

Lockatme  
A screen lock with facial recognition abilities  
Final report

David Anandanadaradja, Sagar Gueye, Bruno Inec,  
Matthieu Kirschleger, Pierre-Louis Sergent

25 septembre 2018

# Table des matières

<b>1</b>	<b>Antériorité</b>	<b>2</b>
1.1	Présentation des membres du groupe . . . . .	3
1.1.1	Contexte . . . . .	3
1.1.2	Organisation et membres . . . . .	3
1.1.3	Compétences . . . . .	3
1.2	Présentation du projet . . . . .	4
1.2.1	Buts . . . . .	4
1.2.2	Motivations . . . . .	4
1.2.3	Linux . . . . .	4
1.2.4	Open Source . . . . .	4
<b>2</b>	<b>Stratégie de développement</b>	<b>5</b>
2.1	Méthodologie voulue . . . . .	6
2.2	Application réelle au projet . . . . .	6
<b>3</b>	<b>Phase du projet – distribution des tâches</b>	<b>7</b>
3.1	Rétrospective - diagramme de Gantt specification . . . . .	8
3.2	Répartition réalisée . . . . .	8
3.2.1	Difficultés rencontrées . . . . .	8
3.2.2	Chronologie . . . . .	9
3.2.3	Liste des tâches approximative . . . . .	9
<b>4</b>	<b>Technologies utilisées</b>	<b>10</b>
4.1	Liste des technologies explorées . . . . .	11
4.2	Technologie utilisée – point technique . . . . .	11
4.2.1	Xlib – python . . . . .	11
4.2.2	Difficulté rencontrée - multithreading . . . . .	11
<b>5</b>	<b>Version finale</b>	<b>12</b>
5.1	Présentation . . . . .	13
5.2	Utilisation - mode d'emploi . . . . .	13
<b>6</b>	<b>Amélioration possible</b>	<b>14</b>
6.1	Interface graphique . . . . .	15
6.2	Le futur de lockatme . . . . .	15

# Chapitre 1

## Antériorité

## 1.1 Présentation des membres du groupe

### 1.1.1 Contexte

Dans le cadre de notre DUT Informatique à l'IUT Lyon 1, nous sommes tenus de réaliser un projet tuteuré durant le second semestre. Ce projet s'étendant également sur le troisième semestre, il a pour but de répondre à une problématique précise et de mettre en oeuvre les compétences acquises au cours de la formation. Il a aussi vocation à faire découvrir de nouveaux domaines et il nous permettra d'élargir nos savoirs à travers une auto-formation.

Ce projet se découpe en deux axes :

- Rédaction du cahier des charges (second semestre)
- Réalisation du projet en lui même (troisième semestre)

Malgré une liste de sujets proposés, notre groupe a voulu suivre ses propres motivations (présentées plus loin dans ce document) et a choisi de proposer un sujet à M. Vidal. L'intitulé est le suivant : Verrouillage et déverrouillage d'écran par reconnaissance faciale sous Linux.

### 1.1.2 Organisation et membres

L'équipe chargée de ce projet est constituée

- Tuteur du projet : M. Vincent VIDAL
- Chef du projet : M. Bruno INEC
- Membres : M. David ANANDANADARADJA, Mme Sagar GUEYE, M. Matthieu KIRSCHLEGER et M. Pierre-Louis SERGENT

### 1.1.3 Compétences

Notre projet comporte deux contraintes principales : il nécessite une bonne connaissance du langage Python et une maîtrise de Linux. L'impulsion de ces choix vient en grande partie du chef de projet qui possède une expérience importante dans ces deux domaines. David et Pierre-Louis possèdent quant à eux une expérience modérée dans l'utilisation de Linux (distribution Arch). L'ensemble des compétences individuelles est résumé ci-après :

#### **Python :**

- Confirmé : Bruno INEC
- Intermédiaire : Pierre-Louis SERGENT
- Débutant : Sagar GUEYE, Matthieu KIRSCHLEGER, David ANANDA

#### **Linux :**

- Confirmé : Bruno INEC
- Intermédiaire : David ANANDA, Pierre-Louis SERGENT
- Débutant : Sagar GUEYE, Matthieu KIRSCHLEGER

Comme le montre le listing précédent, les compétences du groupe sont très disparates. Cela peut apparaître comme une contrainte, mais en réalité cela constitue une véritable opportunité pour tous les membres. Ils vont ainsi pouvoir se former dans les domaines ci-après. Ils sont essentiels pour la suite des études et pour le milieu professionnel.

- Programmation : Linux, Python
- Rédaction cahier des charges :  $\text{\LaTeX}$
- Travail en équipe : réunion, communication, CI, modèle de développement

Nous étions donc motivés pour nous lancer dans un sujet avec nombre d'inconnus mais qui allait être fort enrichissant.

## **1.2 Présentation du projet**

### **1.2.1 Buts**

Le but premier de l'application est de déverrouiller un écran d'ordinateur, à l'aide d'une caméra, par reconnaissance faciale. Cependant cela implique de mettre en place un verrouillage d'écran sous Linux. Les URS spécifiques seront décrit plus tard dans ce document.

### **1.2.2 Motivations**

Trois membres du groupe utilisent Arch Linux qui est une distribution minimale de Linux. Le fait de quitter Windows leur a permis de pleinement se concentrer sur la machine à un plus bas niveau, avec tous les avantages de liberté qu'offre une plateforme open source, mais aussi toutes les contraintes qui sont très formatrices et qui forcent à se pencher d'avantage sur le fonctionnement de ce système d'exploitation. Les trois utilisateurs cherchaient une manière de verrouiller/déverrouiller leur écran de manière sécurisée. Et l'idée de ce projet a fleuri suite à un article présent dans le magazine Linux Magazine/France n°203 : "Mettez en place un système de reconnaissance faciale".

### **1.2.3 Linux**

Le développement du logiciel se fera sur Linux. Un tel projet sur Windows aurait été bien plus difficile concernant l'implémentation système mais aussi le code de l'application. De plus, l'OS est largement privilégié par les développeurs dans le monde de la programmation. C'est pourquoi nous avons choisi de réaliser notre projet sous Linux, qui s'adressera donc à un public familier avec la CLI (Command Line Interface) et les autres aspects techniques. Des interfaces seront potentiellement développées à terme pour les utilisateurs de distributions plus user-friendly (comme Ubuntu).

### **1.2.4 Open Source**

Le développement du projet se fera de manière complètement transparente et donc en open source. Ce choix est assez logique lorsque l'on réalise un programme pour Linux, car il s'inscrit exactement dans la politique des développeurs qui ont réalisé ce dernier. Cela possède de nombreux avantages : possibilité pour la communauté de contribuer au projet au travers de modifications du code, commentaires, rapport de bug, ...

## Chapitre 2

# Stratégie de développement

## 2.1 Méthodologie voulue

Lors de la réalisation du cahier des charges, nous avons réfléchi à la mise en place d'une méthodologie agile : SCRUM, afin de conserver une bonne visibilité sur le projet. Ainsi nous souhaitions réaliser des cycles ; un cycle correspondant au laps de temps entre deux réunions avec le tuteur ; pour que ce dernier soit au maximum impliqué dans le projet et puisse nous aiguiller en cas de problème.

La méthodologie SCRUM implique également une intégration continue pour faire paraître à la fin de chaque cycle une nouvelle version. Pour cela l'idée était d'utiliser un service de décentralisation basé sur Git (GitHub), pour pouvoir merged les différentes branches dans la branche master.

## 2.2 Application réelle au projet

La réalisation du projet en lui même à nécessité de longues heures de recherches individuelles, notamment pour lire de la documentation et faire des "boûts" de code afin de mieux cerner ce que nous allions faire. C'est pourquoi la phase de développement à été plus courte que prévue.

Malgré cela nous avons réalisé de manière régulières des *stand-ups* pour répartir les recherches et pour parler de nos avancements et de nos problèmes. Lorsque nous avions plus de temps (période de vacances), il nous arrivait aussi de travailler quotidiennement en open-space en équipe de 2-3 personnes pour permettre un retour plus rapide sur le travail effectué et une mise en commun.

Pour ce qui est de la méthode SCRUM, nous avons mis en place des sprints de une à deux semaines. À la fin de ces derniers nous nous retrouvions, à l'occasion des stand-ups, afin de fermer les tickets résolus et d'ouvrir de nouveaux tickets pour le sprint d'après. Les outils qui nous ont servi pour exploiter le potentiel de cette méthodologie sont *Taiga* pour le suivi et la gestion des tickets et *Discord* pour la communication par message ou vidéo.

L'intégration continue a aussi été plus ou moins mise en place. Nous avons bien utilisé GitHub pour la mise en commun du travail. Chacun travaillant sur sa branche. Cependant, de beaucoup de recherche résultait assez peu de code. Il était donc difficile de séparer les tâches sur plusieurs branches pour ensuite merge sur master. La plupart du temps c'était donc le chef du projet qui implémentait les features, en se basant sur les recherches de tout le monde.

## Chapitre 3

### Phase du projet – distribution des tâches



### 3.1 Rétrospective - diagramme de Gantt specification

Voici ci dessous le diagramme de Gantt que nous avons réalisé lors du cahier des charges pour le S2 :

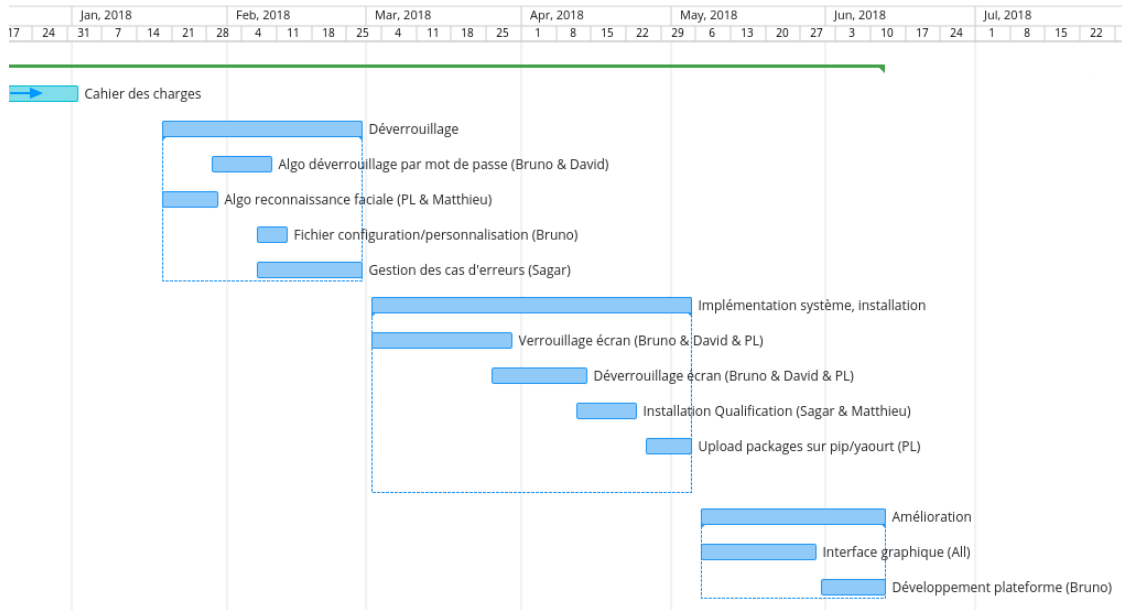


FIGURE 3.1 – diagramme de Gantt

Etant donnée que le sujet était très complexe et comportait de nombreuses zones d'ombres, nous avons fait un diagramme de Gantt, approximatif, quant à la durée des différentes tâches, ainsi qu'à leurs enchainements. Nous allons voir dans la partie qui suit que ce diagramme n'a pas été respecté. Des tâches ont été largement sous-estimé, notamment à cause de l'aspect technique de certains éléments qui ont été bloquant. Certaines tâches ont été supprimé ou requalifié. Suite à nos recherches nous avons changé de stratégie de développement plusieurs fois, il y a donc eu un changement dans les technologies utilisées ainsi que dans les moyens pour arriver à nos fins. Il existe cependant une certaine ressemblance, concernant les grandes parties, entre l'ancien diagramme et le déroulement réel. A savoir :

- Recherche sur l'algorithme pour la reconnaissance faciale
- Verrouillage et déverrouillage de l'écran, implémentation système
- Upload pip, amélioration, test

### 3.2 Répartition réalisée

#### 3.2.1 Difficultés rencontrées

Comme dit auparavant, de nombreuses difficulté ont été rencontré plus tôt que prévu. Voici une liste exhaustive de celles-ci :

- **Répartition des recherches**  
Le sujet abordé était très technique et précis, il nécessitait donc beaucoup de recherche. Mais la répartition était plutôt compliqué étant donnée que idéalement il aurait fallu que

tout le monde possède une connaissance globale des technologies, qui pouvait théoriquement nous servir pour le développement.

— **Exploitation des recherches, mise en commun**

Par la suite, les nombreuses heures de recherches ne nous avançaient guère pour la réalisation de notre projet. Il était difficile d’avoir une vision sur le long terme, quand nous stagnions parfois plusieurs jours sur un aspect compliqué. Il n’était pas évident de savoir concrètement à quoi allait nous servir certaines connaissances (d’ailleurs, un bon nombre n’ont finalement pas été utiles). La mise en commun prenant beaucoup de temps. Il était parfois nécessaire de rédiger un petit document pour résumer les choses apprises durant une semaine afin de mieux pouvoir les exploiter et de déterminer si oui ou non elles seront utiles.

— **Compréhension, avancement éparse**

Et donc suite à cela, venait le temps de la compréhension. Certains aspects techniques liés à l’architecture linux bloquaient certains. L’évolution des connaissances de chacun était assez éparpillé. De plus en plus, chacun recherchait de son propre côté en fonction de ses avancements. Cela à mener à une dispersion néfaste pour l’intérêt commun du groupe.

— **Changement fréquent de stratégie suite à de nouvelle découverte**

Nos avancements nous ont irrémédiablement mené à changer d’idée plusieurs fois pour le développement (specification dans la partie suivante). Il était donc assez frustrant de se rendre compte que des heures de travail ne seront peut être pas utilisés concrètement dans le rendu final. Mais cela fait partie du projet, c’est à dire que pour en arriver au meilleur résultat possible, il était important d’explorer toutes les pistes possibles pour découvrir de nouvelles choses et ainsi réaliser le code le plus pertinent possible, dans le cadre de la philosophie de lockatme.

— **Implication de tout le groupe**

Evidemment vu la liste qui est en train d’être faite, il est assez simple de deviner que cette phase à été bien compliqué pour certains membres du groupe, qui ont pu prendre du retard dans la compréhension de l’avancement. Mais il était quand même important de tenir tout le monde informé, notamment à travers des réunions régulières. Il a également été décidé, que les personnes, pour qui la partie développement pure était trop complexe, se verraient attribuer des tâches lié à la gestion, au déploiement ou à la présentation du projet. Toutefois il est essentiel que chaque membre ait connaissance des aspects techniques de l’application.

— **Répartition des tâches lors du développement**

En lien avec le point précédent et avec un paragraphe du 2.2, dans le développement final il y avait assez peu de code (même si très complexe). Il n’y avait pas de séparation possible avec différentes couches, comme peu nous offrir le modèle MVC par exemple. C’est donc essentiellement le chef de groupe qui a implémenté la partie finale. Cependant les recherches de chacun ont été prise en compte.

### 3.2.2 Chronologie

### 3.2.3 Liste des tâches approximative

## Chapitre 4

# Technologies utilisées

## 4.1 Liste des technologies explorées

## 4.2 Technologie utilisée – point technique

### 4.2.1 Xlib – python

### 4.2.2 Difficulté rencontrée - multithreading

Chapitre 5

Version finale

## **5.1 Présentation**

## **5.2 Utilisation - mode d'emploi**

## Chapitre 6

### Amélioration possible

- 6.1 Interface graphique
- 6.2 Le futur de lockatme