

# Project “SocialDevelop”

Version 2.0

## Preamble

---

The course projects are inspired by real-world needs, and they usually refer to similar sites already on the web. Students must follow the specifications given by this document, but they can also refine them through an interaction with the teacher and the analysis of similar websites. The final website must be completely original, well organized and easily accessible to all the users.

*Thanks to Konstantinos, Svetoslav, Petyo and Danail for suggesting me this idea ☺*

## Site Specifications

---

*SocialDevelop* is a web support tool for collaborative software development. While other sites like GitHub focus on the actual code writing and versioning, our system stays on a “meta” level, namely its aim is to publicize the software projects and identify suitable freelance developers to employ, automatically creating their “references”.

The site is similar to a regular forum, but instead of discussion threads, SocialDevelop contains *projects*.

Each project has a *coordinator* and is associated with a set of *tasks*, i.e., activities to accomplish. Each task is accompanied by a textual *description*, the required *number of collaborators* (i.e., people to employ on the task), and the *time frame* for its completion.

Moreover, each task also has a *type* (very general, and chosen from a predefined list, such as “development”, “graphics”, “documentation”, etc.) and a list of *skills* it requires, each one associated with a *level of expertise* (from 1 to 10, for example, “C++ programming” level 8, or “Responsive design” level 5, etc.). These skills are also chosen from a predefined list, specific to the task type.

Finally, a task can be *open* or *closed*. A closed task does not accept further collaborators, unless they are added directly by the project coordinator. On the other hand, developers, as described later, can send applications requests for open tasks. A task is considered closed if it already has the required number of collaborators or if the coordinator has explicitly closed it. Otherwise, it is open.

Messages (announcements, requests, and internal discussions) can be posted within projects, and they can be marked as public, i.e., visible to any user, or private, i.e., visible only to the project collaborators.

The site has an administrator who is only responsible of entering/editing the list of task types and the list of skills associated with each task type (*optionally*, you may associate the same skill to more than one task type). Skills may *optionally* have a hierarchical structure (for example, “Design Swing Interfaces” could be a “son” of “Java Programming”) to organize them better. A skill, as already described, can be generic or very specific, and can be associated with tasks and with user profiles, as we will see.

The users of our site, that we shall call *developers*, must register themselves by providing their personal and contact information and their curriculum (as plain text or uploaded as a PDF file). They should also indicate their skills, chosen among the global list of skills available in the system, specifying for each of them the achieved level of expertise (through the same scale used for the skills associated with tasks, as specified above). For example, a developer could declare a level-7 skill in “Java Programming”.

The following list contains a brief description of the contents and functionalities required by this site. Obviously, any further refinement or enrichment of these specifications will increase the value of the project.

- All the users, including the *anonymous* ones, can browse the list of projects, see the related tasks (both as a compact list and in a detailed view) and read the public messages posted in the projects.
- All the users, including the anonymous ones, can search on *SocialDevelop* in two ways:
  1. Search for developers: the user indicates a set of skills (taken from the global list) and optionally a minimum level of expertise for each of them. The system will then return a list of developer profiles having all of those skills, with a level of expertise greater than or equal to the given threshold, if specified;
  2. Search for projects: the user enters some keywords, and the system returns a list of projects having these words in their name or description.
- Anonymous users can register on *SocialDevelop* to become registered users, i.e., developers. During the registration process, developers must provide their personal and contact details, their curriculum and their skills, as already described in this specification.
- The public profile of a developer shows, in addition to the above-described information, a list of *automatic references* generated from the projects he/she has worked or is working on. In particular, such references show the project name, the specific task(s) the developer was involved in, and an evaluation (when the task is complete) given by the project coordinator, whose name and email address are also included.
- Developers enrolled in a project can read all messages, both public and private, posted in the project, and post new messages.
- Every developer can create new projects, becoming their coordinator, and populate them with tasks. The definition of projects, tasks and skill requirements must follow the structure already illustrated in this specification. The coordinator should also be able, at any time, to modify the project, for example changing the dates or the skills associated with a task, increasing the number of collaborators required, etc.
- The project coordinator, once a task is fully defined, should be able to view a list of developers suitable for that task (that is, with suitable skills), automatically generated by appropriately using the search function (1) described above.
- The project coordinator can send to any registered developer a *collaboration proposal* for a particular task. Such proposal will be then included in the "*invitations panel*" of the coordinator that reports, for each proposal, the associated task, the contacted the developer, the proposal submission date and its status ( "waiting", "accepted" or "rejected"). The coordinator can cancel a proposal at any time.
- Each developer has a "*proposals panel*" that lists all the collaboration proposals sent by project coordinators as just described. The developer should be able to see the details of the associated project and task, and then accept or decline the proposal. The proposals not considered (i.e., accepted or declined) will be automatically declined after a certain period. If a developer accepts a proposal the system, after changing the status of the proposal in the "*invitations panel*" of the coordinator and in the "*proposals panel*" of the developer, automatically enrolls him/her in the project and assigned task and decreases accordingly the number of collaborators required for that task.
- Developers also have an "*offers panel*" that shows an automatic selection of the collaboration opportunities suitable for their skills. In other words, the system will search and show automatically (using a search system similar to the (1)) the *open tasks* having skills compatible with those of the developer. Developers should be able to send *application requests* for these offers to the corresponding project coordinator. In this case, the offer status in the panel will change from

"available" to "applied". According to the coordinator's decision, the status will then become "accepted" or "rejected". An application request can be cancelled at any time.

- The project coordinator has an "*applications panel*" that lists all the application requests received, with the indication of the developer and task. From here, the coordinator can browse to view the developer profile, and then accept or reject the application. If accepted, the application status will change both in the "*applications panel*" of the coordinator and in the "*offers panel*" of the developer, who will be automatically enrolled in project/task, as already described.
- *Optionally*, every status update in the "*invitations*" "*offers*" and "*applications*" panels just described may generate a notification email sent to the corresponding user. For example, a coordinator will be notified when one of his/her offers is accepted or rejected by a developer or when a developer applies for one of his/her projects, while a developer will be notified of the results of his/her application requests.
- The project coordinator may, at any time, exclude a developer from a task (e.g., if he/she is not working as required), de-enrolling him/her and eventually reopening the corresponding task to new applications requests (i.e., increasing by one the number of collaborators required by the task).
- At the end of a task, the project coordinator has the opportunity to evaluate (on a scale 1-5) all the developers that collaborated to it. This evaluation, as explained above, will then appear in the developer profile.

## Technologies

---

- The basic structure of the site must be created using HTML5 (*with XML or Polyglot syntax*). The validation of all the site pages with respect to the chosen HTML flavour is an important part of the development and **must** be reported in the documentation.
- The site layout must be realised using CCS style sheets. The layout can be freely based on third party layouts available on the web or shown during the lectures. In this case, the degree of **customization** of the layout will be taken into account in the final project assessment. **A responsive layout is not required, but strongly suggested.**
- For the *client-side* programming, JavaScript is the **required** language. It is possible to include libraries developed by third parties, provided that they have suitable cross-browser portability and that they are described in the project documentation. However, **any abuse of these technologies is not recommended**, especially when they can be replaced with a suitable use of HTML, CSS, etc. **In general, your site should be functional also with JavaScript disabled.** The site without scripts may be less “friendly” or allow the access to “core” functionalities only, but sites whose dynamics is entirely script-based are not allowed. However, scripts can play a more important a role in the functionalities whose users are restricted and pre-determined (e.g., in the *back-end* functionalities for administrators, but not in the public *front-end* or in the login procedure).
- For the *server-side* programming, Java (*servlets, JSP*) is the **required** language. Any DBMS and template engine can be employed, if required. Again, it is possible to rely on external libraries.
- In general, the site must work and have a good *rendering* on Internet Explorer/Edge, Mozilla Firefox and Google Chrome, and *possibly* be compatible with older browsers (in this case it should at least *degrade well*) and with the latest versions of other browsers, like Opera. Browser compatibility **must** be explicitly stated in the documentation.

## Project Development and Documentation

---

The specifications may not be exhaustive or completely defined. Every feature added or refined, also through an interaction with the stakeholder or the site end users, will be adequately assessed. All the design choices must be discussed and motivated.

The final project, developed following the guidelines given by the present specification, must be a fully functional website, whose contents and features will be assessed during the examination. The specification parts marked as *optional*, if not developed, will not make the project insufficient but, on the other hand, will not allow your project to reach the highest mark. If you choose to implement an optional feature, the result should not be necessarily perfect or complete: it must only show your commitment to deal with an advanced issue.

The documentation (**in electronic format**) which accompanies the project **must** contain at least the following information:

- Indication of software dependencies (which libraries are required on the client and server side?).
- Indication of the functionalities that have been developed or not developed. Detailed description of any extra or optional functionality added to the project.
- Diagram showing the site structure.
- Relational schema of the database (if any).
- Analytical description of the site layout, also indicating its static/dynamic components.
- Description of any advanced technology (language, framework, plugin, library, etc.) used in the project, specifying the reason of its adoption and the actual contribution given to the project development.

- Description of *any* cross-browser programming/rendering problem encountered, list of compatible browsers.
- Screenshots of the most important website pages (*optional*).

*The actual contribution of each group member* to the project **must** be declared in the documentation (indicating, for example, the members that mainly worked on server and client side programming, the layout designer, etc.). During the examination, each group member will describe its part of the realisation.

## Project Evaluation

---

To evaluate the project, the following issues will be considered (in order of importance):

- 1.Compliance with the specifications.
- 2.Technical correctness.
- 3.Organizational clarity and correctness of the contents.
- 4.Accessibility and standards compliance.
- 5.Appropriate use of static and dynamic contents.
- 6.Design quality.
- 7.Adequacy of the documentation.

This evaluation will be combined with the result of the project discussion.

## Additional Information

---

This specification is available in PDF format on the website of the Web Engineering course at the address <http://www.di.univaq.it/gdellape>. Additional information on the specifications can be obtained directly via email by writing to [giuseppe.dellapenna@univaq.it](mailto:giuseppe.dellapenna@univaq.it).

Please note that projects should be carried out by *small* student groups (three members is the recommended number). Exceptions to this rule must be agreed with the teacher.

## Summary

This summary should be compiled and sent to the teacher, in electronic format, *before* the examination. A copy of this form should be also attached to the project documentation.

**Project name:** \_\_\_\_\_

**Authors:**

First Name	Last Name	ID	Contribution / Role in project development

**Client-side technologies/libraries/frameworks:**

(HTML5, CSS 2/3, Javascript, JQuery, Bootstrap...)

---



---

**Server-side technologies/libraries/frameworks:**

(e.g., JSP, MySQL, JPA, template engines...)

---



---

**Compatible browsers:**

Browser	Version	Compatible	Degrading	Not compatible	Not tried
Internet Explorer Edge					
Mozilla Firefox					
Google Chrome					
Opera					
Safari					

**Project discussion date:** \_\_\_\_\_