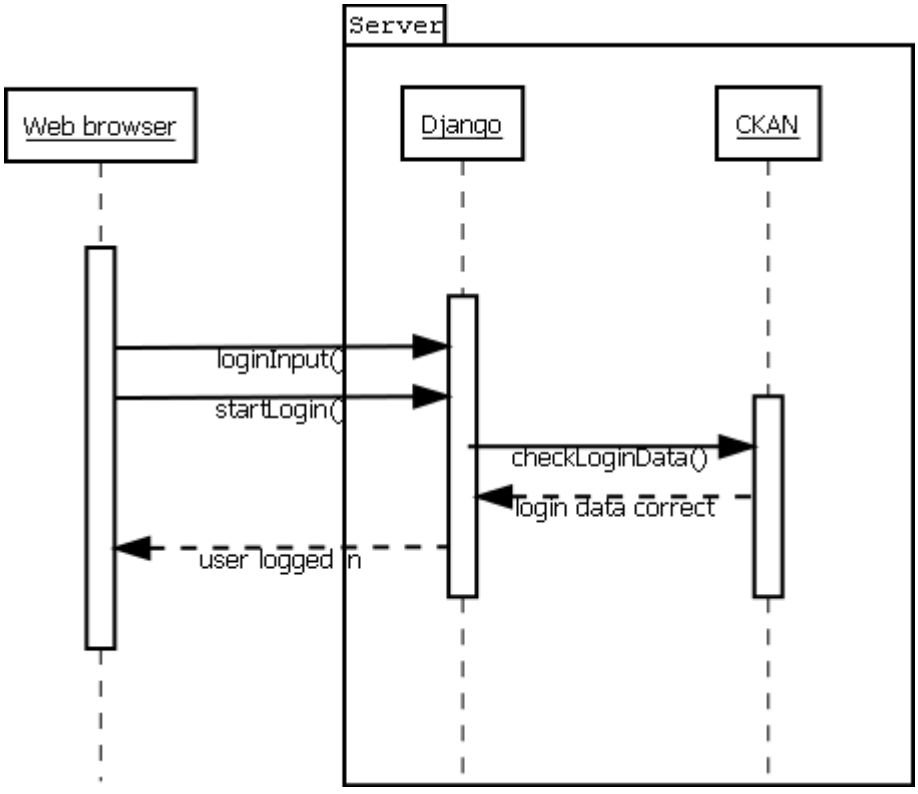
Brief description

User logs in the system using web browser, while Django and CKAN act as link to server database.

Flow of events

1. User will open the website and selects the “Login”-option from the website.
2. User types his/her nickname and password.
3. User selects “Log in the account”
4. The nickname and password are checked from the system database.
5. If the nickname and password are correct the user logs in the system.
6. The “Login” option changes into “Log out”, which logged in user can use to log out from the system website.

Sequence diagram



2. **Use case <Search>**

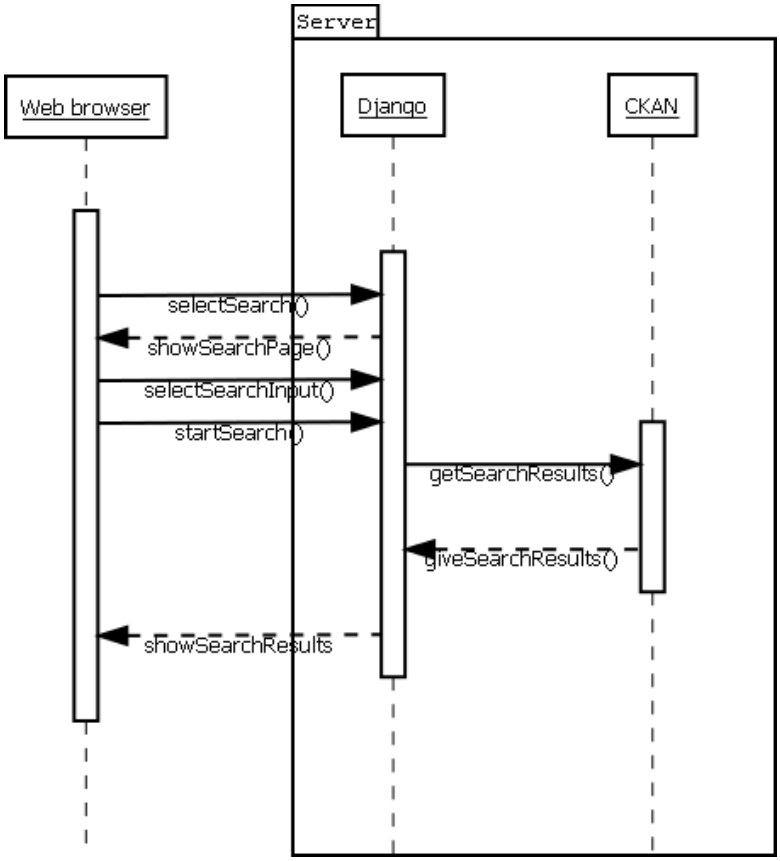
Brief description

User makes a certain type of search from system database by his/her inputs and choises.searches.

Flow of events

1. User will select the “Search”-option from the website.
2. User selects the certain type of search for three options.
   - Company
   - User
   - Job
3. User types the input data for the search depending from the search type.
4. User marks the special sections as negative or positive depending from the search type.
5. User selects “Start search”.
6. Search engine starts to find similar targets from server depending the search input and selections.
7. Search results page is opened and search results are arranged and shown by the most relevant targets according to inputs and selections. Sequence diagram

Sequence diagram



3. **Use case <Add coursework>**

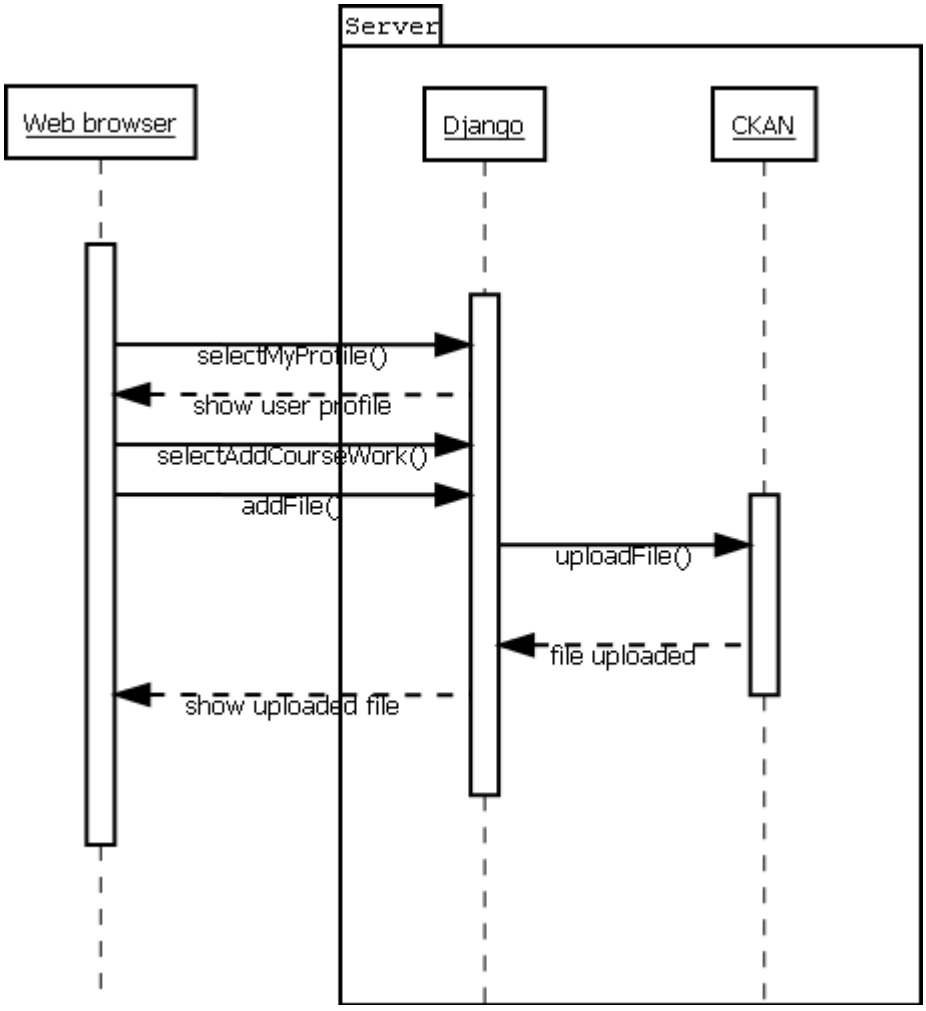
Brief description

Student user saves his/her accomplishments: code, reports, etc. to the system to recruiter and company users to see.

Flow of events

1. User will select the “My profile”-option from the website, which opens section containing information from account and user.
2. User selects the “Add coursework”-option, to add his/her works, documents and programs to be shown in his/her public profile page in the website.
3. User selects skill tags to be linked to the file. This will help to prove his/her certain skills used in to create this certain work.
4. The list uploaded files are shown with “Delete file”-option.
5. User select “Add file” to select the file to be transferred in the server database and to be shown in the portfolio section in the profile.
6. User selects the file from his/her computer to be uploaded in the system.
7. The selected file is uploaded to the system.
8. The uploaded file is shown in the user’s profile.

Sequence diagram



4. **Use case <Contact student>**

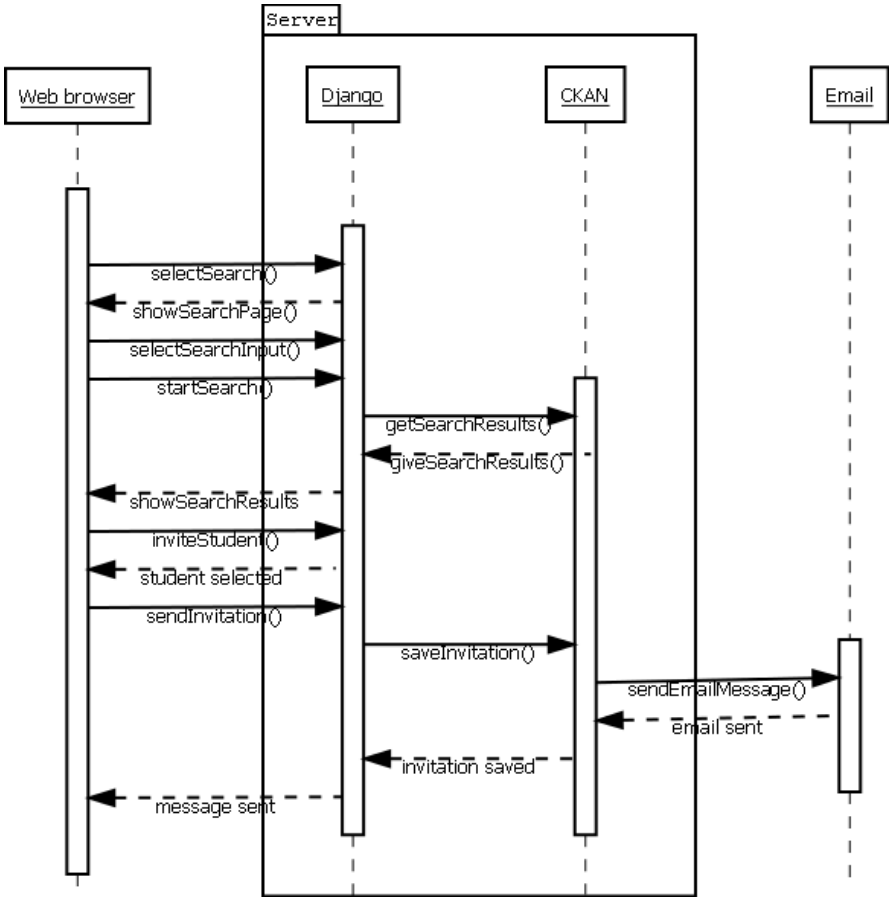
Brief description

User contacts another user selected from user search or his/her friend list. Contacted user gets informed by message from system and email.

Flow of events

1. User will select the “Search”-option from the website.
2. User selects the “Student” type of search from three options.
3. User types the input data for the search.
4. User marks the special sections as negative or positive.
5. User selects “Start search”.
6. Search engine starts to find similar users from server
7. Search results page is opened and search results are arranged and shown by the most relevant users according to inputs and selections.
8. The user selects the “Invite” to contact the student he/she likes. This opens message part.
9. User selects the job offer his/her company likes to give the student.
10. User writes message to message box to give additional information about the job offer.
11. User selects “Send message” to send the message to target user.
12. Student is informed by email about the contact.
13. Student can later respond to the invitation by selecting “I'm interested” or “No” and add message for explanation.

Sequence diagram



5. **Use case <Register>**

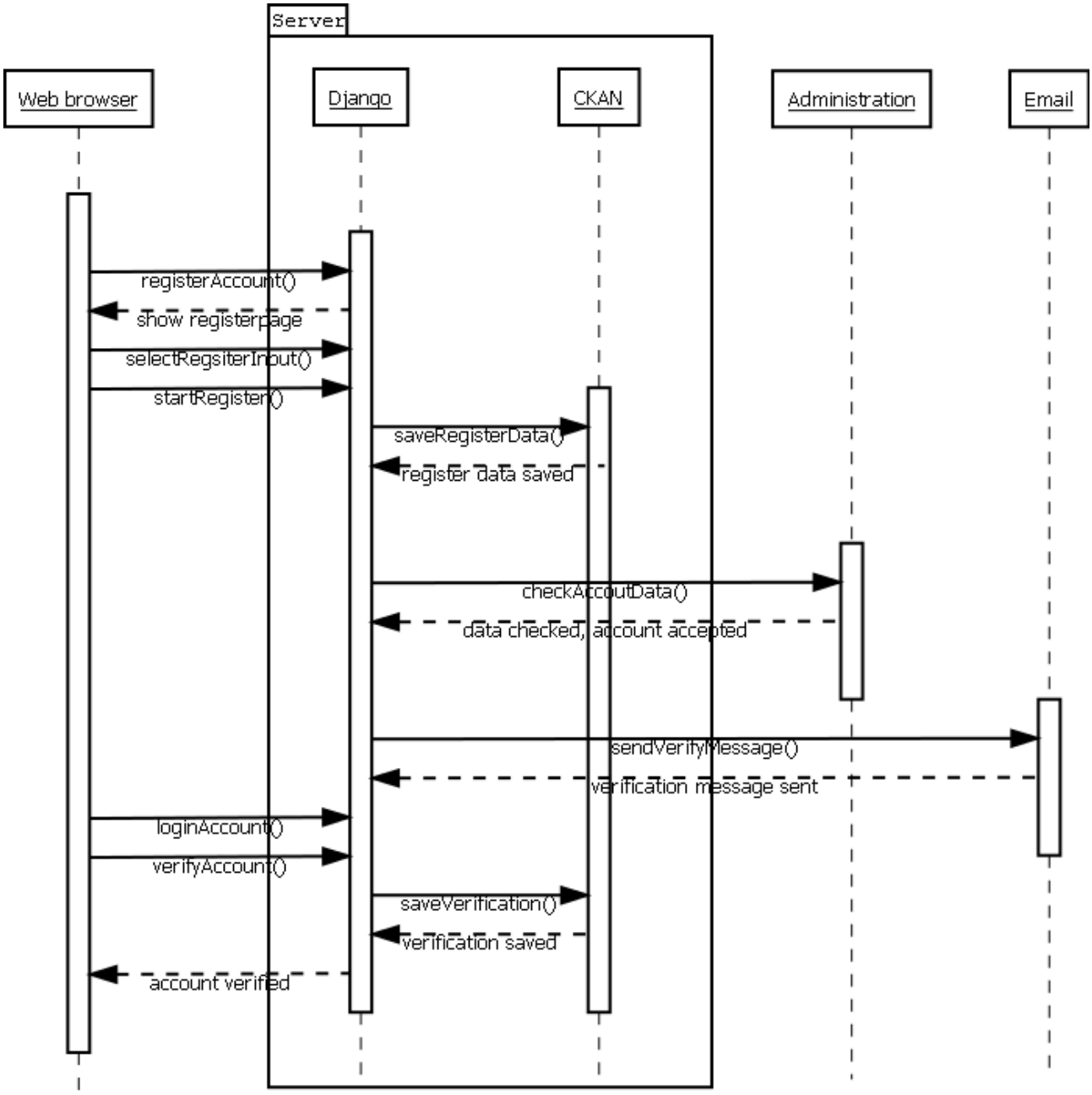
Brief description

User registers in the system, in which his/her information is checked by administration and new account is accepted or cancelled.

Flow of events

1 User will open the website and selects the “Register”-option from the website.
2 User writes all compulsory and optional information depending from his/her user type.
3 User types email account to be contacted by system moderator and administrator for the account verification.
4 User selects “Register account” to send the typed information and account to be verified.
5 Depending from the user type, the moderator or administrator verifies the new user account.
6 When the verification of the account is complete, message with activation code is sent to user’s email address.
7 User checks the email for the message with account activation code and message about verification of the account completed.
8 The user selects the “Login”-option from the website.
9 User types his/her nickname and password first time.
10 User types the account activation code.
11 User gets message that the registration of the account is fully completed.
12 The profile creation page opens to the user.

Sequence diagram



6. **Use case <Edit account>**

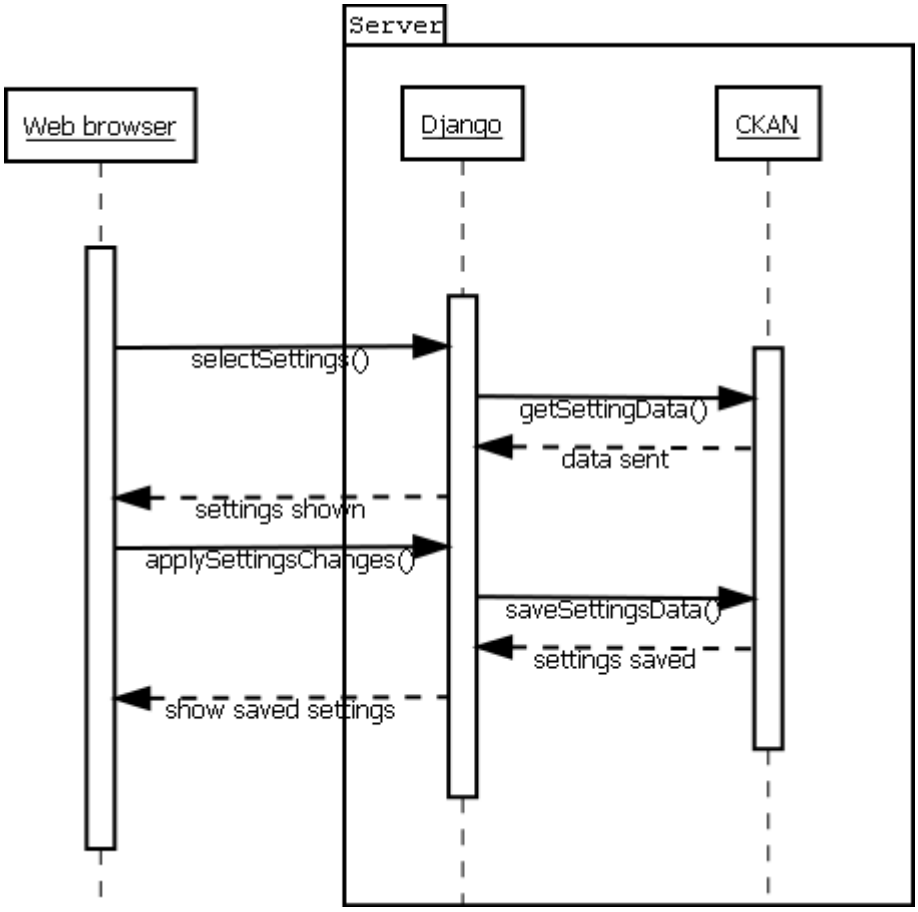
Brief description

User can edit his/her account's information.

Flow of events

1. User selects "Settings"-option from the website to open the settings of the user's account.
2. User will make changes to the sections in the to the settings page. These are for example to change the password or the email address for the user's account.
3. User will select "Apply changes" to apply the changes he/she did in the sections. Password-section has its own apply-button.
4. The changes in the account settings are saved in the system.
5. User gets message that changes have been accomplished.

Sequence diagram



7. **Use case <Create job offer>**

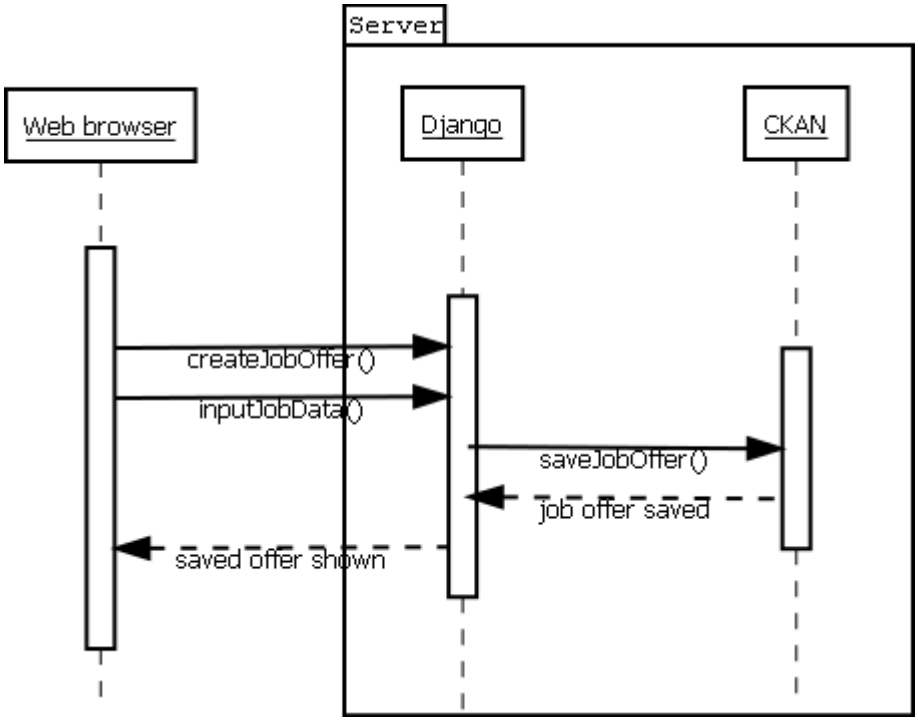
Brief description

Recruiter or company user creates a job offer to the system which students can apply.

Flow of events

1. User selects “Create job offer”-option from the front page of the website, which then opens page to make announcement of the company. Only recruiter- and company-type users have this option.
2. User selects the company, for which he/she makes the job announcement. Only those companies, which company user owns or recruiter user works for can be selected and only one company can be selected.
3. User types all the compulsory and optional description information about the job. For example name, job title and short description.
4. User selects the specific requirements for the job. For example certain skills and coding languages.
5. User will select “Post job offer” to send the job offer in the systems database. The job offer is shown in the job announcement list in the company’s profile page. Company user or recruiter can delete the job offer by selecting “Delete job offer”. The students have option to apply the job offer.
6. The job offer is sent and saved in the system and can be applied by students.

Sequence diagram



8. **Use case <Create profile>**

Brief description

User creates a profile for his/her account after the account has been verified.

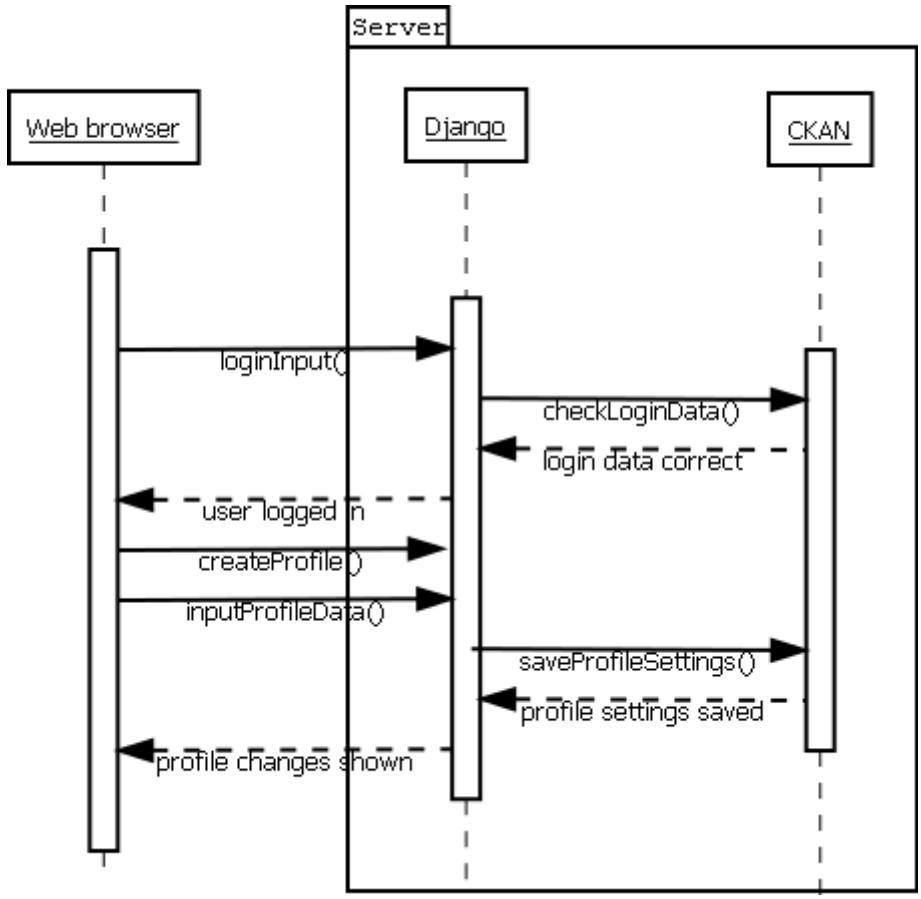
Flow of events

1. The system opens profile page of the user automatically, when the user logs in for the first time. Depending from the account type, every user has different type of profile page.

- Company
- Student
- Recruiter

2. Introduction of making the profile page starts. The introduction shows key sections of the profile page and gives short instructions to modify different parts. The type introduction depends from the user type and the instructions can be skipped.
3. User fills the sections in the profile page using instructions of the introduction.
4. User applies the changes made to the profile page by selecting “Apply changes” and selecting positive option in confirmation window after it.
5. The modifications to the users profile page are saved in the system.

Sequence diagram



9. **Use case <Add recruiter>**

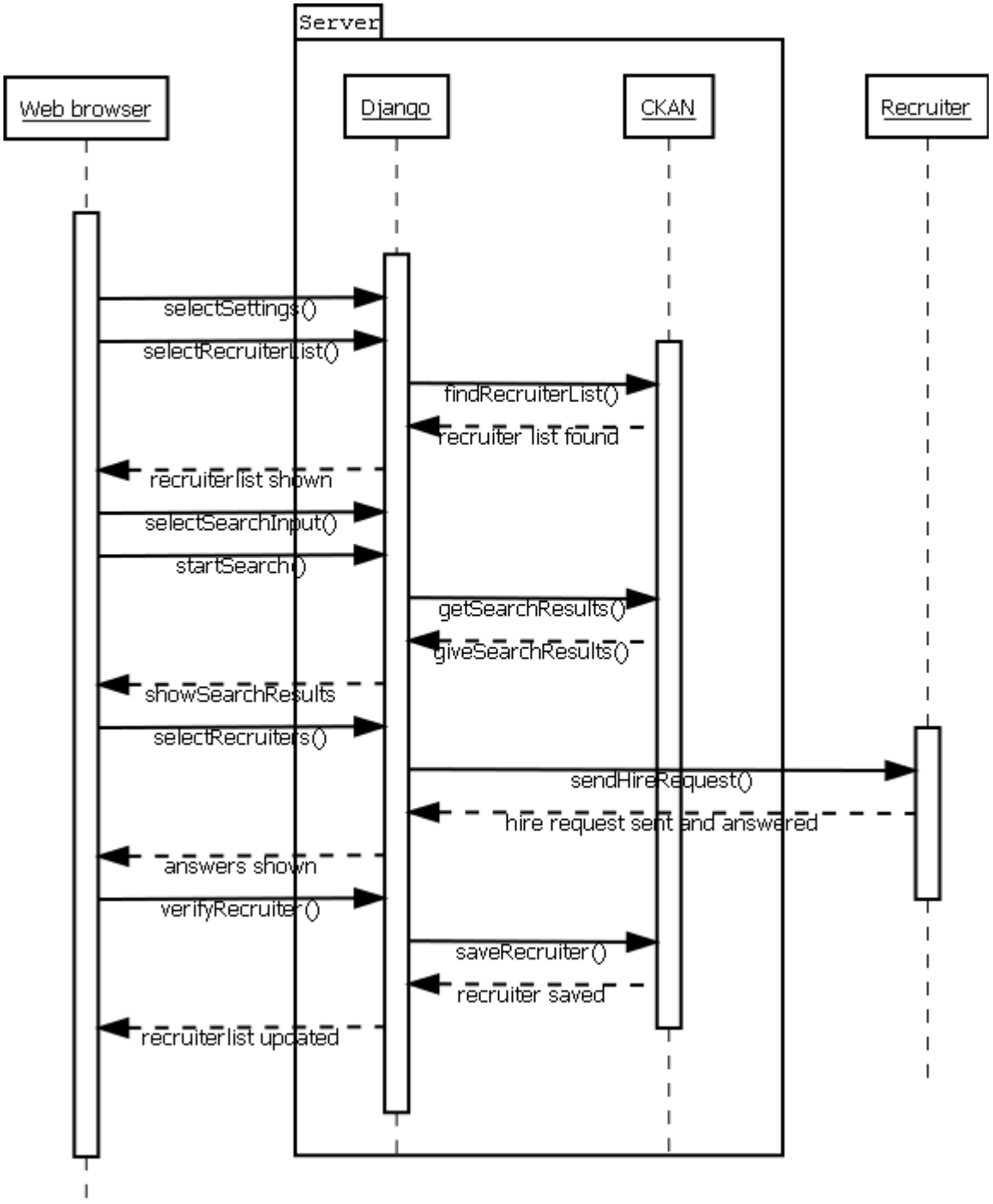
Brief description

Company user will search for suitable recruiter and selects him/her as company's recruiter in the system.

Flow of events

1. User opens the company's settings page by selecting "Settings"-option from the front page.
2. User opens list of company's recruiters by selecting the list called "Recruiters" from the "Settings"-page.
3. User selects "Add recruiter"-option from the "Recruiters"-list, which opens the search function in smaller window with settings to search recruiters already selected.
4. User will fill in the missing information sections of the search.
5. User starts the search by selecting "Start search".
6. Search engine starts to find similar targets from server depending the search input and selections.
7. Search results page is opened and search results are arranged and shown by the most relevant targets according to inputs and selections.
8. User highlights the most suitable recruiter(s) by selecting target recruiter(s) and "Hire selected"-option becomes available.
9. User selects "Hire selected" and hiring messages are sent to recruiters to be accepted.
10. Target recruiters accept the hiring messages and company user gets the messages of accepted offers.
11. Company user finalizes hiring by confirming the message sent from recruiters.
12. Target recruiter is saved in the company's recruiter list.

Sequence diagram



10. **Use case <Verification of the user>**

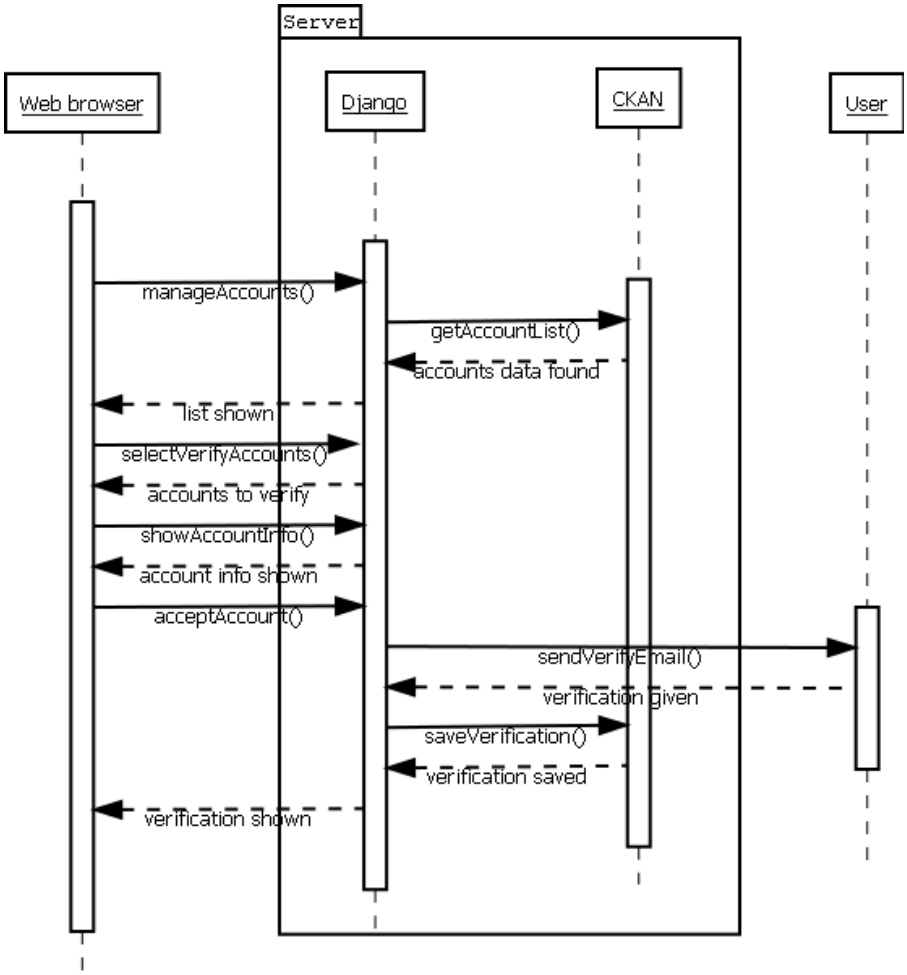
Brief description

Moderator or administrator checks and allows the new user's account to be verified.

Flow of events

1. User selects "Manage accounts"-option from the webpage. Object also shows how many account situations are waiting to be solved.
2. User selects the "Verify accounts" option, which brings up list of all user accounts which need to be verified.
3. User selects the target account, which brings up window of all the data written and selected options in the selected account's data.
4. User checks the information in the account and decides whether to accept or deny the verification of the new account.
5. User selects the "Verify account"-option and account activation code is sent to email address of the new user.
6. The new user uses his/her activation code and finalizes the registration process.

Sequence diagram



11. **Use case <Remove/Hide user>**

Brief description

Moderator or administrator hides or removes the account of target user.

Flow of events

1. User selects “Manage accounts”-option from the webpage. Object also shows how many account situations are waiting to be solved.
2. User selects the “Remove/Hide accounts” option, which brings up window for searching users.
3. User selects the parameters for the search and target account that must be removed or hidden.
4. User selects “Start search” to begin the search.
5. Search engine starts to find similar targets from server depending from the search input and selections.
6. Search results page is opened and search results are arranged and shown by the most relevant targets according to inputs and selections.

7. User selects target users to be removed or hidden. User can select several accounts to be removed of hidden. After selecting the accounts, the user must confirm his/her decision.
8. User confirms his/her decision and selected accounts are hidden and removed from the system.

Sequence diagram