**Decisions:**

* I decided to use Xcode as my development environment since it is the official IDE for developing iOS apps.
* I chose to use Swift as my programming language since it is the primary language for iOS development.
* I decided to use the TMDB API as my source of movie data since it is a popular and reliable source of movie information.
* I chose to use a third-party library like CocoaPods to install and manage the TMDB API client library to make API requests.

**Challenges:**

* One of the challenges I faced was handling the asynchronous nature of API requests. When making API requests, the data is not immediately available, so I had to use closures, completion handlers, or async/await to wait for the response before proceeding with displaying the data.
* Another challenge I faced was displaying the movie poster images in the search results and detail views. The TMDB API provides URLs for the poster images, but I had to download and cache the images to display them in the app. I used third-party libraries like Kingfisher to simplify this process.
* I also faced a challenge of designing the user interface that is intuitive and easy to use. I had to balance the visual appeal of the app with its usability to ensure a good user experience.

**How I overcame them:**

* To handle asynchronous API requests, I used closures and completion handlers to wait for the response before updating the UI. I also used async/await when available to make the code cleaner and more readable.
* To display movie poster images, I used third-party libraries like Kingfisher to download and cache the images. This made it easier to display the images in the app without having to worry about downloading and caching them manually.
* To design the user interface, I used Apple's human interface guidelines as a reference and conducted user testing to ensure that the app was easy to use and intuitive. I also solicited feedback from colleagues and other developers to improve the UI design.