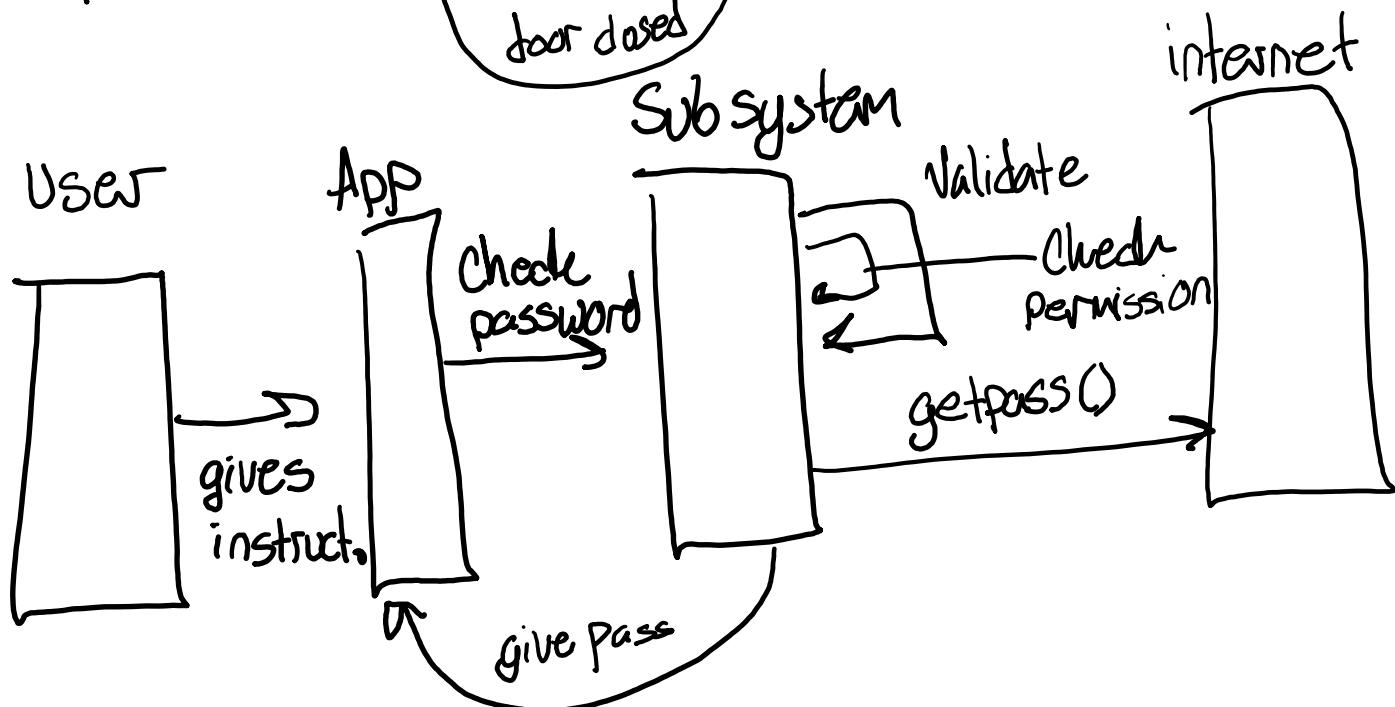
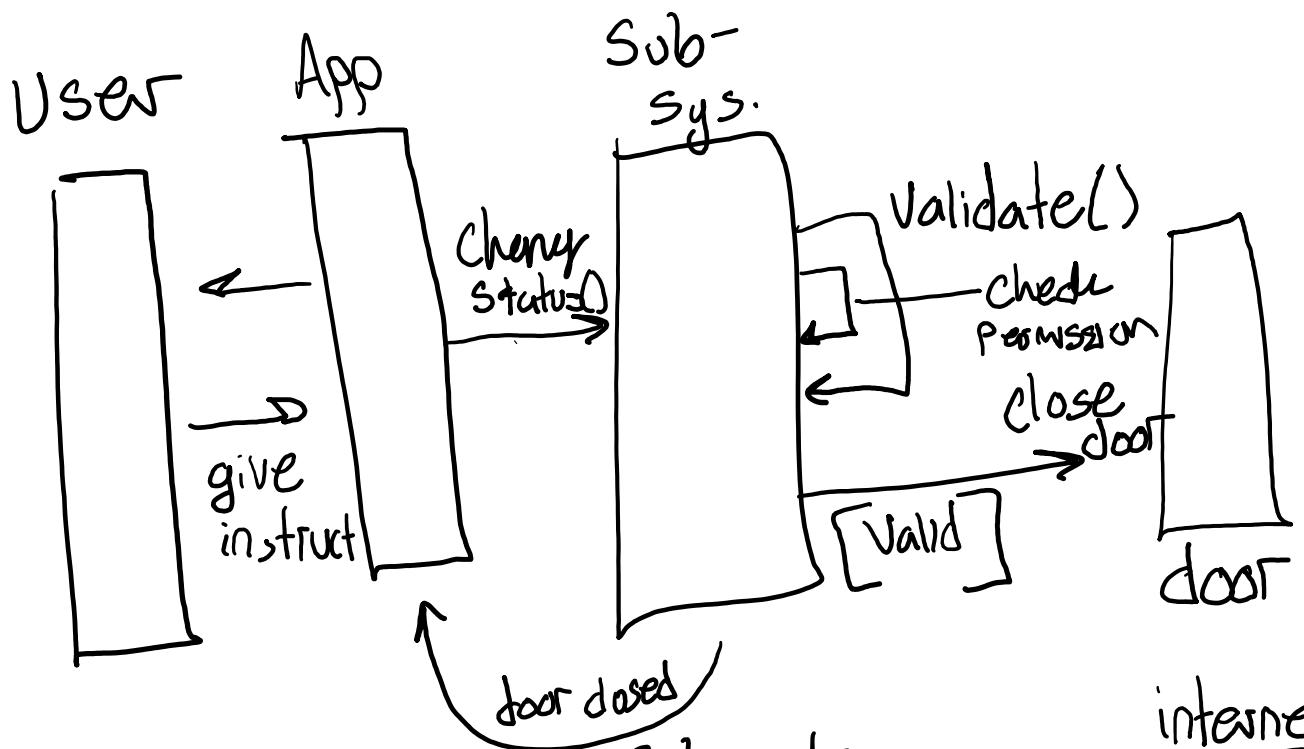


Data Driven UML



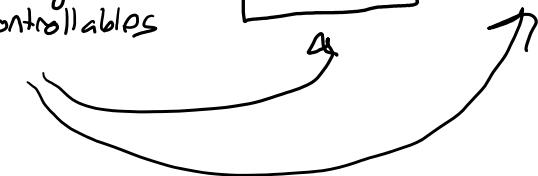
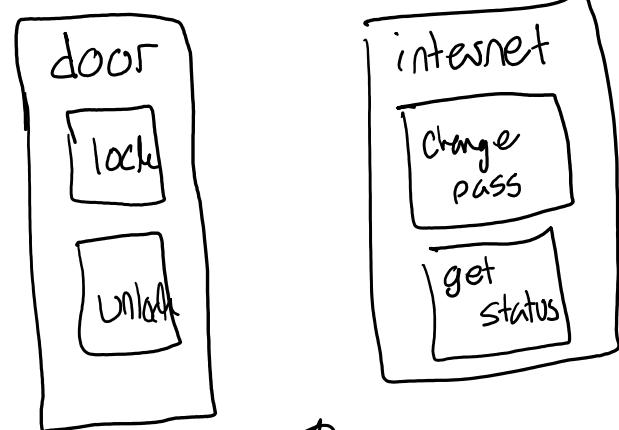
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Stakeholder

Tuesday, September 17, 2024 8:31 AM

element	description
• Stakeholder	home owner, system designer, smart home man.
• Concerns	Show interactions between systems in home + how security works to make things safer
• Modeling	<ul style="list-style-type: none"> • nested boxes - outer : subsys. inner : controllables



Copyright (C) Camilla Lucero, Damon Domke.
 Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

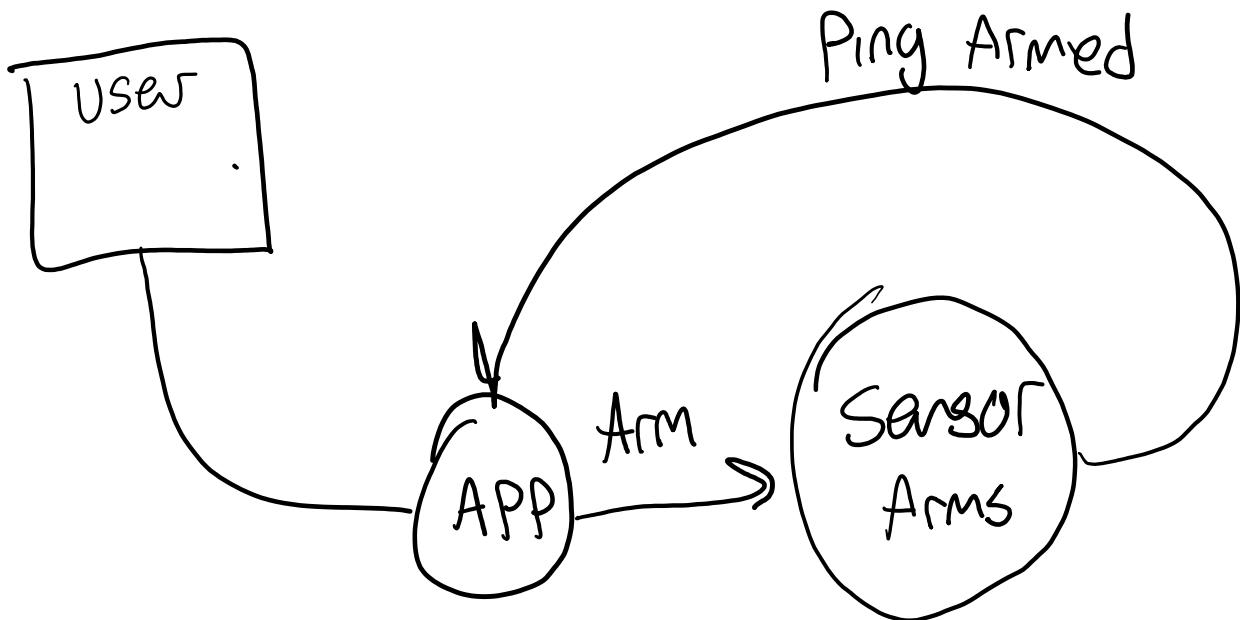
Architectural View Point

Tuesday, September 17, 2024 8:46 AM

Arch. viewpoint → Physical view

◦ Communications

- Dataflow for model based (physical object comm)
- Event based interaction (user → app → object)



Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Our system is a Centralized system

↳ Data is only handled w/i a household

There is no reason for data to be queried over web

↳ other than accessing systems from far away, but that will be controlled w/i overlapping to

* Review design decisions in Lecture 8 slides! * home network

8 Layered Architectural structure

Copyright (C) Camilla Lucero, Damon Domke.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

User stories

Thursday, September 19, 2024 8:51 AM

non functional → sys.

As a homeowner, I want to be able to ~~easily~~^{not measurable} secure my property from my app, so that I can ensure my anxiety about security is alleviated.

As a homeowner, I want my information to be secure behind a login so that I don't need to worry who can access my system

- Yes, these use cases provide necessary abstraction of most important requirements because they state that the system must be efficient, fast, and secure whenever a user changes a part of their system.

↳ App focused, so mobile architecture

↳ Databases?

↳ Can we store our info in the cloud or do we need a physical database?

Functional → User interacts w/

As a homeowner, I want to be able to access my security status and change it, so that I can ensure my most important possessions are secure.

As a homeowner, I want a rundown of all systems, so that I can be up to date on status of all subsystems.

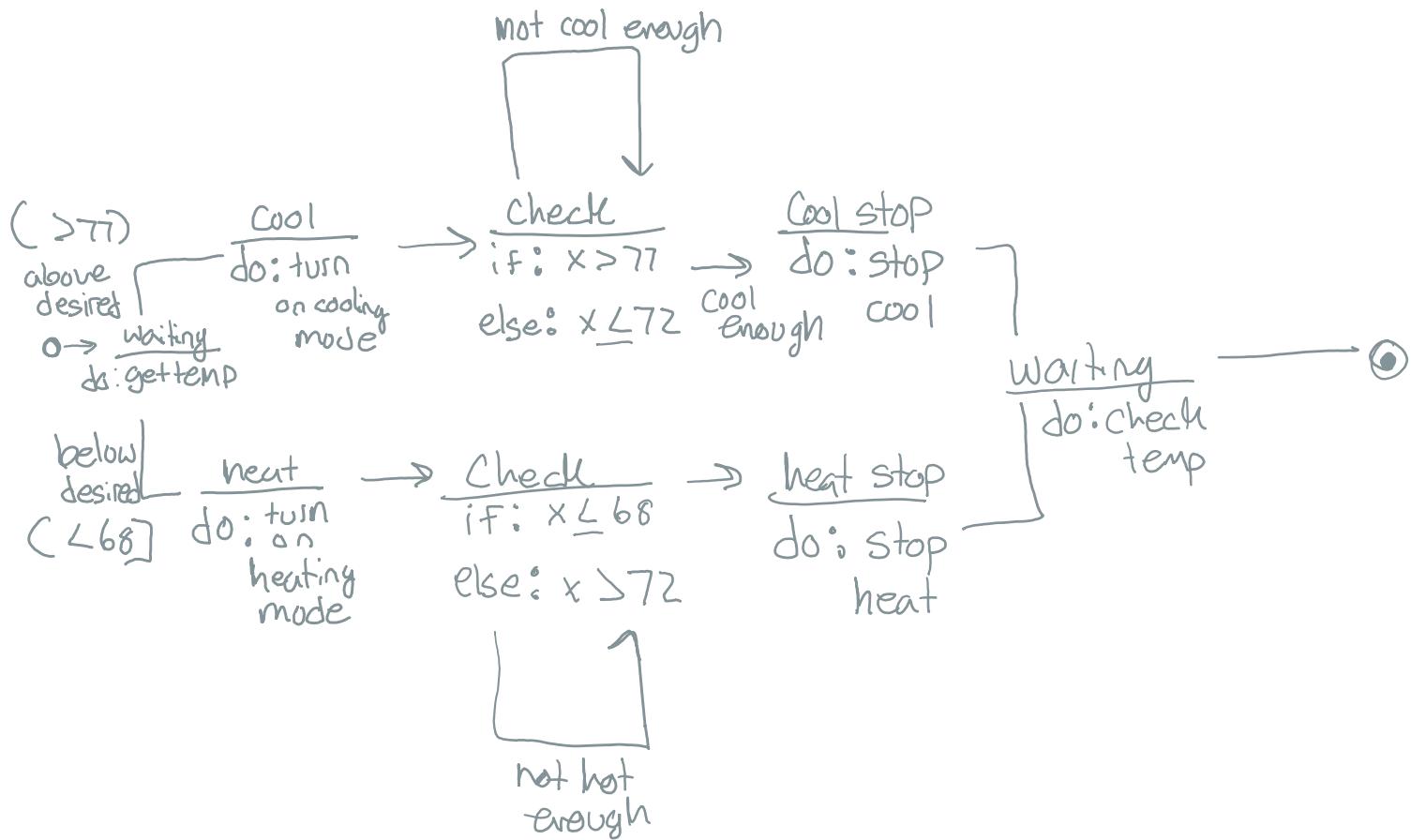
Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Example Of A Event Driven UML diagram

Monday, September 23, 2024 9:07 AM



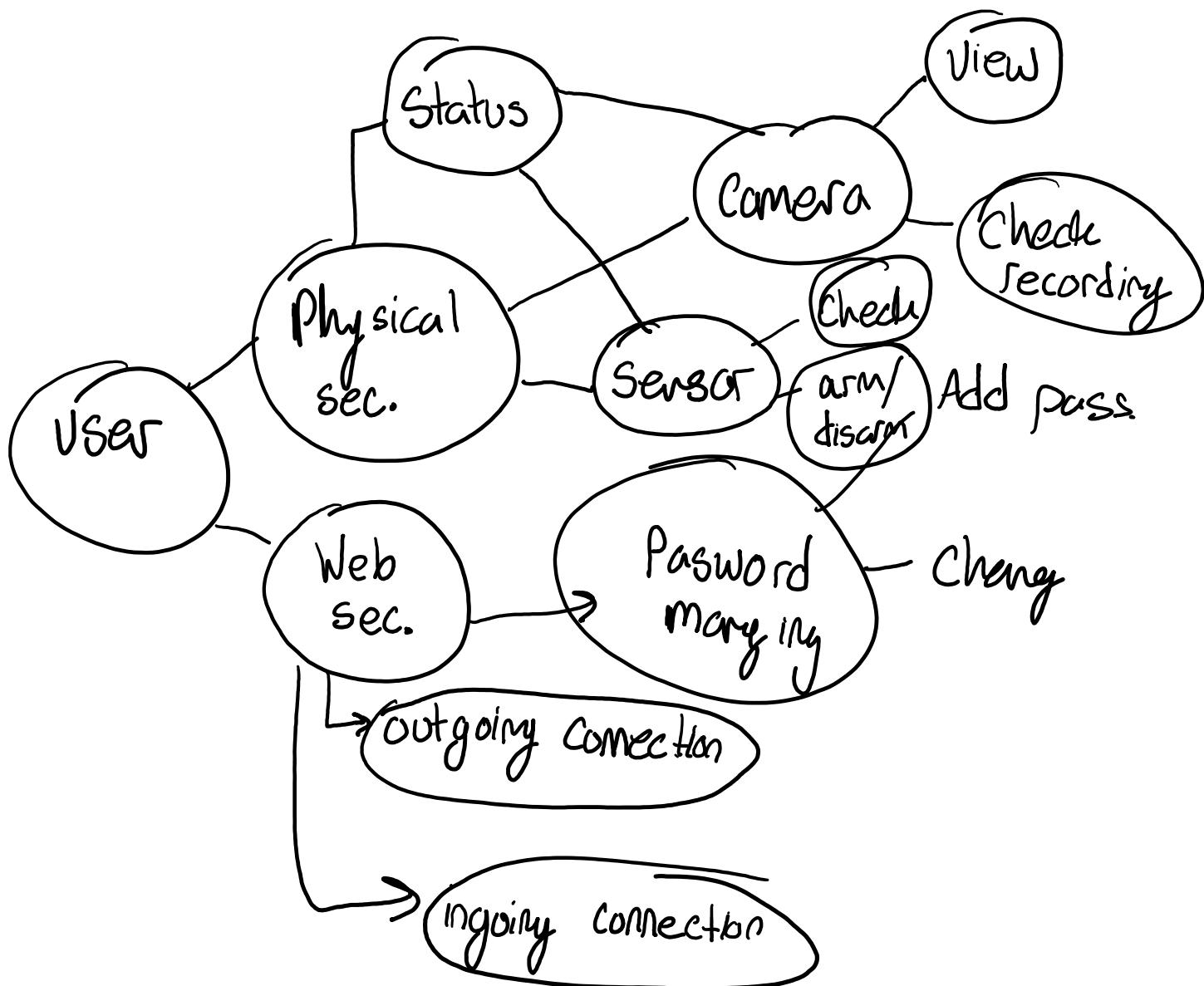
Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Interaction Model (VERY MESSY))

Monday, September 23, 2024 9:08 AM



Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Use Case & UML

Thursday, October 3, 2024 8:36 AM

UC - 1: login

UC - 2: Manage Users

UC - 3: System Status (overall)

UC - 4: View Camera

UC - 5: Subsystem Status (individual)

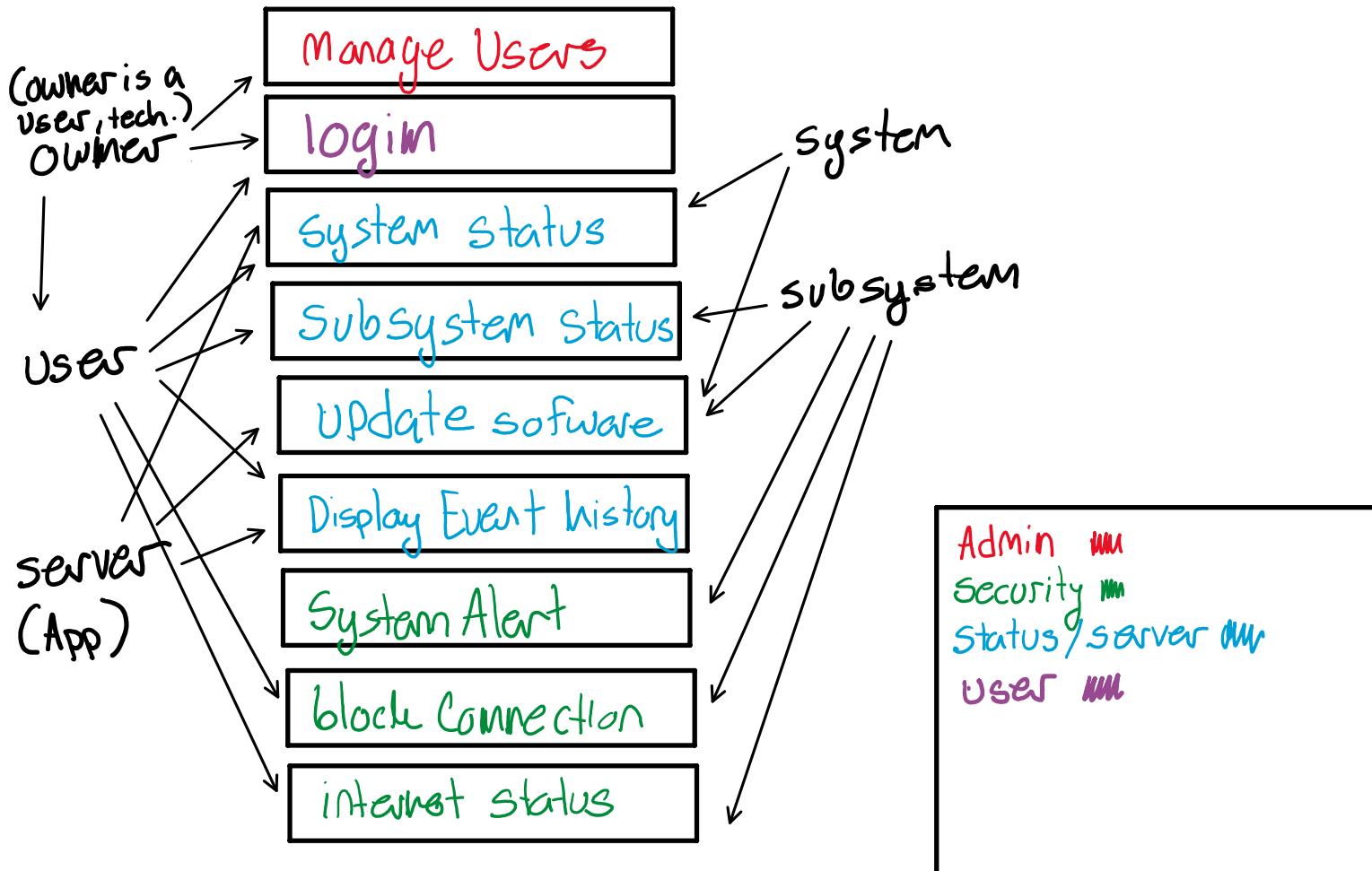
UC - 6: Update Software

UC - 7: System Alert

UC - 8: Display event history

UC - 9: Check internet traffic

UC - 10: Block connection



Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Lab 3 Design Inputs

Tuesday, October 8, 2024 8:37 AM

1. Our project is a brownfield project

↳ we're building off of "abandoned land" and we can pull from other architecture

→ Well understood domain, not "novel"

2. Non functional Requirements

• Reliability → System needs to be functional 100% of the time while providing constant surveillance

• Failure transparency → If system does fail, reports to user quickly + efficiently

• Scalability → Allows systems to be efficiently added

• Disaster Recovery → Attempt to recover by all means either by system reboot or by transferring responsibility to police

Constraints - to be considered (I want to discuss)

• One / little Admin user(s)

• System needs to be Mobile Phone Accessible

• security is top priority

Concerns →

• System needs to be cloud or outside database oriented to not put much pressure on mobile phone

• Baseline cybersecurity to prevent attackess + Collection of user data

3. Goals for Thurs.

↳ security

Inputs → Step 1(A)

1. Our project is a brownfield project
 - ↳ we're building off of "abandoned land" and we can pull from other architecture
 - ↳ Well understand domain, not "novel"

2. Non functional Requirements

- Reliability → System needs to be functional 100% of the time while providing constant surveillance
- Failure transparency → If system does fail, reports to user quickly + efficiently
- Scalability → Allows systems to be efficiently added
- Disaster Recovery → Attempt to recover by all means either by system reboot or by transferring responsibility to police

Constraints - to be considered (I want to discuss)

- One / little Admin user(s)
- System needs to be Mobile Phone Accessible
- Security is top priority

Concerns →

- System needs to be cloud or outside database oriented to not put much pressure on mobile phone
- Baseline cybersecurity to prevent attackers + collection of user data

Assumptions

1. User has preinstalled hardware from 3rd party
 - e.g. cameras, door locks, etc.
2. That system has connections to hardware either that our product is that hardware's main driver software or we have access to driver
3. Security of server hosted on cloud

Use Cases -

UC-3, 5, 7

Goal → establish overall system arch. + Connections to 3rd party

Step 2A

Step 3A

Step 3 A

- Decompose → top down approach
- First iteration → designs top level arch.

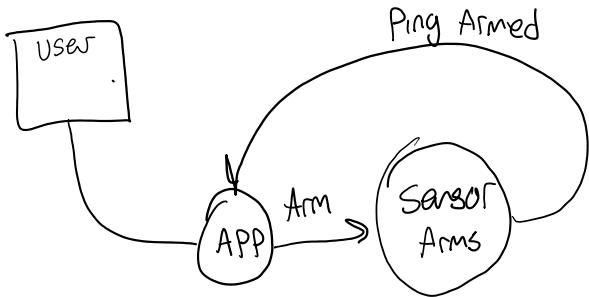
Step 4 A

↳ layered system Arch.

→ Arch. viewpoint → Physical view

◦ Communications

- Dataflow for model based (physical object comm)
- Event based interaction (user → app → object)



→ Component diagram

Step 5 A

User Interface

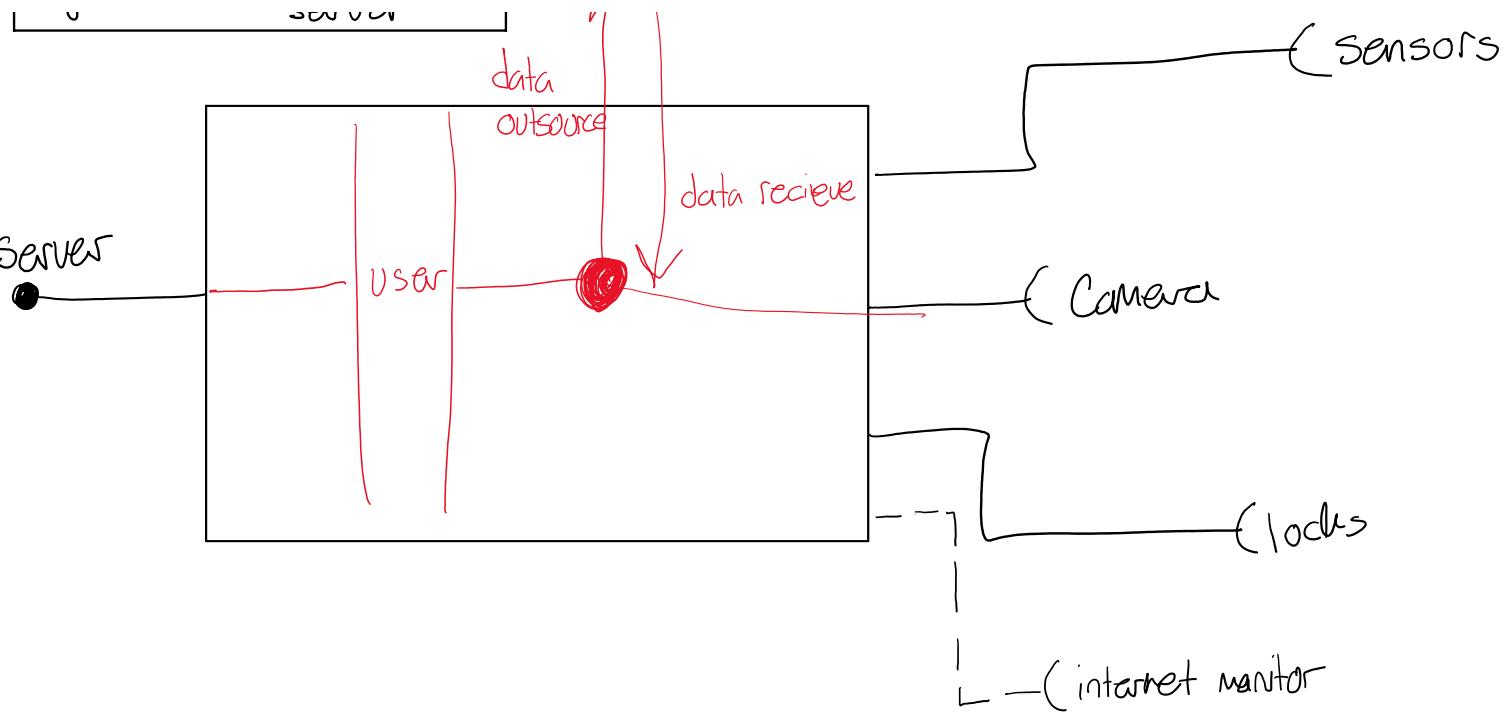
User interface management

Out source to hardware
/software

Business logic + data
Flow control

System support (OS, database)
server





Step 7A

- o Specify component / overall system
- o Data flow specification

Copyright (C) Camilla Lucero, Damon Domke.
 Permission is granted to copy, distribute and/or modify this document
 under the terms of the GNU Free Documentation License, Version 1.3
 or any later version published by the Free Software Foundation;
 with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
 A copy of the license is included in the section entitled "GNU
 Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

our system is decentralized. Since all houses are separate nodes that connect to one main server.

↳ all the server does is push updates + store hardware / software data

transparency → Our system should be transparent

as much as possible, especially to the admin user. Due to system delays, we can attempt to show users a timestamp of the most up-to-date data

- Any user's otherwise should receive least amount of data required by their permissions

↗ for safety

Openness → Code + system should be kept private, but truly does not matter as data cannot be accessed unless connected to home system

Scalability → System is scalable so that more devices can be added + viewed

Security → Security system = as secure as possible

Quality of service → quietest + as efficient as possible w/ a priority on security + communication

Failure management ↴

... - monitored to recover

Failure Management

communication
System is expected to recover
or report issues as soon as
possible

Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Design Decisions

Thursday, October 17, 2024 8:10 AM

1. A leader-follower arch. may be helpful for showing our interactions between client ↔ server and client ↔ subsystem communications. I think it would be a good idea to employ it to show these.
2. A two tier client-serv. Arch is not specific enough for our system. Our app is a smidge to complex.
3. Multi-tier Client Arch. is something we can definitely use to show the interaction between our client server to main database.
4. Peer-to-Peer is too broad for us. We don't have that many arbitrary connections, everything is mostly "linear"

Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Step 1↳ Constraints

• little to no Admin users] → up to 10 Admin users

change

↳ low amount for security,
but enough for wiggle room

• Mobile Accessible

• 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

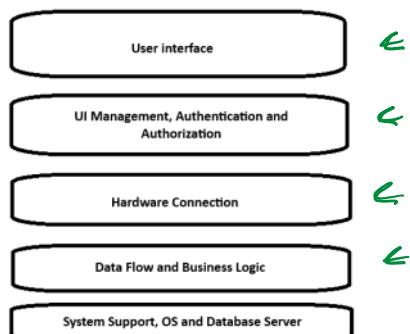
Step 2↳ ASR → SecurityStep 3

↳ User login] → provide user security through keeping data w/i secure data base
User Alerts] → Alerting if something

user ... ↘
 Satisfies driver ↙
 w/i secure data base
 + Alerting if something
 goes dry in sys.

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

Step 5
 This would be part of our



• 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

hard cap of 10 Admins

tUser
 username
 Password

login()
 sys-status()
 sys-change()

Admin users

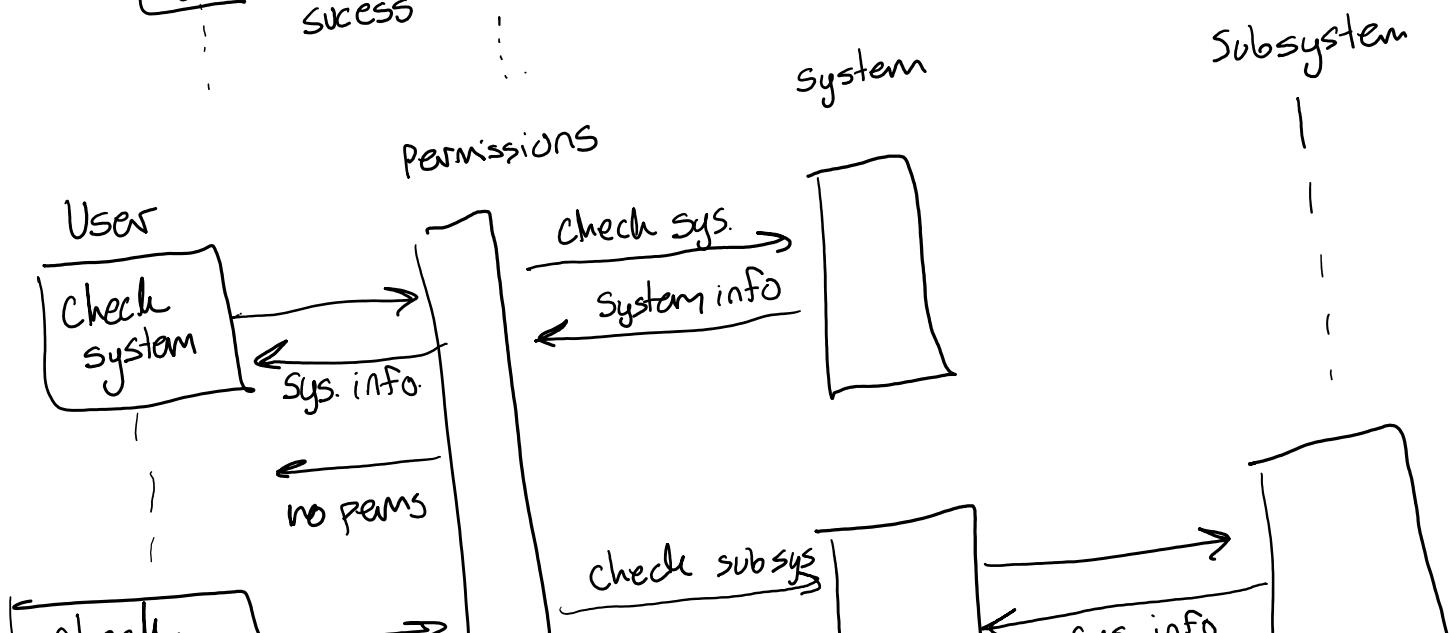
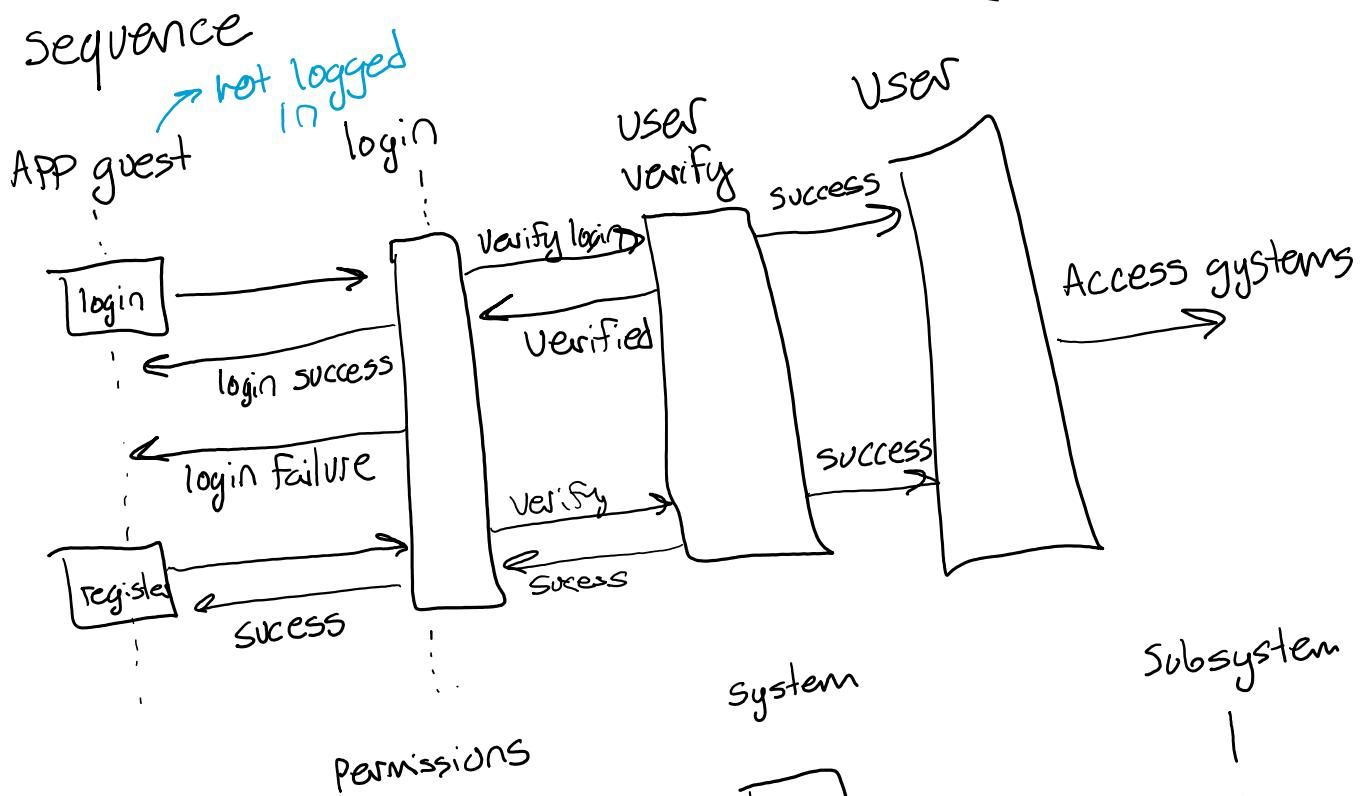
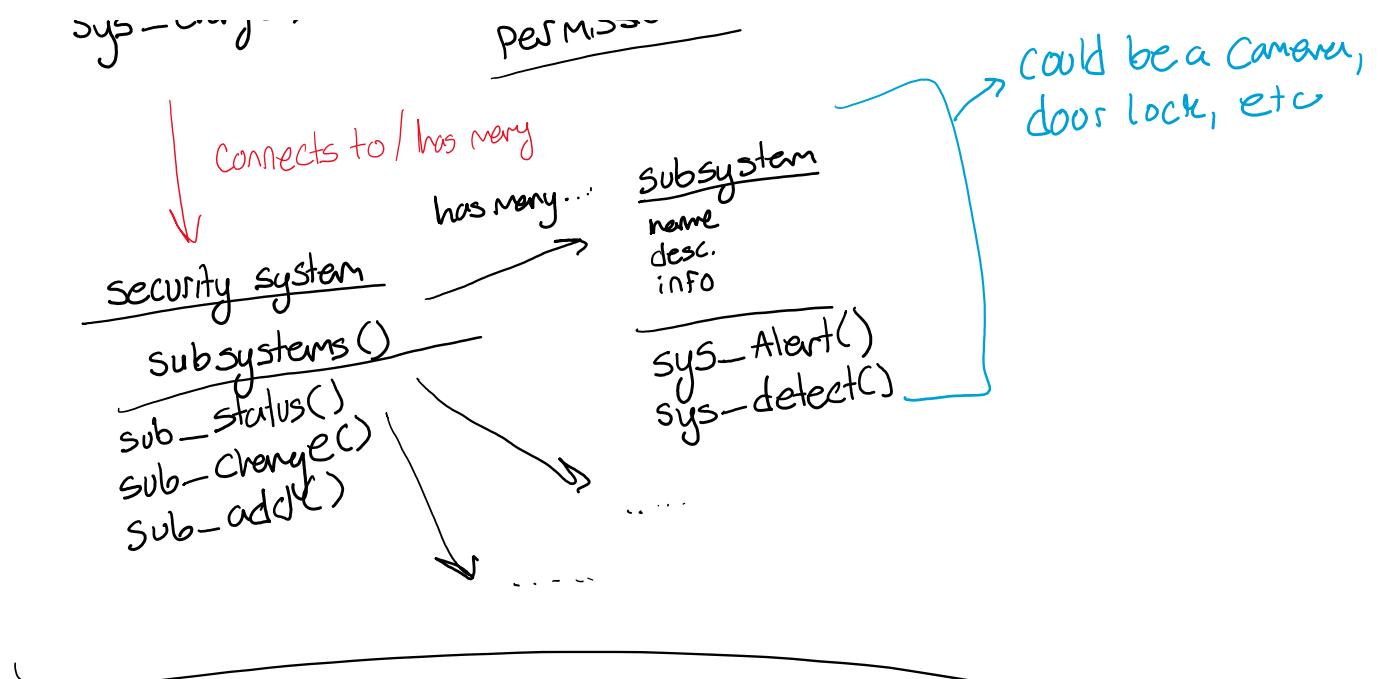
Permissions

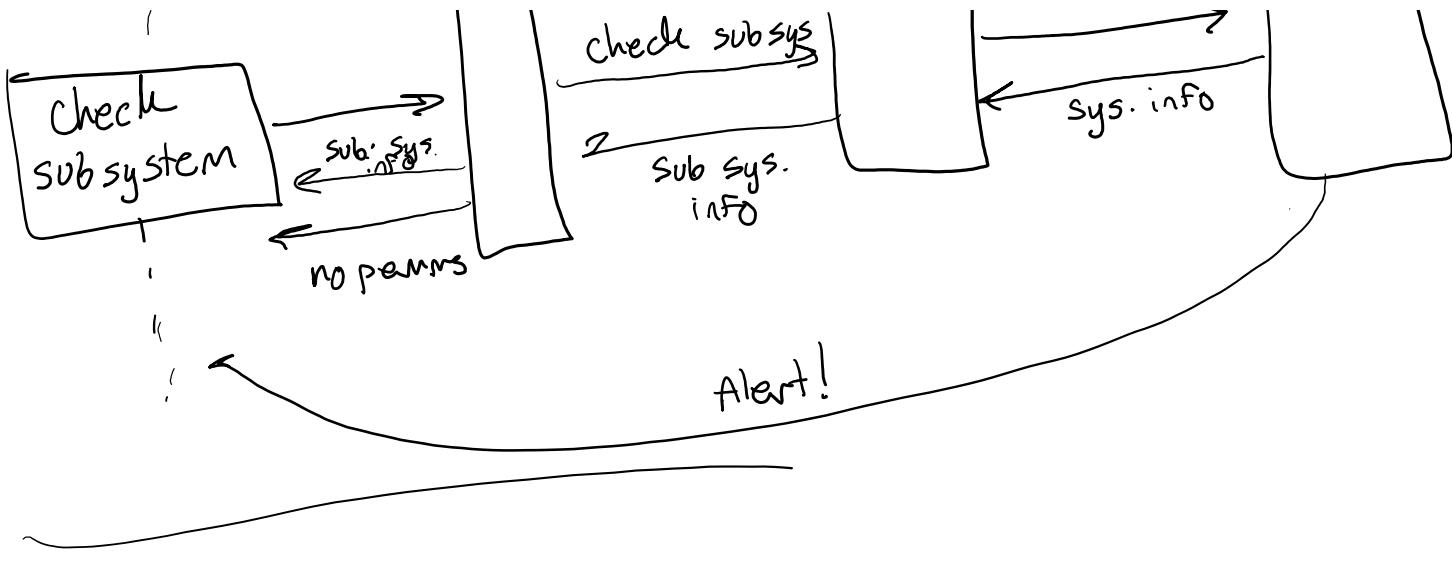
create-temp-user()
 Access-temp-user()
 edit-user-perms()
 sys-Add()

temp-user is a user

temp user
permissions

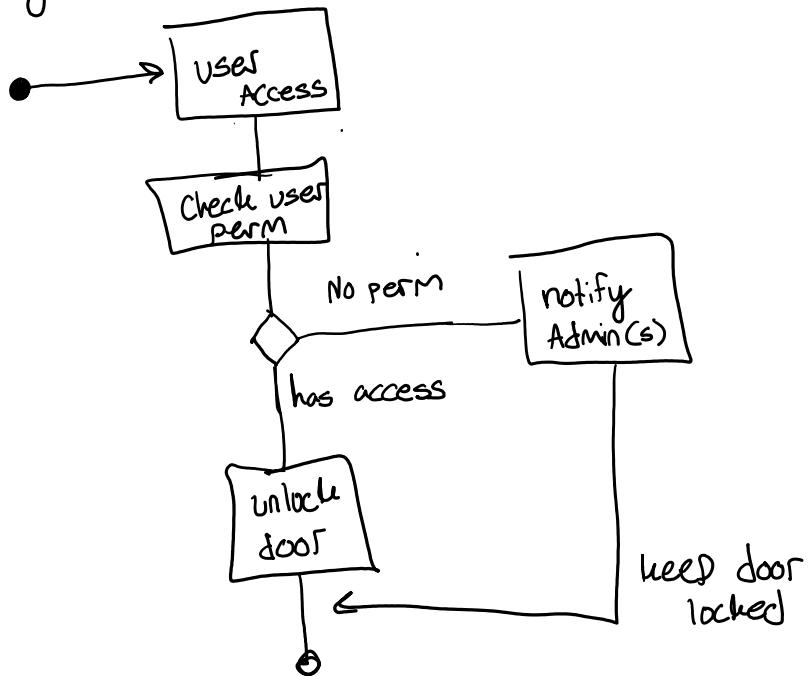
could be a camera,
 door lock, etc





Activity

System check → user door lock

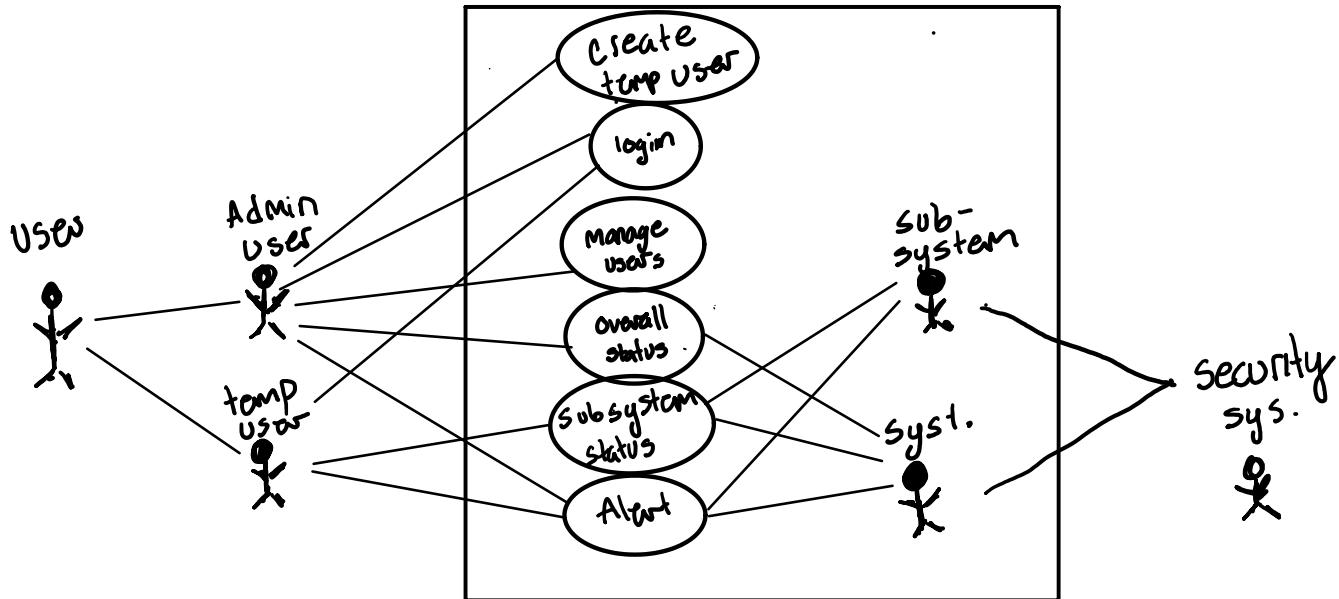


Copyright (C) Camilla Lucero, Damon Domke.
 Permission is granted to copy, distribute and/or modify this document
 under the terms of the GNU Free Documentation License, Version 1.3
 or any later version published by the Free Software Foundation;
 with Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
 A copy of the license is included in the section entitled "GNU
 Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>

Iteration Use Case

Tuesday, November 12, 2024 12:18 PM



Copyright (C) Camilla Lucero, Damon Domke.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

From <<https://www.gnu.org/licenses/fdl-1.3.html>>