# Bilkent University

Department of Computer Engineering

# Senior Design Project

## Detailed Design Report

*VANNY*


**Project Members**

*A A M Jubaeid Hasan Chowdhury - 21503356*

*Abdul Razak DaherKhatib - 21801340*

*Esad Ismail Tök - 21801679*

*Elifnur Alsaç - 21601098*

*Mohamed Wasim Mohamed Akram - 21801103*

**Supervisor**

*Can Alkan*

**Jury Members**

*Erhan Dolak*

*Tağmaç Topal*

**Innovation Expert**

*Faik Berk Güler*

# Table of Contents

# 1. Introduction

This report is the High-Level Design report of the Vanny project. In this report, the aim and design goals of the project will be mentioned. Afterward, the proposed software architecture will be mentioned. In this section, the subjects of the project such as subsystem decomposition, hardware/software mapping, security and control issues, and boundary conditions will be discussed. Following this, subsystem services and its layers will be explained. Afterward, the test cases of the project will be included. After various engineering design considerations, the dynamics and roles of teamwork will be discussed under the title of Teamwork Details.

Babies and toddlers are the main focus, care, and concern of all parents. From providing the best food and medical care to paying babysitters to set up a nanny cam, parents will stop at nothing to make sure their babies and toddlers are healthy and safe. Nanny cams specifically play an increasingly important role in helping parents extend their love and care beyond their immediate physical vicinity, allowing the parents to have a reliable and transparent tool to watch their loved babies and toddlers. The commonly available nanny cams, however, can be optimized to push this reliability and transparency further if the newest and most capable technological tools were integrated in said systems. Furthermore, by making these systems more intelligent and able to communicate with the parents the current nanny cam systems can be extended and transformed into an additional caring eye that is always present in the babies' and toddlers' bedrooms or playrooms.

After conducting market research we observed that most of the current nanny cam systems don't utilize some very useful Artificial Intelligence and Machine Learning algorithms efficiently. Alfred Home Security Camera, for example, is the most popular similar system on Apple's AppStore with over 40 million users, detects intruders by detecting "any movement", with no other AI or ML capabilities utilized. It can also live stream, see in low-light environments, and act as a walkie-talkie, among other common features in security cams[1].

Another example "Annie Baby Monitor" acts as a basic nanny cam with motion detection and the ability to play lullabies[2]. Similarly, other examples of such systems have motion detection with other AI capabilities like detecting noise and crying and notifying the parent that the toddler is awake[3][4][5]. The most capable similar system is the "Mi Home Security Camera". The aforementioned system is a general security camera system. Moreover, it has an angle of 130 degrees, has infrared and night vision scopes, and acts as a two-way voice communication system. In terms of AI capabilities, it has a zoning functionality that makes the system more sensitive to movements in these zones and can detect family members in the camera's range, using Mi Bands or iPhones, and turn them off when
they are in the range[6].

Finally, other specialized systems that focus strictly on a specific problem exist. "Angel Eyes", as an example, watches the baby while sleeping in the crib. The system uses a camera to watch the baby and detects any issues, like objects in the crib, or if the baby is in an unsafe position, and uses sensors to monitor humidity and temperature in the room[7]. Another system detects the hunger of the baby using some cues[8].

A more optimized system will act as a general purpose more capable nanny cam that fits somewhere between “Mi Home Security Camera” and “Angel Eyes”, and is more capable than the other aforementioned nanny cams by utilizing the capabilities of AI and ML and focus on the toddler in their rooms. Ideally, this system will transform nanny cams by pushing them to the limits of possible capabilities. This document describes the specifications of a system that optimizes and transforms these systems into a virtual babysitter. Note that the age group of children that are the target of the system is 0 to 5 years old, the document will refer to this age group as toddlers.

## 1.1 Purpose of the System

In this project, the purpose of the system can be defined under technical and non-technical titles.
For the non-technical purpose of the system, parents can monitor their children when they are not at home so they don't have to worry about their safety. Designed by utilizing the competencies of current technologies, this product will provide comfort for parents and safety for children. At the same time, concentrating the features of other products on the market in a single product provides convenience for users.

Sustainable, efficient, reliable, and error-free operation of the project can be listed for the technical purpose of the system. The safe use of users' data is a technical objective of the project. At the same time, it is important for the purpose of the system that the users can use the product comfortably.

## 1.2 Design Goals

### 1.2.1 Usability

- The system will require a one-time setup afterward users can use the system without directly interacting with the server side.
- The UI of the phone application will be designed so it is used by any user regardless of their tech knowledge.

### 1.2.2 Efficiency

- The system will be able to watch a frame-per-second rate of at least 10.

● The system detects any issues and reports in real-time therefore the notifications sent to the user should arrive within 5 seconds.
● The system will delete full-day recordings in 3 days and highlights in 7 days, enabling efficient memory management.

### 1.2.3 Ethicality

● Parents will be made aware of what the system records and for how long.
● No third-party activity of any kind will have access to the data stored.
● Reports generated will only be shared with parents.
● Encryption and predefined passwords will be implemented to prevent any undesired access and undermine toddlers' safety or privacy.

### 1.2.4 Privacy

● User data will be protected by encryption between the server and the client in the system.
● Recordings will be private with access from any third party or individuals.
● Users can disable the recording whenever desired.
● Users can delete any form of data saved.

## 1.3 Definitions, Acronyms, and Abbreviations

UI: User interface. Anything a user may interact with to use a digital product or service. This includes everything from screens and touchscreens to keyboards, sounds, and even lights.

AI: Artificial intelligence refers to using computers to do things that traditionally require human intelligence.
ML: Machine learning refers to a subfield of artificial intelligence, which is defined as the capability of a machine to imitate intelligent human behavior.

IoT: The Internet of Things, refers to the collective network of connected devices and the technology that facilitates communication between devices and the cloud, as well as between the devices themselves.

## 1.4 Overview

VANNY (Virtual nANNY) is a Machine Learning, AI, and IoT software used as a virtual babysitter. The system will be a wide-range camera, with night vision, fixed in the corner of the toddler's room, along with a microphone and a speaker, the system will connect to a phone application to communicate with the parents. The system will watch a toddler 24/7 in their rooms in order to assure the parents that their toddler is safe and sound. The camera will monitor the toddlers and their surroundings, detecting any possible danger, including, but not limited to, fire and smoke, sharp objects, foreign moving objects, other toddlers, strangers, and loud noises. Additionally, the system will watch the toddlers themselves in case they fall or cry. Once crying is detected the speakers can be used to play a lullaby that is pre-recorded or fetched from the internet. Finally, the system will analyze the toddlers' daytime movement patterns, amount, and patterns; the toddlers' sleep depth, and durations; and the toddlers' crying loudness, duration, and times. Parents will use the mobile application to be notified in case any danger was detected, or if the toddler hurt herself or cries for a certain time or loudly.

The mobile phone will act as a messenger between the system and the parents. Through it, parents can watch their toddlers live, and receive notifications and reports after the analysis. The analysis' aim is, solely, to help observe any unusual patterns in the toddlers' movement, sleep, or crying, not to provide medical advice. For example, an unusually low amount of movement or a long crying at night can provide a hint to the parent's regarding the toddlers' health or uncomforts, i.e growing teeth, bad room temperature, etc.

When all of these components are put together the system will act as a virtual babysitter that helps parents extend their care beyond their physical vicinity, and be at ease when they are not around. Moreover, the system will be completely local, making it more secure and avoiding privacy fears that can turn the system into a source of concern instead of comfort. The innovation type of this system would be "Product Performance" since it focuses on the value, features, and quality of the service, and it attempts to outperform competitors. One important issue regarding this type of innovation is that it is usually easy to copy, and perhaps, be outperformed by other, larger competitors. Additionally, since the needs and features are pre-existing and already realized by the customers but are also adaptive and will continuously improve our innovation is a mix of sustaining and incremental innovation.

# 2. Current Software Architecture

As mentioned earlier, there are multiple systems with similar features and purposes as Vanny. However, together with Vanny, the shortcomings of the mentioned systems will be compensated and the idea will be further improved and perfectionized with the additional features that will be added to the system.
Current Systems:

- Alfred Home Security Camera
- Annie Baby Monitor
- Mi Home Security Camera
- Angel Eyes

# 3. Proposed Software Architecture

## 3.1 Overview

Considering how sophisticated the system would be, we constructed quite sophisticated class models in accordance with the OO principles. The system will consist of three packages that will be explained in the following sections. These packages were separated based on the task they perform and they communicate with each other through specific classes. These packages will represent subsystems in Vanny.

The mobile device will serve as a conduit between the parental unit and the system. Through it, parents can observe their infants in real-time and receive notifications and analysis results. The mobile application will be used to stream the surveillance as well as store critical information captured by the camera.

Tests will be performed to assert the proper implementation of the system and the correctness and completeness of specifications.

## 3.2 Subsystem Decomposition

In the below class diagrams the system's architecture is shown. The system consists of the major packages, these three packages represent the main components of the system based on functionality and hardware. These three packages are also subsystems in our system. In a way, Vanny represents a standard client-server architecture, as will be shown below, with the Raspberry Pi being the server, and the Android phone being the client.

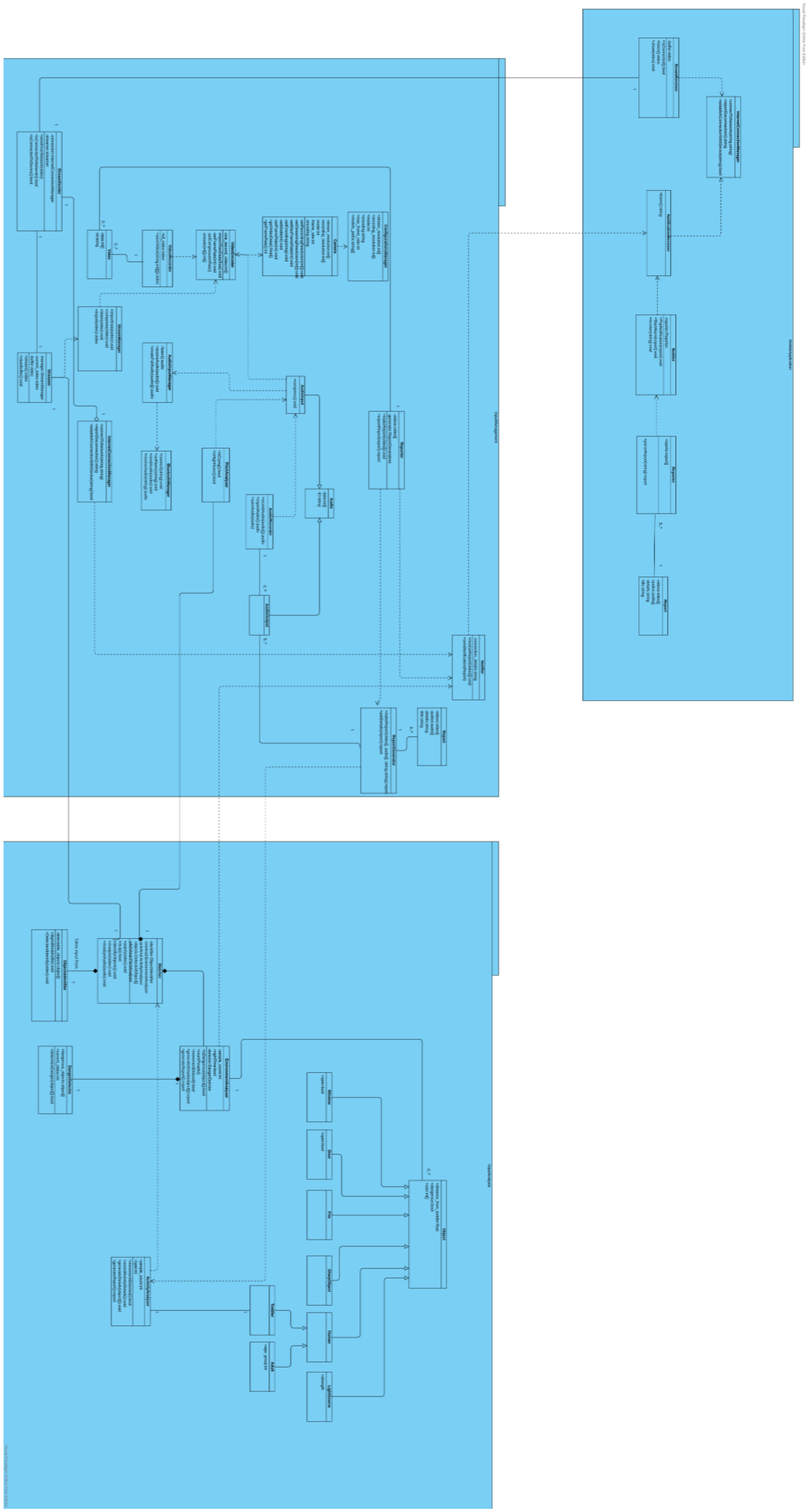

Figure 1: UML Class diagram.

## Input Management Subsystem:

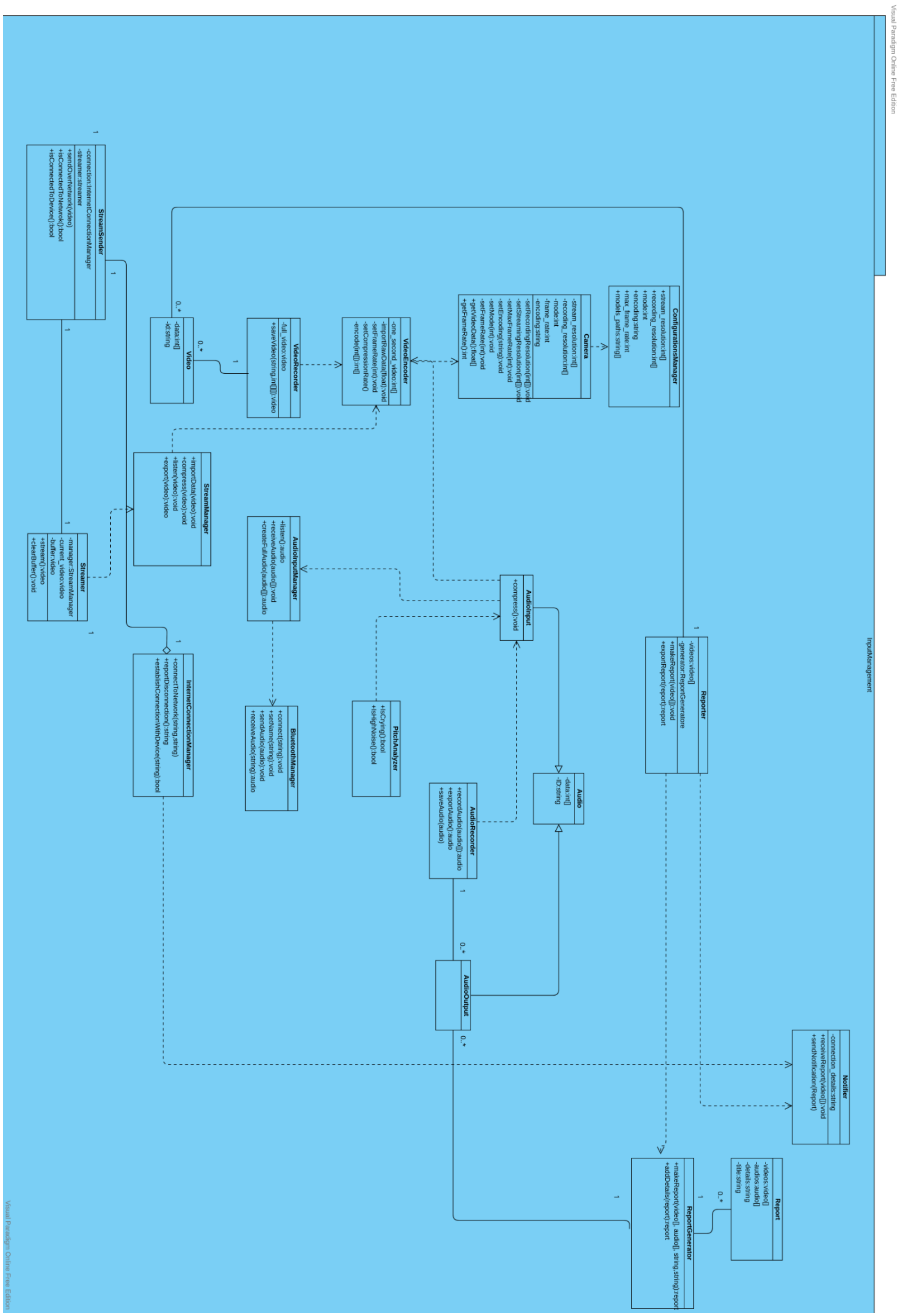


Figure 2: “Input Management” package.

The first sub-system is the Input Manager subsystem is the core subsystem in the system where the majority of the analysis and decision-making will be handled. In this subsystem, we have the Camera as the main source of input and provider of the data to be analyzed and the images to be to detect objects. The camera will use the configuration manager at the beginning of the launch of the system, these configurations include stream and recording resolution, the encoding of them, and frame rates, among other configurations. Another key component in this subsystem is the Stream Manager. The Stream Manager class will handle the process of streaming the live preview to the Android device. This component uses the streamer class to manage the internet connection and send the stream over the internet, as well as give live feedback on the detected dangers. The third main components in the system are the recorders, both for audio and video. In both of these components, the system will record, compress, encode, and store the recordings for later use. The fourth main component is the Reporter, the reporter component is responsible for generating the reports. This includes setting proper explanatory titles for reports, saving videos, and preparing weekly reports for activity analysis, sleep analysis, and for incidents recorder. Lastly, a small but critical part of this subsystem is the Notifier. The Notifier component is responsible for sending all types of notifications to the Android device, as well as sending the reports.

## Input Analysis Subsystem:

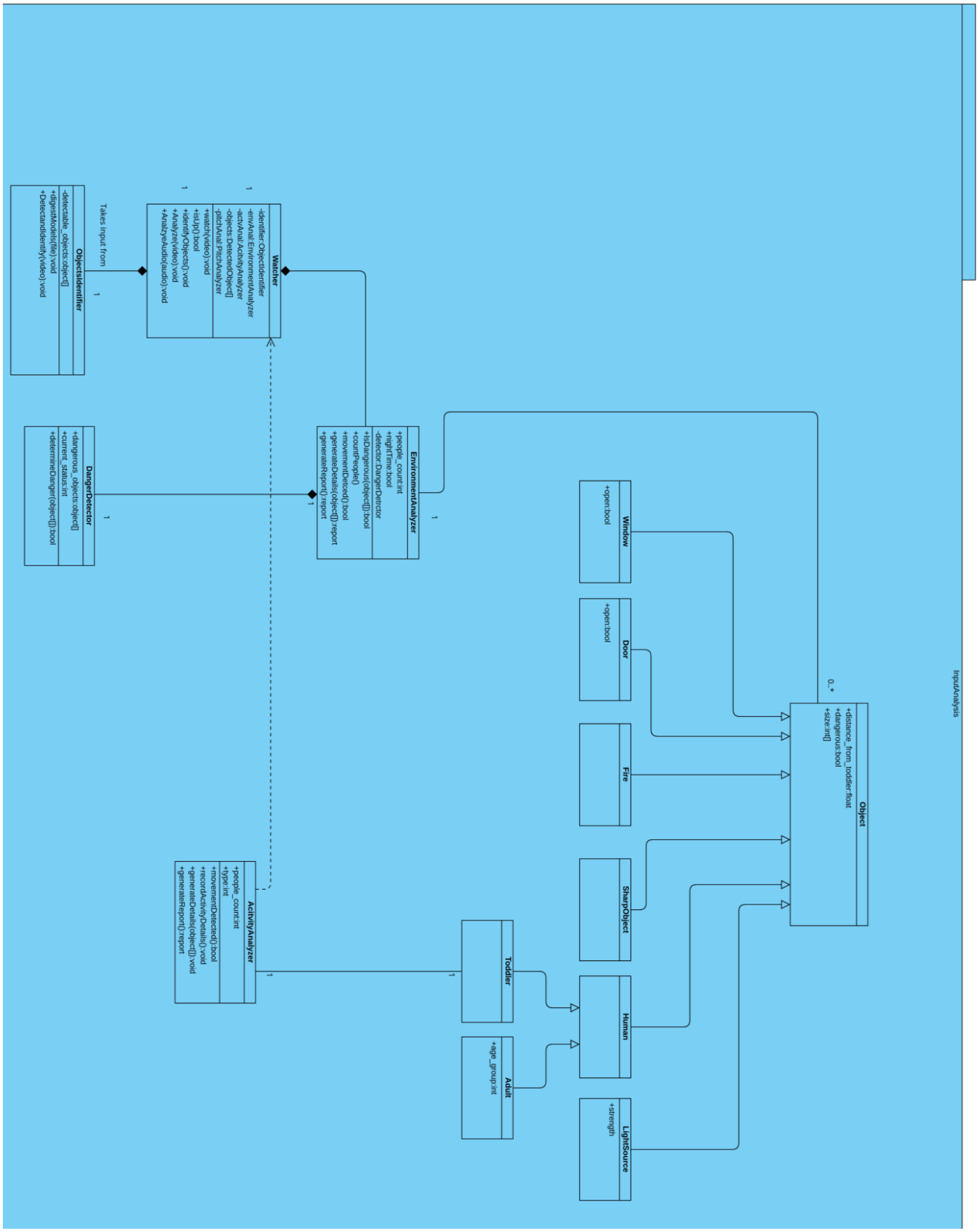


Figure 3: “Input Analysis” package.

The Second subsystem is the Input Analysis. While the Input Manager processes the input, the processing in that system is in the form of data management; no “Intelligent” decisions are made in that subsystem. In the Input Analysis, however, the ML and AI components are used to analyze the input obtained by the Input Manager subsystem.
The core of this subsystem is the Watcher components. This component uses the Object Identifier component, which is used for detecting all forms of objects in the area being watched, as well as digesting and loading the models that the system will detect, decisions will not be made here regarding whether there are any dangers around, or regarding the movement analysis, this component will solely identify the objects and notify, continuously the Watcher. Another component used by the Watcher is the Environment Analyzer. In this component, the major decisions are made regarding whether objects are dangerous, whether any movement is detected when there is supposed to be none, and keeping count of people in the area. It also keeps track of these elements for the report generator to collect and generate its reports. Activity Analyzer is one more component that the Watcher component uses. In this component, movement is recorded along with its details, like duration and time, and the report details are prepared here for the weekly reports that are generated by the reporter. Finally, with all of these components being used by the Watcher component, the Watcher component acts as a manager and an intermediary between all of these components and uses them to evaluate the situation in the toddler’s environment and command the analyzer components to run the analysis on the feedback provided by the objects identifier.

### Mobile Application Subsystem:

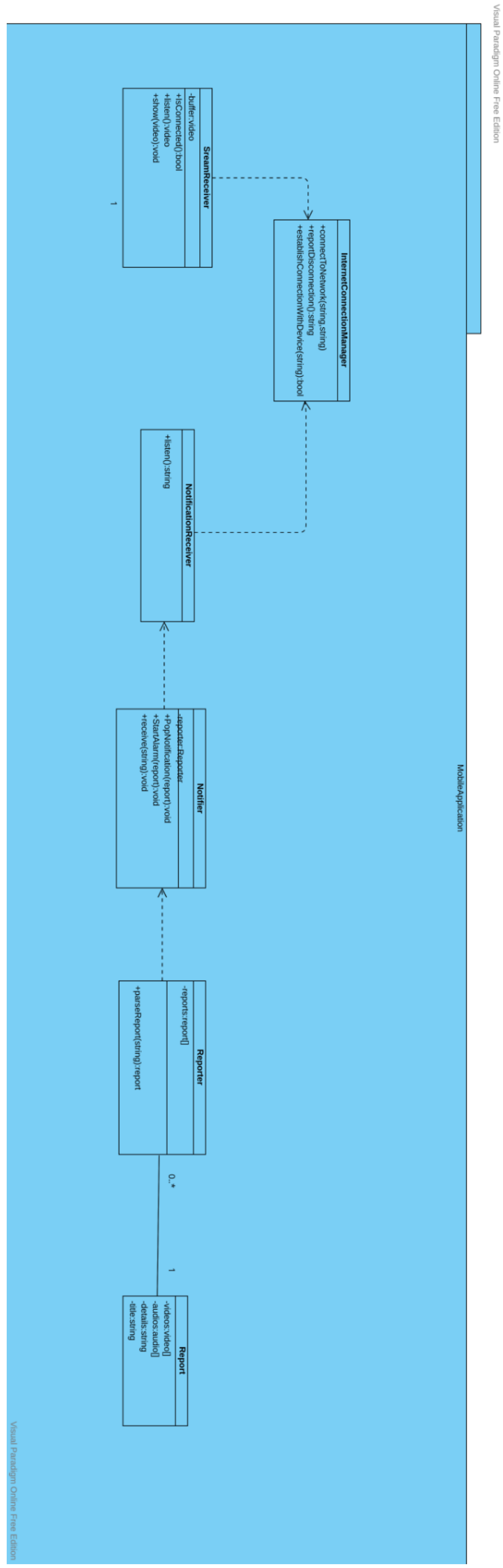


Figure 4: “Mobile Application” package.

The last subsystem is the Mobile Application subsystem. This subsystem is the simplest since the application will be used only for receiving the notifications, stream, and reports from the Input Manger's Notifier and Streamer components. The two key components of the subsystem are Stream Receiver and the Notifications Receiver components. Both of these components act as listeners for any events from the Input Manager subsystem.

## 3.3 Hardware/Software Mapping

Similar to the subsystem decomposition, Class Diagrams will be used to break down the Hardware/Software Mapping. The list of hardware is the following:

1. Raspberry PI 4, Model B, 8 GB of Ram. The following elements are extensions of this hardware element:
    a. Bluetooth speakers.
    b. Bluetooth microphone.
    c. Attached camera.
    d. Infrared sensor.
2. Android Mobile

The Input Manager and Input Analyzer subsystems are both run on the Raspberry Pi using the attached/connected devices. Some elements are mapped directly to their “twin” in the code, like the camera, while others are managed by the OS, like infrared sensors, or used in different components, like speakers. Audio recorders, i.e speakers, are used in both the pitch analyzer, and the video recorder. Speakers are mapped to the Media player's components. The Input Analyzer subsystem is also run on the Pi 4 device. The processing and the decision-making processes being performed in the Pi 4 device make the optimization of all processes, like video capturing, streaming, and storing; ML models and objects detection; and managing connections to other devices and notifying the Android device; of paramount importance considering the limited abilities of the device.

The Android mobile will be responsible for the Mobile Application subsystem. Standard Android mobiles come with capabilities like internet connections and media storage and a player. This means that the main responsibilities of the system like receiving notifications and reports, as well as playing video and audio recordings can be easily performed on such devices.
The load by the subsystem on the hardware component is pretty low and therefore the mobile element is of non-critical importance when it comes to performance, as long as the notifications are received instantly.

## 3.4 Persistent Data Management

Vanny is an application that is extremely dependent on the data that the system receives. Therefore persistent data management constitutes an important portion of the overall software architecture of the application.

There are two kinds of data that the system depends on. One is the video and the other is the audio. A camera device is connected to the Raspberry PI in the system that continually captures the video of the toddler. This is the data that our machine-learning model depends on. It is data that is crucial for both our application and the users of the application. So we need to be careful about both the storage requirements of the data and the potential misuse threats regarding the data. In that sense, our system does not require any SQL or NoSQL DBMS (Database Management System). Vanny is a system that cares about the data and user privacy therefore the captured video is directly processed and used by the machine learning model instead of storing it.

On the other hand, although we do not store the whole record, we store a portion of the record that we call “the highlight moment”. This portion of the record consists of 2 - 3 minutes and includes the recorded moments that will be included in the analysis report of the toddler and sent to the user of the application. Highlight moments are stored in the memory of the Raspberry PI and it is temporary storage in order not to overwhelm the Raspberry PI and provide better memory management.

The other data that the system depends on is the audio. It is the audio record that the user of the application (parent) sends to the toddler via the speaker that is connected to the Raspberry PI. Again we do not use any DBMS in terms of the storage of the audio because of privacy and better memory management. After the record is delivered and used by the speaker, we do not perform any storage regarding the data.

## 3.5 Access Control and Security

Vanny is a data-sensitive application which means that the protection of the data and the privacy of the users are extremely important. If a toddler is to be monitored, we cannot allow anyone else that is not authorized to monitor the toddler or receive the video and audio data. To ensure access control, we emphasize the importance of authentication.

The user of the program needs to set a strong password and should be authenticated before starting to use the application. Anyone else that is not authenticated is restricted from the system. Any other functionality, except for the registration and login screens, is out of use for the users that are not logged in by authentication. There is only one level of authorization and this level provides all of the functionalities to the authorized user who is intended to be a parent of the toddler. Therefore we do not provide a mechanism that gives some permissions for the users that have a particular authorization and some other permissions for the other types of authorizations.

To provide the best possible security for such a data-sensitive system and prevent any possible data leak, we decided not to store our data (video and audio) in an external database. Instead, we only store necessary data (highlight moments) temporarily in the Raspberry PI memory. Moreover, in our system, we do not use any identifier for the toddlers and parents such as their names, ages, and faces to avoid any problem in case of an external threat.

## 3.6 Boundary Conditions

Most of the time it is convenient to structurize the procedures to follow in case of some boundary conditions. In this section, those procedures, namely boundary conditions are specified. There are 5 boundary conditions in Vanny. These are Application Setup, Initialization, Termination, Hardware Failure, and Authentication Failure.

### 3.6.1 Application Setup

Vanny is an application that has several hardware components that need to be set up therefore there should be an application setup stage before directly using the application. The connection between the user and the Raspberry PI should be established in order to grant access to the camera and speaker that are connected to the Raspberry PI.

We provide an authentication key to the users of the application in order to pair the user with the system. At first, the user downloads the Vanny application and in the first run of the application, the user needs to register to the system with the authentication key provided.

Using the authentication key the user can generate his/her password and use this password together with his/her username to complete the registration.

### 3.6.2 Initialization

Vanny requires each user to authenticate to be able to use the system. This authentication is performed by the user with the username and password generated in the application setup procedure. With those credentials, the user needs to be logged in to the system each time he/she initializes the application and starts using it.

In case the integrity of the hardware system is broken or there is an authentication failure, corresponding procedures need to be done which are described in the following sections.

### 3.6.3 Termination

In order to terminate the application, the user can simply quit the application on his/her mobile device. When the application is terminated, all the processes and threads that are related to the running application are removed from the memory, and the resources that are allocated for the application and its processes are freed.

As a security measure, the recorded video and audio data are also removed from the system instead of being stored in an external database system. Also in case of a termination, the highlight moments that are temporarily stored are erased and removed from the application.

### 3.6.4 Hardware Failure

There are several hardware components that are connected to Raspberry PI in Vanny which are the camera, microphone, and speaker. In case of any integration problem of the connection between a hardware device and the Raspberry PI, the system cannot allow the user to keep using the application. In such a case, the connections should be repaired and the authentication step should be repeated to log in to the system again and start using it.

### 3.6.5 Authentication Failure

It is possible that the user of the system forgets his/her credentials. In such a case, as a security measurement, the Application Setup procedure should be repeated. So the user needs to establish a connection with the system using the authentication key that is provided to the user.

# 4. Test Cases

Considering the complexity of using multiple hardware components and several meticulous elements in the system, like the Machine Learning component, the camera capture, the data management, and the streaming service; testing is crucial not just to make sure the developed functionalities work as intended, but also to ensure to help in the development process. Testing elements like Fakes, Mocks, and Stubs, as well as manual testing, will play a major role in the process of testing and development of our system.
In order to accomplish that we are using the following concepts to test our components. Below each concept, a list of specific tests and the mechanism of testing can be found.

## Unit Tests

Unit tests test the functionality of each “unit” in the system, asserting that the unit does what it is expected to do. A “unit” can be defined in many ways depending on the system, conventions, and engineers. We will refer to each function in the system as a unit. Some unit tests are automated and others are manual, and therefore the tests in the first section below would be represented as test code in test suites in the project’s code, while the section afterward, would represent manual tests. Unit tests will follow the Arrange, Act and Assert (AAA) Pattern.

1. Test that the camera works. Mechanism: start it and capture a photo.
2. Test that the camera can capture videos. Mechanism: capture a short video of fewer than 3 seconds.
3. Test that the system can capture the last one minute before and after the incident. Mechanism: camera can be set to “preview” mode, and then capture 1 minute of content, introduce an incident, verify the length of the video by performing video size calculations, keeping in mind encoding, bitrate, resolution, and frame rate.

4. Test that the video captures are of specific size. Mechanism: define the bit-rate, resolution, frame size, length, and encoding, then calculating the expected size and then comparing it to the captured video.
5. Test the audio in the video captures are of a specific size. Mechanism: define the bit-rate, resolution, length, and encoding, then calculating the expected size and then comparing it to the captured video's audio.
6. Test that the image compressor compresses the images to the specified ratio chosen. Mechanism: feed the systema a pre-made image with specific size, compress and calculate against the ratios. The ratios can vary depending on the implementation. Realistically, 0.8-0.5 is the aim.
7. Test that the incident videos are being deleted after a specified time. Mechanism: save a video, set periodic deletion time to 10 seconds, sleep for 10 seconds, verify that the file is deleted.
8. Test that the de-fisheye works. Mechanism: save a raw image of the camera, compare the resolutions and sizes of the images before and after.
9. Test that the object detection functionality works. Mechanism: feed the system a video of objects that should be detected and then logs the events of detection, compare them to predefined expected logs.
10. Test that the accuracy of the detection is above 85%. Mechanism: feed videos of objects that models were trained to detect and then check the logs values.
11. Test that the system can create proper notifications based on the events detected, each object and event will have its own tests. Mechanism: introduce a predefined set of events and compare the logs to predefined log values.
12. Test that the system can connect to other nearby Bluetooth devices. Mechanism: run the bluetooth devices before the start of the testing suites, run the tests, assert that Pi 4 is connected to the specific Bluetooth devices.
13. Test that the system can disconnect from Bluetooth devices already connected. Mechanism: On success of the last test, disconnect from these devices and assert that the system is no longer connected to these devices.
14. Test that the system can connect to the internet. Mechanism: make sure WiFi connection is available before the test suite is run, run script to connect to the internet, ping Google and confirm a received response.
15. Test that the system can perform a WiFi connection. Mechanism: Set up Android device for connections, run script to connect the Pi 4 devices, send a message and receive a reply from Android device.
16. Test that the system can send a notification through the internet. Mechanism: upon success of previous two tests, use stub to create a notification, feed it to the notifier and send the notification, on the Android device feed the same stub, receive, assert that they are identical.
17. Test that the system can "pair" with another device through the internet. Mechanism: Upon success of internet connections of the devices, create sockets, connect them and then bind each device to the other, send multiple messages in both ways, verify reception of each message.

18. Test that the system can stream the camera's content over the internet. Mechanism: launch the Pi 4 device with the camera on, connect to the internet, pair the devices, capture video preview and in real-time, compress send over the internet, receive on the Android device, decompress, compare details. Stop after 1 minute, compare the total sent data v against the total received.
19. Test that the Android device can receive a notification from the internet. Mechanism: Connect the device to the internet, create a testing fake, connect it to it, or connect to Pi 4 if the connection implementation is over. Create a stub of a notification, send it from the testing fake, or Pi, then confirm that the notification is identical to the pre-defined stub.
20. Test that the Android device can "pair" with the system through the internet. Mechanism: similar to the Pi test, create sockets, connect, then pair. Ping the devices to be sure their connection is stable. A testing fake can be used if the Pi 4 device cannot pair yet.
21. Test that the system generates reports of incidents automatically. Mechanism: Launch the system, feed it a pre-recorded video with a detected danger, run the identification on it, logs the results, compare them to pre-defined logs.
22. Test that the system can receive responses from the Android device for the notifications. Mechanism: Use a testing mock to invoke a notification, send a stub response, receive on Pi 4 and compare it to the predefined stub.
23. Test that the system changes the mode as it receives no response from the Android device. Mechanism: Launch the system, bind to Android device, feed it video with a dangerous objects, send notification, confirm reception, don't respond, assert that the status of the system after 30 seconds is "Panic Mode"
24. Test that the system would refuse to connect to devices without the predefined password. Mechanism: Launch device connect to the internet, set up its IP and port to predefined values, attempt connection from Android device, assert the request was sent, then assert that Pi 4 is not connected to any device.
25. Test that the system can compress the live stream data before sending it over the internet. Mechanism: Launch the system, pair with a mobile, capture video of 10 seconds, store the size, compress, calculate the size, decompress, compare the size to confirm it is identical. Assert that the video is of the same size before compression and after decompression.
26. Test that the Android device can decompress the data before viewing it. Mechanism: Send compressed using Pi 4 or through a testing fake, receive, decompress, compare size, and verify that the file is not corrupted.

## The following tests are manual:

27. Test that the de-fisheye image is not limiting the view of the important elements of the room. Mechanism: capture a pic, de-fisheye, visually confirm no important details were removed from the picture after the cropping, and resizing.
28. Test that the infra-red sensor is working properly. Mechanism: Launch the camera, turn lights off, visually assert that the night vision is working.

29. Test that the speakers are producing audio with proper quality. Mechanism: Launch the system, play a high quality audio file, in person listen to the audio and confirm the quality is proper.
30. Test that the microphone can detect noises and record with proper quality. Mechanism: Launch the system, go near the microphone, make noises, play music, and talk, then save the file and listen to it, assert that the audio quality is proper.
31. Test that the audio and video content is of high quality (no choppiness, or electronic noise). Mechanism: Launch the system, go near the microphone, make noises, play music, and talk, then save the file and listen to it, assert that the audio quality is proper.
32. Test that the system can identify movement of the toddler and record it. Mechanism: Run the system, bring a toddler, let the toddler move, check the logs of the system to confirm the detection of movement.
33. Test that the system can identify noises in the night and record it. Mechanism: Launch system and produce loud noises, check the incidents and logs to confirm detection.
34. Test that the system can detect high brightness light in the night. Mechanism: Turn light off, run the system, bring a monitor or laptop set to maximum brightness, confirm from logs that the system detects it.
35. Test that the system generates correct analysis for the time of movement. Mechanism: run the system, bring a toddler, let the toddler move, check the logs of the system to confirm the detection of movement of specified length.
36. Test that the login page functions responsively. Mechanism: run the Android application, enter valid login credentials, assert the success.
37. Test that the details stored in the login page are retrieved and logs the user in. Mechanism: Upon success of previous test, close application, and open it again, assert that no login was required.
38. Test that the user is able to pair the device once logged in. Mechanism: Login, pair to Pi 4, enter the password, then open streaming window, assert live stream by being in the room seeing yourself live in the video.
39. Test that the Android device calibrates the video and streams the content. Mechanism: Login, pair to Pi 4, enter the password, then open streaming window, assert live stream by being in the room seeing yourself live in the video with no delay or choppiness.
40. Test that the Android device displays the identified objects through the stream. Mechanism: Login, pair to Pi 4, enter the password, then open streaming window, assert live stream by being in the room seeing yourself live in the video, bring a dangerous object in hand, check notifications on the phone.
41. Test that the Android device will store the necessary recorded information. Mechanism: Login, pair to Pi 4, enter the password, then open streaming window, assert live stream by being in the room seeing yourself live in the video, bring a dangerous object in hand, check the reports on the phone
42. Test that the Android device will allow the user to toggle between different safety options. Mechanism: Login, pair to Pi 4, enter the password, then open streaming window, set to max alertness, assert live stream by being in the room seeing yourself live in the video, assert notification of intruder, change to low alert, assert no notification, bring a dangerous object in hand, check notifications on the phone.

43. Test that the Android device is able to set lullabies and/or alarms. Mechanism: Login, pair to Pi 4, enter the password, then open streaming window, assert live stream by being in the room seeing yourself live in the video, play a lullaby confirm the audio being played

## Integrations Tests

When working in a team of several developers, each using his/her own style and logic to program the elements they are responsible for, new problems arise, some problems are regarding integration. Different software, or even hardware, components need to communicate with each other. This can be a challenging task, that is why Integration tests are the second level of testing that Vanny will go through to ensure that all components not just work, as tested in unit tests, but also are capable of communicating with each other in the desired and expected way. The following tests will be performed to ensure the proper integration of all elements in the system. The main method of integration testing used will be Functional Incremental Testing. In this method of testing the components' integration will be tested incrementally, piece-by-piece based on their functionality.

1. Test the interface link between the Watcher class and the streamer class.
2. Test the interface link between the Report Generator class and the Activity Analyzer class.
3. Test the interface link between the Watcher and the Environment Analyzer class.
4. Test the interface link between the Watcher and the object identifier class.
5. Test the interface link between the Report Generator and the Activity Analyzer class.
6. Test the format and encoding compatibility between the video encoder and the Stream Manager.
7. Test the compatibility of data format, compression and encoding between the Stream Sender, on Pi 4, and the Stream Receiver, on the Android mobile.

## System Tests

System tests are performed relatively late in the development lifecycle. The main purpose of such tests is to validate the finished software product and evaluate the completeness of specifications. There are many types of system testing. The following tests will be performed as a part of Vanny's system tests.

1. Usability Testing: focuses mainly on the user's ease of use. Mechanism: bring a person unfamiliar with the system, give them a brief explanation and a simple guide on how to use the system then observe them using all the functionalities of the system, repeat with multiple people, take feedback and improve the product accordingly.
2. Load Testing: focuses on the behavior of the system under expected load. Mechanism: run the system with a toddler in a room, pre-recorded or live, introduce dangerous objects to the environment every 15 minutes and observe the reaction time of the system

and the performance when such objects are introduced. Additionally, at the time of the reports' generation an object will be introduced while a live stream is being performed. This is considered an expected load as all of these events can indeed happen at the same time.

3. Stress testing: focuses on the behavior of the system as the load exceeds the expected maximum load. The aim is to help us understand the behavior of Vanny in such incidents. Mechanism: introduce detectable objects and people one by one and keep them in the scope of the camera, use multiple devices for streaming and generate multiple reports at the same time. Keep increasing the number of objects, devices, reports, and people until the system gets overloaded; observe the behavior. Detect bottlenecks and accommodate and optimize accordingly, if needed.
4. Hardware/Software Testing: focus on the interactions between hardware and software. Many types of tests can be performed under this category. The focus for our project would be Performance testing, since the Pi is limited in terms of performance and that it lacks a capable GPU. Considering the large amount of graphic data processed in the system this could become a choking point for the system. Mechanism: Set the frames rate to an acceptable range, like 20 fps. Perform load tests on the system, observe the behavior and the performance, if the system is slow or sluggish, drop the frames rate. Keep repeating until the system becomes stable with a specific frames rate. Perform a stress test, observe the system. Throughout the test keep logs of the systems' element usage rate, as well as memory writing rate.

# 5. Considerations of Various Factors in Engineering Design

## 5.1 Public Health and Safety

Vanny protects babies from any harm. We detect any sharp object near the baby which might cause any harm and warn the parents. We also detect any stranger who might be a baby. It also detects any irregular behavior of the baby which might be due to illness and warns the parents early. The application also ensures the baby has a healthy sleeping pattern.

## 5.2 Public Welfare

Our system is much cheaper than most other available systems in the market as it only uses commodity hardware. Also, Vanny only has a one-time cost as the system doesn't use any remote backend.

## 5.3 Public Information Security

Every data is encrypted and communicated to the client through a secure channel. The application doesn't store any permanent user data. It deletes everything after a maximum of a week. The application is also not connected to a remote backend. Hence, the maker of the system doesn't have access to any user data. This was designed like that intentionally to increase the consumer's confidence in using the system. Vanny provides information about a baby who is alone in his/her home. Hence, privacy issues need to be taken very seriously to prevent any access to user data by any third party including the maker of the system.

## 5.4 Global Factors

The application is available in many countries around the world. The application will be trained in multiple languages. Hence, it can be used in different parts of the world.

## 5.5 Cultural Factors

The application needs audio processing. Although a child might only utter the simplest words, it will still be different for different languages. Hence, we need to train our audio ML model specific to the language the child speaks. This way we may be able to understand the child's words and provide more nuanced information to the parents.

## 5.6 Environmental Factors

No Environmental factors. Vanny doesn't harm the environment in any way.

## 5.7 Economic Factors

Vanny saves the salary that the parents had to pay the real nanny. Parents can work longer hours in their job without much worry because Vanny is always there to look after their children. Hence, they contribute to the economy more. Vanny also costs less than other systems in the market as it uses commodity hardware only. This also saves money for the parents.

## 5.8 Social Factors

Vanny creates a social distance between the baby and the parents as there will be fewer real-life interactions. However, we have taken some steps to ensure that this social distance is reduced to a minimum. Vanny can play the voices of the parents to the baby. It is also possible for the parents to interact with the baby in real-time with live video chat which can be played on the monitor in front of the baby. However, this feature will only be available in future versions.

| | Effect Level (out of 10) | Effect |
|---|---|---|
| Public Health and Safety | 8 | Protects babies from any harm. |
| Public Welfare | 3 | Cheaper than other available options. |
| Public Information Security | 9 | Encrypts everything, has no remote backend and deletes everything after a week. |
| Global Factors | 2 | Applicable in any country |
| Cultural Factors | 4 | Language differences in audio processing are considered. |
| Environmental Factors | 0 | None |
| Economic Factors | 8 | Saves money for the parents as they don't need real money. |
| Social Factors | 4 | Creates a distance between the baby and the parents. |

*Figure 5. Evaluation of factors in engineering design*

# 6. Teamwork Details

Teamwork is one of the most essential parts of any project. Each team member must have nearly equal responsibilities and each member needs to participate in each deliverable. In order to ensure those critical aspects of teamwork, we needed to make necessary measurements and adjustments in order to create the environment that is needed for effective teamwork. In this section, the contribution of the team members, the environment that enables and increases those contributions, and the leading roles and shared leadership status among the team are described.

## 6.1 Contributing and Functioning Effectively on the Team

Our project is a comprehensive one that requires knowledge about both hardware and software systems including the knowledge about setting the communication links between those systems. Each team member has a distinct and unique specialty that he/she gains throughout the university or work life. So in order to achieve the best possible contribution and functioning in the team, we decided which member would be the most knowledgeable one in a specific task. According to those specifications, we formed subgroups that work together to finalize the subtasks of the project. That's how everyone works on the parts that he/she is most suitable and effective with.

As another measurement, we know that each member has several classes and other kinds of heavy work that he/she needs to complete other than the project. So we also need to consider those special situations to ensure that everyone can contribute without being overwhelmed. For example, if some team members have exams or other urgent assignments, we do not schedule any meetings in those weeks to ease the workload of the team members.

## 6.2 Helping to Create a Collaborative and Inclusive Environment

There are several methods to create a collaborative environment for efficient development. In this section of the report, some of those measurements are listed in detail.

- **Regular Meetings:** There need to be regular meetings that are scheduled and synchronous. Only being in communication using chatting platforms such as WhatsApp or Jira is not enough for efficient communication, rather there need to be synchronous meetings where each member can ask questions regarding any aspect of the project to the others that may have the answers to those questions. Those meetings should not be on arbitrary dates but they should be scheduled regularly sufficiently before the meeting day so that each member can be better prepared for the meeting.

- **GitHub:** GitHub is a great platform that we can manage version control over our project. It is also possible to see the commits and progress of the other team members via Github so that we can see what the other members are doing and who needs additional help in his/her current work. GitHub issues also create a great environment where we can note the required changes, implementations, or bug fixes that are necessary for the project. Using Pull Requests from different branches, we are able to review the coded features of team members and suggest changes to those implementations when needed. Overall GitHub is a tool that helps us to create a collaborative environment that every team member can make use of.

- **Slack:** Rather than using WhatsApp as the asynchronous messaging platform for our project, we prefer to use Slack. Using Slack feels more professional in terms of development and it reduces the distraction that comes with the usage of WhatsApp. Being able to create channels in Slack enables us to divide each subtask into channels on Slack and keep the communication in the relevant channel which ensures abstraction. Using the Huddle feature of Slack, we are also able to communicate via voice without spending much time scheduling a zoom meeting or a regular meeting. Instead, we are able to resolve small bugs or small misunderstandings using a quick huddle between the related team members about the issue.

- **Logbooks:** One of the helping regulations of the Senior Project is that each team member needs to have a logbook in which he/she documents his/her progress regarding the project. Keeping a logbook is an encouraging activity that increases productivity since it is valuable to see the work done regularly in the logbook. To see and increment the progress made, each team member is eager to update his/her logbook on new progress. Therefore logbooks are also one of the most valuable things that help create a productive work environment.

## 6.3 Taking Lead Role and Sharing Leadership on the Team

As a group we believe that leadership should be handled in each different subtask in the development. There should not be only one leader that is leading the whole process but rather whoever is capable of a subtask he/she needs to lead the group that is responsible for the relevant subtask. So while we are dividing the team into subgroups for different implementation stages such as machine learning, frontend, and hardware sides, we consider the most eligible participant in the corresponding subtask as the leader of that group. In case the leader thinks that additional help is needed in a specific task he/she can request help from different subgroups and according to the decision of the leader in the other subgroup, the workload can be divided with balance.

Therefore, the leadership is shared among the team and whoever is eligible for the leadership of the relevant task, he/she is eager to take the lead in that task and manage the rest of the team members.

# 7. References


[1] A. S. Inc., “Alfred Home Security camera,” *App Store*, 02-May-2015. [Online]. Available: https://apps.apple.com/us/app/alfred-home-security-camera/id966460837. [Accessed: 09-March-2023].

[2] A. B. A. s.r.o., “Annie Baby Monitor: Nanny cam,” *App Store*, 30-Mar-2015. [Online]. Available: https://apps.apple.com/us/app/annie-baby-monitor-nanny-cam/id944269894. [Accessed: 09-March-2023].

[3] T. LLC, “Baby Monitor Saby + Cry Alarm,” *App Store*, 05-Aug-2020. [Online]. Available: https://apps.apple.com/us/app/baby-monitor-saby-cry-alarm/id1503992310. [Accessed: 09-March-2023].

[4] A. Shabunia, “Camy - live video monitoring,” *App Store*, 09-Sep-2019. [Online]. Available: https://apps.apple.com/us/app/camy-live-video-monitoring/id1475627437. [Accessed: 09-March-2023].

[5] M. V. Solutions, “Video surveillance ivideon,” *App Store*, 16-Sep-2011. [Online]. Available: https://apps.apple.com/us/app/video-surveillance-ivideon/id464234175. [Accessed: 09-March-2023].

[6] “Mi home security camera1080p,” *Mi Home Security Camera Wireless IP Surveillance System 1080p HD Night Vision (US Version with Warranty) - Xiaomi United States*. [Online]. Available: https://www.mi.com/us/mi-home-security-camera. [Accessed: 09-March-2023].

[7] “Angel Eyes Ai - Microsoft AI Lab,” *Angel Eyes AI - Microsoft AI Lab*. [Online]. Available: https://www.microsoft.com/en-us/ai/ai-lab-angel-eyes. [Accessed: 09-March-2023].

[8] D. Maloney, “Machine learning baby monitor prevents the hunger games,” *Hackaday*, 23-July-2022. [Online]. Available: https://hackaday.com/2022/07/23/machine-learning-baby-monitor-prevents-the-hunger-game s/. [Accessed: 09-March-2023].