Smallee Project

Introduction

- The smallee project represents an opening to the future students and day-to-day coders. This web app i made to offer them the opportunity to present their creations and ideas to a larger public and have access to a large network of people sharing the same interest for software design and development.

- The goal of smallee is to propose an interface of exchange and business. This means proposals, sharing, crowd funding and company sponsoring. A sort of reunion of all the actors and potential actors of the field. Smallee does what the other ones don't: create an environment where every member has a place to talk, to share, to learn and cooperate with any other group.

- Smallee ideal is a community of people ready to jump onto new adventures and day after day, realize and conceptualize tools for a better world. Let's see Smallee as the Meetic of the software development. This place is not a market but a "college", where each one can learn and grow, propose and innovate, link and even find a job. The idea is to make all this possible within one place, this place, the best place, Smallee.

Conceptual details

- The smallee app is a web app that will be based on Django for easy development and will integrate the features of the framework Bootstrap. The application will be analyzed under different aspects going from its components to the user interactions.

- Lets be a user. The user has a number of features that are proposed to him. Most of them are free. The user first interaction is the creation of his account. The site must reach college students. Therefore students must enter their personal email and also their student email trough which they will do the second part of the verification process, the first being done on their personal email address.

- What are the informations that we should keep concerning the user? The user information that will be kept are: First name, Last name, birth date (Restrictions will be applied on people less than 16 years old), personal email, student email, phone number, Address, city, country, username (Checked for obscene language), password, linked account (optional), links to realizations (optional), profile picture (optional). This was the definition for a simple user. Organization have their own interface that will be discussed further.

- What are the interactions that will be proposed to the user? The user at first will have to select a set of interest. These are tags representing the different most favored subdomains on the platform. This is starting point of the conversation between the user and the different users populating the interface. The user will have a main page on which the different users and organization to which he is connected can share publication, pictures, events, projects, team membership proposals, applications demo, application release announcements, open source updates, tournaments, videos, lives and articles. The elements listed are the different type of content that will be shared by users and companies on the platform. Their details will be talked further in this document. The user can make an article, a publication, share a picture, a video or a live. It can also launch an open-source project, call for a team membership proposition, publish a demo app, publish a finalized app, He can also make researches with the search engine and find the information following a tag, an user name, a group name, a team name, an organization name or an application name. The search will be done aside of the workflow of the user to give him visibility on the page he was using.

Architectures details

- The different building blocks of this application have each specifications that have to be clearly defined before implementation. The first block is the publication. a publication is a text with particular texture, pattern or font that is limited in size to 256 words. The content will be analyzed to keep away from obscene or abusive language. The publications can have their visibility restrained on following the user wishes. The periodicity of the publication can also be defined with 1h, 2h, 5h, 8h, 12h and 24h. Otherwise the publication is permanent. People can be identified onto publication using their username. The publications can also be set to be published later on. The user then has the pleasure to set the date and the time at which he wants the publication to be performed automatically. The publication can be deleted or modified after creation. Pending publications (waiting to be published automatically) and unsaved publications are saved in drafts.
- The second building block is made of the videos and pictures. Pictures and video are based on the same architecture as the publication and can be accompanied with text. Video and text can also be grouped into albums. And these albums are treated the same way as the individual video and images. These albums can be modified. This means that other pictures and video can be added to the album. The illustration text can also be changed.

- The third building block is the projects. Due to copyright infringement policies and the risk of intellectual property fraud, we will assume all projects ideas written, proposed and shared on this site as open-source projects. This means that as soon as your idea is proposed in the platform, it will be registered as open source idea with your name as project starter. once done all the person having access to this project definition or working under this project close will have to refer to you and their name will be assigned to the list of "threads" linked to the project.The project proposed will have to be defined a name and its repository will immediately be created in a git hub repository having his name and your account as project owner. People wanting to work on the project will then have to refer to this repository and will be defined as threads in the Smallee platform. Each push on the Github repository will result in a project update in Smallee. It will be notified to each of the "threads" and result in an automatic publication on the platform visible to each connected thread. Particular definition on the project has to be made. Each project definition will be submitted to our project description control that will check the project name and description and search for similar projects. It will then show similar projects along with a percentage or similarity. The projects that will be listed as similar will have a similarity percentage of more than 35% with the proposed project, if a similarity of more than 65% is found within two projects, the project starter will be proposed the option to join the project already defined or start its own project. The second option will result in a notification of project start to the owners of more that 65% similarity projects. These ones then have to review your project and submit an approval. If they a 2/3 majority accept your project proposition, you will be able to launch it. If more than 80% similarity is found, the user no more has the option to launch its own project and will be redirected to the most similar project with a threading option. Each time an 80% similarity is detected, a notification is sent to the project owners, this notification is anonymous, but gives to the user a sight of the project description newly defined that have peculiar similarities with its own. Concerning the companies that read the project definition, each time one of them reads it, a notification is sent to the project owner and the company is notified she joined a property group that restrain it from implementing the idea proposed by the user. If this agreement is not respected, the company can be conducted to justice by the project starter. The complete features list will be discussed in the technical design document.

- The fourth building block is the application. Applications are verified and trusted executables that can be downloaded trough the platform and run on the predefined device. If they are not directly downloadable trough the platform, a link can be submitted. Each app and link will be priorly verified by a team of reviewing. If the application or link is judged not viable or risky, the app submission will be rejected and the app submitter will be notified of the detailed reasons of the rejection. If the application is judged to have a really great level of risk (virus, malware, phishing, worm, trojan, ransomware, etc…), the user proposing the application will be sanctioned by temporarily removing his right to publish anything on the platform. If the unfaithful action continue, the user will be definitively banned from the platform and share to black-list with the companies present on the platform. An app is joined, if not preceded by an app demo; it is the lightweight minimalistic version of the application that presents the most basic and fundamental features of the application. It is the version that can be presented to a tech investment group or for a prototype demonstration. Our intent is to propose multiple emulators that will run the demo applications. These emulators will be included in further versions of the platform. An application can be set for free download, paid download (including percentage that will go to the company), or crowdfunded for large distribution (including a percentage for the company). Open source applications will both include a link to the executable file and the link to the Github directory. The crowdfunding and selling options won't be available for open source projects. Applications classified as open source will be applied a version control that will allow to keep track of the different versions and updates released. Open source application and demo applications will propose an application release announcement that will be published in the feed.

- The fifth building block is the team membership proposal. A user can create a team of coder in one click and then send proposal trough the feed. Users can be invited to the team and other will go through the team membership proposal. Those ones will then be presented to the proposal launcher that will review them and select those who show a better profile for the task. Once the team proposal is launched, a Slack team will be created and within that Slack connection will be crafted the channel #Smallee. This is the channel that will contain the team interactions that are made on the Smallee platform. This means that a group interaction chat will be created directly on Smallee and will be duplicated on the channel #Smallee of the newly created Slack team. Each new member and will be asked a slack profile. If it does not have a Slack account he will be asked to create one. Without a Slack account the user cannot join the team. You can see that Github and Slack APIs will be greatly used within this web app. Team members can interact within a private feed with publication, video, pictures and articles. A member of the team can leave the team but will have to make a justification that will be notified to the other members of the team; a team member can be directly kicked out of the team by the founder of the team but other members of the group wishing the removal of a team member must go through a justification and a simple majority will result in the exclusion of the exclusion of that member. Excluded members will have a their team membership status updated with an “Excluded” red tag aside from the name of that team. As user can define projects, teams can also present projects and applications release. These projects and applications will be set as creation of each team member.

- The sixth building block is article. An article is a well structured document that contains text, images, code, videos, links and other features. Articles will be implemented using a particular WYSIWYG. Articles can include users references. This means that a content of something said in the article represents something belonging to the intellectual property of another user. Therefore including the user reference can be a good way to show the attention to the work of everyone around the platform. Articles can be modified or deleted. Their publication time and date can also be defined. Each article images, video and images will be analyzed for obscene or abusive content.
- The seventh building block is the lives. Live transmissions will not be implemented in the core of the application. Each user who launch a live will be provided a Youtube live Stream. The link will then be share on the feed and your followers will receive notification that you started a live diffusion. Diffusions will be kept in memory and progressively reviewed for content verification. Abusive or obscene content live streaming will result in sanctions concerning the streamer like temporary rejection of the rights to start live transmission. In case of recidivism, the user will be definitely banned.
- The eight block is the one concerning the search engine. The search engine will divide the results in 5 groups: users, teams, organizations, groups, applications (projects). Each time the user will launch a new search command, the search engine will launch threads that will look for the different informations that can be found in each of the categories and present them to the user. Each research made by each user will be kept anonymously to see the trends in the researches of the users along with their age, college, city and other activity made on the platform.
- The ninth building block is the agenda. Each user dispose of an agenda and this one will be organized depending of the companies that person follows. The agenda will then keep the different events proposed by the different companies followed by the user and tournaments to which the user registered. A simple algorithm will provide the possibility to link it our Android or IOS calendar.

- Now that we are done defining the building blocks linked to the simple users, we can define the building blocks that are linked to the organizations, second type of users that are present on the platform. The tenth building block is the events announcements, These are official announcements made by companies and which are placed in a particular place in their calendar of programs. The users who follow that company will have their agenda automatically updated. Events can be modified or deleted. Events provide day, time, locations and description. It can also precise users that are present on the platform and that will be leading the event.

- The eleventh building block is the job proposal. Organizations have the capacity to set job proposals that will be visible in the feed of all the persons following them and also be notified to them. Job proposal contain location of the job, job name, job description, job salary, job contract type, and the link to the job postulation. The platform does not directly manage the job postulation. That is why we have it done in the link proposed by the organization.

- The twelfth building block is the tournaments. Some companies can also propose competitions that they organize. The competitions are pretty simple to define. Each competition has as parameters the day, the time, the duration, the participants type, the number of members per team if they have to be teams, the different prizes, the title of the competition, the description of the tournament, the link to the competition registration and the tournament officials if they are users in the community. The tournaments are directly notified to the person following the company proposing the event. The competition parameters can be modified or the competition can simply be cancelled. The reason of the cancellation have to be defined. The notification of cancellation will be sent to each user connected to it. On the other hand organizations can do all what the users can do except on the part concerning the applications, projects, groups and teams. Those options are not offered to companies.

- The thirteenth building block is the private projects. Those projects have the same attributes as the other projects but reveals to be more protected than the other ones. In effect, when a user propose a new private project, He has no restriction upon the public projects but is compared to the other private projects. Any similarity close to 50% is automatically notified to all the other users having the 50% similarity and the project is simply rejected. The private projects can be crowdfunded. This means that a community of people can fund the project up to a funding goal that will enable the the project to be completed. The funding goal should then be presented, along with the different use of the funding money that will be made by the project owner, and the different gifts given to the most generous users. When the project is accepted and reviewed by our team (for security purposes and project establishment), the project is created in GitHub as a private repository, a Slack group will also be provided. Creating a private project comes then with some fees. The GitHub private repository comes with a monthly fee; The user will have to pay the first month and Smallee fee after the project be validated and simply keep with the GitHub fee as long as the project goes. The payment can be done through the Smallee platform. The private projects GitHub links are not visible to any user. Only the creator of the project can see them. He will be provided a link that he will send to the members of the team he will create for the project. That link will generate a code that they will use to connect to that project. A code can only be used by a particular user for a particular project. This will avoid project leaking. Ads, private projects, crowd funding, and donations will be the main sources of revenue for the Smallee platform

System requirements

- The website will be based on Django + Bootstrap + PostgreSQL architecture to ensure robustness and development efficiency. The Django architecture is put in place to propose an easy evolution in the project entities. The project entities will be multiple and we really need to be able to produce them quickly and efficiently going through a class model definition.
- The website will run on a server that will be remotely managed via an SSH port connection. This SSH connection will be discussed further. The website must run on all the different navigators without any latency or modification of the user interface base. That is why we use Bootstrap as user interface generation motor for consistency and proficiency.
- The system has to work 24h/24 and we have to define a very efficient monitoring base for the website. The site will have two server from the beginning, a live server and a 12H backup server. The implementation of this server will be discussed in the technical design document. The monitoring part of the architecture will be based on important statistic informations useful to the development and improvement of the site. An help and support architecture must also be put in place for the satisfaction of the different users.