Bilkent University

CS353 Group 1 - Oh No

Suit Up

Project Proposal

Group Members:

Ergün Batuhan Kaynak	21501178
Beyza Kalkanli	21600944
Kerem Yilmaz	21601223
Muhammed Salih Altun	21602314

March 4, 2019

Contents

1 Introduction

Suit Up is a website which offers a platform that connects people who are seeking for employment and the companies. The website enables people who were working in a company to post a review about that company. Also, companies can create posts for events. The aim of this project is to build a database system for Suit Up and design a user friendly website.

In the next section, an overview of the project and its features will be given under the project description title. This will be followed by Functional Requirements, which will be an in-depth explanation of the main functionalities of Suit Up. In the Nonfunctional Requirements section values which are authentication & security, privacy, usability, capacity, reliability, performance, concurrency, error-handling, scalability and pseudo requirements(constraints) will be addressed. Limitations will be given in order to underline the restrictions. Then the Entity/Relationship diagram of Suit Up will be given. The website that we will post our reports of the project will be given in the last section. The references can be found at the end of the report.

2 Project Description

Suit Up is a website that is supported by a database system and created in order to provide users an opportunity of easily getting in touched with companies and a place for companies to post job offers and events.

Suit Up includes five types of accounts in order to increase the usability of the platform by providing different interfaces to them: Super Admin, Admin, Editor, User and Office Account. The users can search for a job, apply to a job, write reviews about their experiences at their old jobs, subscribe the companies or the branch offices of the companies, create subscriptions for specific jobs at specific locations and configure their profile. The offices can post job offerings and send request to editors to delete a review about them using their office accounts. Also, the companies can post events and users can participates these events. Admins are going to be added only by super admin. They can add/remove editors and evaluate the delete request of comments that come from the offices. Super admin is the most authorized person. Super admin can manage any operation that is done by editors and admins. In addition to them super admin is responsible for the addition and removal of admins.

For the implementation of the project we will use MySQL for the database and Flask micro frame for building the web server.

3 Requirements

In this section, first, the functional requirements, then the non-functional requirements will be listed and explained.

3.1 Functional Requirements

Functional requirements will be discussed in the following subsections.

3.1.1 Account Management

All accounts other then super admin requires an approval process in order to ensure the consistency of the system. Users can browse through the website without logging in. However, they need to log in for specific functionalities such as writing a review for an office. The account types extend Account in order to avoid redundancy.

- **Super Admin**

  Creation of the super admin account will be managed directly from the code base using SQL queries. Super admin cannot be removed through admin panel.

- **Admin**

  There can be multiple number of admins. Admin accounts can only be created and removed by super admin accounts.

- **Editor**

  Both super admins and admins can create accounts for editors and remove them.

- **User**

  The user signs up to the system and an email for verification is being sent to the user in order to ensure the correctness of the email that the user used for the register. A user can be banned by an editor.

- **Office**

  Since the office is the branch of a company in a specific location, each office can only have one office account. The office account is being created after the completion of a process. An office send a request for an account and after the approval, their accounts are being created. The approval of the office account can be done by a super admin or an admin or an editor.

3.1.2 Account Functionalities

The functionalities for the five different account types are listed below.

- **Super Admin**

  We decided to create the super admin account type in order to be able to control the add/remove of the admins and the stability of the system. The developers of the project will have the super admin accounts and they have the privilege of doing operations that any authorized account can do.

- **Admin**

  Admin accounts are the second most privilege accounts after the super admins. We decided to include them to the system to prevent super admins from dealing with excessive add/remove operations.

- **Editor**

  Editor accounts are needed to manage the organization of the website. Editor accounts can evaluate reports against reviews and decide if they should indeed be removed. A review is removed if two editors approve the removal request. Editors can also ban users accounts.

- **User**

  A user can post reviews to companies, apply to jobs offered by offices and participate in events hosted by offices. Users can also make three types of subscriptions. A subscription can be made for a company to get notified if that company has any new job offers or events, regardless of which office of the company posts them. Second type of subscription can be made for an office of a company to get notified if that company has any new job offers or events posted by that office only. The last kind of subscription can be made to a location to get notified when any office of any company has new job offers or events in that location.

- **Office**

  Office accounts are special accounts that are verified by Suit Up to be representatives of company offices. These accounts are used to post new job offers and events to Suit Up. These accounts can also report reviews made against their offices.

3.1.3 Review Removal Requests

A representative of a company can report a review to ask for its removal. Editors check reviews and take action if they find the review to contain sensitive information or offensive language.

3.1.4 Office

Each company can have different offices in various locations. Suit Up allows companies to have office accounts for offices and these accounts act as representatives. Users of these accounts can post job offers, host events and report reviews made against the company.

3.1.5 Event

Events are hosted by company offices and posted to Suit Up by office representatives. Users can participate in these events and show their participation on their profiles.

3.1.6 Job

There are two kinds of jobs in the system. One of them is a job being offered by an office. A job that is offered by a company can be applied by users as long as it is available. Total number of users that have applied to a job is also displayed. The other role of jobs in the system is being listed as past or current employment of a user.

3.1.7 Review

Users can make reviews on offices. Users can describe their work and interview experiences in text, select predefined pros-cons (positives-negatives) about the office to add to their reviews, give ratings to 5 predefined aspects (e.g. work environment) on a scale from 1 to 5. Also, they can explicitly specify their position, job type (e.g. full-time), starting date, and ending date along with their salaries. Other users can mark the review as helpful and increase the review popularity, so that it shows higher up the list when users sort it with respect to popularity.

3.2 Nonfunctional Requirements

The system's aimed non-functional requirements are written below.

3.2.1 Authentication & Security

- Users must authenticate themselves before making reviews to companies or applying to jobs in the system.
- No user should be able to edit information or reviews of other users.
- User login data should be hashed to make sure they are safe even in an event of security breach in the server.

3.2.2 Privacy

- Review removal requests are not shown publicly, only accounts related with their evaluation can see them. (Editors, admins and super admins).
- Users can post anonymous reviews if they wish to not disclose their identities in their reviews.

3.2.3 Usability

- Users should be able to navigate through the clients without having problems on how to use it, the process should seem natural without a high learning curve.
- There should be appropriate description to each action so that the user can clearly understand its functionality and consequences.

3.2.4 Capacity

- The database must have sufficient storage to handle at least 10.000 registered users and 10.000 reviews.

3.2.5 Reliability

- System uptime should be 99.9% to not cause any inconvenience to the user.
- Database anomalies should be handled concisely not to cause any conflicts.

3.2.6 Performance

- Each query must not take more than 2.500 milliseconds to return its result.
- Results of any human interaction (other modules) such as clicks or selections should not take more than 1 second to provide good user experience.

3.2.7 Concurrency

- The server should be able to handle requests in an either a concurrent or parallel manner. Whenever possible, the users should not have to wait for other requests to be handled in a first in first out type of sequential manner.

3.2.8 Error-Handling

- Errors must be recognized and accounted for as much as possible. The user should see an error message that clearly and concisely explains what went wrong to understand what the problem is without being presented with too much needless info, with steps to take to solve the problem, if the nature of the problem permits.
- Any unforeseen errors should be presented to the user with an appropriate message to the user, with an option to the user to report the problem to the developers.

3.2.9 Scalability

- We will be using Amazon's elastic compute cloud (EC2) instance to deploy our project. If it is needed in the future we will use Kubernetes which is a container orchestration application to setup a load balancer which will divide the incoming load by launching additional EC2 instances and deploying Docker containers. This will allow us to handle scaling loads easily.[1][2][3]

3.2.10 Pseudo Requirements(Constraints)

- For the implementation of the database MySQL will be used.
- A user friendly website will be designed.

4 Limitations

The system limitations are listed as follows.

- Users can only post one review to each company for one position and one job type.
- User can list minimum of one and a maximum of three positive and negative aspects (pros and cons) for each of them while writing a review.
- A job offer of a company needs to have a duration of three days at minimum.
- A review can only be removed if two editors approve the removal request.
- Users must provide one of the following in addition to job position and type
  - company
  - office
  - location

  information when subscribing.
- When subscribing based on location, users must provide either a city or a country.
- Each event must be hosted by a company.
- For online events location attributes will be treated as a link to the related page.
- Each office can have exactly one office account associated with it.
- Each review must be related with exactly one office only.
- Each report must have a single review associated with it.
- Each user can have at most one current employer.

5 Entity/Relationship Diagram

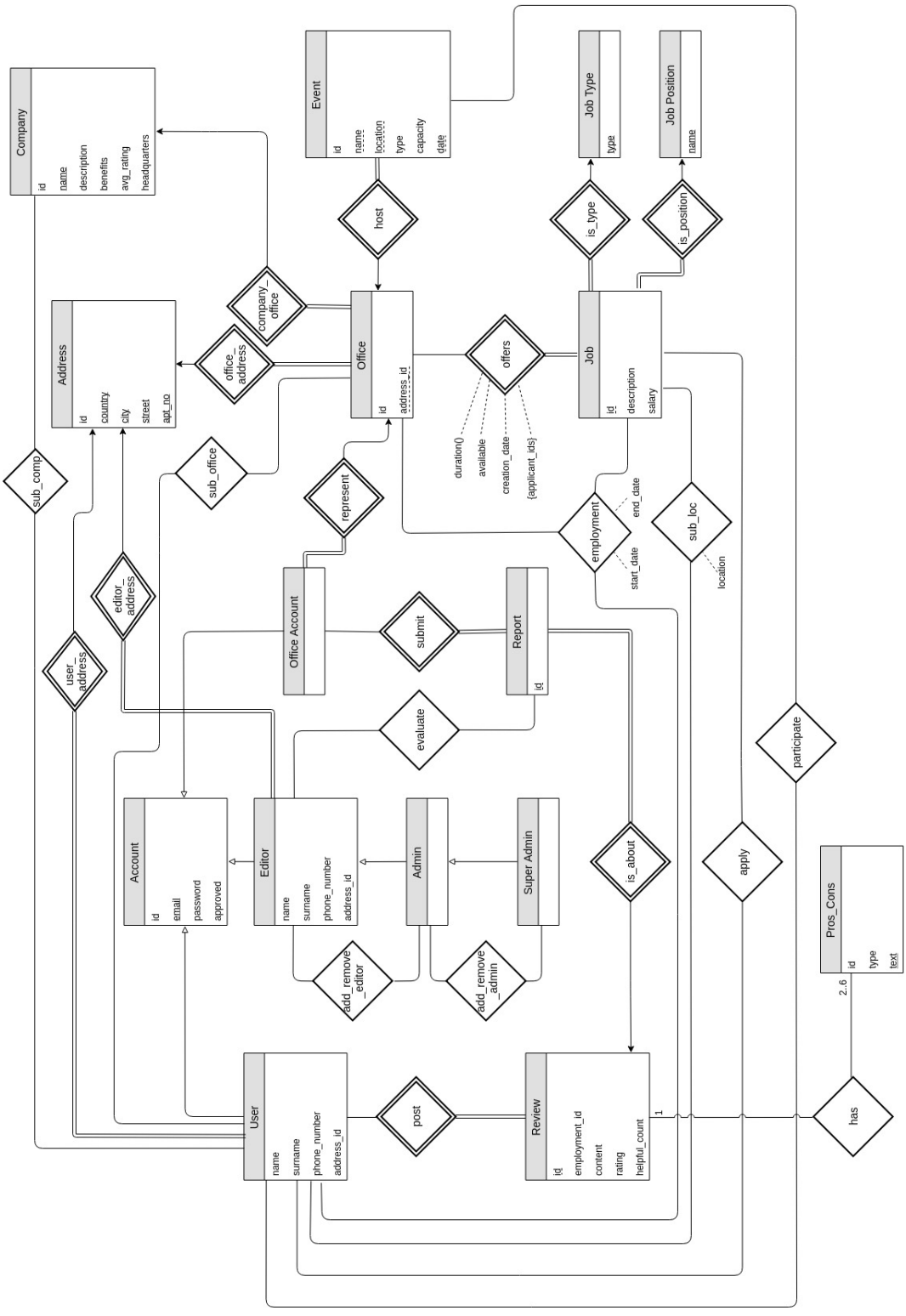

Figure 1: Entity Relationship diagram for Suit Up [4][5]

6 Website

Our website is: https://mrkeremyilmaz.github.io/suitup/

References

[1] "What is amazon ec2?" [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

[2] "Production-grade container orchestration." [Online]. Available: https://kubernetes.io/

[3] "Enterprise application container platform." [Online]. Available: https://www.docker.com/

[4] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database system concepts*, 6th ed. New York: McGraw-Hill, 2010. [Online]. Available: http://www.db-book.com/

[5] R. Elmasri and S. Navathe, *Fundamentals of database systems.* Addison-Wesley Reading, Mass, 2000.