

# TableCast

Authors: Caroline Crooks, Tahaseen Shaik, Sumayya Syeda  
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

**Abstract**—TableCast is an interactive table projector designed to assist users in the kitchen via computer vision, voice commands, and gestures. Users will choose a recipe and follow instructional content projected on their kitchen countertop, including video demonstrations and user responsive feedback, calibrated to their specific countertop. We accomplish this by using high quality off-the-shelf peripheral devices, extensive computer vision algorithms, open source speech recognition engines, and web application development software.

**Index Terms**—Computer Vision, Cooking, Gesture Recognition Tracking, Homography, MediaPipe, Object Detection Tracking, Projections, SIFT, Speech Recognition

## 1 INTRODUCTION

TableCast is an interactive table projector designed to assist users in the kitchen via computer vision and voice commands. Users will be guided through recipes of their choosing via text and video steps without having to touch any sensitive devices.

Currently, most people watch cooking videos or read recipe articles online while they are cooking. There are a few problems with this. In many instances, fingers become messy from cooking. In order to look at the recipe again, you must constantly wash your hands or risk your device also becoming messy. This process is frustrating and unsanitary. Secondly, some people have trouble following the steps due to the differences between the instructor's tools/setup and one's own. There is no guide on how to cook something in one's own kitchen.

In order to create a intuitive and user-friendly interface for TableCast, we will be utilizing computer vision extensively and incorporating user interactions via voice commands and projected buttons. TableCast will initially calibrate to the user's kitchen space before beginning the cooking process. The recipes will be broken into steps and then displayed on the table in text and video format for users to follow. Users will have the option of using voice commands or projected buttons in order to go back a step, pause and play. Furthermore, TableCast will use object re-identification to track the ingredients as they are prepared for the cooking process in order to best direct the user. This design is meant to make cooking new dishes easier and avoid the issues of contamination of devices and overall inefficiency in the kitchen.

## 2 USE-CASE REQUIREMENTS

### User-Interaction:

- UR1. Physical interaction with the device or web-app must be limited to 4 unique interactions
  - UR1a. Start, stop, select recipe, change recipe
- UR2. User should be able to interact with device using projected buttons.
- UR3. User must have a seamless experience with the system with low latency.

### Projection:

- PR1. Perimeter of projection must be within 2 cm of the counter
- PR2. Section projections on counter must be within 1 meter of the users starting position
- PR3. Sections should not overlap and should have at least 1 cm spacing between.

### Device:

- DR1. Voice command recognition must have an accuracy rate of at least 90%
- DR2. Survey of user experience must have at least 85% satisfaction rate
- DR3. Accuracy of tracking must be at least 90%

## 3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

### 3.1 Architecture Overview

The system components comprise of digital hardware components, software components, and mounting mechanisms. Our processing unit is the Jetson AGX Xavier, which will send and receive information to the peripheral hardware devices. A camera and microphone provide input data to the computer. A projector and stereo speaker are output devices. An external PC is both an input and output device that will communicate with a web server. Within the AGX, primary software tasks comprise of computer vision processing, voice recognition, the web server, user interface control, and overall system command processing. Mounting devices are tripods to hold the camera, AGX, and projector, along with supportive casings to provide additional protection. One tripod will support the AGX and camera, while another tripod will support the projector. These components are further described in the block diagram (Figure 7) and in Section 6, System Implementation. We are keeping version control using Atlassian

Suite's BitBucket. This enables us to consistently track and merge our progress to make system integration easier.

### 3.1.1 Hardware Components

The camera is placed above the center of the table. It must have a full view of the entire table. Real time data will be sent from the camera to the AGX's computer vision processing unit. The camera will be turned on throughout the duration of the recipe, from the calibration step to the completion of the recipe.

The microphone is a wireless, lavalier microphone that a user will attach to their clothing. The user will turn it on after they pick the recipe and may turn it off after the recipe is complete. The microphone will send audio data to the voice command software module to be processed.

The projector displays the user interface onto the table. It will receive image data from the projection module.

The stereo speaker will sound an alarm when a timer counts down to zero.

A PC communicates with the web server on the AGX. This PC will be a user supplied device that will enable the user to browse recipes and start to cook.

### 3.1.2 Software Components

The vision module receives the video feed from the camera. It will use MediaPipe and OpenCV as computer vision libraries to do object detection, tracking, and gesture recognition. Ingredients on the table will be monitored in real time, and user gestures will be interpreted as commands to the program.

The voice command module receives data from the microphone. Speech is interpreted with speech-to-text tools within the module to interpret commands.

The projection module receives data from other modules to create a responsive user interface. The interface includes recipe video controls, timer tools, buttons, and ingredient placement guidelines. After an image is created, the image is then warped according to a computed homography. This warped image is sent to the projector.

The command processing module is the central control system for the overall software system and coordinate the modules. For example, when a user selects a recipe, the web server will pass information to the command processing module, which will initiate the other modules.

The web server starts upon powering on the AGX. It hosts the web application that communicates with the user's PC. The user will browse and select recipes on the application.

### 3.1.3 Mounting Components

There are two tripods. One tripod supports the projector and hold it in place so that it is secure at an angle. Another tripod has an extending arm that protrudes over the table. This will support the camera that is facing the table. The AGX will also be securely attached to the same

tripod, on the part of the arm closer to the base of the tripod.

## 3.2 Principle of Operation

### 3.2.1 Installation

The user will place the tripods next to their kitchen countertop, across the side of the table where they will be cooking. The camera should be angled down, facing the countertop. The projector should be angled down towards the countertop.

### 3.2.2 Solution Flow

**Set Up.** The user will turn on the AGX. Upon boot, a script will automatically set up a network interface and a web server. The user will connect their own personal laptop to the network, named TableCast\_Device, which will be broadcast from the AGX once it is. This will be a Peer-to-Peer (P2P) connection between the laptop and the AGX. The user will then navigate to the TableCast web application running on the web server. On the application, users can scroll through recipe options and select one to cook with TableCast. The user will then attach the microphone to their clothing.

**Recipe Task.** A flow diagram is displayed in Figure 1. With the recipe selected, TableCast will guide users through a calibration step. The computer vision and projection modules will be used to complete this step. Once calibrated, the device will project the user interface on the table. Users will place each ingredient in a box with the corresponding label. With all the ingredients detected, the user can start the recipe. The user will follow a video projected on the table, using gestures or voice commands to aid them while cooking. This will repeat for each step until the recipe is complete.

**Shut Down.** The user can stop the program at any time. If the user has paused the program, they will then be given the option to exit the recipe or shut down the device.

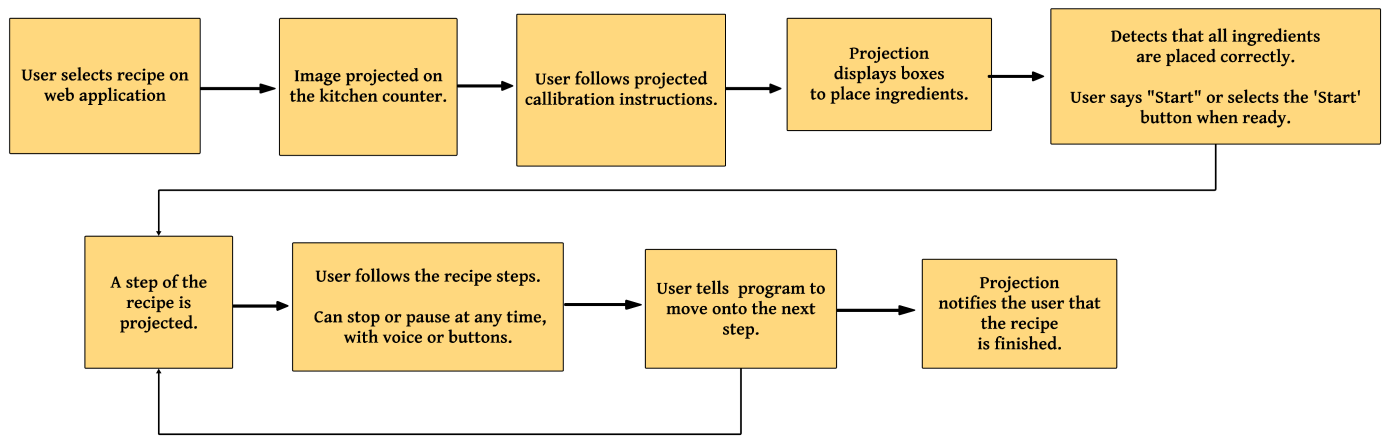


Figure 1: Solution Flow

## 4 DESIGN REQUIREMENTS

IR1. Projector mount must be adjustable.

- This relates to UR2 as the projector must be mounted properly to have projected buttons.

IR2. Warped projection on table looks flattened.

- Ensures PR1, PR2 and PR3.

IR3. Voice commands must recognize wake word to complete any command with an accuracy of 90%.

- Ensures DR1.

IR4. Camera must be mounted such that it points down and is 3-5 feet above the counter.

- Ensures PR1, PR2, PR3. To meet all the proper projection requirements, the camera must be mounted directly above the table. .

IR5. Gesture recognition must have 90% accuracy.

- Ensures DR3.

IR6. The latency of the video instruction must be within 3-5 seconds.

- Ensures UR3.

IR7. The gesture and voice commands must be recognized within 3 seconds.

- Ensures UR3.

### 4.1 Software Components

- OpenCV
- MediaPipe
- Porcuine Wake Word
- SpeechRecognition Python library

- OpenAI Whisper
- Python SimpleHTTPServer
- isc-dhcp-server
- Flutter
- Flask

### 4.2 Hardware Components

- Arducam IMX219 Low Distortion M12 Mount Camera Module
- Viewsonic Projector
- Jetson AGX Xavier
- Electrical connections
- Laptop
- Wireless Lavalier Microphone
- Stereo Speaker

## 5 DESIGN TRADE STUDIES

### 5.1 Gesture Recognition Libraries

Gestures are used as inputs to execute system commands such as start, stop, next and previous. The categories that mattered the most were computing power, latency, and precision. These categories ensure that the system is efficient, fast and precise which allows for a more satisfactory user experience. To score these categories, we used a study[2] that compared the two models with metrics. It was clear that although OpenPose was more robust, MediaPipe would have the specs required for our system while remaining light weight and reliable.

**Gesture Recognition**

|                 | MediaPipe | OpenPose |
|-----------------|-----------|----------|
| Computing Power | 5         | 3        |
| Latency         | 5         | 3.5      |
| Precision       | 4         | 5        |
| Total           | 14        | 11.5     |

Figure 2: Gesture Recognition Trade Study

**5.2 Projector**

The projector is one of the cornerstones of our project. Without a high quality projector, it is impossible to meet our project requirements. With our limited budget, it was very difficult to find a satisfactory projector, but we were able to narrow down the decision to be between 2 projectors: the VANKYO Performance V700W and the ViewSonic 4500 Lumens SVGA Projector. In order to choose the most compatible projector, we considered 4 different factors: ANSI Lumens, projection size, projector size and cost. Between the two options, we found that the ViewSonic projector had the highest amount of ANSI Lumens (4500) which would be the minimal amount needed to work well in relatively bright environments. The promised projection size was higher by a little bit for the VANYKO, but ultimately it's large form and low ANSI Lumens pushed us to select the ViewSonic projector instead.

**Projector**

|                 | ViewSonic | VANYKO |
|-----------------|-----------|--------|
| ANSI Lumens     | 5         | 3      |
| Projection Size | 3         | 4      |
| Projector Size  | 4         | 2      |
| Cost            | 4         | 4      |
| Total           | 16        | 13     |

Figure 3: Projector Recognition Trade Study

**5.3 Speech Recognition Engines**

The software is primarily being programmed in Python, and the SpeechRecognition Python library is

well documented. The goal was also to use a free, open source engine. The library simplifies accessing and using several speech recognition models. There are 13 engine options for speech recognition. Out of those 13 options, only 4 options function offline. The system will not be connected to the Internet, so this is a necessary feature. These four options are CMU Sphinx, Snowboy Hotword Detection, Vosk API, and OpenAPI Whisper. Snowboy is a hotword detector, for detecting phrases like "OK Google" or "Alexa", so this engine will not be considered, out of a preference for longer phrase recognition.

To compare accuracy rates, we found a research paper that compared Sphinx, Vosk, Whisper, and two other engines [3]. A graph from this paper is featured in Figure 4. The Whisper engine had the lowest error rate, followed by Vosk, then Sphinx. Another part of the paper researched the execution time of each engine. Vosk was the fastest, followed by Whisper, then Sphinx. For our system, the accuracy of the voice commands is a slightly higher priority than the speed, which was put into consideration when selecting the engine.

**Speech Recognition Engines**

|             | OpenAI Whisper | Vosk API | CMU Sphinx |
|-------------|----------------|----------|------------|
| Accuracy    | 5              | 3        | 2          |
| Speed       | 4              | 5        | 2          |
| Ease of Use | 5              | 5        | 5          |
| Total       | 14             | 13       | 9          |

Figure 4: Speech Recognition Engine Comparison Table

**6 SYSTEM IMPLEMENTATION**

**6.1 Website Module**

**6.1.1 Network Interface**

The AGX will host an ad-hoc network that will enable a P2P connection with the user's laptop. On boot, the AGX will launch a bash script that enables its network interface and create a dhcp server, using the package isc-dhcp-server package. The user's computer will be able to select the 'TableCast\_Device' network to connect.

**6.1.2 Web Server**

The Python library SimpleHTTPServer will host the web application on the AGX. This application will be accessible via web browser URL on the user's laptop.

### 6.1.3 Web Application

The web application will be developed using Flutter because it is simple to use and lightweight. User actions will include:

- Scroll through a list of recipes.
- Click a recipe to read more details about the recipe.
- Click the 'start' button to select the recipe to cook.
- Click the 'stop' button to stop the currently running recipe.

When a user selects 'start', the backend system handles this input. Flask will be the backend software employed to handle the web application interactions. The 'start' input will be sent to the Command Processing unit, which will handle the start up behavior of the other modules.

## 6.2 Speech Module

### 6.2.1 Hardware

The microphone (BOYA Wireless Lavalier Microphone) must be powered on and clipped onto the user. The receiving device has a USB C plug in that will be connected to the AGX.

### 6.2.2 Software

There will be two libraries used to process speech recognition: the Picovoice Porcupine Wake Word and the Python library SpeechRecognition with OpenAI Whisper. A free developer Picovoice account is used to create the recognition model for the wake word. The tiny.en model of Whisper is employed in this program. This way, when the user says "TableCast [command]", the program will quickly detect an instance of a command, avoiding accidental command recognition during a normal conversation. Through testing, this method has been found to be more reliable than just transcribing speech with Whisper while searching for the wake word. Once the Wake Word is detected, Whisper's transcription of the phrase after the wake word is searched for a valid command. In this case, the Central Processing module is notified, which will direct the other modules to carry out the corresponding tasks.

### 6.2.3 Voice Commands

- "TableCast stop/pause video" - pauses the recipe video. Do nothing if already paused.
- "TableCast resume/play video" - plays video the recipe video. Do nothing if already playing.
- "TableCast restart/replay video" - replays the video from the beginning.
- "TableCast stop/pause timer" - pauses the timer. Do nothing if already paused.

- "TableCast resume/start timer" - resumes the timer. Do nothing if already running.
- "TableCast next step" - switch to the next step of the recipe.
- "TableCast previous step" - switch to the previous step of the recipe.

## 6.3 Vision Module

### 6.3.1 Hardware

The Arducam IMX219 Low Distortion M12 Camera Mount Module must be connected to the camera port on the Nvidia Xavier AGX.

### 6.3.2 Object Tracking

We plan to use the SIFT (Scale Invariant Feature Transform) algorithm for object detection and object tracking. Once the first image of the object has been taken, SIFT will be used between each frame to determine the new location of the object.

SIFT extracts features from the target object in an image and doesn't rely on the object's orientation or size to recognize it in another image. This is especially useful in our scenario as ingredients are very likely to rotate and change form throughout the recipe.

To make object detection easier, we initially take an image of the object in a designated labeled section of the counter. With this first image, SIFT will know what to look for in the initial frame and continue to build upon the reference images for subsequent frames.

### 6.3.3 Gesture Recognition

Users have the option to make commands using gestures. We will be using the following gestures:

- Open Palm Select
- Swipe Left
- Swipe Right

MediaPipe already has a trained model that can quickly detect an open palm. When the user places an open palm over a projected button, the system knows they are selecting that button.

Mediapipe also tracks landmarks on the hand as the hand is in motion. We will be using this feature to detect x, y, z coordinates of the landmarks. The displacement of the landmarks in the x direction while then hand is open will determine if the user is swiping right or left. Of course, the user may be moving their hand in either direction as they are cooking. In order to avoid false positives of gestures, we will only consider the gesture in designated sections on the counter.

## 6.4 Command Processing Module

This module will take processed data from the vision and speech module to output commands. Each input will be mapped to a command such as stop/pause, resume/play etc. Since output from both the vision and speech module can both map to same command, a central command processing module is necessary to manage these controls. Furthermore, the Command Processing Module is responsible for executing the actions determined by gestures and voice commands recognized in the vision and speech module.

## 6.5 Projection Module

The projection module involves three main steps: (1) coordinating projector calibration (2) receiving data from other modules as user input to the interface and (3) warping the image according to the corresponding homography.

### 6.5.1 Hardware

The primary hardware component for the projector module is the projector and the AGX Xavier. The AGX must process the UI before communicating the data to the projector via HDMI in order to be projected onto the kitchen countertop. This is the most key aspect of the project as it is what enables the user to interact with the product.

### 6.5.2 Projector Calibration

The projector calibration procedure is designed to ensure that the resulting projection of the UI fits the table according to the accuracies listed in our requirements. The calibration will consist of using the camera to detect the outline of the table and project a red outline to enable the user to best calibrate the projector with aid from TableCast. Since our product is reliant on a tripod system, it is our intent to make the calibration system as intuitive and user friendly as possible by ensuring appropriate messages are conveyed through our projection. The user can simply follow the instructions and no prior experience will be required. The height and angle of the projector will be in quantized intervals to reduce the amount of variability in the calibration. After verifying the outline is accurate, various geometric shapes will be projected to ensure the image warping is being handled correctly. This will be verified by the camera. Additionally, once calibrated the system need not be altered on later recipes in the same environment.

### 6.5.3 Flutter API

Flutter is used for the projected user interface that the user will interact with on the table.

## 6.5.4 UI Design

The interface is divided into three sections corresponding to the layout of the table.

- **Burner Area.** There will be no content displayed above the burner due to user safety concerns. Users will not be directed to reach above or across the burner.
- **Workspace Area.** The user is allocated space on the table to cook. Above that, video content, buttons for video control, notifications, and a timer.
- **Ingredient Area.** Users place their ingredients in the grid section. The user also has table space for other non-ingredient materials, like plates and utensils. At the top of this section, there is a settings bar.

There are three possible layout options, according to the burner position on the table: (1) burner center, (2) burner left, and (3) burner right. This layout is chosen once the computer vision module has determined the position of the burner. Figures 5 and 6 show the user interface design. The burner right design is a reflection of the Burner Left design.

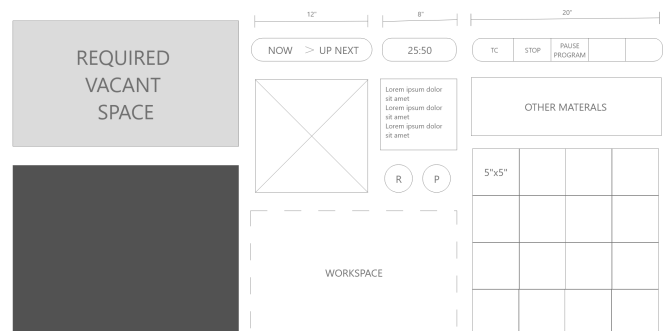


Figure 5: Table Interface, Burner Left

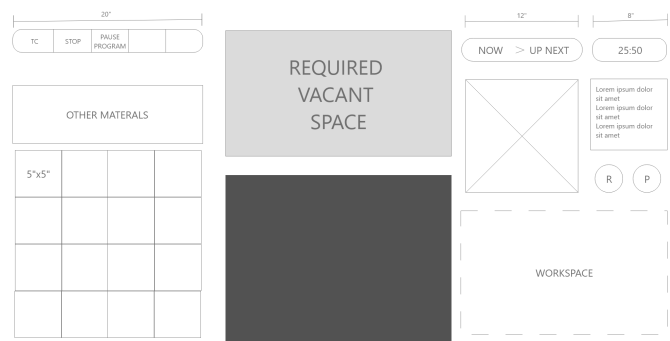


Figure 6: Table Interface, Burner Center

### 6.5.5 Interactions and Backend

The Flutter application will be launched when it received a command from the Command Processing mod-

ule to begin the application. Flask will be used to handle data to send to the Flutter application. Components controlled by user input are:

- **Video Controls.** The video can be paused or replayed, either by gesture (tapping on the button) or voice command.
- **Timer Controls.** The user will be suggested to use timers at certain steps. This timer will be displayed in as a widget. The user can pause the timer and resume it, either by gesture or voice command.
- **Ingredient Guidance.** After the calibration stage, each square in the grid will be labeled with the name of the ingredient. During each step, grids that contain ingredients required at that step will be outlined in green. Additionally, if the user picks the wrong item, the box will flash red, as an indicator that they should place the item back in the box and pick the correct ingredient.

### 6.5.6 Data Storage

Data that is preinstalled on the device are recipe videos and their corresponding instructional text. This data will be stored in an assets folder that has a subfolder for each recipe. Videos will be given an id in order of each step. A text file will contain the written instructions, with each written step labeled with the same corresponding video id. The Flutter application will obtain text content and video paths from this file.

### 6.5.7 Homography Calculations

The UI for the selected recipe must be processed via a homography calculation in order to account for the projector warping during projection. This is largely a result of deciding to project onto the table from the side of the table rather than hooking the projector directly above and perpendicular to the table. To account for the specific distance, height and angle ( $\theta$ ) down of the projection. The homography is calculated as follows:

$$\mathbf{H} = \begin{bmatrix} \frac{D}{D-H} & 0 & 0 \\ 0 & \frac{H}{D-H} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Multiplying each  $(x, y)$  coordinate in the frame by the homography will result in a new  $(x_p, y_p)$  to create the warped image for projection.

## 6.6 Societal Factors

### 6.6.1 Economic

TableCast offers a low cost alternative that can be efficiently packed.

### 6.6.2 Public\Health\Welfare

TableCast has a user-friendly interface that encourages beginners to cook and lead healthier lifestyles. It is designed such that it has a secure mounting mechanisms to ensure user safety.

### 6.6.3 Social\Cultural

TableCast allows users to easily attempt recipes from across cultures creating a space for better appreciation of the diversity that exists in food from all the world.

## 7 TEST & VALIDATION

### 7.1 Tests for TR1

**TR1. Projector mount must be adjustable.**

We plan to test this requirement by ensuring the selected tripod has several quantizable heights the user can manually adjust the projector height to while fully supporting the projector weight.

### 7.2 Tests for TR2

**TR2. Warped projection on table looks flattened**

This will be tested extensively by using standardized geometric shapes and quantitatively evaluating the error between the original image and a camera capture of the warped projection onto the countertop. This is also a stage in projector calibration so it will be tested in every new environment.

### 7.3 Tests for TR3

**TR3. Voice commands must recognize wake word to complete any command with an accuracy of 90%**

There will be 4 tests to ensure the voice commands meet the required accuracy. The includes (1) testing commands in environments with different noise levels, (2) testing different kinds of voices, (3) conversing without saying commands to check for false positives, (4) stress testing with repeated usage of commands within a short time frame.

### 7.4 Tests for TR4

**TR4. Camera must be mounted such that it points down and is 3-5 feet above the counter.**

Camera must be setup and measured such that it meets this requirement.

## 7.5 Tests for TR5

**TR5. Gesture recognition must have 90% accuracy.**

There will be 6 tests to ensure the gesture recognition commands meet the required accuracy. The includes (1) testing in different noisy environments, (2) different hands (3) literal testing of the commands repeatedly (4) making sure that similar gestures did not get picked up (5) d. Additional (5) stress testing so repeating commands one after another.

## 7.6 Tests for TR6

**TR6. The latency of the video instruction must be within 3-5 seconds**

The latency of the video at it's slowest, should be sufficiently low latency similar to a YouTube video. This measurement can be done by imaging the worst lab in the project.

## 7.7 Tests for TR7

**TR7. The gesture and voice commands must be recognized within 3 seconds.**

In order to determine the largest latency gap in understanding the voice/gesture commands, we will time any lag and buffering as they are used to improve recognition and reduce the compute load.

## 7.8 Tests for TR8

**TR8. User study must achieve 85% user satisfaction.**

We will invite five people to use our product. We will be using the System Usability Scale (SUS) [1] in order to assess the user's satisfaction with the overall product. SUS relies on a 1-5 scale for a range of 10 questions to evaluate the overall ease of use and required expertise.

the yellow indicates Shaik's tasks, and the red indicates Syeda's tasks.

## 8.2 Team Member Responsibilities

**Caroline:**

- Speech Processing
- User Interface
- Backend

**Tahaseen:**

- Projector Homographies
- Projector Calibration
- Hardware Integration

**Sumayya:**

- Gesture Recognition
- Object Detection
- Object Tracking

## 8.3 Bill of Materials and Budget

See Table 1 for the Bill of Materials. Most of our budget is allocated toward the projector because the quality of the projection is vital to the user experience. Other major components, such as the AGX and camera, are borrowed from the department.

*\*Note: These items are borrowed from the course inventory or other campus resources and therefore cost \$0.*

# 8 PROJECT MANAGEMENT

## 8.1 Schedule

Our schedule is being tracked using Atlassian Suite's Jira Software and was designed to complete our build process by the Interim Demo deadline of April 1, 2024. After this date, we plan to conduct our testing and verifications to ensure our product is functioning as anticipated. This is necessary to ensure that we can integrate all systems together in a timely and concise manner. All testing and verification will be complete by the Final Presentation date of April 24, 2024 We anticipate this will leave us enough slack to wrap up any loose ends and create a product that we are satisfied meets our outlined requirements. The schedule is shown in further depth in Fig. 10. The colored circles near each Jira issue, indicates the owner of the task. For ease, the blue circle indicates Crooks' tasks,



Table 1: Bill of materials

| Description   | Model #            | Manufacturer            | Quantity | Cost @   | Total           |
|---|--------------------|-------------------------|----------|----------|-----------------|
| *Arducam IMX219                                     | SKU:B0183          | Arducam                 | 1        | \$0.00   | \$0.00          |
| *NVIDIA Jetson AGX Xavier                           | 900-82888-0040-000 | Nvidia                  | 1        | \$0.00   | \$0.00          |
| *PlatinumPlus 5858D 58" Tripod                      | 620-585BB          | Sunpak                  | 2        | \$0.00   | \$0.00          |
| BOYA Wireless Lavalier Microphone                   | BY V20             | BOYA                    | 1        | \$39.95  | \$39.95         |
| *Mini External USB Stereo Speaker                   | 3369               | Adafruit Industries LLC | 1        | \$0.00   | \$0.00          |
| 4,500 ANSI Lumens SVGA Business/Education Projector | PA700S             | ViewSonic               | 1        | \$350.99 | \$350.99        |
|   |                    |                         |          |          | <b>\$390.94</b> |

## 8.4 Risk Mitigation Plans

The following risks were identified as primary risks and placed in a severity/consequence matrix in order to set priorities. The matrix can be seen in 9.

- R1. Camera falls due to poorly designed mounting. (Mitigation: redundant mounting security.)
- R2. Projector falls due to poorly designed mounting. (Mitigation: redundant mounting security.)
- R3. Camera is damaged by food splatter or any additional spillage. (Mitigation: cover camera with thin plastic wrap.)
- R4. User's gesture is ignored by chosen algorithm. (Mitigation: increase algorithm sensitivity.)
- R5. User's gesture is recognized incorrectly by chosen algorithm. (Mitigation: increase testing to meet required accuracy.)
- R6. User's vocal wake word is ignored by vocal command algorithm. (Mitigation: increase algorithm sensitivity.)
- R7. User's vocal wake word is recognized incorrectly by vocal command algorithm. (Mitigation: increase testing to meet required accuracy.)
- R8. Server hosting UI crashes in the middle of a recipe. (Mitigation: have a graceful fail state with safe system shutdown.)

## 9 RELATED WORK

There are few similar products to TableCast which make our application unique. The most common method of creating a large digital tabletop is by replacing the table with a large screen. However, we aim to make sure that TableCast is functional for most kitchen counters without requiring expensive replacement. On the other hand, there do exist a few table projectors in prototype phases in the 2010s, but these were never pushed to the market and/or do not cover a large table surface like we intend to do. The primary example for this is Sony's Experia Touch [4]. However, our product is specific to kitchen environments and guiding users through cooking on a large and chaotic environment. This is in contrast to Sony's small and simple environments.

## 10 SUMMARY

TableCast uses projections and computer vision to help anyone learn how to cook, personalized to their own kitchen. Recipes are segmented into individual steps, easy for the user to digest. Each step has both video and written steps, easy to see on the table. High quality projections will provide guidance to users, reactive to user actions. Users can have meaningful interactions with the interface, gestures and voice commands, enabled by computer vision and speech recognition software systems. TableCast enriches a user's kitchen experience, inspiring further interest in cooking.

There will be challenges in implementing this design. One such challenge is developing the calibration step. The goal of the calibration step is to meet the requirement of being able to use the device on variable sized tables. Another notable challenge is object tracking, particularly detecting the same object between frames. There will be much testing and fine tuning involved in this step so that the algorithm can detect all of the objects it needs to track throughout the duration of the entire recipe. Although these tasks are challenging, the goals and requirements are still reasonable, given the time and resources allocated. TableCast is an innovative tool used to make cooking easier and enjoyable for people of various backgrounds. Our technology is applicable beyond the kitchen where projections, computer vision and voice processing come together to create an seamless user experience.

## Glossary of Acronyms

Include an alphabetized list of acronyms if you have lots of these included in your document. Otherwise define the acronyms inline.

- UR# - User Requirement
- PR# - Projector Requirement
- DR# - Device Requirement
- IR# - Implementation (Design) Requirement
- P2P - Peer-to-Peer
- AGX - Naming for Tegra chips from Nvidia

- SUS - System Usability Scale

## References

- [1] John Brooke. “SUS: A quick and dirty usability scale”. In: *Usability Eval. Ind.* 189 (Nov. 1995).
- [2] Kien Nguyen Phan et al. “Assessing Bicep Curl Exercises by Human Pose Application: A Preliminary Study”. In: Mar. 2023, pp. 581–589. ISBN: 978-3-031-27523-4. DOI: 10.1007/978-3-031-27524-1\_55.
- [3] Akshara Pande et al. “A Comparative Analysis of Real Time Open-Source Speech Recognition Tools for Social Robots”. In: *Design, User Experience, and Usability*. Ed. by Aaron Marcus, Elizabeth Rosenzweig, and Marcelo M. Soares. Cham: Springer Nature Switzerland, 2023, pp. 355–365. ISBN: 978-3-031-35708-4.
- [4] Nick Statt. *Sony’s prototype projector turns any tabletop into a touch-sensitive display*. The Verge, Mar. 2016. URL: <https://www.theverge.com/2016/3/13/11215454/sony-interactive-projector-future-lab-sxsw-2016> (visited on 03/01/2024).

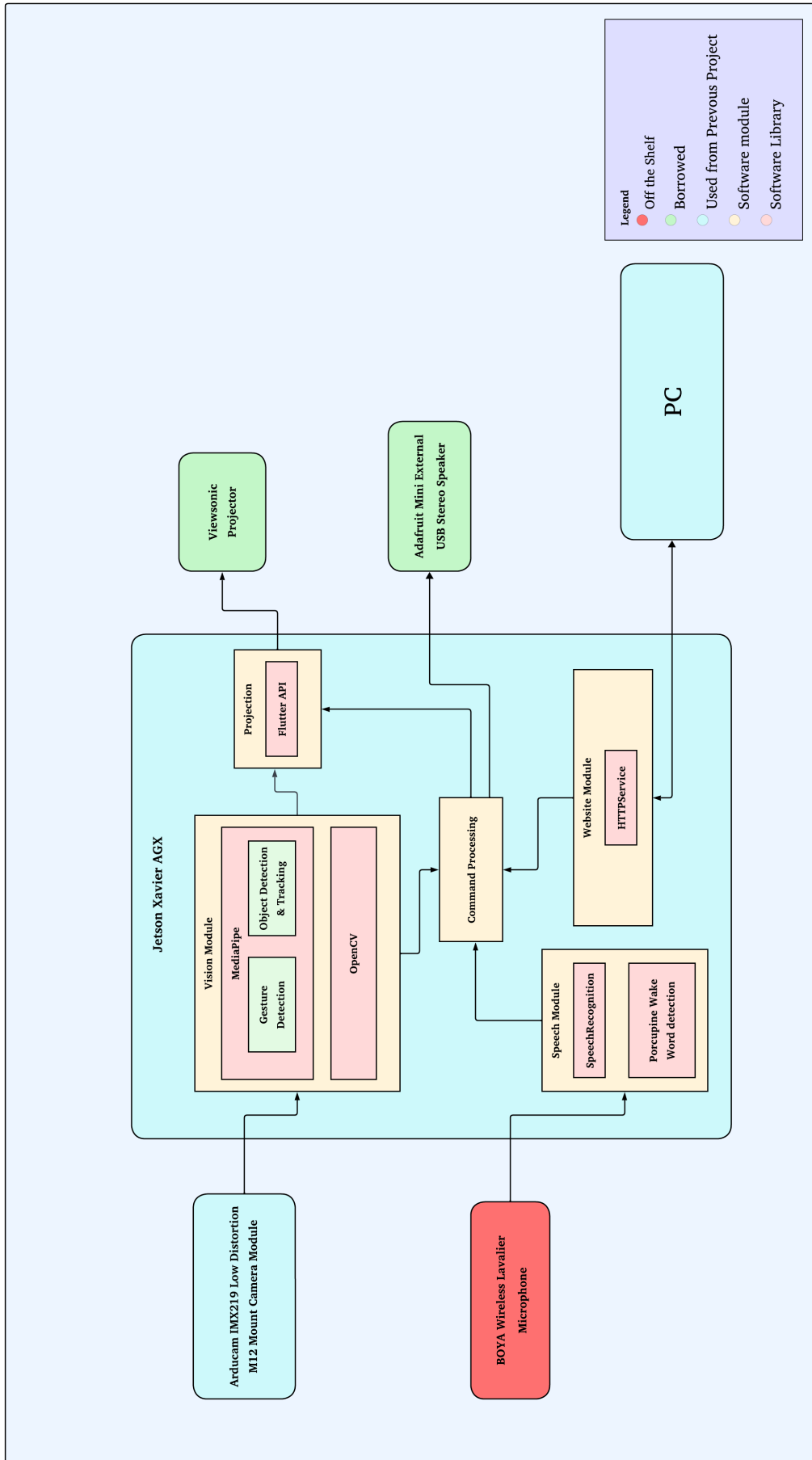


Figure 7: A full-page version of the same system block diagram as depicted earlier.

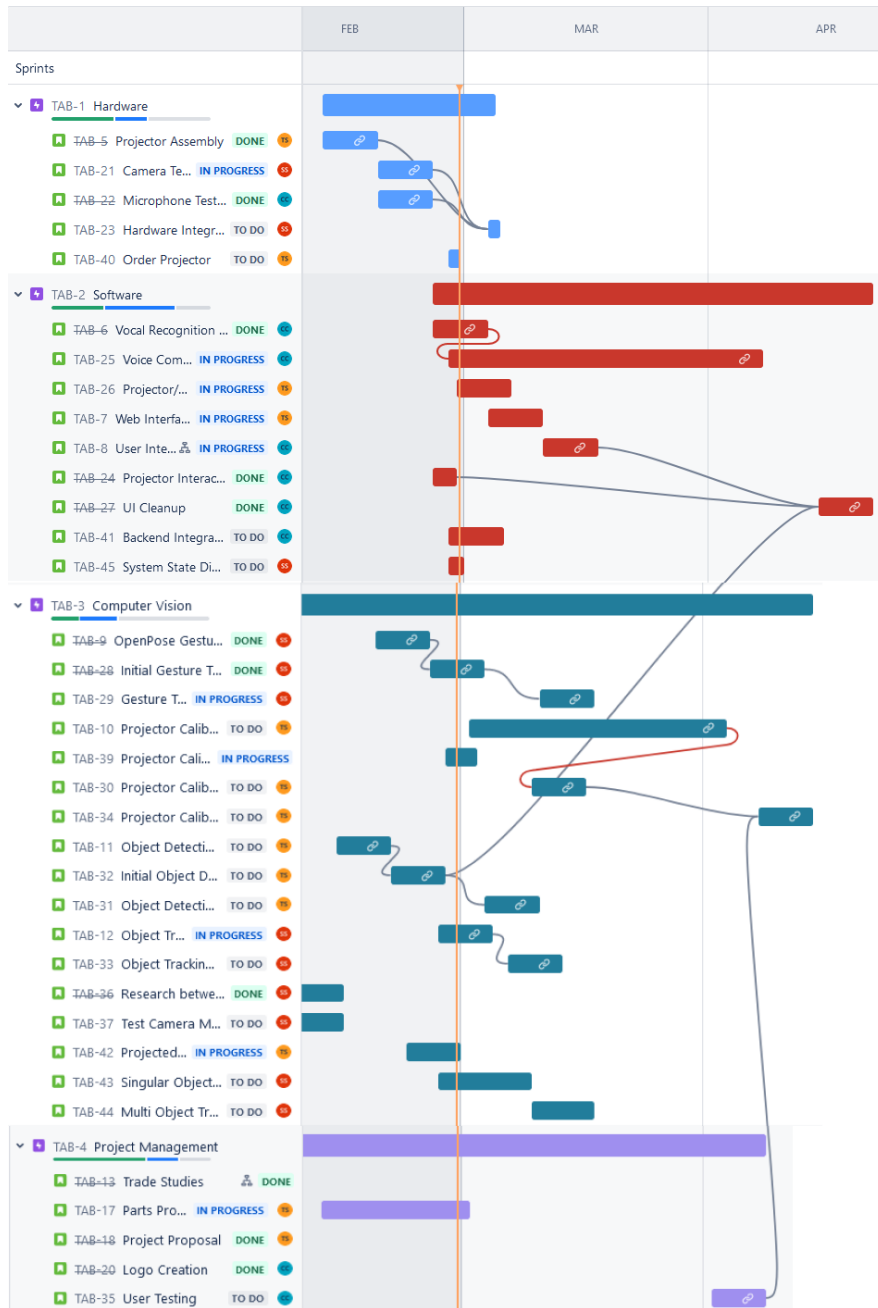


Figure 8: Gantt Chart

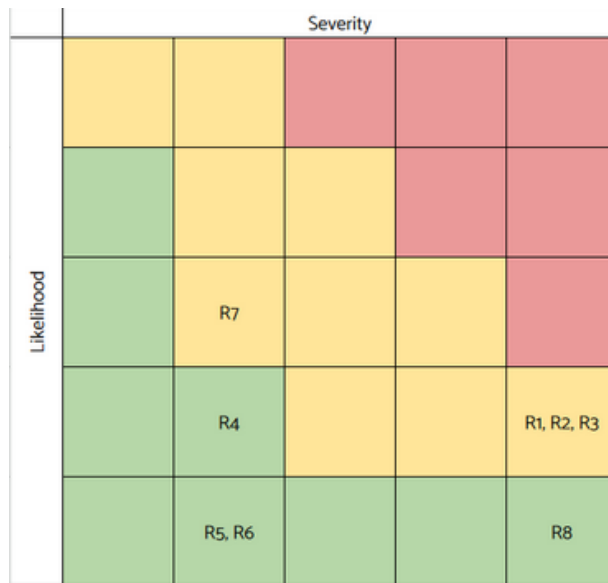
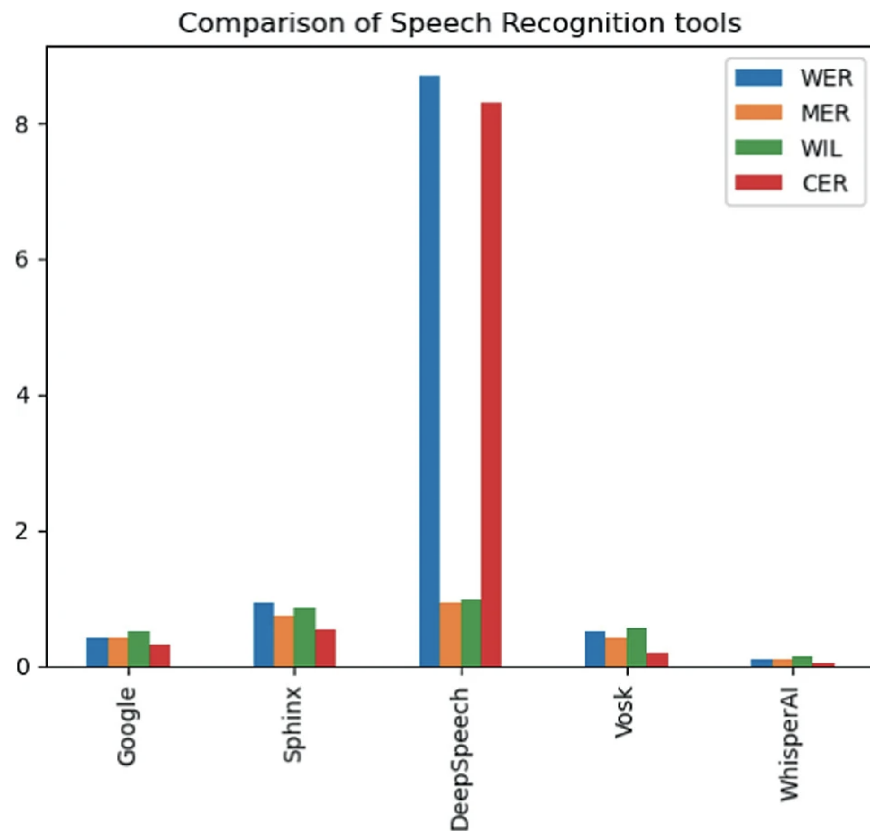


Figure 9: Risk Matrix for Identified Risks

**From: A Comparative Analysis of Real Time Open-Source Speech Recognition Tools for Social Robots**



Comparison of speech recognition tools based on evaluation measures

Figure 10: Open Source Speech Recognition Error Rate Comparisons (Pande, et al.)