Group 6 USE E-Learning	Authors
 Design and Testing 2 D05 - Hackathon	- Ángel Delgado Luna (angdellun@alum.us.es) - María De Gracia Piñero Pastor (marpinpas@alum.us.es) - Belén Garrido López (belgarlop3@alum.us.es) - Ezequiel Portillo Jurado (ezeporjur@alum.us.es) - Alejandro Rodríguez (aleroddia1@alum.us.es)

DEL 5

DEL5 is an application whose goal is educational. The student´s representation comissions have a lot of topics to manage in a short period of time. Some of these activities can be proclaims about teachers, building problems, management of different associations, treasure functions like inventory control, changing students between associations, and some more.

Information requirements

1. The actors of the system are administrators, members, and students. For every actor, the system must store a name, an optional middle name, a surname, an optional photo, an email, an optional phone number, an optional address, and an arbitrary number of social profiles. The system must store the following data regarding such profiles: a nick, the name of the social network, a link to a profile in that social network.

2. Actors can exchange messages. For every message, the system must keep track of the sender, the recipient, the moment when it was sent, the subject, the body, its priority, and some optional tags. Priorities are HIGH, NEUTRAL, or LOW, other are not expected. Every actor has the following message boxes: in box, out box, trash box, notification box, and spam box. When an actor receives a message, it gets to the in box unless the system flags it as spam, in which case it gets to the spam box. When he or she sends a message to another user, it's saved to the out box. When an actor removes a message from a box other than trash box, it is moved to the trash box; when he or she re-moves it from the trash box, then it is actually removed from his folders; when a message is not in any folder, it's actually removed from the system. The previous boxes are pre-defined, and the actors must not be allowed to delete them, to change their names, or to move them. Actors are allowed to create new boxes that they can manage arbitrarily; managing boxes includes nesting a folder within another folder. Note that a message may be stored in several boxes, but the system must keep a unique copy.

3. Phone numbers should adhere to the following patterns: "+CC (AC) PN", "+CC PN", or "PN": "+CC" denotes a country code in range "+1" up to "+999", "(AC)" denotes an area code in range "(1)" up to "(999)", and "PN" denotes a number that must have at least four digits. Phone numbers with pattern "PN" must be added automatically a default country, which is a parameter that can be changed by administrators. Note that phone numbers should adhere to the previous patterns, but they are not required to. Whenever a phone number that does not match this pattern is entered, the system must ask for confirmation; if the user confirms the number, it then must be stored.

4. Students can publish proclaims. Every proclaim must store a ticker, a title, a description, the moment when it is created, and some attachments as extra content. Every time a student publishes a proclaim, he/she will have to provide a Student Identity Card. This consists on the following data: The centre where is studying, a code of four numbers which is placed on the back – right – top and the identification number. A proclaim can be saved in final mode. If it is saved on final mode, cannot be edited or deleted. Once a proclaim is definitely saved, this takes pending status.

5. Every member can self-assign any proclaim that is not attended on the system. If necessary, a proclaim can be manage by more than one member.

6. Proclaims will be attended by members. For every proclaim, members will have to accept or reject it. In the first case, a law must be attached. If rejected, a reason must be provided.

7. Accepted proclaims will be shown (when listing) in green, rejected in red, and pending in orange.

8. There are a lot of kind of proclaims understood under Category´s name. Teachers proclaims, Building Infrastructure Proclaim, Teaching Schedule or even bullying. They are organised into a hierarchy controlled by the administrators. belongs to a parent category, but the root category, which is called "CATEGORY".

9. Every member has a finder. With it, they get the proclaims whose status is "pending". Finder will search according to the following criteria: registered date, a ticker, description, proclaim type. If criteria is not found, all results will be given.

10. If a proclaim is accepted, member and student are allowed to create commentaries about it while a proclaim has not been closed. Every commentary must have a short description and some optional attachments.

11. There is a new kind of Actor: Collaborators. They help members joining to the different comissions and preparing events.

12. Collaborators can manage their portfolio. Every portfolio has an only personal record, zero or more work reports, zero or more study report, or more miscellaneous report.

13. For every report, the system must store the title and the moment when it is registered. For every personal report, it must be saved the full name of the actor, his/her address and a phone number. For every work report, users will have to register the start and the end date (this last can be optional), the name of the business where the member has worked or is working and a short piece of the text about the job done. For every study report, it must be saved the current course (the current course is understood as the highest course in which is registered), the moment when he/she began the degree and the percentage of credits course, an optional average and an optional end date. For every miscellaneous report must be provided a short piece of text and some optional attachments.

14. Members can create comissions. Every comission must have a name, a description and the moment when it was created. Furthermore, the different interested collaborators can join to any comission depending on its wishes.

15. Once a collaborator is in an comission, he/she can create events. For every event, the system must store the title, description, the celebration moment and a state. When an event is created takes pending status and is registered in draft mode. It is only gets final mode when a member accepts or rejects it

16. Students, collaborators and members who go to any event can create notes. A note consists on a value from 0 to 10 and a short optional description.

17. Collaborators can swap from one comission to another. A swap consists on the collaborator who requests for a swap, the collaborator who is in other comission, the comission to swap, a phone number, a status and an <u>optional</u> description. The status, when it is sent, has pending status. The collaborator who receives it decides if accepts or no.

18. There's a new kind of actors in the system: sponsors, who support some events by means of sponsorships.

19. The system must store the following data regarding sponsorships: a URL to a banner, a link to a target page, and a valid credit card.

Functional requirements

20. An actor who is not registered must be able to:
    - ✓ Register to the system as member or student
    - ✓ Register to the system as collaborator.
    - ✓ List and show the different events and its notes. Furthermore, the actor must be able to navigate to the profile of the collaborator who created the event. That include his/her personal information.
    - ✓ Register to the system as sponsor
21. An actor who is registered must be able to:
    - ✓ Do the same as an actor who is not authenticated
    - ✓ Edit his/her personal data.
22. An actor who is registered as Member must be able to:
    - ✓ Manage of his/her proclaims.
    - ✓ Creating, Updating, Deleting and Listing the comments of every proclaim.
    - ✓ Self-assigned a proclaim which has not been assigned yet by another member.
    - ✓ Find all proclaims has no member assigned.
    - ✓ Manage the different comissions. This consists on listing, showing, creating, updating and deleting them. Until an comission is not declared in final mode, collaborators are not allowed to join them.
    - ✓ List the created events grouped by its status with edition possibility. An event cannot be published if it is in draft mode.

23. An actor who is registered as Student must be able to:
    - ✓ Create, Update, Delete, List and Show the different proclaims that he/she may have.

- ✓ Make comments on those proclaims which are not closed. Members can closed those which they consider are finished.
- ✓ Create and List notes about the events that he/she has gone.

24. An actor who is authenticated as collaborator must be able to:

- ✓ Manage his/her portfolio. That includes listing, creating, updating and deleting the different reports.
- ✓ List the different comissions created. He/she should have the possibility to join other.
- ✓ Manage his/her events. That includes listing, creating, updating, deleting and showing actions. When an event is created, its status is pending.
- ✓ List the different swaps received and sended. Moreover, a list of the collaborators join in every comission.
- ✓ Create a swap and send it to another collaborator.
- ✓ Edit a swap received. That means changing its status from pending to accepted or rejected.

25. An actor who is registered as Administrator must be able to:
- ✓ Register new administrators to the system.
- ✓ Ban and Unban spammer actors
- ✓ Create, Edit, List and Delete category of proclaims. A category can be deleted as long as it is not used by more than two proclaims.
- ✓ Launch a process that computes a score for every event. The score is computed building on the notes that they've got. The system must analyse the notes in the events and compute their average. If average is higher than 5, the score shown is 1, otherwise –1.
- ✓ Manage a dashboard with the following data:
  1. Average, Standard deviation, minimum and maximum of published proclaims by students.
  2. Average, Standard deviation, minimum and maximum of taken proclaims by members.
  3. Show with a histogram the number of proclaims per type registered.
  4. Average, Standard deviation, Minimum, Maximum of the number of finder results.
  5. The minimum, the maximum, the average, and the standard deviation of the number of collaborators per comissions.
  6. The minimum, the maximum, the average, and the standard deviation of the number of notes per event.
  7. The ratio of members with an comission.
  8. The top-three of nearest events to the current system date.

26. An actor who is authenticated as sponsor must be able to:
- ✓ Manage his/her sponsorships. That includes showing, listing, creating, updating them. A sponsorship can be deactivated or re activated according to user´s will.

Non-Functional requirements

27. The system must be available in English and Spanish. (Unless otherwise stated, the data are not required to be available in several languages, only the messages that the system displays.)
28. The system must be easy to customise at run time. The customisation includes, but is not limited to: the name of the system (it's "DEL5 – Students Representation" by default); the banner shown at the header (it's the one available at https://i.imgur.com/eDCGHR9.png by default); the message that is shown on the welcome page ("Welcome to DEL5! The best website for student rights defense" is the default welcome message in English; "¡Bienvenidos a DEL5! El mejor lugar para la defensa de los derechos de los estudiantes" is the default welcome message in Spanish); a list of spam words (it's "sex", "viagra", "cialis", "one million", "you have been selected", "Nigeria", "sexo", "un millón", and "ha sido seleccionado" by default).
29. Photos are not required to be stored in the database, but links to external systems like Pin-terest.com or Flickr.com, just to mention a couple of examples.
30. When an actor gets a message that contains a spam word, it must be stored in the spam box instead of the input box.
31. Tickers must adhere to the following pattern: "yymmdd-xxxxxx", where "yymmdd" refers to the year, month, and day when the corresponding entity is registered, and "xxxxxx" to a random uppercase alpha-numeric string. No two entities may have the same ticker since it's assumed to be a unique external identifier.
32. The maximum number of results that a finder returns is 10 by default. The administrator should be able to change this parameter in order to adjust the performance of the system. The absolute maximum is 100 results.
33. Attachments are not required to be stored by the system, but their URLs to external storage systems like Drive.com or Dropbox.com, to mention a few examples.
34. The results of a finder are cached for one hour by default. The administrator should be able to configure that period at will in order to adjust the performance of the system. The mini-mum time's one hour and the maximum time's 24 hours.
35. The percentage of credits must be between 0 and 100.
36. Personal data must be coherent with the provided personal data given by the user logged.
37. Take into account that notes cannot be update or delete once are saved. As consequence, a double confirmation is required.
38. Whenever an event is displayed, a random sponsorship must be selected and its banner shown, if any. Banners must be shown as little intrusively as possible.
39. The system must accept the following credit card makes: VISA, MCARD, AMEX, DINNERS, and FLY. Other makes are expected to be accepted in future. The administrator should be able to easily manage the list of accepted credit card makes.

Level A+

40. JSON is a kind of data used for exchanging data between applications. This is one of the foundations from every API Rest which consists on making a structure of your Web Information System that lets user to consume its data. Spring has a library which allows programmers to create their own api rest. That will allow people to use the data provided by your system. So this is why you are intended to produce, choosing any of the use cases from this document, a integration with this tool.

For this level, it must be produced:

- ✓ Implementation of this tool in any of the use case.
- ✓ A report no longer than 1000 words. This report must contain an explanation about its implementation and the way that Spring works with this system.

Level A++

41. There are sometimes that loading data is a too cost and heavy task. The reasons can be: too many users logged or simply the big amount of data to load. As Spring good practices is recommended to use a cache system. That avoids the repetitive calls to database, keeping data available for the application while the content doesn´t get modified.

For this level, it must be produced:

- ✓ Implementation of this tool in any of the use case.
- ✓ A report no longer than 1000 words. This report must contain an explanation about its implementation and the way that Spring works with this system.