**Problem Statement**

Students need a platform to connect with startups for part time work during their studies to fund them. Startups need smart creative employees who are willing to work at a cheap rate.

**Requirements**

**Phase I**

studentsmeetstartups.com is a website that connects students (with or without experience) with local startups for part time work and helps startup businesses to meet with students who are informed on cutting edge technologies.

A student has a name, age, short personal description, an address, personal website (optional), email, telephone number, academic record, resume and a user account. In order to apply for a meetup, a student must be registered to Students Meet Startups. An academic record contains a transcript pdf document.

A startup has a name, description, address, telephone number, fax, website (optional), email and a user account. In order to post meetups, an employer must be a registered member.

An address has a street number, apartment number (optional), street name, postal code, province and municipality.

A meetup has a start time, end time, address(optional), an employer and participants (students). A meetup has startup representants reviews and participants (students) reviews. A meetup review has a rating, a comment.

**\*Phase II**

Meetups can be physical or virtual ( a Skype meetup for example).

An academic record contains all the academic information about a student: CGPA, GPA per term, course followed/completed /failed/incomplete (with grades), and expected graduation date.

A startup has employees. Employees can be of type x and type y. Each type has different access modifiers.

Meetups can be of type “invite only” or public. Invite only meetups can be viewed by invited students only, public ones can be viewed by every student member. A student meetup review has a “hired” status icon that shows if a student was hired or not. A startup review has a “hired someone” status icon that shows if the startup hired this student at this particular meetup.

**\*Phase III**

The system will have a matching algorithm for particular startups and student skills.

\*Phase II and III will only be implemented if we have time inside the time constraints of the project.

**Architecture & Technologies**

Fat-client architecture is going to be used for the system. Javascript will be used accross the stack. The client side computations are going to be done with KnockoutJS and jQuery. On the server side we will use ExpressJS(NodeJS). MySQL is going to be used as persistent storage solution.

We will use Git as version control system and GitHub for hosting the repository.

Client-server will always communicate back and forth using HTTP protocol in a RESTful way. Types of objects sent and received by both server and client will almost all be in JSON format.

We’re going to use Bootstrap 3 as our responsive CSS/HTML/JS framework to reduce the testing effort to make our product work on different types of devices.

**Use Case Stories**

1. As a student, I would like to create a profile my general information, transcript and resume in order to meet with startups.

2. As a startup employee, I would like to create a profile with general information about my business in order to meet potential students.

3. As an startup employee I would like to schedule a meetup in order to set up a place and time to meet students.

4. As a student I would like to enroll to a meetup in order to meet a startup representant.

5. As a non-member user of the website, I would like to consult the startups profiles in order to explore studentsmeetstartups.com.

6. As a student I would like to modify my profile in order to display more up to date information about me.

7. As a startup employee I would like to modify my profile in order to display more up to date information about my business.

**Server-Client Message Examples**

|  |  |  |  |
| --- | --- | --- | --- |
| **Message example # (matches use case # )** | **Request** | **Response** | **Description** |
| 1 | POST /student  (html form values as a JSON object) | \* | Student signs up. |
| 2 | POST /startup  (html form values as a JSON object) | \* | Startup signs up. |
| 3 | POST /meetup  (html form values as a JSON object) | \* | Startup schedules a meetup. |
| 4 | PUT /meetup/:id (JSON object containing student unique account id) | \* | Student enrols in a meetup. |
| 5 | GET /startup/:id | \* | Non-member user want to check out startups profiles. |
| 6 | PUT /student/:id (html form value as a JSON object) | \* | Student modifies his profile. |
| 7 | PUT /startup/:id (html form value as a JSON object) | \* | Startup modifies his profile. |

\* JSON object containing a confirmation or an error message.