

CyberKey

Rapport de fin de projet : rappel des objectifs et description des solutions apportées.

Rappels

Le but de ce projet est de faciliter l'accès à la salle D111 du bâtiment ENSIBS (nommée CyberLab) pour le personnel, les enseignants et les étudiants de l'Université Bretagne Sud. Pour cela, il est proposé de fournir à tous ces acteurs un outil numérique leur permettant de partager l'accès à cette salle.

En effet, tous ont besoin d'accéder à cette salle couramment, que ce soit pour des cours, des travaux pratiques, des travaux de recherche ou du travail personnel. Cela dit, la salle a justement besoin d'être verrouillée car elle contient du matériel ne devant être mis à disposition qu'aux personnes autorisées.

C'est pour cela que cette salle est toujours verrouillée par une serrure à clef sauf pendant les heures de cours utilisant cette salle ou sur demande des étudiants. Le système actuel est tel qu'un jeu de clef existe et que les professeurs souhaitant accéder à cette salle peuvent l'ouvrir en récupérant la clef au préalable. En ce qui concerne les étudiants, cet accès est plus restreint et demande une organisation. Actuellement, les étudiants doivent répondre à un sondage en ligne sur le site web "Framadate" où ils indiquent quels sont les créneaux où ils souhaitent travailler dans la salle puis au moment venu, ils vont chercher la clef du CyberLab auprès d'Axel Jacome, l'ingénieur en charge de la D111.

La solution technique proposée était de créer une sorte de clef virtuelle pouvant être activée selon des contraintes d'horaires. Pour cela, la clef virtuelle est implémentée sous la forme d'une application mobile Android, la gestion des horaires par une application web permettant d'administrer l'accès des clefs et la serrure par un système de déverrouillage électronique capable de communiquer avec les clefs et les données de l'ensemble du système.

Ainsi, ce projet est découpé en trois parties. La première partie à réaliser était le développement de l'application d'administration, la seconde le développement de l'application mobile Android et la dernière la réalisation du système de serrure électronique.

Description de la solution technique

Gestion des données du système

Pour toutes les parties du projet (l'application Android, l'application web et le système de serrure électronique), les données devant être partagées sont sauvegardées grâce à la base de données en tant que service proposé par Firebase.

L'avantage de cette base de données est qu'elle permet à plusieurs clients (peu importe la plateforme) de partager une instance de cette base de données, ainsi les données des différentes applications sont mises à jour en temps réel (de l'ordre de la dizaine de millisecondes). Cette base de données est de type NoSQL et les données y sont sauvegardées sous forme de données JSON.

Un autre avantage de ce système de base de données est qu'il permet pour une application l'utilisant de travailler hors-ligne. En effet, l'application reste réactive aux événements utilisateur même sans connexion car les changements sont temporairement sauvegardés sur disque. Et, une fois que la connexion est rétablie, l'application reçoit tout d'abord tous les changements opérés lorsqu'elle était sans connexion puis se synchronise avec l'état courant du serveur.

Application web d'administration

Cette application est destinée à un administrateur. Elle permet de gérer les créneaux et gérer les accès à la salle. Son développement a été effectué à l'aide du cadre de conception React JS.

Gestion des créneaux

Dans l'application d'administration, la gestion des créneaux, c'est-à-dire la création de nouveaux créneaux et la suppression de créneaux existants est implémentée sous la forme d'un calendrier. Une journée débute à 8h00 et se termine à 20h00, les créneaux sont de 1h30. Ces paramètres ne sont pas configurables depuis l'application elle-même car ils ne changent pas régulièrement. Cela dit, ces paramètres sont très facilement modifiables depuis le code de l'application sans impacter l'utilisation du système.

Toujours depuis le calendrier, il est possible de parcourir les semaines précédentes et suivantes et de rapidement revenir à la semaine courante. Il est possible de télécharger un fichier de données JSON représentant tous les créneaux du calendrier. A l'inverse il est possible de charger un fichier de données JSON afin de peupler le calendrier. Ainsi, par exemple, il est très facile pour l'administrateur de charger un fichier de créneaux qui serait le même pour les quatre semaines suivantes, etc.

En ce qui concerne le calendrier en lui-même, on retrouve deux types de créneaux, les créneaux de cours et les autres. Les créneaux de cours sont automatiquement peuplés par l'application en effectuant une requête depuis le planning de la salle Cyberlab de l'environnement numérique de travail de l'UBS. Ces créneaux sont écrits en orange pour les différencier. De plus, ils ne peuvent pas être supprimés par l'administrateur et il ne peut y avoir d'autres créneaux au même moment.

Les autres créneaux, écrits en blanc, sont des créneaux libres créés par l'administrateur (en cliquant sur une zone libre du calendrier). Ce sont ces créneaux qui seront proposés aux utilisateurs de la salle depuis leur application Android. Sur ces créneaux, l'administrateur a la possibilité de voir le nombre de demandes d'accès pour ce créneau ainsi que le nombre d'accès accordés. Il est également possible de supprimer un créneau.

Les ajouts ou les suppressions de créneaux ne sont pas immédiatement répercutés en base de données. Ces changements ne sont que visuels tant que l'administrateur ne les a pas sauvegardés via le bouton correspondant. Visuellement, le bouton devient vert lorsqu'un changement n'est pas encore sauvegardé. Avant de fermer l'application et si des changements n'ont pas été sauvegardés, l'application demande une confirmation.

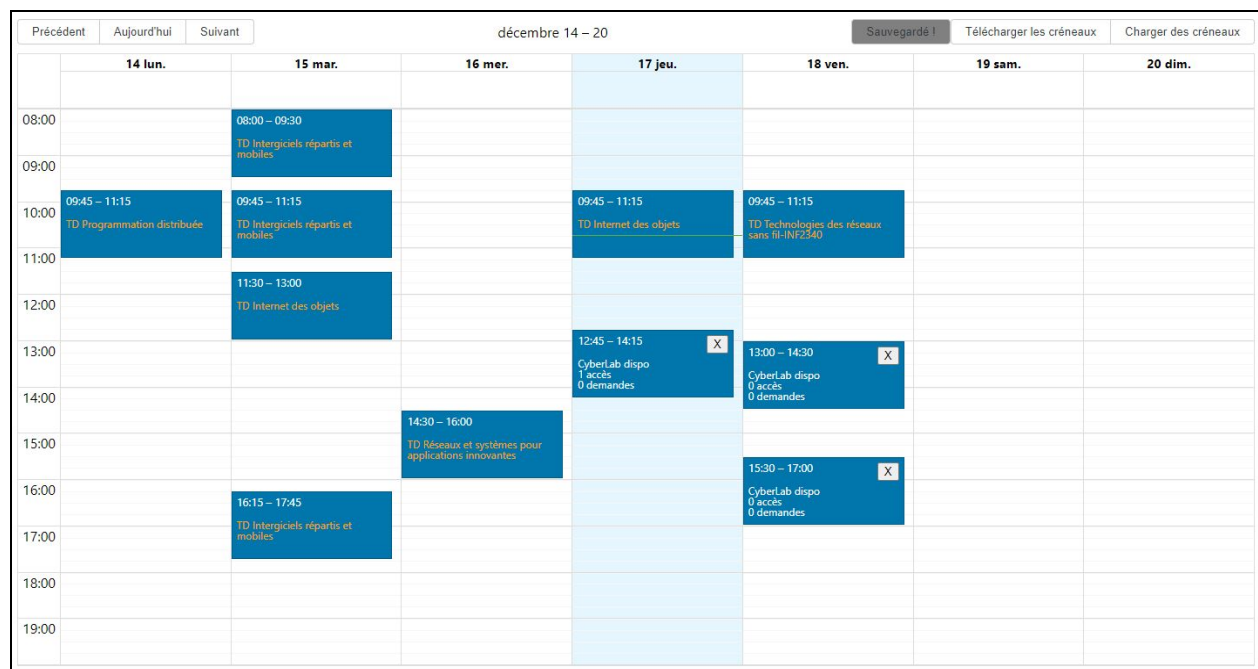


Figure 1 - Vue du calendrier

Gestion des accès

Dans l'application d'administration on distingue deux types d'accès. L'accès pour un horaire précis et l'accès libre. L'accès contraint par un horaire est destiné à la majorité des utilisateurs, les étudiants principalement. L'accès libre est quant à lui réservé aux utilisateurs qui ont besoin

d'accéder à la salle sans contrainte d'horaire. Cet accès est destiné principalement aux professeurs et aux personnels de l'université.

De ce fait, les utilisateurs souhaitant accéder à un créneau défini par l'administrateur le font via l'application mobile Android. Cette demande d'accès arrive presque instantanément sur l'application d'administration. Ensuite, c'est à l'administrateur d'accepter ou de refuser cette demande.

Demandes d'accès au CyberLab :

Créneau du 18/12/2020 à 15:30 - 18/12/2020 à 17:00

☒ ☐ allain.e1602246@etud.univ-ubs.fr

Créneau du 18/12/2020 à 13:00 - 18/12/2020 à 14:30

☒ ☐ allain.e1602246@etud.univ-ubs.fr

Créneau du 21/12/2020 à 13:00 - 21/12/2020 à 14:30

☒ ☐ allain.e1602246@etud.univ-ubs.fr

Figure 2 - Vue des demandes d'accès

Les demandes acceptées peuvent être également visualisées sous forme d'une liste. Il est possible pour l'administrateur de révoquer un accès qu'il avait précédemment accordé.

Accès autorisés pour le CyberLab :

Créneau du 09/12/2020 à 08:00 - 09/12/2020 à 09:30

☒ allain.e1602246@etud.univ-ubs.fr

Créneau du 17/12/2020 à 12:45 - 17/12/2020 à 14:15

☒ allain.e1602246@etud.univ-ubs.fr

Figure 3 - Vue des accès accordés

De même, pour les demandes d'accès libre et les accès libres accordés. Il est possible pour l'administrateur d'accepter ou de refuser une demande et de révoquer un accès.

Demandes de libre accès au CyberLab :

☒ ☐ default

Utilisateurs en accès libre :

☒ allain.e1602246@etud.univ-ubs.fr

Figure 4 - Vue des accès libres

Il est également possible de visualiser les demandes d'accès et les accès accordés depuis le calendrier en double-cliquant sur un créneau. Ainsi, on accède rapidement à toutes ces informations pour le créneau sélectionné. Il est possible d'accepter ou de refuser une demande et de révoquer un accès.

Journalisation

Il semble important pour un système d'accès numérique d'être capable de tracer les accès. Pour cela, l'administrateur peut visualiser tous les accès à la salle. Chaque trace comporte l'adresse mail de l'utilisateur qui a accédé à la salle, la date et l'heure de l'accès.

Ces traces sont ajoutées à la base de données par la serrure électronique elle-même. Ce qui permet même en cas d'ouverture malveillante d'être capable de connaître au moins l'heure d'accès. Cela permet aussi par exemple, dans le cas où du matériel aurait été dégradé, de connaître quelles ont été les personnes qui ont accéder à la salle.

De plus, il est possible de télécharger ce journal dans un fichier au format JSON.

Exporter
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:31:28
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:28:43
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:28:25
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:28:07
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:27:24
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:27:00
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:26:43
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:26:25
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 15:25:40
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 14:53:40
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 01/12/2020 à 14:53:25
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:40:50
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:38:53
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:38:37
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:14:07
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:13:48
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:12:19
. allaine1602246@etuduniv-ubsfr est entré dans le CyberLab à la date du 26/11/2020 à 22:11:57

Figure 5 - Vue du journal

Application mobile Android utilisateur

Cette application est destinée aux utilisateurs de la salle, que ce soit les professeurs, le personnel ou les étudiants de l'UBS. Elle permet de visualiser tous les créneaux libres définis par l'administrateur, de demander l'accès pour un ensemble de créneaux. Elle permet aussi de demander un accès libre. Il est possible de supprimer son compte et toutes ses informations.

L'application permet évidemment , si l'utilisateur a le bon accès, de déverrouiller la serrure électronique.

Cette application a été développée à l'aide du cadre de conception React Native.

Création de compte et connexion

La gestion des comptes utilisateurs est confiée à un service de Firebase qui implémente le protocole OAuth 2.0.

Afin de créer un compte permettant d'utiliser l'application. Il est nécessaire de posséder une adresse mail du domaine de l'UBS (@univ-ubs.fr). De ce fait, on interdit à ceux qui ne font pas partie de l'UBS d'utiliser l'application. La seule contrainte de mot de passe est qu'il doit être d'au moins huit caractères.

Pour se connecter, il suffit d'utiliser la même adresse mail et le même mot de passe utilisés lors de la création de son compte.

Lors de la création d'un compte, un couple de clefs asymétriques RSA de 1024 octets est créé. La clef publique est sauvegardée sur la base de données et la clef privée est sauvegardée dans le système de "keystore" d'Android. Ce processus est de nouveau effectué à la connexion permettant de renouveler le couple de clefs régulièrement.

Ce couple de clefs permet par la suite d'authentifier l'utilisateur lorsqu'il souhaite déverrouiller la serrure électronique.

L'algorithme RSA a été choisi car la carte à microcontrôleur utilisée pour piloter la serrure possède une accélération matériel en son sein pour ce type d'algorithme. De plus, la taille de clef choisie (1024 octets) est un compromis permettant une sécurité relativement robuste tout en conservant un temps de calcul raisonnable tant du côté de l'application utilisateur mais surtout du côté de la serrure électronique.



Figure 6 - Création de compte

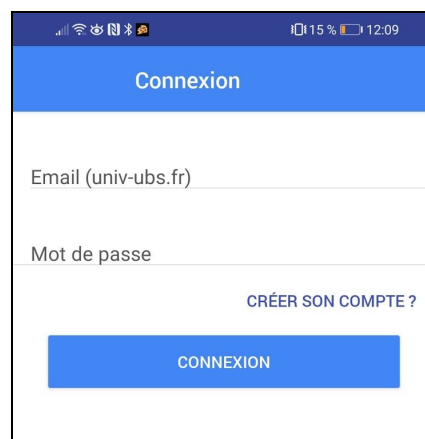


Figure 7 - Connexion

Visualisation des créneaux et demande d'accès



Figure 8 - Liste des créneaux

Tous les créneaux peuvent être visualisés sous forme d'une liste. Cette liste est divisée par semaine, on retrouve dans chaque semaine qui comporte des créneaux des informations sur ceux-ci tels que la date, l'heure, et l'état de la demande d'accès de l'utilisateur. Cet état peut être de trois sortes, l'accès n'a pas été demandé du tout, la demande d'accès est en cours ou l'accès a été accordé. A noter qu'il n'est pas possible de voir quand l'accès a été refusé. En effet, un accès refusé revient à être un accès qui n'a pas été demandé.

Lorsqu'un créneau est ajouté par l'administrateur, celui-ci apparaît directement dans cette liste. De même lorsqu'un créneau est supprimé celui-ci disparaît de la liste. Si le créneau supprimé comportait des demandes d'accès ou des accès, ceux-ci sont de plus automatiquement supprimés.

A noter que seuls les créneaux futurs sont affichés à l'utilisateur, en effet il n'est pas judicieux d'afficher les créneaux passés car ceux-ci ne pourront de toute manière pas être exploités par l'utilisateur.

Suppression de compte

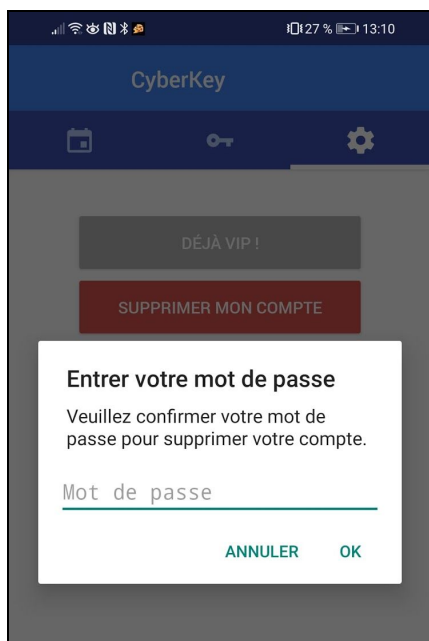


Figure 9 - Suppression de compte

L'utilisateur peut supprimer son compte ainsi que toutes les informations liées à celui-ci. De ce fait, son email sera supprimé de la base de données de même que toutes ses demandes d'accès, ses accès, sa clef publique et sa clef privée. Pour valider cette opération, l'utilisateur devra rafraîchir sa connexion en rentrant à nouveau son mot de passe.

Demande d'accès libre

L'utilisateur peut demander un accès libre sans contrainte de créneaux. De même que pour une demande pour un créneau, cette demande peut être acceptée ou refusée par l'administrateur. A noter qu'une demande refusée n'est pas indiquée mais qu'elle redevient une demande non effectuée.



Figure 10 - Etats accès libre

Déverrouillage

A ce moment, l'application utilisateur doit communiquer avec le système de serrure électronique. Le Bluetooth 4.0 dit BLE est utilisé pour ce faire. En effet, BLE se prête parfaitement à cette utilisation. La procédure de déverrouillage demande peu de débit, il ne faut pas qu'un utilisateur puisse déverrouiller la serrure depuis n'importe où, pour cela la portée de BLE qui est de l'ordre de la dizaine de mètres convient.

La procédure de déverrouillage suit un déroulement précis.

Dans un premier temps, l'application vérifie si l'utilisateur a effectivement le droit d'accéder à la salle. C'est-à-dire, soit l'utilisateur dispose d'un accès libre, soit l'utilisateur dispose d'un accès pour le créneau courant, si un tel créneau existe.

Si l'utilisateur peut effectivement accéder à la salle, alors l'application commence par scanner les périphériques BLE aux alentours pendant 2 secondes. Cette durée de recherche a été trouvée en testant au fur et à mesure différentes durée de recherche.

Si après cette recherche, l'adresse MAC du système de serrure a été trouvée l'application peut débiter la communication avec le système de serrure. Sinon la procédure est réinitialisée en indiquant à l'utilisateur que la serrure n'a pas pu être trouvée.

Si la serrure a été trouvée, l'application commence par négocier un MTU pour les

communications à suivre de 512 octets qui est l'ordre de grandeur estimée des communications qui suivront.

L'application utilisateur initie la communication en envoyant sur une caractéristique BLE de la serrure l'identifiant en base de données de l'utilisateur.

L'application attend 10 ms que la serrure réponde en ayant écrit sur une autre caractéristique BLE une chaîne de caractère challenge. Si au bout de ces 10 ms rien ne se passe la procédure est réinitialisée.

L'application signe la chaîne challenge lue avec la clef privée de l'utilisateur qui se trouve dans le "keystore" du système Android. La signature résultante est écrite sur une autre caractéristique BLE. A ce moment, si la signature a été correctement vérifiée par la serrure, alors la porte doit être ouverte. On bloque la procédure pendant 10 secondes le temps que le système de serrure ouvre la porte et pour éviter que l'utilisateur ne renvoie une demande trop rapidement.

Si pour l'une des raisons énoncées ci-dessus la procédure n'aboutit pas, un message est affiché à l'utilisateur avec la raison. Les raisons sont les suivantes : "Serrure non trouvée", "Erreur de connexion à la serrure" et "Erreur de procédure, recommencer". La dernière raison est plus générale car elle exprime le fait que la serrure a mis trop de temps à répondre ou que la négociation du MTU n'a pas abouti.

Ci-dessous, les phases de la procédure affichée à l'utilisateur lorsqu'elle aboutit. Avec dans l'ordre, procédure non démarrée, recherche de la serrure, connexion à la serrure, phase de communication et porte ouverte.

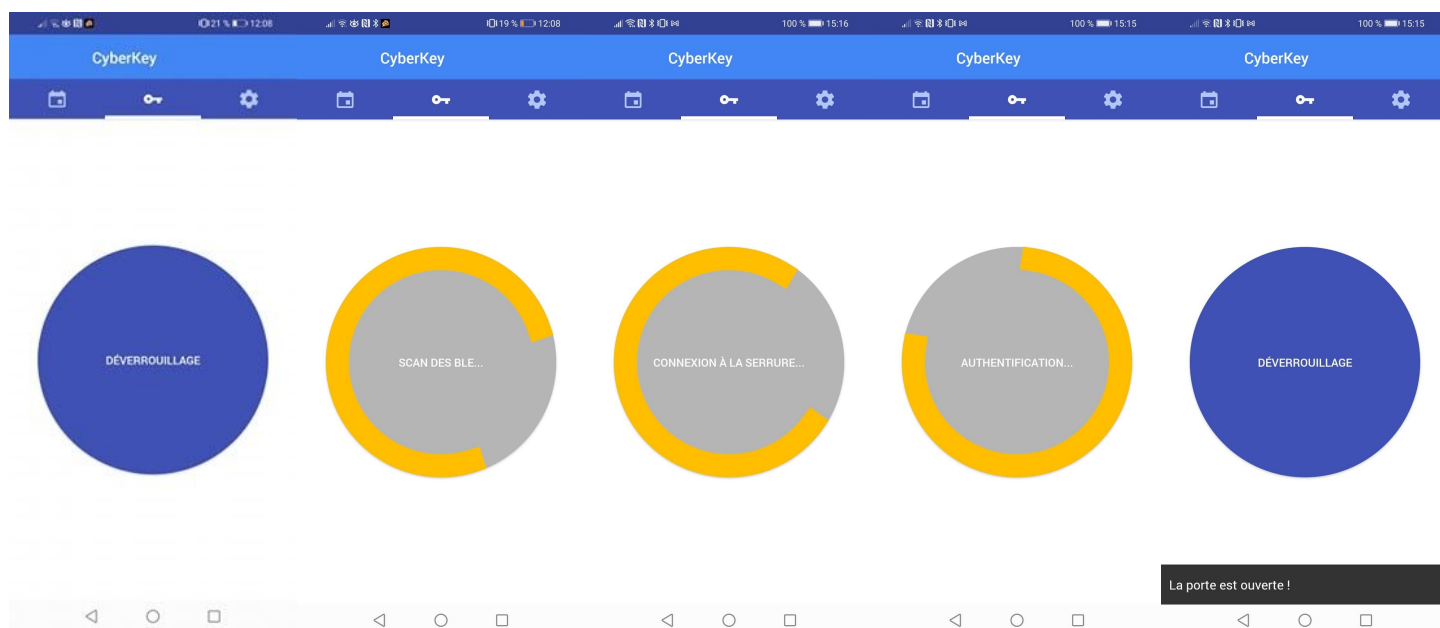


Figure 11 - Phases de déverrouillage : Début, scan, connexion, authentification, porte ouverte

Système de serrure électronique

Le système de serrure électronique est composé de plusieurs éléments. La carte à microcontrôleur qui est un ESP32 DevKit permettant de gérer tous les autres composants, un écran OLED SSD1306 permettant d'afficher l'état de la serrure, un relais 3.3 V et une serrure électrique. Il était précisé dans la proposition de projet que ce devait être une serrure munie en plus d'un déverrouillage mécanique pour des raisons de sécurité. Étant donné le confinement, il a été difficile de pouvoir acquérir une telle serrure. C'est pour cela que cette serrure a été remplacée par une simple gâche électrique (matériel qui était déjà en notre possession). Celle-ci ne peut se déverrouiller mécaniquement mais elle permet quand même de bien visualiser le fonctionnement.

La carte à microcontrôleur ESP32 DevKit

Cette carte a pour rôle de contrôler tous les autres composants du système de serrure. Elle a de plus un rôle de serveur BLE répondant aux requêtes des utilisateurs via l'application présentée précédemment dans le but de déverrouiller la serrure électrique.

La structure du code qui tourne sur cette carte est la suivante.

Tout d'abord, un serveur BLE est initialisé en créant un service BLE comportant trois caractéristiques BLE, la première caractéristique sert à recevoir l'identifiant en base de données de l'utilisateur, la seconde sert à envoyer la chaîne de caractère challenge à l'application utilisateur et la dernière sert à recevoir la signature de l'utilisateur de la chaîne de caractères challenge. Enfin, on initialise l'écran OLED en affichant un message permettant aux utilisateurs de savoir que le système de serrure est prêt à fonctionner.

A ce moment la carte est en attente d'une connexion BLE.

Lorsqu'un client se connecte, la carte attend 3 secondes avant de se réinitialiser si le client n'envoie pas son identifiant en base de données sur la bonne caractéristique.

Si avant ce délai le client effectue cette démarche alors la carte répond en écrivant une chaîne de caractères challenge sur la caractéristique qui correspond.

Après l'écriture de cette chaîne de caractère challenge, la carte attend encore 3 secondes avant de se réinitialiser si le client n'envoie pas la signature de la chaîne sur la bonne caractéristique.

Si avant ce délai le client effectue cette démarche alors le client a bien envoyé toutes les informations nécessaires au déverrouillage de la serrure. A ce moment la carte arrête le serveur BLE prévenant ainsi toutes connexions d'autres clients.

Ensuite, la carte se connecte à un réseau Wifi puis se connecte à un serveur NTP afin de récupérer la date et l'heure courante.

Puis, la carte se connecte à la base de données afin de récupérer la clef publique de l'utilisateur qui correspond à l'identifiant en base de données reçu précédemment. Cette clef publique est composée d'un module et d'un exposant.

Grâce à la clef publique récupérée depuis la base de données et avec la signature reçue précédemment, la carte vérifie cette signature grâce à un algorithme de vérification de signature RSA. Si cette vérification est validée, alors la carte déclenche le relais qui actionne la serrure électrique pendant 3 secondes puis ajoute une trace de cette ouverture dans la base de données (avec la date mise à jour grâce au serveur NTP évoqué précédemment). Si la vérification n'est pas validée alors la porte reste verrouillée. De plus, pendant l'ouverture de la porte un indicateur lumineux (une led) clignote.

Un retour de la procédure est effectué continuellement via l'afficheur OLED SSD1306.

Lorsque la carte se réinitialise, elle redémarre complètement. Cela permet d'éviter les éventuelles fuites de mémoires. Cela ne pose pas de problème majeur puisque le redémarrage s'effectue en seulement quelques millisecondes.

Après avoir effectué environ une vingtaine de tests, le délai moyen entre l'envoi d'une requête de déverrouillage et le déverrouillage effectif de la serrure est de 7 secondes avec un écart-type d'environ 1 seconde. Ceci en étant à une porte inférieure à 10 mètres.

Montage des composants

Pour ce qui est du montage du système, il est nécessaire d'alimenter la serrure en 12 V. Pour les tests cela s'est fait grâce à un transformateur d'alimentation de 220 V à 12 V. La carte à microcontrôleur est alimentée en USB et se charge elle-même de réguler la tension à 3.3 V.

La carte à microcontrôleur est chargée de piloter un afficheur OLED SSD1306 grâce au protocole de communication I2C. La carte alimente cet afficheur en 3.3 V par l'une de ses broches. La ligne de données (SDA) de l'afficheur est branchée sur la broche 21 de la carte et la ligne d'horloge de synchronisation (SCL) de l'afficheur est branchée sur la broche 22 de la carte.

La carte est également chargée de piloter un module relais permettant de contrôler l'alimentation et donc le déverrouillage d'une serrure électrique. Ce module relais est alimenté par la broche de la carte fournissant une tension de 3.3 V de la carte. Le contrôle du relais est effectué par la broche 4 de la carte. Lorsque cette broche passe à l'état haut en fournissant une tension de 3.3 V, l'électro-aimant du relais se déclenche et ferme le circuit. Cela permet à la serrure électrique d'être alimentée par l'alimentation 12 V et ainsi d'ouvrir la porte. Comme dit précédemment, il n'a pas été possible d'acquérir une serrure électrique plus complexe dotée d'un déverrouillage mécanique et à déclenchement temporaire après l'envoi d'un signal. Celle-ci a donc été remplacée par une gâche électrique plus simple à déclenchement permanent. C'est-à-dire que tant que la gâche est alimentée en courant celle-ci reste déverrouillée. Par exemple, pour déverrouiller la gâche 3 secondes, il faut l'alimenter pendant 3 secondes.

Ci-dessous, voici un montage des composants du système de serrure.

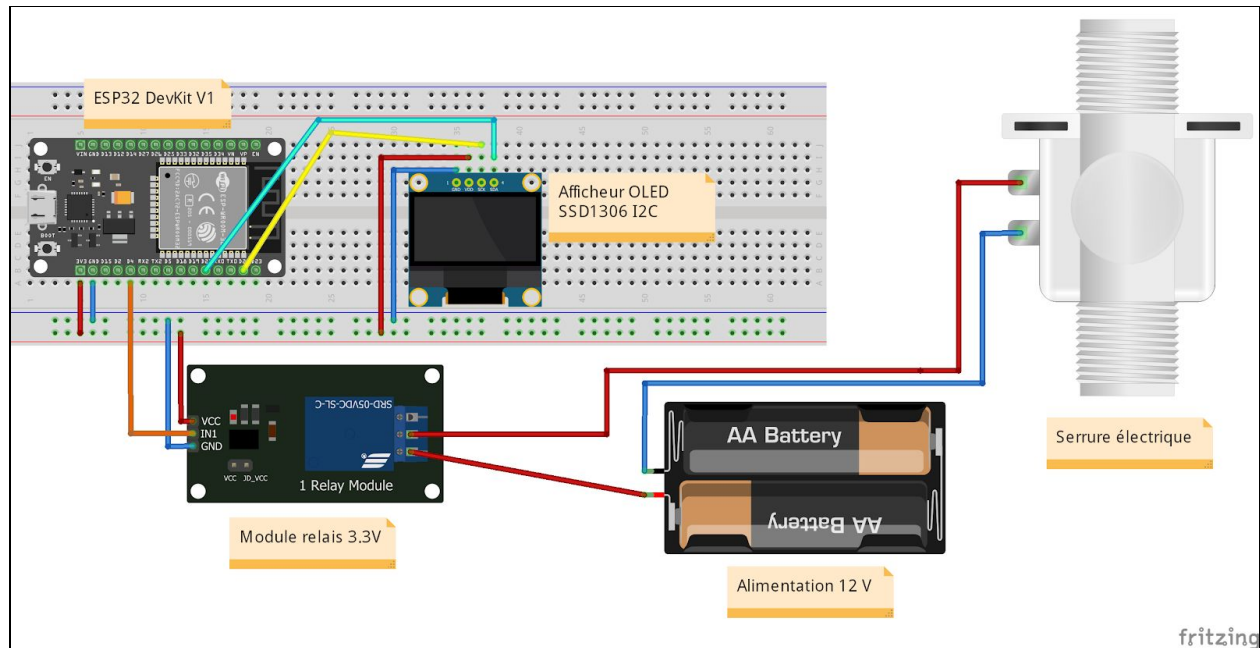


Figure 12 - Schéma du montage du système de serrure

Rappels des livrables

Le premier livrable fourni comportait l'application web d'administration. Celle-ci a été donnée sous la forme d'un URL et d'une documentation expliquant son fonctionnement.

Le second livrable fourni comportait l'application mobile Android utilisateur. Ce livrable prenait la forme d'un fichier apk et d'une documentation expliquant son fonctionnement.

Le troisième et dernier livrable comporte le système de serrure. Celui-ci prend la forme d'un fichier comprenant le code de la carte et d'un schéma des composants du système de serrure. En effet, vu le contexte actuel, il n'est pas vraiment possible de fournir le système de serrure complet. La documentation de ce livrable est donnée en intégralité dans ce rapport.

De plus, ce dernier livrable comprend également une mise à jour des deux livrables précédents. En effet, il a été nécessaire de les mettre à jour pour qu'il puisse communiquer au mieux avec le système de serrure développé à la fin. L'URL fourni au premier livrable n'est plus le même car nous avons changé de domaine. L'application Android a aussi été mise à jour, cela se traduit par un nouveau fichier apk à installer.

Enfin, l'entièreté des codes sources mise à jour des différentes parties du projet est également fournie dans le dernier livrable.