Sprint 1
Alzheimer's Facial Recognition

Gabriel Thomas, Irald kollcinaku, Robert Turczyn, Tyler Schauss, Jacob Carpenter, Zilin Li

Vision Statement

Our vision is to build a bridge between humans and machines through software technology. Through a skilled team of developers, we acknowledge the need for a change in medical conditions that affect not only the patients but also their loved ones. Our team strives to build a superior app to empower Alzheimer's patients recognize people in their daily routine.

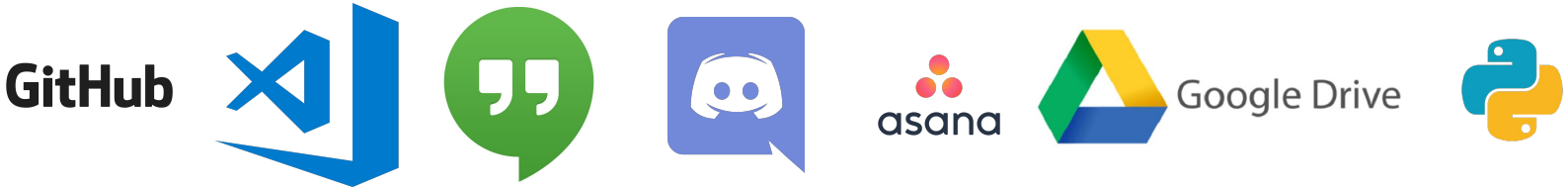
Incorporating facial recognition, tracking and reminders is our main focus in constructing this helpful app.

Project Objectives

- Setup a Firebase hosted server
  - Will host NoSQL Database
    - Will hold data for client's contacts
    - Will hold images of client's contacts
- Use the device camera to achieve facial recognition
- Utilize text to speech to upon facial recognition
  - Will tell client the recognized contact's information
- Allow clients to upload images and data of contacts
- Allow caretakers to upload images and informations that is connected to a client's account
- Use PWA to create a web portal as well as Android and iOS apps
- Possible implement a location setting that can show the location of a client to a caretaker.

Devops Toolchain

Tools



Platforms
Libraries
Languages

Functional Requirements

- REQ1- The system shall use the device camera to run Open CV code
- REQ2- The system shall recognize faces seen by the device camera
- REQ3- The system shall use TTS (text-to-speech) via device audio output to say the name of the person recognized
- REQ4- The system shall have a web portal
- REQ5- The system shall have an app version that runs on Android and iOS
- REQ6- The system shall have a caretaker portal that can connect to a clients account
- REQ7- The system shall have a way for the client to add many pictures to a Database for Open CV to recognize faces.

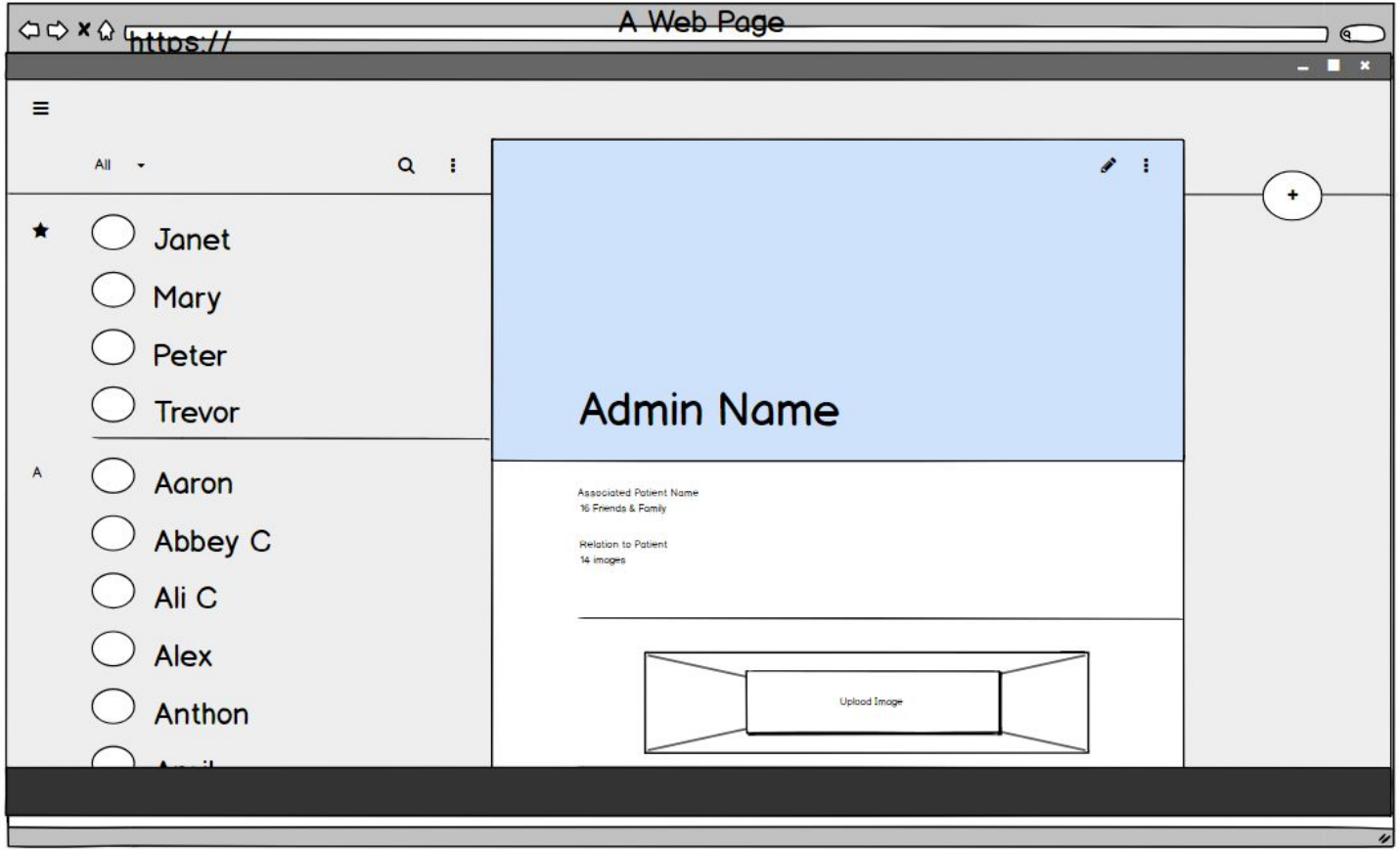
Functional Requirements Cont...

- REQ8- The caretaker portal shall have an option to enter pictures for the client's app to recognize faces
- REQ9- The system shall have a feature that turns the device's screen black while in the clients pocket to preserve battery life.
- REQ10- The system shall alert the client upon facial recognition
- REQ11- The system should have a location setting that can be turned on and off to allow a caretaker to see where the client is

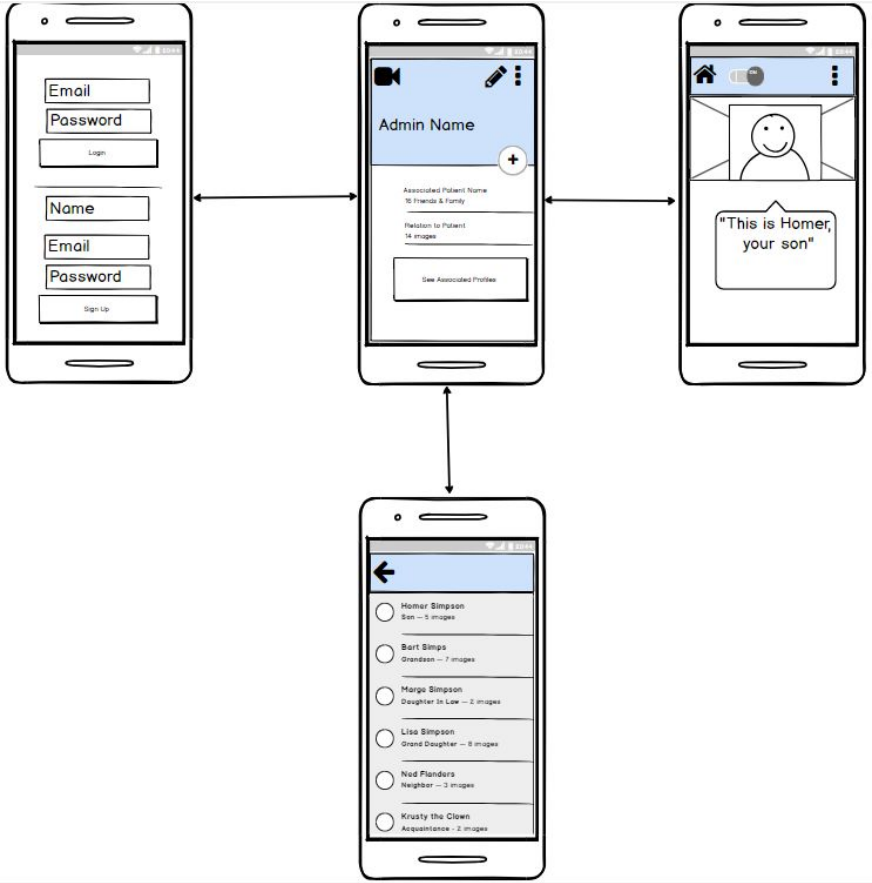
Non-Functional Requirements

- NFR1- The system shall have a NoSQL Database to store images
- NFR2- The system shall have a NoSQL Database to store text information, such as, name, relation, etc...
- NFR3- The system shall be able to send a facial recognition message with TTS data in less than a minute
- NFR4- The system shall have a backend server setup to host necessary applications

Views


A Web Page
https://
All
Janet
Mary
Peter
Trevor
A
Aaron
Abbey C
Ali C
Alex
Anthon
Admin Name
Associated Patient Name
16 Friends & Family
Relation to Patient
14 images
Upload Image

Views


Email
Password
Login
Name
Email
Password
Sign Up
Admin Name
Associated Patient Name
16 Friends & Family
Relation to Patient
14 images
See Associated Profiles
"This is Homer, your son"
Homer Simpson
Son — 5 images
Bart Simps
Grandson — 7 images
Marge Simpson
Daughter In Law — 2 images
Lisa Simpson
Grand Daughter — 8 images
Ned Flanders
Neighbor — 3 images
Krusty the Clown
Acquaintance - 2 images

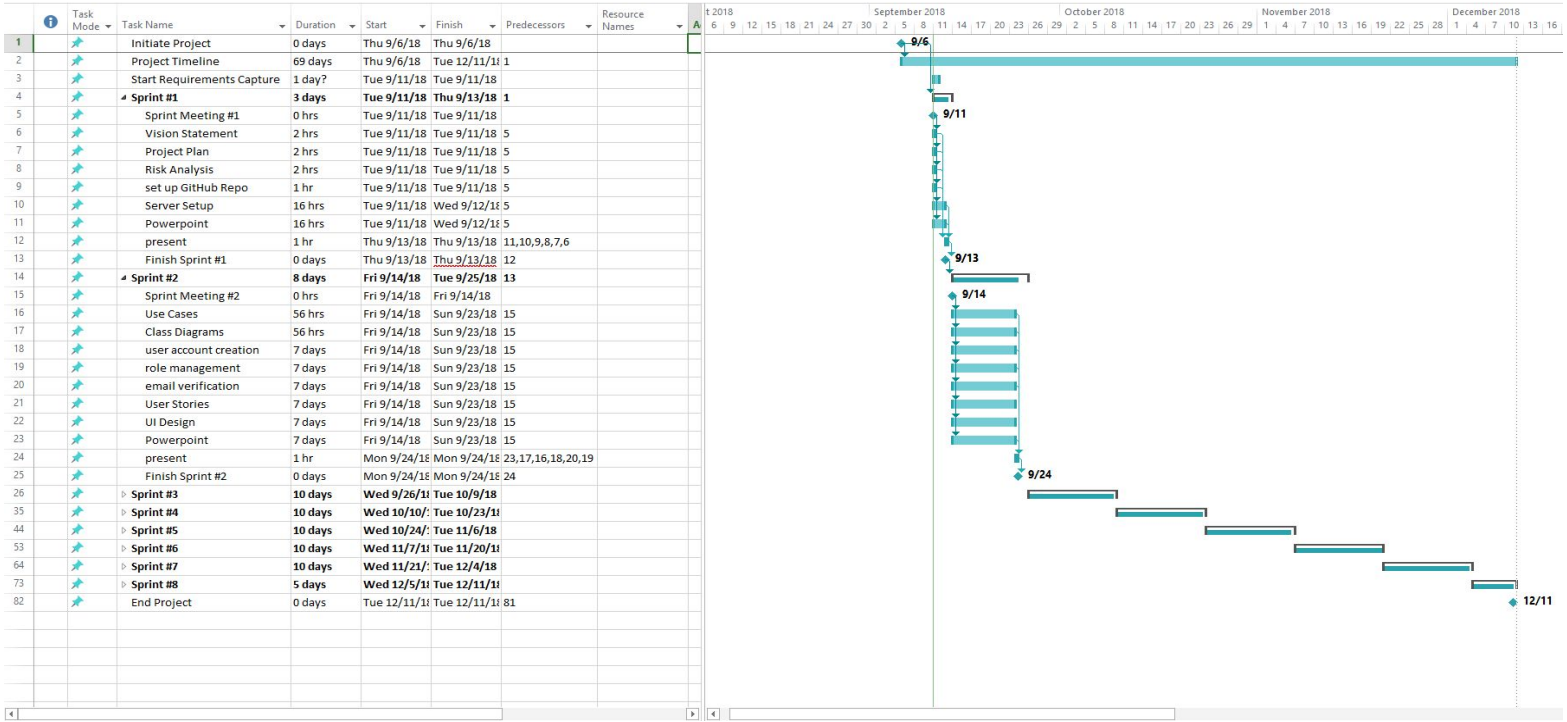
Possible Risks

- Time
- Most of us unfamiliar with PWA apps
- None of us have used this tech stack before
- Performance
  - Battery Life of Phone
  - How long it takes to recognize a face
  - How lighting conditions will affect recognition

Mitigation Plan

- Schedule one in-person meeting and at least one online meeting a week to track project to solve issues and ensure we finish related works within every phase.
- We may use another facial detection and recognition android sdk provided by Arcsoft if we cannot implement OpenCV.
- Keep screen dimming until client start to use this application.
- We will constantly update the risk mitigation plan in the following development life cycle.

Project Plan



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Initiate Project	0 days	Thu 9/6/18	Thu 9/6/18		
2		Project Timeline	69 days	Thu 9/6/18	Tue 12/11/18	1	
3		Start Requirements Capture	1 day?	Tue 9/11/18	Tue 9/11/18		
4		**Sprint #1**	**3 days**	**Tue 9/11/18**	**Thu 9/13/18**	**1**	
5		Sprint Meeting #1	0 hrs	Tue 9/11/18	Tue 9/11/18		
6		Vision Statement	2 hrs	Tue 9/11/18	Tue 9/11/18	5	
7		Project Plan	2 hrs	Tue 9/11/18	Tue 9/11/18	5	
8		Risk Analysis	2 hrs	Tue 9/11/18	Tue 9/11/18	5	
9		set up GitHub Repo	1 hr	Tue 9/11/18	Tue 9/11/18	5	
10		Server Setup	16 hrs	Tue 9/11/18	Wed 9/12/18	5	
11		Powerpoint	16 hrs	Tue 9/11/18	Wed 9/12/18	5	
12		present	1 hr	Thu 9/13/18	Thu 9/13/18	11,10,9,8,7,6	
13		Finish Sprint #1	0 days	Thu 9/13/18	Thu 9/13/18	12	
14		**Sprint #2**	**8 days**	**Fri 9/14/18**	**Tue 9/25/18**	**13**	
15		Sprint Meeting #2	0 hrs	Fri 9/14/18	Fri 9/14/18		
16		Use Cases	56 hrs	Fri 9/14/18	Sun 9/23/18	15	
17		Class Diagrams	56 hrs	Fri 9/14/18	Sun 9/23/18	15	
18		user account creation	7 days	Fri 9/14/18	Sun 9/23/18	15	
19		role management	7 days	Fri 9/14/18	Sun 9/23/18	15	
20		email verification	7 days	Fri 9/14/18	Sun 9/23/18	15	
21		User Stories	7 days	Fri 9/14/18	Sun 9/23/18	15	
22		UI Design	7 days	Fri 9/14/18	Sun 9/23/18	15	
23		Powerpoint	7 days	Fri 9/14/18	Sun 9/23/18	15	
24		present	1 hr	Mon 9/24/18	Mon 9/24/18	23,17,16,18,20,19	
25		Finish Sprint #2	0 days	Mon 9/24/18	Mon 9/24/18	24	
26		**Sprint #3**	**10 days**	**Wed 9/26/18**	**Tue 10/9/18**		
35		**Sprint #4**	**10 days**	**Wed 10/10/18**	**Tue 10/23/18**		
44		**Sprint #5**	**10 days**	**Wed 10/24/18**	**Tue 11/6/18**		
53		**Sprint #6**	**10 days**	**Wed 11/7/18**	**Tue 11/20/18**		
64		**Sprint #7**	**10 days**	**Wed 11/21/18**	**Tue 12/4/18**		
73		**Sprint #8**	**5 days**	**Wed 12/5/18**	**Tue 12/11/18**		
82		End Project	0 days	Tue 12/11/18	Tue 12/11/18	81	