**E1**

*a. Expliquez à quoi sert le dépôt gerrit exactement. [ /1]*

Gerrit permet de faire de la révision de code qui est stocké sur un serveur Git. Elle permet entre autre de discuter du code, visualiser les différences entre les différentes versions, gérer les accès et bien plus.

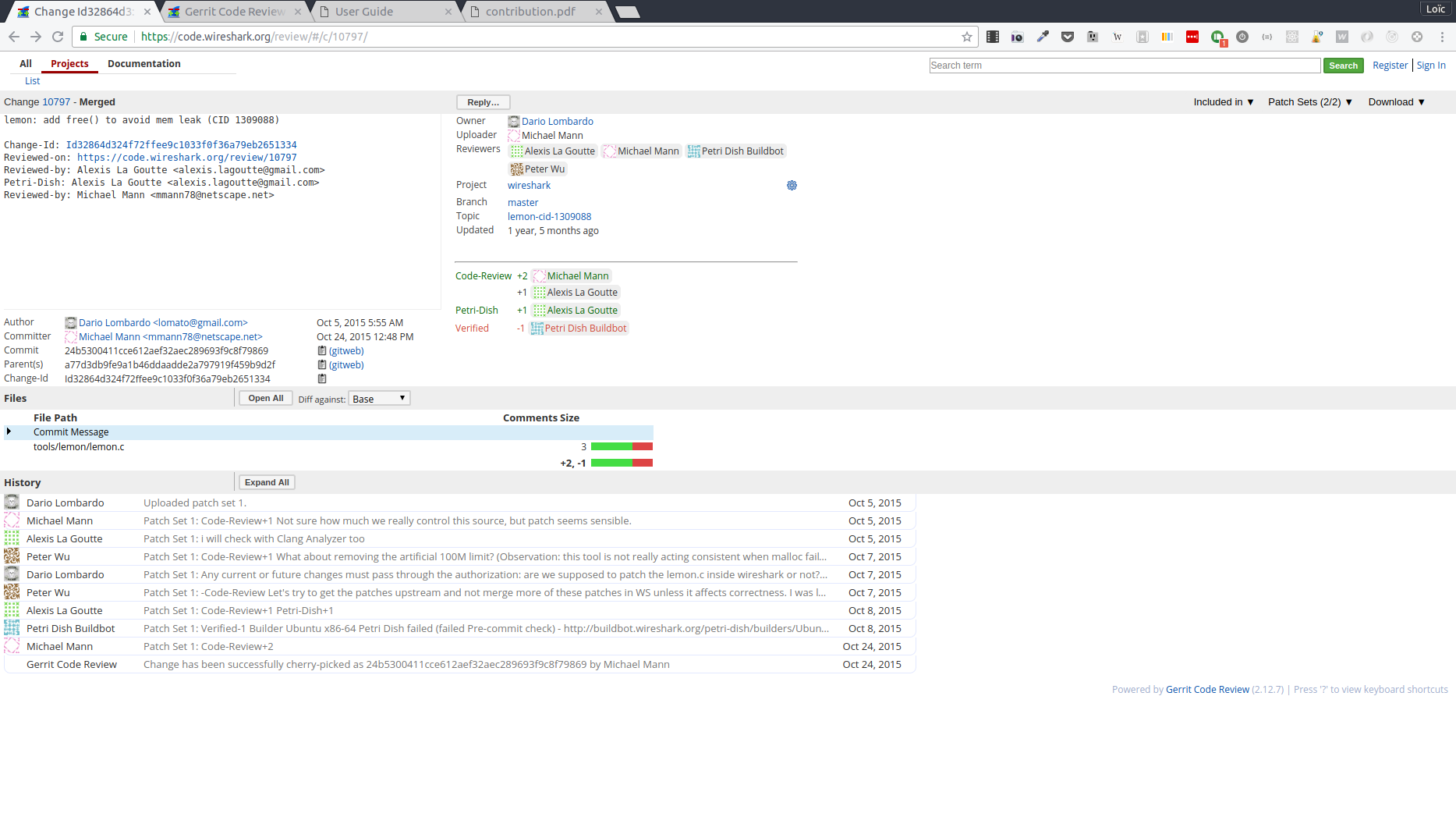
*b. Que signifient “open“, “merged” et “abandoned”? [ /1]*

* Open signigie qu’une modification a été poussé à un répertoire avec le “namespace” refs/for/. Si le refs/for/ n’est pas inclus dans le HEAD, le code est directement poussé vers le serveur final. Un changement marqué open indique que celui-ci doit être révisé avant d’être accepté.
* Merged signifie qu’un changement a été révisé et accepté et qu’il a été combiné avec le code final.
* Abandoned signifie que le changement a été abandonné puisqu’il n’était plus nécessaire.

*c. Qu'est-ce qu'un “patch”? [ /1]*

Un patch est un numéro dans le ref. Il correspond au ZZ dans refs/changes/XX/YYYY/ZZ. Le numéro de patch change lorsqu’une nouvelle version du code est téléchargée.

*d. Cherchez la révision technique Id32864d324f72ffee9c1033f0f36a79eb2651334.*

*e. Identifiez les fichiers qui ont été modifiés et décrivez brièvement ce que fait ce patch. [/1]*

Le fichier modifié est “tools/lemon/lemon.c”. Les modifications changement la vérification de la valeur “filebuf” dans le “if” et désallouent le buffer du fichier.

*f. Est-ce que le patch a été accepté? Comment est-ce que l’on peut voir ça? [/1]*

Oui. Son état est “merged” donc la modification de code a été acceptée et ajouté au projet.

*g. Identifiez les développeurs qui critiquent ce patch. [ /1]*

Alexis La Goutte, Michael Mann, Petri Dish Buildbot, Peter Wu, Michael Mann et Dario Lombardo.

*h. Identifiez les valeurs de la section “Code-Review” à droite. Que signifient ces valeurs? [ /1]*

+2 et +1 Les valeurs de la section “Code-Review” indiquent les personnes qui ont approuvé le code ou refusé le code

*i. Naviguez la partie “History” en bas, qu’est-ce qui s’est passé pour cette révision? Pourquoi cette partie est importante? [ /1]*

Cette section est importante puisqu’elle permet d’avoir un historique de toutes les actions qui ont été posées par les membres impliqués dans le changement. Dans la discussion ici, Micheal, Dario et Peter se demandent où le code review devrait être fait. Les autres approuvent les changements et proposent des modifications au code.

*j. Faites le même travail pour la révision technique « Ic34c9fd45740b834ce826d2ce083c615191bdef3 » du développeur “Michael Oed”. [/5]*

Fichier modifié : packaging/nsis/common.nsh

Change du texte

Le patch a été refusé. Le statut est « abandoned »

Les développeurs qui critiquent :

Evan Huus  
Pascal Quantin

Valeur dans la section code-review : -1

Les commentaires demandes des modifications pour que le «automatic updater» soit pris en compte, sinon ils demandent l’abandon. Finalement c’est abandonné.

**E2**

*a. Selon vous, est-ce que cet auteur contribue beaucoup au développement du logiciel*

*(selon l'ensemble de ses “commit”) en comparaison avec le committer de ce commit?*

*Comment est-ce que vous avez trouvé la réponse? [ /2 ]*

*Oui. Il a soumis plusieurs révisions qui ont étés acceptées. Il semble en avoir un nombre comparable au committer. On a trouvé la réponse en regagrdant l’historique des deux développeurs.*

b. Copiez le message complet de ce commit. [ /1 ]

Parent: a77d3db9 (PER: fix ASAN runtime error: left shift of negative value -1)

Author: Dario Lombardo <lomato@gmail.com>

AuthorDate: 2015-10-05 11:55:09 +0200

Commit: Michael Mann <mmann78@netscape.net>

CommitDate: 2015-10-24 16:48:09 +0000

lemon: add free() to avoid mem leak (CID 1309088)

Change-Id: Id32864d324f72ffee9c1033f0f36a79eb2651334

Reviewed-on: https://code.wireshark.org/review/10797

Reviewed-by: Alexis La Goutte <alexis.lagoutte@gmail.com>

Petri-Dish: Alexis La Goutte <alexis.lagoutte@gmail.com>

Reviewed-by: Michael Mann <[mmann78@netscape.net](mailto:mmann78@netscape.net)>

c. Pourquoi cette section est utile? [ /1 ]

Pour observer les différences entre deux versions.

d. Copiez les lignes de code qui ont été modifiées par l'auteur. (Cela inclut les lignes

enlevées et ajoutées) [ /1 ]

Parent: a77d3db9 (PER: fix ASAN runtime error: left shift of negative value -1)

Commit: Michael Mann <mmann78@netscape.net>

CommitDate: 2015-10-24 16:48:09 +0000

Reviewed-on: https://code.wireshark.org/review/10797

Reviewed-by: Alexis La Goutte <alexis.lagoutte@gmail.com>

Petri-Dish: Alexis La Goutte <alexis.lagoutte@gmail.com>

Reviewed-by: Michael Mann <mmann78@netscape.net>

a. Est-ce que ce commit ajoute une nouvelle fonctionalité ou résoud un bogue? Comment

est-ce que vous avez déterminé ça? [ /2 ]

Résout un bogue. Le titre l’indique.

**E3**

a. Trouvez le bogue #9887 (astuce: il y a une barre de recherche). Quel est le nom de la personne qui a trouvé le bogue? Est-ce que ce rapporteur est un être humain? [ /1 ]

Buildbot Builder

Ce n’est pas un être humain

b. Quel est le niveau d'importance de ce bogue? Comment est déterminé le niveau d'importance d'un bogue? Est-ce que ce bogue est critique à un tel point qu'il compromet le fonctionnement du logiciel? (dites pourquoi) [ /3 ]

High major

Il est déterminé en fonction de l’impact sur les utilisateurs.

Il ne compromet pas le fonctionnement global du logiciel, mais a un impact important sur les utilisateurs.

c. Combien de personnes on commenté ce problème? (à part la description originale) [ /2]

15 personnes

d. Est-ce que la communauté a confirmé la validité du bogue? Comment? [ /2 ]

Oui car le bug a été classé WORKSFORME

e. Est-ce que le bogue a été résolu? Comment est-ce que vous avez déterminé ça? [ /2]

Non. Comme le bogue est classé WORKSFORME, les tentatives pour reproduire le bogue n’ont pas toujours étées fructueuse et on ne sait pas comment le code pourrait produire une telle réaction. Si jamais le bogue réapparait, il pourrait être réouvert

**E4**

a.Expliquez l'utilité de “Buildbot“ dans le contexte d'un projet “open source“. [ /2]

Buildbot permet l’automatisation des builds. C’est intéressant dans un cas de logiciel open source car il y a beaucoup de nouveau code ajouté constamment, donc ça requiert une gestion avancée des builds. Buildbot simplifie donc cette gestion et élimine le besoin d’avoir un humain qui doit gérer cela en permanence.

b. Identifiez les différents “builders“ disponible pour ce projet cliquant sur “Builders” en

haut de la page. [ /1]

Clang Code Analysis  
OSX 10.6 x64  
Ubuntu 16.04 x64  
Visual Studio Code Analysis  
Windows 8.1 x86  
Windows Server 2012 R2 x64

a. Que signifient les couleurs vert, rouge et orange? [ /1.5 ]

rouge = build failed  
jaune = build pending  
vert = build successful

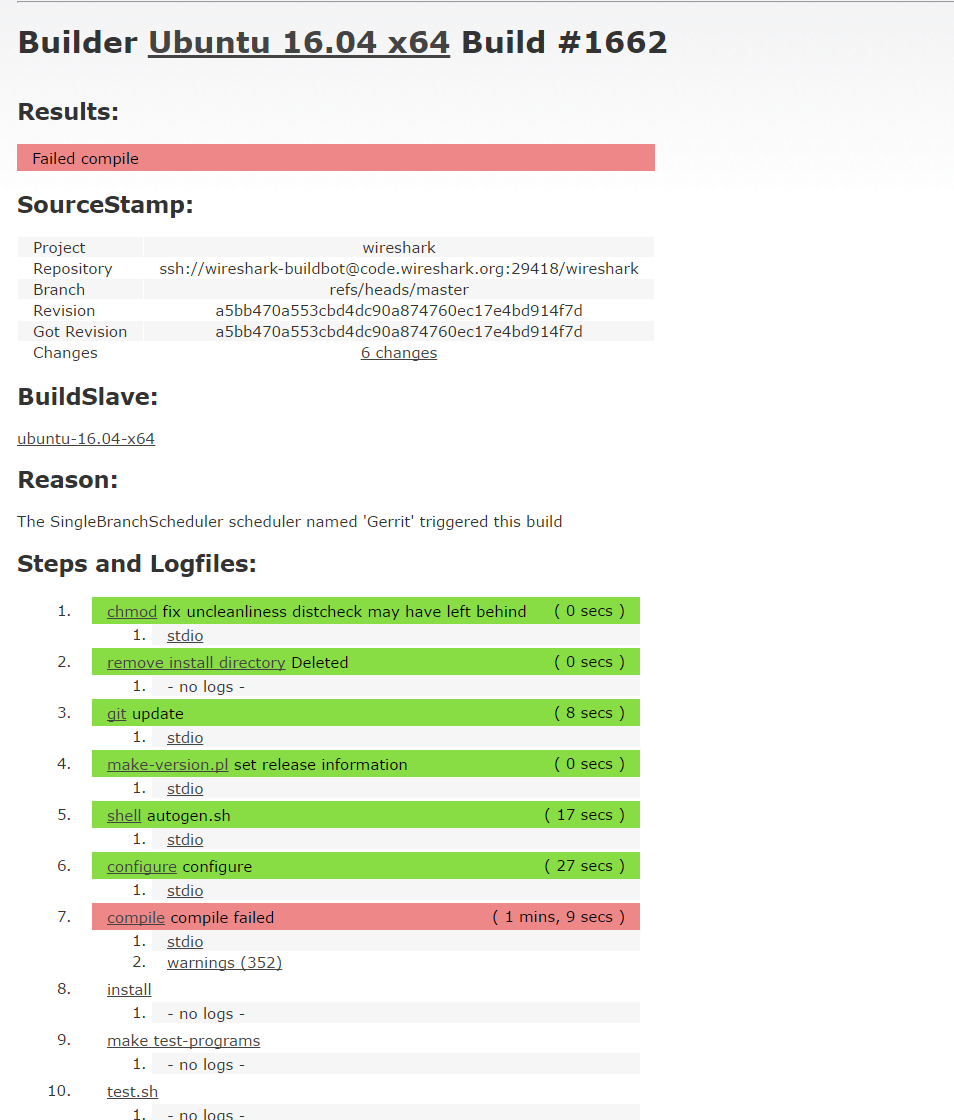
b. Que signifient les différentes colonnes? [ /1.5 ]

les différents builders (listés dans la sous question E4.a)

c. Identifiez un build échoué ou avec des avertissements, faites un screenshot et expliquez

la cause des problèmes en cliquant sur le lien pour voir des messages du build. [ /2 ]

Le build a échoué car la compilation n’a pas réussi, soit l’étape 7. Il y a 352 warnings dans le code, disponible à cette adresse : https://buildbot.wireshark.org/wireshark-master/builders/Ubuntu%2016.04%20x64/builds/1662/steps/compile/logs/warnings%20%28352%29



Le build a échoué car la compilation n’a pas réussi, soit l’étape 7. Il y a 352 warnings dans le code, disponible à cette adresse : https://buildbot.wireshark.org/wireshark-master/builders/Ubuntu%2016.04%20x64/builds/1662/steps/compile/logs/warnings%20%28352%29

d. Quel est le numéro de la révision technique de ce build échoué, et qui est le

développeur? [ /2]

Revision a5bb470a553cbd4dc90a874760ec17e4bd914f7d

Martin Kaiser  
Michael Mann

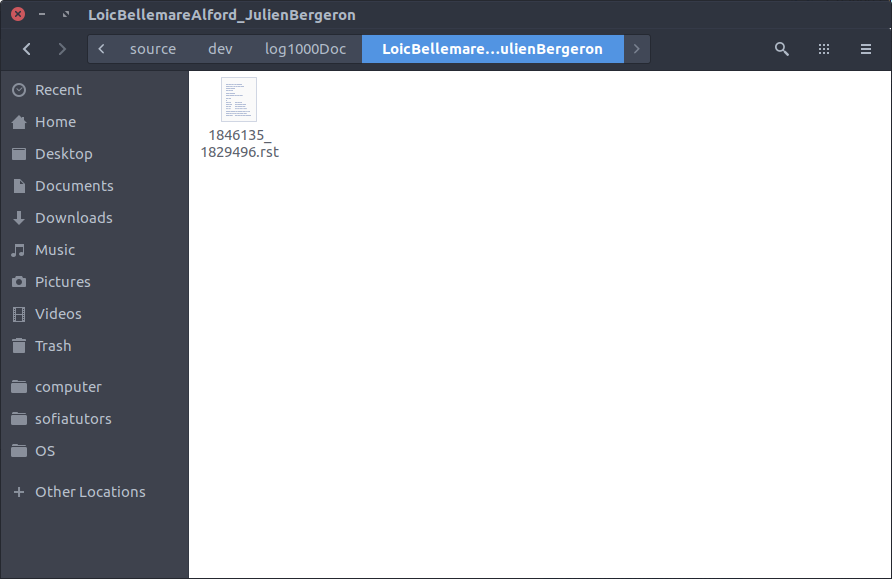
**E5**

*Copiez le résultat de la commande « git clone » dans votre rapport. [2]*

Cloning into 'ring-project'...  
remote: Total 929 (delta 0), reused 929 (delta 0)  
Receiving objects: 100% (929/929), 246.01 KiB | 0 bytes/s, done.  
Resolving deltas: 100% (480/480), done.  
Checking connectivity... done.

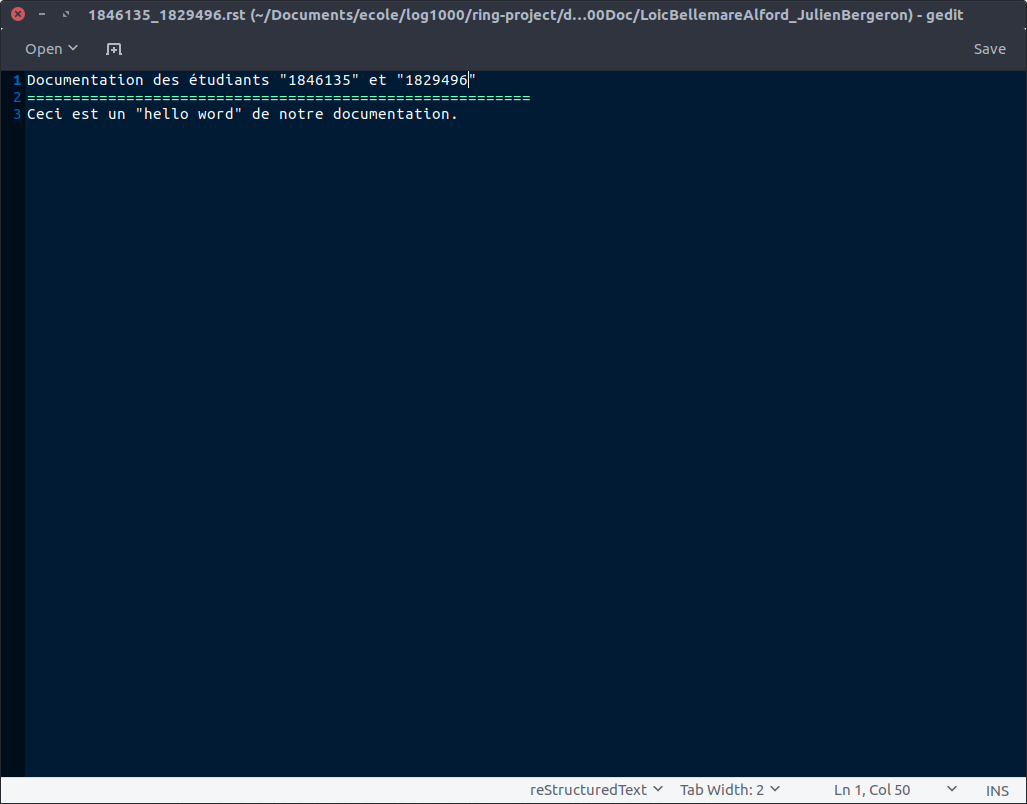
*a. Dans le dossier que vous avez créé, ajoutez un fichier dont l’extension est « .rst » qui*

*contiendra votre documentation en respectant le format du nom de fichier suivant*

*« matriculeEtudiant1\_matriculeEtudiant2.rst ». Faites une capture d’écran du fichier*

*ajouté dans votre rapport. [1]*

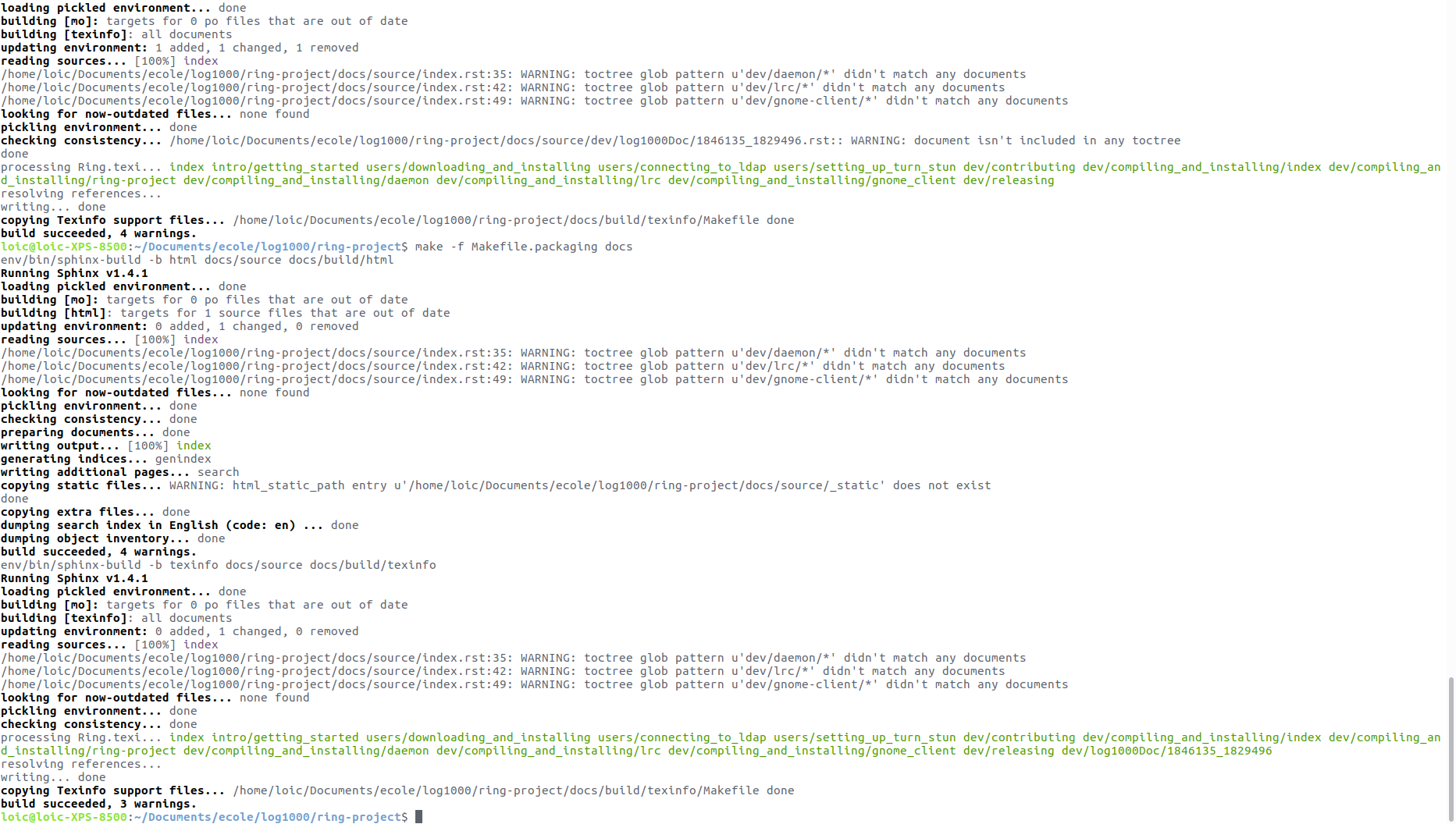
b. Insérez dans ce fichier le texte suivant : [1]



*c. Revenez à la ligne de commande et exécutez « make -f Makefile.packaging docs »*

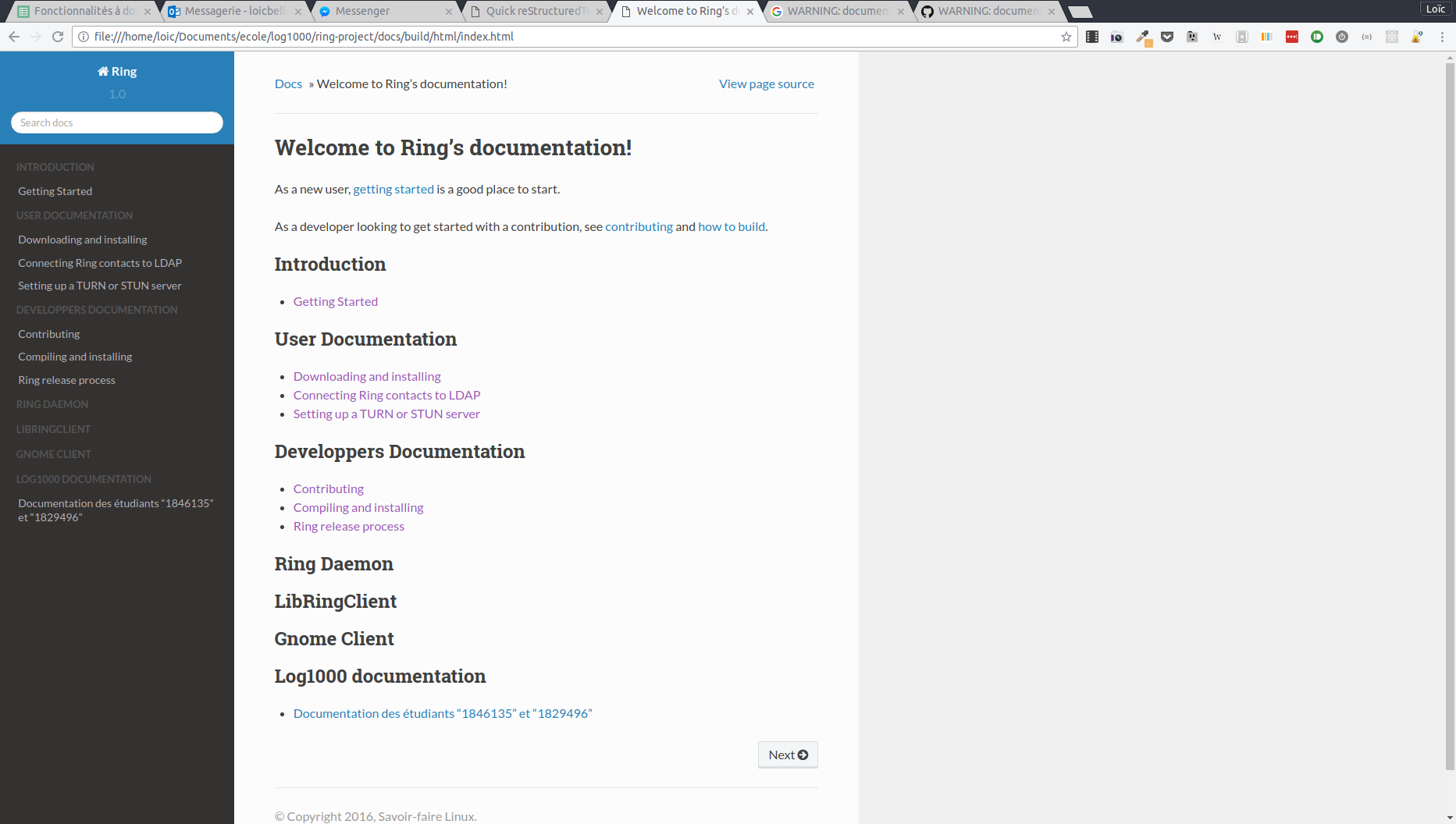
*comme indiqué précédemment. Faites une capture d’écran du résultat de cette*

*commande (make ...) dans votre rapport. [2]*

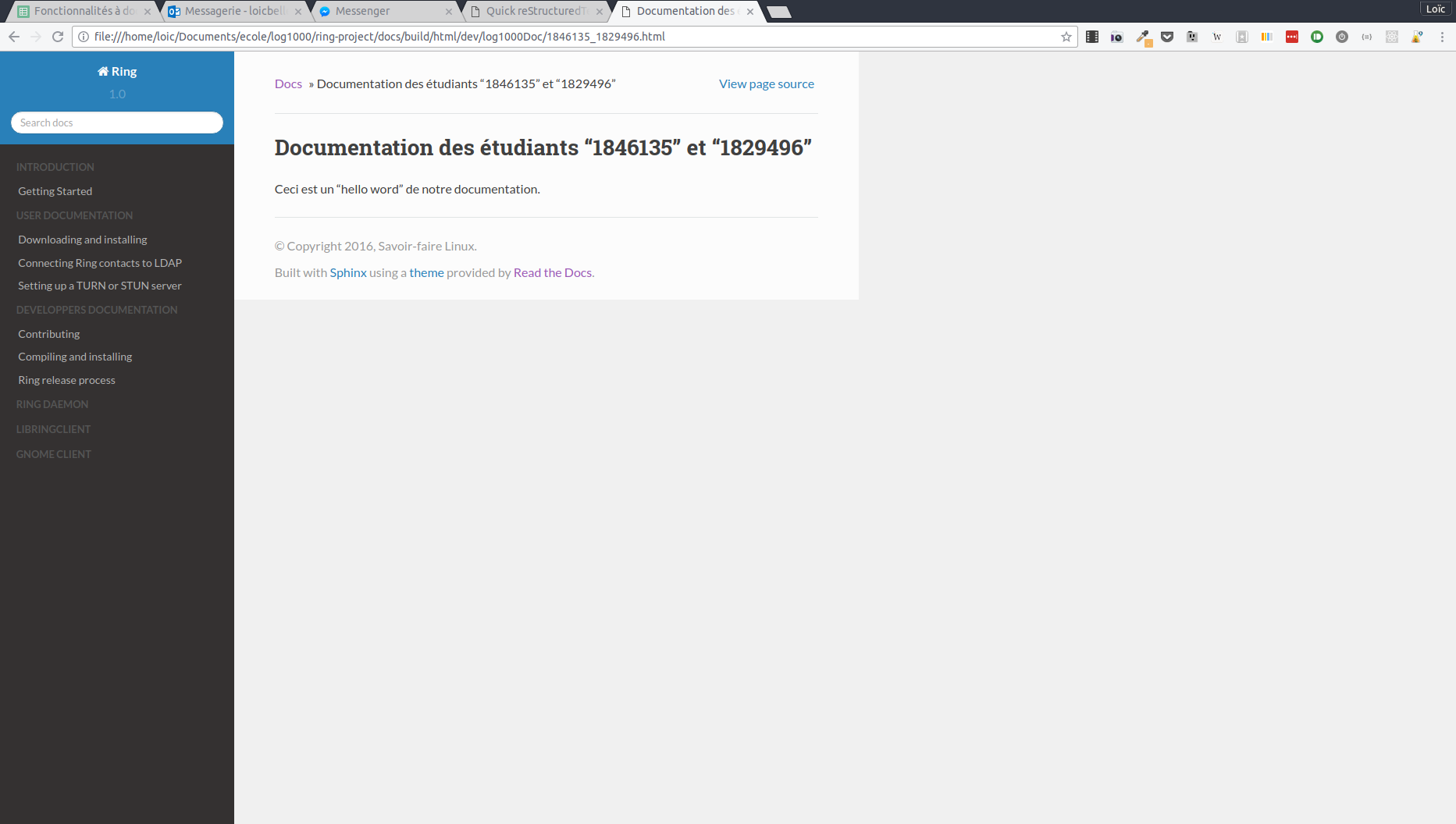


*d. Ensuite, comme indiqué précédemment, ouvrez la page « index.html » pour voir vos*

*modifications. En bas de page, vous pouvez trouver un lien vers votre documentation*

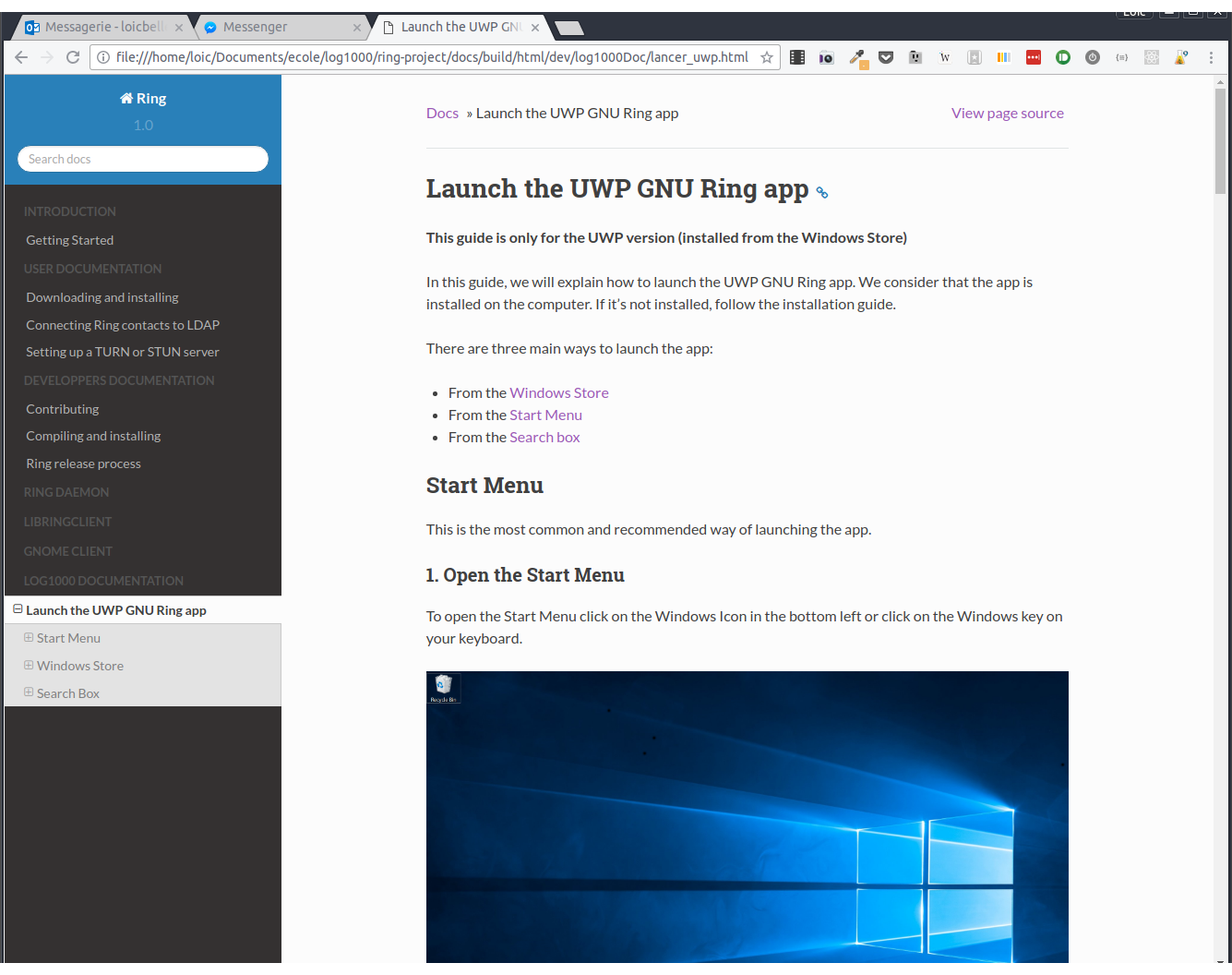
*qui est ajouté. Faite une capture d’écran du lien ajouté dans votre rapport. [2]*

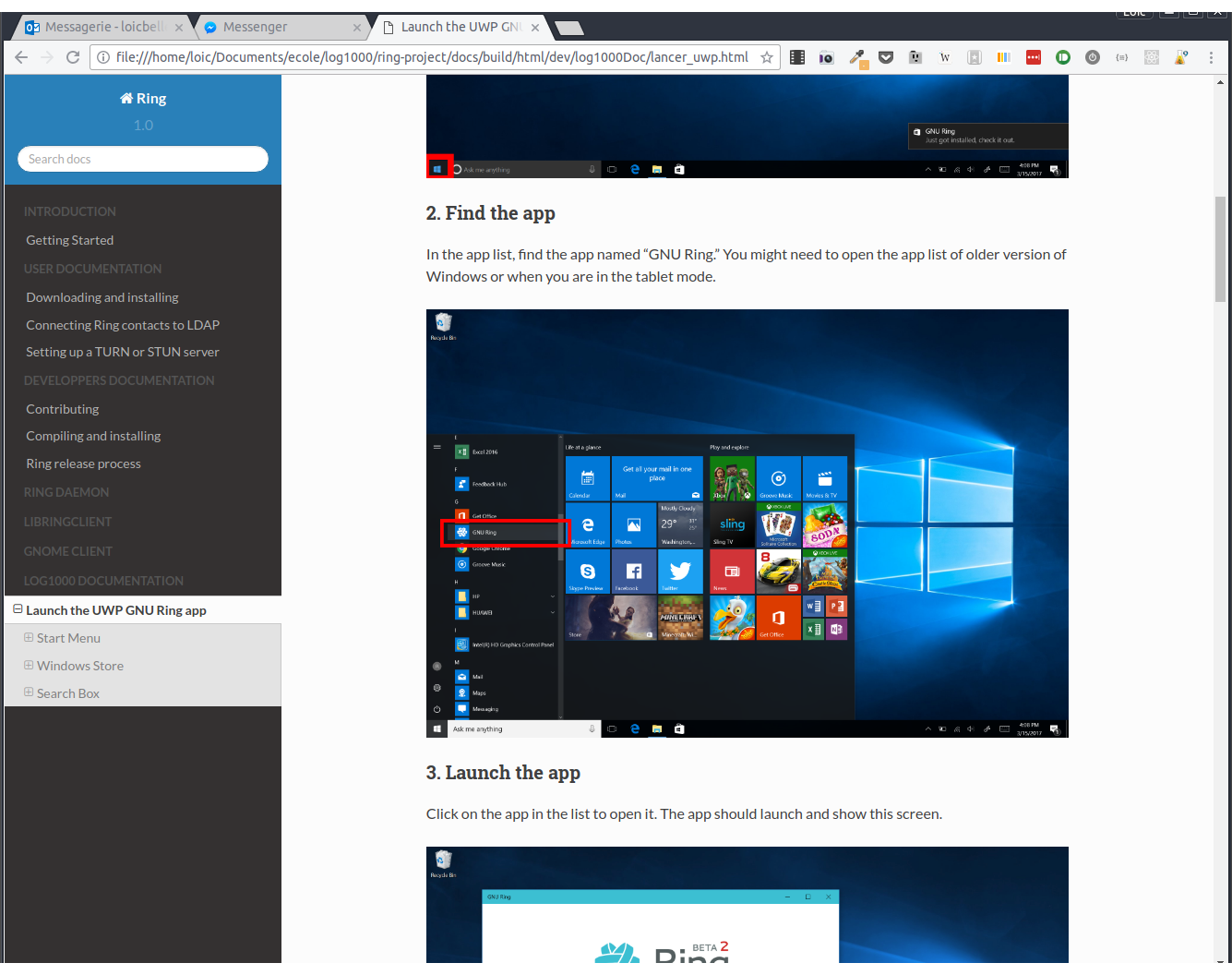
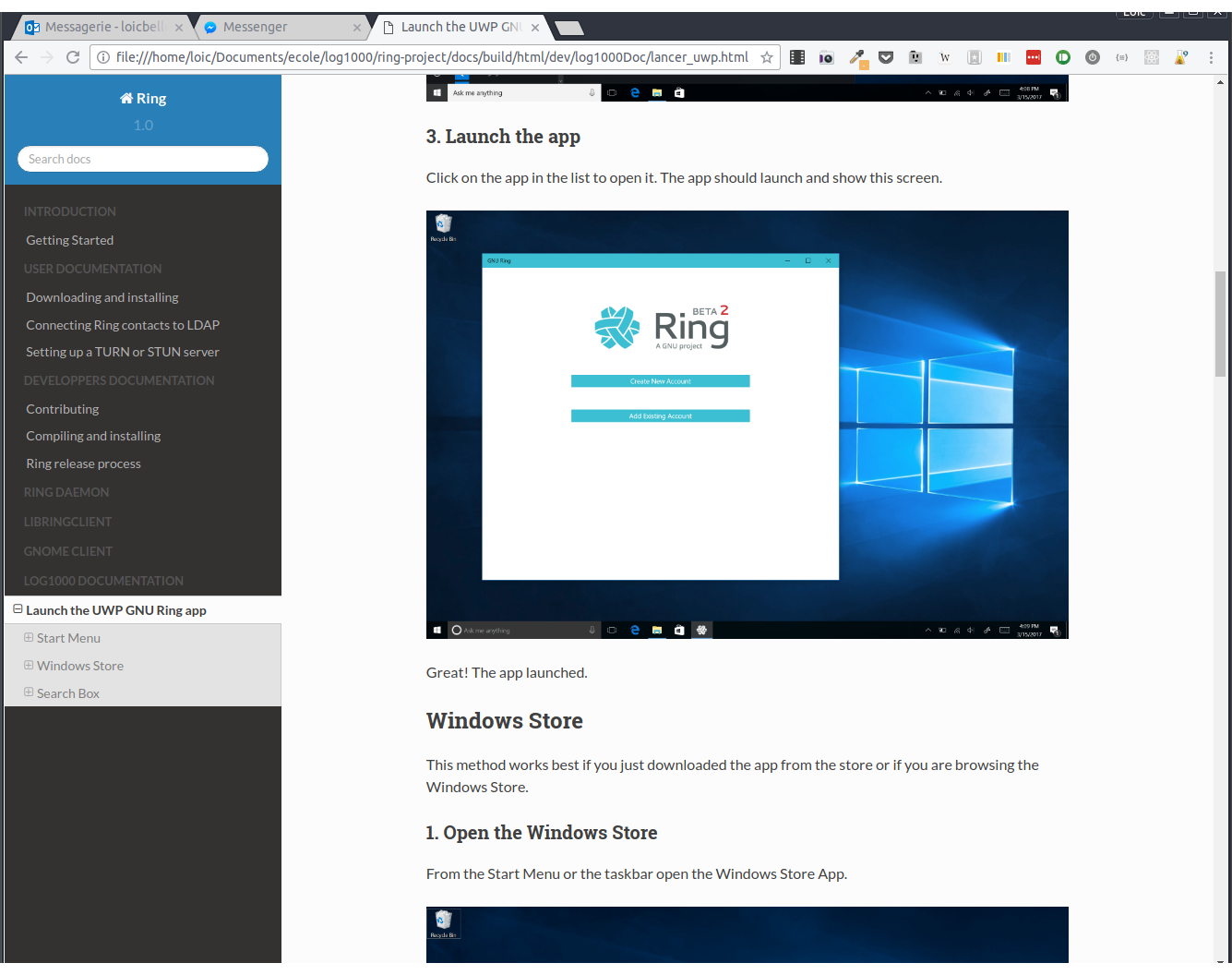
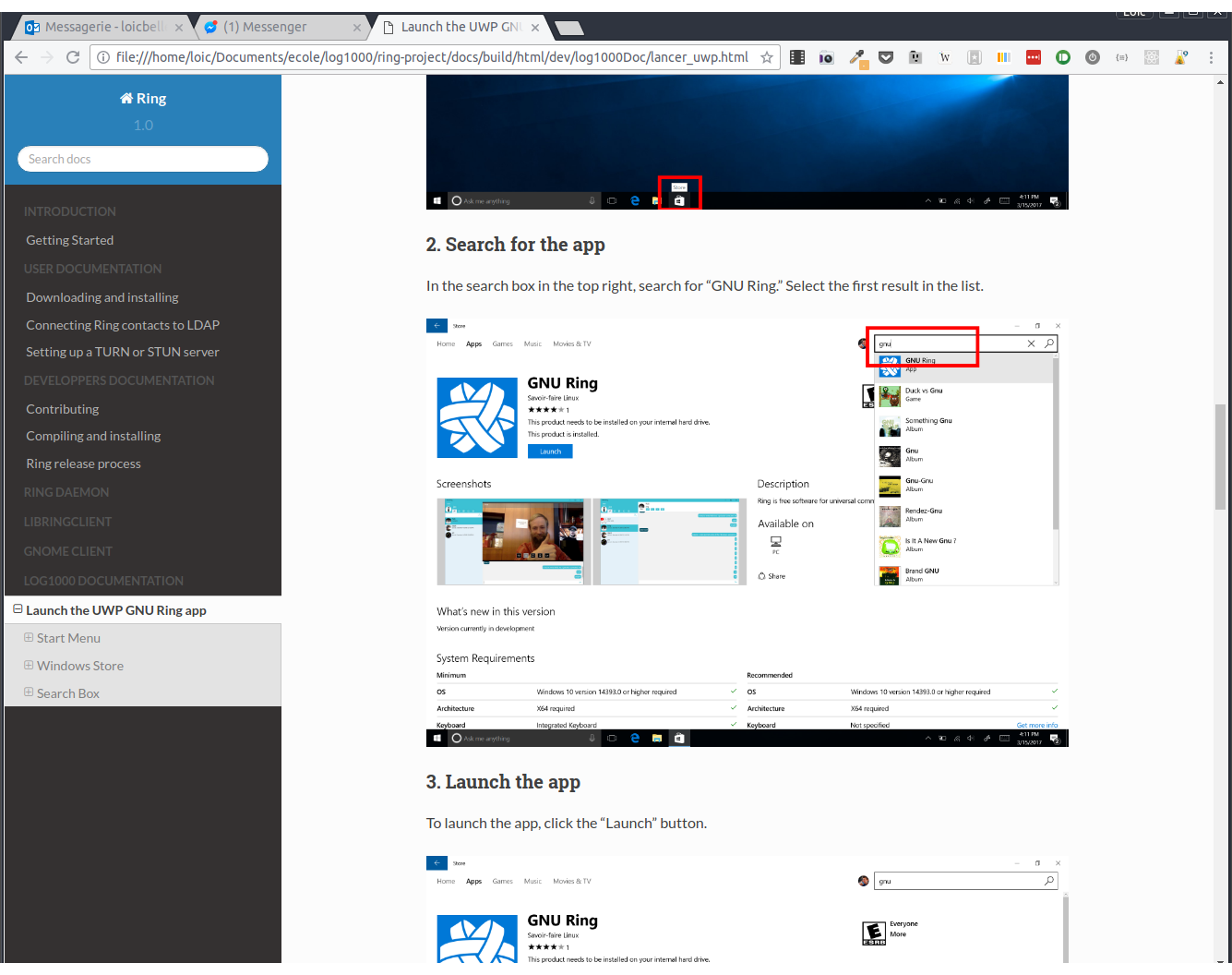
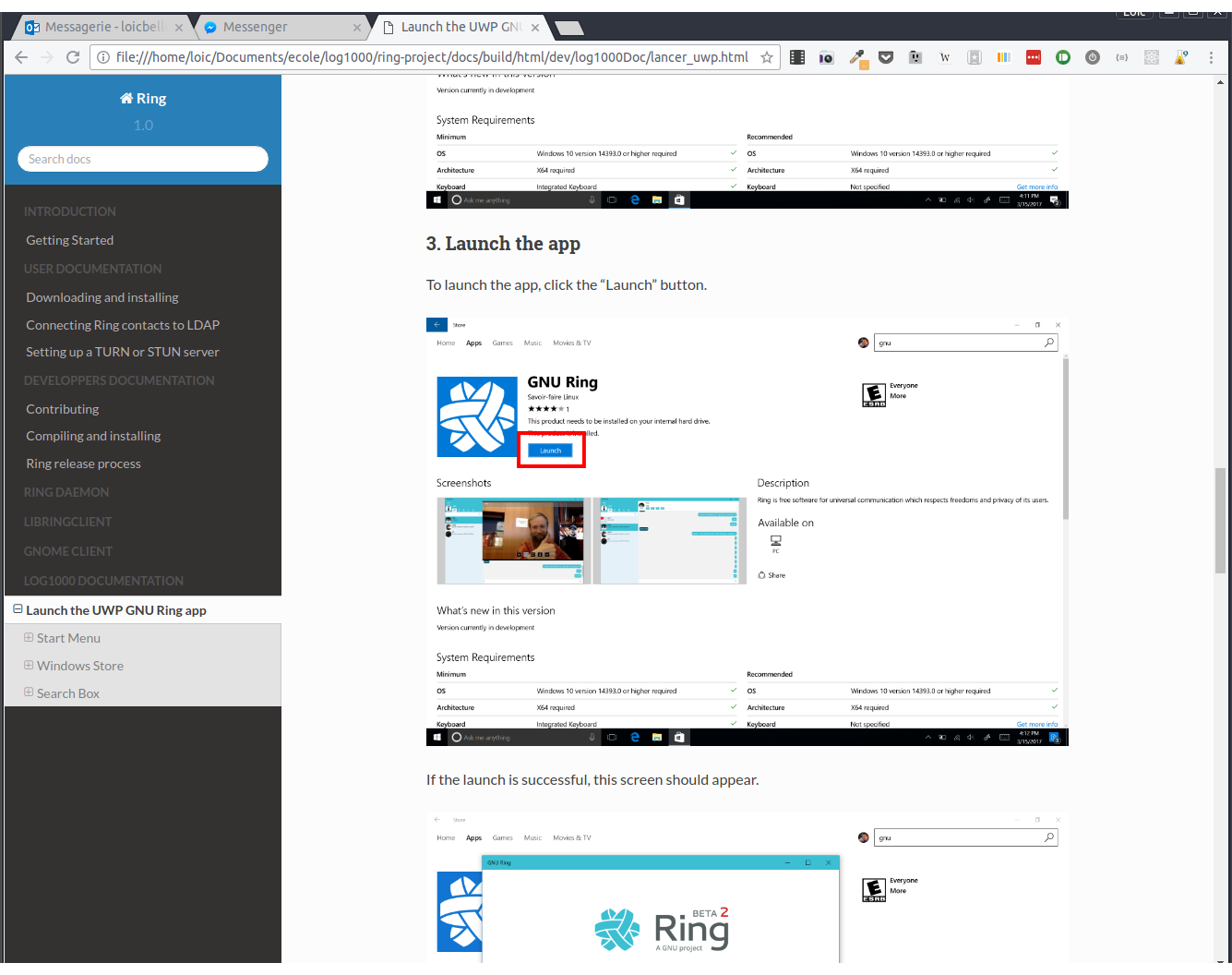
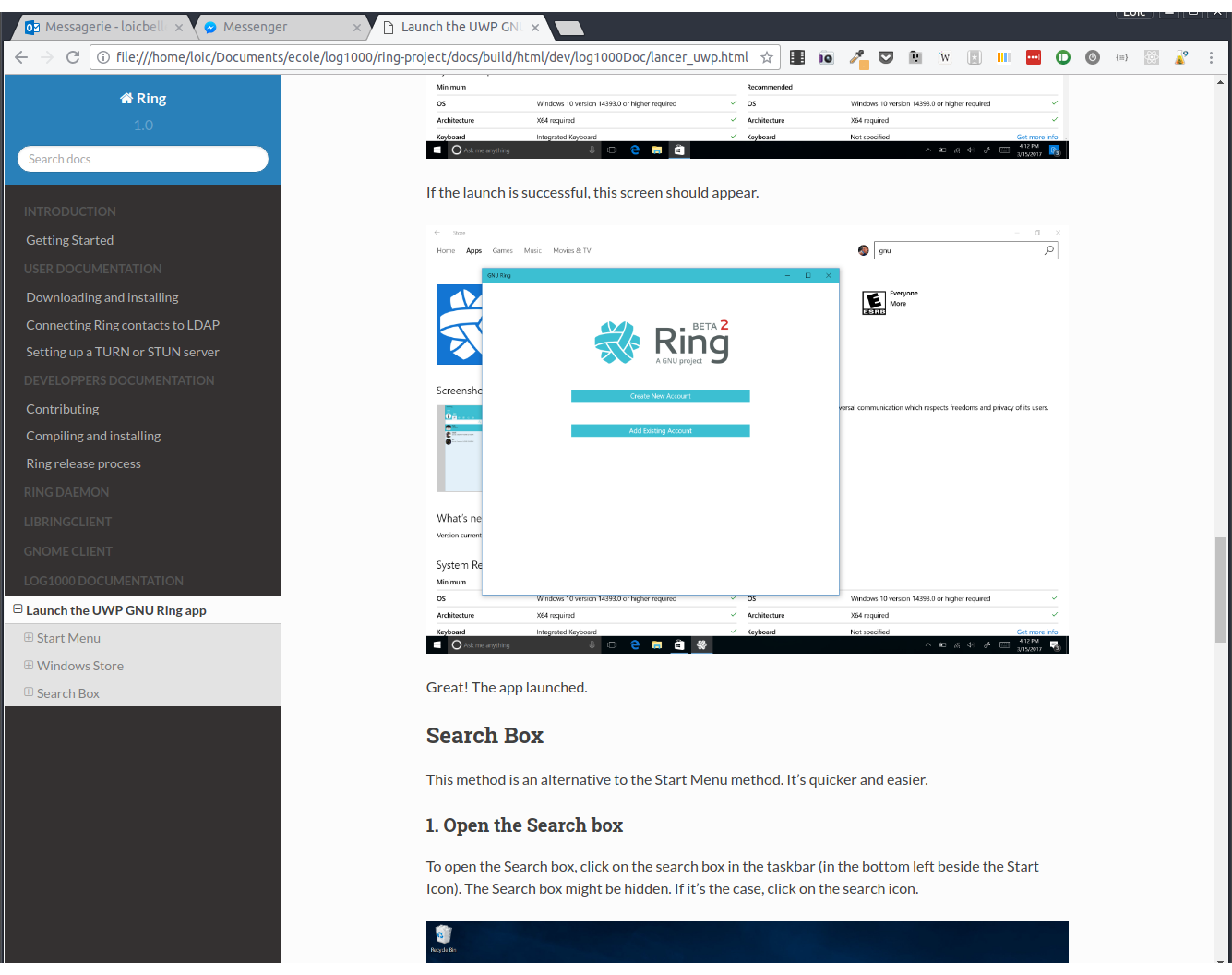
*e. Cliquez sur l’un de ces deux liens pour voir votre modification comme indiquée sur la*

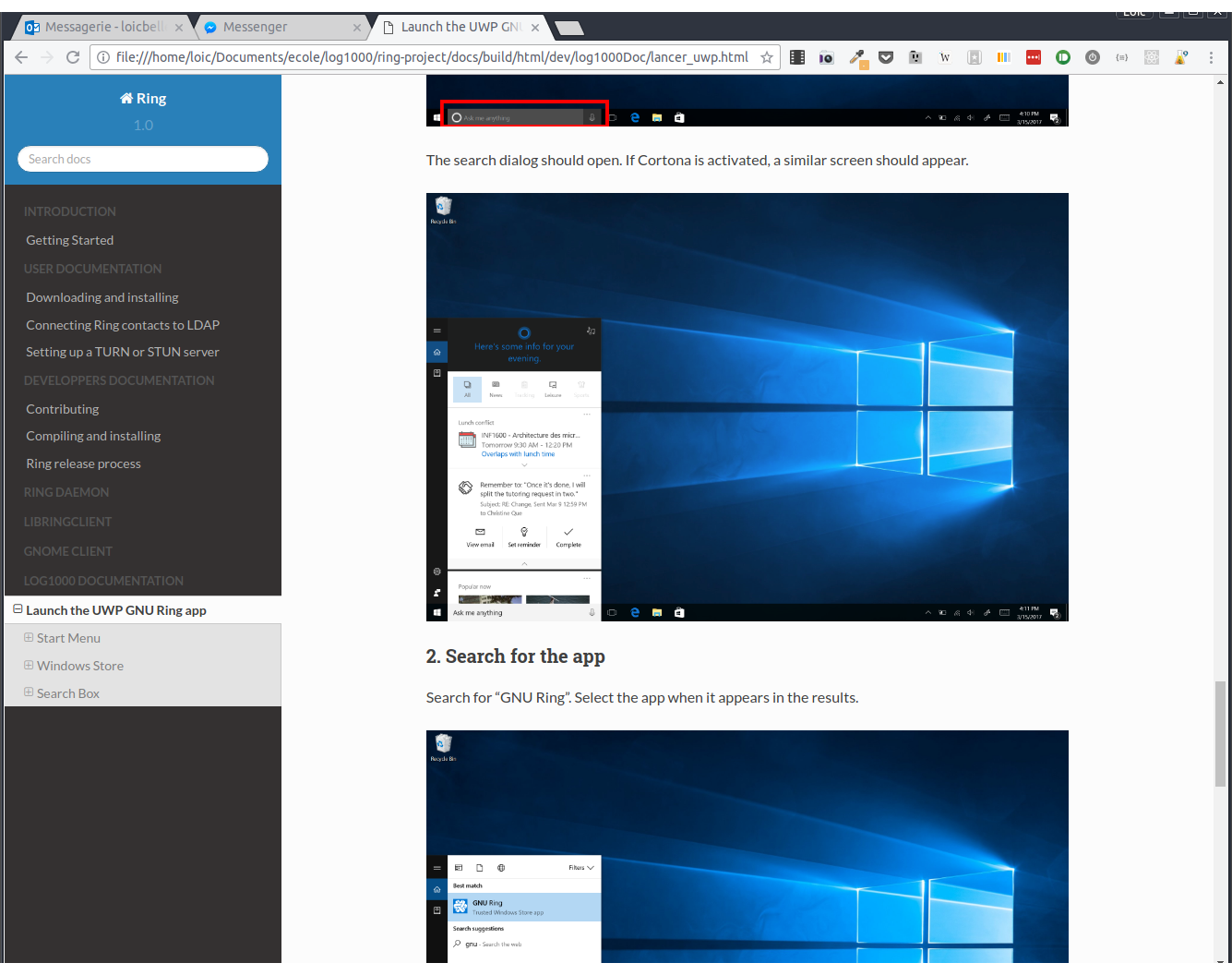
*figure suivante. Faites une capture d’écran de la page créée dans votre rapport. [1]*

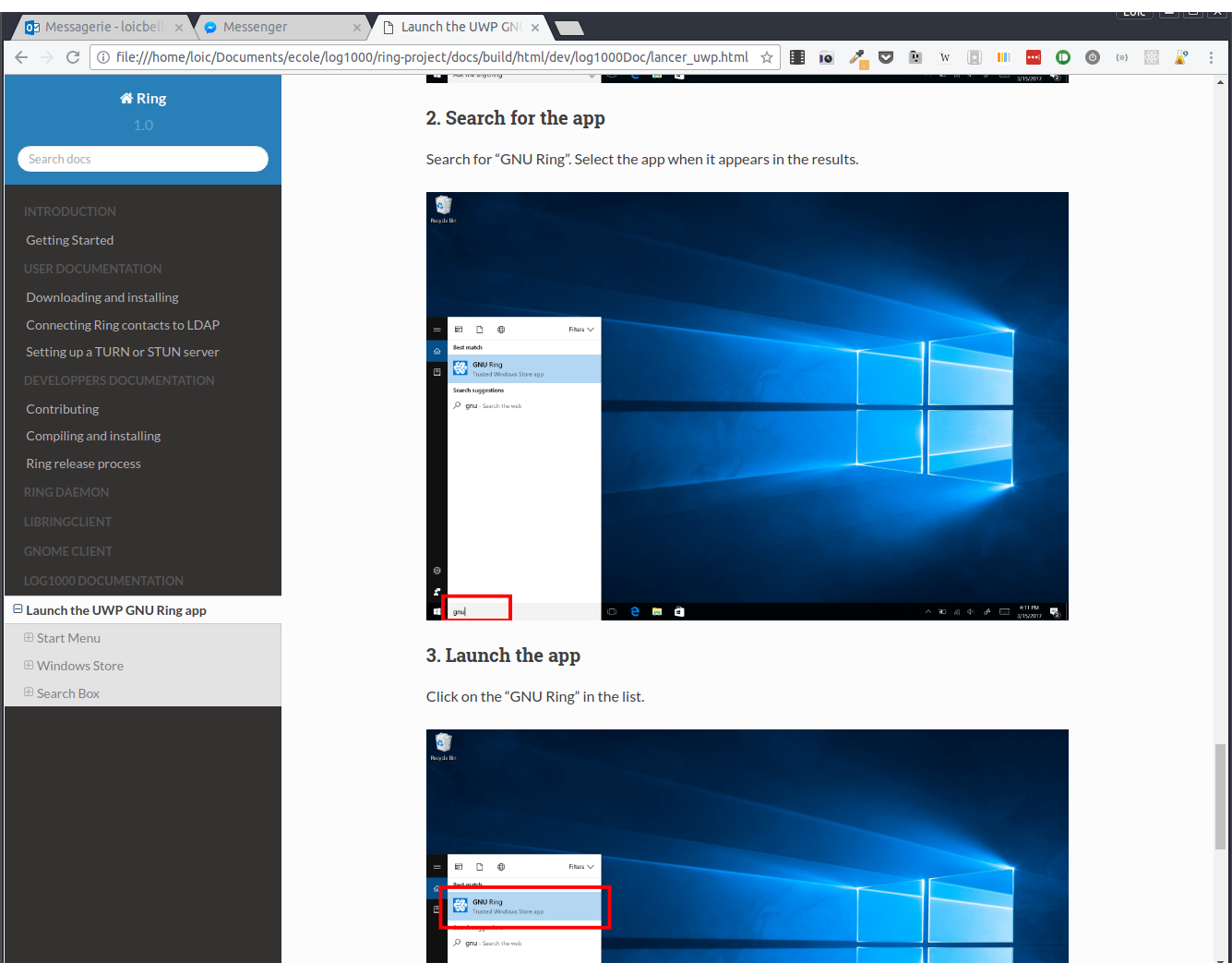
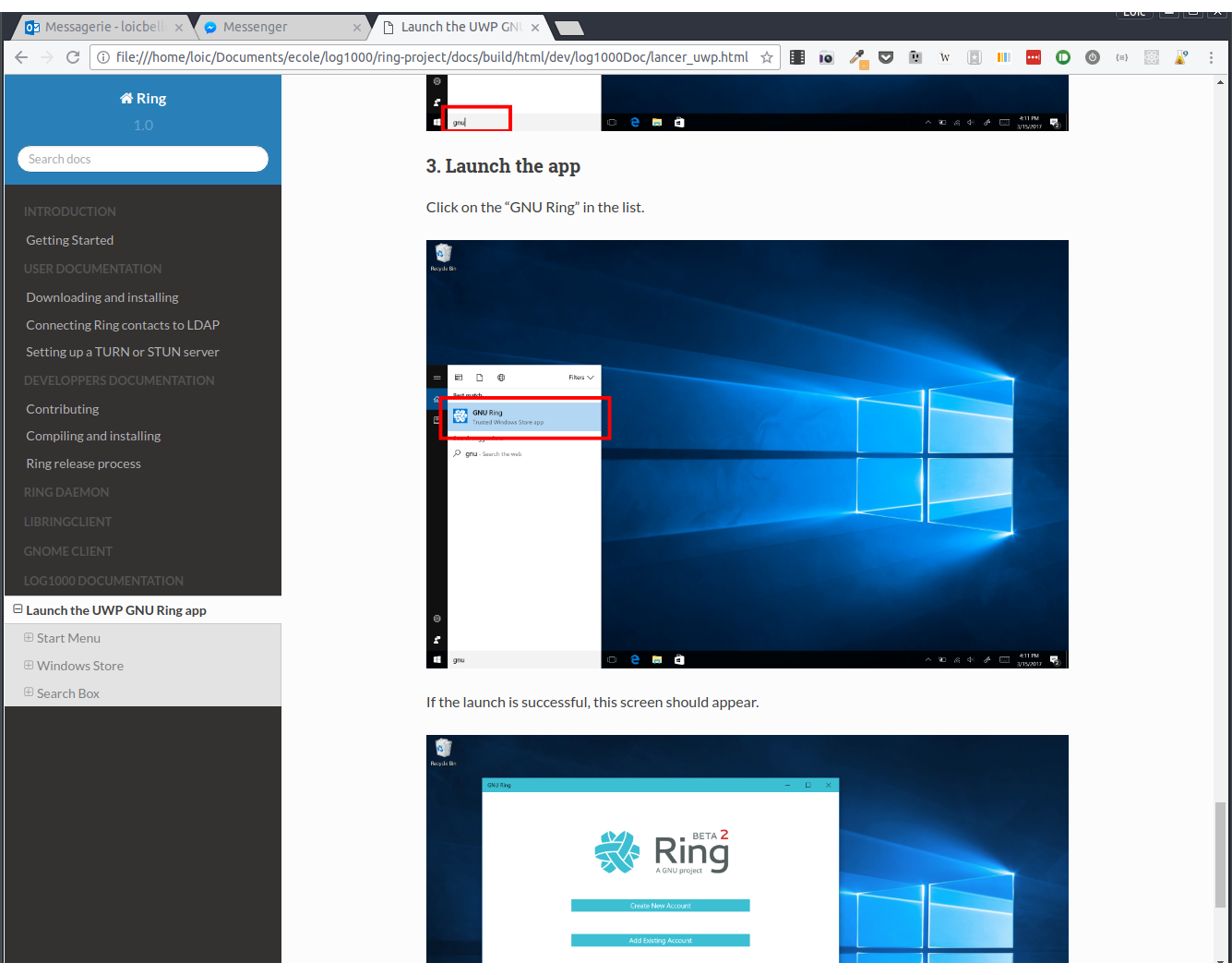
*Une fois vous avez terminé votre documentation, faites une capture d’écran de la page web que*

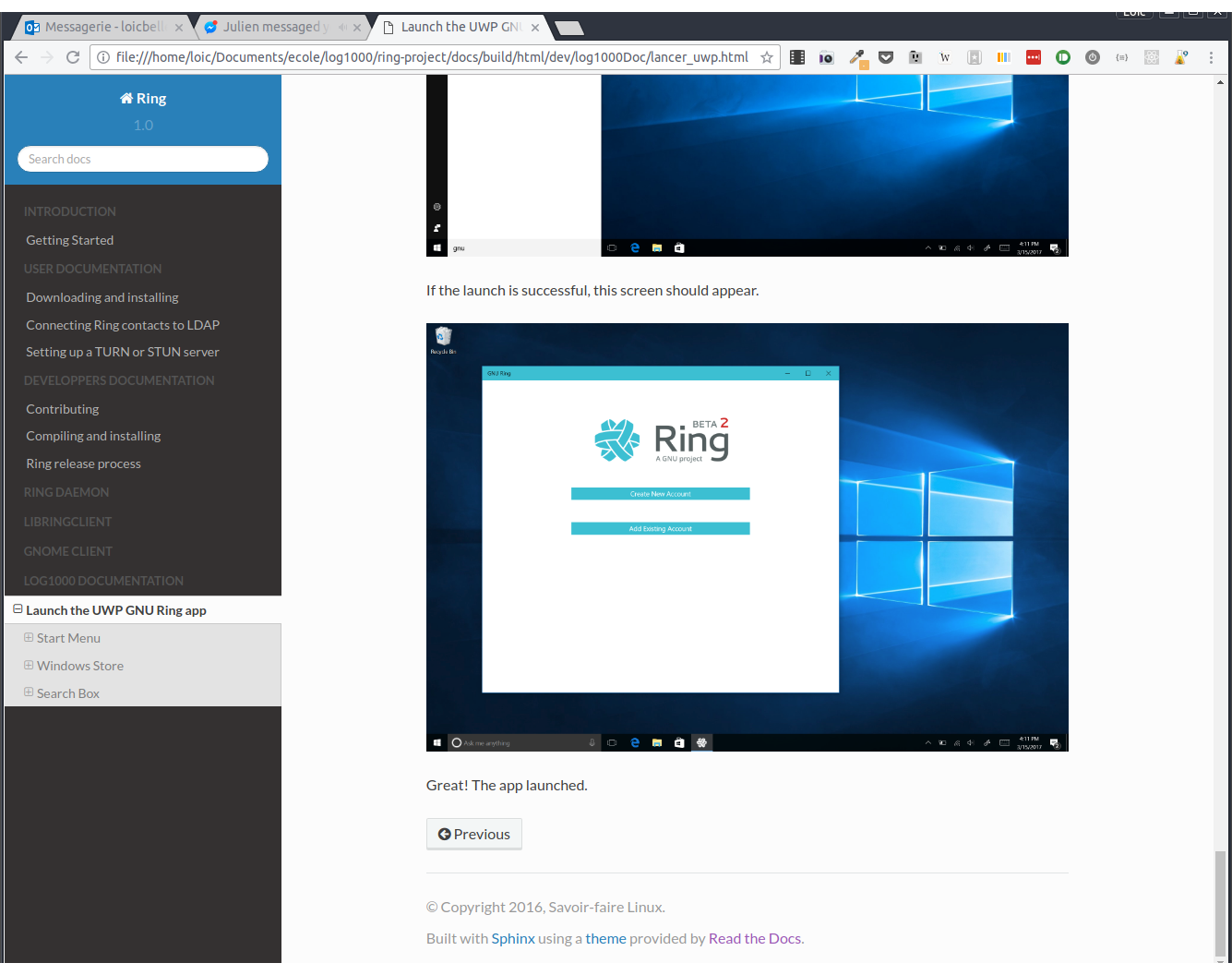
*vous avez créée (la page de documentation). Notez que vous serez évalué aussi par la qualité*

*de votre documentation. [10]*









*Prenez des captures d’écrans de la procédure de révision technique (chacune des étapes du*

*tutoriel joint à l'énoncé du TP, et aussi des captures d'écrans pour les feedbacks que vous allez*

*probablement recevoir sur votre contribution) pour le rapport du TP. [5]*

