step 1
↳ Constraints
   change
   • little to no Admin users] → up to 10 Admin users
                                    ↳ low amount for security,
                                       but enough for wiggle room

   • Mobile Accessable

   • security is priority

   • Cloud based

↳ Concerns
   • time → lots of detail for small bit of time
   • Internet → wifi can be unreliable
   • Ambiguity → we rely off ambiguity

↳ QA

Non-Functional

Security – Making sure our user's property & data is safe and protected behind our system.

Reliability – User's can rely in our systems to do their job and report when they fail

Scalability – System can accommodate different hardware and software to make a streamlined application.

Efficiency – Connection, Notification, etc. must be fast

Disaster Recovery – Attempt to recover or notify home-owner or police of an issue.

Design Purpose → Add system for login + Alerts
                    Add class diagrams

— Assumptions
   ↳ No Assumptions since we are building off
      current + Adding common features

Step 2
   ↳ ASR ⟶ Security

Step 3
   ↳ User login   ⎤→ provide user security
      User Alerts ⎦   through keeping data
                         w/i secure database
                ↙   + Alerting if something

user ... ~   w/i secure database
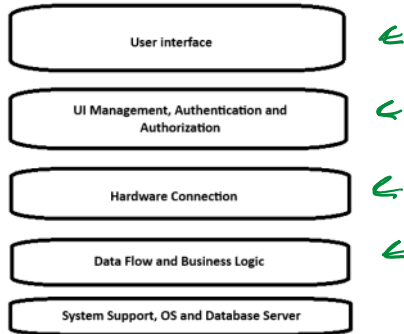   + Alerting if something
Satisfies driver ←   goes dry in sys.

Step 4
implement user class diagram
implement system class diagram
sequence for login + Alert

Step 5

This would be a part of our

User interface ←

UI Management, Authentication and Authorization ←

Hardware Connection ←

Data Flow and Business Logic ←

System Support, OS and Database Server

• Responsibilities

↳ UI for login + Alert

↳ Business logic for creating + storing user login info

↳ Authorization for login

↳ Hardware connection for system status

Admin is a user

Step 6   Admin Users
hard cap of 10 Admins   Permissions

User   create_temp_User()
username   Access_temp_user()
Password   edit_user_perms()
                          sys_Add()
login()   temp_user is a user
sys_status()   temp User
sys_charge()   Permissions

                                    could be a Camera,
                                    door lock, etc

sys—chg... Permiss...

Connects to / has many

security system
subsystems()

sub—status()
sub—change()
sub—add()

has many...

subsystem
name
desc.
info

sys—Alert()
sys—detect()

could be a Camera, door lock, etc

---

sequence

not logged in

APP guest   login   User verify   User

login

login success

login failure

register

sucess

verify login

Verified

verify

sucess

success

success

Access systems

User

Check system

Permissions

Sys. info.

no perms

check sys.

System info

check subsys

System

Subsystem

sys. info...

check
subsystem

Sub. Sys.
info

no perms

check subsys

Sub Sys.
info

sys. info

Alert!
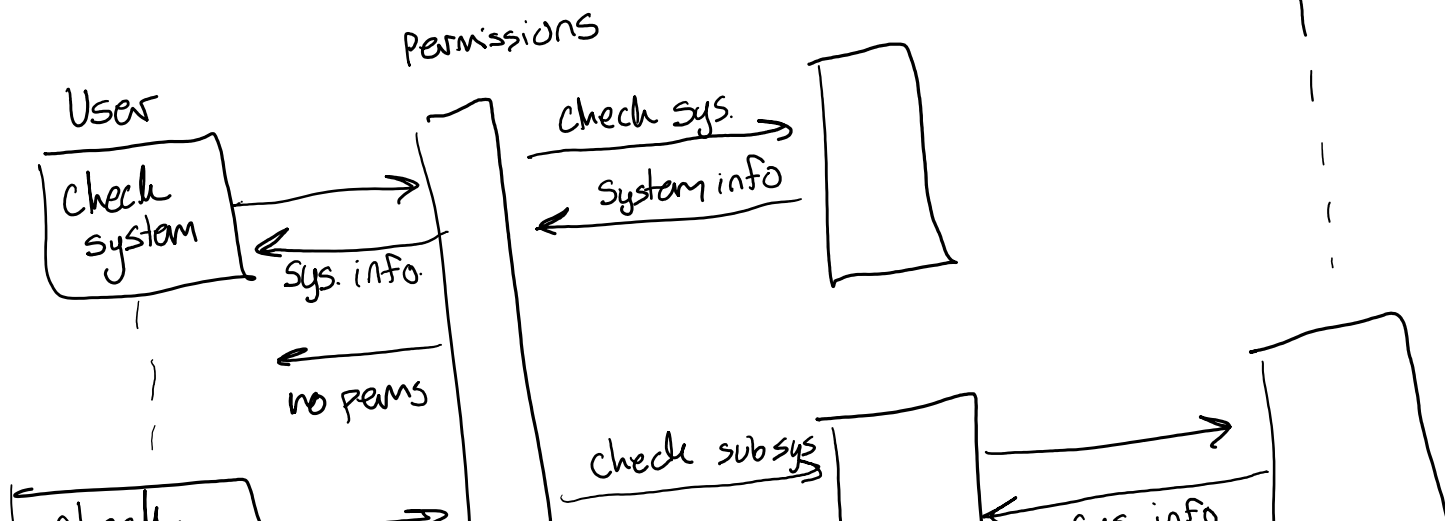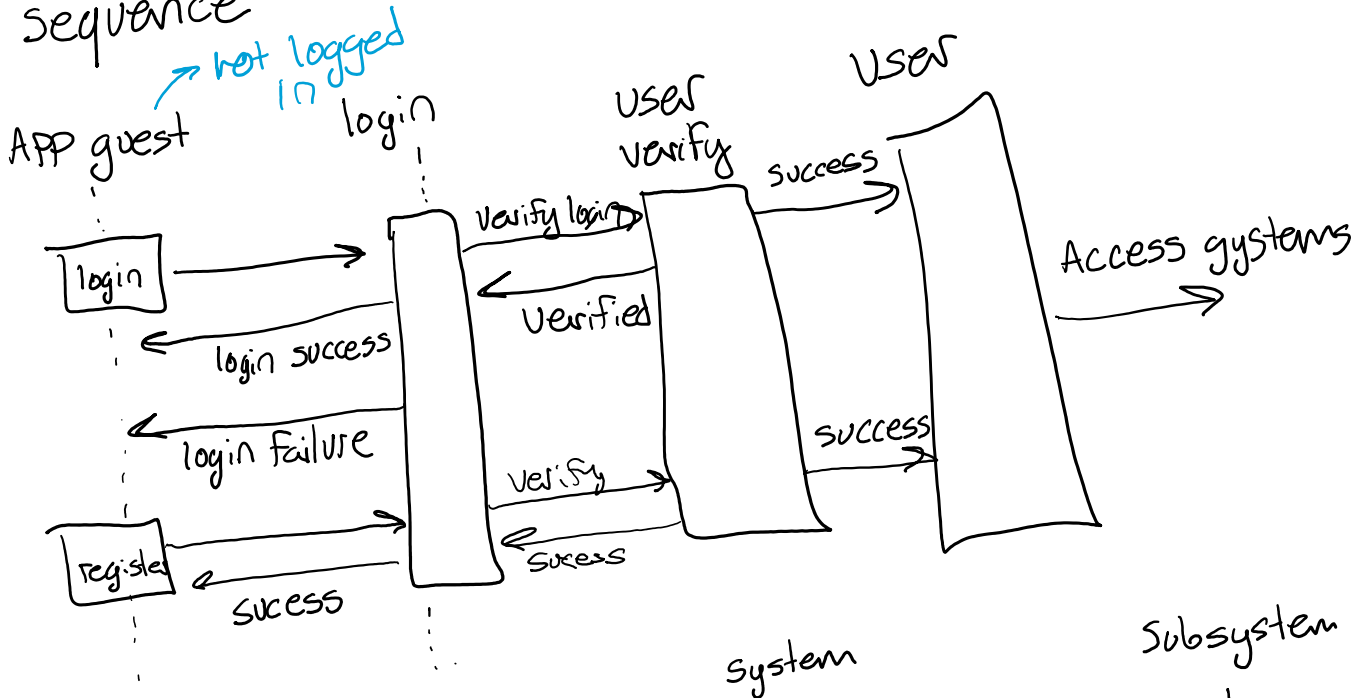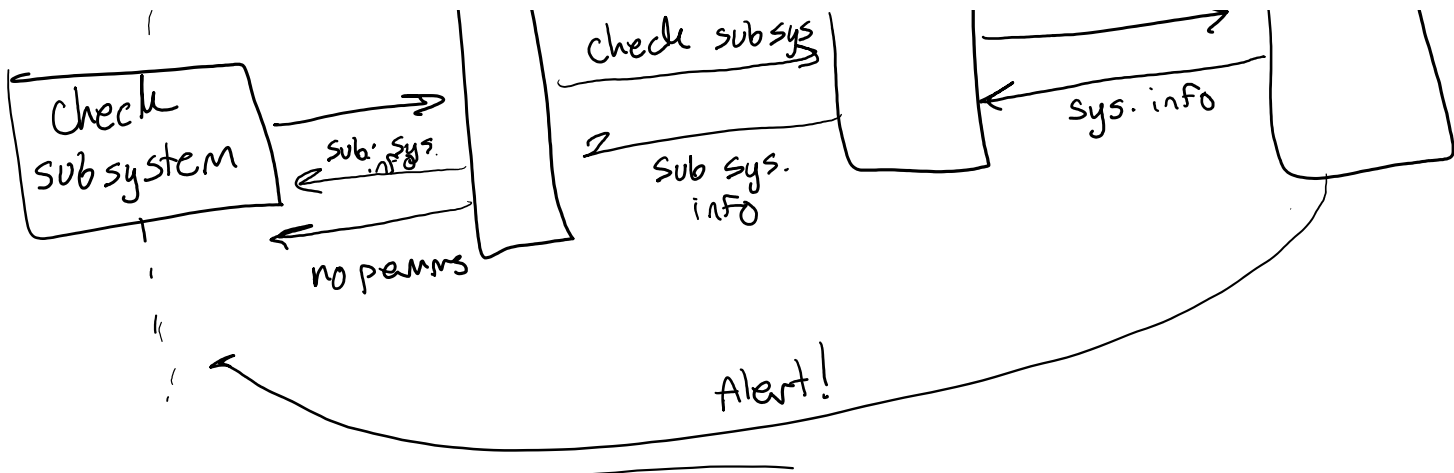
Activity

System check → user door lock

User
Access

Check user
perm

No perm

has access

notify
Admin(s)

unlock
door

keep door
locked