1) Our goal for Astropath is to have a one-stop-shop application for astrophotography. We hope to be able to edit, stack, and generate timelapses of a user's star pictures. We will have a simple interface where a user can upload and select images and then utilize our editing suite to generate professional grade images.

2) **Tyler** - Team Lead - Tyler is a Senior in Computer Science who hopes to graduate in December of 2025. He is interested in Astronomy, landscape photography & enjoying his free time in the backcountry snowboarding.

   Greg - Frontend Lead - Greg is a senior in Computer Science who hopes to graduate in May of 2025. He Is interested in astronomy, Bitcoin, and all outdoor activities.

   Nolan - Backend Lead - Nolan is a junior in Computer Science who hopes to graduate in May of 2026. He is interested in astronomy, Fortnite, all things sports, and weightlifting.

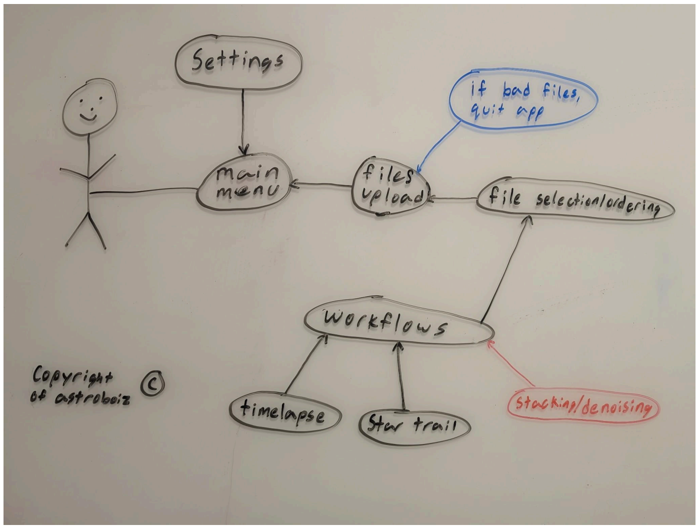
3) **What we plan to get done:**
   Astropath will be a streamlined, user-friendly astrophotography tool designed to simplify the creation of star trails and starlapses. By April 2024, our goal is to deliver an application that enables users to batch import TIFF images, quickly select and deselect images for processing, and apply automated processing modes like gap filling, lighten, and darken to produce clean star trail effects.

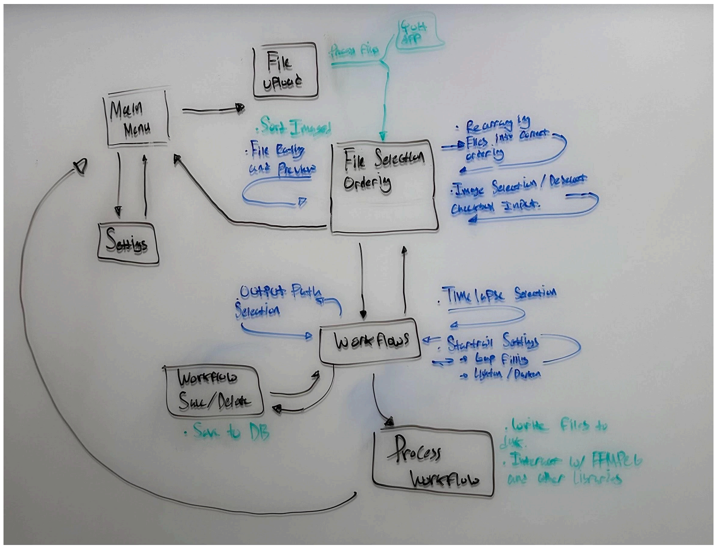
   Essential features will include a Lightroom-style interface for easy image selection, basic batch import functionality focused on TIFF files, and preset options for users to save and quickly access their preferred processing configurations. *Astropath* will also include a basic starlapse creation feature, allowing users to convert sequences of night-sky images into time-lapse videos. This version will emphasize efficient workflows for astrophotographers, allowing them to create polished visual effects with minimal manual intervention and without the need for additional tools.

   Expanding RAW file support (such as ARW) will be considered for future releases, with TIFF-focused workflows prioritized to ensure quality and timely delivery. The MVP will not include stacking, focusing instead on delivering an intuitive, reliable, and high-quality tool for astrophotography enthusiasts by our target date.

   - Image Upload
   - File Conversion. (TIFF)
   - Image Selection & Sequencing
   - Automated Star lapses and Star trails
   - Packaging For (Windows & Mac)

4) **Systems overview**





Uploading Systems:

- During file upload, Astropath will check each file dimensionality and verify filetype.
- TIFF Conversion. If a file is (.ARW/JPEG) -> Convert to .TIFF
- If files are not a known format, or any dimensions don't match then Exit with error message to user.

File Selection & Ordering System

- The user can select & deselect files (checkbox) which will be included in the batch processing stage.
- File sequencing. The user can select the proper ordering of the files which can then be the ordering of the timelapse.

Workflows System

- The user can select Workflow from the list of workflows.
- The user can select Workflow options like startrail(darken/lighten)
- The user can select a lapse mode (Whether to create copy files for lapse files or just create the lapse and then delete the temp files).

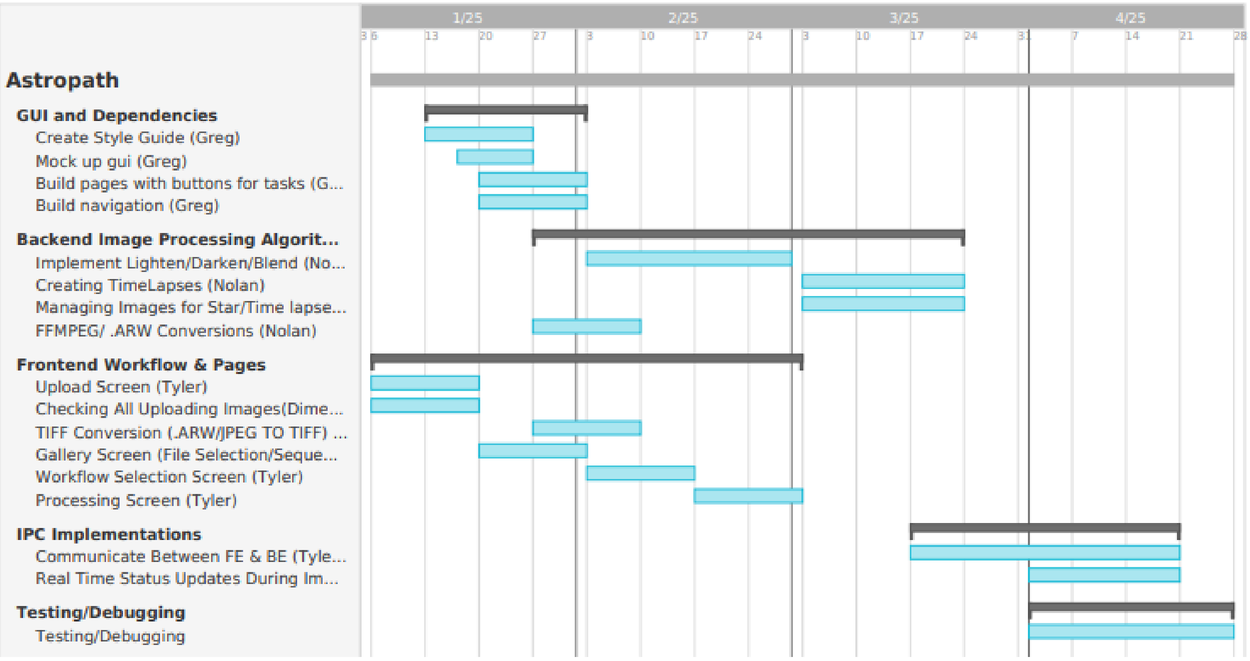
Processing System

- During image processing, show progress in a user friendly way. This includes an accurate description of what task is being completed and what percentage is done (eta if possible?)
- Show the image as it’s being processed. A progress bar is cool, but showing the image as it’s being built is nifty.

5) **Major MIlestones**

- GUI & Dependencies
  - This milestone includes basic setup and design of how our app is going to work. We will know that this is done when we are ready to add functionality to the buttons and we are able to navigate from page to page.
- Backend Image Processing Algorithms
  - This milestone is what controls the functionality of the app processes. This includes algorithms to change the brightness of images and create our StarLapses and StarTrails. We will know that this milestone is done when we put images into our editing functions and they are returning properly processed images
- Frontend Workflow & Pages
  - The User can navigate between each of the pages of the application. The UI should have a clean UX and provide a minimalistic feel.
- IPC Implementations
  - The implementation of the IPC, should allow the FE(Frontend) and BE(Backend) to communicate with each other with no limitations. This means the Go backend can communicate with the Typescript frontend. Wails will handle the conversion of types and runtime support, however additional data structures may be required to facilitate real time display of processed images.

6) **Major tasks to meet milestones**
   a) GUI & Dependencies
      i) Come up with style guide (Greg)
      ii) Use Figma to mock up GUI (Greg)
      iii) Build pages with buttons for tasks (Greg)
      iv) Build navigation (Greg)
   b) Backend Image Processing Algorithms
      i) Implement Lighten/Darken/Blend (Nolan)
      ii) Creating TimeLapses (Nolan)
      iii) Managing Images for Star/Time lapse (Nolan)
      iv) FFMPEG/ .ARW Conversions (Nolan)
   c) Frontend Workflow & Pages
      i) Upload Screen (Tyler)
      ii) Checking All Uploading Images(Dimensions/Formats) (Tyler)
      iii) TIFF Conversion (.ARW/JPEG TO TIFF) (Tyler)
      iv) Gallery Screen (File Selection/Sequencing) (Tyler)
      v) Workflow Selection Screen (Tyler)
      vi) Processing Screen (Tyler)
   d) IPC Implementations
      i) Communicate Between FE & BE (Tyler)
      ii) Real Time Status Updates During Image Processing (Tyler)



7) **Stretch goals**

Our stretch goal is to get long term image stacking and denoising algorithms to work. This is a challenge because the earth’s rotation causes the stars to move relative to the landscape. Another goal is to get a unique way to display real time star streaking during the processing stage. Instead of showing a simple progress bar, showing a real time visualization of the image being built is a result we aim to have done.

A last stretch feature which we would like to implement but are not sure if we can, is the ability to give an accurate estimate to processing to the user. The idea is to use the user’s previous computations combined with the real time results to produce an estimate for the remaining time. This can be done, but the question is regarding how accurate we can make it.