

“Web-based Development” Project

Universidade da Madeira

2020-2021

1. Introduction

The Web-based Development project's goal is to implement a set of requirements, described on point 2.

This Project has two parts, each of them making for 25% of the final mark for this course (5+5 points). In the first part, the goal is to implement the project using Node.js technologies, with the Express library. In the second part, the goal is to implement the same requirements using PHP and Laravel technologies.

Part of the code developed by each group in the first part can be reused in the second part (namely HTML, CSS, and JavaScript on the browser side).

Each project group should have 4 people and the groups will be the same for the two parts of the project.

1.1 Delivery requirements

The minimum mark in each of the parts is 8 points. Each part is assessed with a mark from 0 to 20 points, with no decimal points.

In each part, there are **mandatory requirements and optional requirements**. The mandatory requirements add up to a total of 8 points. The possible optional requirements add up to more than 12 points. The groups can choose which of the functionalities to implement (or try to implement).

Functionalities may be implemented that result in a sum greater than 20, but the grade for each part will never exceed 20. When delivering the project a form should be filled in (see section 6) in which the group declares which functionalities they implemented and to which level ("initial", "partial", "complete", or "complete and well tested").

Failure to implement the mandatory requirements results in the project failing. This is the case even if the total achieved with the optional requirements exceeds 8 values.

2. Project Description

The project aims to develop a "social network" where different **people** interact in asynchronous **conversations**, sending **messages** to other people, who participate in the respective conversation. Each person has a set of personal data that they can edit. Among this information is the login/password. History should be kept of the name and icon/image with which each participant introduces him/herself. This name and image are visible to the other participants in the conversations.

2. Common Requirements to be implemented in the Project

The project deliverable will be a combination of the implementation of the requirements below:

2.1 Mandatory Requirements

1. It should be possible for a user to create an account with: username, password, and profile picture.

2. The system must present a login screen, only after a successful login can a user start or access the conversations.
3. Several people can participate in each conversation, and each person can participate in several conversations. Anyone can create a conversation. The person who creates a conversation chooses which other people to invite to join that conversation initially and sets up the conversation rules.
4. Each message necessarily belongs to a conversation, but there may be shared links to specific messages. A conversation has a descriptive name, which by default is the date it was started. The conversation name can be changed later, but the name history (and date) is kept.
5. Each person can decline the invitation to join a conversation. A person can leave a conversation at any time. Having already participated in the conversation he can later ask to be readmitted.
6. Messages are added to the "flow" of a conversation based on the date, but it is also possible to have messages in reply to another message, also sorted by date, which creates a message tree (like in Facebook).
7. At least 2 DRY principles or sass techniques should be used for styling the HTML content. The code can be pre-compiled into a .css, but keeping the original .scss in the final delivery. These principles or techniques are not limited to those taught in class, so there is freedom of choice and implementation, provided they are adequate and properly documented.
8. An external module/dependency for node.js should be developed and used. The exported functions must be coherent within the module, i.e. serve a common purpose.
9. All messages should be stored in the provided database.
10. Use selectors for class and id in both jQuery and CSS rules

2.2 Optional Requirements

Each group may implement any combination of the extra requirements below:

11. Additionally, a participant can add a reference to a previous message, but add it to the main "flow" (not creating a hierarchy - tree), as in WhatsApp.
12. Within a conversation, direct messages can be sent between participants (TO and CC, but not BCC). These messages are completely private and cannot be seen by administrators.
13. Messages can be text, contain one or more files, of various types (images, zip, powerpoints), audio, video, or even links to external content, or links to messages from the same conversation or messages from other conversations.
14. All messages must be saved in the provided database.
15. Configurable conversation rules include:
 - a. Conversations can be:
 - i. Public (all content is visible). Public chats can be presented as a blog.
 - ii. Reserved (the chat name is public, but contents are only visible to participants of the chat)
 - iii. Private (not even the chat name is public)
 - b. Invitation to new participants can be:
 - i. All participants can invite other participants at any time.
 - ii. Invitations to new participants must be pre-approved (see polls).
 - iii. Only administrators can make invitations (the creator is initially the only administrator).
 - iv. Only the creator of the conversation can make invitations.
 - c. About the promotion to administrator level (or demotion from administrator level):
 - i. It is done by a vote among all participants (see voting).

- ii. It is done by voting between all the current administrators (except the one targeted in case of promotion or demotion). In case the chat's creator is demoted, the administrator in charge for the longest time takes on the role of creator. No conversation may be left without a chat organiser.
 - iii. It is done only by the chat creator.
 - d. About the visibility of the conversations to the participants:
 - i. All messages in the conversation are visible to all participants.
 - ii. All the messages of the conversation are visible to each participant from the date of the first participation.
 - iii. Only messages in the periods in which the participant was authorised are visible (this can be more than one date range).
 - e. About editing messages in a conversation:
 - i. Previously written messages can be edited at any time (but the history of previous versions remains available)
 - ii. Previously written messages can be edited within 15 minutes of the last edit.
 - iii. Messages cannot be edited.
 - f. About deleting messages:
 - i. Message authors can delete their messages.
 - ii. Message authors can delete their messages within 30 minutes of the last edit.
 - iii. Administrators can hide messages (moderation), keeping the information that the message existed (and its content is visible for administrators)
 - g. When deleting a message with replies (and replies to replies) the whole branch of the tree will have the same destination as the reference message.
 - h. About terminating a conversation:
 - i. Any guest can request a vote for conversation termination.
 - ii. Only administrators can request a vote (among themselves) for conversation termination. Conversations will be archived, not being possible to add messages, but they can be accessible to administrators or also to participants.
 - iii. Only the creator can terminate conversations.
 - i. About changing conversation settings:
 - i. (the same options as in the previous option, with the voting including information on what changes are proposed)
 - j. About the votes in the conversations
 - i. (see detailed description below)
- 16. A poll is a specific type of message.
 - a. The poll has a title, a description (optional), and a closed enumeration set of possible answers (by default: yes, no, abstention).
 - b. Polls have an opening and closing date/time.
 - c. Each participant/admin in the conversation can vote, and change his mind by voting again during the voting period.
 - d. The creation of polls has the same options as invitations (*mutatis mutandis*)
 - e. Votes can be visible (the equivalent of a show of hands) or private (secret ballot).
 - f. People who have already voted/are yet to vote can be visible only to administrators or to all voters.
 - g. Voting results can be invisible, visible only to administrators, or visible to all participants in the poll. In the case of private votes, only the results are shown periodically and just in case the number of voters in this period ensures the secrecy of the vote (more than 5 and that not all have voted the same way).

- h. A vote can be cancelled using the same options as for deleting messages (*mutatis mutandis*)
 - i. The participants in a vote are defined at the moment the poll message is created (entry/exit of administrators and participants in the conversation does not affect voting rights).
 - j. The settings of each vote are visible to the voters and cannot be changed during the process.
 - k. Private votes guarantee the privacy of the voters.
- 17. Reactions to a message
 - a. Allow adding different emoticons to a message (1 per person, for example, "thumbs up", a smiley, or a heart)
 - b. Count the reactions
 - c. List the names of people who responded by tapping on the corresponding emoticon
- 18. Possibility to add a friend/contact. In this case, not all users in the database will appear as an option that can be added to a conversation, but only those that have been previously added as friends. For this you will need:
 - a. An option to add friend/contact by searching in the general users' list
 - b. After the person accepts the request they will be available to be added to conversations

5. Delivery

The first delivery of the project is scheduled for **03-05-2021** closing on 23:59. This delivery should be made by emailing the teachers with the following subject in the message: **[DBWP1GX]** where X indicates the group number. In order to reduce the size of the deliverables, it will not be necessary to include external libraries as deliverables, all content should be archived in a .zip or .rar file with the name GX, where X indicates the group number.

Together with the project, students should share a Readme file, with instructions on how to configure and run the project (for example, the need to install libraries). This is where students should, for example, indicate names of users already created for testing and other points that may facilitate the implementation and evaluation by teachers. Besides the project, a form describing the implemented requirements should also be submitted (see the section below).

Each project group will be provided with a connection string to access a MongoDB database. The connection is exemplified in the git repository of the practical classes (mongoConfigs.js, in ExExpress1 branch, last commit). This DB is mandatory and can be easily accessed through Compass. As for the Laravel project, the migrations and seeds mechanism should be sufficient.

6. Monitoring requirements

In order to facilitate the monitoring of the requirements, the submission should be accompanied by the completion of a form with the same structure as the form below. This form will indicate the progress of implementation of the project requirements

Requirement	Status
1	complete and thoroughly tested
2	complete and thoroughly tested
3	complete and thoroughly tested
8	partially complete
15 d ii	started

(the data presented in the table is a mere example of a possible reporting of the progress of the requirements)

7. Code of academic ethics and integrity

In this course, each student is expected to abide by the highest standards of academic honesty. This means that every idea that is not the student's idea must be explicitly credited to its author(s). Failure to do so constitutes plagiarism. Plagiarism includes using ideas, code, or solutions from other students or individuals, or any source other than the course texts, without crediting those sources. Citing sources does not change the course grade, but students should not copy code from other students, or give their own code to other students under any circumstances. Note that it is every member's responsibility to maintain access to the code for the members of their group only.