In creating my app, the goal was to have a pipeline process for the user to follow. To start, we are presented with the landing page:

Graphical user interface, application

Description automatically generated

The user is presented with a small prompt, and a button to view the available course list for selection. Upon pressing the button, the user is directed to a new activity using the Intent process.

The next activity is a course selection menu:

Table

Description automatically generated

The list was populated using recyclerView, as well as an adapter and data class to contain the data for each course.

The data structure of each Course object is:

* CourseID: the course code, ex. CS161
* CourseTitle: the full course title, ex. Intro to programming
* Prereq: required prior courses, ex CS161
* Term: term course takes place in, ex 1

Course’s checkboxes can be checked to indicate the choice to enroll in each course, and when this is finished the floating action button labelled “Submit” can be pressed to proceed to the next activity, where courses registered in are displayed.

Note: the selection must pass the desired constraints:

* Prerequisite must be met for each course
* Each term must have 3 or less courses

This is combined with the initial conditons of:

* CS161, CS162, Math101 already completed
* CS255, CS263 already registered

The selected courses are passed to the course schedule activity, and an intent with extras is passed.

Test cases:

1. Registering for CS277:

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Shown success and displayed in next activity Course Schedule along with preregistered courses.

1. Registration in CS356:

Graphical user interface

Description automatically generated

Error message displayed stating missing prerequisite, next activity of Course Schedule not shown.

1. Registration for CS225:

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Shown success and displayed in next activity Course Schedule along with preregistered courses.

1. Testing course overload error:

Graphical user interface, application

Description automatically generated

1. Full courseload populated:

Graphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generated

No errors, shows all registered courses correctly.

Difficult parts:

* Intents, specifically, extras. Was tricky to make typing work
* Adapters and recyclerView had many startup functions that were hard to figure out at first
* Handling errors efficiently in the handleCourses function.

Design Cues:

* Changed colors to STFX colors, seemed fitting
* Pipeline approach made process feel very step by step, more intuitive
* Scrollable list for courses made the most sense with amount of courses given; had it been the entire list I would have added dropdowns for each department and a search bar to quickly find courses
* Checkbox system instead of “ADD” buttons made more sense, as it’s faster to select all wanted courses at once rather than one by one
* Would like to turn course list items red if they caused an error if given enough time
* Would like to create to separate columns for each term of courses, again not enough time
* Floating action button was ideal, so you don’t have to scroll to bottom to submit choices.

Main intution of design was to create something that could be completed quickly, and felt step-by-step; all to make user experience more natural and intuitive.

If you have any issues running the app or if my submission formatting wasn’t correct, please let me know 😊

Jay