

GNG1503

**Manuel d'utilisation et de produit pour le projet de conception**

**Technologie de messagerie : eMessage PRO**

Soumis par:

JIBMM Solutions – Groupe FC32

Boukary Timi Kaoura, 300093867

Mustapha Ezzeddine, 300260202

Ibrahim Gbané, 300210971

Joel kasong Tshisos, 300143694

Le 9 avril 2023

Université d'Ottawa

# Table des matières

---

Table des matières.....	ii
Liste de tableaux .....	iv
Liste d'acronymes et glossaire.....	v
1 Introduction.....	1
2 Aperçu.....	2
2.1 Conventions.....	5
2.2 Mises en garde & avertissements .....	5
3 Pour commencer .....	6
3.1 Considérations pour la configuration .....	6
3.2 Considérations pour l'accès des utilisateurs.....	7
3.3 Accéder/installation du système.....	8
3.4 Organisation du système & navigation .....	11
3.5 Quitter le système.....	12
4 Utiliser le système.....	13
4.1 Page de connexion :.....	13
4.2 Pages de navigation :.....	14
4.2.1 Messages .....	15
4.2.2 Nouvelles .....	17
4.2.3 Calendrier.....	18
4.2.4 Paramètres.....	18
5 Dépannage & assistance .....	19

5.1	Messages ou comportements d'erreur.....	19
5.2	Considérations spéciales .....	19
5.3	Entretien .....	19
5.4	Assistance.....	20
6	Documentation du produit .....	21
6.1	Interface eMessage Pro .....	21
6.1.1	NDM (Nomenclature des Matériaux) .....	22
6.1.2	Liste d'équipements .....	22
6.1.3	Instructions.....	22
6.2	Essais & validation.....	39
7	Conclusions et recommandations pour les travaux futurs .....	40
8	Bibliographie.....	41

## Liste de tableaux

---

Table 1. Acronymes ..... v

Table 2. Glossaire ..... v

.

# Liste d'acronymes et glossaire

---

**Table 1. Acronymes**

Acronyme	Définition
Info.	Information
app	Application
LCO	Logement communautaire d'Ottawa
API	Application Programming Interface
iOS	iPhone operating system
CRUD	Create, Read, Update, Delete

**Table 2. Glossaire**

Terme	Acronyme	Définition
Page de Navigation	Pdn.	Les pages de l'application ayant accès après la connexion à un compte

# 1 Introduction

Ce livrable a pour but de montrer l'utilisation de notre application ainsi que notre site internet afin de permettre à notre client de pouvoir bien comprendre le fonctionnement de celle-ci.

Notre site web et notre application ont été testé à de nombreuses reprises lors de phases de prototypage. Aux termes de ces essais, il est possible d'affirmer que notre solution finale répond correctement aux besoins de notre client la LCO. Les agents de **LCO** avaient besoin d'une méthode de communication facile avec les locataires, leur indiquant le temps des constructions, les aidant à la nouvelles technologie des édifices. Suite à une présentation globale de la solution lors de la Journée de la conception, il est possible de certifier que ce projet a été apprécié de notre client tant par son efficacité, sa simplicité et son respect des normes imposées.

Ainsi nous allons vous fournir notre explication détaillée sur chaque point pertinent à la réalisation de l'interface de messagerie entre les gestionnaires et les locataires.

## 2 Aperçu

Dans le cadre de ce projet de conception, notre client, Logement Communautaire d'Ottawa (LCO) avait besoin d'une application de messagerie pour pouvoir communiquer avec leurs locataires. Ces messages seront ciblés en fonction de la nature du problème, ils seront personnalisables dans le cas de situations uniques. En effet, la LCO a pour objectif de réduire l'émission de gaz à effets de serres lors de leurs divers travaux de rénovations : rendre les nouveaux logements plus écolo-conscient. Dans ce sens, cette solution de messagerie numérique permettra de réduire la production de papier de distribution dans les édifices. Cette nouvelle technologie de messagerie se présente alors comme un outil efficace de communication avec les locataires dans les édifices à haute performance. Notre application facilite donc le travail des gestionnaires de la LCO.

Cette technologie de messagerie, nommée eMessagePro, est capable de simplifier encore plus le travail en créant des groupes de locataire. Par exemple les locataires de la rue X étage Z. De plus cette application est munie d'un haut niveau de sécurité empêchant toutes personnes non-locataire de créer un compte pour la plateforme mobile. Quant au site web des gestionnaires, l'utilisateur ne pourra pas créer s'y connecter durant le processus de création ou connexion de compte tant que ses données personnelles ne sont pas validées avec celles présentes dans la base de données dans la branche de gestion de LCO.

L'application mobile des locataires est disponible depuis n'importe quel téléphone portable ou encore tablette compatible. Elle se présente telle que l'indique les images ci-dessous.

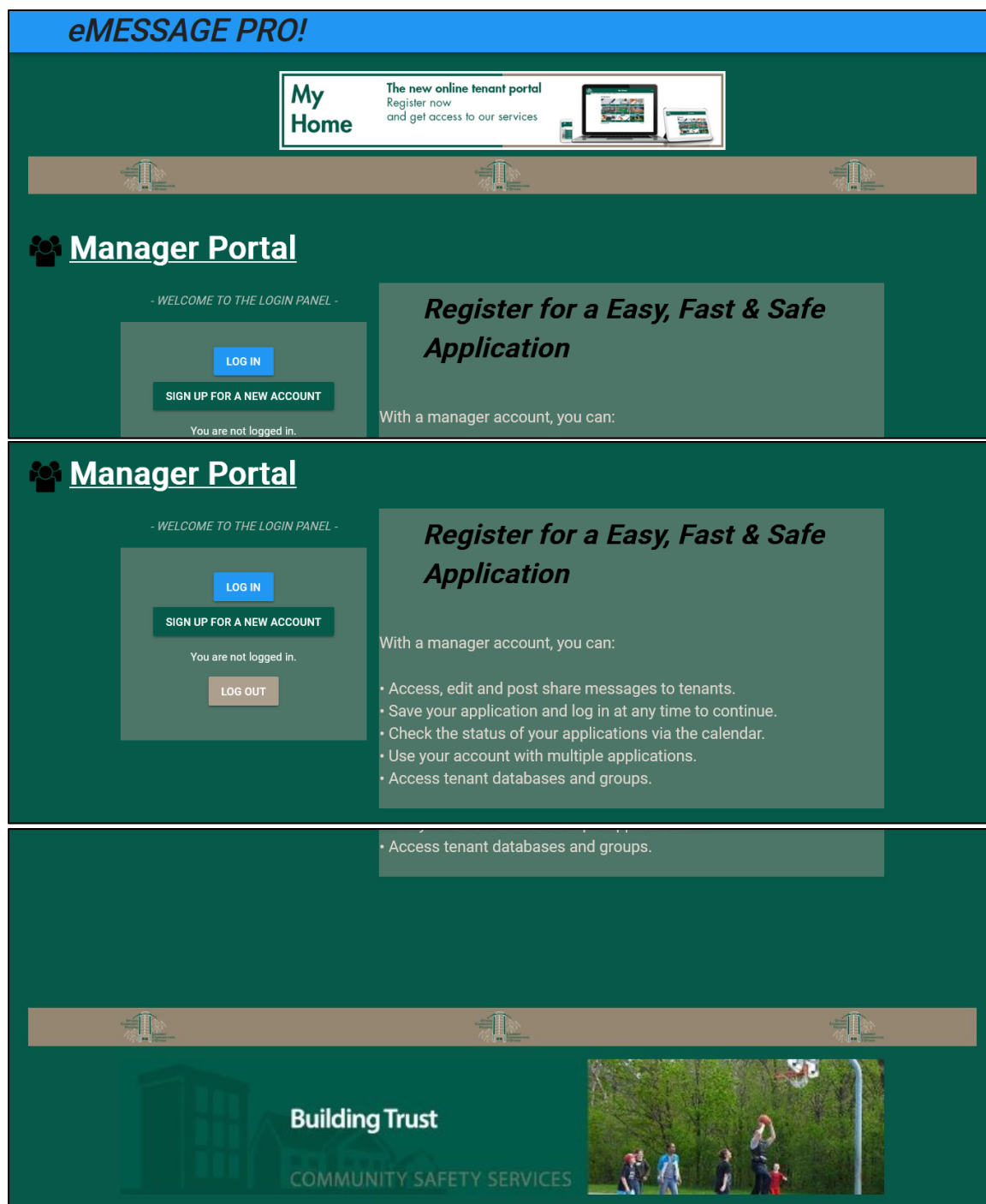


La version mobile de eMessagePro se divise en 4 pages de navigations :

1. Les Messages
2. Le Calendriers
3. Les nouvelles
4. Paramètre

Le site web des gestionnaires est disponible sur le web depuis n'importe quel ordinateur et se présente telle que l'indique les images ci-dessous.





La solution finale du site web a permis d'aboutir à une interface proposant dans un premier temps une page de connexion et dans un second temps une page de modification et de personnalisation de messages de partage.

eMessagePro se distingue des autres applications présentes sur le marché par de nombreux avantages:

- Son temps d'exécution et de partage de messages;
- Sa sécurité;
- Son coût de production et de commercialisation;
- Sa modernité;
- Sa portée;
- Son esthétique.

## **2.1 Conventions**

Les conventions utilisées lors de la réalisation de ce projet ne concernaient que les normes graphiques imposées par la LCO. Les couleurs utilisées dans les différentes sections de navigation sont celles utilisées par la société de logement. La langue utilisée pour la présentation finale est l'anglais.

## **2.2 Mises en garde & avertissements**

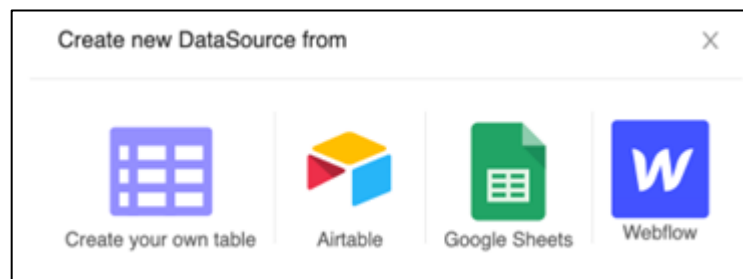
Pour l'application mobile des locataires, si l'utilisateur se connecte à son compte eMessagePro et qu'au bout de 5 min aucune action n'est exécutée, l'utilisateur sera directement déconnecté. De plus, dans le cas de l'utilisation d'un réseau Wi-Fi pas assez fort pour l'application, il sera fort probable que l'application soit lente à répondre aux actions et aux cliques. Les liens tels que « Help » ne seront pas disponibles également.

### 3 Pour commencer

Le produit final qui a été développé par notre équipe de conception se présente sous la forme de deux interfaces de navigation. De ce fait, les étapes vous seront décrites pour les deux séparément.

#### 3.1 Considérations pour la configuration

Pour l'application Thunkable, des blocs de sources de données permettent de travailler avec les informations stockées dans des bases Airtable connectées, des feuilles de calcul Google ou des tables locales. Ainsi ces blocs permettent d'effectuer diverses opérations CRUD sur les enregistrements qui ont été stockés.



Pour l'application Anvil, le langage de codage utilisé est Python. La version payante personnelle propose des outils de stockage divers:

- Des bibliothèques de données Python ;
- Du stockage de fichiers statiques;
- 150 000 lignes de tableau de données;
- 10 Go de stockage de table de données;
- 1 000 e-mails sortants (mensuel)

Personal	
✓	Everything in Free plus...
✓	More Python Libraries
✓	Custom Domains
✓	Static Files Storage
✓	Remove Anvil Banner
✓	Scheduled Tasks
✓	150,000 Data Table Rows
✓	10 GB Data Table Storage
✓	Unlimited Background Task Timeout
✓	1,000 Outgoing Emails (monthly)

Veillez noter que dans le cadre précis de ce projet, la version payante personnelle était suffisante afin d’aboutir à un résultat pertinent. Toutefois, en fonction des besoins de l’utilisateur, Anvil sera en mesure d’apporter davantage de bases de données et de fonctionnalité de traitement.

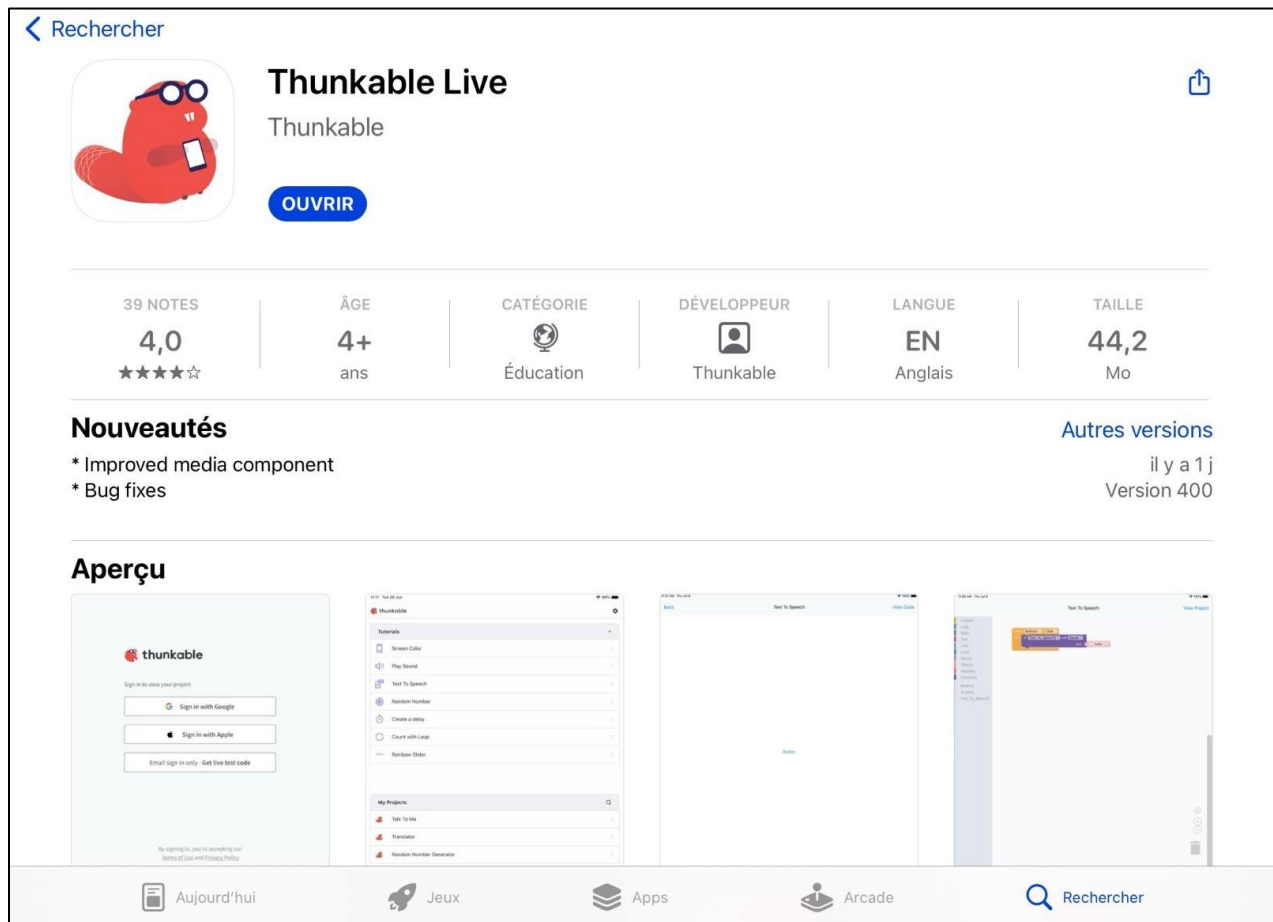
### 3.2 Considérations pour l’accès des utilisateurs

Aucune restriction pour l’accès de Thunkable ou Anvil. Les deux sont accessibles à tout utilisateur.

### 3.3 Accéder/installation du système

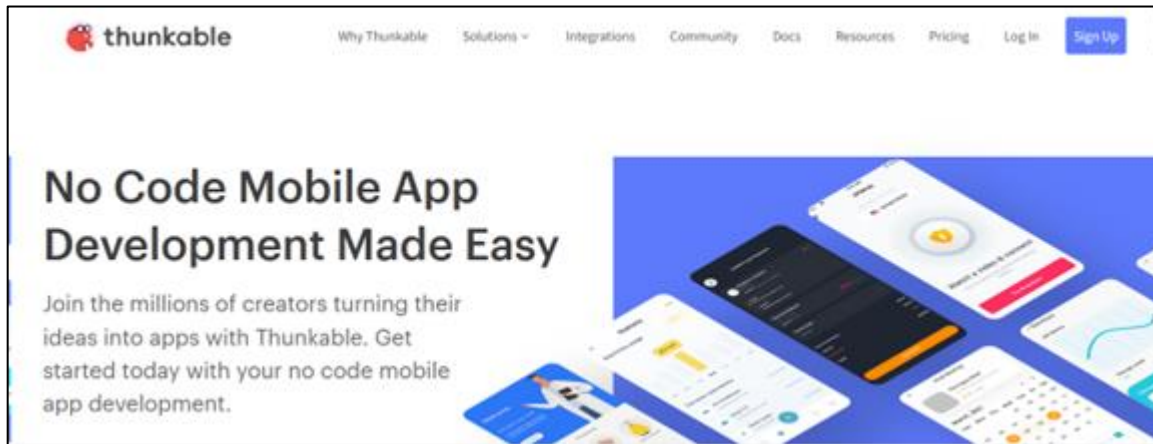
#### Application Mobile des locataires:

Afin d'accéder à cette interface, il faut tout d'abord télécharger l'application mobile Thunkable depuis votre téléphone (ou tablette). Elle est disponible sur la plateforme de téléchargement qui varie selon l'outil utilisé (ex. App Store ou encore Google Play).

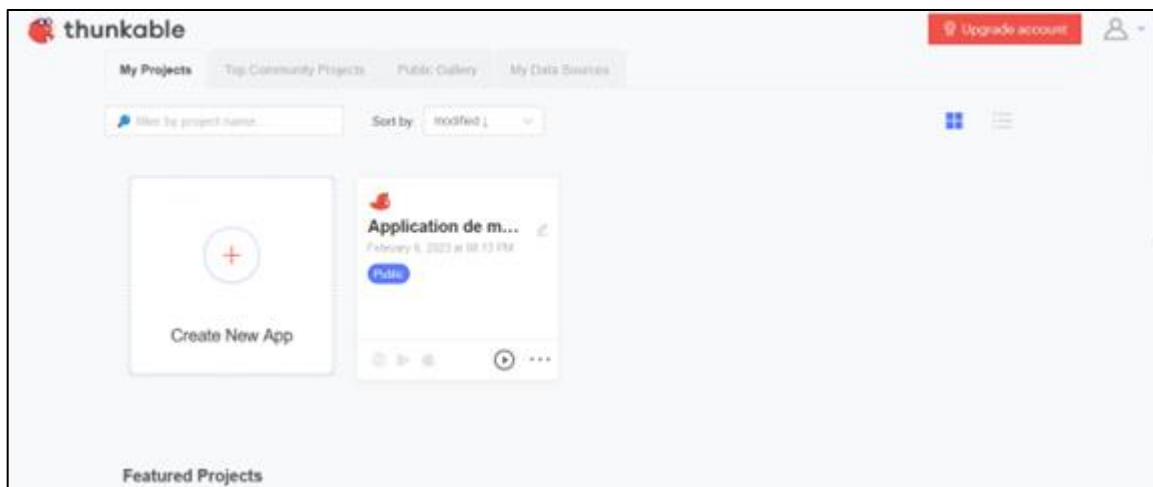


Ensuite, ouvrir l'application et suivre les étapes décrites afin de lier l'application développée depuis l'ordinateur par programmation à l'application Thunkable du portable. Il sera alors possible à l'utilisateur d'utiliser l'interface des locataires depuis son outil.

Pour ce qu'il en est de l'application web de Thunkable, il faudra tout d'abord créer un compte, puis accéder à l'édition d'application.

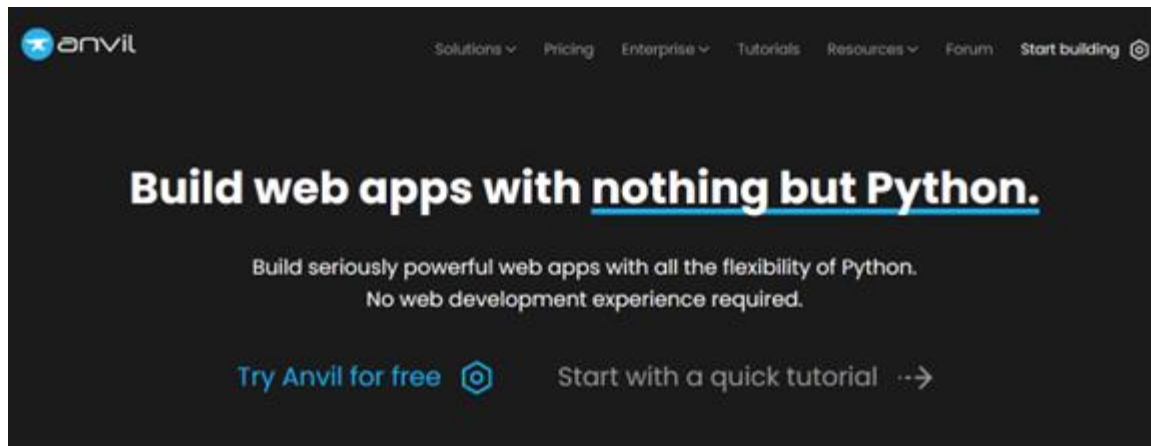


Après la création d'un compte:

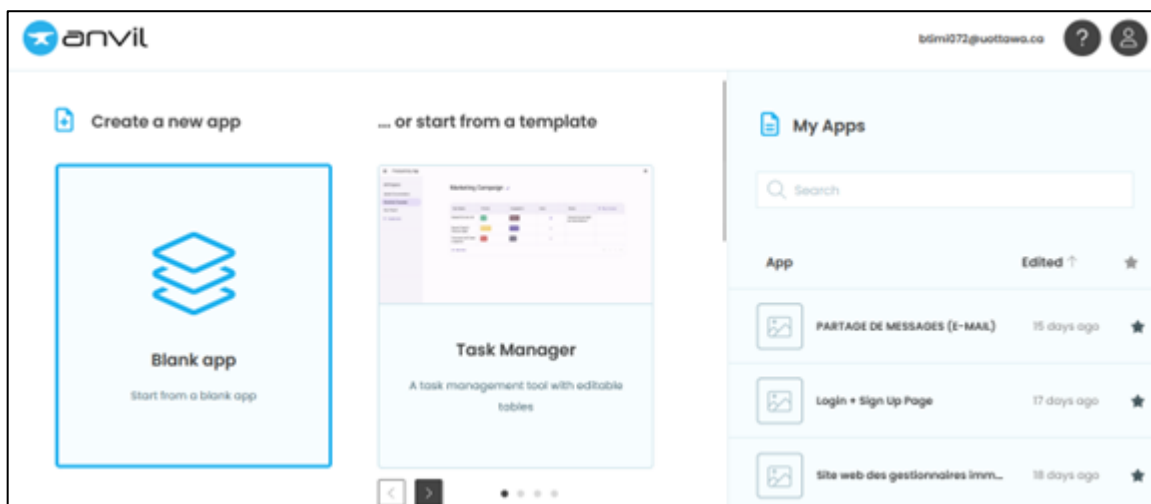


### **Site web des gestionnaires:**

Le site web des gestionnaires a été développé et tester depuis l'application Anvil. Accessible depuis le web, il s'agit d'une nouvelle façon de créer d'autres applications web, avec rien d'autre que le langage de codage 'Python'. Veuillez noter que l'utilisateur doit avoir des bases de compréhension écrite de l'anglais car Anvil n'est disponible que dans cette langue.



Afin de comprendre le fonctionnement détaillé de l'application, nous vous invitons à explorer les documentations disponibles sur la page de Anvil ainsi que les vidéos de tutoriel ou encore les gabarit et exemples de sites déjà conçu. Une fois que l'utilisateur accède à la page de connexion de Anvil, il sera en mesure de créer un compte puis de naviguer à travers les options disponibles et pertinentes à la réalisation de ce projet. Nous pouvons citer celle de créer une nouvelle application.



Une fois que l'utilisateur accède à cette page, il lui sera possible de créer l'application destinée aux gestionnaires.

### **3.4 Organisation du système & navigation**

#### **Application mobile :**

##### **Page de connexion**

Une fois Thunkable ouvert, l'utilisateur aura une page de création de compte ou de connexion. Une fois connecté il aura l'option de langue : français ou anglais. Puis il pourra accéder à eMessage Pro et voir directement la page d'affichage des messages de partage. Un onglet d'affichage de calendrier est également disponible pour pouvoir accéder à l'historique des messages. Enfin, la plateforme offre une option de paramètres.

#### **Site web des gestionnaires:**

##### **Page d'accueil de la plateforme:**

Cette page permet de se connecter ou de créer un compte. Les options de déconnexion et d'oubli de mot de passe y sont également incluses.

##### **Page de création/modification de message:**

Cette page se présente en trois sous parties. Tout d'abord, l'utilisateur sera amené à choisir le groupe de locataire auquel le message sera destiné, ensuite il aura le choix du thème et de l'image qui seront affichés en lien avec la nature du message. La dernière section concerne la modification du message d'envoi. Le gestionnaire devra mettre ses informations personnelles (son nom et son email) afin que le locataire y ait accès et communiquer avec lui par la suite. Par la suite, le gestionnaire pourra rédiger son message de communication et le soumettre.



### **3.5 Quitter le système**

Pour correctement quitter le système de l'application mobile des locataires, l'utilisateur doit fermer l'application Thunkable. Quant au site web des gestionnaires, il faudra cliquer sur le bouton de déconnexion "log out" afin de pouvoir correctement quitter le système.

## 4 Utiliser le système

Les sous-sections suivantes fournissent des instructions détaillées, étape par étape, sur la façon d'utiliser les diverses fonctions ou caractéristiques de eMessage Pro.

### 4.1 Page de connexion :

Le site web des gestionnaires possède une page de connexion assez simplifiée qui permet de créer un compte, s'y connecter ou se déconnecter:



D'autre part, la page de connexion de l'application mobile des locataires se présente de manière plus simple et minimaliste afin d'alléger l'affichage.

La première page de l'app est la page de connexion. Vous avez besoin de cliquer sur le bouton « se connecter » puis choisir la langue d'utilisation soit français ou anglais pour se rendre à la page suivante :

Connectez-vous en insérant votre courriel et votre mot de passe.




## 1. Nouveau compte :

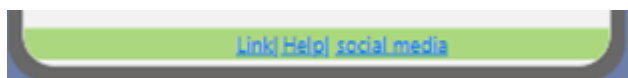
Si vous n'avez pas de compte, vous aurez besoin d'en créer un en cliquant sur « Nouveau compte » et insérer les informations demandées dans la page suivante :



## 4.2 Pages de navigation :

Pour naviguer à travers les Pdn, vous devez cliquer sur l'icône  présent dans le coin haut à gauche de chaque page et puis choisissez la page où vous souhaitez vous rendre.

Vous remarquerez en bas de chaque page qu'il se trouve la section :



Ceci représente des liens utiles dont vous pouvez avoir besoin dans le moment de vos navigations.

- Help : vous conduit au lien d'aide en cas de n'importe quel problème relié à l'app, au logement ou autre...
- Link : vous conduit au lien officiel de la société LCO
- Social média : vous conduit à la page des média-sociaux de LCO.

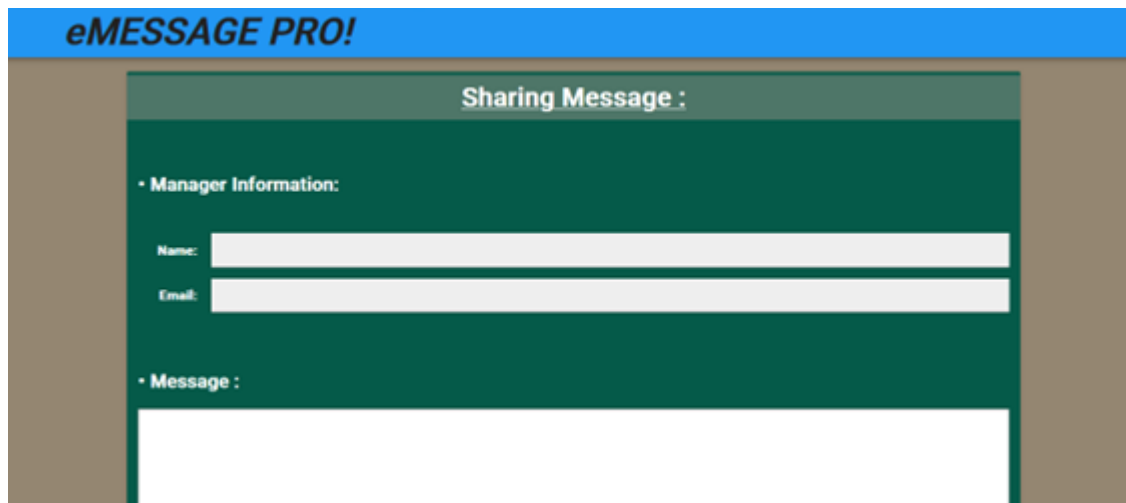
#### 4.2.1 Messages

Une fois le gestionnaire connecté à l'application web, il aura accès à la page d'édition des messages. Comme le présente l'image ci-dessous, le gestionnaire pourra sélectionner le groupe de locataires auquel le message sera destiné. Ces groupes sont prédéfinis et concernent les locataires d'une section en particulière à un étage cible. Ensuite l'image du message de partage est sélectionnée selon la nature du message.



Le gestionnaire devra par la suite inclure son nom ainsi que son adresse email afin de recevoir lui aussi un modèle du message et ainsi avoir une idée de l'affichage du message auquel aura accès le locataire sur l'application mobile.

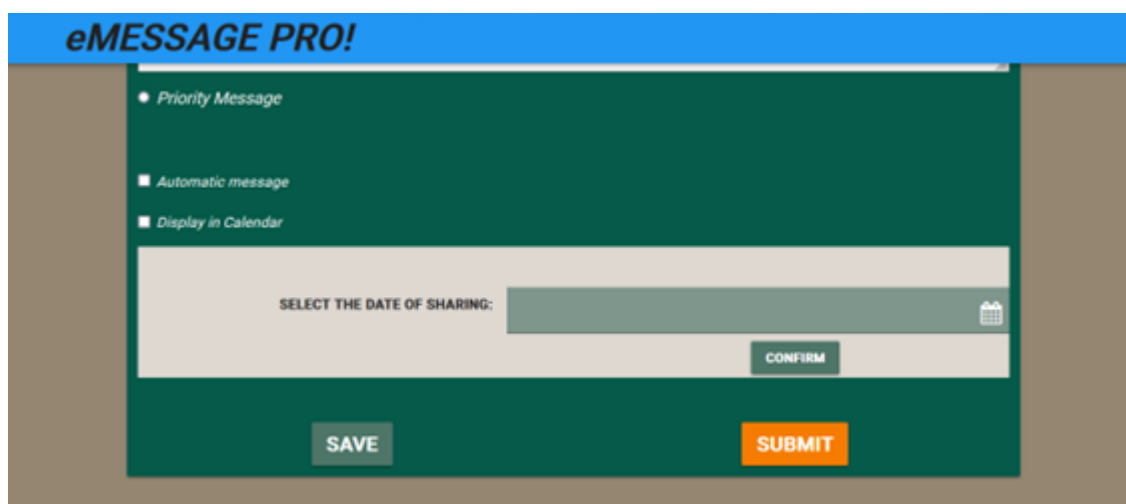
Ensuite l'espace réservé au message sert à rédiger le message de partage.



The screenshot shows a web interface for 'eMESSAGE PRO!'. The main heading is 'Sharing Message :'. Below this, there are two sections: '• Manager Information:' and '• Message :'. The 'Manager Information' section contains two input fields: 'Name:' and 'Email:'. The 'Message :' section is a large text area for writing the message.

Des options de priorité de message sont disponibles afin que ce dernier soit plus visible du côté du locataire. Il est également possible d'automatiser le partage du message ou encore de l'afficher dans le calendrier du locataire dans le cas d'une requête non ponctuelle.

Une fois chacune des fonctionnalités sélectionnées, le gestionnaire sera capable de choisir une heure et une date de publication, ensuite les confirmer, sauvegarder le message et le soumettre.



The screenshot shows the same 'eMESSAGE PRO!' interface, but with additional options. Below the 'Manager Information' section, there are three checkboxes: '• Priority Message' (selected), '■ Automatic message', and '■ Display in Calendar'. Below these is a date selection area with the text 'SELECT THE DATE OF SHARING:' and a calendar icon. A 'CONFIRM' button is next to the date selection. At the bottom, there are two buttons: 'SAVE' and 'SUBMIT'.

La page de messagerie de l'application mobile se présente telle que le montre l'image ci-dessous:


La page suivante représente les messages envoyés par la compagnie de logements. C'est également la page affichée quand vous y êtes connecté.



#### 4.2.2 Nouvelles

C'est une page qui donne des renseignements sur les nouvelles de la société de logement, de leur projet/but ou une brève description de sujet pertinent. Cliquez sur « en savoir plus » en bas de la page pour vous rendre sur leur site web et avoir accès à plus d'informations.

### 4.2.3 Calendrier

C'est l'emplacement où les événements clés seront postés (ils sont de même poster dans la page des « Messages ») vous pouvez vous rendre sur la semaine passée ou prochaine en cliquant sur le bouton  situé à gauche et à droite du calendrier.



### 4.2.4 Paramètres

Vous avez le droit d'activer ou de désactiver vos notifications de messages sur cette page en cliquant sur le bouton « on/off ». Quand « on » s'affiche ça veut dire que les notifications sont activées et vis-versas. Vous pouvez de même vous rendre sur les média-sociaux de la compagnie en cliquant sur le logo du réseau désiré. Le numéro posté sur cette page est le numéro d'aide en cas de problème relié à l'application. Enfin, un bouton de déconnexion permet de se rendre à la page de connexion de l'application mobile.



## **5 Dépannage & assistance**

### **5.1 Messages ou comportements d'erreur**

Sur la page de création de compte ou de connexion, toutes les données fournies doivent être correctes sinon un message d'erreur va être affiché en vous indiquant l'erreur. Cela est valable pour l'application mobile ainsi que pour le site web des gestionnaires. Toutefois un système de modification de mot de passe est disponible en cas d'oubli.

### **5.2 Considérations spéciales**

Dans le cas de l'application mobile des locataires, il se peut que le wifi ne soit pas assez puissant donc des considérations doivent être prises :

- Dans le cas où vous cliquez sur « connect » et ça ne fonctionne pas essayez de sortir puis de rentrer à nouveau dans l'application et essayez de vous connecter.
- Si vous cliquez sur un des boutons qui vous font sortir de l'app (comme ceux des médias sociaux) il se peut que rien ne se passe si votre connexion n'est pas forte donc rafraîchissez la page en faisant défiler vers le haut et réessayez.

### **5.3 Entretien**

Pour éviter tout problème avec le système, assurez-vous que votre connexion wi-fi est forte et correcte.

N'oubliez pas de rafraîchir la page de messages quand vous vous y rendez en faisant défiler vers le haut pour se rassurer que cette page soit mise à jour.



## 5.4 Assistance

En cas de problème urgent, appelez le numéro suivant :613-731-1182 présent dans la page des paramètres.



Vous pouvez, de même, cliquer sur le lien « help » présent en bas de tous les pages.

## **6 Documentation du produit**

### **6.1 Interface eMessage Pro**

Dans le cas de ce travail de groupe, le prototype était uniquement constitué de 2 éléments d'interface : l'application mobile et le site web des gestionnaires. Ainsi la solution proposée n'est comprise que dans une catégorie de logiciel. Toutefois, pour que ces deux logiciels soient utilisés il faudrait pouvoir accéder à un matériel de base; à savoir un portable (iOS, android etc) et un ordinateur (fixe ou mobile).

Le matériel numérique pris en compte dans la réalisation de l'application mobile était uniquement Thunkable. Cette application se sert de modules d'actions afin de faire opérer l'ensemble de l'application mobile. Tandis que le site web a été conçu en se basant sur la version personnelle de Anvil. Ce dernier est basé sur un langage de programmation Python.

Dans le cas où l'une de ces deux applications proposées, à savoir ANVIL ou Thunkable, semble constituer un problème d'utilisation ou de compatibilité, veuillez utiliser l'autre application afin de créer soit le site web soit l'application mobile des locataires.

### 6.1.1 NDM (Nomenclature des Matériaux)

#### ❖ Thunkable:

Lien:

[https://thunkable.com/?utm\\_term=thunkable&utm\\_campaign=US/UK/CA%20|%20Search%20|%20Brand&utm\\_source=adwords&utm\\_medium=ppc&hsa\\_acc=2723599162&hsa\\_cam=15000671177&hsa\\_grp=127293870903&hsa\\_ad=554377436828&hsa\\_src=g&hsa\\_tgt=kwd-360955260416&hsa\\_kw=thunkable&hsa\\_mt=e&hsa\\_net=adwords&hsa\\_ver=3&gclid=CjwKCAjwov6hBhBsEiwAvrvN6ND-a3rk5E-A4n1QaO0Wv1Gedb6Zmi\\_8wgQw4rLvA-St2h4gu2WDyRoCdj4QAvD\\_BwE#/](https://thunkable.com/?utm_term=thunkable&utm_campaign=US/UK/CA%20|%20Search%20|%20Brand&utm_source=adwords&utm_medium=ppc&hsa_acc=2723599162&hsa_cam=15000671177&hsa_grp=127293870903&hsa_ad=554377436828&hsa_src=g&hsa_tgt=kwd-360955260416&hsa_kw=thunkable&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gclid=CjwKCAjwov6hBhBsEiwAvrvN6ND-a3rk5E-A4n1QaO0Wv1Gedb6Zmi_8wgQw4rLvA-St2h4gu2WDyRoCdj4QAvD_BwE#/)

Prix: Gratuit

#### ❖ Anvil – version personnelle:

Lien: <https://anvil.works/>

Prix: 21.23\$ CAD (ou 15\$USD + frais de conversion)

### 6.1.2 Liste d'équipements

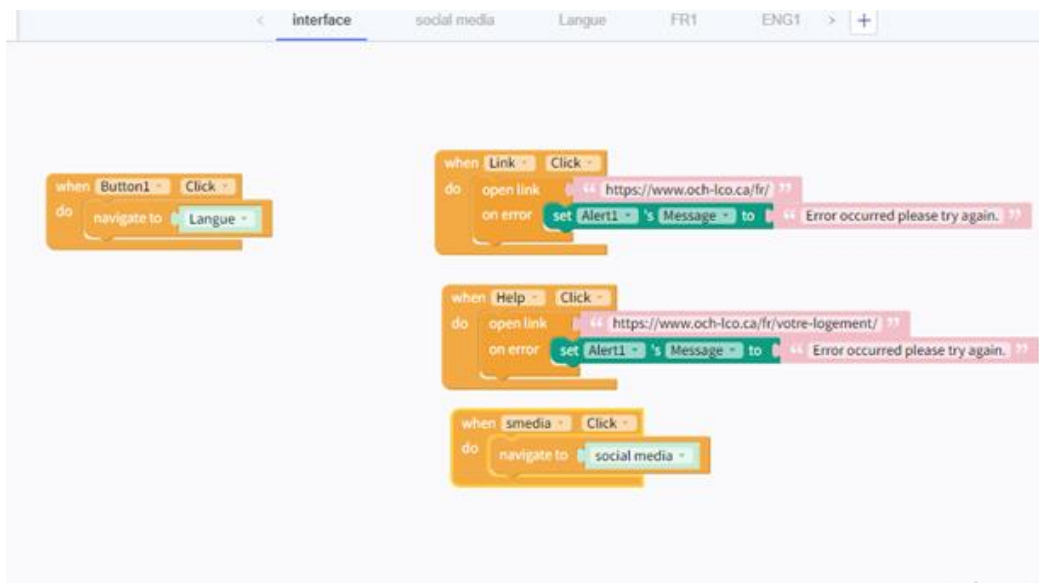
- Ordinateur (portable ou fixe) capable de supporter et exécuter des langages de programmation.
- Téléphone portable (Android ou iOS)

### 6.1.3 Instructions

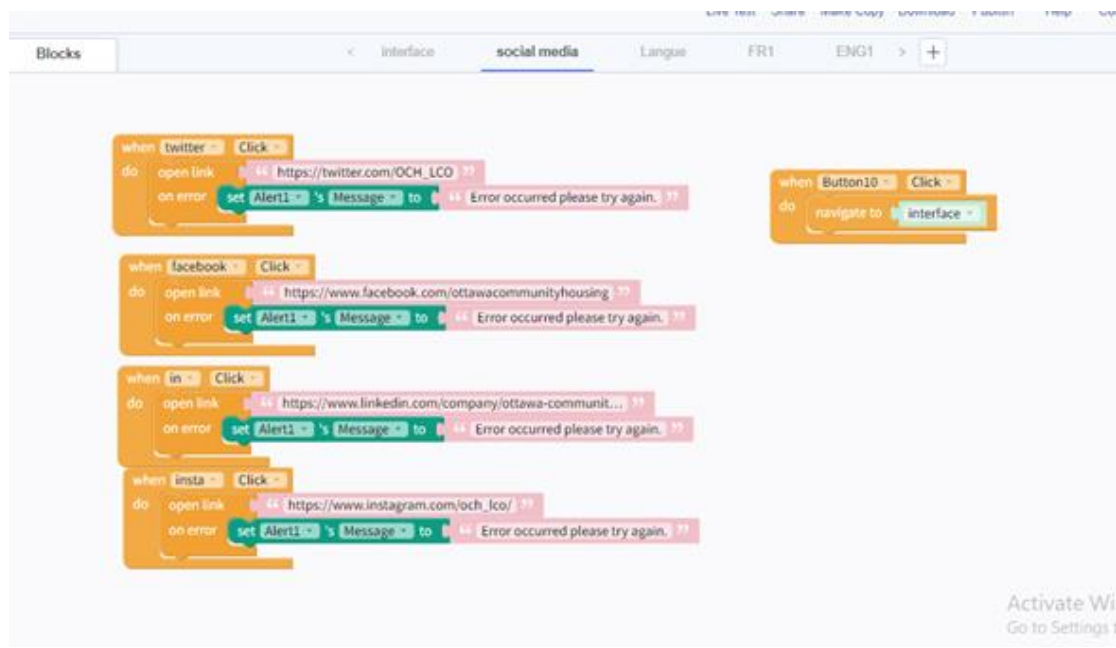
#### **Application mobile:**

Ce sous-système du produit final a été conçu à l'aide l'application Thunkable de manière simple et en suivant les instructions retrouvées sur la page d'accueil de la plateforme et en suivant des vidéos tutoriels sur YouTube pour s'aider en cas de bugs. Les étapes de programmation suivantes se présentent telles que:

## Codage – Page d'accueil



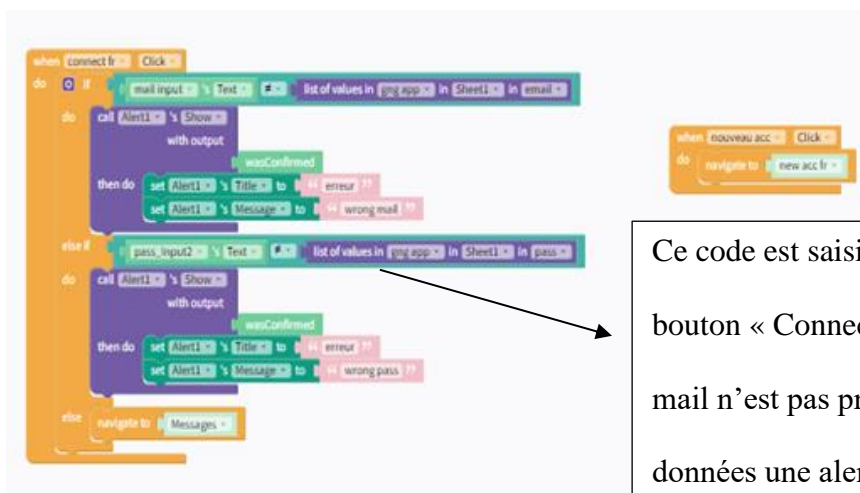
## Codage – Page de liens



### Codage – Écran 3



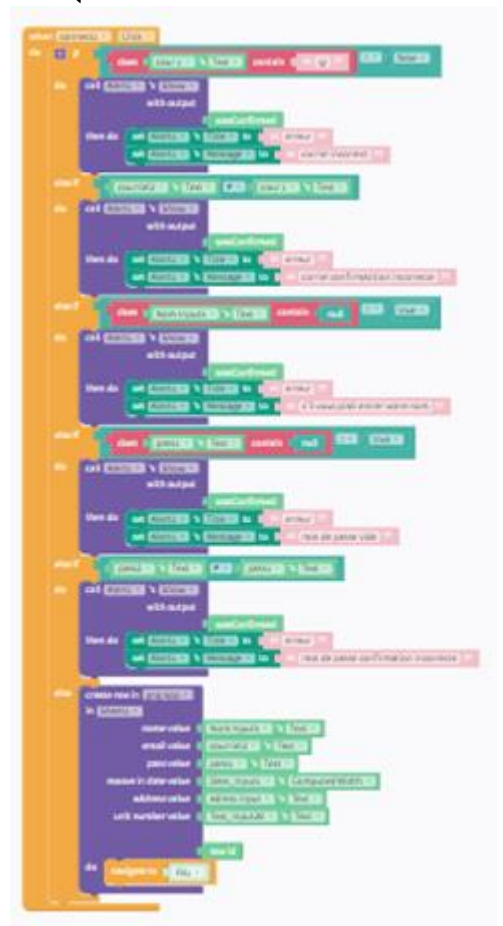
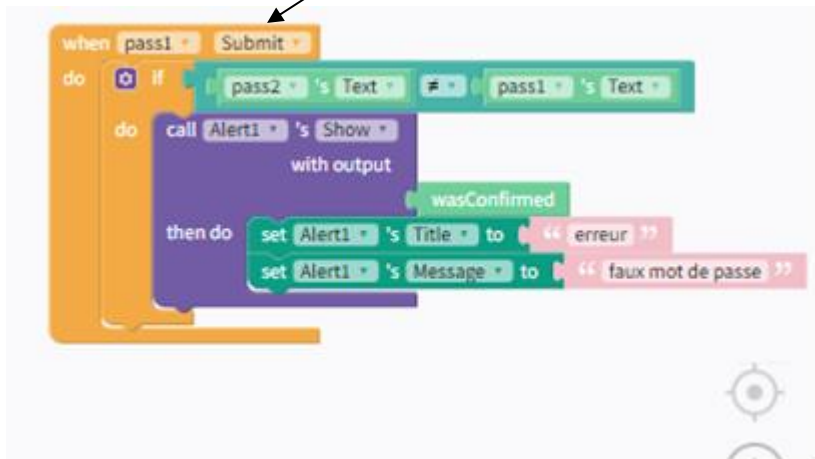
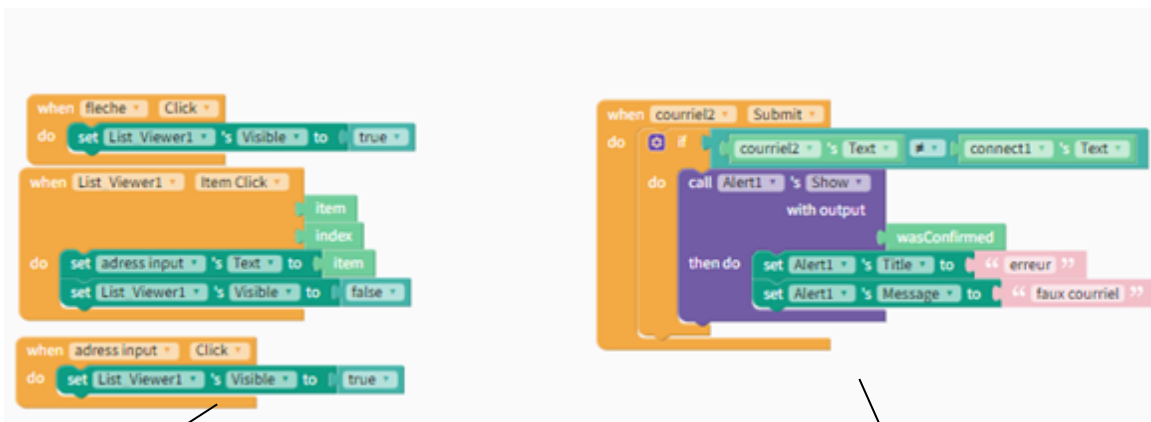
### Codage – Page de connexion



Ce code est saisi si un click était fait sur le bouton « Connect ». Le 1<sup>er</sup> if signifie : si le mail n'est pas présent dans la base de données une alerte d'erreur sera affiché

### Écran d'entrée d'informations de l'utilisateur (création de nouveau compte):

## Codage – Page de création de compte



## Page des Messages



## Codage – Page de messages



Code initialisant les  
boites de messages, photo

Code indiquant le  
processus de  
changement de

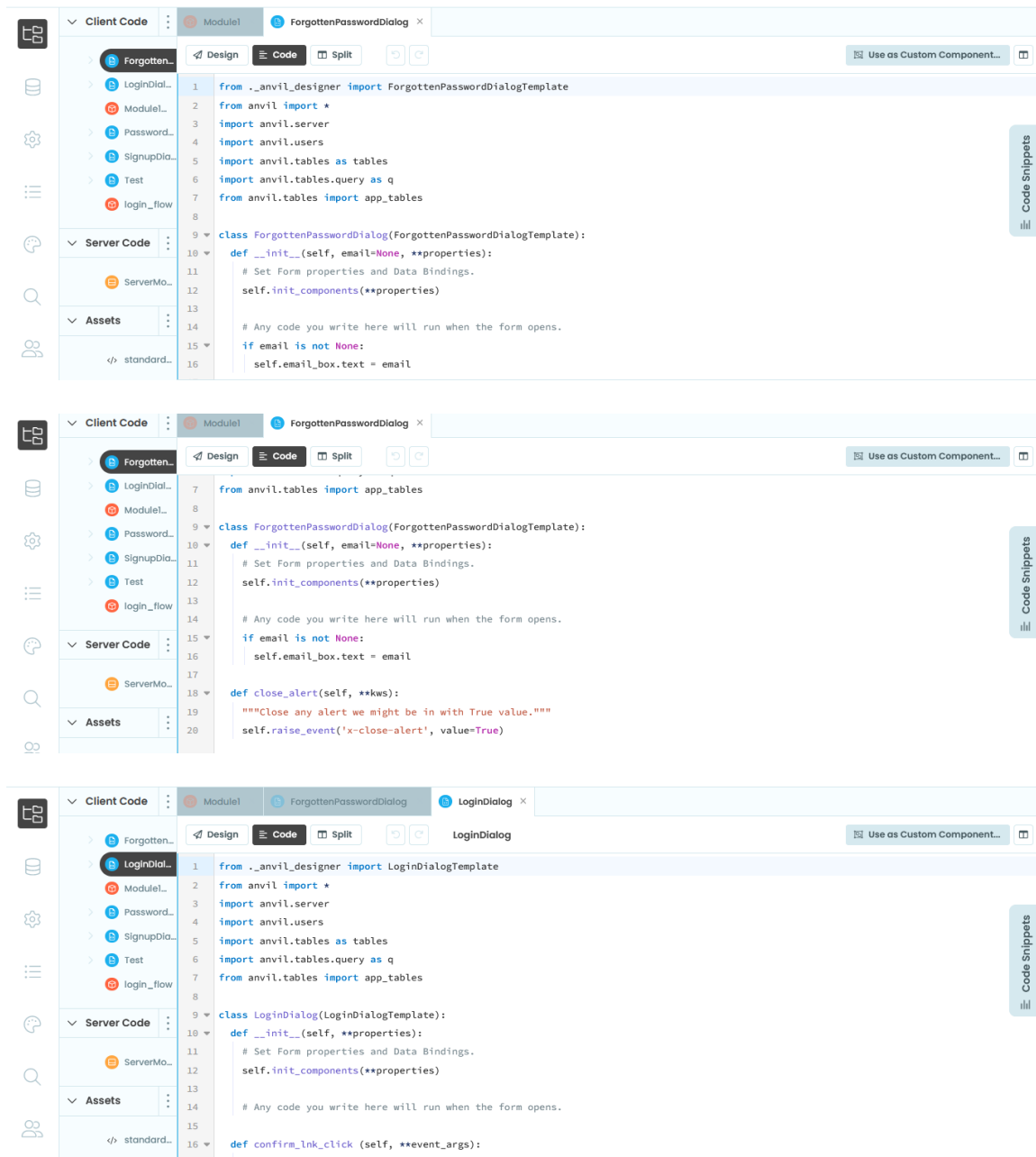




[illegible]

Ce sous-système a été conçu de la base jusqu'au produit final par programmation. En effet, il a fallu se servir de vidéos tutoriels disponibles directement sur le site de Anvil ainsi que sur YouTube en cas de bugs. Pour la création de prototypes, la version gratuite de l'application est suffisante tant elle offre les outils nécessaires à la création de la page d'interface et à l'utilisation de couleurs respectant les normes imposées par notre client, et bien d'autres fonctionnalités de Anvil. Pour la solution finale du produit, la version personnelle payante de Anvil a été utilisée. Les images ci-dessous présentent les codages Python employé afin d'obtenir chacune des fonctionnalité du site web des gestionnaires.

## Codage de la page de connexion :



Client Code

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

Design

Code

Split

Use as Custom Component...

LoginDialog

def confirm\_lnk\_click(self, \*\*event\_args):

"""Close any alert we might be in with 'confirm\_email' value."""

self.raise\_event('x-close-alert', value='confirm\_email')

def reset\_pw\_link\_click(self, \*\*event\_args):

"""Close any alert we might be in with 'reset\_password' value."""

self.raise\_event('x-close-alert', value='reset\_password')

def focus\_password(self, \*\*kws):

"""Focus on the password box."""

self.password\_box.focus()

def close\_alert(self, \*\*kws):

"""Close any alert we might be in with 'login' value."""

self.raise\_event('x-close-alert', value='login')

Code Snippets

Client Code

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

Design

Code

Split

Use as Custom Component...

1

import anvil.server

2

import anvil.users

3

import anvil.tables as tables

4

import anvil.tables.query as q

5

from anvil.tables import app\_tables

6

from anvil import open\_form

7

import custom\_signup.login\_flow

8

custom\_signup.login\_flow.do\_email\_confirm\_or\_reset()

9

open\_form('Test')

10

Code Snippets

Client Code

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

Design

Code

Split

Use as Custom Component...

PasswordResetDialog

1

from ..anvil\_designer import PasswordResetDialogTemplate

2

from anvil import \*

3

import anvil.server

4

import anvil.users

5

import anvil.tables as tables

6

import anvil.tables.query as q

7

from anvil.tables import app\_tables

8

9

class PasswordResetDialog(PasswordResetDialogTemplate):

10

def \_\_init\_\_(self, \*\*properties):

11

# Set Form properties and Data Bindings.

12

self.init\_components(\*\*properties)

13

14

# Any code you write here will run when the form opens.

15

16

def focus\_pw\_repeat\_box(self, \*\*kws):

Code Snippets

Client Code

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

Design

Code

Split

Use as Custom Component...

PasswordResetDialog

8

9

class PasswordResetDialog(PasswordResetDialogTemplate):

10

def \_\_init\_\_(self, \*\*properties):

11

# Set Form properties and Data Bindings.

12

self.init\_components(\*\*properties)

13

14

# Any code you write here will run when the form opens.

15

16

def focus\_pw\_repeat\_box(self, \*\*kws):

17

"""Focus on the password repeat box."""

18

self.pw\_repeat\_box.focus()

19

20

def close\_alert(self, \*\*kws):

21

"""Close any alert we might be in with True value."""

22

self.raise\_event('x-close-alert', value=True)

Code Snippets

30

Client Code

Forgotten...

LoginDia...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

SignupDialog

Design

Code

Split

Use as Custom Component...

```

1 from .._anvil_designer import SignupDialogTemplate
2 from anvil import *
3 import anvil.server
4 import anvil.users
5 import anvil.tables as tables
6 import anvil.tables.query as q
7 from anvil.tables import app_tables
8
9 class SignupDialog(SignupDialogTemplate):
10     def __init__(self, **properties):
11         # Set Form properties and Data Bindings.
12         self.init_components(**properties)
13
14         # Any code you write here will run when the form opens.
15
16     def focus_email(self, **kws):

```

Code Snippets

Client Code

Forgotten...

LoginDia...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

SignupDialog

Design

Code

Split

Use as Custom Component...

```

15
16     def focus_email(self, **kws):
17         """Focus on the email box."""
18         self.email_box.focus()
19
20     def focus_password(self, **kws):
21         """Focus on the password box."""
22         self.password_box.focus()
23
24     def focus_password_repeat(self, **kws):
25         """Focus on the password repeat box."""
26         self.password_repeat_box.focus()
27
28     def close_alert(self, **kws):
29         """Close any alert we might be in with True value."""
30         self.raise_event('x-close-alert', value=True)

```

Code Snippets

Client Code

Forgotten...

LoginDia...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

SignupDialog

Test

Design

Code

Split

Test

Use as Custom Component...

```

1 from .._anvil_designer import TestTemplate
2 from anvil import *
3 import anvil.server
4 import anvil.users
5 import anvil.tables as tables
6 import anvil.tables.query as q
7 from anvil.tables import app_tables
8 from .. import login_flow
9
10 class Test(TestTemplate):
11     def __init__(self, **properties):
12         # Set Form properties and Data Bindings.
13         self.init_components(**properties)
14
15         # Any code you write here will run when the form opens.
16         self.update_login_status()

```

Code Snippets

31

Client Code

Forgotten...

LoginDia...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

SignupDialog

Test

Design

Code

Split

Test

Use as Custom Component...

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

# Any code you write here will run when the form opens.

self.update\_login\_status()

def update\_login\_status (self):

user = anvil.users.get\_user()

if user is None:

self.login\_status\_lbl.text = "You are not logged in."

else:

self.login\_status\_lbl.text = "You are logged in as %s" % user['email']

def login\_btn\_click(self, \*\*event\_args):

"""This method is called when the button is clicked"""

login\_flow.login\_with\_form()

self.update\_login\_status()

def logout\_btn\_click(self, \*\*event\_args):

"""This method is called when the button is clicked"""

Code Snippets

Client Code

Forgotten...

LoginDia...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

SignupDialog

Test

Design

Code

Split

Test

Use as Custom Component...

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

self.update\_login\_status()

def logout\_btn\_click(self, \*\*event\_args):

"""This method is called when the button is clicked"""

anvil.users.logout()

self.update\_login\_status()

def signup\_btn\_click(self, \*\*event\_args):

"""This method is called when the button is clicked"""

login\_flow.signup\_with\_form()

def image\_3\_mouse\_down(self, x, y, button, \*\*event\_args):

"""This method is called when a mouse button is pressed on this component"""

pass

Code Snippets

Client Code

Forgotten...

LoginDia...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

ServerMo...

Assets

standard...

Module1

ForgottenPasswordDialog

LoginDialog

PasswordResetDialog

SignupDialog

Test

login\_flow

Design

Code

Split

Test

Use as Custom Component...

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

from anvil import \*

import anvil.server

import anvil.users

import anvil.tables as tables

import anvil.tables.query as q

from anvil.tables import app\_tables

from LoginDialog import LoginDialog

from SignupDialog import SignupDialog

from ForgottenPasswordDialog import ForgottenPasswordDialog

from PasswordResetDialog import PasswordResetDialog

def login\_with\_form(allow\_cancel=False):

"""Log in by popping up the custom LoginDialog"""

d = LoginDialog()









Client Code

Module1

LoginDialog

PasswordResetDialog

SignupDialog

Test

login\_flow

ServerModule1

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

Assets

standard...

theme.css

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

```

def hash_password(password, salt):
    """Hash the password using bcrypt in a way that is compatible with Python 2 and 3."""
    if not isinstance(password, bytes):
        password = password.encode()
    if not isinstance(salt, bytes):
        salt = salt.encode()

    result = bcrypt.hashpw(password, salt)

    if isinstance(result, bytes):
        return result.decode('utf-8')

@nвил.server.callable
def _do_signup(email, name, password):
    if name is None or name.strip() == "":
        return "Must supply a name"

```

Client Code

Module1

LoginDialog

PasswordResetDialog

SignupDialog

Test

login\_flow

ServerModule1

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

Assets

standard...

theme.css

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

```

pwhash = hash_password(password, bcrypt.gensalt())

# Add the user in a transaction, to make sure there is only ever one user in this database
# with this email address. The transaction might retry or abort, so wait until after it's
# done before sending the email.

@tables.in_transaction
def add_user_if_missing():
    user = app_tables.users.get(email=email)
    if user is None:
        user = app_tables.users.add_row(email=email, enabled=True, name=name, password_hash=pwhash)
        return user

    user = add_user_if_missing()

    if user is None:
        return "This email address has already been registered for our service. Try logging in."

```

Client Code

Module1

LoginDialog

PasswordResetDialog

SignupDialog

Test

login\_flow

ServerModule1

Forgotten...

LoginDial...

Module...

Password...

SignupDia...

Test

login\_flow

Server Code

Assets

standard...

theme.css

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

```

        return "This email address has already been registered for our service. Try logging in."

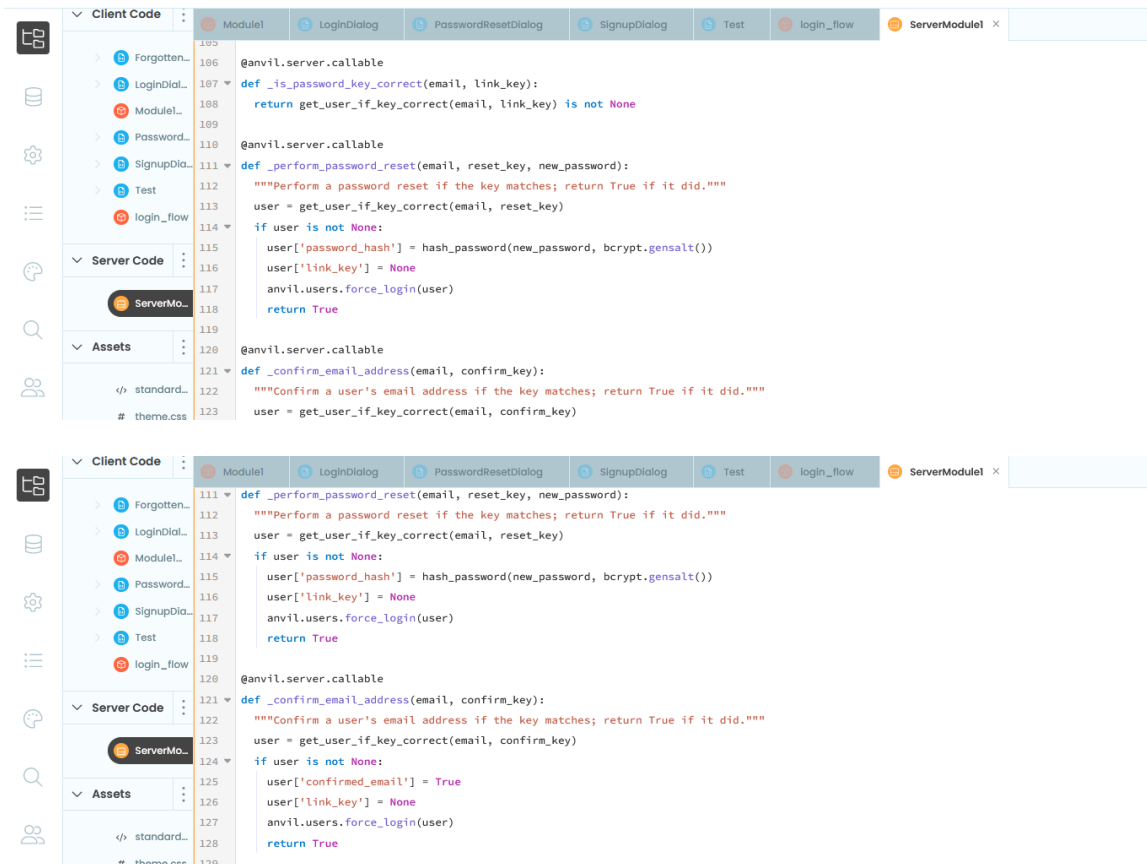
_send_email_confirm_link(email)

# No error = success
return None

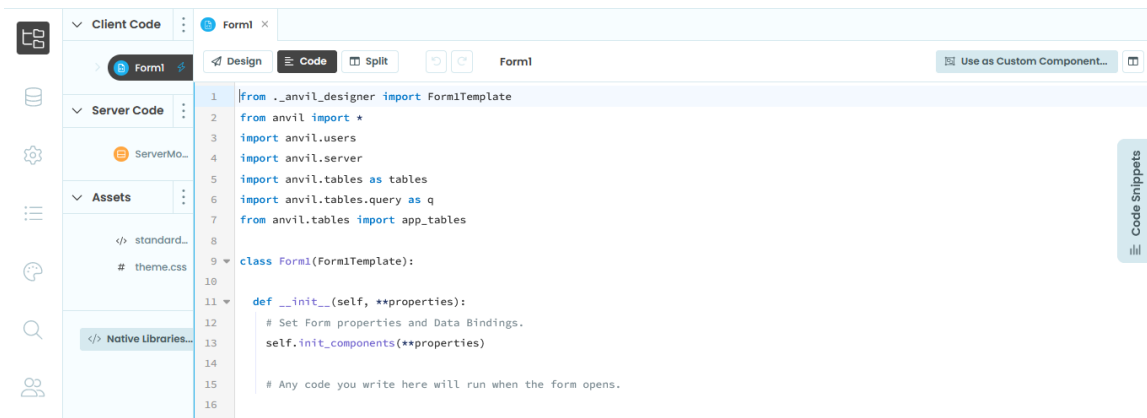
def get_user_if_key_correct(email, link_key):
    user = app_tables.users.get(email=email)

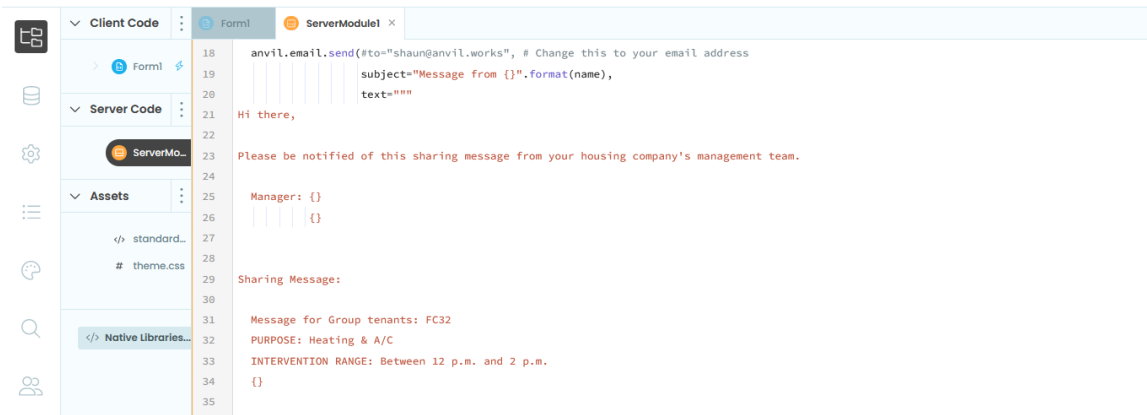
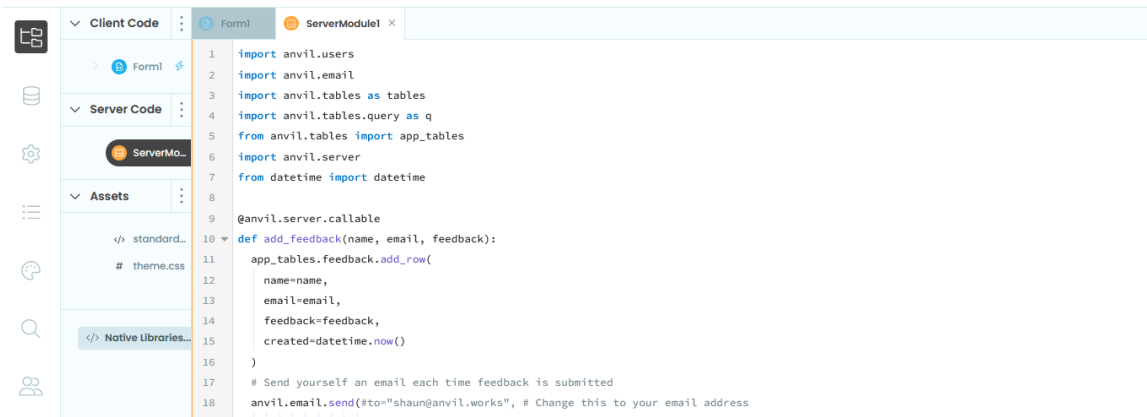
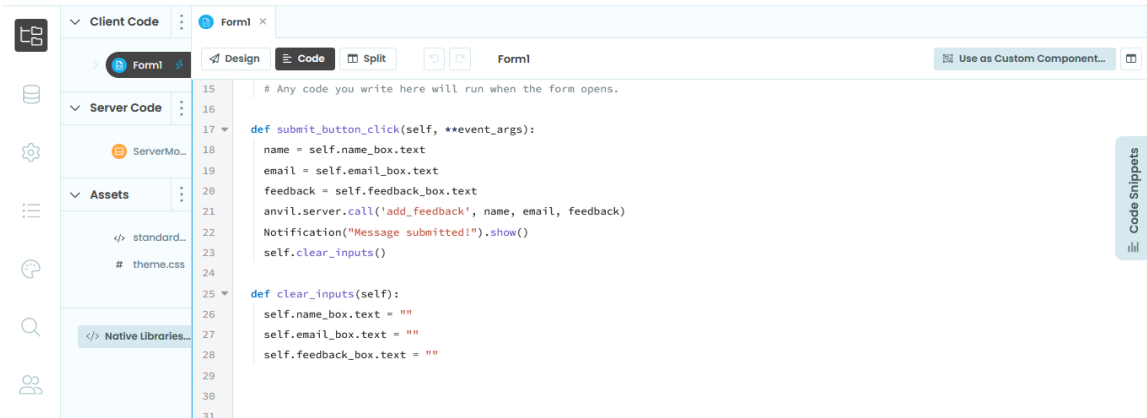
    if user is not None and user['link_key'] is not None:
        # Use bcrypt to hash the link key and compare the hashed version.
        # The naive way (link_key == user['link_key']) would expose a timing vulnerability.
        salt = bcrypt.gensalt()
        if hash_password(link_key, salt) == hash_password(user['link_key'], salt):
            return user

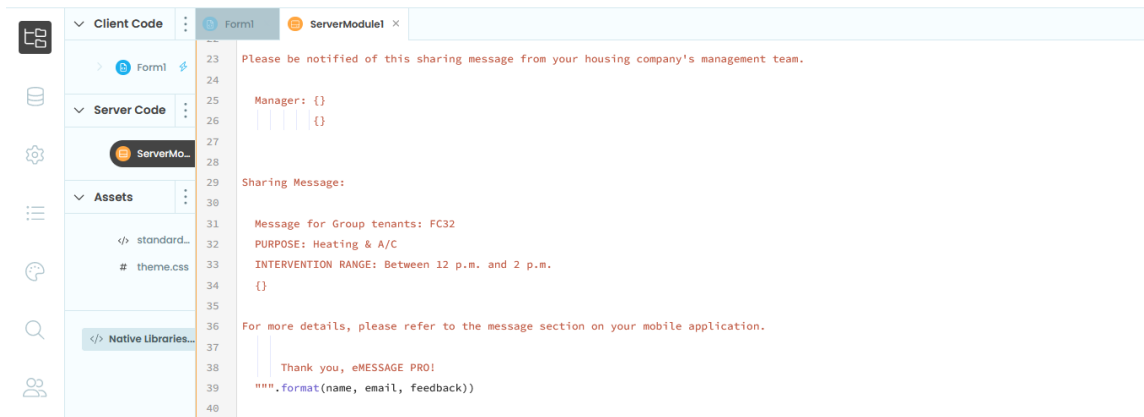
```



## Codage de la page de partage de message :







## 6.2 Essais & validation

Les essais qui ont été réalisés sont simples. Ils étaient constitués dans un premier temps d'essais de navigation dans les sous-systèmes élaboré. Lorsqu'un problème se pose, le code de programmation doit être revu avec précision et fidélité afin de le détecter, le corriger puis refaire le test. Lorsque le prototype fonctionne alors il est validé. Ensuite des tests d'envois de messages courriels ont été effectués afin de s'assurer que les messages aux locataires étaient bien reçus au courriel indiqué lors de la procédure de connexion. Cette vérification a été faite sur les messages de sécurité pour changer de mot de passe en cas d'oubli de l'utilisateur. Pour une utilisation prolongée ou pour un nombre accru de locataires, il faudra passer à une version payante supérieure à celle utilisée lors de ce projet de conception car cette version offre un nombre limité d'e-mail ou messages envoyés à l'application mobile, à savoir 1000 mails.

## **7 Conclusions et recommandations pour les travaux futurs**

En fonction de la toute première rencontre avec notre client, il a été question de définir tous ses besoins et de trouver une solution favorable à ses besoins. Toutes les rétroactions de notre client nous ont permis d'être plus précis dans notre processus de conception.

Plusieurs objectifs ont été fixés et atteints durant chaque étape de la phase de prototypage. Au fur et à mesure plusieurs fonctionnalités ont été mise à jour en fonction des besoins du client et aussi en tenant compte de toutes nos capacités techniques bien que nous fussions une équipe multidisciplinaire.

Tout ce que nous avons appris dans la gestion de notre projet nous seront très utile pour projets avenir ou encore dans le cadre d'un travail en groupe. Cela étant dit, les pistes qui restent intéressantes à explorer à l'avenir sont la liaison des deux interfaces présentées et l'ajout de plus d'onglets de navigation sur le site web avec des modules.

Si l'on avait suffisamment de temps, nous aurions pu terminer la liaison de l'application et du site web des gestionnaires. D'autre part, le site web des gestionnaires aurait pu être encore plus détaillé en termes de contenu. Il aurait été possible d'y ajouter des fonctionnalités d'accès à un calendrier des messages, des paramètres, des groupes de locataires ainsi qu'une banque de données avec des messages prédéfinis ou automatisés. Mais également acheter la licence pro de l'application afin de la commercialiser.

## 8 Bibliographie

Lien officiel de LCO : [Ottawa Community Housing \(och-lco.ca\)](http://OttawaCommunityHousing.org)

Anvil: <https://anvil.works/>

Thunkable:

[https://thinkable.com/?utm\\_term=thinkable&utm\\_campaign=US/UK/CA%20%20Search%20%20Brand&utm\\_source=adwords&utm\\_medium=ppc&hsa\\_acc=2723599162&hsa\\_cam=15000671177&hsa\\_grp=127293870903&hsa\\_ad=554377436828&hsa\\_src=g&hsa\\_tgt=kwd-360955260416&hsa\\_kw=thinkable&hsa\\_mt=e&hsa\\_net=adwords&hsa\\_ver=3&gclid=Cj0KCQjwxYOiBhC9ARIsANiEifaPV80PCpRAlloPMwnDXm6LtQ\\_yKleodLrfFdsKcw6kR\\_BR9dzFC-EaAm2eEALw\\_wcB#/](https://thinkable.com/?utm_term=thinkable&utm_campaign=US/UK/CA%20%20Search%20%20Brand&utm_source=adwords&utm_medium=ppc&hsa_acc=2723599162&hsa_cam=15000671177&hsa_grp=127293870903&hsa_ad=554377436828&hsa_src=g&hsa_tgt=kwd-360955260416&hsa_kw=thinkable&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQjwxYOiBhC9ARIsANiEifaPV80PCpRAlloPMwnDXm6LtQ_yKleodLrfFdsKcw6kR_BR9dzFC-EaAm2eEALw_wcB#/)