EECS 473 Project Proposal:
OmniView, The Facial Recognition Glasses

Team Members:
Loren Wang, wloren
John Connolly, johnconn
Tzu-Fei Yu, tzufeiy
Garrison Bellack, gbellack
Zheng Hao Tan (Bobby), tanzhao

Executive Summary

Since the beginning of human interaction, names have played a crucial role. Names serve as an identifier for a person, and without them people have no identity. In our modern society dominated by social media, interactions have become more frequent but less substantial, which has sparked a phenomenon where it has become harder to remember people's names. We have decided to address this issue by applying it alongside the recent surge in popularity of smart, wearable tech. We aim to create a pair of smart glasses with facial recognition capabilities, that can serve as an aid to the user in identifying a person and their name. Using a pre-existing database of face-name pairs, when you see a person, the glasses will recognize them even if you don't, and display their name for you. This will help you to remember the person and boost social interaction. This implementation also provides us a platform to further expand upon, such as adding phone or social media integration for other notifications, or sending additional information about the person you are looking at into your view.

High Level Description

One of the most important things in the social world is remembering people's names. To quote Dale Carnegie, "A person's name is to him or her the sweetest and most important sound in any language". However, too often we can't remember the names of people we meet, even after multiple interactions.

Our vision is to create a product, the OmniView, that aids people in remembering names. We believe this product could see widespread casual use, but is specifically useful for people with a bad memory or those entering a large/new social environment such as school, work, or a party. We believe it will be used in these situations because the price of not remembering someone's name can be quite high. Especially in business, where having positive work relationships is critical, a lapse in social graces could mean a very real financial setback.

The requirements for this product are not that stringent. It obviously needs to be safe and manufacturable. Reliability is interesting; failing to know someone's name is not horrible as often using their name can be avoided, but calling someone the wrong name, a false positive, is extremely bad. The price should be relatively low as this product is not something anyone needs, but more of a want. It should also not be apparent that you are using anything other than your memory to remember a person's name.

One solution would be an app that ties in with Facebook, quizzing you on people's names. Whenever you are going to a Facebook event, it uses the guest list to exercise your memory with pictures of the attendees. There are many advantages to this idea. It is completely safe and manufacturable. The cost is extremely low; potentially one dollar on the app store. However, there are a few downsides. The first is that it would only help remember people's names in the context of a Facebook event. The second is that it requires the premeditated action of quizzing yourself on people's names, and even then you may not remember their name. The last is minor, but it may actually increase the rate of false positives by getting some names from the quiz stuck in your head. Overall, while this helps, it doesn't solve the problem.

The next solution we came up with is an earpiece that recognized a person's name via their voice. A microphone would record audio from a person introducing themselves and send the audio to a server where the person's name and unique voice characteristics are extracted. The next time you hear that person talking, a small speaker in your ear would tell you their name. The pros of this idea are that it is fairly obscure. Having what would look like a hearing aid attached to your ear is not very noticeable. Even if it is noticed, its use is not apparent. The cons are that this product is more expensive. Another issue is that for you to get the person's name, they have to be speaking with you already, meaning you can't go strike up a conversation with that person's name. The final issue is that it is very hard to uniquely identify a person based on audio; noise, multiple people speaking, and lack of defining audio characteristics would all lead to a higher rate of failure.

We proposed a real-time solution, the OmniView, which is a pair of glasses with a camera that can instantly display the name of the person you are currently at. Using an existing database of face-name pairs such as a school registry, the glasses could take a picture of the person you are looking at and send the data to a server where the person's unique facial structure is analyzed. The server would then return the name of the matching entry, showing the name of the person in front of you on an display in the corner of your vision. A new entry could also be dynamically created by taking a picture and sending that information to the server along with accompanying audio of their introduction or only sending the picture and adding their name to the server later. The pros are that facial recognition have the highest rate of success and the lowest rate of false positives. In addition, this could be used right before the interaction (looking across the room) to get the person's name. The cons are that it is very noticeable you are wearing a device. It is also an expensive solution.

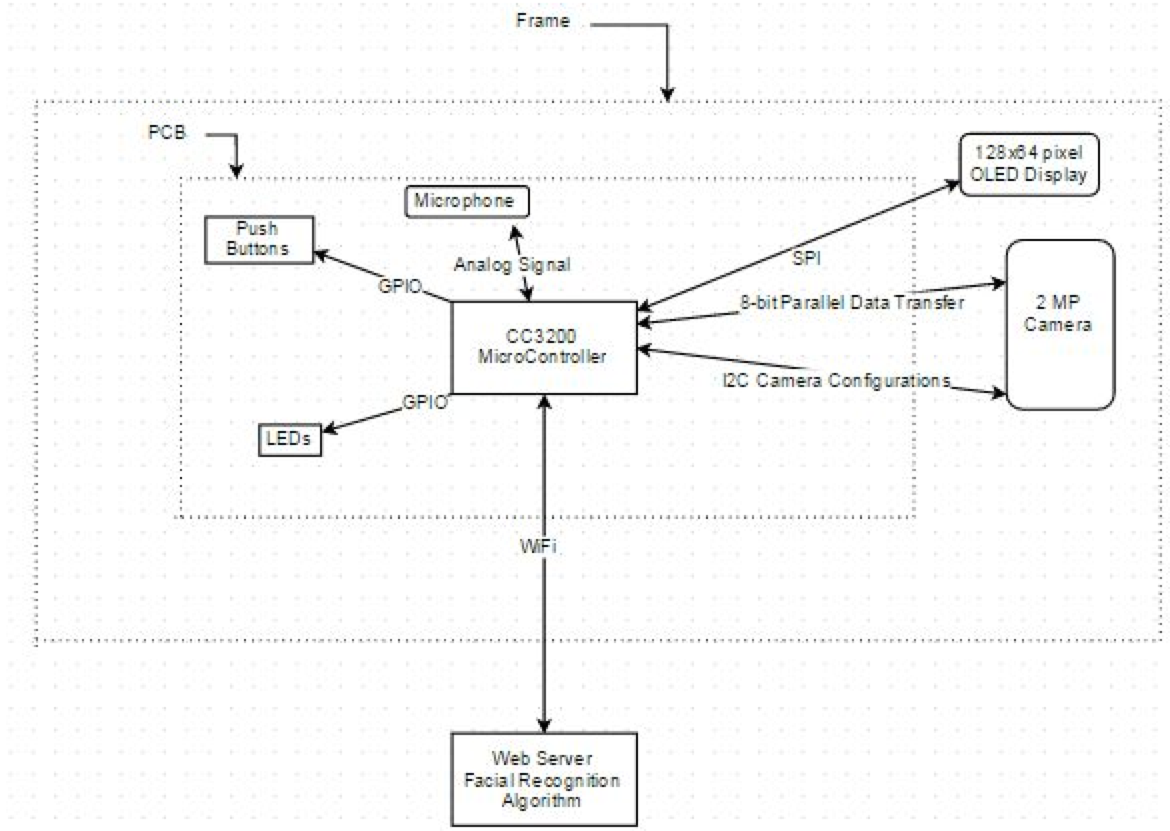
To remedy these issues, our idea in practice would be part of a larger product capable of acting as a camera, mobile notifications center, and augmented reality platform. This would all use the same hardware but the expanded feature set would bring far more value, justifying the high price. There is also the added bonus that it would be unclear what you were currently using this multi-featured device for, obscuring the fact you were using it to discover someone's name. With these issues assuaged, this is the solution we are proposing to implement.

Implementation Issues

To achieve our high-level goals for this project, we need to settle a number of low-level details. The image that is going to be captured will need to be high enough resolution for our face-recognition algorithm to work. The system needs to be fast enough such that there is no significant latency when the user sees a person and that person's name pops up on the display. OmniView needs to be light enough and small enough so that it is not irritating to be worn by the user. The power draw of the device also needs to be low, since we are powering via a small battery attached to the headset.

Our primary mode of operation will involve taking an image from the camera at a rate of 1Hz and sending the image to the web server. The web server will be running our facial recognition algorithm. When the web server identifies the person in the received image, it will send the name of that person back to the microcontroller. The microcontroller will then drive that name onto the OLED display. In images with more than one person, the algorithm will identify the largest face in the image. If the server does not have a high enough confidence level in its identification of a person, it will send an error message indicating that no face was detected or the face did not match any existing entries. The OLED would continue to display the name of the last identified person until a timeout of 15 seconds, at which point the display would be cleared. In this mode, the facial recognition algorithm will be comparing images against a pre-loaded database of image-name pairs. The entire device will be turned on and off through a push button and this will be the default mode of operation.

Our secondary mode of operation will involve the use of a microphone and camera in tandem, and is initiated with a second push button. This button push will start a data capture which is 10 seconds of microphone data, and one picture taken 5 seconds after the push. This allows the button push to be done prior to the interaction, making it more subtle and preventing it from interrupting the interaction itself. The device will then send both the captured image and the ten seconds of the microphone signal to the web server. The web server will perform natural language processing to parse the spoken name from the microphone signal. If the web server does not have a high enough confidence level, it will reject the image and name. If there is no rejection, the web server will dynamically add the received image and processed name to the database of image-name pairs. The web server will then send a response to the device. If the web server failed to identify a face in the image or name from the audio, it will send the device an error message which will be displayed. If the web-server successfully added a name-image pair to the database, it will send the name that the database was updated with. During an introduction, usually two names are said, the user's and the target's; however, the user's name can be excluded because it is previously known. One issue is the exact spelling of the target's name is impossible to determine from audio but that is acceptable and the server can be updated later if the exact spelling is different. For the duration of the secondary operation, primary operation will be paused.



We have chosen to use an external server in our application. This server will perform the facial recognition algorithm and query the database of face-name pairs. Using an external server and database allows us to keep the device worn on the face small and light. It also reduces the power draw of the PCB, which is being powered by a small battery on the headset. We will rely on open-source software for our facial recognition. To ensure feasibility, we have already selected our facial recognition algorithm, openBR, and successfully had 1.3 MP images identified against our sample database of face-name pairs within 1 second.

The communication between the server and microcontroller will be conducted using WiFi. This is done for two reasons: it expands the maximum communication distance between the server and the device, and it allows for a high throughput of data. Images are fairly data intensive, and we want the latency between an image sent to the server and the corresponding text sent back to the microcontroller to be as short as possible. Another wireless protocol with high throughput and even higher range than WiFi is cellular. We chose not to proceed with a cellular protocol, because of the increased power draw that cellular would bring[1]. In addition, the intended use case for our device makes the increased range of cellular unnecessary. Using WiFi does introduce at least one added layer of complexity to the device: the registering of the device as a station in a network. Instead of creating a dedicated UI on the glasses for setting up connectivity, we would instead allow for OmniView to be connected to as an access point, and communicate with a user's personal computer. Through this method, OmniView can be registered on the network through the use of a personal computer.

The features we are looking for in a microcontroller includes an interface for a high resolution camera, and support for a wireless protocol capable of streaming data. After some research, we determined the primary candidates to be the CC3200 and the STM32F479N. We chose to use TI's CC3200 microcontroller. This microcontroller has a dedicated core that implements the WiFi WLAN and TCP/IP stacks. It also has an on-chip 80 MHz Cortex M4 processor to perform other application logic and interface with peripherals. The microcontroller includes 2 UART modules, 1 SPI module, 1 I2C module, and an 8-bit parallel camera interface[2]. Freescale's STM32F479N processor was given brief consideration for it's MIPI controller[3]. MIPI is a camera interface used by many high resolution cameras. We decided on the TI CC3200 because it has built in support for a 2 megapixel camera and a core that handles the WiFi protocol. It is worth mentioning that TI has a surface mount antenna solution for the CC3200 development board. No one on the development team has experience with antenna design, so we intend to use the same surface mount antenna found on the CC3200 dev board. We believe this will result in the antenna being a non-factor in our list of foreseeable difficulties with developing OmniView.

There are several advantages to having WiFi capability within the microcontroller rather than through an external module. The most important reason is the form factor. With WiFi being handled by the MCU, we don't have to route a dedicated WiFi module. This would take up a lot of space in an application prioritizing low weight and size. Another important reason is that the vendor (Texas Instruments) has provided sample code for implementing WiFi on the microcontroller, which we can use as a reference in order to reduce development time. If we used a dedicated module, it is unlikely that we would be able to find sample code ported to our specific microcontroller.

We decided to use an Organic Light Emitting Diode (OLED) display over the popular Liquid Crystal Display (LCD). One reason for this decision is that OLED displays require less power than LCDs. OLED displays only draw power based on the section of the display that are non-black. In contrast, LCDs filter a backlight that is always on, and must use power to make the screen dark. Another important factor is weight of the display; OLED displays are lighter

---

1 http://www.computerworld.com/article/2506011/wireless-networking/mobile-users-prefer-wi-fi-over-cellular-for-lower-cost--speed--reliability.html

2 http://www.ti.com/product/CC3200/datasheet/abstract#XXXX0015868

3 http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN1876/PF260695

than LCDs. Communication between the display and the microcontroller will be done via a common serial protocol. OLED displays in our size range support I2C, SPI, or both. Since we only have 1 SPI / 1 I2C to work with, and the camera requires I2C, we can use the remaining SPI pins for the OLED display.[4]

All of the electronic components will be contained on one PCB, with the exception of the OLED display and the camera. The OLED display will be mounted on the frame, and connected to the PCB. This decision was made for the sake of the user being able to read the words being driven onto the display. The camera will be mounted directly on the PCB in a 90 degree angle so that we can place the camera in an area on the frame (perpendicular to the temple) that will result in optimal image quality. The PCB will sit within the frame. The frame will be created using a 3D printer. We estimate the frame(we define frame as the 3D-printed glasses frame with the unpopulated PCB included), to weigh around 50g. We have discovered a kickstarter project for 3D-printed glasses frames that claims a weight of 11g[5]. They use nylon as their plastic of choice. Due to the constraints of the 3D-printers at the university, we intend to use ABS plastic. ABS is slightly less dense than nylon[6]. Even though we're using a lighter material than the other frame developers, we expect our inexperience with 3D-printing CAD tools will result in a frame that is 3-4 times as heavy as the 11g nylon frame.

The size and weight of the entire device are two very important application parameters. If either value is too high, the head-mounted device is unusable. We will need to ensure that the weight of OmniView is not uncomfortable to the user. In addition, we need to make sure that the size is reasonable. Size also affects whether or not OmniView will be socially acceptable. We recognize that for a prototype, the device will probably not have the necessary form factor to be socially acceptable. Even google glass has problems with social acceptability, and that device was developed with a much larger budget and longer time frame than our device.

Given our peripheral choices, we estimate that our electronic devices will have a combined weight of 23 grams. Our chosen battery weighs 17 grams. Our total estimate for the weight of our device comes out to be 90g. We researched a similar device - Google Glass - and found that it weighs 43 grams. Given the weight of Google Glass, which we consider to be the minimum for complex electrical devices worn on the face, we consider the estimated weight of our device to be acceptable.

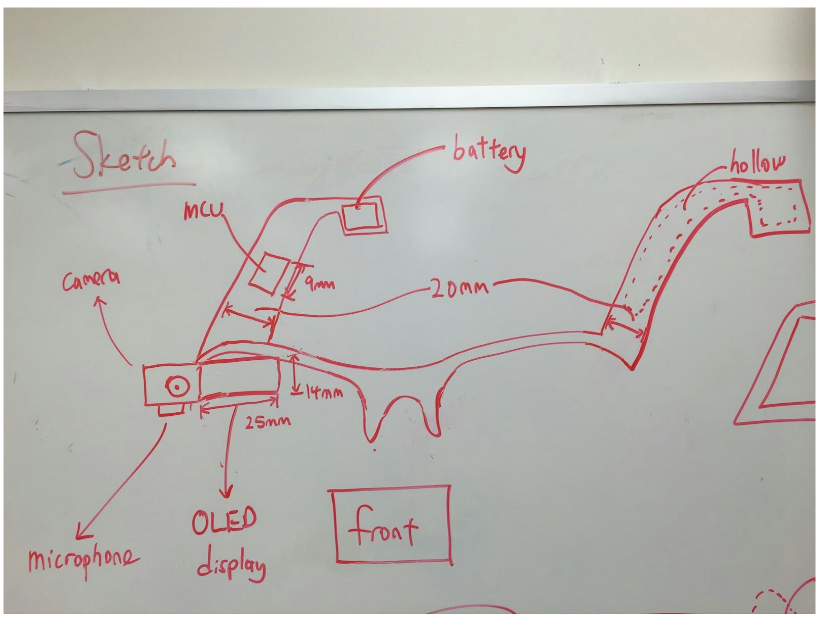
The glasses will have dimensions that are slightly larger than Google Glass. We plan to fit all the electronics on the right side of the frame, and hence, we expect a thicker and heavier temple on the frame. The temple is going to be 20mm wide (top to bottom of the temple piece) and approximately 10mm thick (left to right of temple piece). We wanted the glasses to be symmetric on both sides of the frame, but still not compromise on weight. Therefore, we decided

---

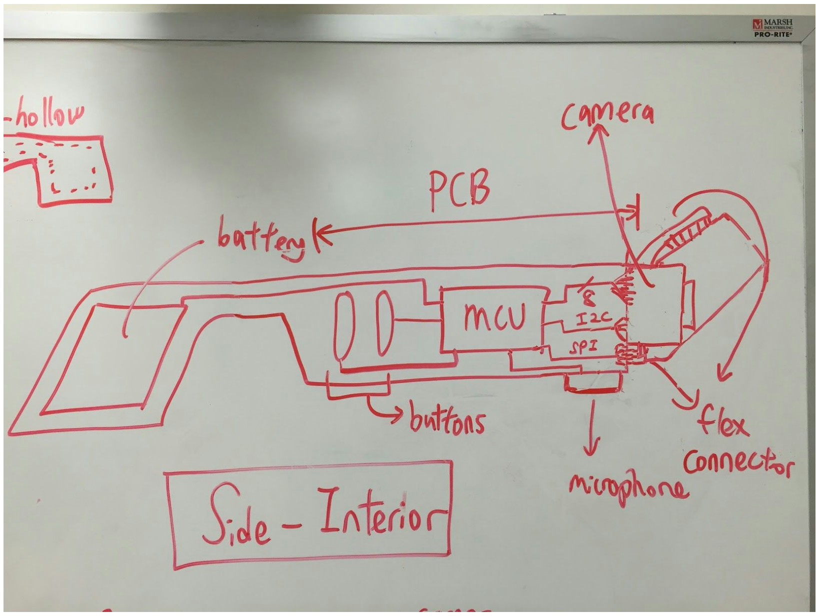
4 http://electronics.howstuffworks.com/oled5.htm
5 https://www.indiegogo.com/projects/mono-an-eyewear-3d-printed-to-fit-your-face#/
6 http://www.dotmar.co.nz/density.html

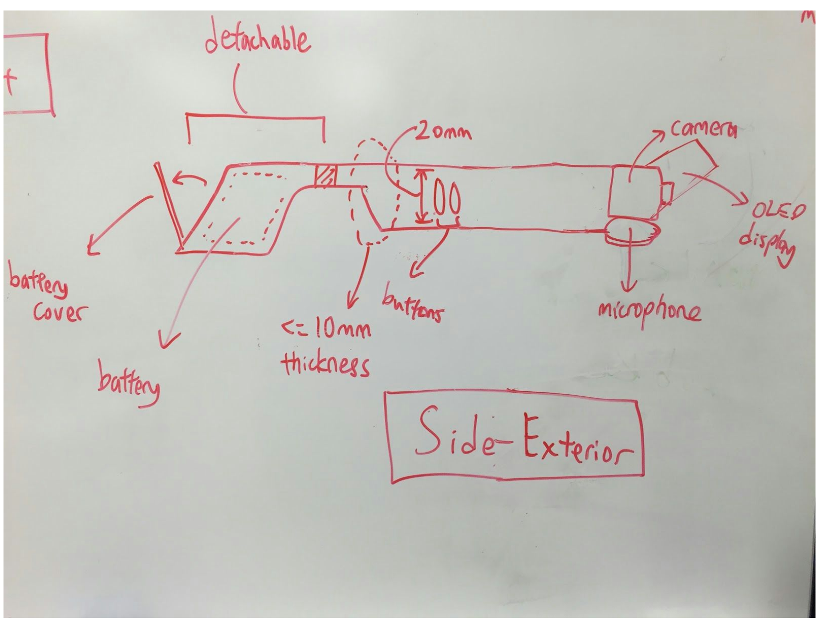
to design the left temple piece identical to the right, but have it be hollow to reduce the overall weight.

We have also considered non-technical issues that might arise after the product has been built and used. One of them is the privacy laws pertaining to the recording and taking of photographs of people in public. It is generally legal in the United States to take photographs or record videos of objects and people when you are within public property. The right to do so in a private property is ultimately decided by the property owner.[7] However in practice, this law does not inhibit smartphones users and we do not see it as an issue for our device either. There are currently many different approaches and proposed solutions regarding privacy concerns, but the laws have not caught up with wearable technology yet. There is no law that recording devices must have a light on or make a noise when taking a picture. Nevertheless, we will continue to develop the product with these laws in mind.

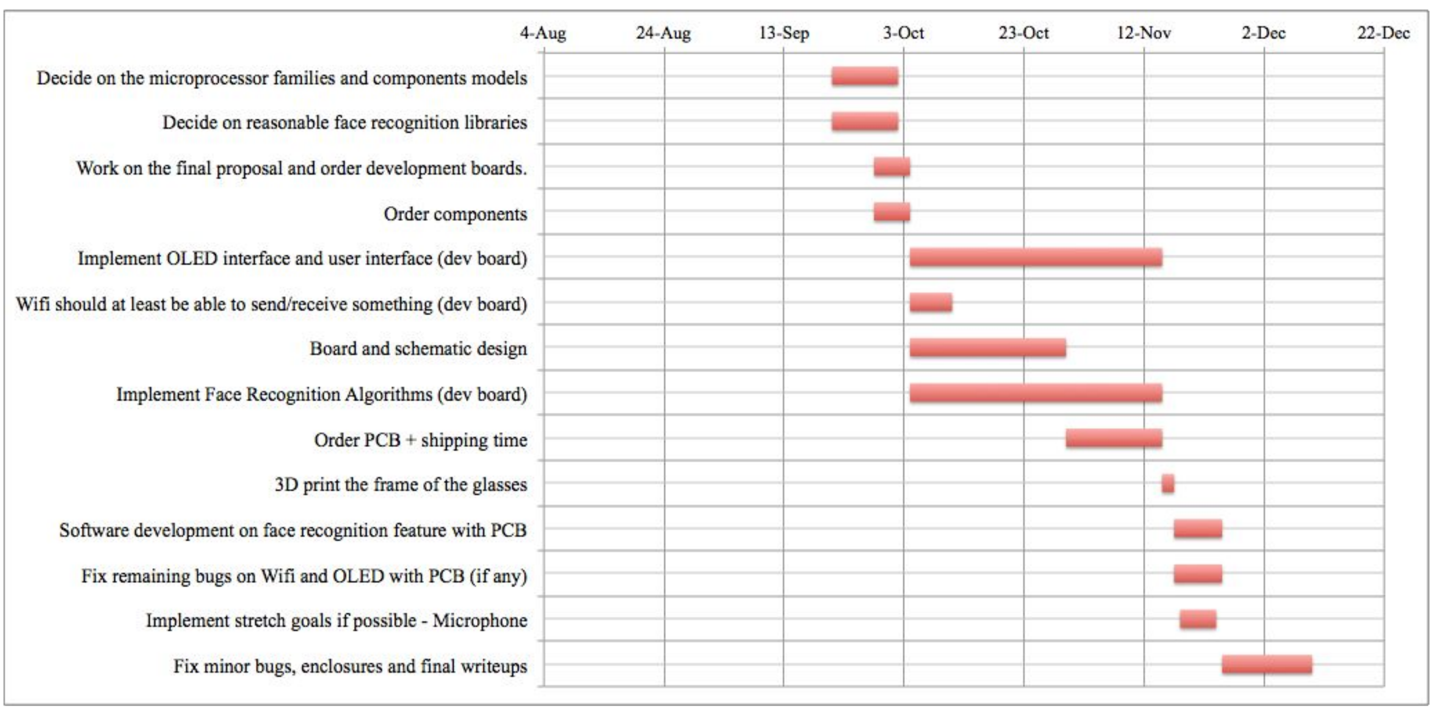


[7] http://www.krages.com/ThePhotographersRight.pdf


hollow
camera
PCB
battery
mcu
I2C
SPI
buttons
flex connector
microphone
Side - Interior


detachable
20mm
camera
OLED display
battery cover
battery
<= 10mm thickness
buttons
microphone
Side-Exterior

Task	Start Date	Duration (days)	End Date
Decide on the microprocessor families and components models	Sept. 21	11	Oct. 02
Decide on reasonable face recognition libraries	21	11	02
Work on the final proposal and order development boards.	28	06	04
Order components	28	06	04
Wifi should at least be able to send/receive something (dev board)	Oct. 04	07	11
Board and schematic design	04	16	30
Implement OLED interface and user interface (dev board)	04	42	Nov. 15
Implement Face Recognition Algorithms (dev board)	04	42	15
Order PCB + shipping time	30	16	15
3D print the frame of the glasses	Nov. 15	02	17
Software development on face recognition feature with PCB	17	08	25
Fix remaining bugs on Wifi and OLED with PCB (if any)	17	08	25
Implement stretch goals if possible - Microphone	18	15	Dec. 03
Fix minor bugs, enclosures and final writeups	25	15	10



The main bottlenecks of this project are going to be the development of the PCB and 3D printing the frame of the glasses.

Since the PCB we are dealing with is going to be small, we might need to be careful when we solder on it. That's why we allocated a lot more days for PCB ordering and shipping time, as well as testing it when it arrives before mid November. We generally expect it to arrive in the first week of November, and if anything goes wrong, we can reorder them again and still make it in time for our deadline.

Another thing that could go wrong is with the components and dev boards are not shipped in time for us to begin the project. However, we have already selected all the components we will use and ordered them. In addition, a few of us have these exact components at home, so we will not be idle while waiting for the parts to be shipped.

We will need a 3D printer to create the frame of the glasses. The 3D printers in the Dude are a little bit difficult to work with, and we will talk to people in the mechanics lab who have access to more advanced 3D printers. We have allocated time to get this done after we receive our PCB just in case anything is wrong with the dimensions of the PCB can be accounted for by changing the enclosure around it. For milestone 1, the prototype will be capable of displaying meaningful text on the OLED display. Basic Wi-Fi module will be able to send/receive data from server. We hope that PCB schematic design will be done. The image recognition algorithm should already be implemented and able to show meaningful information. For milestone 2, the PCB will have been ordered. The OLED display and Wi-Fi software will be finished with only minor bugs left, and we will have decided on whether we want to implement functionalities related to the microphone. In addition, the frame of glasses will have been 3D printed.

Design Expo & Presentation

By the time of the design expo, we will have a complete PCB for the device. We will have also created a 3D-printed set of glasses to mount this PCB onto. We will demonstrate OmniView by allowing people to wear the glasses, then identify people in our group who will all be registered in the face-name database. If we have completed the stretch goal of dynamic addition to the database, we will also use this feature and show them the glasses now recognize them and display their name.

In the auditorium-style demonstration to the rest of the class, we will use a member of the audience to repeat the above demo, but also have the server display the name associated with received images to a terminal and project that on the screen behind us. The only need we have for the demonstration is that the room the demonstration is being done in is well-lit, so that the images being captured are of high quality.

Since our project uses WiFi on University grounds, we will also need make sure that our device is made known to the university. We intend to have Omniview function as an access point for the purposes of the demonstration, so we don't have to deal with the added complexity of setting it up Omnview a station and connecting to a university access point.

Budget & Materials

The components needed to implement this project are listed in the table below. We focused on choosing components that would minimize our weight, size, and power draw.

Components	Description	Quantity	Cost/item	Total Cost
Micron MT9D111 Camera	● Supply Voltage 2.5 V ~ 3.1 V ● 2 Megapixels ● Size:8mm x 6 mm ● Weight: 5g ● Power: ○ 348mW at 15 fps, full resolution ○ ≈94mA	2	$10.19	$20.38
Univision OLED display	● Supply Voltage 3.3 V ● Monochrome ● 128*64 pixels ● Size: 25mm x 14mm ● Weight: 8.5g ● Power: ○ sleep mode 5uA ○ max 28.9mA	2	$14.95	$29.9
Microphone	● Supply Voltage 2.7~5.5 V ● SNR 64 dB ● Size: 9.7mm x 4.5mm ● Weight: 0.8g ● Power: ○ sleep mode 1uA ○ max 6mA	1	$7.95	$7.95
Texas Instruments CC3200 MCU with Wifi Network	● ARM Cortex M4 Core 80 MHz ● Built in Wi-Fi 802.11 b/g/n ● 8-bit parallel camera interface ● 27 GPIO pins ● Size: 9mm x 9mm ● Weight: 2.5g ● Low Power Modes	5	$18.92	$94.60

	○ Hibernate: 4 μA ○ LPDS: 250 μA			
PCB	● Weight: PCB + Misc. Parts: ≈ 50g	5	$20.85	$104.25
SimpleLink Wi-Fi CC3200 Dev Board	● Development board for MCU	2	$29.58	$59.16
Camera Booster Pack	● Camera “shield” for CC3200	2	38.06	$76.12
Lithium-polymer charger	● Charges Lipo Battery	2	$7.95	$15.8
Lithium-polymer battery	● Output: 680mAh ● 3.7 VDC. ● Size:40 x 25 x 8.5mm ● Weight: 17g	5	$16.99	$16.99
Misc. Parts	● LED, Buttons, 3D printing materials and other passive components			$200.00
Shipping	----	----	----	$200.00
Total	----	----	----	$822.15

Camera:
http://www.newegg.com/Product/Product.aspx?Item=9SIA5W02EF6650&nm_mc=KNC-GoogleMKP-PC&cm_mmc=KNC-GoogleMKP-PC-_-pla-_-Gadgets-_-9SIA5W02EF6650&gclid=CPH4uq2JjMgCFdW7GwodIDoNFA&gclsrc=aw.ds

Univision OLED Display:
http://www.adafruit.com/products/326

Texas Instruments CC3200 MCU:
http://www.digikey.com/product-search/en?mpart=CC3200R1M2RGC&v=296

Texas Instruments CC3200 Dev Board:
http://www.ti.com/tool/CC3200-LAUNCHXL?keyMatch=cc3200%20launchpad&tisearch=Search-EN-Everything#buy

Texas Instruments CC3200 Camera Booster Pack
http://www.aliexpress.com/item/CC3200-Camera-booster/32276520437.html?src=google&albch=search&acnt=479-062-3723&isdl=y&aff_short_key=UneMJZVf&albcp=266121556&albag=7593673036&slnk=&trgt=dsa-42862830006&plac=&crea=64152518476&netw=g&device=c&mtctp=b&memo1=2o1&gclid=COvzov_TqcgCFQcOaQodG9QPPw

Microphone:
http://www.adafruit.com/products/1713

Lihium-ion polymer Battery:
http://www.miniinthebox.com/syma-x5c-x5c-1-explorers-parts-x5c-11-3-7v-500mah-update-3-7v-680mah-lipo-battery-3-in-1-cable-line-x-5pcs_p2041286.

3-D printing plastic:
http://www.makershed.com/products/1-75mm-abs-1kg-spooled-filament?gclid=Cj0KEQjw-b2wBRDcrKerwe-S5c4BEiQABprW-IJFh5Xj-NJSy5-yB43XXVX5OqK1UEVK9iAmf0rJn1MaAnZA8P8HAQ

As mentioned previously, the target weight of OmniView is less than 100g and the target power consumption is less than 450maH. The device should be small enough to be comfortably worn. All the components listed in the table above are selected based on satisfying these criteria. The major components that will be included in the final product are TI's CC3200 microprocessor, Micron's MT9D111 camera, a 128x64 pixel OLED display, the microphone, battery, PCB, and 3D printed frame.

TI CC3200 Microprocessor + Wi-Fi

As mentioned in the Implementation Details section, the Texas Instruments CC3200 MCU comes with a Wi-Fi network processor. OmniView will be transmitting 2 MP images to the server at a frequency of 1 Hz and returning the facial recognition result back to the system. With 802.11 b/g/n specification, and using the TCP networking protocol, the processor is capable of transmitting up to 13 Mbps. This transmission rate is sufficient for our application. The low power consumption of the CC3200 MCU makes it more attractive, which takes about 350mA in the worst case.

Micron MT9D11 Camera

A high quality camera is required for this application, since the captured image will be used for facial recognition. Using pictures taken from a 1.3 megapixel front camera on a Nexus 5, the facial recognition algorithm successfully identified the members of the development team from about 10 feet away. Given this experiment, we believe a reasonable target resolution is 1.3 megapixels or higher. Low power consumption is also an essential criteria for our camera module.

We researched a couple of camera modules such as the OV5460, OV2722, OV3660, Micron MT9M11 and MT9D11 that have the potential to support the TI CC3200 MCU and meet our constraints. The OV Cameras have superior performance (above 2 megapixels with lower current drain), however, they only support either serial interface (CSI) or 10-bit camera parallel interface (CPI) data output. This is problematic, as our MCU has an 8 bit CPI data input. Of the cameras we evaluated, only the Micron MT9M11and MT9D11 use an 8 bit CPI data input. We found that the MT9D has 2 megapixel resolution while the MT9M has only 1.3 megapixels. Even though the power consumption for the D version (≈94mA) is about twice as much as the M (≈45.9mA), we believe the additional resolution of the MT9D will lead to increased success for the facial recognition algorithm, and have decided to use the Micron MT9D11 for this project. Another advantage of the D version is that CC3200 MCU happens to have existing MT9D11 drivers, which means using the MT9D11 will save us a lot of development time.

Univision OLED display

We need a small display that can exhibit a decent amount of information (high pixel density), as well as being power efficient when sleep mode is activated. We evaluated several OLED displays and decided that the Adafruit 128x64 OLED display is well suited for our application. The screen also ships with flexible connectors, which we will incorporate into our 3D printed casing. The Sparkfun display was a little too expensive given our budget.

PCB

The number of PCBs required is a rough estimate, and depends mainly on the quantity the manufacturer can supply. However, having a couple backup PCBs will give us more space when dealing with unpredictable failures.

Battery

We currently propose to use a 680mAH battery for our face recognition glasses. Since the battery is one of the heaviest components that we are going to mount on the glasses, we need to keep track of the battery size and weight without compromising the usability and practicality of the glasses. We mapped out our power draw of all the electronic components, and concluded a worst case current draw of 456.05mA.

current(battery)
= current(camera) + current(display) + current(microphone) + current(wifi module + processor)
= 94.05mA + 6mA + 6mA + 350mA
= 456.05mA

This calculation assumes all our components are operating at the same time, and indicates that we will have at least 1.5 hours of battery life. We are not going to use all our components all at once, so we can expect to have a longer battery life than this.

Google Glass has a 570mAH Lithium-polymer battery which fits in the right end of glasses. We plan to do a similar setup for the 680mAH battery, which has a mass of 9g.

Conclusion

Overall, we believe that remembering names and faces is a real challenge that exists in our world. Our project simplifies the process and gives you a visual aid to help with that problem. The project is realistically accomplishable and our main risks remain in getting everything accomplished in our timeframe, with the device being small, light, and fast. If the software and hardware are not able to perform at a reasonably high level, it will affect the performance of our whole project by attaching a significant lag to the retrieval of a person's name. Despite this potential issue, we as a group are very dedicated and passionate about getting the project working, and are prepared to invest significant hours to make this happen.

Appendix A

Design Criteria	Importance	Will/Expect/Stretch
Facial recognition correctly determines new/old faces > 80%	Fundamental	Will
OLED displays name and info quickly enough such that user will not have to wait	Fundamental	Expect
Battery power > 1 hour	Important	Will
User has simple/efficient control of the glasses	Important	Will
Weight should not exceed such a limit that it will cause pain or discomfort to the user	Important	Expect
Software & Microphone are able to capture and person's name	Optional	Stretch
Flexible PCB	Optional	Stretch

Appendix B

Milestone 1:

Prototype is capable of displaying meaningful text on the OLED display.
Basic Wi-Fi module is able to send/receive data from server.
PCB schematic design should be done.
The image recognition algorithm should already be implemented and able to show meaningful information.

Milestone 2:

PCB should have been ordered by now.
The OLED display and Wi-Fi software should be finished with only minor bugs left.
Have decided on whether we want to implement functionalities related to the microphone.
Frame of glasses should be 3D printed by now.

Appendix C

```
class Camera {
	public:
		Camera(CameraMode mode);
		void takePicture(int* buf, int& bufSize);
		void sleep();
		void wakeUp();
	private:
		void initCameraMode(CameraMode mode);
		enum CameraMode {RGB565, RGB555, RGB444, JPEG422, JPEG420,
Raw10Bit};
		uint8_t parallel_port; // 8-bit camera parallel port.
};

class Display {
	public:
		Display(DisplayMode mode);
		void writeLine(int line, string message);
		void clearLine(int line);
		void clearScreen();
		void sleep();
		void wakeUp();
	private:
		void initDisplayMode(DisplayMode mode);
		enum DisplayMode {SPI, I2C};
};

class Microphone {
	public:
		Microphone();
		void record(int seconds, int* micBuf, int& bufSize);
		void sleep();
		void wakeUp();

	private:
		void initMicrophone();
		void startRecording(int* micBuf, int& bufSize);
		void endRecording();
};

class WifiModule {
	public:
```

```
        WifiModule();
        void sendPicture(int* picBuf, int bufSize);
        void sendAudio(int *micBuf, int bufSize);
        string getResponse();
        void sleep();
        void wakeUp();
    private:
        void initWifiModule();
        void sendTCP(int* buf, int bufSize, int ipAddr);
        void receiveTCP(int* buf, int bufSize, int ipAddr);
};

class MCU {
    public:
        MCU();
        void sleep();
        void wakeUp();
    private:
        void boot();
};

class FaceRecognitionSoftware {
    public:
        void receiveRequest(RequestType& type, int* buf, int& bufSize);
        // returns result string. ex: “Garrison” success, “No Face Found” failure
        string comparePicture(int* picBuf, int bufSize);
        string addToDatabase(int* picBuf, int bufSize, string name);
        string addToDatabase(int* picBuf, int bufSize0, int* micBuf, int bufSize1);
        void sendResponse(string response);
    private:
        enum RequestType {compareFace, addFace};
        void openBrInit();
        void openBrAnalyze(string imgPath);
        void openBrCompare(string imgPath, string galleryPath);
        void addFaceNamePair(string imgPath, string name, string galleryPath);
        void sendTCP(int* buf, int bufSize, int ipAddr);
        void receiveTCP(int* buf, int bufSize, int ipAddr);
};
```

Appendix D

Each person in the group has agreed to the following:

- 25 hours/week are expected of each person
- The group is free to meet on Tuesdays, Wednesdays, Saturday afternoons, and Sundays
- The team plans on getting most of its work done as early as possible.
- The team will work together as much as possible and individually only when necessary (or away).
- We all agree to meet every Sunday to work together, and this will also us to do weekly progress checks.
- Tasks responsible for:
    - Mechanical: Tzu-Fei
    - PCB & Soldering: John, Tzu-Fei
    - Programming: Garrison, Loren, Bobby
    - Scheduling Meetings: Garrison
    - Minutes: John, Bobby
    - Requirement Gathering: Garrison
- Known Conflicts/Missed Time:
    - Interviews: Everyone
    - Thanksgiving: Garrison, John
    - Badminton Tournaments: Loren
- Team meetings will be held in lab.