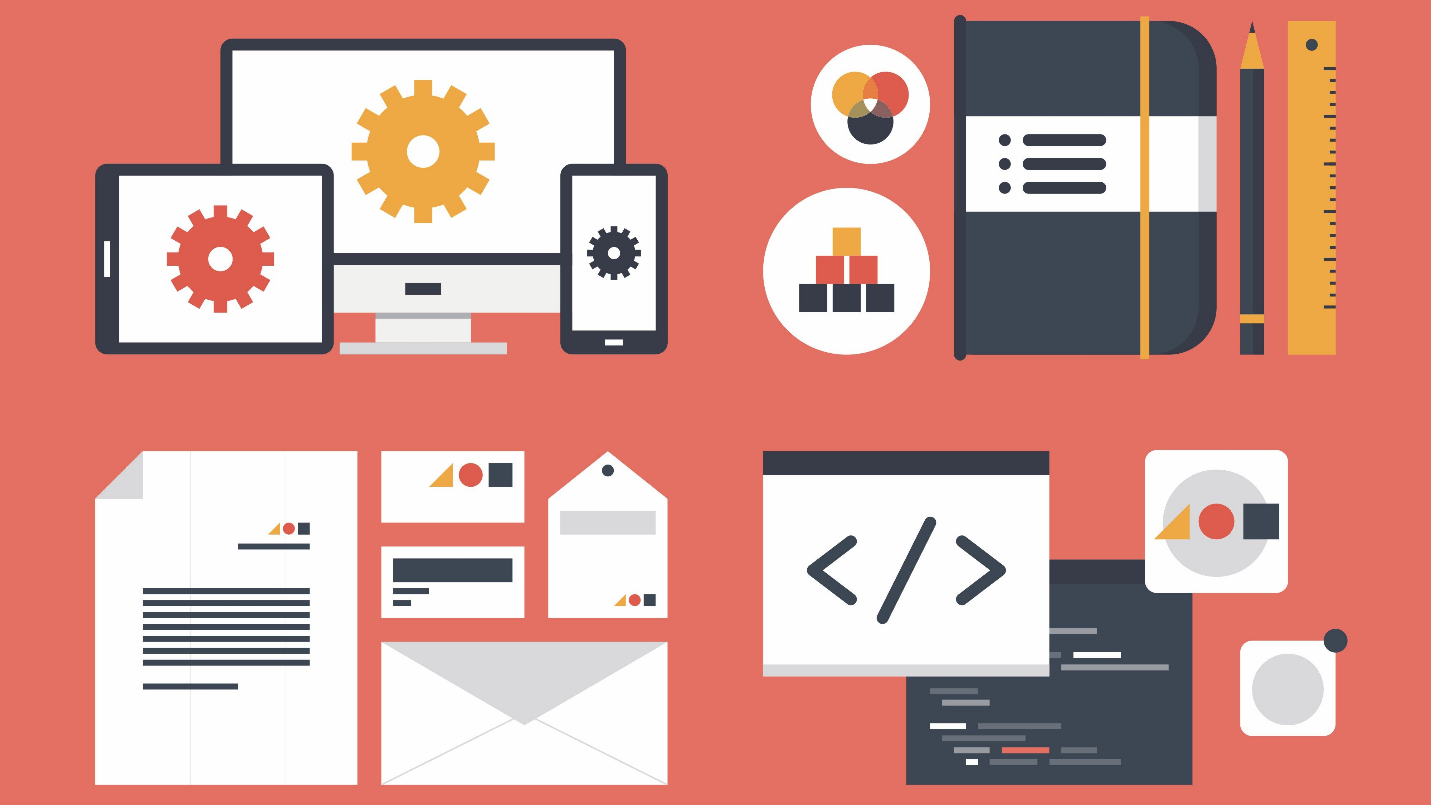
Design document



**Name: Tony Jiang**

**Date: 19 Sep 23**

**Project: Video call system**

| **Versions** | **Date** | **Description** |
| --- | --- | --- |
| **0.1** | **19 Sep 23** | **Initial document** |
| **0.2** | **6 Oct 23** | **Correct spelling errors.**  **Explained the UML diagram.**  **Make single quotes to double quotes.** |
| **0.3** | **12 Oct 23** | **Change the ERD section.** |
| **1.0** | **13 Oct 23** | **Initial documentation complete.** |
| **1.1** | **2 Nov 23** | **Update the section for “User acceptance test”.** |
| **1.2** | **13 Dec 23** | **Update the section for “User acceptance test”,**  **added a header for “Prototype concept 1” and added user acceptance test for prototype concept 2.** |
| **1.3** | **3 Jan 24** | **Update level 1, 2, 3 and 4 of System Architecture.** |

Contents

[Introduction 5](#_Toc155359894)

[Context 5](#_Toc155359895)

[Goals 5](#_Toc155359896)

[Scope 5](#_Toc155359897)

[Functionality 6](#_Toc155359898)

[System Architecture 6](#_Toc155359899)

[Level 1: System Context (C1) 6](#_Toc155359900)

[Prototype 1 7](#_Toc155359901)

[Prototype 2 8](#_Toc155359902)

[Level 2: Containers (C2) 9](#_Toc155359903)

[Level 3: Component (C3) 10](#_Toc155359904)

[Frontend Windows Form Application [container] 11](#_Toc155359905)

[Backend [Container] 12](#_Toc155359906)

[Prototype 1 13](#_Toc155359907)

[Prototype 2 14](#_Toc155359908)

[Level 4: Code (C4) 15](#_Toc155359909)

[Prototype 1 16](#_Toc155359910)

[Prototype 2 17](#_Toc155359911)

[User Interface Design 19](#_Toc155359912)

[Wireframe 19](#_Toc155359913)

[User Flowchart 19](#_Toc155359914)

[Database Design 20](#_Toc155359915)

[Testing Strategies 21](#_Toc155359916)

[Test approach 21](#_Toc155359917)

[Unit test 22](#_Toc155359918)

[User acceptance test 23](#_Toc155359919)

[Prototype concept 1 23](#_Toc155359920)

[Prototype concept 2 24](#_Toc155359921)

[Additional feature for concept 1 27](#_Toc155359922)

# Introduction

This document covers all the technical aspects of this project, including its structure and the design decisions made throughout the development. It helps to further reflect on the design choices within this project. By doing so, it helps with planning how everything should be configured and address any potential development challenges. Additionally, it helps in conveying the intended design to other developers, ensuring a shared understanding and agreement on the design approach. Some information may refer to other documents where the information is fully detailed. New content or components will be added to this document throughout the project, and it is intended for technical developers.

The project’s objective is to create two or three solutions for a video call system within the PRAS application.

# Context

This is to give general information about the project. What’s the project about and from whom. For context, please refer to the **Project Plan** document, on “*1.1 Context*”.

# Goals

This is to give information about what the project is trying to achieve and what the purpose of the project is. For the goals of this project, please refer to the **Project Plan** document, on “*1.2 Goal of the project*”.

# Scope

This is to define what needs to be delivered for the entire project, to ensure clarity for all stakeholders. The scope of the project can be found in the **Project Plan** document, on “*1.3 Scope and preconditions”*.

# Functionality

This is to define the core functionality of the system, so that it’s clear to everyone. To achieve a clear understanding of the functionality of the system, we use something called “user story”. User story describes the functionality of the product, the expected behavior, and the required components. You can find the user stories in the **User Story** document.

# System Architecture

The purpose of System Architecture is to describe the internal system’s overall structure and establish an agreement on the desired design of the system. To make it easy to describe and communicate the system’s architecture, we’ll use a C4 architecture diagram. It’s an architecture design that is easy to understand. The design approach is straightforward and helps us communicate how each part of the system should be set up, even to a non-technical person. It’s like using Google Maps and you’re trying to zoom in on a place in Aruba. The closer you zoom the more details you see. It’s like that. The C4 architecture diagram helps us do the same for our software’s structure.

The C4 architecture diagram has 4 level:

* Level 1: System Context (C1)
* Level 2: Containers (C2)
* Level 3: Component (C3)
* Level 4: Code (C4)

We will go to each level and describe what they do when we reach there. The tool that was used to create the C4 model is [Visual Paradigm online](https://online.visual-paradigm.com/diagrams/features/c4-model-tool/). The free version.

## Level 1: System Context (C1)

At this level you’ll discover which intended user is going to use the system and what the system does. This provides an overview of the entire system. Additionally, it helps to describe what the system does to non-technical people.

### Prototype 1

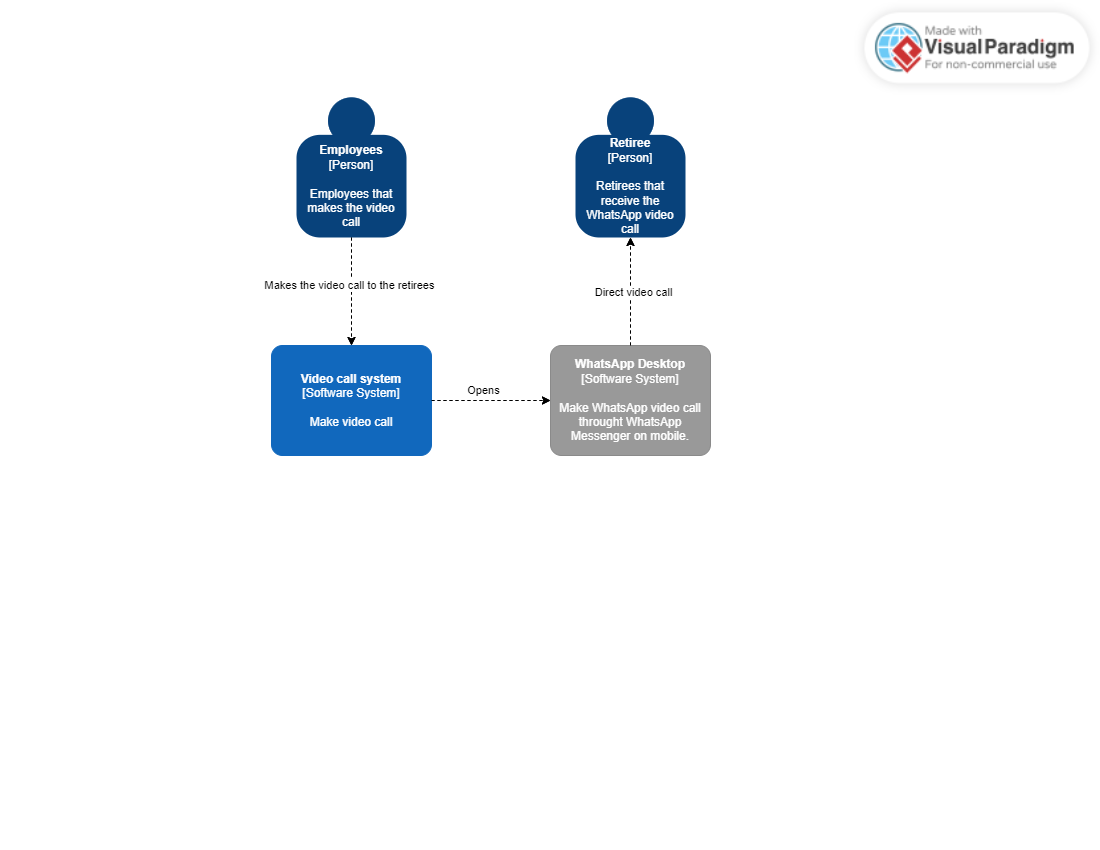


Figure : C1 model prototype 1

In this C1 model of prototype 1, you can see that employees can make video calls to retirees using the video call system. The system will open WhatsApp desktop to video call the retiree on WhatsApp mobile. This provides an understanding of which users can use the system, what kind of system it is and the action they perform on the system.

### Prototype 2

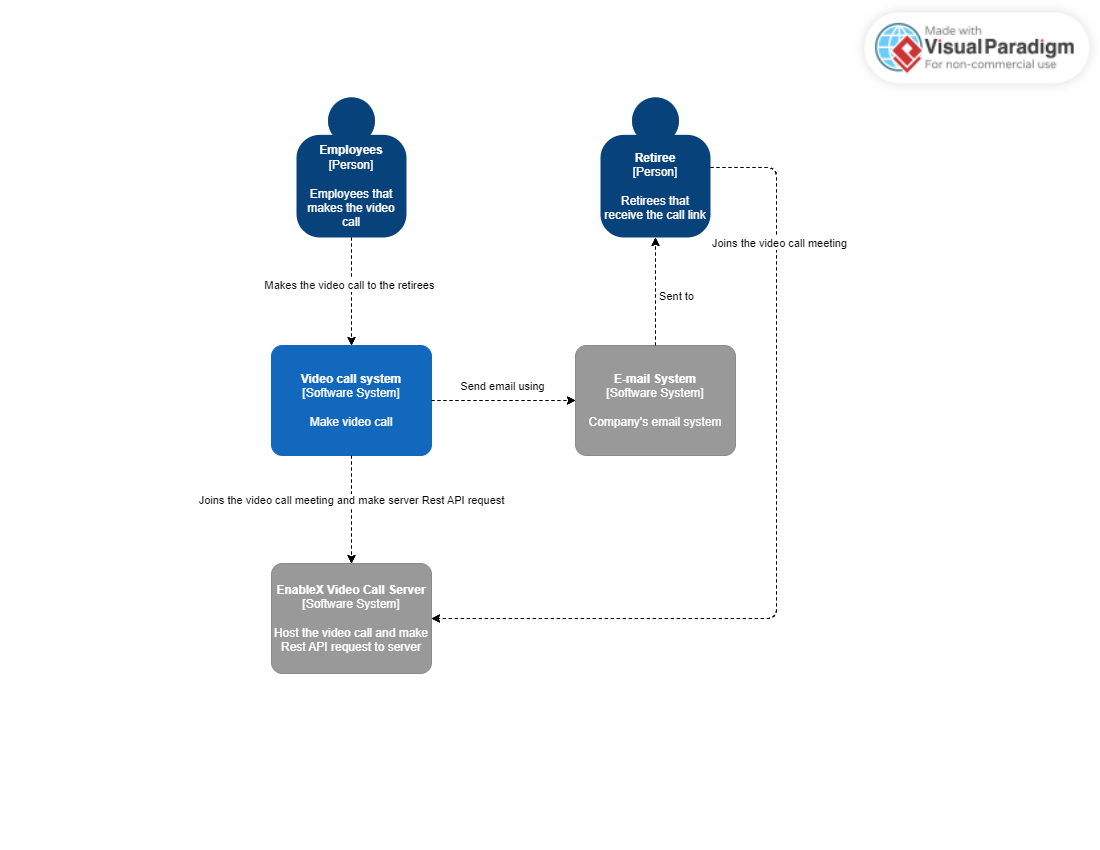


Figure : C1 model prototype 2

In this C1 model of prototype 2, you can see that employees can make video calls to retirees using the video call system. The system will send an email of the video call URL, where the retiree will be able to join the video call meeting. This video meeting is hosted through EnableX Server. The system is able to make Server API request to the EnableX Server. This provides an understanding of which users can use the system, what kind of system it is and the action they perform on the system.

## Level 2: Containers (C2)

At this level, we delve deeper into the system’s architecture, moving beyond the high-level system context of level 1. Here, we identify the high-level technical building blocks or “containers” that make up the system and understand the relationship between them. These containers represent the applications or databases used to build the entire system. It’s important to note that the information presented from this level onwards is intended for the technical audience. This model is the same for both prototypes.

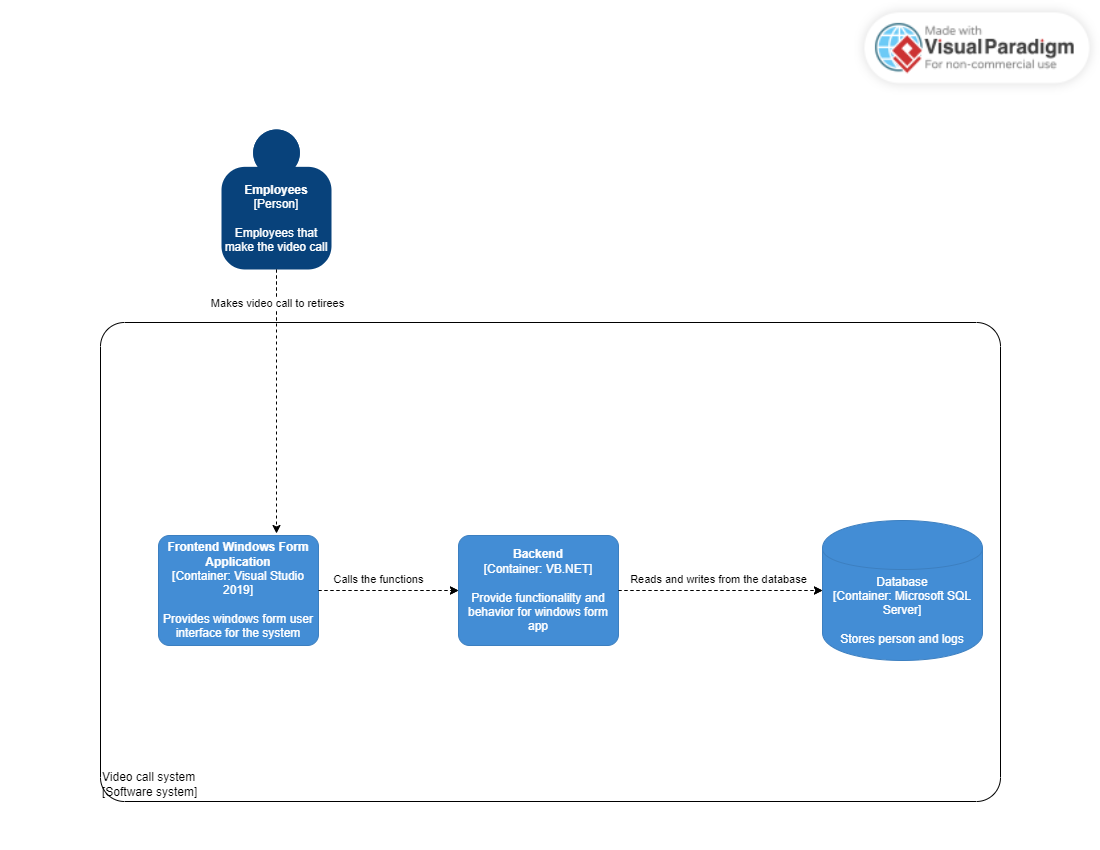


Figure : C2 model for both prototypes

It’s pretty understandable how the system is put together from the C2 model. There are 3 containers that make up the system.

* **Frontend Windows Form Application**
* This component serves as the user interface, where the users interact with the system.
* It communicates with the backend to process user’s actions.

**Technology: Visual Studio 2019**

* The PRAS application is developed in Visual Studio 2019
* The PRAS application uses Windows Forms App (.Net Framework)
* It ensures seamless integration within the same IDE for the video call system.
* **Backend**
* “All the functional code responsible for the system’s operations, resides here.
* It manages data flow between the front end and the database.

**Technology: Visual Basic .NET (VB.NET)**

* The backend is written in VB.NET language for the PRAS application.
* Utilizing the same language for simpler integration into the PRAS application.
* **Database**
* This container stores all the retirees’ information.
* It provides data to the backend as needed.

**Technology: Microsoft SQL Server**

* The PRAS application relies on Microsoft SQL Server as its database system.
* Utilizing the PRAS application’s existing database simplifies the process of fetching and adding data and avoid the necessity of making a new database and populating it.

## Level 3: Component (C3)

Level 3 represents the component (C3) of the container from level 2 of the system’s architecture. These components are the building blocks that make up the containers of level 2 and they interact with each other. These components are categorized by the function they are assigned to do.

### Frontend Windows Form Application [container]

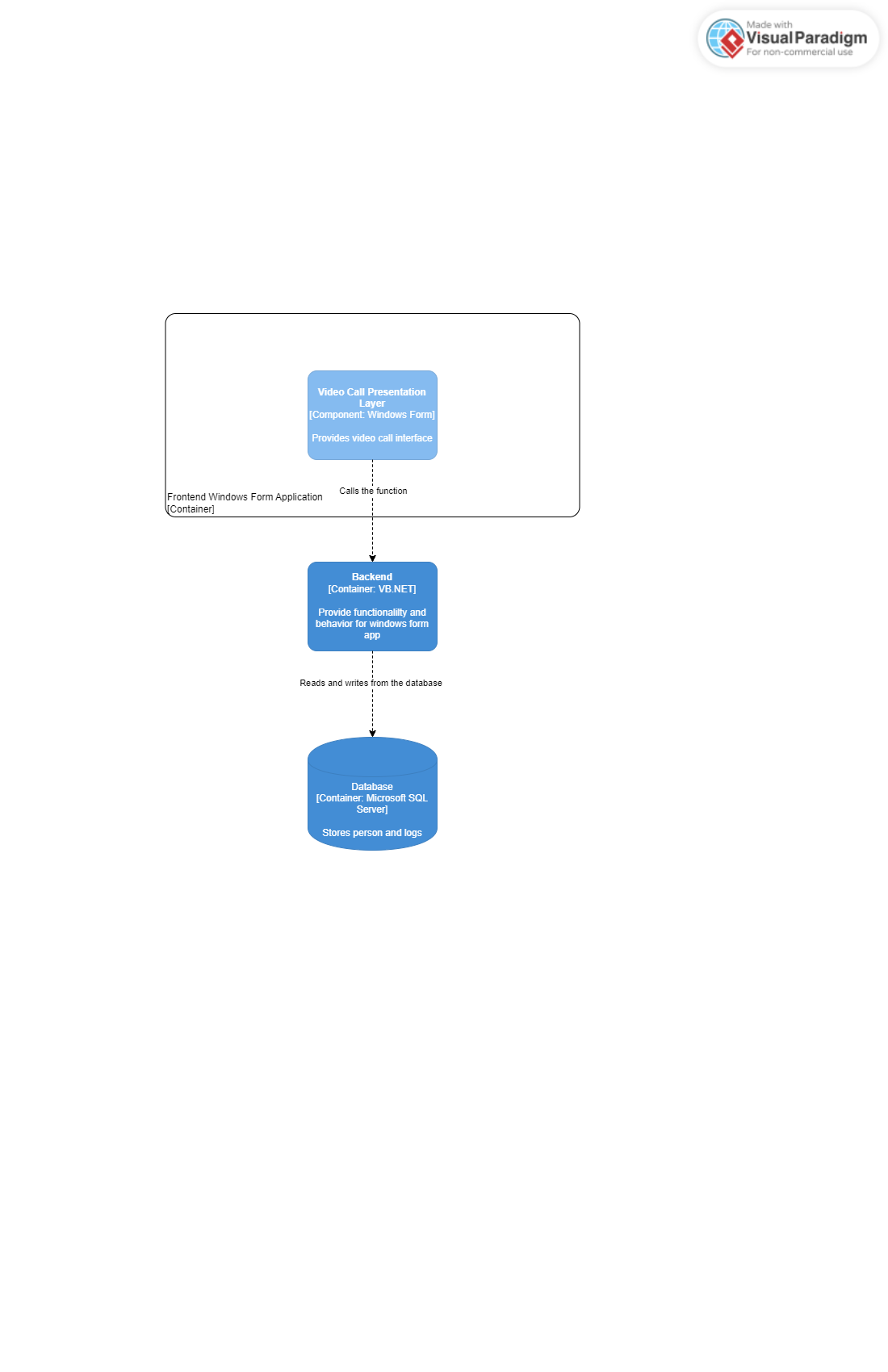


Figure : C3 model from Frontend Windows Form Application for both prototypes

This is the C3 model of the “Frontend Windows Form Application” container. This represent for both prototypes. It contains 1 component. The design idea here is to sperate the user interfaces with destinct functions.

* Video Call Presentation Layer: Here lies all the video call user interfaces.

### Backend [Container]

The backend container for the two prototypes is a bit different from each other.

### Prototype 1

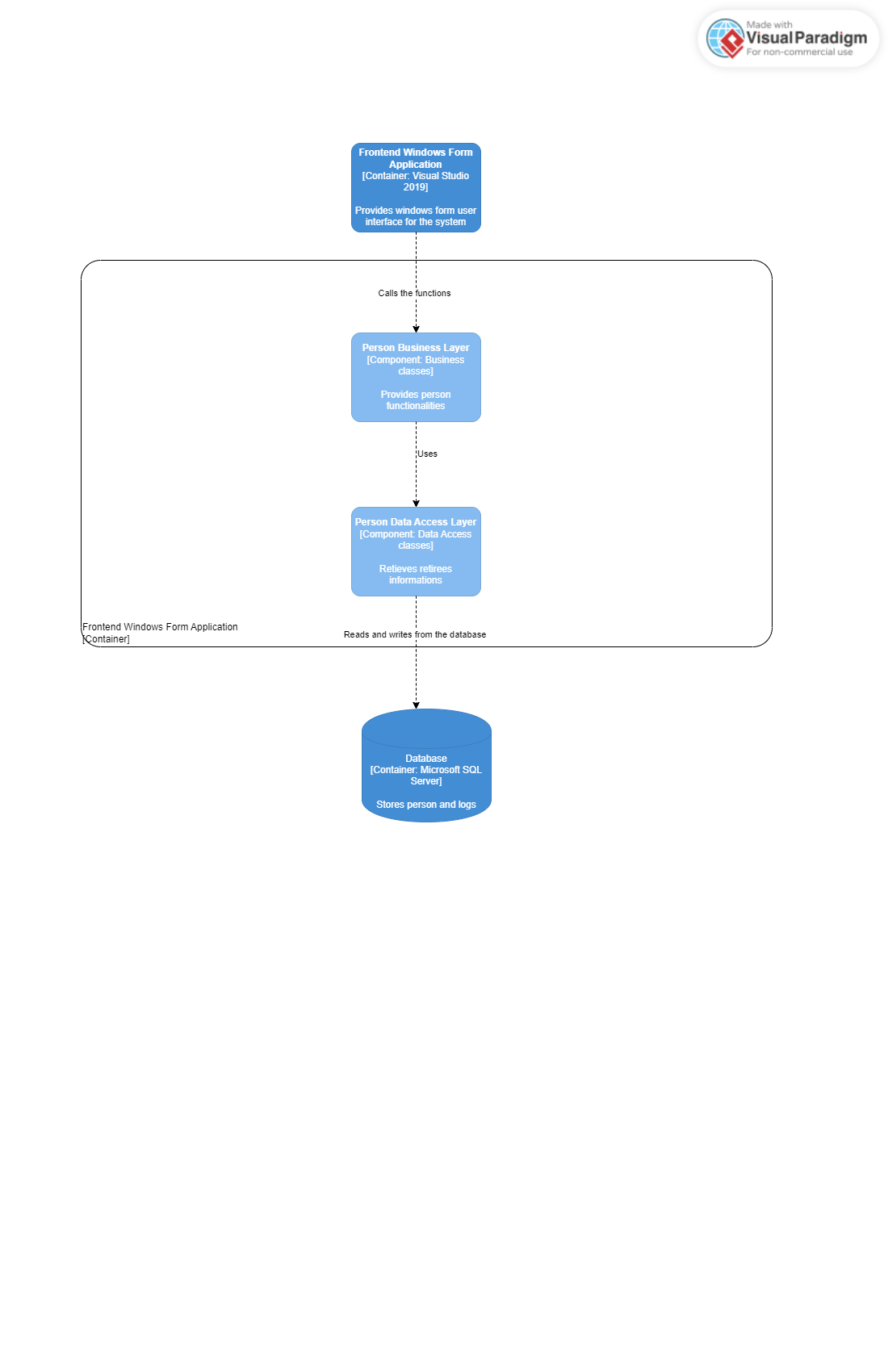


Figure : C3 model from the backend container of prototype 1

This diagram shows the structure of the Backend container for prototype 1, which consist of 2 components that do their own task. This separation of tasks simplifies the understanding of the design and for future additional components.

* Person Business Layer
* Contains all the person functions.
* It uses the “Person Data Access Layer” to retrieve retiree data.
* Person Data Access Layer

Manages data flows from the database to specific video call functions.

### Prototype 2

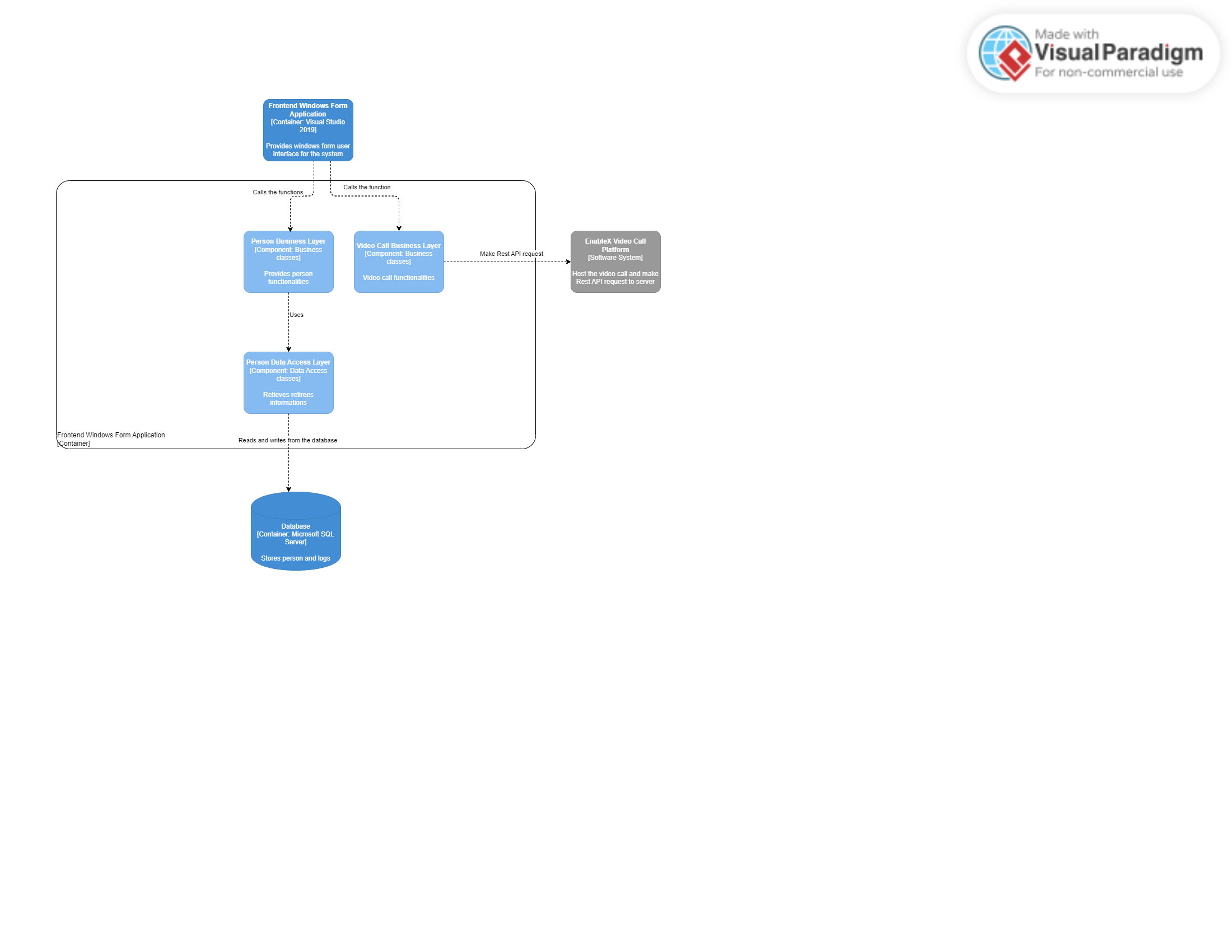


Figure : C3 model from backend container of prototype 2

This diagram shows the structure of the Backend container of prototype 2, which consist of 3 components that do their own task. This separation of tasks simplifies the understanding of the design and for future additional components.

* Person Business Layer
* Contains all the person functions.
* It uses the “Person Data Access Layer” to retrieve retiree data.
* Video Call Business Layer
* Contains all the video call functions.
* Make Server API request to EnableX Server.
* Person Data Access Layer
* Manages data flows from the database to specific video call functions.

## Level 4: Code (C4)

This level includes UML (Unified Modeling Language) diagrams. UML diagrams are a type of visual tool used in software engineering to illustrate the structure and relationships of classes within a system or software application. They provide a high-level overview of the system by displaying the attributes and methods of each class within it. UML diagrams are valuable for planning and conveying how a system is built, and they are easily understandable, even for newcomers.

In our documentation, each component will include a UML diagram. This level is typically considered optional and is primarily used to illustrate complex class structures and explain their functionality. It will be used to depict the structure of the video call system in the backend and provide insights into the design reasons.

Please note that this UML diagram will be continually maintained throughout the project, ensuring that it accurately reflects the system's structure. We will be using [Lucidchart](https://lucid.app/lucidchart/be5d23f6-d824-4ac6-a638-4433423f997f/edit?viewport_loc=-502%2C-338%2C2560%2C1154%2C0_0&invitationId=inv_5ec458a2-d156-4647-8bcc-0a516a275dc4) as the tool for creating and updating these UML diagrams.

### Prototype 1

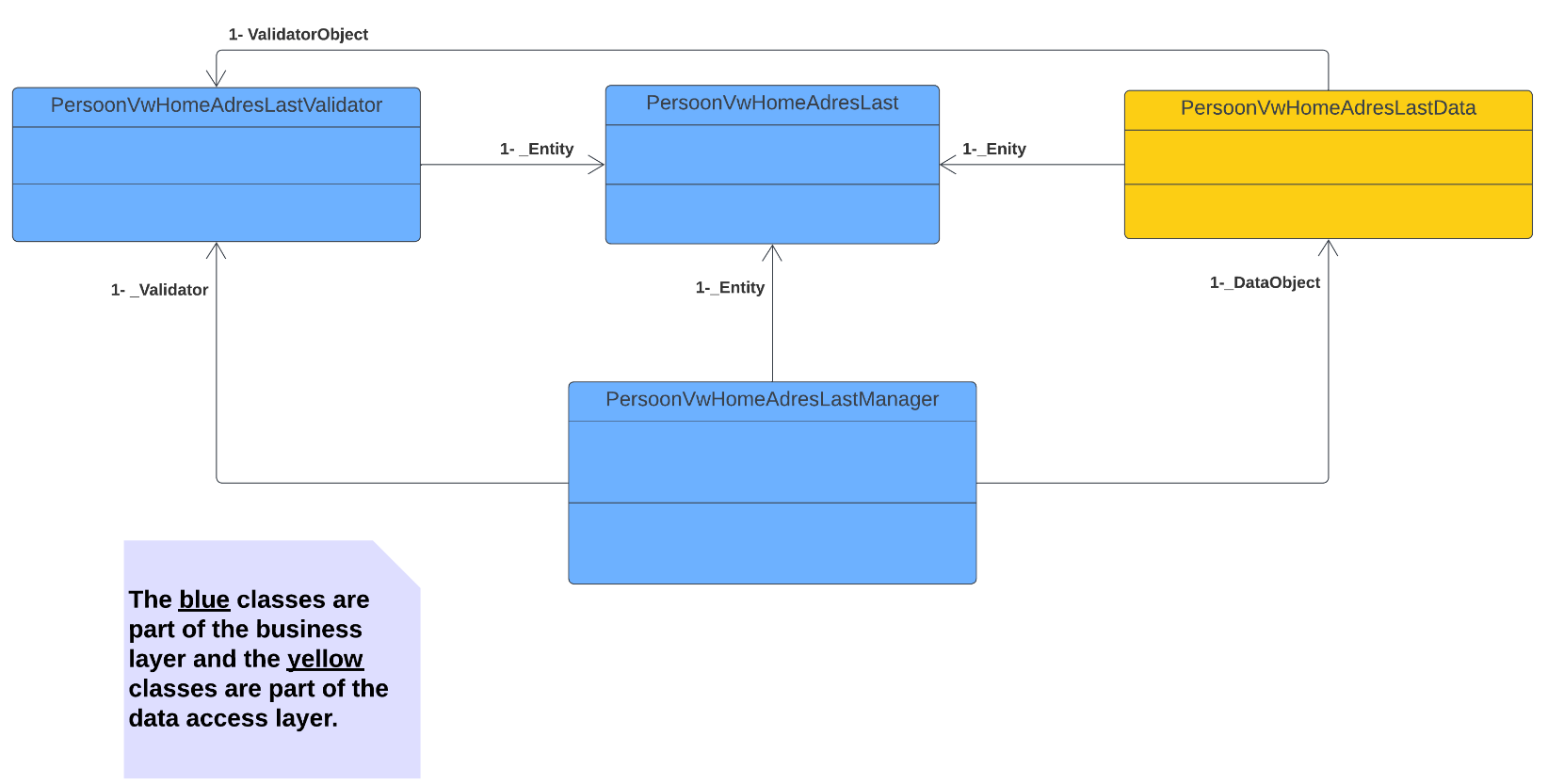


Figure : UML diagram of retiree

For Prototype 1, the UML diagram is already define in the project; I’m showing which classes I’m using to retrieve the retirees latest information. I’m going to define what each classes do in this diagram; you can see the diagram at figure 7.

PersoonVwHomeAdresLast class is the entity class.

PersoonVwHomeAdresLastValidator class is to validate all the properties of PersoonVwHomeAdresLast class.

PersoonVwHomeAdresLastData class is part of the data access layer. It makes connection from the system to the database. It interacts with the database to execute queries, insert updates, delete records, and retrieve data. It validates the data before passing it to the entity class, and vice versa.

PersoonVwHomeAdresLastManager class establish communication and collaboration between the backend and frontend. It also coordinates the classes on what to do.

### Prototype 2

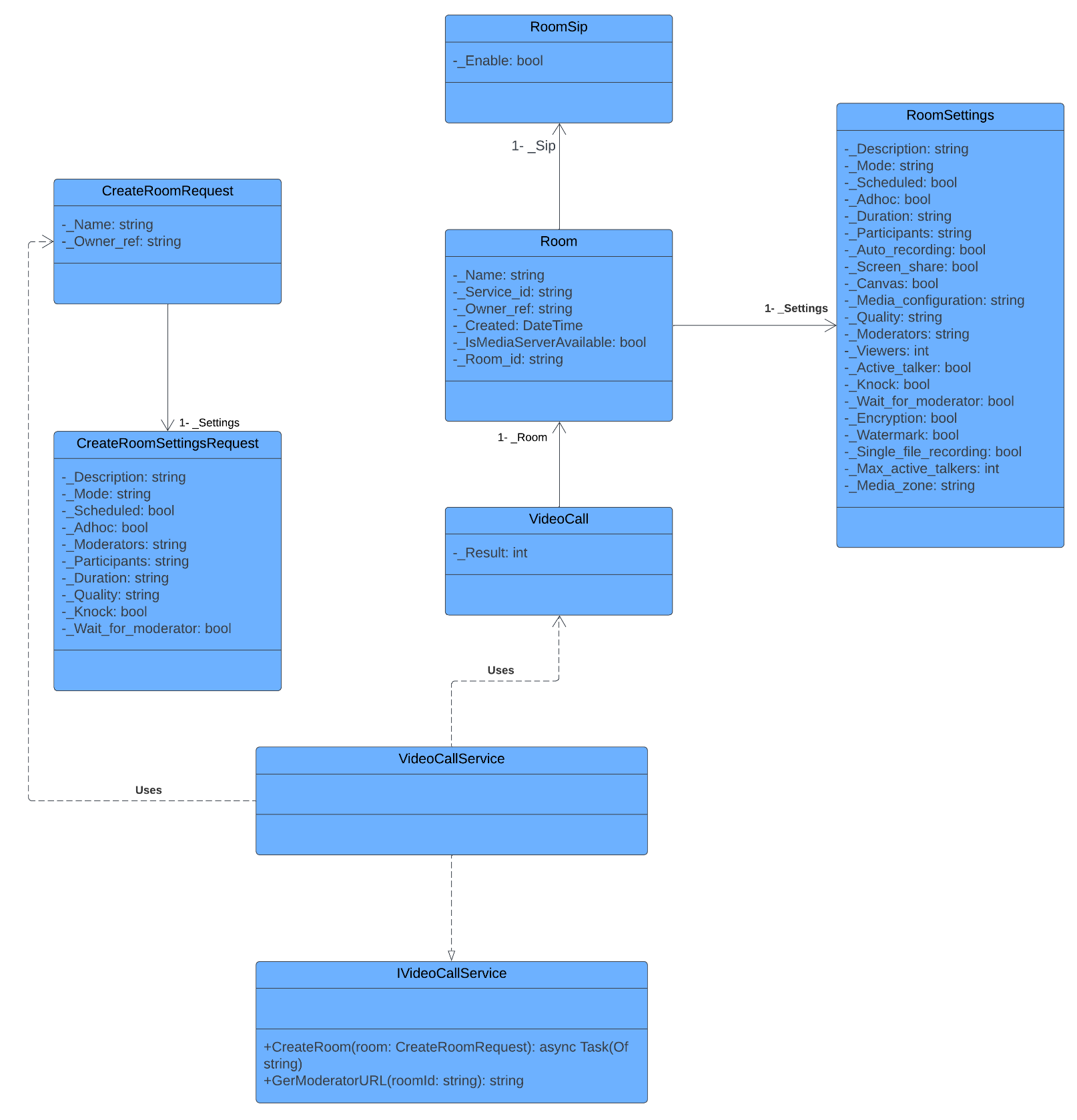
****

Figure : UML diagram of video call

For prototype 2, I’m trying to structure the UML diagram to follow the SOLID principles, which helps in maintaining, extending, and understanding the UML design. This approach helps in building an adaptive, effective, and agile software application, primarily in object-oriented software development. Additionally, it simplifies the process of extending, modifying, testing, and refactoring the code.

This prototype utilizes the diagrams from Figures 6 and 5. The explanation for the diagram in Figure 5 is defined in Prototype 1

**SOLID** stands for:

* **S**ingle Responsibility Principle

It states a class should only have one reason to change, meaning it should have a single job or a single responsibility. The UML diagram follows this principle. For example, the “VideoCall” class is responsible for the core domain model and actions related to the video room meeting, while the “VideoCallService” class handles higher-level logic and operations for managing collections of room meeting, such as “CreateRoom” and “GetModeratorURL” methods. The “CreateRoomRequest” class is where the input request for the “VideoCallService” class.

* **O**pen-Closed Principle

It states that classes should be open for extension but closed for modification. This means that they should be open to extending their functionalities but closed to making changes directly to the class's code. It is implemented in this UML diagram. For example, the “IVideoCallService” Interface can add new functionalities and can be implemented by other classes without altering the existing functionalities in the interface or the “VideoCallService” class.

* **L**iskov Substitution Principle

It states that objects of a derived class should be able to replace objects of the base class without affecting the correctness of the program. In other words, the sub-class that extends from the base class should inherit the functionalities and properties from the base-class and should be able to replace methods without causing any problems for the base-class. This is not implemented in the UML diagram because currently there are no classes that inherit from a base class.

* **I**nterface Segregation Principle

It states that a client should never be forced to implement methods of an interface that it doesn't use. In other words, an interface should be specific to the needs of the implementing class. In the UML diagram, this principle is implemented with the “IVideoCallService” interface, which is designed specifically for the “VideoCallService” class. No other classes implement this interface.

* **D**ependency Inversion Principle

It states that High-level modules should not depend on low-level modules. Both should depend on abstractions, such as interfaces or abstract classes. In other words, classes should depend on interfaces or abstract classes instead of concrete classes and functions. It encourages a more flexible and maintainable software design by reducing tight coupling between components and allowing for easier substitution of implementations. This is implemented in the UML diagram, as you can see that the “VideoCallService” class implements from the “IVideoCallService” interface. You can create another class to implement this interface.

# User Interface Design

The purpose of this is to present the user interface to all the stakeholders and reach an agreement on the product’s visual design. For illustrating and conveying the idea, we’ll make wireframes and user flowcharts.

## Wireframe

This is to illustrate the user interface design, we made use of wireframes, which are mockup designs of an application or website. There are websites or applications that allow you to easily create wireframes for free. You can create, export, demonstrate, and get feedback on it instantly. With the wireframe you can also provide detailed descriptions for each component on the wireframe. The wireframes can be found in the **Wireframe document**. Tool used for creating the wireframe: [Figma](https://www.figma.com/file/HlLEH2MbfMa5GbZs4ryU4N/Internship-semester-5-project?type=design&node-id=111%3A4&mode=design&t=nlHN2i57nTdvGZfc-1).

## User Flowchart

The user flowchart is a flowchart that illustrates the interaction between a user and the system. It provides visual representation of how they interact with each other. It serves as a useful tool for all stakeholders, including non-technical ones, to understand and communicate how these interactions should function. The user flowcharts can be found in the “**User Flowchart**” document, which was created using the [draw.io](https://app.diagrams.net/) tool.

# Database Design

This process helps us design, identify, and manage the database. It involves designing the overall table structure, identifying the types of tables required for the database, establishing the relationships between them, and maintaining and improving the database's design and data structure. The database design we are using is based on the Entity Relation Diagram (ERD). It gives a better understanding of how the database is designed. Some entities that are in the diagram are predefined in the company’s database, and we will utilize these entities to retrieve data. Our focus is to design the database for the video call system in the PRAS app. The tool for creating the ERD is an inhouse company used tool: [Aqua Data studio](https://www.aquafold.com/).

A diagram of a computer generated data

Description automatically generated with medium confidence

Figure : ERD

The entities that I’m going to use is already define in the company’s database and they are written in Dutch. The “Persoon (Person)” , “PersoonAdres (PersonAddress)” and “Adres (Address)” are entities from the company’s database. They are used to get retirees’ information.

For prototype 1, it’s to retrieve the retiree’s telephone type and the mobile numbers. For prototype 2, is to retrieve the retiree’s email address.

# Testing Strategies

This is to plan out how I’m going to test the video call system, what type of strategy I’m going to use for testing, and the reasons for the tests. For these tests, you would need to follow step-by-step instructions on how to test the system. The summarized version of testing strategies is in the project plan document in section 4.1,” Testing strategies”.

## Test approach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test type | Explanation | When should you start | Test written by | Tested by |
| Unit test | This is to test the functionality of the system’s components and ensure that everything is working as expected, even after a new line of code is added. It serves to prevent any code error. | This should be written at the start of the initial documentation phase. To get a clear idea of the possible problem that may occur to the application. After that, it should be written iteratively during each sprint, and it can be tested at the end of each sprint. | Tony | Tony |
| User acceptance test | To test the expected user outcomes and prevent any unexpected results to show on the system. | The test cases need to be written before testing. After each function is finished on the user interface, we conduct the test at the end of each sprint. | Tony | Edwin |

## Unit test

Here, we will be defining all the unit tests that need to be written and tested on the system. Please note that new unit tests may be added, modified, or deleted as the project progresses.

|  |  |
| --- | --- |
| **Method for getting phone number** | **Status (done or not yet)** |
| ShouldGetPersoonPhoneNumbersByTelType() | Not yet |
| ShouldGetPersoonPhoneNumbersByStartingDate() | Not yet |
| ShouldGiveAnExceptionWhenThereIsNoPhoneNumber() | Not yet |

|  |  |
| --- | --- |
| **Method for calling the video call** | **Status (done or not yet)** |
| ShouldInitializeVideoCall() | Not yet |

|  |  |
| --- | --- |
| **Method for ending video call** | **Status (done or not yet)** |
| ShouldEndVideoCall() | Not yet |

|  |  |
| --- | --- |
| **Method for getting email** | **Status (done or not yet)** |
| ShouldBeAbleToGetPersoonEmail() | Not yet |
| ShouldGiveAnExceptionWhenThereIsNoEmail() | Not yet |

|  |  |
| --- | --- |
| **Method for adding date and time of the log** | **Status (done or not yet)** |
| ShouldBeAbleToAddDateAndTimeOfLog() | Not yet |

|  |  |
| --- | --- |
| **Method for getting the date and time from the video call** | **Status (done or not yet)** |
| ShouldBeAbleToGetTheDateAndTimeOfVideoCall() | Not yet |

## User acceptance test

Here, reside all the user acceptance tests and will be tested once each function is complete at the end of each sprint. The written user acceptance tests will be stored here, and the test reports for the user acceptance tests will be located in the “Test Report” document. Please note that new user acceptance tests will be added, modified, or deleted as the project progresses.

### Prototype concept 1

***Created date: 27 Sep 23***

| ID | User story | Description | Expected result |
| --- | --- | --- | --- |
| 1 | US 1: As an SVb employee, I want to start the video call in the appointment agenda form, so that I don’t need to initiate the call in another window, or form. | This is to see if a from appears. | A form should appear |
| 2 | US 8: As an SVb employee, I want to see the retiree’s available mobile phone numbers on records, so that I can decide which number to video call. | This is to show the available mobile phone number of the retiree. | A form should appear with the available mobile phone number. |
| 3 | US 8 | This is to show that you can’t interact with other things outside the form. | You can’t interact with anything outside the form in the application. |
| 4 | US 9: As an SVb employee, I want to be notified that the retiree doesn’t have a phone number, so that I can ask the retiree’s phone number. | This is to show a message when the retiree doesn’t have a mobile phone number. | A message should appear stating that: “This retiree doesn’t have a mobile phone number in the records.” |
| 5 | US 9 | This is to show that you can’t interact with other things outside the form. | You can’t interact with anything outside the form in the application. |
| 6 | US 2: As an SVb employee, I want to make a video call using the PRAS desktop application, so that everything that needs to be done for the retirees is done on the desktop application. | This to see live feed of the video call. | I can see the live video call. |
| 7 | US 2 | This is to see if you can hear any sounds from the live feed. | I can hear the sound from the video call. |
| 8 | US 2 | This is to see if your camera is working. | I can see my camera on a small screen in the video call. |
| 9 | US 4: As an SVb employee, I want to end the video call on the video call form, so that I don’t need to end the video call on another form/ window. | This is to see if you can end the video call on the end video call icon. | The video call will end, and the form will close/ disappear. |
| 10 | US 4 | This is to see if closing the form would ends the video call. | The video call will end, and the form will close/ disappear. |

### Prototype concept 2

***Created date: 13 Dec 23***

| ID | User story | Description | Expected result |
| --- | --- | --- | --- |
| 1 | US 5: As an SVb employee, I want a confirmation box to appear when I end the video call, so that I don’t accidentally hit the end video call. | This is to see if a confirmation box appears when you press the close button on the video call form. | A confirmation box should appear stating that “Are you sure you want to end the call?” and you should also see a “Yes” and a “No” button. |
| 2 | US 5 | This is to see that the video call form should not close when pressing the “No” button on the confirmation box. | The confirmation box should disappear, and the video call form should still be there. |
| 3 | US 5 | This is to see that the video call form should closed when the “Yes” button is pressed. | The confirmation box and the video call form should disappear. |
| 3A | US 5 | This is to see if you get a message box when click the “Sign Out” button in the call, it has an arrow icon. | The message box should appear stating “Are you sure to exit this session?” and there is a check box for ending the meeting when exit and two buttons for end call and cancel. |
| 4 | US 6: As an SVb employee or a retiree, I can mute my microphone, so that the other person will not be bothered by the background noises. | This is to see if you can mute your microphone when the microphone icon is pressed in the video call form. | The microphone icon should give a visual indication that the microphone is muted, and the other participant can’t hear you. |
| 5 | US 7: As an SVb employee or a retiree, I can turn off my camera, so that I don’t show any personal stuff. | This is to see if you can turn off your video camera when the camera icon is pressed in the video call form. | The camera icon should give a visual indication that your camera is turned off and you can’t see yourself on your own display screen. |
| 6 | US 11: As an SVb employee, I want to get a confirmation box when I start a video call, so that I don’t accidentally video call the retiree. | This is to see if a confirmation box appears when the “Start Video Call” | A confirmation box should appear stating “Are you sure you want to make this video call?” and you should also see a “Yes” and “No” buttons. |
| 7 | US 11 | This is to see that the video call form should not appear after you pressed the “No” button on the confirmation box. | Confirmation box should disappear, and the video call form should not appear. |
| 8 | US 11 | This is to see if the video call form should appear when the “Yes” button is pressed on the confirmation box. | The confirmation box should disappear, and the video call form should appear. |
| 9 | US 12: As a SVb employee, I want to be notified that the email has been sent when pressing the “Start Video Call”, so that I know that the video call link has been sent to the retiree. | This is to see if the message box appears. | A message box should appear stating that “The email has been successfully sent.” |
| 10 | US 13: As a SVb employee, I want to be notified if the retiree has an email address when pressing the “Start Video Call”, so that I know that the video call link is sent to their email address. | This is to see if the message box appears when the retiree doesn’t have an email address. | A message box should appear stating that “This person doesn't have an email.” |
| 11 | US 14: As a SVb employee, I can admit a specific retiree to join the call, so that no other person can join the call. | This is to see who is waiting to join the call. | You’ll be able to see the participants’ name that wants to join the call. Either through a pop-up in the call or if you click the three dots and then click lobby, you should be able to see the participants’ name in the waiting list. |
| 12 | US 14 | This is to see if you can allow the participant to join the call. | The participant should be able to join the call when you click the “Allow” button. |
| 13 | US 14 | This is to see if you can deny the participant to join the call. | The participant should not be in the call when you click on the “Deny” button. |
| 14 | US 15: As a retiree, I can join the video call through the link provided via email, so that I don’t need to download any external apps on the mobile. | This is to see if you get an email with the invite link. | You should see an email with a link. |
| 15 | US 15 | This is to see if you can join the video call by clicking on the link through mobile. | You should be able to join the video call on the mobile via the link provided in the email. |
| 16 | US 15 | This is to see if you can leave the call on mobile | You should be able to leave the call on mobile. |

### Additional feature for concept 1

***Created date: 28 Dec 23***

| ID | User story | Description | Expected result |
| --- | --- | --- | --- |
| 1 | US 10: As an SVb employee, I want to be able to input the mobile number that is written on the description , so that I can make the video call to that mobile number. | This is to see if you can see the field and the text on the field. | You should be able to see a field and text on the field stating, “Please enter a country code and a mobile number”, when you press the “Start Video Call” button. |
| 2 | US 10 | This is to see what would happen if you press the “Video call” button when you didn’t input anything in the field. | A warning message should appear when you press the “Video call” button when you didn’t input anything on the field. |
| 3 | US 10 | This is to see if you can copy and paste the number or type the number manually. | You should be able to copy and paste the number from the description or enter it manually into the number field. |
| 4 | US 10 | This is to see if the number would correct itself in the number field when pressing the “Video call” button. | The number field should remove any symbols and spaces when you press the “Video call” button. |