DOCUMENTATION TP D’OUTILS FORMELS

**Introduction**

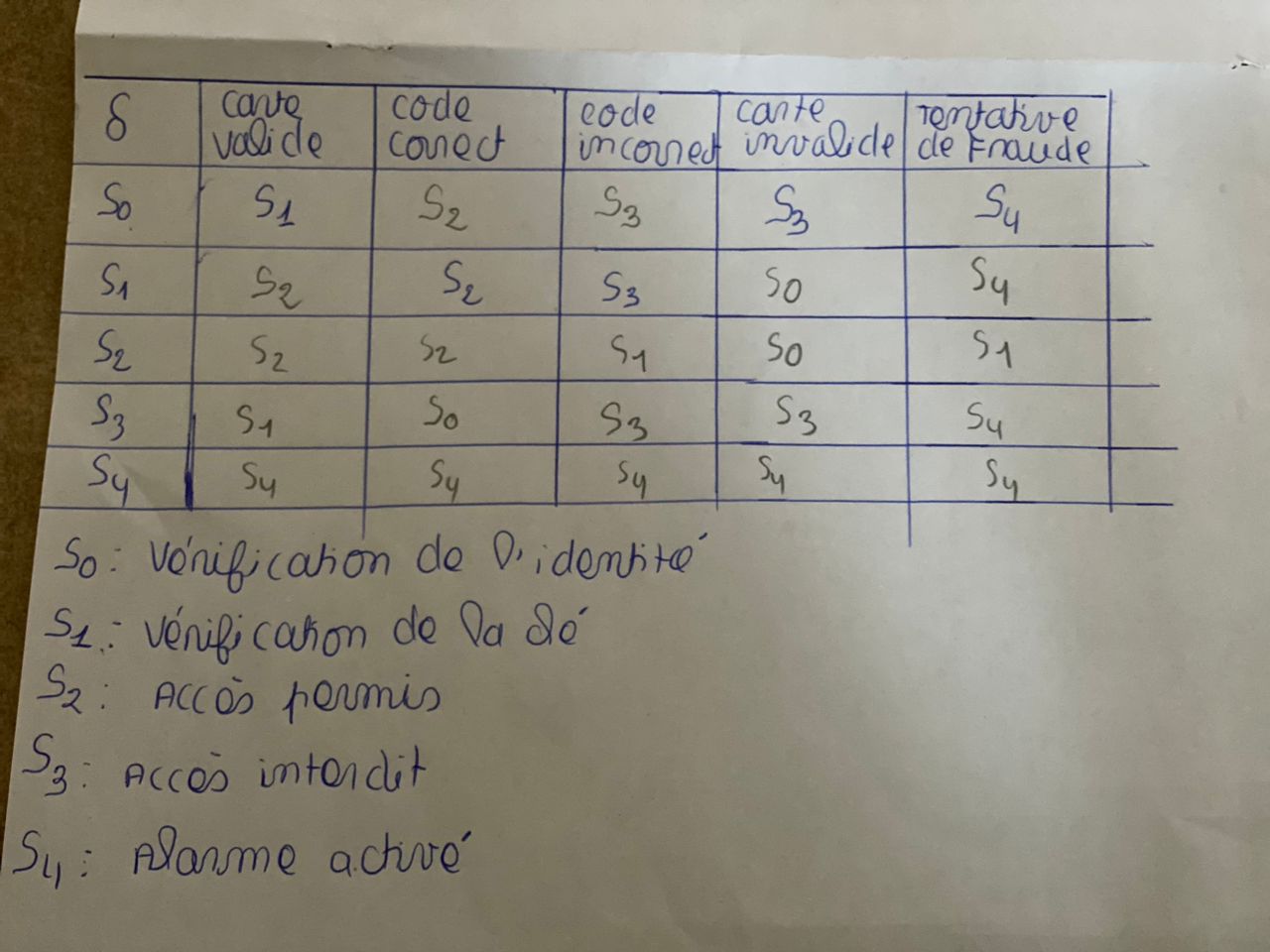
Le projet consistait à spécifier, modéliser, vérifier et optimiser un système de contrôle d'accès pour un bâtiment sécurisé. Le système utilise des règles de logique formelle, des automates pour gérer les états du système, des circuits logiques pour la réalisation matérielle, et des méthodes formelles pour vérifier la conformité du système avec les spécifications.

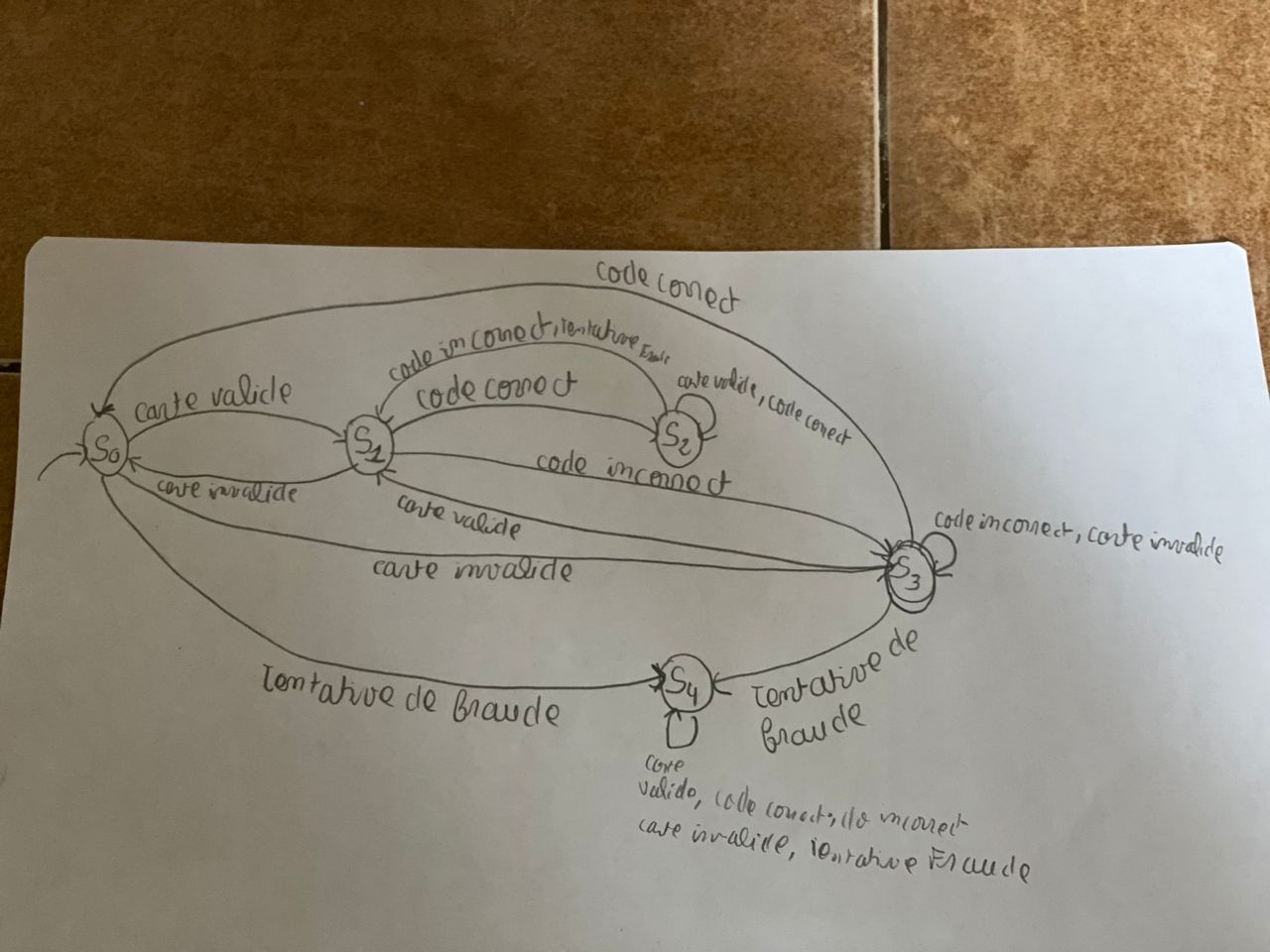
Pour réaliser ce projet, j'ai eu à créer plusieurs classes :

1. **Classe** Carte : Qui gère les informations sur les cartes d'accès.
2. **Classe** Utilisateur : chargée de contenir les informations sur les utilisateurs.
3. **Classe** Porte : Qui gère l'état de la porte (soit ouverte ou fermée).
4. **Classe** Automate : Qui gère les états et transitions du système.
5. **Classe** Historique : Enregistre les événements en quelque sorte un journal.
6. **Classe** Transition : Représente une transition entre deux états avec une condition.
7. **Enum** Etat : Qui permet d’énumérer des différents états du système.
8. **Classe Principale** ControleurAcces : Contient le menu interactif et les méthodes pour vérifier une carte, ajouter un utilisateur, consulter les logs et afficher l'état du système.

Comme difficultés rencontrées : j’ai eu quelques difficultés au niveau des concepts théoriques tels que la logique formelle, les automates finis déterministes, et l'algèbre de Boole.

Pour bien comprendre les automates et pour voir l’adapter au projet, j'ai au préalable dessiné un automate sur le papier pour voir quel exemple de sortie j'allais avoir.





L'utilisation d'outils de vérification formelle peut sembler difficile au début. J'ai regardé un tutoriel sur la logique formelle sur YouTube pour pouvoir bien faire. Voici la page que j'ai pu regarder et qui m'a servi : [Cours sur la logique partie 1](https://www.youtube.com/watch?v=Z3u_d6ZOd-s) ainsi que j’ai eu à lire d’autre blog sur un tel cas