

Phishlify MVP #3

This is the third MVP for the Phishlify application. This document should help guide you on what is currently implemented, what you can test, and the next steps to achieve the final product. Note, this document does NOT have to be filled out, it is more of a guide and recommended to fill out, to help improve my next development procedure.

APPLICATION LINK: <https://svencarstenrasmusen.github.io/Phishlify/build/web/index.html#/>

1. Implemented Requirements:

Here below are the functional and non-function requirements that need to be met by the Phishlify application. Note, all requirements that are marked **YELLOW** are the requirements that I strive to complete by the end of the project, as I am working alone now and fulfilling all requirements will be out of scope for the remaining time of the semester.

TFR-01	Login And Registration - The tool must provide the user with the ability to create an account. Here it is crucial to check whether an account for an email already exists or not. The name, surname, email, and password must be required to create an account. The tool must allow the user to perform an authorized login if they already have an account.
TFR-02	Manage Project - For users to be able to keep a manageable overview, the system must allow the user to create projects that the user can then associate with their "victims". Here, a project name, time frame, description, and the responsible user's name are required from the user. The UI must provide the user with the required fields to enter the specific information. The deletion of a project also falls under the managing of a project, meaning the project creator must be able to delete an existing project.
TFR-03	Manage Campaign - Within each project, the user must be able to create one or more campaigns. To maintain an easy overview, only the selected project's campaigns are shown to the user to avoid information overflow. Each campaign must contain fields for the campaign name, the email template that is sent to each "victim", a link to the landing page of the phishing website, the start and end dates of the campaign, and a list of emails to which the phishing mail is sent to. After a campaign has been created, the Phishlify tool is able to provide the project owner to delete a campaign within a project.
TFR-04	Webview First - The Frontend must first be usable to the user from a web browser. All mentioned function requirements have to first be met on the browser version of the frontend before the cross-platform code is adjusted to a responsive mobile version. Creating and managing each project and campaign must be easy to view with the least amount of clicks possible.

Table 1: The first four functional requirements. TFR stands for "Technical Function Requirement".

TFR-05	Expandability - The Phishlify application in all of its entirety, meaning the frontend the middleware, and the database, must be flexible and expandable. Such that, the system can be easily further improved in the future, allowing new features and supported platforms.
TFR-06	Emailing - The application must be able to send phishing emails to all of the added recipients within a campaign, with one simple click from the user of the application. This also includes, the functionality for the user to be able to create and edit emails within the Phishlify tool.
TFR-07	Email Scheduling - The Phishlify application must be able to send the emails to the recipients in a timely manner. This means, that not all emails are sent at once, otherwise, the sender (i.e., Phishlify) will be marked as a spammer. The application will send off a batch of 10 emails every 60 seconds for a duration of 8 hours per day, or at least until the email queue is empty.
TFR-08	Admin User - The Phishlify system must provide support for an admin user. This user can then view all projects and campaigns, with the ability to overwrite, execute or remove them. Additionally, the admin user must be able to manage the normal users within the Phishlify system, allowing the admin user to either create or delete users.
TFR-09	Statistics - For the responsible user of a project, the Phishlify system must show detailed statistics of the campaigns within the project. These statistics must show the click rate and the success rate for each campaign. A click rate would be the linked clicked within the phishing email and the success rate represents how many users fell for the phishing attack.
TFR-10	Hosting - The Phishlify system must be able to host a phishing website, providing a landing page view within the frontend itself. The landing page can be viewed within each campaign improving the overview for the end user.

Table 2: The next six functional requirements, with TFR-05, TFR-06, and TFR-08 marked to be done.

NFR-01	User-Friendly Interface - The frontend of Phishlify should provide the user with a clean and easy-to-understand user-interface. This includes displaying alert, information and confirmation dialogs to the user before important actions are executed. An example situation is the deletion of a project, here a confirmation dialog should pop-up asking the user for a confirmation of the deletion. Keeping the user informed about their actions and potential mistakes.
NFR-02	Runtime - For the tool to help a Kapsch employee manage their simulated attacks, the tool must be effective and performant, with minimal loading times during the use of the tool. The loading times of each function within the tool must not be longer than 3 seconds. This provides a good user-experience (UX).
NFR-03	Secure Channel - The client-server architecture of Pishlify should use the HTTPS protocol to ensure secure communication between the frontend and backend.

Table 3: The three "Non-Function Requirements" that are to be completed. The notion of non-function in this context means, that the application would still work without them, but not in a secure, efficient and usable way.

From the above listed requirements within Table 1, Table 2, and Table 3, the following requirements are implemented (unless testing shows otherwise) in this third MVP (Minimal Viable Product):

- TFR-01
- TFR-02
- TFR-03
- TFR-04 (Continuous Work)
- TFR-05 (Continuous Work)
- TFR-06 (Through dummy GMAIL account)
- NFR-01
- NFR-02
- NFR-03

2. Testing:

Below are the functionalities you may test as much as possible, it is recommended that you use the example inputs to avoid any security/privacy issues. Please tick if a feature is successful, faulty, needs more user-interface feedback (example: dialog for errors) or add some comments if needed. At the end of this section is a listing of currently not implemented minor features.

- **Register**
 - Please register an account with a dummy email and dummy password.
 - Example **Correct** Input:
 - Name: Max Mustermann
 - Email: Max.Mustermann@email.com
 - Password: 1234
 - Repeated Password: 1234

- Example **Incorrect** Input:
 - Name: Max Mustermann
 - Email: Max Mustermann
 - Password: 1234
 - Repeated Password: 12

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Login**

- Either use a newly created account. Or use a sample existing user listed below.
- Example **Correct** Input (Existing User):
 - Email: toolbox@email.com
 - Password: 1234
- Example **Incorrect** Input (Non-existing user):
 - Email: illusive.man@email.com
 - Password: 1234

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Create Project**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.
- Example **Correct** Input:
 - Project Name: My First Project
 - Person In Charge: toolbox

- Language: English
- Customer: Some Customer
- Domain: www.some-customer.com
- Start Date: 03.06.2022
- End Date: 21.06.2022

○ Example **Incorrect** Input:

- Currently no specific incorrect input is known. Meeting needed.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

• **Create Campaign**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.

○ Example **Correct** Input:

- Campaign Name: My First Campaign
- Domain: www.someOtherDomain.com
- Description: This is a description for my first campaign.
- Start Date: 03.06.2022
- End Date: 21.06.2022

○ Example **Incorrect** Input:

- Currently no specific incorrect input is known. Meeting needed.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Add Email Recipient**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.
- Example **Correct** Input:
 - Email: sven@home.lu (or you may use your email. The sender is sven.the.eagle@gmail.com)
- Example **Incorrect** Input :
 - Email Regex not yet implemented.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Send Email**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.
- Example **Correct** Input:
 - Subject: My First Email
 - Text: This is a test email for trying out the Phishlify application.
- Example **Incorrect** Input :
 - No HTML Code reader implemented.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Delete Email Recipient**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Delete Campaign**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Delete Project**

- Please specify in the comments section if the navigation and user-interface is intuitive or not.

Success: _____

Faulty: _____

Feedback: _____

Comment: _____

- **Missing Features:**

- If not listed here and a feature is not yet implemented. Please add the missing feature in the comment field.
 - Email Regex when adding recipients.
 - Regex checks for creating a campaign.
 - Regex checks for creating a project.
 - Customer in campaign is currently still NULL due to old database table version.
 - Dashboard does not list any information. A meeting is required to ask what it should display specifically.
 - Comment: _____

3. Next Steps:

For the next MVP a meeting is required to go into the specifics of what specific information should be shown in specific parts of the application. I will go deeper into the smaller details, such as Regex checks, empty text checks, etc. to further refine the Phishlify application. I will also look into how to containerize the entire Phishlify system.