Project Deliverable F: Prototype Ⅰ & Customer Feedback

Team 7

October 29th, 2024

Abstract

*The document outlines a project deliverable focusing on prototype realization and customer feedback. It includes the complete documentation for prototype development which includes a user interface and administrative dashboard, test plan as well as the core system architecture for SIM swap detection and alerting. A Critical Component Analysis and gathered feedback are also included. Finally, notable changes to the Bill of Materials and project specifications are incorporated. The deliverable aims to complete a prototype, testing a design specification and plan for future prototypes.*

Table of Contents

[**Introduction 4**](#_uo1v4mstxso5)

[**Prototype Development 4**](#_u40nvl7t94qd)

[Documentation for Prototyping 4](#_1o7n56acrug0)

[User Interface and Administrative Dashboard 4](#_o7noycv418b)

[SIM Swap/Duplication Core Detection Algorithm 5](#_eq8ivvqsyneq)

[System Architecture for SIM Swap & Alerting 8](#_cmkkjo81sapt)

[Prototype Test Plan Outline 9](#_nhibdoaqy49)

[**Critical Components Analysis 12**](#_w7dye1cnrd6p)

[**Gathering Feedback 13**](#_53z2yd4k3ckb)

[**Updates 13**](#_43eikup6yldy)

[Target Specs 13](#_t3zy3gk1fnxi)

[Design 13](#_8vb6c5mftrn)

[Bills of Materials 13](#_bnzax5qbafvw)

[**References 13**](#_7ycqltdohcf8)

# 

# 

# Introduction

The objective of this deliverable is to develop the first prototype, devise a test plan for the second, and gather customer feedback on the initial design to inform improvements in future iterations.

# Prototype Development

In the prototype documentation, all of the team members in charge of a certain subsystem this week both wrote their code and included comments throughout it. All team members updated their pseudocode/code to the shared AANNG GitHub file. Below is a written summary of what each team member did, including a description/snippet of their code and an explanation of their testing methodology.

## Documentation for Prototyping

## User Interface and Administrative Dashboard

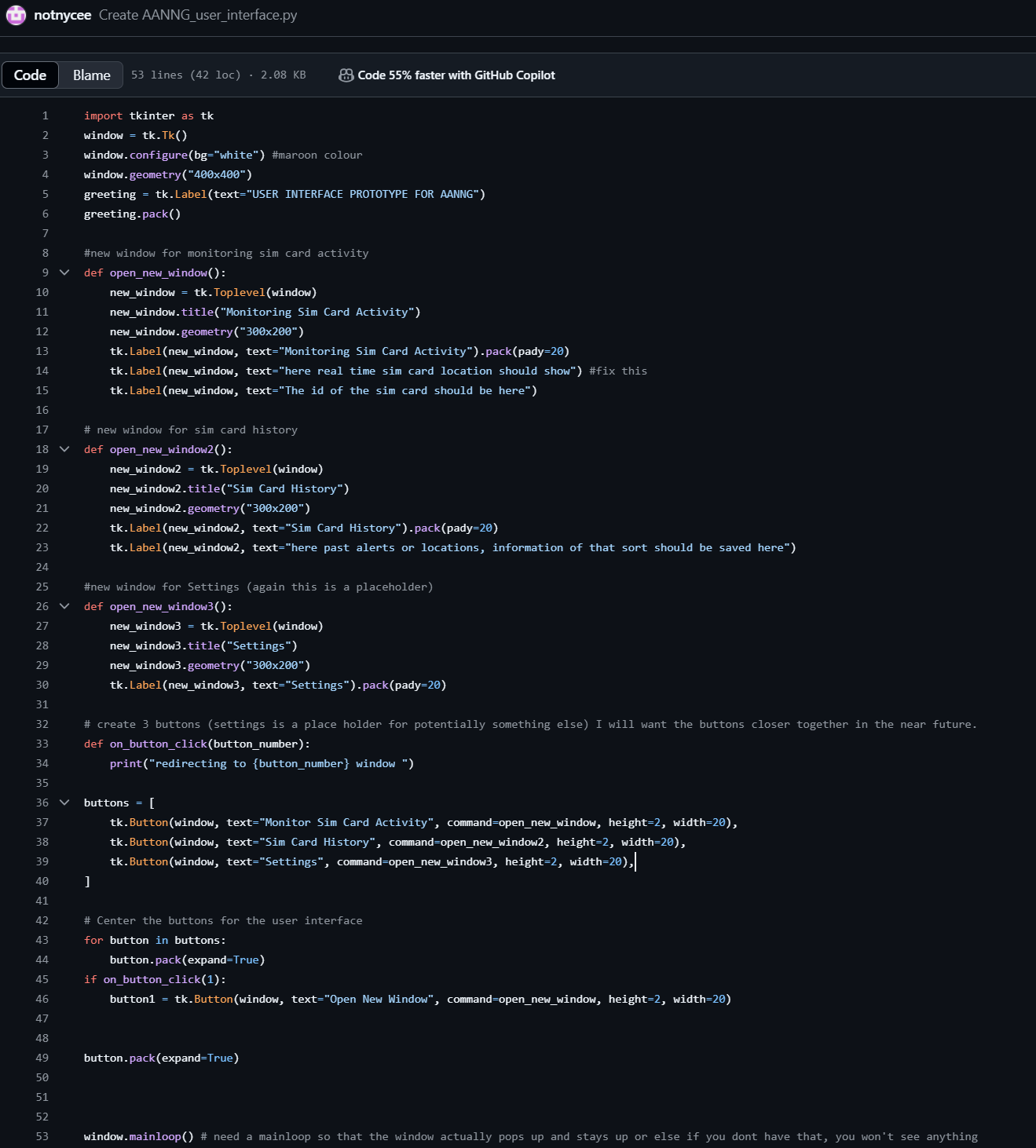
The intention was to create a simple user interface. Shabodi had mentioned that they wanted minimal UI so that the product focuses on functionality and is easy to maintain.

Using Python’s tkinter, the starting screen you would see as you start up the program was coded.

This is the start of the development. Later on, code needs to be developed for the code to connect with the interface and show an alert message. Another thing that is wanted is to be able to display the location, sim card ID, and current condition of a sim card in the ‘monitoring sim card activity’ window.



Current user interface code :



## SIM Swap/Duplication Core Detection Algorithm

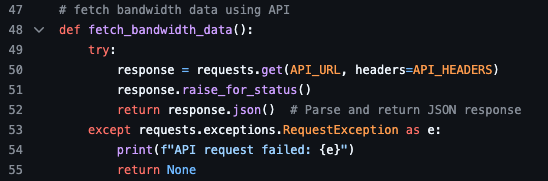
When working on the SIM Swap/Duplication Core Detection Algorithm, I made sure to create an outline of the pseudocode for the prototype based on what the subsystem is intended to do in terms of functionality as gathered from client feedback and criteria.

## 

The prototyping test plan for this subsystem is essentially the approach of programming basic functionalities (as per the design criteria and user feedback from Client Meeting #2) and testing it repeatedly to ensure functionality. I also identified the stopping criterion for the core system. This subsystem is especially important since it influences the other subsystems as well and some of their functionalities (such as for the Location subsystem) depend on this detection algorithm.

I also identified the core stopping criterion for this subsystem, which is the successful parsing of the JSON file and identification of a swap/duplication within the standard 25 Mbps run time on a 5G private (enterprise) network. When this is successfully done, the detection algorithm can be considered to “pass” its test. Any testing that I have done so far has been focused, since this is a focused subsystem prototype.

In terms of the code that I did, I made sure to include comments throughout so that, for future prototype testing or potential debugging, it would be clear what each code block was intended to do. Below is a screenshot of an example of such a block as part of the code:



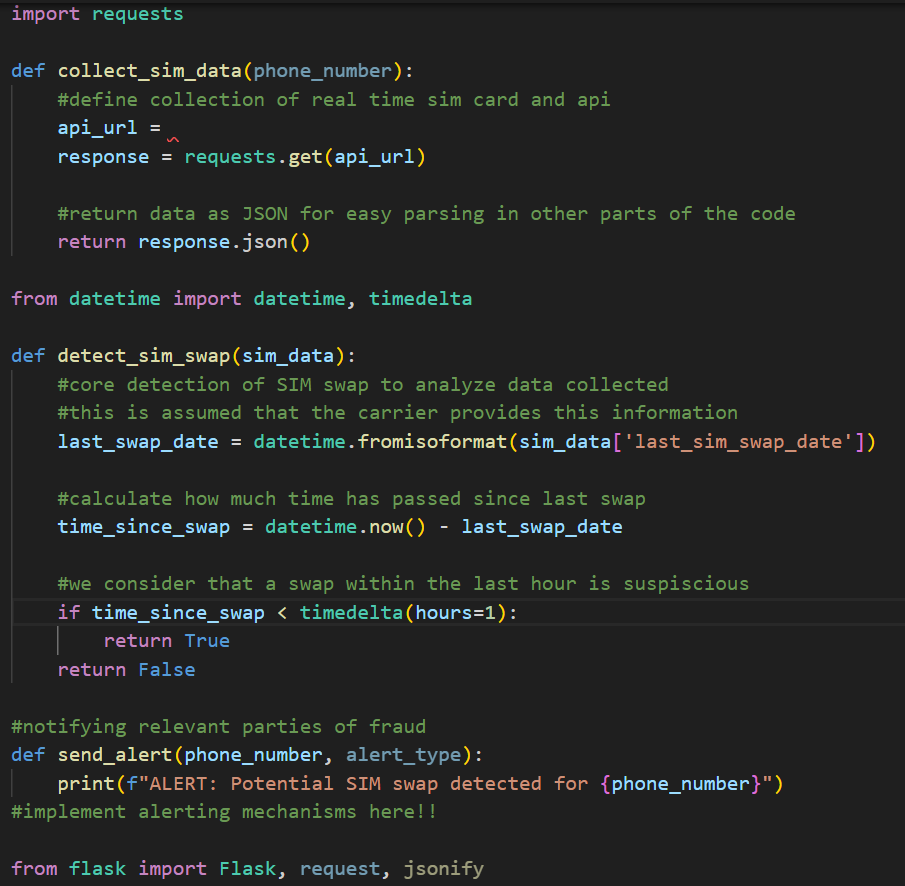
As per recommended industry standards, I have avoided using specific line comments where possible. This is also to promote consistent neatness in the code. Below is a copy of the code in the SIM\_swap\_legitimacy.py Python file on the GitHub repository.

|  |
| --- |
| # Detection of SIM Swap Fraud or Duplication  # Blame: Nava  """  1. Import JSON library and reader to read files from the network provider on registered SIMs  2. Before parsing, check if user has been authenticated prior to accessing  3. Measure current bandwidth vs capacity (Bandwidth API) (once an hour, looping function)  4. Parse JSON file and read data: ICCID and Location of ICCID (use location API) (once an hour)  5. Function to detect any duplicated ICCID OR a location change of a registered SIM  """  #loop function to continuously read the data  # SIM cards currently registered in this enterprise's bandwidth  # write a function to obtain the SIM of the device from settings OR just have a list and parse the b2andwidth data of registered SIMs in the data  # import the JSON file, parse the JSON file loop once every hour  # Pseudocode \*INTERNAL\*  # Assumption: All devices under bandwidth enterprise network have registered SIM IDs (ICCIDs) under a JSON file  import json  import numpy as np  import time  import requests  from datetime import datetime  # initialize JSON file containing the registed SIMs (assume provided by network)  json\_file\_path = 'registered\_sims.json'  # put data into a NumPy array  def load\_iccids(file\_path):  try:  with open(file\_path, 'r') as json\_file:  registered\_iccids = json.load(json\_file)  return np.array(registered\_iccids)  except Exception as e:  print(f"Error loading ICCIDs: {e}")  return np.array([])  # configure Shabodi's Bandwidth API  API\_URL = "https://api.shabodi.com/bandwidth"  API\_HEADERS = {  "Authorization": "Bearer YOUR\_API\_TOKEN", # API TOKEN  "Content-Type": "application/json"  }  # fetch bandwidth data using API  def fetch\_bandwidth\_data():  try:  response = requests.get(API\_URL, headers=API\_HEADERS)  response.raise\_for\_status()  return response.json() # Parse and return JSON response  except requests.exceptions.RequestException as e:  print(f"API request failed: {e}")  return None  # log bandwidth data  def log\_bandwidth\_data(data):  data['timestamp'] = datetime.now().isoformat() # include a timestamp  with open('bandwidth\_log.json', 'a') as log\_file:  log\_file.write(json.dumps(data) + "\n")  # detect duplicated or swapped sims  def detection(current\_iccid\_data, registered\_iccids):  detected\_issues = []  for entry in current\_iccid\_data:  # get SIM number and location of SIM  iccid = entry.get('iccid')  location = entry.get('location')    if iccid not in registered\_iccids:  detected\_issues.append(f"Unregistered SIM detected in network: {iccid}, {iccid}: {location}")  else:  registered\_location = registered\_iccids.get(iccid)  if location != registered\_location:  detected\_issues.append(f"Location change detected for SIM {iccid}: {location}")  if detected\_issues:  print("Detected SIM issues:", detected\_issues)  # loop to perform these checks once an hour  while True:  # update registed ICCID information from network  registered\_iccids = load\_iccids(json\_file\_path)  print("Registered ICCIDs:", registered\_iccids)  # fetch current bandwidth data  bandwidth\_data = fetch\_bandwidth\_data()  if bandwidth\_data:  log\_bandwidth\_data(bandwidth\_data)  print("Logged bandwidth data:", bandwidth\_data)  # check for suspicious activity  if 'sims' in bandwidth\_data: # Assuming 'sims' key holds the current SIM data in API response  detection(bandwidth\_data['sims'], registered\_iccids)  time.sleep(3600) |

## System Architecture for SIM Swap & Alerting

The intention was to design system architecture for SIM swap detection and alerting. This outlined the functions we needed to use to create or code, and what kind of data we needed to include. The api sections of the code are empty due to being unable to access Shabodi’s api’s.

I imported datetime to be able to calculate the time between when the user was alerted, and when the fraud/swapping took place. I also implemented a repeat of the code every hour, as requested, to ensure safety.



## Prototype Test Plan Outline

As part of the prototyping test plan, our team has outlined a few key factors that we would like to keep in mind for current and future analysis as we continue the development phase of this project. Among the factors we would like to consider, the objectives of our prototype planning is based on the feedback we received directly from Shabodi in Client Meeting #2. We would like to carry out both focused and comprehensive prototypes based on the nature of our objectives.

It is worth noting that, as per the approval of Shabodi, our team has begun to use GitHub as a mechanism for collaboratively working on code and being able to share files related to the project with one another. We have created a private repository to do this.

| **Prototyping Objective** | **Prototype Form** | **Testing Procedure** |
| --- | --- | --- |
| SIM Swap/Duplication algorithm | Focused | * The SIM Swap/Duplication algorithm is a focal point for the comprehensive system in terms of both the prototyping objective and the application as a whole. * We plan to test the SIM Swap/Duplicationalgorithm **after the addition of every major functional code block**. Frequent testing will allow us to simultaneously monitor potential syntax/logical errors as well as errors in functionality. * The Python libraries used will be datetime, json, numpy, time, and requests.      * We will be checking the algorithm to make sure we consistently meet the following standards set by the client:   + Standard encryption is successful for the JSON file of the ICCIDs in the network.   + APIs are properly integrated and function as intended.     - We will use the client’s overall approval of the level of functionality as a stopping criterion.   + The time complexity of the code remains constant.     - O(n) time will be the worst-case scenario, where ‘n’ is the length of the entire string in the given prototype.     - We will simulate a fake JSON file containing a high volume of SIMs to mimic reality.   + The parsing function to read and analyze SIM/ICCID data loops **once an hour** at a minimum network speed of 25 Mbps. |
| Location subsystem prototype | Focused | * The first prototype for the location subsystem should be able to identify an ICCID from the JSON file and return its location (using the JSON library in Python and Shabodi’s Location API). * Once we are able to confirm that the Location subsystem functions as intended with the overall core algorithm, we will subject it to user testing.   + In the first iteration of testing, we will have potential users test the subsystem.   + In later versions, due to time constraints, we will have Shabodi provide their feedback on its operability. * The stopping criterion will be when the location bandwidth successfully and accurately outputs the current location of the ICCID and is able to track it in real-time (if it moves) and share the location radius in the GUI user prompt. * The location radius returned should be **within 15km of the actual ICCID**.   + This also simplifies the algorithm since the actual coordinate location is more difficult to obtain. |
| Bandwidth analysis subsystem prototype | Focused | * The Bandwidth subsystem’s first prototype should be able to extract network (**5G**-oriented) data about the current internet speed and maximum storage. Other relevant data should also be able to be extracted, even if it will not be output (for the purpose of the overall algorithm’s functionality). * The Shabodi Bandwidth API will be used for this subsystem. * Once we have tested that the subsystem is able to (internally) output the relevant bandwidth data, we will ensure that it is functionally compatible with the other subsystems, including the core detection algorithm (this was also mentioned in Deliverable E). * In the first iteration of testing and feedback, we will be testing using potential users.   + In later iterations (primarily the 3rd/4th), we will ask for Shabodi’s feedback, since these prototype versions will likely most closely resemble the final subsystem prototype. * The stopping criterion for this will be when the bandwidth data is able to be consistently and successfully output, even with fluctuations in the data (which, as a reminder, will be tested through emulation).   + As well, once Shabodi is satisfied, we can make minor final adjustments, but essentially have the final product. * The bandwidth data should be able to return and simulate an actual enterprise's data. The goal for enterprise speed should be **25 Mbps.** |
| GUI and user alert/administrative control panel interface prototype | Focused | * The GUI and user alert/administrative control panel subsystem should be able to act as the sole UI/UX interface for this application.   + In accordance with Shabodi’s user needs and the design criteria, the GUI should be very minimalistic in terms of function.   + The application is not meant to be focused on user interactions with the machine; rather, it should be focused on machine-machine interaction with minimal user involvement.     - This was exemplified in the user feedback that we received from Shabodi in Client Meeting #2. * In the first iteration of prototype testing, we will be testing to see if the GUI functions properly in terms of intaking user input and providing the relevant output. The GUI will be using tkinter.   + We will also be testing for user feedback to determine if the functioning of the GUI is straightforward and simple to use.   + We will use different users every time to see if, upon first seeing the GUI, the user can figure out how to use it immediately. This will objectively tell us how straightforward it is to use the system. If we used the same user, they would already know which button(s) to press in order to get the data that they want. * The prototype should be able to alert users of SIM suspicious activity and administratively control it (accept/deny its acceptance into the bandwidth). * The prototype should also be able to provide data on the bandwidth/location/etc. upon request based on what was extracted during the last scan.   + The data should be **from the most recent hourly loop**. * In later iterations, the visual aesthetic of the interface will also be changed (for marketing purposes) to match Shabodi’s colours (using the hexcode of their logo colours The first prototype will be focused on having functional buttons which opens a window to display real time information. * We plan to test the first prototype by how long it takes for windows to open after the buttons are clicked. * Whether the sim card history updates properly. * If current information is seen to update every hour in the GUI. |
| SIM Swap/Duplication alert application | Comprehensive | * The first iteration of the comprehensive system test will be carried out once all subsystems have some level of functionality and can carry out **at least** their most primitive functions (as they have been outlined in Deliverable E). * Testing will focus on the more broad, overall goals of the system.   + We will test to make sure the program can function (ie. detect SIM swap and allow for administrative control of it) for its fundamental purpose based on the design criteria and user needs.   + The stopping criterion will depend on the level of development for the subsystems for each iteration of testing. It may be more broad in the beginning and then get more specific in later assessments (as well as when more client feedback comes in). * Iterations will continue for future variations as the development of the subsystems progresses.   + We will carry out a prototyping test of the comprehensive system **at least 4 times prior to presenting the final product**. This does not include additional tests/adjustments made for/by Shabodi. * The initial prototype should be finished within the next week. The iterations of the second/third/etc. should be what we work on for the next two/three weeks. |

# 

# Critical Components Analysis

A simple analysis of critical components or systems should also be included, based on

your current knowledge of engineering science or other knowledge.

| **Critical Components / Systems** | **Analysis** |
| --- | --- |
| Authentication Mechanism | * **Strength**: provides secure access control * **Potential Vulnerabilities:** token theft or interception * **Improvement**: create an implementation of token refresh mechanism that will securely store tokens |
| SIM Swap Detection API | * **Strength**: Direct communication with network for real-time detecting * **Potential Vulnerabilities:** we are dependent on a third-party * **Improvement**: implementation of a “fallback” mechanism to check the health of the API |
| Phone Number Validation | * **Strength**: prevents invalid inputs of numbers * **Potential Vulnerabilities:** could be a possible way to bypass this mechanism * **Improvement**: consider international number formats to increase unique patterns |
| Error Handling & Logging | * **Strength**: helps debug and increase our security within the code * **Potential Vulnerabilities:** possible information leakage * **Improvement**: create a structured logging to ensure the sensitive data is not logged and can’t be viewed by the public |
| User Interface & SIM Swap Alerts | * **Strength**: informs user of security threats * **Potential Vulnerabilities:** if we alert the user too often, it can cause alert fatigue * **Improvement**: make sure the alerts are only active when there is a real threat |

# 

# 

# Gathering Feedback

| **Feedback** | **Key Comments** | **Area of Improvement** | **Priority Level** |
| --- | --- | --- | --- |
| The overall idea of the project and the code prototype. | * “Overall, I was very impressed.” | “What if someone disconnects a sim card, duplicates it, and then reconnects the duplicated sim card to the network? Would the network detect that duplicated sim card”  They wondered how we plan to get around that. | MEDIUM |
| Client meeting #2  With Shabodi  APIs | * Very content * Jitter and QoS don’t seem to have a purpose for the client and the project | Look into insights API | HIGH |
| Client meeting #2 with Shabodi  PRODUCT | * Shabodi wondered whether we were building an application or a component to a larger system. | Define the product.  There are two ways to go about it.  They want us to think of real-world business applications of the product. | MEDIUM |

# Updates

The relevant updates that were made this week are listed as follows:

## Target Specifications

After the initial prototype of the SIM Swap/Duplication core detection algorithm, the stopping criteria has been adjusted to be not only the ability to detect a swap/duplication, but to be able to successfully interoperate between the other subsystems (ie. be able to detect the location and bandwidth data given the extracted information).

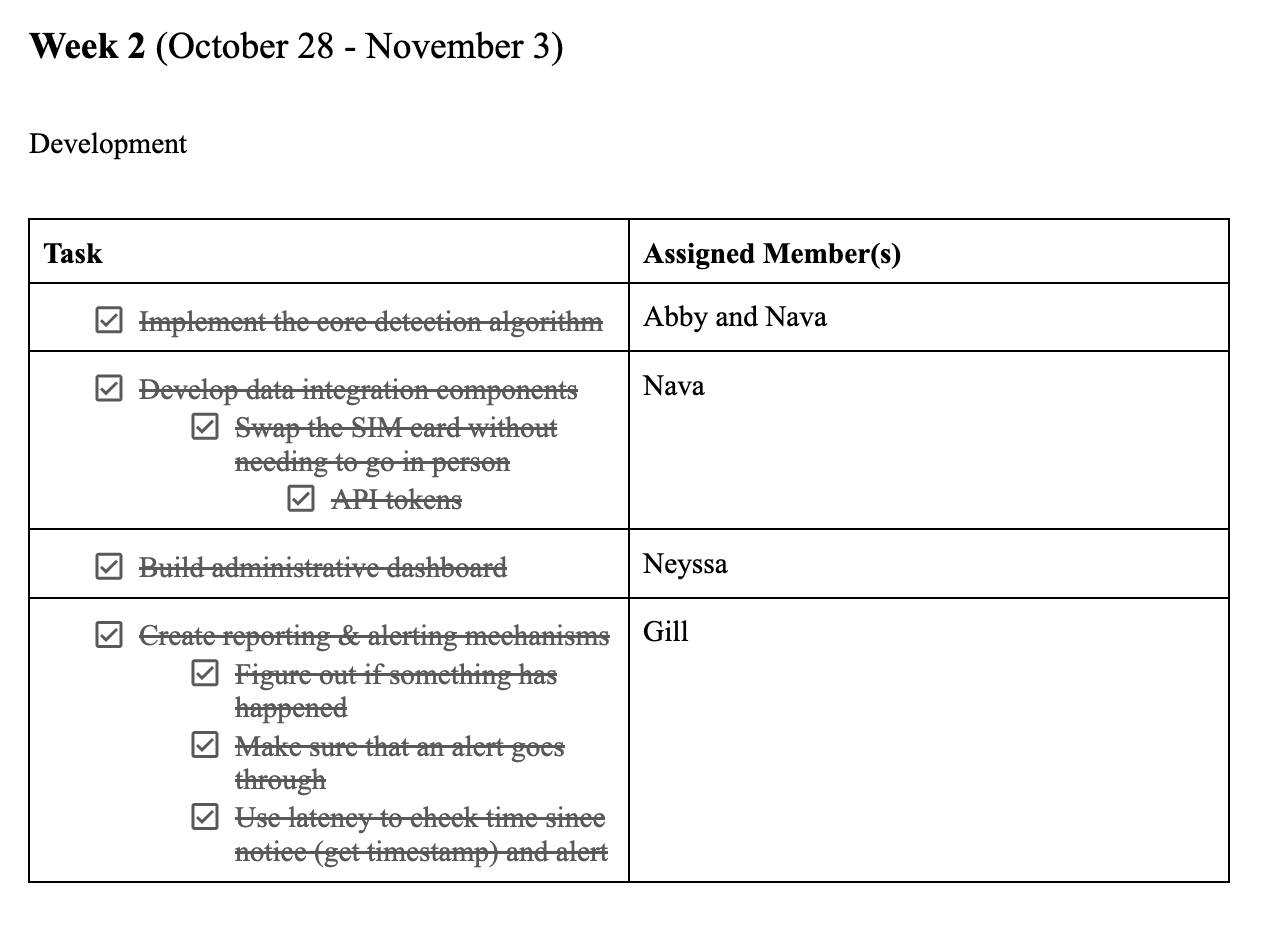
Bills of Materials

In the Bill of Materials we updated the basic user devices needed to start our project which included adding to the list the Shabodi equipment, individual laptops and the Microsoft OS/Mac iOS. As well, we included the use of certain Python libraries/tools, including **tkinter, requests, datetime, json, numpy, and time.**

# Trello Board (Task Distribution)

# 

This week, our team listed our tasks in checklists under the Deliverable F card in our Trello. This was done in accordance with our plan on the Prototype Calendar from Deliverable E.



The estimated duration of all tasks this week was **5-6 days** for code-related tasks and **2-3 days** for documentation/written tasks. All members completed their tasks within this time constraint with no issues.