



CEN 308 SOFTWARE ENGINEERING

PROJECT DOCUMENTATION

Letter Yourself

Prepared by:
Kasim Suleyman Oner
Sanjin Stevanovic

Proposed to:
Nermina Durmić, Assist. Prof. Dr.
Aldin Kovačević, Teaching Assistant

22.06.2022

Table Of Contents	2
1. Introduction	3
1.1. About the Project	3
1.2. Project Functionalities and Screenshots	3
2. Project Structure	6
2.1. Technologies	6
2.2. Database Entities	6
2.3. Architectural Pattern	6
2.4. Design Patterns	6
3. Conclusion	7

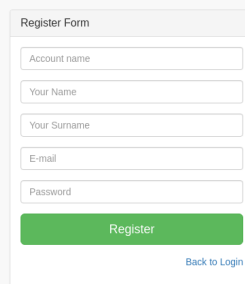
1. Introduction

1.1. About the Project

[LetterYourself](#) is a web application where users can write emails and schedule to send them at a later time. They can send emails to themselves or to someone else. Users add a recipient, write the letter, select the date and time, and on the selected date and time the email will be sent.

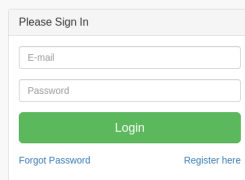
1.2. Project Functionalities and Screenshots

Users can create an account, or log in if they already have an account.



A screenshot of a web form titled "Register Form". It contains five input fields: "Account name", "Your Name", "Your Surname", "E-mail", and "Password". Below the fields is a green "Register" button and a blue link "Back to Login".

Register



A screenshot of a web form titled "Please Sign In". It contains two input fields: "E-mail" and "Password". Below the fields is a green "Login" button. At the bottom, there are two blue links: "Forgot Password" and "Register here".

Login

Users can schedule a new email or change an already scheduled email.

The screenshot shows the 'Add Letter' modal form. The form has the following fields:

- Title:** A text input field with the placeholder 'To Me'.
- Send date:** A date and time picker showing '07/19/2022, 03:14 AM'.
- Letter Body:** A large text area containing the text 'One Month From Now'.
- Receiver email:** A text input field with the placeholder 'example@email.com'.

At the bottom of the modal, there are two buttons: 'Close' and 'Save Changes'.

Add email

Users can see a list of their scheduled emails.

The screenshot shows the 'Letters' list view. The table has the following columns: ID, Title, Body, and Send At. The table contains 11 rows of data. The first 10 rows have a 'Send At' date of '2021-03-31 22:15:00', and the last row has a 'Send At' date of '2021-06-26 01:08:00'.

ID	Title	Body	Send At
66	My Letter11	My Dear friend..	2021-03-31 22:15:00
65	My Letter10	My Dear friend..	2021-03-31 22:15:00
64	My Letter9	My Dear friend..	2021-03-31 22:15:00
61	My Letter7	My Dear friend..	2021-03-31 22:15:00
60	My Letter3	My Dear friend..	2021-03-31 22:15:00
59	My Letter	My Dear friend..	2021-03-31 22:15:00
58	My Letter2	My Dear friend..	2021-03-31 22:15:00
57	NEW	new letter..	2022-05-27 20:41:00
56	adsadasd	asdasd	2021-06-26 01:30:00
55	aasdx	casdasd	2021-06-26 01:08:00

Letters

Users can see a list of communications which shows who will receive the letters.

Letter Yourself

Letters

Communication

Admin Panel

Communication

Show 10 entries

Letter Title	Receiver Email
editedNEW	edited@galp.com
edited2	edited2@galp.com
edited3	edited3@galp.com
new2	ss@gapl.com
edited3	daqs@gapl.com
vvv	ccc@galp.com
aaaz	saqw@gapl.com
edited3	swqas@gapl.com
ccc	zxcas@gapl.com
cxazac	czqwas@gapl.com

Showing page 1

PreviousNext

Communications

Admin can see the user list on the admin panel.

Letter Yourself Admin

Users

Back

Users

Show 10 entries

Search:

User ID	Name	Surname	Email	Status
1	Suleyman	Oner	suleyman@galp.com	ACTIVE
2	Ali	Akcam	alihan@galp.com	PENDING
3	Huseyin	Gulecen	huseyin@galp.com	PENDING
4	Furkan	Erkan	furkan@galp.com	PENDING
6	Oguz	Tilisim	oguz@galp.com	ACTIVE
45	Suleyman	Oner	suyo571oner@gmail.com	ACTIVE
46	Sanjin	Stevanovic	sanjin.stevanovic@stu.ibu.edu.ba	ACTIVE

PreviousNext

Admin-Panel

2. Project Structure

2.1. Technologies

For the backend we used PHP with: FlightPHP framework for RESTful API, Swiftmailer for sending emails, PHP-JWT for JSON Web Tokens, Swagger for API documentation.

For the database we use MySQL.

For the frontend we used Javascript with: JQuery for API requests.

For tests we used PHPUnit with Selenium.

In the backend, for PHP, we used the PSR-12 coding style. In the frontend, for Javascript, we used the Airbnb Javascript coding style.

2.2. Database Entities

This project has the these database entities:

- receiver stores information about email recipient
- persons stores information about users
- letter stores information about the email
- communication stores connections of email and email recipient
- accounts stores information about user accounts

2.3. Architectural Pattern

This project uses the Layered Architectural Pattern. This pattern is used because it provides great flexibility, so that a layer can be modified without affecting other layers as long as the layer provides the same functionality to the layer above.

2.4. Design Patterns

This project has these design patterns:

- **Template pattern:** used in the backend, in the files:
api/services/BaseService.class.php and *api/dao/BaseDao.class.php*
Because all of the service and dao classes have the same structure and use many same methods, it would be messy to repeat so much code, so we use the template method, separating the common logic into BaseService and BaseDao respectively.
- **Bridge pattern:** used in the backend, in the files in the directory:
api/services
The service classes use the methods in the dao classes to interact with the database.

3. Conclusion

Because this project is fairly simple it already has all the functionalities implemented, but its appearance currently is a basic template, and in the future we could redesign the appearance of the application to give it a more unique look.