SMART LOCK

REPORT #1

Software Engineering - S'19 - Group #10

https://software-engineering-s19-group10.github.io/website/

MEMBERS:

*Ted Moseley (trm124), Michael Truong (mt842), Daniel Nguyen (dnn21),
Eric Lin (ekl40), Mohit Khattar (mk1483), Mohammad Nadeem (mn535),
Andrew Sengupta (ans165), Jasjit Janda (jsj87), Jeffrey Lu (jl2088),
Amandip Kaler (ak1415)*

Table of Contents:

0. Individual Contribution Breakdown

All members contributed equally on the first report.

1. Customer Statement of Requirements

1.a. Problem Statement

Imagine that you are coming home from a long day at work. On the way back you went to the grocery store. You reach your home and are carrying a multitude of bags in your hand. As you are heading towards your door, you try to fumble into your pocket to retrieve your keys. Unfortunately, because of the bags, your keys suddenly fall to the ground and into a sewer. Now, you have to call the locksmith to open your door or somehow get your keys out of the sewer. You get to your door and realize someone has stolen your Amazon package that was delivered today.

Unfortunately, the case described is a very common and possible scenario. What if all of this could be curtailed? A good solution to this is using a smart lock with a camera for authentication.

With a growing number of online retailers and the convenience of having any product delivered to one's doorstep at the click of a few buttons, the number of packages being delivered each year is on the rise. Unfortunately, these packages are also commonly left unguarded, leaving the perfect opportunity for a malicious individual to steal the package while you're away from the house. It becomes an even bigger problem when a package contains a costly item. A major issue with package thieves and home burglary type crimes, in general, is the lack of evidence. Only a very small number of such crimes actually have enough evidence to identify a suspect. One existing solution to the package theft issue is Amazon Key, a system by which authorized Amazon deliverers can enter your home to deposit your package. This solution, however, is hindered by both it's cost and limitation to one online retailer.

A large majority of home security systems today are both expensive and complicated. Two improvements to home security developed in recent years, namely keyless locks and home security cameras, are both sold separately, and individually pricey. These systems attempt to modernize home security by providing a product that allows homeowners to monitor and control their home remotely while also increasing home security to allow entrance into the home. We would like a cost-effective system that integrates the keyless lock and home security camera solutions to home security, and use them to simultaneously improve upon a consumer's experience with package delivery.

In order to rectify the aforementioned problems, homeowners need an easy-to-use home security system, where a video camera is used to both monitor the house and authenticate users using facial recognition, condensing the functionality of both the keyless lock and home security camera systems. Such a system will be an improvement on current home security systems by allowing homeowners easy entry into their homes while also allowing many other features.

Conventional locks rely on physical keys. However, these can either be easily lost or easily replicated which is a cause for concern. Smart locks, on the other hand, are much more secure because they use other means of authentication. Most current smart locks do not necessarily have an option for unlocking using facial authentication (see Ring). Rather, they rely on apps or pass codes which can be broken into relatively easily. Facial authentication, on the other hand, allows for the user to easily unlock the door.

Then, there exists the need for security for a homeowner. Using the camera, a user can monitor the home remotely by viewing the video feed from the camera. The system can also monitor movements and faces using facial and motion detecting. Current home security systems either require a subscription or charge an exorbitant amount of money for cloud storage of video. Our proposal is less costly for the user while providing a secure, user friendly system.

For the purposes of this project, the current plan is to use a computer webcam for the video camera, and the lock will be assumed to already have been set up such that the lock defaults to being locked, and unlocks upon the sending of an unlock signal. Residents of the household will be able to enter their house through facial authentication, while package couriers will be able to scan the package barcode to allow themselves into the house to deposit the package, thus eliminating package theft. Records of all authentications will be able to be monitored through the use of a web application tied to the camera, with the option to be notified in the event of suspicious activity. A user will be notified of events through SMS and/or push notifications through a (mobile) browser which is the best option to communicate effectively with the user.

Users will be classified into three categories. The first category consists of homeowners. Users in this category will have the option to add or remove authenticated users and monitor the camera. Homeowners can monitor the camera through a live video feed. They will also be able to see a list of all the people that entered the home within a given timeframe in the past. All of this shall be available to homeowners through a web app. They can opt in for notifications on their browser or through SMS.

The second category will consists of residents of a home. Note that this category does not include homeowners. These users will be able to unlock the door for an indefinite amount of time, but do not have the ability to monitor or control the lock itself. That is, residents of a home do not have administrative privileges over the lock.

The third category of users consists of visitors. Couriers can be considered a subset of visitors as well. Visitors will be able to temporarily unlock the door for either a determinate amount of times or until a given date.

When a homeowner wants to allow authentication for temporary visitors, the homeowner can login to the web app and add the visitor using the web page. The homeowner can add a picture for facial authentication of the visitor. Or, the homeowner can create a temporary URL that they can send to the visitor. The homeowner can also set an expiration date for the authentication of the visitor.

When a homeowner wants to allow a courier to enter the package, the homeowner will enter the tracking number on the web app and the company that is shipping the package. Then, when the courier comes to the door, the courier can scan the shipping label of the package using the camera. Our system then verifies with the shipping company that the label corresponds to the tracking number. The door then unlocks for the courier to be let inside and drop off the package. The door will automatically lock once the courier exists and closes the door. This will prevent the homeowner's packages from being stolen.

If an unauthorized or unrecognized visitor approaches the door, a notification will be sent to the homeowner. The homeowner can view a live video and choose to report the stranger if they perform suspicious activity. The neighbors will be notified about the events that suspicious visitors show up so everyone will be aware of potential robbers and other threats in the neighborhood. This smart lock not only keeps the home safe but the community as well.

For a more secure neighborhood, homeowners need a new network called the Stranger Reporting Network. This network allows homeowners to report suspicious activity that they detect in their area. Other users in the neighborhood can then view a heatmap of suspicious activity in the neighborhood. This feature will be very useful in getting homeowners to be more vigilant to protect homes.

In order for the user to effectively use this product, the interface will be designed to be as user friendly as possible. The interface will revolve around a modern and simplistic design, in order to reduce the clutter that the user may become confused by. Different menus with categories such as statistics and options will be tucked away in a side menu, accessible with one click of a button.

This product is a much needed improvement upon existing home security systems and smart locks. Which provides the user friendly interface with features that are helpful to the homeowners such as SMS notifications, a live video feed, cloud storage (to a specified limit), visitor and courier authentication and the Stranger Reporting Network.

1.b. Glossary of Terms

- **Resident**: A person permanently authorized to open the Lock via facial recognition.

- **Homeowner**: A Resident of the home with the ability to log into the Web Application, and thus has the ability to add users to the system, change settings regarding the lock, and perform any other task allowed via the Web Application.

- **Authorized Visitor**: A person who is provided temporary access to the Lock via a temporary URL or given the ability to be temporarily authenticated via facial recognition by the homeowner.

- **Couriers:** People who deliver packages to the door. These are a special type of Visitor who is allowed short term access so that they can drop packages inside the home.

- **Web Application:** A client-server application presented over the web. This is primary platform on which the Smart-Lock interface accessed by the Homeowner will operate.

- **Application Programming Interface (API):** A shared interface that allows different subsystems to communicate between each other without concern for the underlying implementation of other subsystems.

- **RESTful API**: A shared interface which functions through stateless operations.

- **Front-end**: The interface with which the user interacts directly. In a RESTful API, this includes the user interface as well as the utilities necessary to send requests to the Back-end.

- **Back-end**: The interface which handles the significant operations that should not concern the end user like querying the database and transmitting data. In a RESTful API, the server is accessed through API calls directly.

- **HTML:** The standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

- **CSS:** A style sheet language used for describing the presentation of a document written in a markup language like HTML.

- **JavaScript**: The Front-end scripting language of the internet. This is the language we will use handle client-side functionality in the Web App.

- **WebSocket**: A bidirectional communication protocol via a TCP connection. Has been adopted by JavaScript as a standard means of stateful communication.

- **Python**: A general-purpose scripting language with a focus on readability. This is the language we will use to handle server-side (Back-end) functionality.

- **PostgreSQL**: A widely used open source relational database system.

- **Django**: A monolithic Python based web framework for developing Web Applications. Includes a system for user created extensions for specific purposes.

- **Flask:** A microframework for Python based on Werkzeug and Jinja 2.

- **Computer Vision**: The field relating to computers acquiring, processing and analyzing digital images in order to solve problems.

- **Computer Vision Module:** Part of the Back-end server that handles Computer Vision related processes, including but not limited to facial recognition, object detection, motion detection.

- **Facial Recognition**: A segment of Computer Vision which uses biometric data to analyzes the face of the person.

- **Motion Detection:** A set of monitoring algorithms which detects when there is movement in front of the camera.

- **OpenCV:** An open source Computer Vision and machine learning software library. In this project the library is used for its mature Facial Recognition features.

- **Temporary URL:** A temporary "key" that allows a user to unlock the door via a hyperlink to a QR code.

- **SMS:** The system by which text messages are transferred over the cellular network. In context of the project, these messages are an option for notifying the Homeowner.

- **Twilio**: A service which provides SMS and voice communication APIs.

- **Stranger Reporting Network:** An integrated network of strangers reported by the users and saved in the server's database.

2. Systems Requirements

(note: 5 is the highest priority)

Req. Identifier	Priority	Requirement
REQ-1	5	Residents shall be able to unlock the door through facial recognition.
REQ-2	3	Residents shall be able to unlock the door through a web app without facial authentication in the event that face recognition fails.
REQ-3	4	Residents shall be able to register other household members able to unlock the door via facial recognition.
REQ-4	2	Residents should be able to create a temporary URL to allow a visitor to unlock the door.
REQ-5	4	Residents shall be able to temporarily add people that can unlock the door.
REQ-6	4	Residents shall be able to control and get SMS messages for events that occur.
REQ-7	2	Residents should be able to report suspicious activity to the SRN.
REQ-8	2	Couriers should be able to open the door based off information found on a package expected by the user.
REQ-9	2	The User should be able to enter tracking information for a package to allow authentication of package Couriers

2.a. Enumerated Nonfunctional Requirements

Req. Identifier	Priority	Requirement
REQ-10	4	The system should provide desktop and mobile-friendly web pages.
REQ-11	4	The web interface should be intuitive and user-friendly.

REQ-12	5	The web application should be quick and easy to access and log into.
REQ-13	3	The user should be able to see an organized list of all the people that entered the home.
REQ-14	3	The system should be able to log all events related to the Lock, and be able to present this to the homeowner.
REQ-15	2	Residents may receive notifications via push notifications on a desktop.
REQ-16	1	Residents may receive notifications via push notifications on mobile.
REQ-17	5	The system should have security features to prevent unauthorized access into the Web Application and into the homeowner's house.

2.b. On-Screen Appearance Requirements

Req. Identifier	Priority	Requirement
REQ-18	5	The web page shall have a login page for homeowners to access and configure their Smart Lock.
REQ-19	4	The web page shall have the main page with a sidebar that links to other features of the web application.
REQ-20	3	The main page should have a button that allows homeowners to unlock the lock (remotely) on the main page.
REQ-21	2	The 'Stranger Reporting Network' page should provide homeowners with a map of suspicious activity in their area and provide a field to manually report suspicious activity.
REQ-22	3	The Live Feed page should allow users to see a live feed of the camera.
REQ-23	3	The Settings page should have tabs corresponding to each category of settings, with checkboxes to enable or disable settings.

REQ-24	3	The Data Visualization page should contain clickable tabs that each provide a different graphic visualizing statistics generated by the lock.
REQ-25	4	The Visitors page should provide buttons to add the photos of visitors the homeowner wishes to allow into the home, and to generate a temporary URL to send to a visitor that grants access into the home. The visitor profile has its own settings/configurations of authority.
REQ-26	3	The Deliveries page should provide a box giving a brief overview of scheduled deliveries, and a button to allow homeowners to add the tracking numbers of packages.
REQ-27	2	Residents can provide Couriers with another temporary code easily.
REQ-28	4	There should be a help button which will bring up a detailed user manual.

3. Functional Requirements Specifications:

3.a. Stakeholders

- A Homeowner / Landlord
  - Homeowner should be able to efficiently and easily safeguard, monitor, and grant accessibility to their home. The homeowner should be able to achieve such tasks with minimal human intervention. Additionally, the homeowner should be able to report and convey the presence of potentially malicious actors, and safely receive any package to their home without needing to be present at home.

- Residents
  - Other residents of the home should be able to easily unlock the lock device an unlimited amount of time through facial recognition. Residents should also be able to receive notification, watch live video feeds, and allow guests in the house. However, residents are limited in their ability to change the options and settings.

- Visitors/Courier
  - Visitors are allowed temporary access to the house. This can be over a certain period of time or a limited amount of access. Visitors can be authorized through facial recognition or temporary URL. Couriers must scan the barcode of the package and are allowed a one time access into the house afterwards.

3.b. Actors and Goals

Actors	Goals
Homeowner (Initiating)	To safeguard and monitor their home against package theft and home intrusions in addition to having a secure and simple method of entering their own home.
Resident (Initiating)	To easily enter their own home without needing a key or other physical object.
Visitor (Initiating)	To easily enter the home of a homeowner using the smart lock if authorized.
Stranger (Initiating)	To perform suspicious activity or break into the home of the homeowner.
Courier (Initiating)	To deposit packages safely inside the homeowners house when the homeowner is not present.
Camera (Participating)	Capturing and delivering video footage of the home entrance to the server. Used for motion detection and facial recognition as well.
Database (Participating)	SQL database storing information associated with households including but not limited to faces, authentication URLs, and logs of events.
Lock Device (Participating)	To keep the home secured by unlocking for only authorized individuals.
Server (Participating)	To manage the network between the devices, software, and database.
Scanner (Participating)	To temporarily unlock the door by scanning the barcode of a delivered package or QR code from the temporary URL.
Neighbors (Participating)	To have access to reported suspicious activity by the homeowner.

3.c. Use Cases

3.c.i Casual Description

UC-1: Unlock - Gives the homeowner and residents ability to unlock the door through facial recognition.
Derived from requirements REQ-1, REQ-2, REQ-3, REQ-5, REQ-6.

UC-2: Notifications - Allow homeowners to receive notifications about events.
Derived from requirements REQ-6, REQ-13, REQ-14, REQ-15, REQ-16, REQ-18, REQ-19.

UC-3: Stranger Reporting - Allow homeowners to report suspicious activity in the neighborhood.
Derived from requirement REQ-7, REQ-13, REQ-14, REQ-18, REQ-21.

UC-4: Temporary Visitor Authentication - To allow visitors to the home in one(or more) time(s) using a URL or temporarily adding them to trusted faces.
Derived from requirement REQ-4, REQ-5, REQ-13, REQ-14, REQ-18, REQ-19, REQ-25.

UC-5: Package Delivery - To allow couriers to unlock the door so they may deposit packages.
Derived from requirements REQ-8, REQ-9, REQ-26.

UC-6: Add Visitors - To add residents as visitors with permission to enter the house with facial recognition.
Derived from requirements REQ-1, REQ-2, REQ-3, REQ-4, REQ-25.

UC-7: Data Visualization - To allow homeowners to view data pertaining to who enters the house and when then enter.
Derived from requirement REQ-13, REQ-14, REQ-24.

UC-8: Live Video Feed - To obtain a live feed of the home entrance through the camera.
Derived from requirement REQ-22.

UC-9: Delete Visitor - To delete users that had permission to enter the house.
Derived from requirements REQ-1, REQ-2, REQ-3, REQ-4, REQ-25

UC-10: Options/Settings - To allow homeowner to change his notification and lock settings.

Derived from requirements REQ-6, REQ-23

UC-11: User Manual / Help - To aid in the use of the Smart Lock interface.
Derived from requirement REQ-28

3.c.ii Use Case Diagram

Use Case UC-1: Unlock (through face ID)	
Related Requirements	REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-6
Initiating Actor	Any of: Homeowner, Residents, or authorized visitor
Actor's Goal	Homeowners and residents unlock the door through facial recognition.
Participating Actors	Camera, lock device, database, server
Preconditions	Facial Recognition Camera video upload to server Residents data stored in database
Postconditions	Door locks again after entering
Flow of Events: 1. Initiating actor approaches the door. 2. Computer vision module detects motion and triggers facial recognition. 3. The facial recognition algorithm identifies the face as an authorized person. 4. System signals the lock device to unlock. 5. After door is closed again, lock automatically	

Use Case UC-2: Notifications	
Related Requirements	REQ-6, REQ-13, REQ-14, REQ-15, REQ-16, REQ-18, REQ-19
Initiating Actor	Visitor, courier, or other events
Actor's Goal	Allow homeowners to receive notifications about events.

Participating Actors	Camera, server, and homeowners receiving the notification
Preconditions	An event occurs.
Postconditions	The owner is informed of the event.
Flow of Events 1. An event occurs such as someone coming up to the door or a package being delivered. 2. This information is sent to the server. The server then sends a notification (SMS or push) to the user’s device. 3. The user sees the notification and is informed of the event.	

Use Case UC-3: Stranger Reporting	
Related Requirements	REQ-7, REQ-13, REQ-14, REQ-18, REQ-21
Initiating Actor	Stranger
Actor’s Goal	Allow homeowners to report suspicious activity in the neighborhood.
Participating Actors	Homeowner, server, and neighbors
Preconditions	Suspicious Activity occurs
Postconditions	Suspicious activity is now viewable by others in the neighborhood.
Flow of Events 1. Stranger arrives in front of door. 2. Homeowner notified of unknown individual in front of door. 3. Homeowner sees suspicious activity. 4. Reports it to the network. 5. A snapshot of the individual is saved by camera. 6. Now, anyone in the neighborhood can view this on the map. Alt. Flow of Events 1. Stranger arrives in front of door. 2. Stranger attempts to break into house. 3. Both homeowner and local police notified.	

4. Snapshot of individual and clip saved by camera. 5. Anyone in neighborhood can view this on map.

Use Case UC-4: Temporary Visitor Authentication	
Related Requirements	REQ-4, REQ-5, REQ-13, REQ-14, REQ-18, REQ-19, REQ-25
Initiating Actor	Homeowners
Actor's Goal	To allow visitors to the home in for one time using a URL or temporarily adding them to trusted faces.
Participating Actors	Temporary visitor; lock device; Database
Preconditions	Temporary visitor must be authenticated.
Postconditions	Temporary visitor authenticated.
Flow of Events 1. Homeowner accesses authentication tab in web app UI; selects url authentication selection. Clicks on Create New URL. 2. Homeowner adjusts duration (or amt of uses) parameters and sends a generation request. 3. Database stores this information, and URL is returned. 4. Homeowner sends the URL key to temporary visitor. 5. Temporary visitor clicks on URL. 6. Temporary visitor activates the URL (clicking unlock on the page). 7. Use is recorded in database, and visitor is authenticated. Alt. Flow of Events (Temporary Facial Recognition Authentication) 1. Homeowner accessing authentication tab in web app UI; selects facial recognition authentication selection. Clicks on Create New Facial Recognition Authentication. 2. Homeowner adjusts duration (or amt of uses) parameters. 3. Homeowner uploads pictures (headshots) of the person. 4. Database stores this information. 5. Temporary visitor is authenticated upon facial recognition.	

Use Case UC-5: Package Delivery	
Related Requirements	REQ-8, REQ-9, REQ-26
Initiating Actor	Courier
Actor’s Goal	Unlock to deposit package
Participating Actors	Scanner, lock device, server
Preconditions	The package is being delivered.
Postconditions	The delivery person is let into the home to put the package inside.
Flow of Events: 1. Homeowners give the service tracking number. 2. A delivery person comes to the door with the package(s). 3. Scans package barcode. It is authenticated through servers. 4. The delivery person is let inside. Deposits package and leaves.	

Use Case UC-6: Add Visitors	
Related Requirements	REQ-1, REQ-2, REQ-3, REQ-4, REQ-25
Initiating Actor	Homeowner
Actor’s Goal	Add residents as visitors with permission to enter the house with facial recognition.
Participating Actors	Residents, visitors
Preconditions	The ability to store a user with their own facial id tied to them.
Postconditions	None worth mentioning
Flow of Events: 1. A resident/visitor wants access to enter house. 2. Homeowner adds the individual as a user. 3. Camera scans face of user and stores it. 4. User now has access to enter house.	

Use Case UC-7: Data Visualization	
Related Requirements	REQ-13, REQ-14, REQ-24
Initiating Actor	Homeowner
Actor’s Goal	Allow homeowners to view data pertaining to who enters the house.
Participating Actors	Residents, Visitors
Preconditions	Homeowner, Resident, or Visitor unlocks the door.
Postconditions	Name of Visitor, time entered, and data is viewable by the Homeowner.
Flow of Events: 1. Homeowner opens web app. 2. Homeowner selects the time, date, or visitors. 3. App displays the desired data and homeowner can view easy-to-read logs, graphs, and charts on who entered the house and when.	

Use Case UC-8: Live Video Feed	
Related Requirements	REQ-22
Initiating Actor	Homeowner
Actor’s Goal	To obtain a live feed of the home entrance through the camera.
Participating Actors	Camera
Preconditions	Homeowner is logged into the website and camera is functioning.
Postconditions	None
Flow of Events: 1. Homeowner selects the live streaming tab on the web app. 2. Camera sends current video feed to the server. 3. Homeowners can view the video on web app.	

Use Case UC-9: Delete Visitor	
Related Requirements	REQ-1, REQ-2, REQ-3, REQ-4, REQ-25
Initiating Actor	Homeowner
Actor’s Goal	Delete users that had permission to enter the house.
Participating Actors	Residents, visitors, database
Preconditions	Exist visitors with access to the house
Postconditions	Visitor is no longer part of the database and does not have access to the house.
Flow of Events: 1. Homeowner opens visitor settings and select authorized visitors to delete 2. System deletes visitors from database, and they no longer have access to house. 3. System send notification to the user for successful removal.	

Use Case UC-10: Options/Settings	
Related Requirements	REQ-6, REQ-23
Initiating Actor	Homeowner
Actor’s Goal	Allows the homeowner to change his notification and lock settings.
Participating Actors	Database
Preconditions	Homeowner is logged into the web app.
Postconditions	Setting gets updated
Flow of Events: 1. User selects the settings page. 2. User chooses the setting category. 3. User can enable/disable whichever settings they desire.	

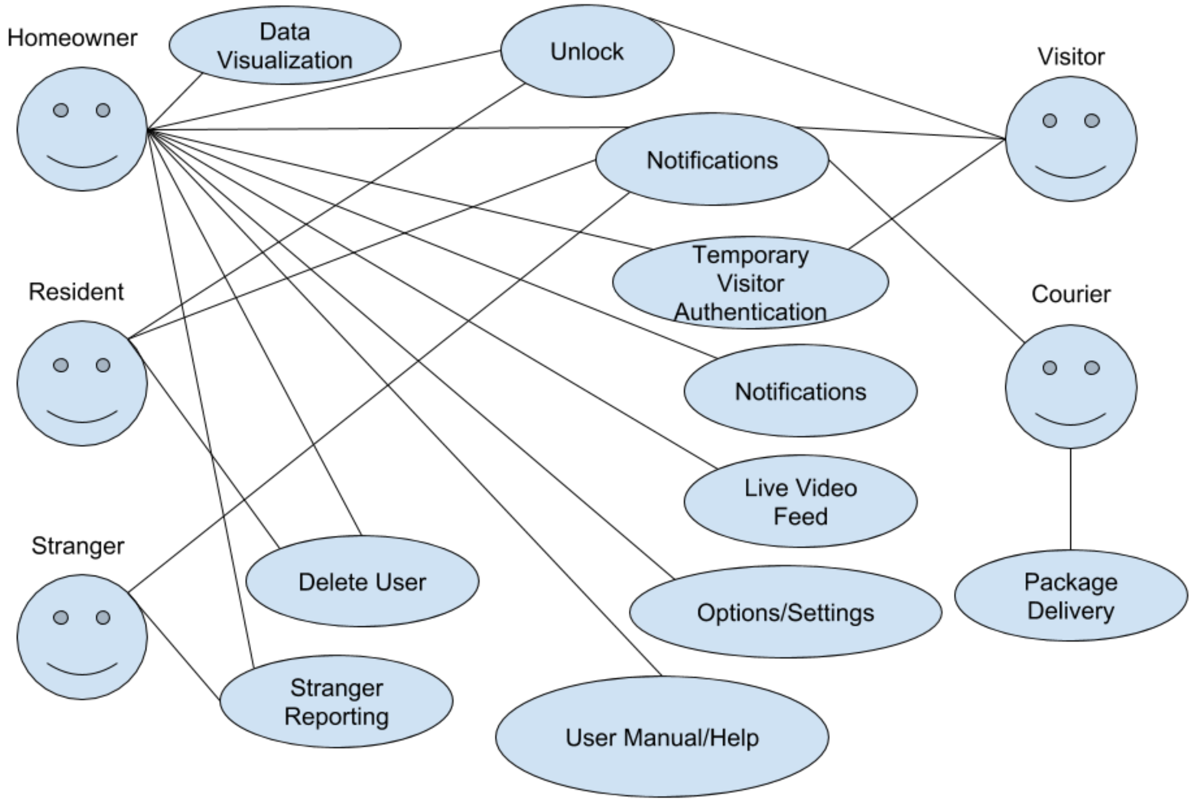
Use Case UC-11: User Manual / Help	
Related Requirements	REQ-28
Initiating Actor	Homeowner
Actor’s Goal	Open a manual in order to figure out a feature the user is unfamiliar with.
Participating Actors	Server
Preconditions	Homeowner is confused about how to use a feature or wants to find out what a feature does.
Postconditions	Homeowner receives all information needed to navigate the system
Flow of Events: 1. User selects the help icon (?). 2. The web app displays a user manual 3. User may browse through or search a specific feature	

3.c.iii Traceability Matrix

REQ-#	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
REQ-1	5	X					X			X		
REQ-2	3	X					X			X		
REQ-3	4	X					X			X		
REQ-4	2	X			X		X			X		
REQ-5	4	X			X							
REQ-6	4	X	X								X	
REQ-7	2			X								
REQ-8	2					X						
REQ-9	2					X						
REQ-10	4											
REQ-11	4											
REQ-12	5											
REQ-13	3		X	X	X			X				
REQ-14	3		X	X	X			X				
REQ-15	2		X									
REQ-16	1		X									
REQ-17	5											
REQ-18	5		X	X	X							
REQ-19	4		X		X							
REQ-20	3											
REQ-21	2			X								
REQ-22	3								X			
REQ-23	3										X	

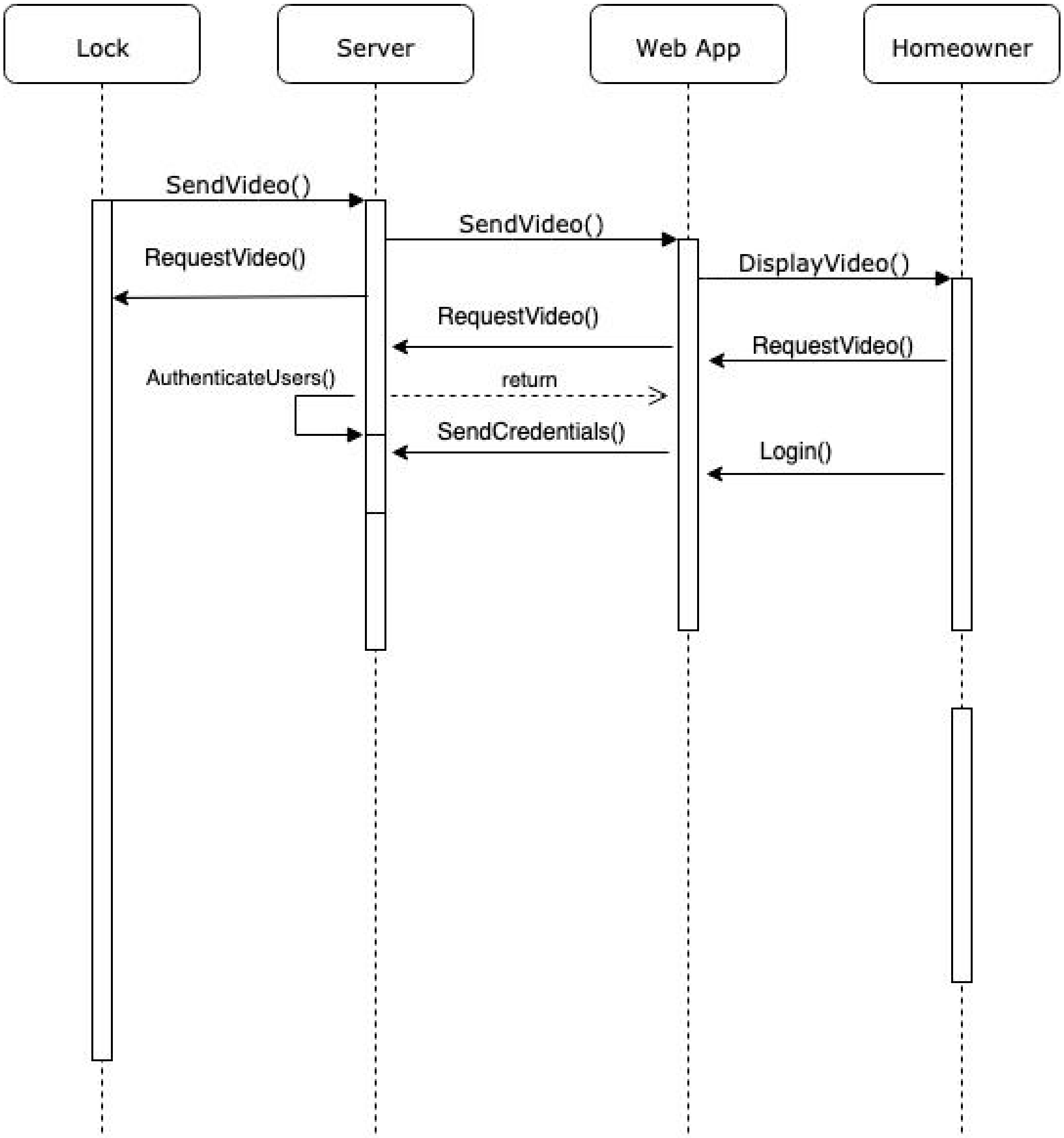
REQ-24	3							X				
REQ-25	4				X		X			X		
REQ-26	3					X						
REQ-27	2											
REQ-28	4											X

3.c.iv Fully-Dressed Descriptions

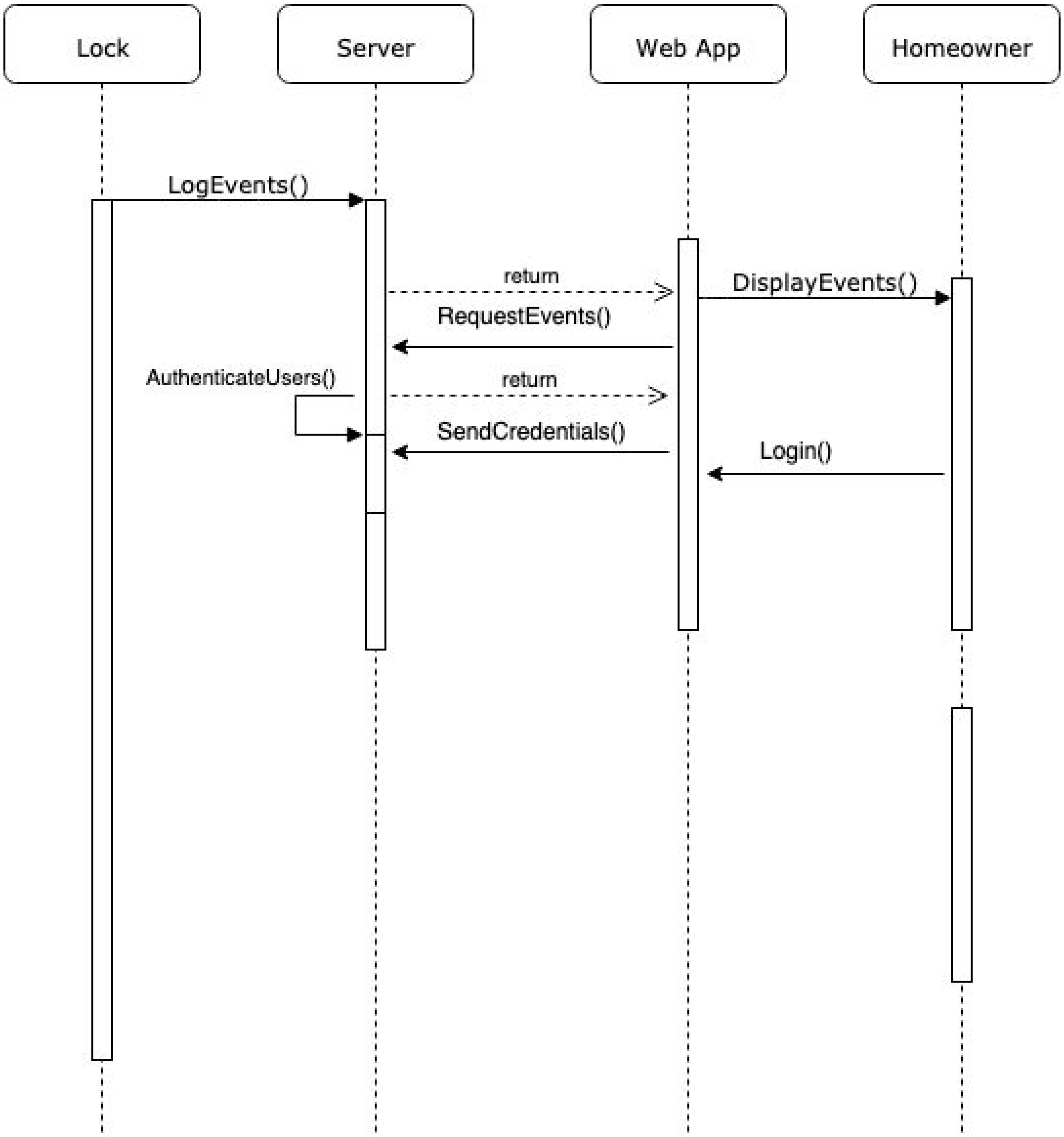


3.d. Systems Sequence Diagrams

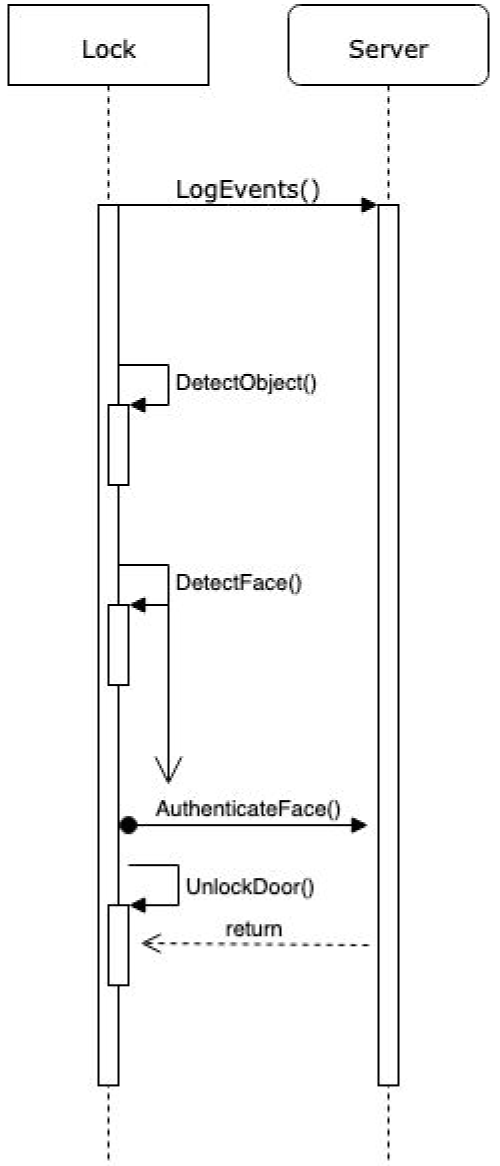
3.d.1 Sequence Diagram for UC - 8



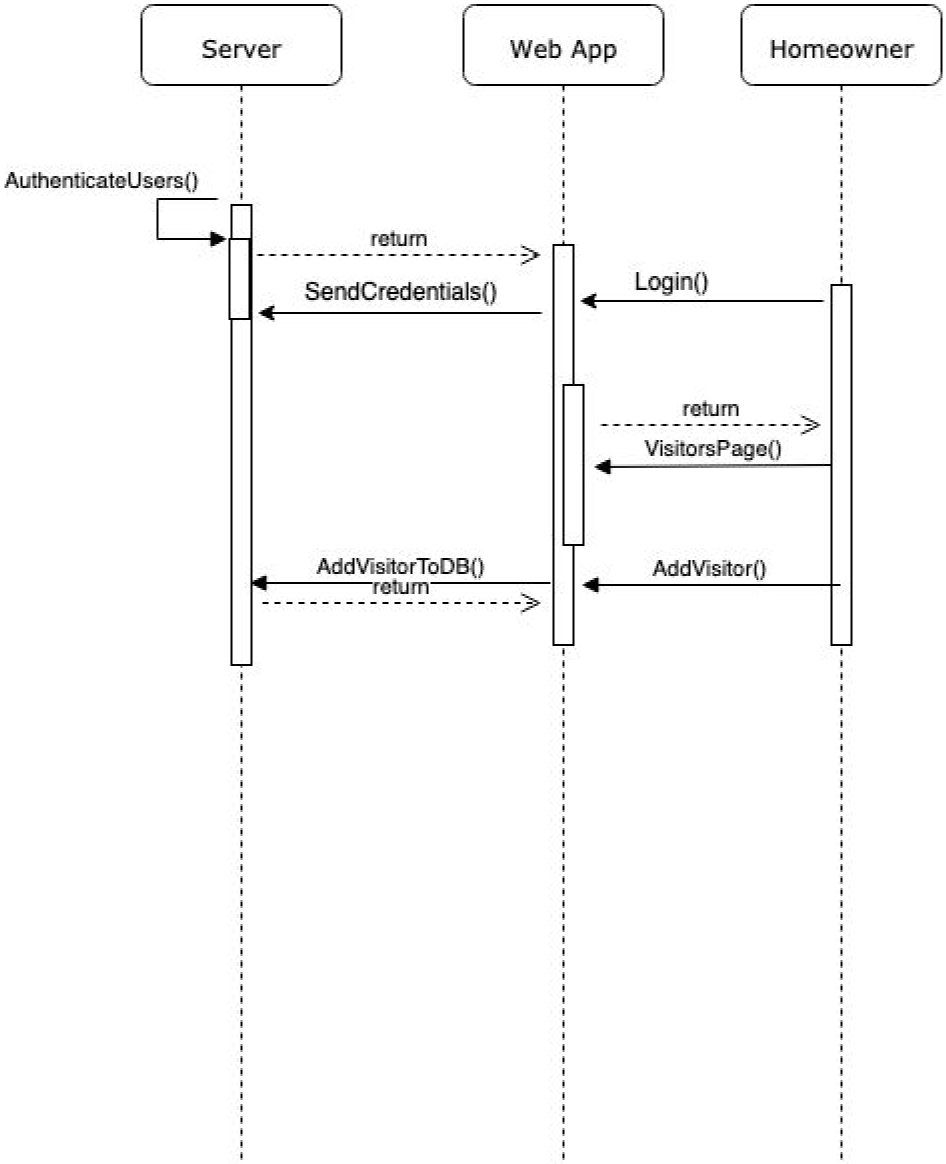
3.d.2 Sequence Diagram for UC - 7



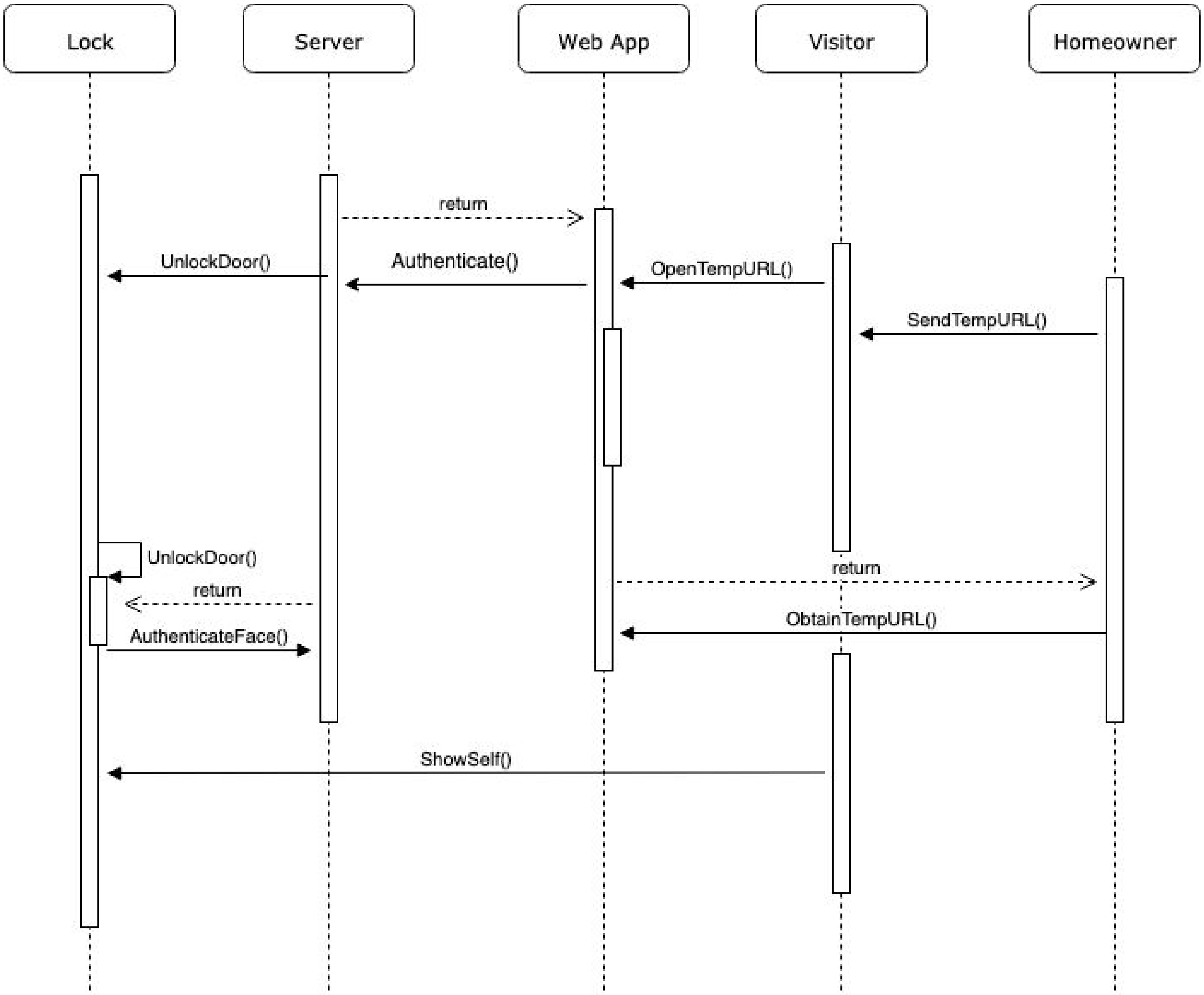
3.d.3 Sequence Diagram for UC - 1



3.d.4 Sequence Diagram for UC - 6



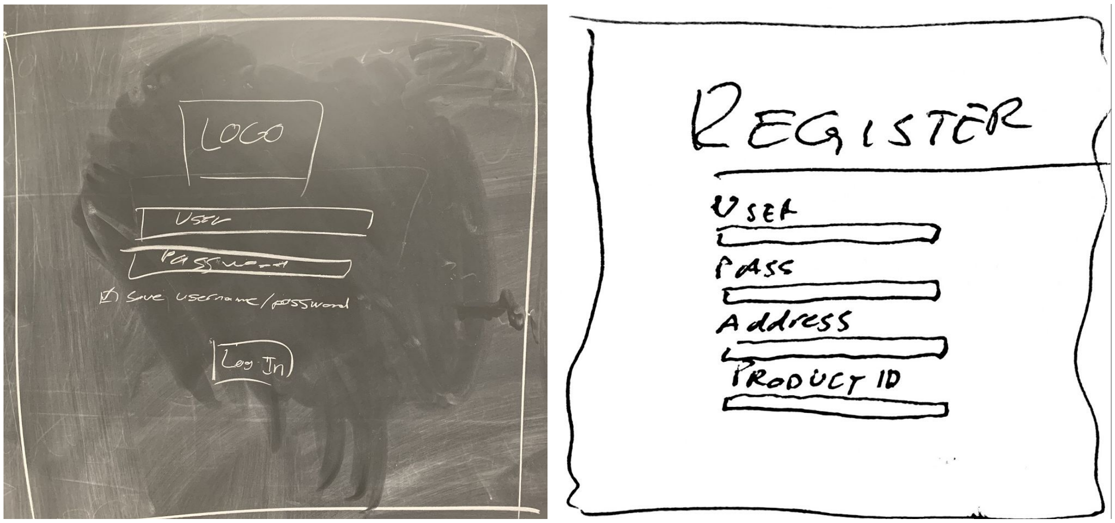
3.d.5 Sequence Diagram for UC - 4



4. User Interface Specification

4.a. Hand Drawn Designs & User Effort Estimation

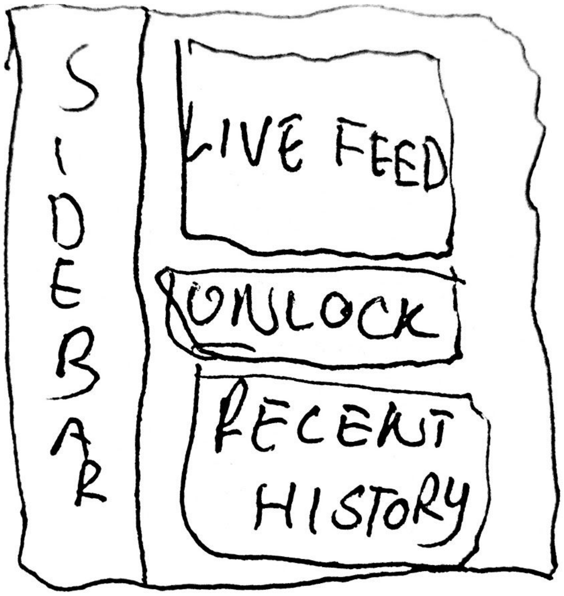
Login / Register



In order to authenticate the owner the owner must first enter their username and password. The layout consists of the username and password field also with a login button. Although not drawn on the layout, the UI will also have a register button next to the login button that will allow for the user to register for an account by entering their username, password, street address, and Product ID. This will allow for the customer to link their smart lock with their account.

User Effort Estimation
Authentication will require four clicks. The user will first click the login button, enter his/her username and password, and click the sign in button. Then, the user will click the the register button, enter in the additional information to link the smart lock to the account, and click register.

Main Menu

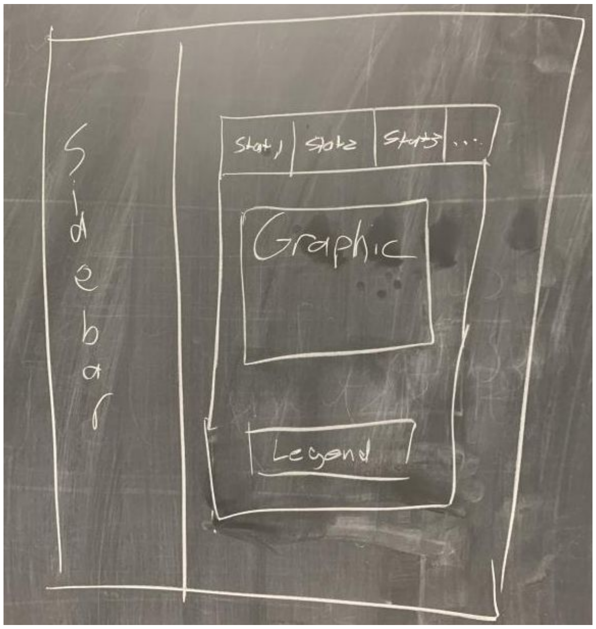


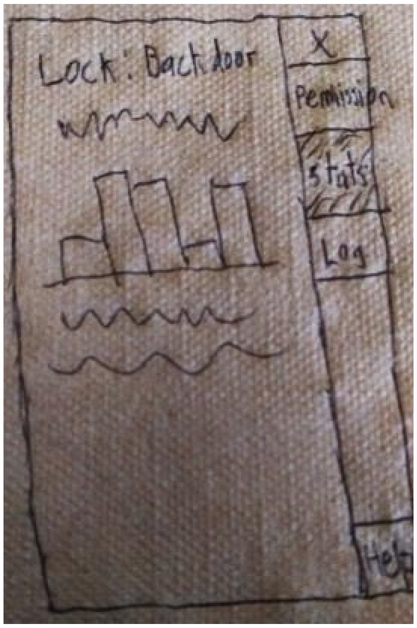
The main menu (once the user has signed in) will consist of a sidebar displaying links to other important screens such as the log, settings, visitors / authentication, and statistics (data visualization). The home page will be very minimalistic. It will contain live feed of the lock (camera), an unlock button which will allow for the owner to manually unlock their door without having to use the facial authentication or URL authentication. Under the unlock button the user can preview some recent history (only the first few entries; the rest of the history will only be shown when the user clicks on “logs” [in the sidebar]).

User Effort Estimation

The main menu will require two mouse clicks to access. The user will first click the login button, then enter his/her username and password and click the sign in button.

Data Visualization



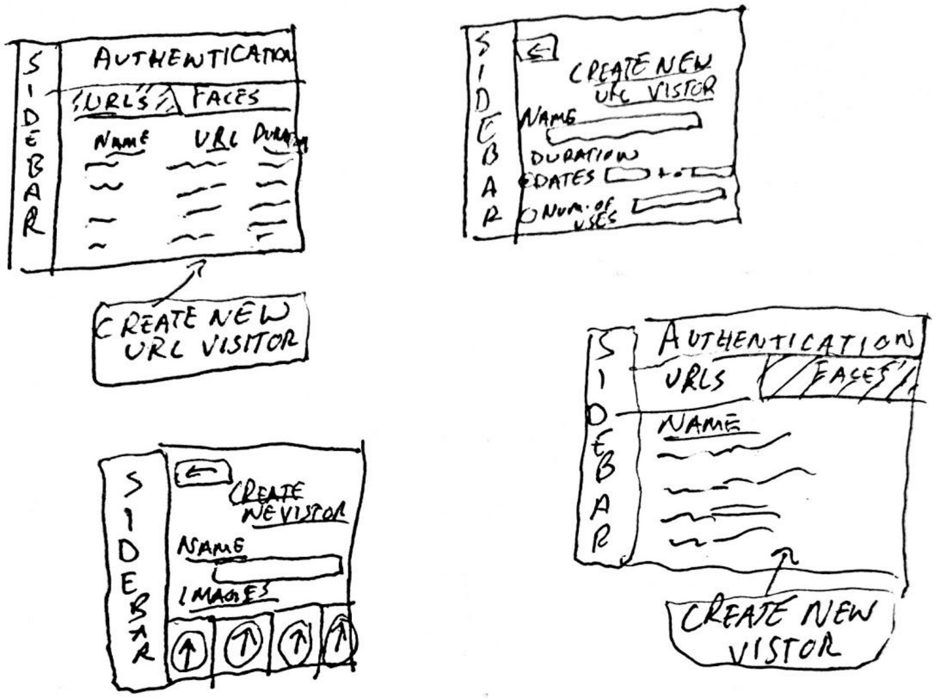


Data Visualization will conveniently be located under the sidebar for the user; labeled "stats." This page will contain graphics centered right in the middle of the viewport, with legends / any other information on the bottom. On the top, there will be tabs to other statistical information that would be important to visualize. The statistics displayed will include total number of visitors, time of day visitors are most active, and organized chronological history of when visitors are most active.

User Effort Estimation

From the main menu, the user will only require one left click to access the data visualization tab.

Automatic Visitor Authentication

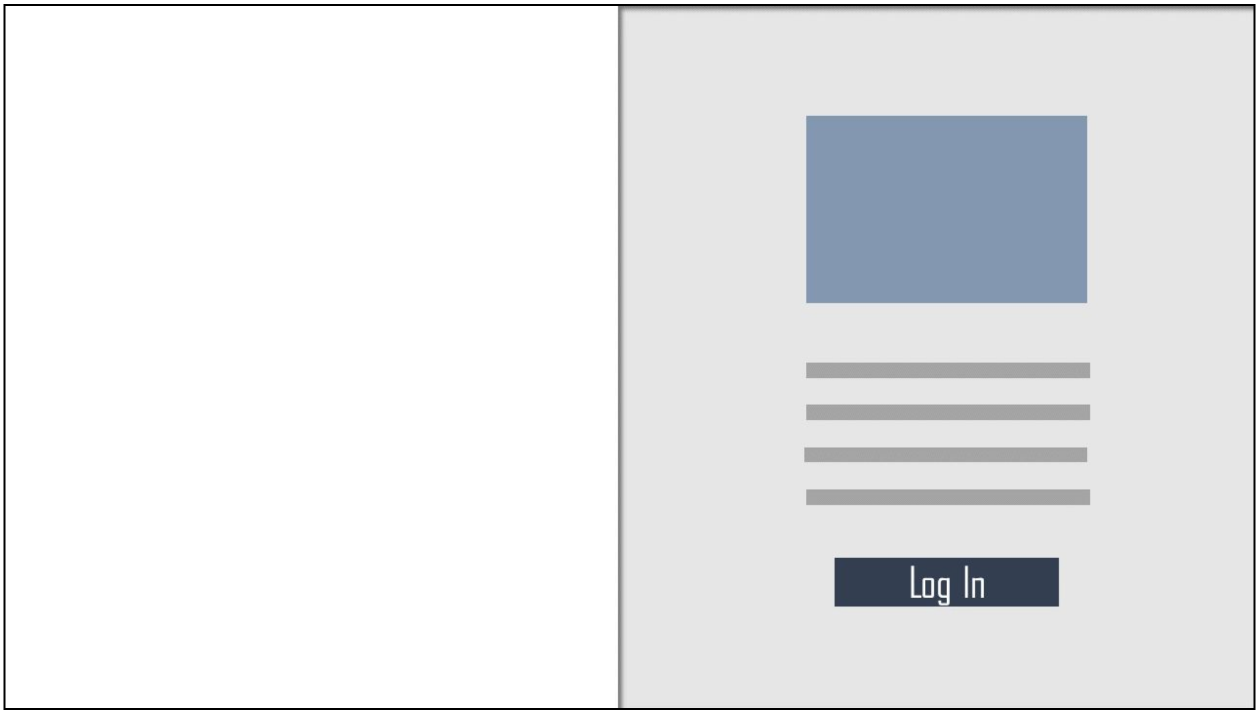


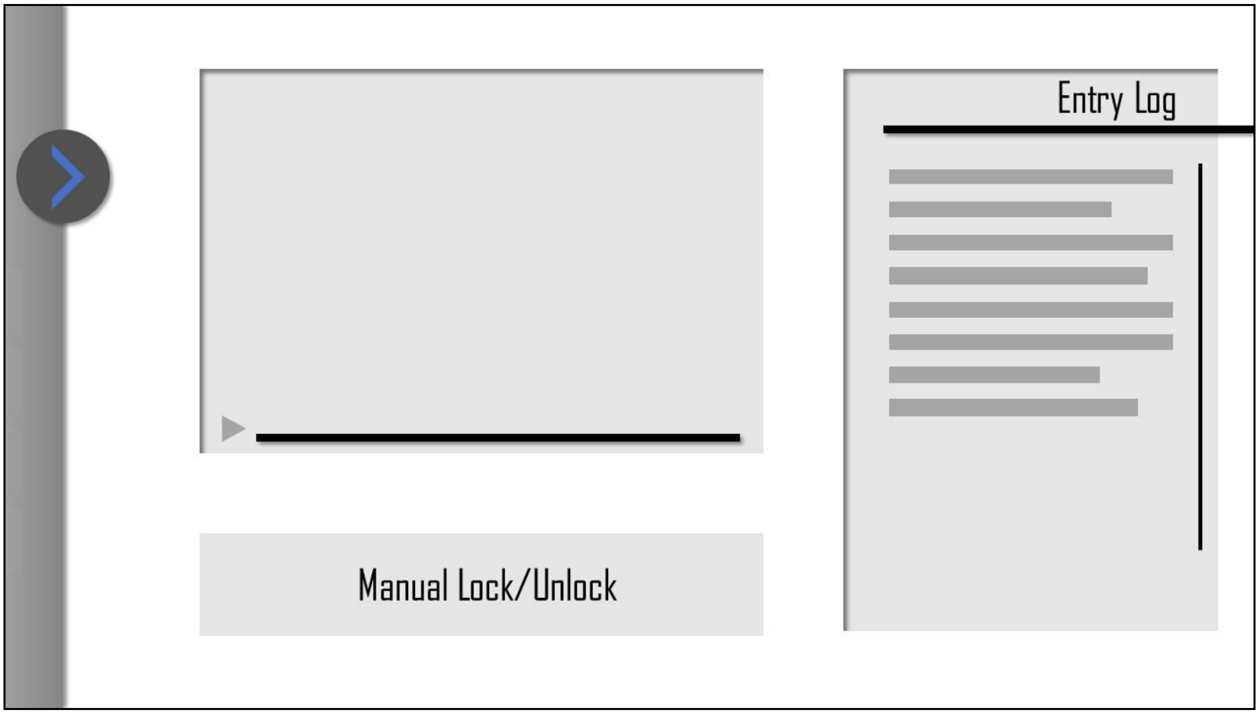
Automatic Visitor Authentication can be accessed by the user by navigating to the "Permissions" tab on the sidebar. This page will host two tabs, URLs and Faces. URLs contains active visitors that were permitted by the URL system. Faces contains visitors that were permitted by inputting images. The user may also create new visitors using the button below the entries or delete entries as well. When creating new visitors using the visitor for with URLs the user needs to specify a name as well as duration (the duration can be specified in different ways including dates and number of uses). When creating a visitor by inputting faces, the user may specify name and then upload multiple images which the system may catalog for facial recognition. The user may also specify duration for facial recognition as well if so desired (not shown).

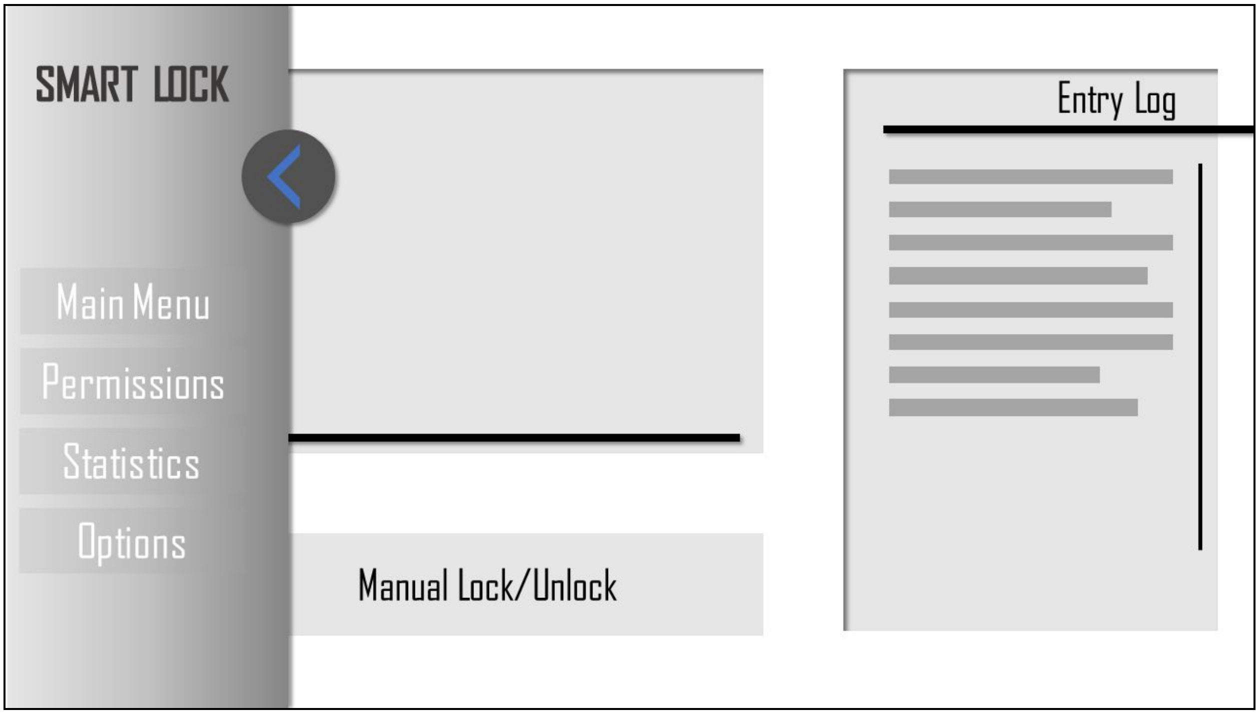
User Effort Estimation

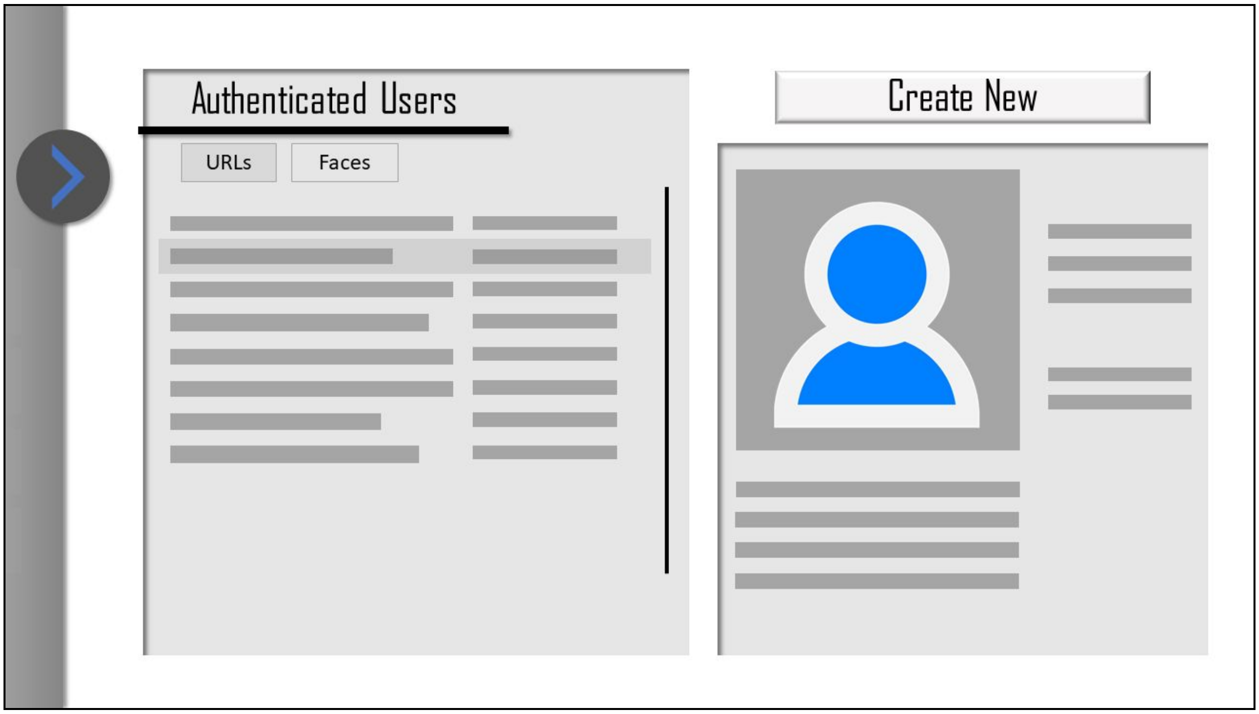
From the main menu, the user will only require *one left click* to access the "Permissions" tab. And then depending on the task the user wishes to complete, they may require anywhere from 0 to 6 interactions (depending on whether they wish to simply view visitors or enter a new person and specify specific durations).

4.b. Conceptual UI Mockup (Extremely Basic)






SMART LOCK
Main Menu
Permissions
Statistics
Options
Manual Lock/Unlock
Entry Log


Authenticated Users
URLs
Faces
Create New

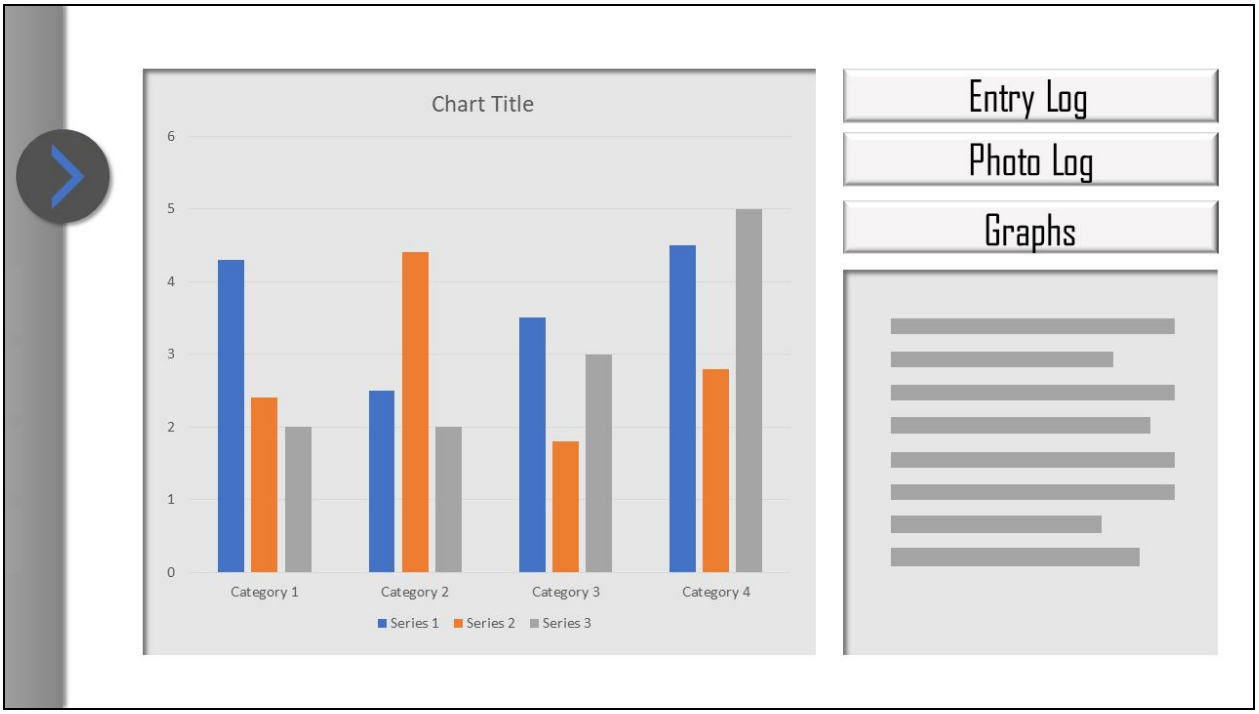


Chart Title
6
5
4
3
2
1
0
Category 1
Category 2
Category 3
Category 4
■ Series 1
■ Series 2
■ Series 3
Entry Log
Photo Log
Graphs

5. Domain Analysis

5.a. Domain Model

5.a.i. Concept Definitions

Note: The "Type" column denotes the category of responsibility.
D = "doing", K = "knowing", N = "none"

Responsibility Description	Type	Concept Name	Related Use Cases
To route requests from the user interface to the appropriate handler to service the request.	D	Controller	ALL EXCEPT UC-11
To log information needed for use cases in persistent storage, and retrieve information as needed by the handlers for use cases.	K	CentralDatabase	ALL EXCEPT UC-11
To detect if an object has passed in front of a camera.	D	ObjectDetector	UC-1, UC-3, UC-4, UC-5
To detect if a face is being shown in front of the camera.	D	FaceDetector	UC-1, UC-3
To analyze the face that has been detected and see if the face is associated with any existing users who have access to the lock.	D	FaceRecognizer	UC-1, UC-3
To unlock the door upon successful authentication.	D	DoorUnlocker	UC-1, UC-4
To log events for a lock in the database.	D	EventLogger	UC-2
To send an SMS notification to a user's phone number upon an event occurring.	D	SMSNotifier	UC-2
To send a push notification to a user's device upon an event occurring.	D	PushNotifier	UC-2

To report an unidentified person trying to gain access to the house associated with a lock.	D	StrangerReporter	UC-3
To display the reportings of strangers within an area.	K	StrangerMap	UC-3
To assign a temporary URL containing a QR code that can unlock the lock exactly once per QR code.	D	TempAuthAssigner	UC-4
To detect if a temporary authentication code is being displayed in front of the camera.	D	TempAuthDetector	UC-4
To verify that the temporary authentication code being presented is in fact valid for one-time unlocking.	D	TempAuthVerifier	UC-4
To detect if a barcode is being displayed in front of the camera.	D	BarCodeDetector	UC-5
To verify that the bar code being displayed is that of an expected package to be delivered to the house.	D	BarCodeVerifier	UC-5
To add packages and tracking number	D	PackageAdder	UC-5
To register allowed visitors along with access permissions for a certain lock.	D	VisitorAdder	UC-6
To log images of a newly added visitor's face for future use in facial recognition unlocking.	D	FaceScanner	UC-6
To display lock data to the user.	D	DataGenerator	UC-7
To fetch specific lock data based on specified fields.	D	DataFetcher	UC-7
To mirror the video feed from the camera wherever the user is	D	LiveFeedViewer	UC-8

viewing his/her lock information from.			
To delete a given visitor from the list of allowed visitors for a particular lock.	D	VisitorDeletor	UC-9
To update the settings of a user.	D	SettingsUpdater	UC-10
To show relevant information about the specific use case to the user and allow for user input related to the use case.	K	WebInterface	UC-2, UC-4, UC-6, UC-7, UC-8, UC-9, UC-10, UC-11

5.a.ii. Association Definitions

Concept Pair	Association Description	Association Name
Controller ←→ CentralDatabase	Controller sends an arbitrary request for data to the database and receives the data it requests.	Database request
ObjectDetector ←→ FaceDetector	If an object is detected, determine if the object is a face.	Detect face
FaceDetector ←→ FaceRecognizer	If a face is detected, determine if it is recognized in the list of visitors.	Recognize face
FaceRecognizer ←→ CentralDatabase	Verify the facial detection with the permissions in the central database for the particular lock.	Verify face ID
FaceRecognizer ←→ DoorUnlocker	If the face is recognized, unlock the door.	Face ID unlock
EventLogger ←→ SMSNotifier	Upon an event being logged, notify the user via SMS.	SMS notify
EventLogger ←→ PushNotifier	Upon an event being logged, notify the use via push notification.	Push notify

FaceRecognizer ←→ StrangerReporter	If a face is not recognized, report the unrecognized face to the Stranger Detection Network (SRN)	Report stranger
StrangerReporter ←→ StrangerMap	Updates the stranger map with the location of the stranger and the time of the stranger's visit.	Update stranger map
WebInterface ←→ TempAuthAssigner	Access the temporary authentication interface through the web page.	Access temp auth
ObjectDetector ←→ TempAuthDetector	If an object was detected, see if it is a temporary authentication code.	Detect temp auth
TempAuthDetector ←→ TempAuthVerifier	If a temporary authentication code was detected, see if it is valid to unlock the lock.	Verify temp auth
TempAuthVerifier ←→ CentralDatabase	Check the temporary authentication code with the database for permissions and extra verification	Confirm temp auth
TempAuthVerifier ←→ DoorUnlocker	Unlock the door after temporary authentication was confirmed.	Unlock temp auth
ObjectDetector ←→ BarCodeDetector	If an object was detected, see if it is a barcode.	Detect barcode
BarCodeDetector ←→ BarCodeVerifier	If a barcode was detected, see if it is valid.	Verify barcode
BarCodeVerifier ←→DoorUnlocker	If the barcode is valid, signal controller to unlock.	Barcode Unlock
WebInterface ←→ VisitorAdder	Access the visitor adder interface through the web page.	Access visitor adder

VisitorAdder ←→ Controller	Adds a visitor to the database with the given permissions.	Add visitor
VisitorAdder ←→ FaceScanner	Registers a newly added visitor’s face to the database, for facial recognition purposes.	Scan visitor face
FaceScanner ←→ Controller	Preps the database request to add a face image for a new visitor	Add new face
WebInterface ←→ DataFetcher	Access the data visualization interface from the web page, in the form of a data fetcher query.	Access data visualization
DataFetcher ←→ Controller	Prep the data fetcher query for sending to the database.	Prep data query
DataFetcher ←→ DataGenerator	Update data graphs based on the results of the query from the data fetcher	Update data graphs
WebInterface ←→ LiveFeedViewer	Access the live feed streaming from the camera.	View live feed
WebInterface ←→ VisitorDeletor	Access the control panel that removes authentication permissions from select person.	Access auth control
VisitorDeletor ←→ Controller	Prep visitor delete request for sending to the database	Prep visitor delete
WebInterface ←→ SettingsUpdater	Access the settings update interface from the web page.	Access settings
SettingsUpdater ←→ Controller	Prep the updated settings for update in the database	Prep settings update
Controller ←→ WebInterface	Updates web interface based on results of a database query.	Update web page

5.a.iii. Attribute Definitions

Concept	Relevant Use Case(s)	Attribute	Attribute Description
SMSNotifier	UC-2	User phone number	Phone number which SMS notifications should be sent to.
StrangerReporter	UC-3	Time of visit	The time the stranger approached the house.
		Stranger image	A captured image of the stranger from the camera.
StrangerMap	UC-3	User location	Location which the map should be centered around.
		Map radius	Max radius around the current location to show.
		Heatmap opacity	Determines the opacity of the heat
		Stranger filter	Filters the heatmap by specific stranger sightings.
TempAuthVerifier	UC-4	Temp code	Temporary authentication code generated for one-time unlocking.
BarCodeVerifier	UC-5	Tracking numbers	List of expected tracking numbers for the lock (retrieved from database).
VisitorAdder	UC-6	Visitor first name	First name of the

			visitor to be added.
		Visitor last name	Last name of the visitor to be added.
		Is visitor allowed	General access allowance (boolean). If this is “false”, the visitor is not deleted from the list of visitors, but he/she is not allowed access to the lock until this field is set to “true” again.
		Start time	Earliest time on a given day where a visitor can unlock the lock.
		End time	Latest time on a given day where a visitor can unlock the lock.
DataGenerator	UC-7	Data sets	Sets of data to display.
		Data formats	List of formats which the data should be displayed in. Each item in the list corresponds to a data set.
SettingsUpdater	UC-10	New settings	List of new user settings that should be applied on a settings update.
		Current settings	Current settings for the user, before any settings update

5.a.iv. Traceability Matrix

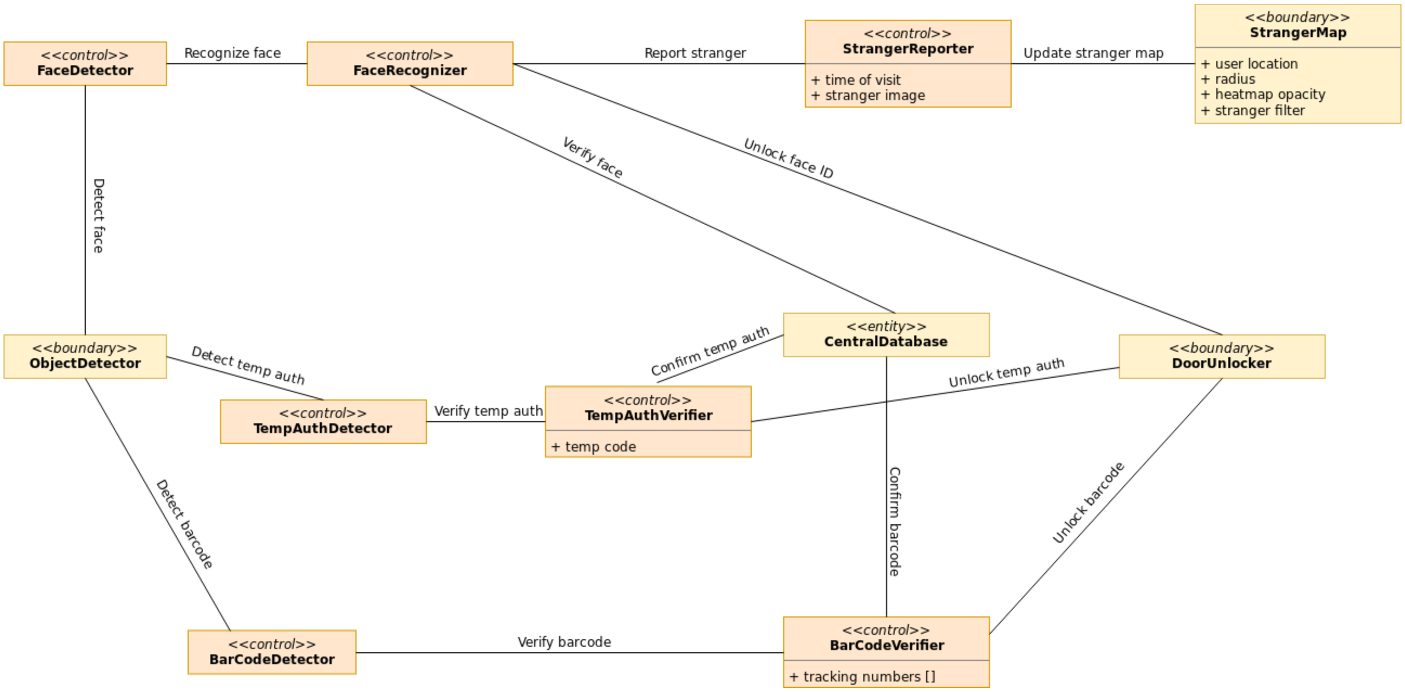
Domain Concepts	Use Case	1	2	3	4	5	6	7	8	9	10	11
	PW	15	10	10	5	10	10	15	15	5	5	5
Controller		X	X	X	X	X	X	X	X	X	X	
CentralDatabase		X	X	X	X	X	X	X	X	X	X	
ObjectDetector		X		X	X	X						
FaceDetector		X		X								
FaceRecognizer		X		X								
DoorUnlocker		X			X							
EventLogger			X									
SMSNotifier			X									
PushNotifier			X									
StrangerReporter				X								
StrangerMap				X								
TempAuthAssigner					X							
TempAuthDetector					X							
TempAuthVerifier					X		X					
BarCodeDetector						X						
BarCodeVerifier						X						
PackageAdder						X						
VisitorAdder					X							
FaceScanner							X					
DataGenerator								X				
DataFetcher								X				

LiveFeedViewer									X			
VisitorDeletor										X		
SettingsUpdater											X	
WebInterface			X		X	X	X	X	X	X	X	X

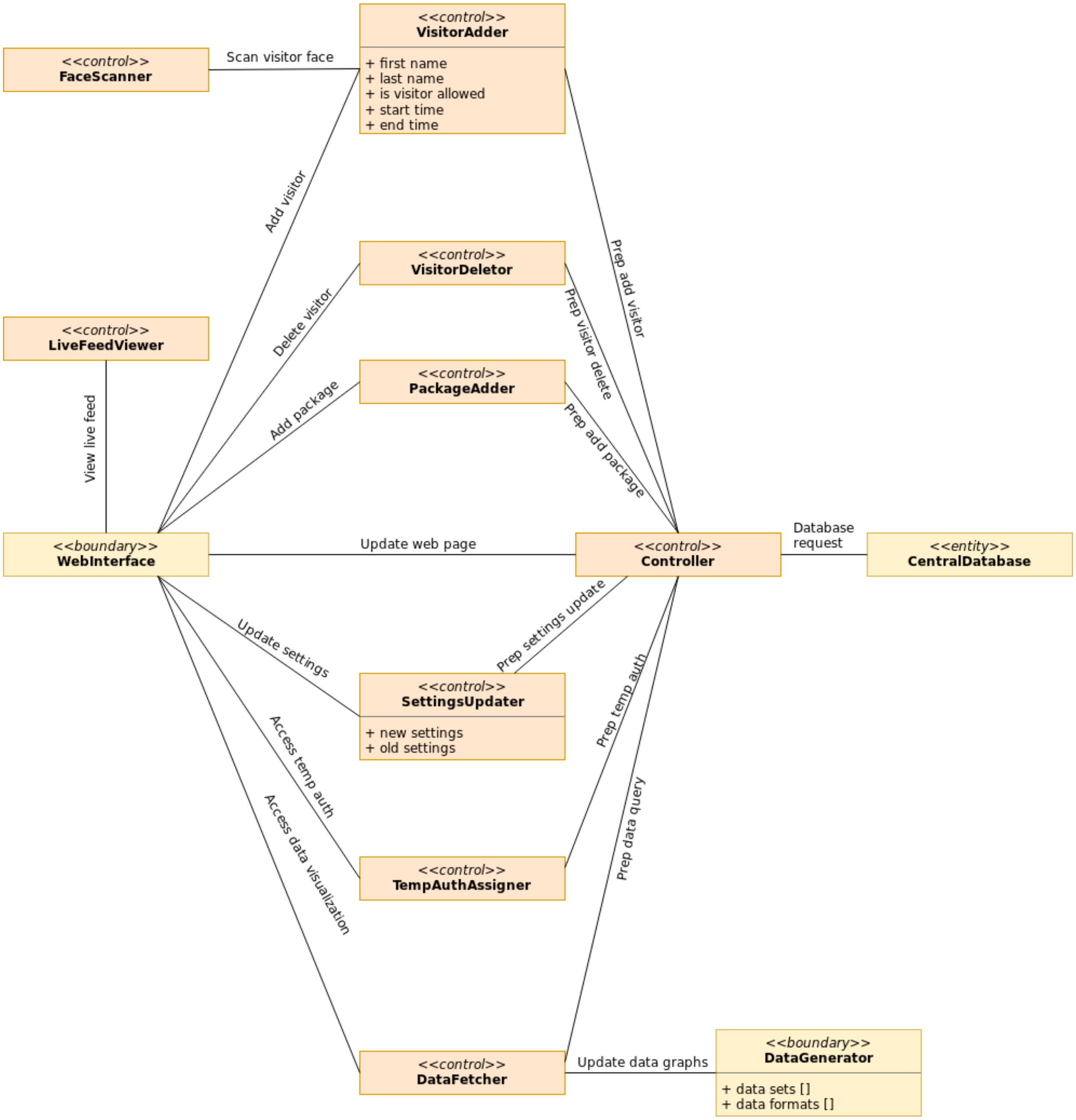
5.a.v. Diagrams

Due to the the large scope of the project, we have decided to split up a full representation of the problem domain into separate spaces. Each chart shown below is completely orthogonal to the others. That is, each chart can be treated as a “subsystem” for the total Smart Lock product. In general, all subsystems start from the same *boundary* of either the ObjectDetector or WebInterface concepts. From there, paths diverge depending on the goal of the use case.

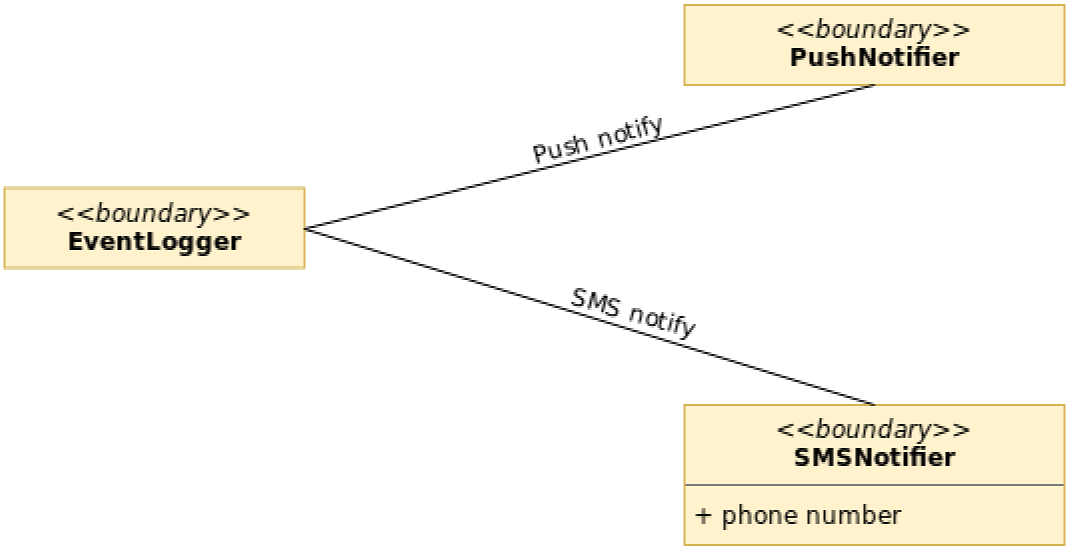
5.a.v.i. Boundary: ObjectDetector



5.a.v.ii. Boundary: WebInterface



5.a.v.iii. Boundary: EventLogger



5.b. System Operation Contracts

Operation	Unlock [UC - 1]
Responsibility	Automatically authenticate visitor using facial recognition.
Output	Make autonomously make a judgement on whether to unlock and door or not, and notify user.
Preconditions	Visitor approaches entrance. FaceDetector detects face.
Postconditions	Notify Homeowner. If (visitor ∈ authenticated_users) Unlock Else Reject

Operation	Temporary Visitor Authentication [UC - 4]
Responsibility	Visitor is authenticated temporarily using facial recognition or a URL key.
Output	Make autonomously make a judgement on whether to unlock and door or not, and notify user.
Preconditions	Visitor approaches entrance. User holds up QR code.
Postconditions	Notify Homeowner. If (visitor_qr ∈ authenticated_tokens) Unlock; key_uses-- Else Reject

Operation	Add Visitors [UC - 6]
Responsibility	Add visitor's face to database of faces for facial recognition authentication.
Output	Add user's faces to database. Ready for CV.
Preconditions	Homeowner inputs / uploads approximately 5 images of visitor.
Postconditions	Preprocess images. Communicate images to CentralDatabase's repo of images.

6. Project Size Estimation

Unadjusted Use Case Weight

Simple Use Cases	UC-4,9,10,11
Average Use Cases	UC-2,3,5,6
Complex Use Cases	UC-1,7,8

= (Total No. of Simple Use Cases x 5) + (Total No. Average Use Cases x 10) + (Total No. Complex Use Cases x 15)
=(4)*5+(4)*10+(3)*15
=125

Unadjusted Actor Weight

Simple Actors	Camera, Lock
Average Actors	Local Processor
Complex Actors	Server, Homeowner

=(Total No. of Simple Actors x 1) + (Total No. Average Actors x 2) + (Total No. Complex Actors x 3)
=(2)*1+(1)*2+(2)*3
=10

Technical Complexity Factor

Factor	Description	Weight	Assigned Value	Weight x Assigned Value
T1	Distributed system	2.0	5	10.0
T2	Response time/performance objectives	1.0	5	5.0
T3	End-user efficiency	1.0	4	4.0
T4	Internal processing complexity	1.0	4	4.0
T5	Code reusability	1.0	3	3.0
T6	Easy to install	0.5	1	0.5
T7	Easy to use	0.5	5	2.5
T8	Portability to other platforms	2.0	2	4.0
T9	System maintenance	1.0	4	4.0
T10	Concurrent/parallel processing	1.0	5	5.0
T11	Security features	1.0	5	5.0
T12	Access for third parties	1.0	1	1.0
T13	End user training	1.0	1	1.0
			TOTAL (TF)	**49**

TCF = 0.6 + (TF/100)
For the Smart Lock System, TCF = 0.6 + (49/100) = 0.99
TCF = 1.09

Environmental Complexity Factor

Factor	Description	Weight	Assigned Value	Weight x Assigned Value
E1	Familiarity with development process used	1.5	2	3.0
E2	Application experience	0.5	3	1.5
E3	Object-oriented experience of teamy	1.0	2	2.0
E4	Lead analyst capability	0.5	4	2.0
E5	Motivation	1.0	1	1.0
E6	Stable requirements	2.0	1	2.0
E7	Part-time staff	-1.0	0	0
E8	Difficult programming language	-1.0	1	-1.0
TOTAL (TF)				**10.5**

ECF = 1.4 + (-0.03 x EF)
*For the Smart Lock System, ECF = 1.4 + (-0.03 * 10.5) = 1.085*
ECF = 1.085

Use Case Points

UCP = (UUCW + UAW) x TCF x ECF
For the Smart Lock System, UCP = (125 + 10) x 0.99 x 1.085 = 135.06
UCP = 135.06
Estimated Effort = UCP x Hours/UCP
For the Smart Lock System, Estimated Effort = 135.06 x 28
Estimated Effort = 3782 Hours

7. Plan of Work / Project Management

Group 1: (Ted Moseley, Michael Truong, Daniel Nguyen)

- Authentication subsystem for temporary access to the home for Visitors
  - Generate temporary URLs that Homeowner send to Visitors
  - Produce a web Front-end for generating QR codes that Visitors scan
  - Produce a web Front-end for Homeowner to get temporary URLs for Visitors
- Authentication Configuration (for temporary access URLs/QR codes)
  - Allows the Homeowner to configure a time-frame, number of uses, and/or expiration date.
- Smart Package Tracker and Barcode Authentication (*stretch goal*)
  - Allow Homeowner to enter data about incoming packages
  - Determine when a Courier may enter with a package based on tracking data
  - Produce a method for Couriers to scan into the home

Plan of work:
Group 1's initial goal is to decide upon the differentiation between visitors and package couriers when it comes to temporary access. We are going to use multi-factor authentication for both. Then, we will deal with the technical details of transmitting/generating data for these parties.

We finished designing set of user friendly and simple interfaces to display visitor/courier data for the upcoming demo. We will then implement configuration options that a homeowner would like to have access to when adding visitors and package details.

Group 2A: (Eric Lin, Mohit Khattar)

- Database Structuring (Shared Infrastructure) and communicating structure to other subteams.
- Assist group 2B with their responsibilities.

Plan of work:
The primary goal of Group 2A is to design the database for the whole application. This involves setting up a RESTful API that will allow other groups to access and modify the database in any ways necessary. This will be an ongoing task; it is not likely that all groups will know what they need (in terms of database design and API endpoints) in the early

stages of the project. Continuous work will be required on this front as a result, and the goal is to have the back-end API fully ready and functional by March 1.
Other than configuring the database and adding API endpoints, the group will be assisting Group 2B in any aspects of their goals that they need help with.

Group 2B: (Mohammad Nadeem, Andrew Sengupta)

- Data Visualization (be creative; using data on who and when [events log]; e.g. how many people visited this month? who visited this month?)
- Create backend and frontend for "events log" (a log of significant events (masked person detected, face detected, motion detected, facial recognized))
- Create SMS notification handler (handles any sort of notification (notify user of any event))
- Make all classes of notifications toggleable
- SRN (Stranger Reporting Network) Server API and connect it to the server
- Live feed communication between the server, embedded computer, and web client

Plan of work:
First, Group 2B has to brainstorm and decide on what will be in the data visualization (web app). We will create a mockup of the interface of the visualization. We will then create the web pages for data visualization using HTML, CSS, and JS.

Concurrently, we will be developing the database for Stranger Reporting Network and will use SQL to do so.. The SRN server will use Python and a Django framework. To visualize the Stranger Reporting Network, we will use the Google Maps API to display a heatmap of suspicious activity in the homeowner's video. Then, we will create a system for the SMS Notification network. The tools used will be Twilio which is an external API used for sending and receiving SMS messages used by many services. Twilio will allow us to program the SMS notification network to allow responses to user messages as well. Twilio uses Flask so we will use Flask as a framework.

We will also work on the live video feed from the camera. This technology will be implemented using WebSockets and will stream to both the server and the web app.

Group 3: (Jasjit Janda, Jeffrey Lu, Amandip Kaler)

- Motion Detection
- Facial Detection
- Facial Recognition
- Masked person Detection (must implement own interface/unlock signal to lock)

- Create backend and frontend for Auth Permissions (add active members)
- Add detail Authentication Configuration (for people recognized by face) (e.g. you may only want certain people to gain access for a certain period of time, certain amount of days)

Plan of work:
We have decided to use computer vision library OpenCV to implement motion detection first. When test runs are successful, we will proceed to implement object detection.
Future: Implement facial recognition to the system.

8. References

- **CSS:** https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- **Django:** https://www.djangoproject.com/
- **Django REST Framework:** https://www.django-rest-framework.org/
- **Flask:** http://flask.pocoo.org/
- **HTML:** https://en.wikipedia.org/wiki/HTML
- **JavaScript:** https://en.wikipedia.org/wiki/JavaScript
- **Maps API:** https://developers.google.com/maps/documentation/javascript/
- **OpenCV:** https://opencv.org/
- **Python:** https://www.python.org/
- **PostgreSQL:** https://www.postgresql.org/
- **REST:** https://en.wikipedia.org/wiki/Representational_state_transfer
- **Twilio:** http://twilio.com/
- **WebSocket**: https://en.wikipedia.org/wiki/WebSocket