

2023

Observe



OBSERVA

READY.SET.PROTECT.

CSED-2026

HCI Project

12/9/2023

Table of Contents

Team Members: (Team 5)	3
Analyzing User's Requirements and Task Description:	3
User Stories	3
Data Gathering	7
Surveys	7
Team Collaboration.....	Error! Bookmark not defined.
Prototyping	9
Wireframing.....	9
UI Design.....	11
Iterative Testing and Feedback	15
Project Management	15
Tools Used.....	15
Notion with Scrum Framework	15
Figma	15
GitHub	15
Development Phases	16
First Sprint - Research.....	16
Minimum Viable Product (MVP).....	16
Task Distribution and Epics.....	16
User Stories	16
Progress Tracking.....	17
Iterative Development.....	17
Collaborative Environment	17
Database Management	18
Description.....	18
Visual Representation.....	19
Request Cycle	19
Next Milestone	20

Visitor History 18

User Notifications 20

WebApplication UI 18

Optimize User Experience 20

Two-Way Communication..... 20

Team Members: (Team 5)

- | | |
|---------------------------------|----------|
| • Ahmed Mustafa Elmorsy Amer | 21010189 |
| • Ahmed Youssef Sobhy Elgoerany | 21010217 |
| • Moustafa Esam El-Sayed Amer | 21011364 |
| • Ebrahim Alaa Eldin Ebrahim | 21010017 |
| • Ahmed Ayman Ahmed Abdallah | 21010048 |
| • Ali Hassan Ali Mohamed | 21010837 |

Analyzing User's Requirements and Task Description:

User Stories

- ✚ I am Abdallah, I work till late hours, so I want to view the camera when I am away, so that I can see what/who is at my door at any time.

Tasks Included

- Receive the stream on the server.
- Stream Camera from bell app.
- Send the stream to the front end.
- Show the received stream to the user.

- ✚ I am Ahmed, I want to get notified with the name of the visitor, so that I know who is at the bell without opening the camera.

Tasks Included

- Facial Recognition of visitors.
- Database with saved visitors faces.
- Add database to model.
- Upload images of new visitor faces.
- View current visitor's name in dashboard.
- Send notifications through the web.

- ✚ I am Ayman, I work multiple jobs, so I am a bit forgetful. I want to change my email address and password because I forgot them.

Tasks Included

- Add settings page to edit the user info.
- Add API requests for updating user info.

- ✚ I am Said, I want to see who the last people who visited me are because I am sometimes busy and didn't see who was at my door.

Tasks Included

- Add visitors page to show the known visitors history.

✚ I am Mohammed, I don't want to enter my email and password every time I login because it is repetitive and tiring.

Tasks Included

- Generate an authentication token.
- Add remember me option.
- Save the authentication token to a cookie.

✚ I am Alaa, I want no one to be able to view my data as it invades my privacy.

Tasks Included

- Add authentication layer for each user for all requests.
- Create login and signup pages.
- Add login to bell app using api key.
- Scan API key using QR code.

✚ I am Mustafa, I am a 60-year-old senior. I want to have an easy way to access the camera, visitors, etc. because I am a bit tech illiterate.

Tasks Included

- Create a navigation bar for the users.
- User-manual for the website.

✚ I am Ebrahim, I am a person who gets confused and doesn't like complicated things. I want the application interface to be simple and beautiful.

Tasks Included

- Create wireframe designs for website pages.
- Implement the pages designs with HTML, CSS, JS.
- Implement the pages designs using VueJS.
- Test the UI and UX of the pages.

Functional Requirements

1. User Interface and User Experience (UI/UX):

- The application should have a simple and aesthetically pleasing interface.
- Wireframe designs for website pages should be created to plan the layout and structure.
- The implemented pages should adhere to the designed UI using HTML, CSS, JS, and VueJS.

- User testing should be conducted to ensure a positive and intuitive user experience.

2. **User Settings and Information Update:**

- The system should provide a settings page allowing users to update their email address and password.
- API requests for updating user information should be securely implemented.

3. **Visitors History:**

- The system should maintain a history of known visitors.
- Users should be able to view a list of the last people who visited their door.

4. **Facial Recognition and Visitor Identification:**

- The system should employ facial recognition technology to identify and label visitors.
- A database of saved visitors' faces should be maintained for accurate recognition.
- The application should tell the description of the visitor if they aren't saved on the system.
- Notifications containing the visitor's name should be sent to the user through the web interface.

5. **Emotion Recognition:**

- The application should have the capability to identify the emotion of the visitor.
- This feature can aid in assessing whether the visitor is distressed and may require assistance.

6. **Remote Voice Interaction:**

- Users, especially those facing health constraints or during quarantine, should have the ability to interact with visitors remotely using voice features.

7. **Authentication and Privacy:**

- Login and signup pages should be created with secure authentication mechanisms.
- User data should only be accessible after successful authentication.
- An authentication layer must be implemented for all user-related requests to protect privacy.
- An option for users to generate an authentication token and save it for future logins should be provided.

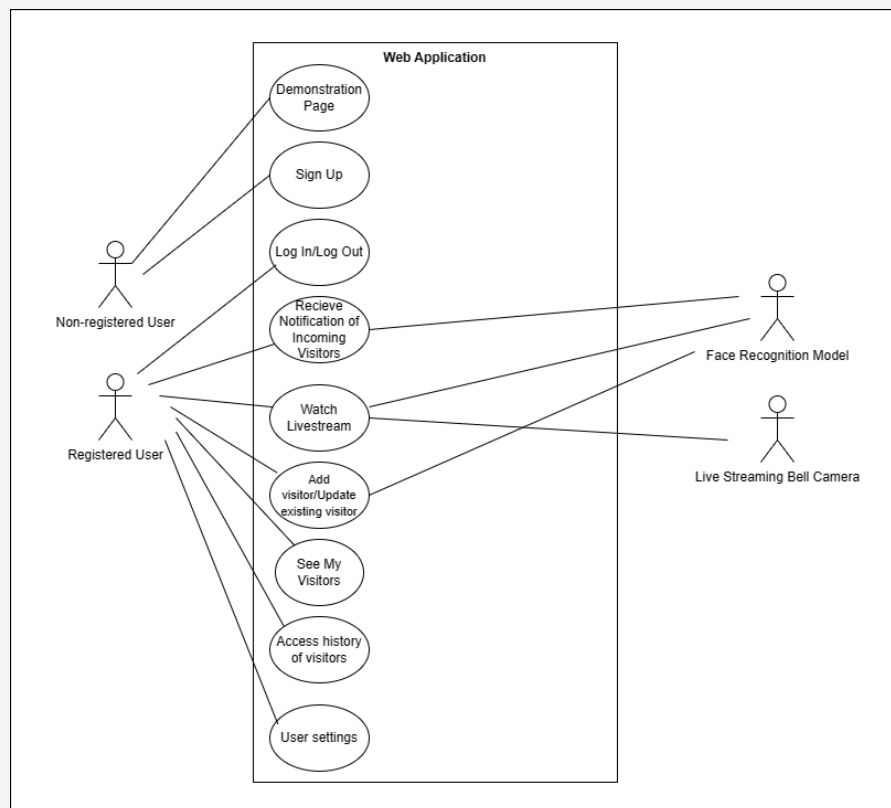
8. **Video Streaming and Camera Interaction:**

- The system must enable users to receive a live stream from the smart doorbell camera.
- The server should efficiently handle the streaming process.

Non-Functional Requirements:

1. **Security:**
 - All user data, including facial recognition data and emotion analysis, must be stored securely.
 - Encryption should be implemented for data in transit to prevent unauthorized access.
2. **Performance:**
 - The video streaming should have low latency to provide a real-time experience.
 - Response times for facial recognition and emotion analysis should be optimized.
3. **Usability:**
 - The UI should be designed with simplicity and ease of use in mind.
 - A user manual for the website should be provided to assist users, especially those who may be less tech-savvy.
4. **Reliability:**
 - The system should be highly reliable, ensuring consistent access to video streams and accurate visitor identification.

Use Cases



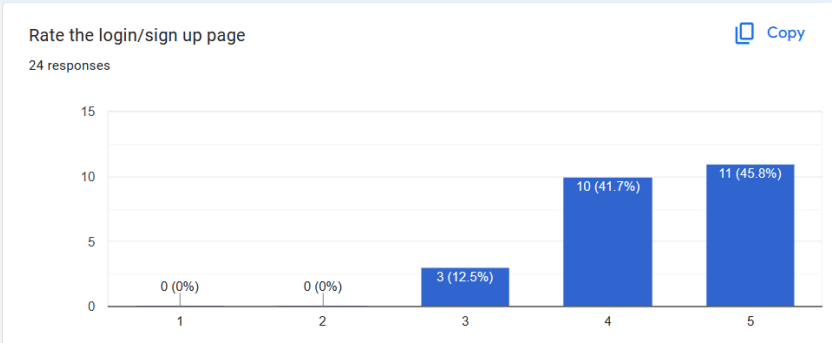
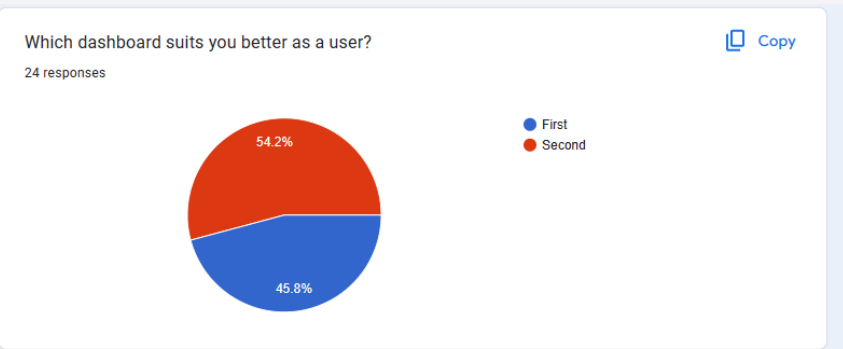
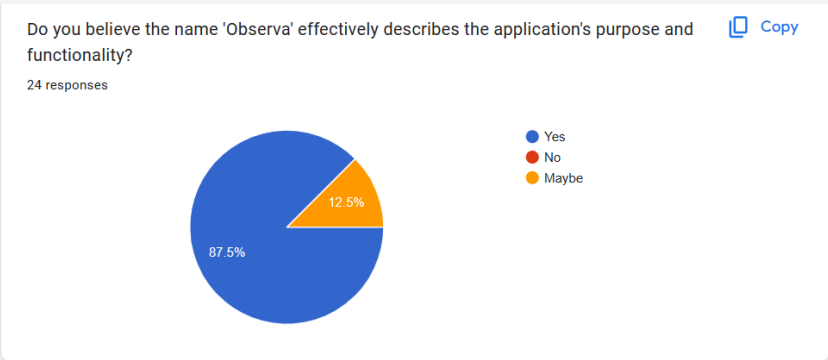
Use case diagram showing the user's interaction with the web application

Data Gathering

In the process of gathering data for our *Observa* Smart Doorbell System project, we employed a combination of surveys and collaborative teamwork. The data collection strategy involved two primary methods:

Surveys:

We conducted surveys to systematically gather insights from potential users, aiming to understand their preferences, expectations, and specific requirements regarding home security systems. The survey responses provided valuable qualitative and quantitative data, informing the design and functionality aspects of the *Observa* system.



Team Collaboration:

Internally, our team engaged in collaborative discussions and workshops to leverage the collective expertise and diverse perspectives within the group. Through brainstorming sessions and collaborative work, we were able to extract valuable insights, identify key challenges, and establish consensus on various design and functionality aspects of the smart doorbell system.

Interviews:

In the context of establishing requirements, it is equally important for development team members to meet stakeholders and for users to feel involved. This on its own may be sufficient motivation to arrange interviews. Therefore, we conducted a few interviews with potential users and testers. Here's how interviews benefited our project, specifically in adding more features and refining the UI for the Observa Smart Doorbell System:

I. User-Centric Insights:

Interviews provided direct access to the perspectives and experiences of potential users. By engaging with users, the team gained deep insights into their preferences, expectations, and pain points related to home security and smart doorbell systems. This helped us to see from the eyes of our users and escape the amazed look at our product.

II. Feature Enhancement:

Through user interviews, the team was able to identify user needs that may not have been apparent during the initial research phase. Users often provide nuanced feedback and suggest features that align with their daily routines and security concerns. This information was instrumental in adding more features to the Observa system, ensuring that it meets the diverse needs of its user base.

III. UI/UX Refinement:

Understanding how users interact with technology and what interfaces resonate with them is crucial for creating an intuitive and user-friendly UI. User interviews helped in uncovering

aspects of the UI that may have been unclear or confusing to potential users. This insight guided the refinement of the UI, making it more intuitive and aligned with user expectations.

IV. User Journey Mapping:

User interviews contribute to the creation of detailed user journey maps. Understanding the various touchpoints and interactions users have with the smart doorbell system helped in optimizing the UI for a seamless user experience throughout the entire interaction cycle, from initial setup to daily use.

V. User Empathy:

Interviews create a sense of empathy within the development team. Understanding the challenges and needs of users on a personal level fosters a deeper connection with the end-users. This emotional connection can drive the team's motivation to deliver a product that genuinely addresses user concerns and provides a positive experience.

VI. Validation of Design Choices:

Through interviews, the team was able to validate design choices and hypotheses. Direct feedback from users helped in confirming whether design decisions resonated with the target audience or if adjustments were necessary.

Exploring Similar Products:

This phase proved to be highly significant as it granted us access to user feedback on comparable products. Leveraging this information, we enhanced certain features, aiming to elevate the overall user experience and iteratively improve upon existing solutions.

Prototyping

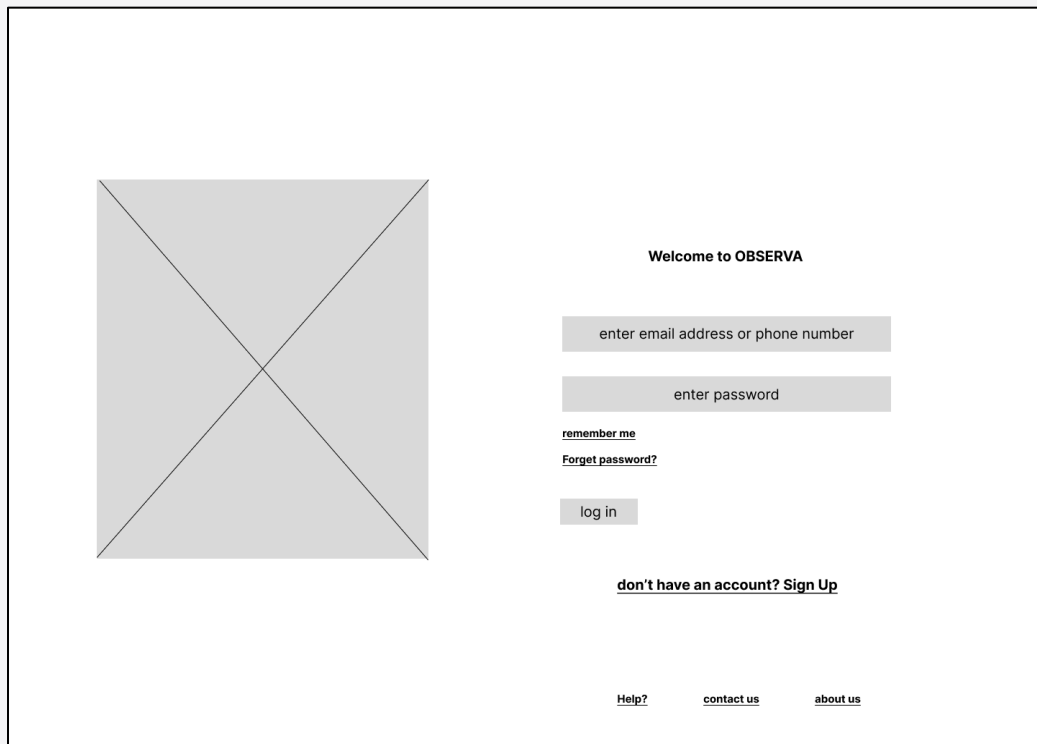
Regarding the UI/UX design process, we first designed the wireframes, then came up with UI Design, then implemented some unfinalized front-end codes and got feedback from potential users, so we will work on the enhancements before the next milestone.

Wireframing

The initial phase involved the creation of wireframes. Wireframes are low-fidelity, basic representations of the user interface layout and structure. They serve as a blueprint for the design, focusing on functionality and content placement without delving into visual details.

Login Page

The Log-in page where the user enters his e-mail and password to access his data.



A login page mockup for 'OBSERVA'. On the left is a large grey square with a black 'X' across it. To the right, the text 'Welcome to OBSERVA' is centered. Below it are two input fields: 'enter email address or phone number' and 'enter password'. Under the password field are links for 'remember me' and 'Forgot password?'. A 'log in' button is below these. At the bottom, a link says 'don't have an account? Sign Up'. The footer contains links for 'Help?', 'contact us', and 'about us'.

Welcome to **OBSERVA**

enter email address or phone number

enter password

[remember me](#)

[Forgot password?](#)

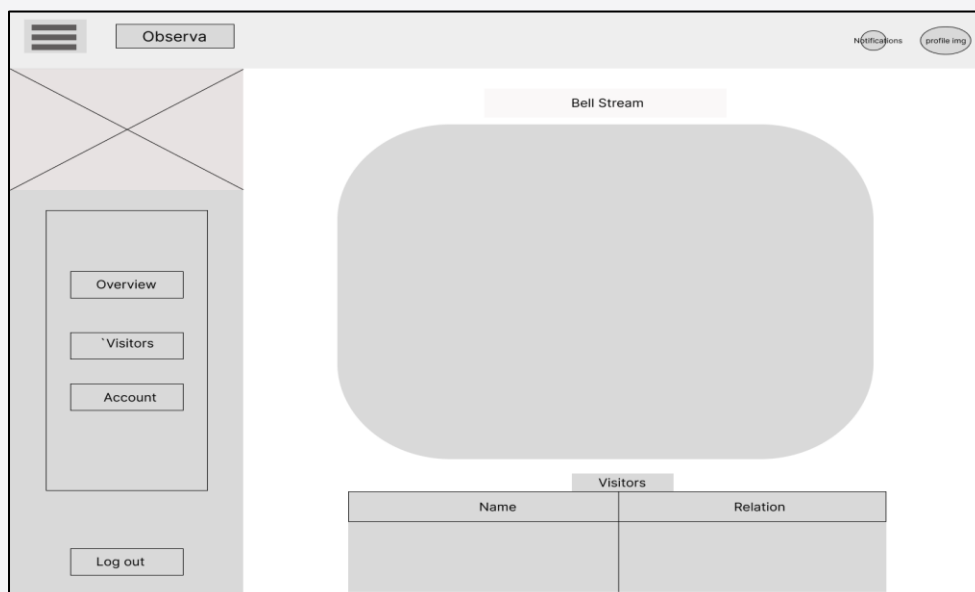
log in

[don't have an account? Sign Up](#)

[Help?](#) [contact us](#) [about us](#)

Dashboard

The main page, where the stream is viewed, and the user gets an overview of his door-front visitors.



A dashboard mockup for 'Observa'. The top bar has a hamburger menu, the 'Observa' logo, and 'Notifications' and 'profile img' icons. The left sidebar contains a large 'X' placeholder and a menu with 'Overview', 'Visitors', 'Account', and 'Log out'. The main area is titled 'Bell Stream' and contains a large rounded rectangle placeholder. Below this is a 'Visitors' table with columns 'Name' and 'Relation'.

Observa

Notifications profile img

Bell Stream

Overview

Visitors

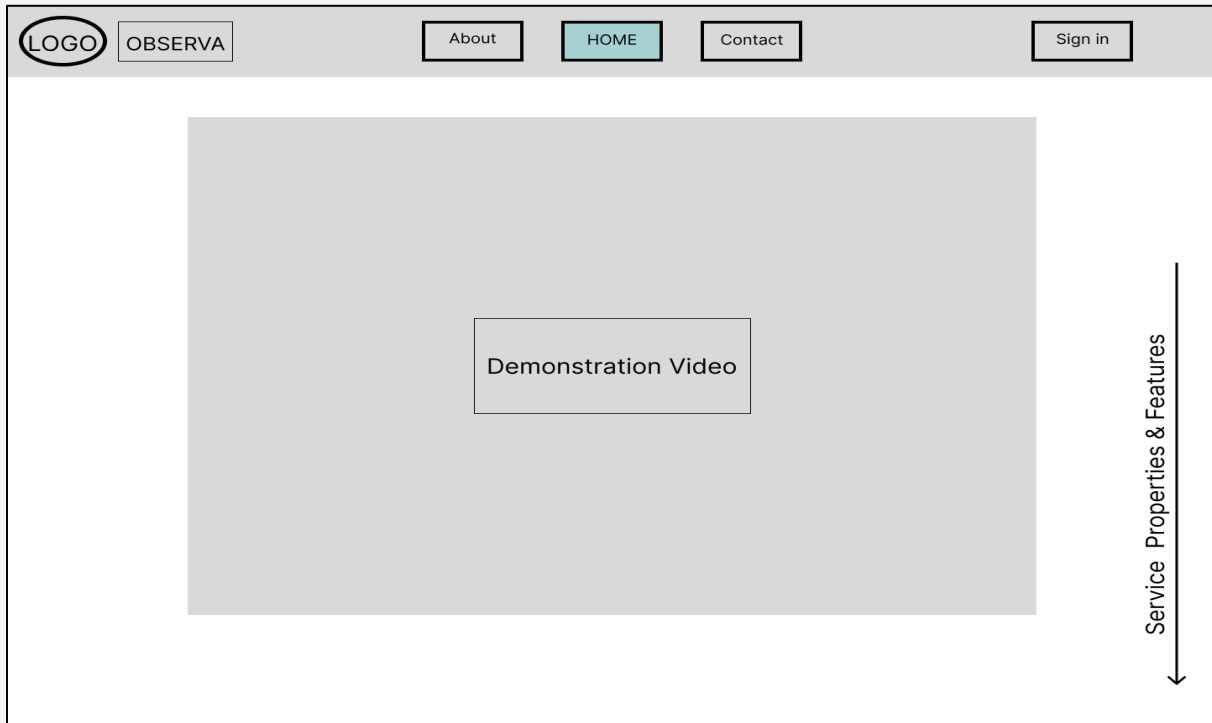
Account

Log out

Visitors	
Name	Relation

Home page

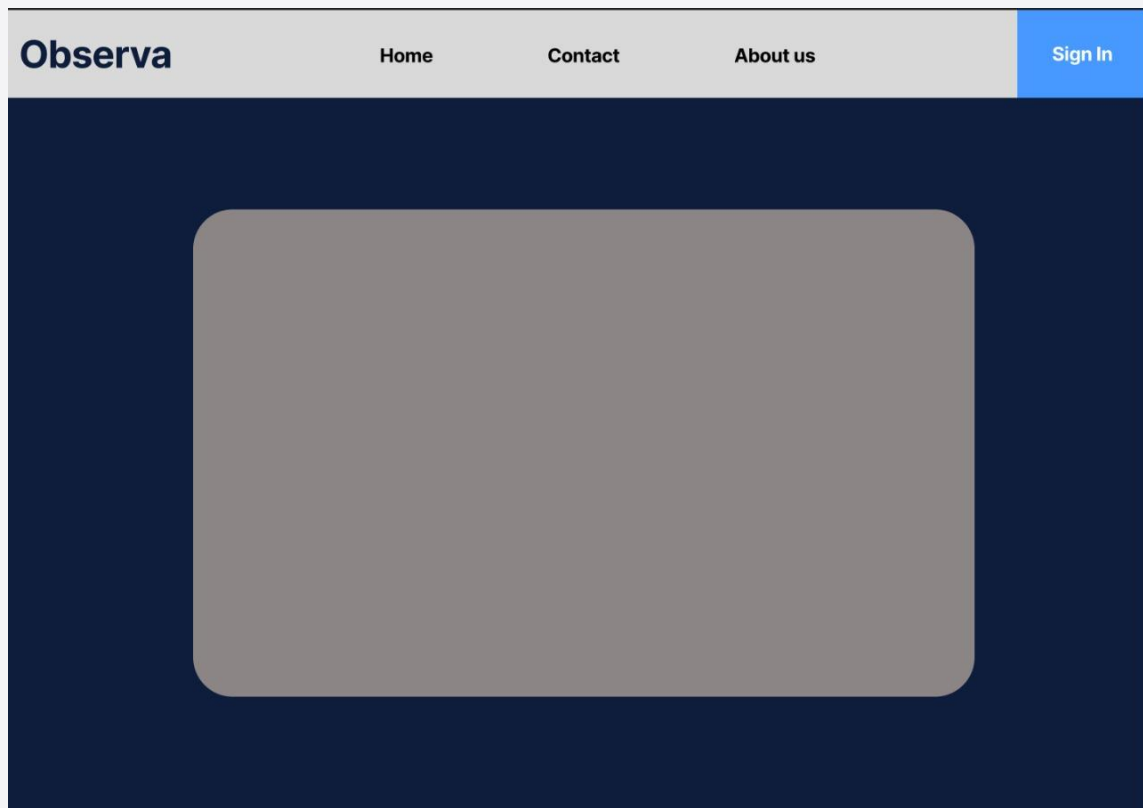
Shows a demonstration video of how to handle the stream and navigate the dashboard.



UI Design

Building upon the wireframes, the next step was the UI design. This phase focused on adding visual elements to the wireframes, such as color schemes, typography, and imagery. The goal was to create a visually appealing and cohesive design that aligns with the brand identity and user preferences.

Home page




Visitors Page



Log In

Welcome TO OBSERVA



Username


Password

Login

[don't have an account?](#)

Sign Up

Welcome TO OBSERVA



Username

Email Address

Password

Birthday

City

Full Name

Phone Number

Gender

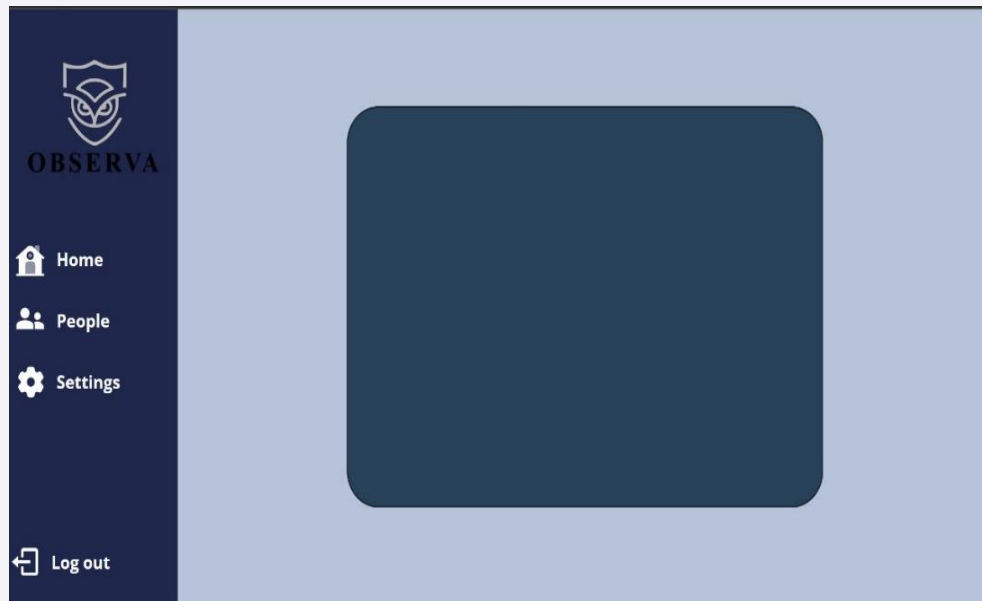
Address

Sign Up

[already have an account?](#)

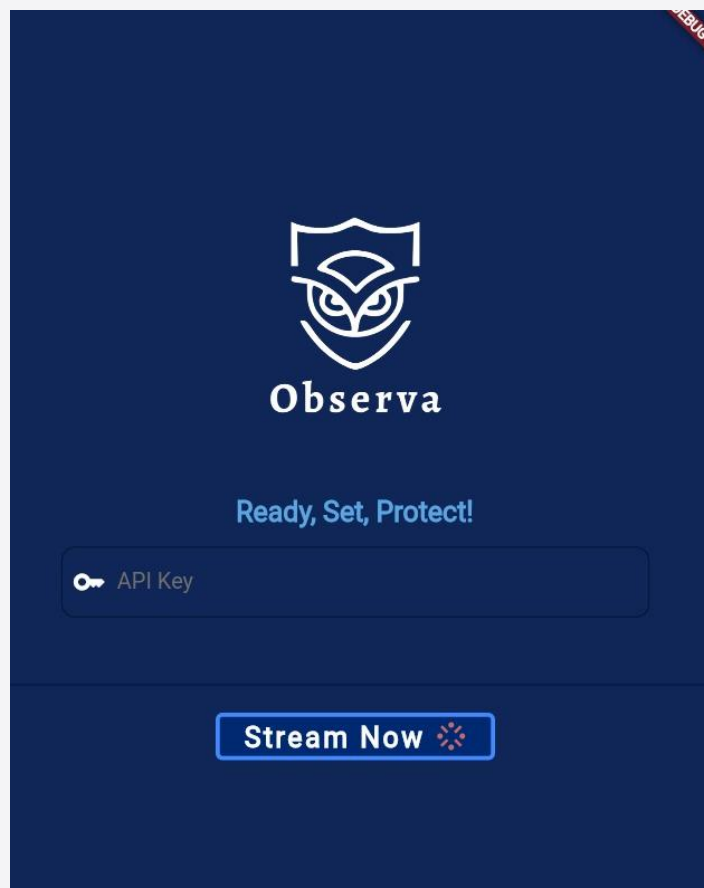
Observa

Dashboard



Mobile Bell Application

- Using Flutter framework



Iterative Testing and Feedback

The UI design needed a lot of testing and feedback from potential users, so this is not the final design, and we plan to employ additional data gathering techniques to fulfill all requirements that satisfy user's experience and expectations.

Project Management

Tools Used

Notion with Scrum Framework

- Notion served as the central project management hub, leveraging the Scrum framework. Scrum principles provided a flexible and iterative approach, enabling the team to adapt to changing requirements and continuously improve the development process.
- [Notion Workspace](#)

Figma

- Figma played a crucial role in the UI/UX design process. It allowed the team to collaboratively create and iterate on design prototypes. Figma's cloud-based platform facilitated real-time collaboration, streamlining communication among team members involved in the design phase.
- [Figma Workspace](#)

GitHub

- GitHub served as the version control system for the project, enabling collaborative development. It allowed the team to manage and track changes to the codebase, facilitating seamless collaboration among front-end and back-end developers.
- [GitHub Repository Link](#)

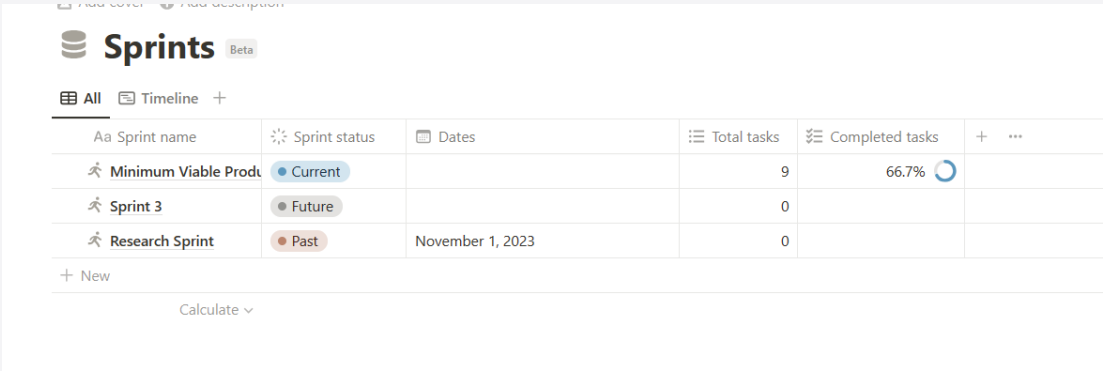
Development Phases

First Sprint - Research


- The project kicked off with a research sprint, focusing on understanding user needs, market trends, and technological requirements. This information guided the subsequent phases of development, ensuring a user-centric and technically feasible solution.

Minimum Viable Product (MVP)

- The second sprint focuses on developing the Minimum Viable Product. This involved implementing the core functionalities necessary for the system to be functional and valuable to users. The MVP approach allows for quicker delivery of a basic yet functional version for testing and validation.



The screenshot shows the 'Sprints' application interface. At the top, there's a header with the 'Sprints' logo and a 'Beta' tag. Below the header, there's a navigation bar with 'All' and 'Timeline' tabs. The main content is a table with columns: 'Aa Sprint name', '⚙️ Sprint status', '📅 Dates', '⋮ Total tasks', '📊 Completed tasks', and '+ ...'. The table lists three sprints: 'Minimum Viable Product' (Current status, 9 total tasks, 66.7% completed), 'Sprint 3' (Future status, 0 total tasks), and 'Research Sprint' (Past status, 0 total tasks, dated November 1, 2023). At the bottom, there's a '+ New' button and a 'Calculate' button.

Aa Sprint name	⚙️ Sprint status	📅 Dates	⋮ Total tasks	📊 Completed tasks	+ ...
🚀 Minimum Viable Product	● Current		9	66.7% 	
🚀 Sprint 3	● Future		0		
🚀 Research Sprint	● Past	November 1, 2023	0		

Task Distribution and Epics

- Tasks are distributed across different epics to streamline development efforts. Epics were categorized into front-end and back-end, aligning with the division of responsibilities among team members. This approach facilitated focused development and allowed team members to work efficiently within their respective domains of expertise.

User Stories

- User stories are employed to define and communicate specific features and functionalities from the end-user's perspective. This user-centric approach ensures that development

efforts are aligned with the actual needs and expectations of the target users.

№ Task ID	Aa Task name	Status	Assignee	Priority	Tags	Sprint	Epic
OHT-37	Send notifications through web	Not started					Front-end Back-end
OHT-36	Scan api key using qr code	Not started					Bell App
OHT-35	Add login to bell app using api k	Not started					Bell App
OHT-33	Implement the pages designs usi	Not started					Front-end
OHT-32	Test the UI and UX of the pages	Not started					Front-end
OHT-30	User-manual for the website	Not started					Front-end
OHT-28	Create login and signup pages	Not started					Front-end
OHT-27	Add remember me option	Not started					Front-end
OHT-26	Save the authentication token to	Not started					Front-end
OHT-24	Add api request to get visitors so	Not started					Back-end
OHT-23	Add authentication layer for each	Not started					Back-end
OHT-22	Add visitors page to show the kn	Not started					Front-end
OHT-21	Add api requests for updating usi	Not started					Back-end
OHT-20	Generate an authentication token	Not started					Back-end
OHT-19	View current visitor's name in das	Not started					Front-end Back-end
OHT-18	Add settings page to edit the use	Not started					Front-end
OHT-16	Upload images of new visitor face	Not started					Front-end Back-end
OHT-11	Stream Camera from bell app	Not started					Bell App

Progress Tracking

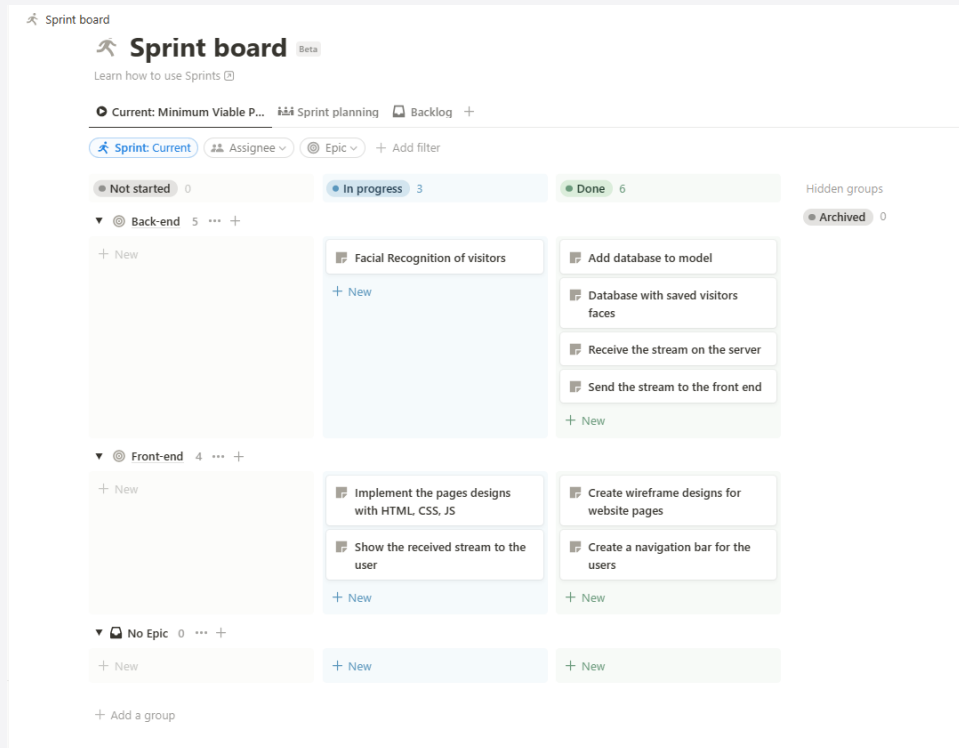
- Progress was tracked using a Kanban board. This visual representation allowed the team to monitor the status of tasks immediately, from "To Do" to "In Progress" and finally "Done." The Kanban board facilitated transparency and enabled quick identification of bottlenecks or areas that required additional attention.

Iterative Development

- The use of Scrum principles, user stories, and iterative development cycles allowed the team to adapt to changing requirements and continuously improve the product. Regular sprint reviews and retrospectives ensured that the team learned from each iteration, fostering a culture of continuous improvement.

Collaborative Environment

- The combination of tools like Notion, Figma, and GitHub contributed to a collaborative and efficient work environment. Real-time collaboration in design, version control in development, and organized project management in Notion collectively supported the team throughout the project lifecycle.



Visitor History

Users can now seamlessly review past interactions, providing an extra layer of security. The application ensures that each visit is logged and stored, allowing users to access and manage their historical data effortlessly, by storing videos and visitors' history.

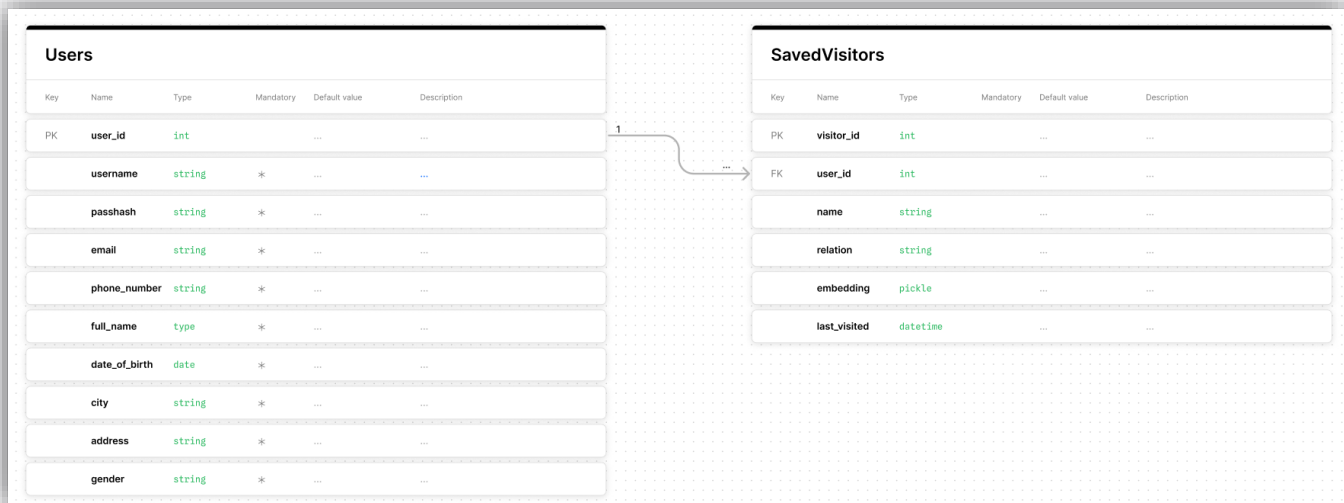
Web Application UI

The UI now combines aesthetics with functionality, offering users an intuitive and visually pleasing experience. The thoughtful arrangement of elements ensures easy navigation and access to the application's enhanced features, creating a seamless and enjoyable interaction.

Database Management

Description

As shown in the diagram below, the database consists mainly of two tables one for the users signed up to the platform and the other is for their visitors. Each table has its own public key with additional one to many relationships between the Users' dataset and Visitors' dataset.



Visual Representation

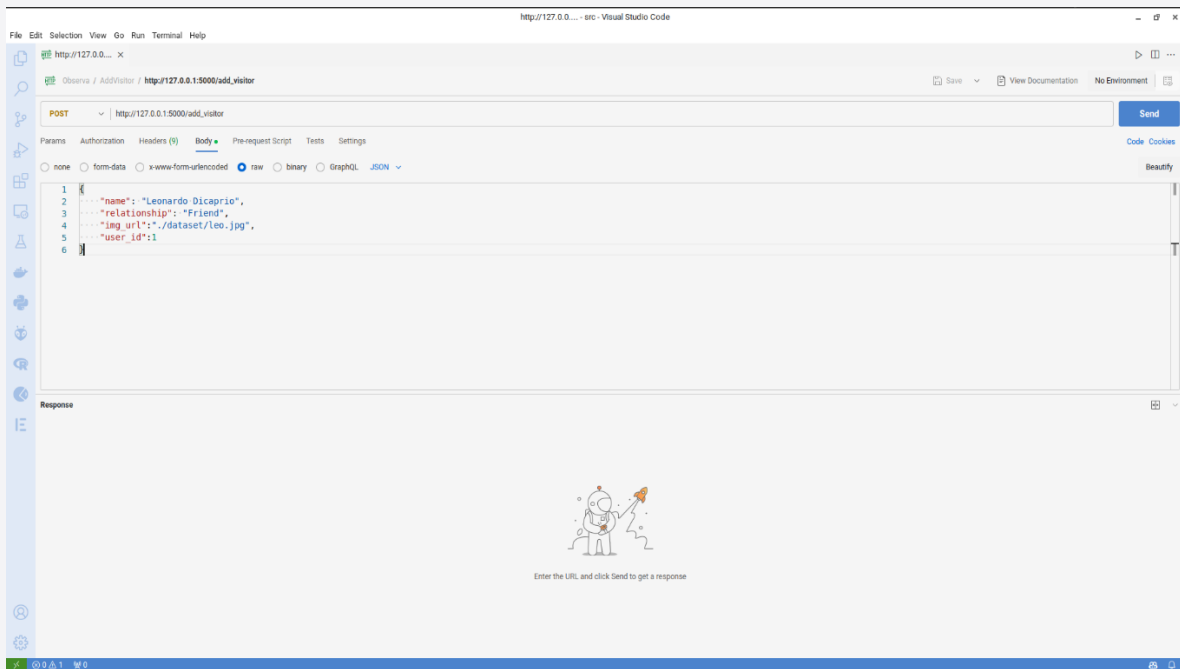
We use graphical interface to help us navigate the database, graphically through **pgadmin** platform.

An example is shown below of our visual representation of the dataset:

id	username	passhash	email	phone_number	full_name	date_of_birth	city	country	gender
1	ayman	password	ayman@example.org	3781434396627	Ahmed Ayman	1952-11-30	North Jenniferborough	Belize	male
2	lauren79	password	vruiz@example.net	2890290316586	Zachary Dunn	1970-04-30	Pittmanville	San Marino	male
3	zward	password	bmartin@example.com	4849363110741	Charles Perez	1911-10-25	South John	Chile	male
4	zsteale	password	ariana60@example.net	2326962092807	Peter Moclure	1915-11-19	North Margaret	Finland	male
5	jacob06	password	debra29@example.org	4366883757875	James Orozco	1969-08-01	Lake Ashley	Rwanda	male
6	hhill	password	johnwolfe@example.org	4991631612196	William Guerrero	1962-10-22	Bishopville	Moldova	male
7	kingkevin	password	wagnerjoseph@example.com	5103519679961	Anthony Johnson	1969-09-11	Thompsonstown	Palestinian Territory	male
8	javier85	password	thompsonmatthew@example...	8233142149705	Cody Whitehead	1990-04-05	Rossshire	Swaziland	male
9	jbass	password	jbryant@example.com	1165106192070	Tony Brown	1941-01-24	Port William	Benin	male
10	vlopez	password	mhill@example.com	5267035835680	Allen Williams	1997-05-27	Maynardmouth	Trinidad and Tobago	male
11	samueltaylor	password	burnsgina@example.com	80803223643517	Douglas Griffin	1922-03-27	New Jamie	Argentina	male

Request Cycle

We use a graphical interface to handle the request cycle like postman.



Next Milestone

In the following phase of our smart-door bell application development, we are going to introduce some enhancement features, that improves the functionality of the project in addition to the user experience.

We will implement the following features:

User Notifications

The application will be enhanced with a notifications system, which notifies users and keeps them up to date with their visitors whenever someone arrives at their doorstep.

Optimize User Experience

Conducting interviews with potential users will take place, to benefit from their experience to optimize the user experience to what relieves them. Collecting these data will help us improving the UI.

Two-Way Communication

Adding another layer of interactivity, the Smart Doorbell Application now features two-way communication with visitors. Users can seamlessly engage in conversations, enhancing security and convenience. This bidirectional communication capability not only serves as a deterrent for potential intruders but also facilitates smooth communication with expected visitors.

Full UI Implementation

Apply the designed UI for the web application, ensuring that it aligns with the wireframe designs and provides an intuitive and aesthetically pleasing interface.

Functionality Testing

Thoroughly test the implemented visitor history and video storage features to ensure their accuracy, reliability, and seamless integration into the system.

Key Activities:

- Conduct thorough testing of visitor history and video storage functionalities.
 - Integrate user feedback from interviews to optimize the UX.
 - Activate and test user notification features.
 - Implement and test two-way communication with visitors.
 - Apply the finalized UI design to the web application.
- This phase is critical for ensuring the functionality of Sprint 2 tasks and preparing for the subsequent milestone, Sprint 3 Finalization. By rigorously testing and optimizing the features, we aim to deliver a robust and user-friendly Observe Smart Doorbell System that meets the project's objectives and exceeds user expectations.