



# HermanMiller

Team Herman Miller  
Michigan State University  
AR Adjust App  
Project Plan  
Spring 2018

**Herman Miller Contacts:**

Matt VanOpstall

**MSU Capstone Members:**

Mike Bremiller

Kevin Gaban

Han Huang

Kyle Kinsey

Jacob Weber

# Table of Contents

## Contents

Table of Contents .....	2
Executive Summary .....	3
Functional Specifications .....	4
Design Specifications .....	5
Chair Detection Interface .....	5
Interactive Adjustment Interface .....	6
Technical Specifications .....	9
Software Technologies .....	9
Image Recognition .....	9
System Architecture.....	9
Risks .....	11
Recognizing Chair Models via Camera .....	11
Minimizing Application Size .....	11
Multiple Configuration Testing .....	11
Cross-Platform Variation .....	12
Schedule.....	13

## Executive Summary

Since its founding in 1905 in Zeeland, Michigan, Herman Miller has been the industry standard in office furniture innovation. In 1992 they introduced the Aeron chair, which quickly became the most popular ergonomic chair of all time. Herman Miller has since introduced a remastered version of the Aeron chair, as well as other successful lines including Embody, Celle, and Mirra. These chairs are precisely engineered and made-to-order to provide an unparalleled level of comfort and personal configuration to users.

With an advanced level of configuration comes a more involved setup process for consumers who want proper ergonomic support. Herman Miller currently provides resources to help with configuration via in-person training seminars led by ergonomic experts. These seminars, while interactive and useful, are time-consuming, and many of the nuances of the specific adjustments can be quickly forgotten. Herman Miller seeks to solve these problems with Adjust AR: a mobile app for its customers.

Adjust AR uses “augmented reality” to identify a user’s chair through the camera and provides specific adjustment instructions and information via a virtual model of the user’s chair. This allows the user to sit in their physical chair and make appropriate adjustments while being guided by the AR model. The app serves as a persistent reference for the user, with no need for seminars or paper / online reference texts.

## Functional Specifications

Along with the proper training, Herman Miller ergonomic office chairs mitigate the negative effects of sitting for extended periods of time. However, it is very common for users to misuse or altogether ignore these ergonomics features. Historically, Herman Miller has led training sessions in person, with industry experts giving demonstrations and information to users. Today, the increasing popularity of ergonomic solutions coupled with high turnover rates at many organizations has made it increasingly difficult to keep consumers up-to-date and properly informed about correct usage of their products.

The Adjust app seeks to solve this problem by providing near-instantaneous resources and support for consumers. The application is designed to be a supplement to existing in-person ergonomic training sessions. After the initial training provided by a Herman Miller representative, the user can interact with their chair in the application and make any adjustments at their leisure. In addition, the application is especially useful to new employees who may miss company-wide training seminars.

To accomplish this, the Adjust app is designed to provide adjustment instructions for any user of Herman Miller office chairs. The native iOS app uses the iPhone camera and computer vision technology to identify the model of the user's chair. The user can then choose to begin a guided ergonomic adjustment session. In the guided ergonomic adjustment session, the user is shown an interactive 3D model of their chair, with the current part highlighted. Tapping on the part gives them adjustment instructions and tips for proper ergonomic setup.

If a user already knows what chair they have, or simply wishes to explore all models, there is an additional Gallery feature. This feature allows a user to interact with any supported model and use a search functionality to find parts by name or function.

Additionally, the app can store useful analytics for Herman Miller, including data about location, usage time, and statistic for each individual model. This information can be leveraged across the company for sales, marketing, and engineering purposes.

## Design Specifications

The design of the Adjust app revolves around two core components: the chair detection interface, and the 3D chair adjustment interface. Upon entering the app, the user can use the forward camera to detect their model of Herman Miller office chair, and begin a guided ergonomic adjustment session using a 3D representation of their chair.

If the user has already identified their chair, or wishes to explore the components of any model, the application also provides a Gallery interface. The Gallery allows users to view a 3D model of any supported Herman Miller chair, with an option to begin a guided adjustment session or to view all adjustments in any order. Additionally, a search functionality provides the user with the ability to search for a part by name or by functionality and have it dynamically highlighted on the model.

### Chair Detection Interface

By default, the app will open to the chair detection interface. This section prompts the user to point the camera at their chair to identify the model. When the user is ready, they are prompted to press the “Detect” button (see Figure 1). After the button is pressed, the image detection algorithm is run for the next 2 seconds.

Each frame of the camera input is run through the CoreML model, which outputs a “best guess” and a confidence level for that prediction. After the detection is finished, the app aggregates the predictions over the detection interval, and if all the predictions are the same and pass the confidence threshold (in our case, 95%), then the chair will be identified by name on screen (see Figure 2). If the chair could not be identified, or did not pass the confidence threshold, then a message indicating that “no chair was detected” will appear, and the app will re-prompt the user to detect again.



FIGURE 1: CAMERA INTERFACE



FIGURE 2: CAMERA INTERFACE WITH CHAIR DETECTED

Once a chair is successfully identified, the “Detect” button is replaced with the “Adjust” button. When the user presses the “Adjust” button, a prompt appears which asks the user if they would like to begin a guided adjustment session. Pressing “yes” will take the user to the interactive adjustment interface and starts a step-by-step ergonomics session, while pressing “no” will highlight all the parts, allowing the user to adjust them freely.

If the user prefers to not use image recognition, they can instead choose to select a chair from the gallery (see Figure 6).

## Interactive Adjustment Interface

Upon pressing the Adjust button or choosing a model from the gallery, the app will prompt the user to sit in their chair. At this point, an interactive 3D representation of the chair will appear on-screen, which can be rotated via user input. A guided ergonomics walkthrough begins for

the user by telling them which adjustment should be made and highlighting it brightly on the chair for them. To help the user easily locate the current adjustment, the chair is rendered slightly transparent, allowing the adjustment to be seen from any orientation (see Figures 3 and 4).



**FIGURE 3: GUIDED ADJUSTMENT SCREEN 1**



**FIGURE 4: GUIDED ADJUSTMENT SCREEN 2**

Tapping on the highlighted part will open a screen with details about the selected part. At the top of the screen is the name of the part, followed by a video of the adjustment. These videos provide a quick, visual demonstration of the adjustment. Below the video is a quick description of any ergonomic “best practices” for this adjustment, typically describing how the user should be positioned after performing the adjustment. Finally, a text description of how to perform the adjustment is also included (see Figure 5).

During the guided adjustment walkthrough, the user navigates forward and backwards between parts using the “Next” and “Back” buttons, respectively. Due to the advanced level of configurability on some models, the users chair may not have all the configuration options as

the model; in this case, the user can simply use the “Next” button to skip to the next step in the adjustment process.

Additionally, the user can press the “Back to Camera” button to return the detection interface to choose another chair. To view all the adjustments for the current chair, the user can simply respond “No” when prompted for a guided session, or they can access the Gallery from the hamburger menu.

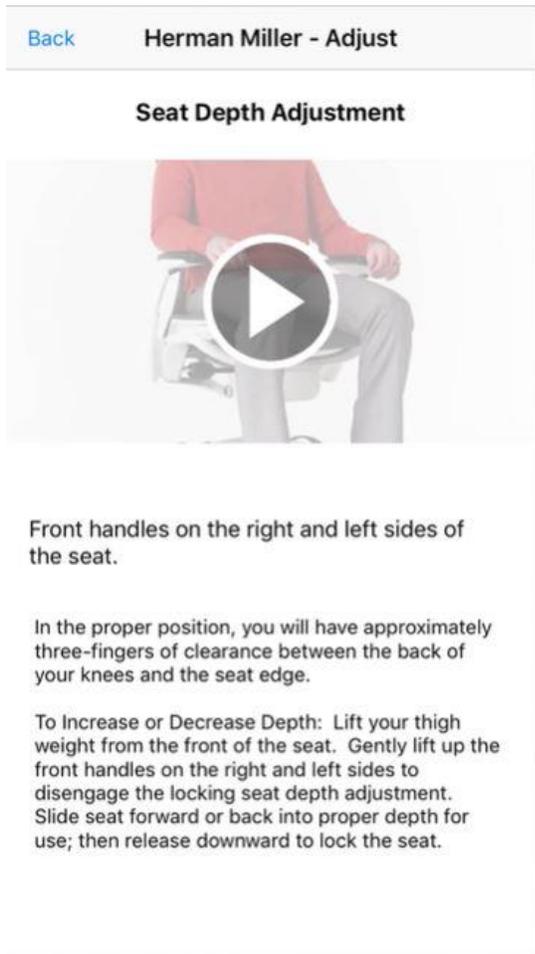


FIGURE 5: ADJUSTMENT DETAIL SCREEN

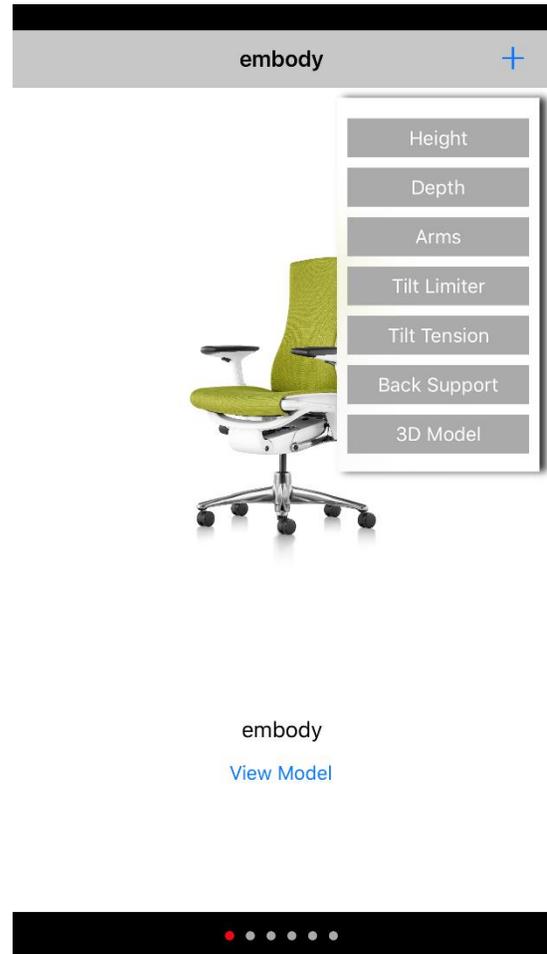


FIGURE 6: CHAIR GALLERY SCREEN

Allowing the users to switch between the guided ergonomic tutorial is useful for the situations in which the client knows what adjustment they need to make, but simply needs to locate it. Additionally, the Gallery option is a convenient alternative for users who already know which model they are using, or who have previously identified their chair.

The Gallery features screenshots of every chair currently supported in the app. Clicking on the “view model” button brings up the guided adjustment session for the selected chair. Alternatively, the user can press the small plus button to view a list of attachments for the

selected chair. Choosing one of these attachments takes the user to the 3D model with the selected part highlighted, allowing them to easily find the part they are looking for.

## Technical Specifications

### Software Technologies

- **Xcode**
  - The Adjust app was developed entirely in Xcode and written in Swift to create a responsive native iOS application.
  - Swifts built-in “property lists” are used to store local static information, including details about the chairs and adjustment instructions.
- **SceneKit**
  - SceneKit is a built-in library for Swift which is used to render the interactive 3D models in the application. SceneKit renders COLLADA (.dae) files, and uses the metadata stored in them to dynamically render all of the parts of the chair as one singular model, while still retaining each pieces information.
  - Blender, an open-source 3D modelling software, is used to modify the 3D model before being imported into SceneKit.

### Image Recognition

- **Microsoft Custom Vision**
  - Cloud-based computer vision and machine learning service. To utilize Custom Vision in the Adjust app, hundreds of labelled photos of each chair are uploaded to the service, which outputs a machine learning model (in the form of CoreML, detailed below).
- **CoreML**
  - CoreML is a machine library created by Apple. In our application, the CoreML model that is output by Custom Vision is imported into Xcode as its own class, takes in still frames from the camera input, and outputs a prediction for one of the available tags.

## System Architecture

The core system architecture of Adjust is based on two key components: The Image Recognition and the Interactive 3D Model. These components are tied together with the rest of the application using Swift (see Figure 7).

The image recognition component starts with Microsoft Custom Vision. When an image of a chair is uploaded to Microsoft Custom Vision, it is tagged with the name of an existing chair, and Custom Vision creates a new machine learning model to classify future images. By uploading hundreds of pictures of each chair (taken from different lighting setups, angles, and colors), the model can distinguish between any of the supported chairs.

Custom Vision then exports a CoreML model for use in the application. The model is exported as a custom Swift class, which takes in an image and returns its “best-guess” and a given confidence level. This guess is used as the identification for the chair.

On the other side of the application is the functionality to display an Interactive 3D Model. To accomplish this, 3D renderings of each Herman Miller chair were used. Each model was modified by grouping the polygons of each adjustment into their own “layer”. These models, which are imported to Xcode as COLLADA files, are then rendered using Apples SceneKit technology. Scenekit loads the 3D model into the view, and can detect what part is being touched by referencing the metadata of each layer.

Both components are tied together using Swift in Xcode, performs the rest of the application logic, as well as creates all the User Interfaces.

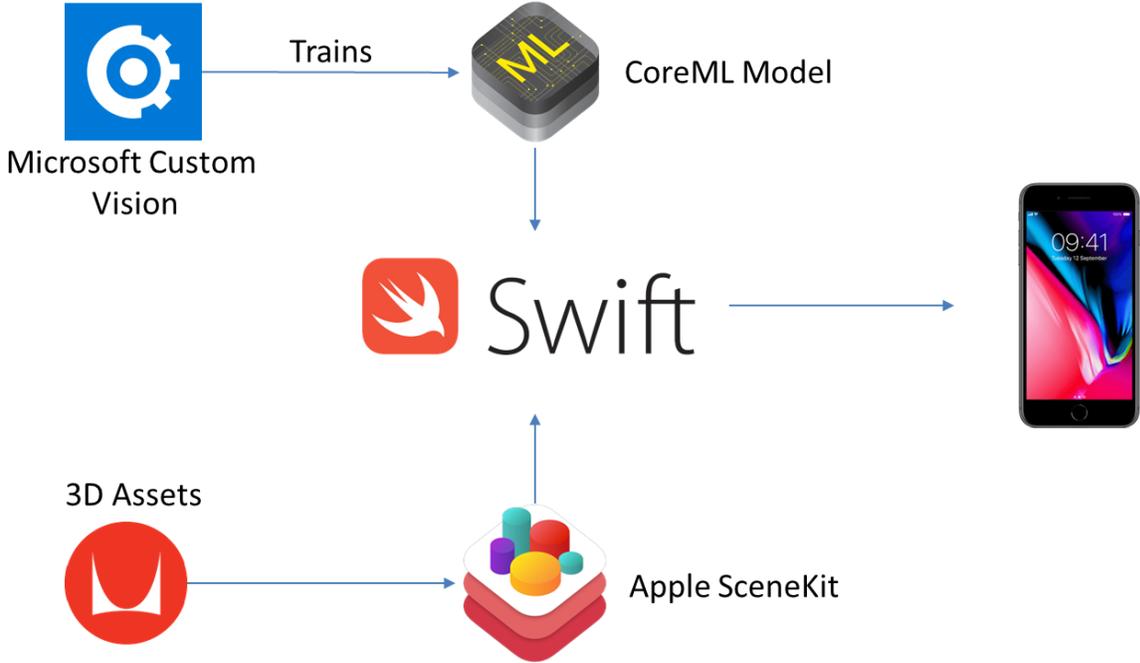


FIGURE 7: SYSTEM ARCHITECTURE DIAGRAM

## Risks

### Recognizing Chair Models via Camera

**Difficulty:** Hard

**Description:** The ability for the user to identify their chair using the mobile app. Recognition is done using the camera from the device being used. The app must not only be able to correctly identify the model of Herman Miller chair out of the set of possible models, but also reduce “false positives” of other chairs and of scenery.

**Mitigation:** Microsoft Custom Vision and CoreML are used to perform image recognition. Custom Vision’s web interface allows simple uploading and tagging of target images, and can be exported with one click, and then loaded into XCode. To reduce false positives, the detection process is started with a button press. After running for 2 seconds, the application aggregates predictions from every frame of video input, and chooses the model with the highest confidence over the entire interval.

### Minimizing Application Size

**Difficulty:** Medium

**Description:** For each of the Herman Miller chairs there are multiple assets required to properly provide the ergonomic tutorial. Each chair has a 3D model that ranges from 4 to 7 MB. Each adjustable attachment on a chair requires a video tutorial and text description. The videos for each chair can add up to 100 MB. The size of the app will grow quickly as we add more chairs.

**Mitigation:** One solution to this problem is moving the videos to a hosting platform such as YouTube. This will drastically reduce the size. However, our client has requested that we do not host the video online due to branding and intellectual property issues. We have also discussed the possibility of changing the videos to an animated gif format.

### Multiple Configuration Testing

**Difficulty:** Easy

**Description:** Each one of Herman Miller’s chairs comes in several configurations, which may or may not include certain adjustments. These may be difficult to visually identify, or to handle during the guided adjustment tutorials.

**Mitigation:** Herman Miller has provided the team with two fully-loaded models. This provides unlimited access to pictures of the adjustments, in addition to the photos captured during on-site visits. Human input can also be used to verify the existence of an adjustment.

## Cross-Platform Variation

**Difficulty:** Medium

**Description:** Planning for potential future Android port. If the app is eventually expanded to Android, substitutions for CoreML and SceneKit needed to be developed, and the interface must be recreated for Android.

**Mitigation:** Microsoft Custom Vision is able to export a Tensorflow model, which is compatible with Android. This means that there is no new “retraining” of the model, and it stays consistent across both platforms. Additionally, a plan is being drafted for future developers on how an Android application could be completed.

# Schedule

## Week 1 (1/9 – 1/15)

- Start Project, initial meeting with group members
- Onsite Herman Miller Visit
  - Visit Herman Miller Greenhouse Site
  - Tour chair facility and learn about project goals
  - Define scope of project with client contacts
- Research AR and Image recognition technologies

## Week 2 (1/16 – 1/22)

- Begin work on Unity3D UI
- Begin work on project plan presentation
- Test Vuforia Model Target Generator
- Begin work on 3D models in Unity3D

## Week 3 (1/23 – 1/29)

- Continue work on UI
- Set up JIRA and OMNI from Herman Miller
  - 1<sup>st</sup> Sprint begins
- Continue work on Project Plan Document
- Research alternatives for Vuforia
  - Microsoft Custom Vision
  - CoreML / TensorFlow
  - 8<sup>th</sup> Wall
  - ARKit

## Week 4 (1/30 – 2/5)

- Prepare for the Project Plan Presentation
- Prototype with Microsoft Vision and CoreML
- Experiment with XCode for native iOS
- Image recognition working on iPhone
- Prototype Apple SceneKit for interactive 3D models
- Modify Embody and Aeron interactive models

## Week 5 (2/6 – 2/12)

- 1<sup>st</sup> Sprint ends, 2<sup>nd</sup> Sprint planning
- Pivot from Vuforia and Unity to Native iOS and CoreML
- Present Project Plan

- Begin building CoreML model using Microsoft Custom Vision
- Add video player functionality to app
- Added interactive Embody and Aeron models

### Week 6 (2/13 – 2/19)

- Work on Alpha Demonstration
- Work on Alpha Presentations
- Customized User Interface to Herman Miller branding
- Added detailed descriptions of Embody and Aeron adjustments
- Created local information storage using built-in Swift Property Lists
- Team Status Report – Mitigation of Risks

### Week 7 (2/20 – 2/26)

- 2<sup>nd</sup> Sprint ends
- Expand Gallery Interface
- Update adjustment descriptions
- Added part descriptions to property list
- Create TestFlight deployment for Beta Testing
- Alpha Presentation

### Week 8 (2/27 – 3/5)

- Client visit at Herman Miller Design Yard
  - Present to client contacts and VP of Seating
  - Feedback on use cases and UI / UX
- Begin setup of Firebase for Analytics
- Update flow of Guided Adjustment tutorial
- Consider animated adjustments within tutorial space

### Week 9 (3/6 – 3/12)

- Spring Break

### Week 10 (3/13 – 3/19)

- Implement detect button (vs. auto-detect)
- Automatic rotation to current adjustment
- Animated rotation

### Week 11 (3/20 – 3/26)

- Improve Gallery features
  - Option to focus on any listed part
  - Pull-up menu for each chair
- Prepare for Beta Presentation

- Implement analytics

### Week 12 (3/27 – 4/2)

- Beta Presentations
- Expand Analytics Features
- UI Modifications

### Week 13 (4/3 – 4/9)

- Present at Herman Miller
- Bug fixes and modifications from TestFlight feedback
- Project Video Begin

### Week 14 (4/10 – 4/16)

- Add Herman Miller branding (fonts, images)
- Final TestFlight deployments
- Project Video

### Week 15 (4/17 – 4/23)

- Project Videos Due
- App finalization
- Finish Branding and finalize assets

### Week 16 (4/24 – 4/30)

- Design Day
- Design Day Setup
- Project Deliverables Due

### Week 17 (5/1 – 5/3)

- Videos Shown in All-Hands on Meeting
- Project Completion