

Scoping task

For this task, we have an existing system sending messages from the system to the company. The messages are of different type, like marketing and billing and more. The requirement is that we need to add a new functionality to the existing system where each company can add multiple emails, and they can be used for different types of messages.

Firstly we need to make changes to the database, so that it is able to store multiple emails for one company.

Then we need to make changes to the backend and maybe add APIs that will have the ability to insert new mails to the database for the companies.

Then at the end we need to make changes to the UI so users can add and configure emails for their purposes.

When the changes are made we need to write tests, and then test the new functionalities but also test the other functionalities that were already working.

After everything is tested update the existing documentation for the system and deploy the new changes to production.

Lets begin with the assumptions.

1. Assumptions

- The system uses a relational database where the company profile is stored.
- The company mail is stored in the table company in a simple field.
- Different modules of the code send different mail, for example sending marketing messages and billing messages are handled by different modules.
- The system does not have the option to add, change or delete the email address.
- The company uses the existing address for logging in.
- The system already has some kind of UI that manages the company profile.
- The system already has implemented user authentication and authorization.
- There are already automated tests.
- It has existing documentation.

2. Changes

- Database changes
 - For our new requirement we need to make a new table called `company_email`, and in that table we will store all emails of a company
 - The relation will be One-to-Many: One company can have multiple emails, but one email can belong to only one company.
 - The table will have the next fields:
 - `Id`: PK
 - `company_id`: FK (to the company)
 - `email`: email address
 - `msg_type`: For what types of messages it is used for
- Backend and API
 - We will need to add new APIs for CRUD operations on the emails.
 - Create: API for adding new email for the company.
 - Read: API for getting all or some of the emails of the company.
 - Update: Edit email of the company (Example: change for which type of messages it is used for).
 - Delete: Remove an email from the company.
 - Also we need to change and add backend functionalities.
 - Add form validation. Implement validation when adding or updating email.
 - Change code modules for sending emails. Each module should check which email should be used for sending emails.
 - Limit the change of email functionality to only users with admin privileges.
- Frontend and UI
 - Add the option for adding new emails. We will need new form for adding emails to companies.
 - UI for managing emails for companies. The system needs to have the ability to select which emails are for which types of messages.
 - Show success or error messages depending on the outcome. If there was an error tell the user with a user-friendly message. Same when the operation was successful.
- Testing
 - Write automated tests for the new features and run them.

- Test the old features that are working and not broken by the new changes.
- Documentation
 - Update the documentation. Record all changes in the documentation.
- Deployment
 - Deploy the new features to the users. Push the new version to production so the users can use the new features.