Degree Designer

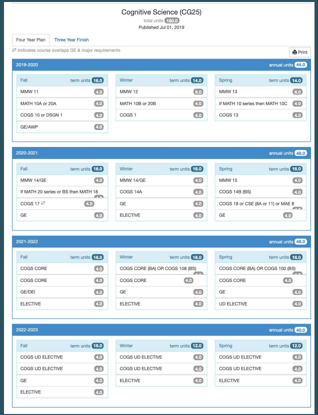
Created by Quaranteam

Nice to meet you!

- Our team:
    - Aneesha Ramaswamy (CSE)
    - Aven Huang (Cognitive Science)
    - SK Bost (Computer Science and Design)
    - Zoe Wong (Cognitive Science)
    - Mentor: Sydney Wong (Computer Science)

The problem

- It's often stressful and confusing to plan your classes
- Involves many different sources that aren't all easy to use
- This summer, we wanted to tackle this problem by designing an app!



Cognitive Science (CG25)
total units 180.0
Published Jul 01, 2019

Four Year Plan | Three Year Finish

⇄ indicates course overlaps GE & major requirements
Print

2019-2020 annual units 44.0

Fall (term units 16.0)		Winter (term units 14.0)		Spring (term units 14.0)	
MMW 11	4.0	MMW 12	6.0	MMW 13	6.0
MATH 10A or 20A	4.0	MATH 10B or 20B	4.0	If MATH 10 series then MATH 10C	4.0
COGS 10 or DSGN 1	4.0	COGS 1	4.0	COGS 13	4.0
GE/AWP	4.0				

2020-2021 annual units 48.0

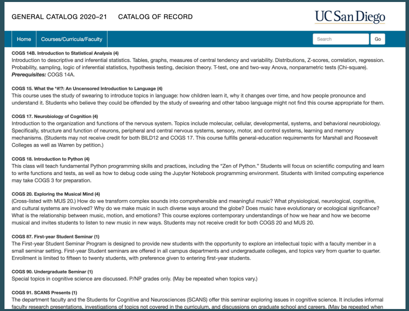
Fall (term units 16.0)		Winter (term units 16.0)		Spring (term units 16.0)	
MMW 14/GE	4.0	MMW 14/GE	4.0	MMW 15	4.0
If MATH 20 series or BS then MATH 18		COGS 14A	4.0	COGS 14B (BS)	4.0
COGS 17 ⇄	4.0	GE	4.0	COGS 18 or CSE (8A or 11) or MAE 8	
GE	4.0	ELECTIVE	4.0	GE	4.0

2021-2022 annual units 48.0

Fall (term units 16.0)		Winter (term units 16.0)		Spring (term units 16.0)	
COGS CORE	4.0	COGS CORE (BA) OR COGS 108 (BS)		COGS CORE (BA) OR COGS 100 (BS)	
COGS CORE	4.0	COGS CORE	4.0	COGS CORE	4.0
GE/DEI	4.0	GE	4.0	GE	4.0
ELECTIVE	4.0	ELECTIVE	4.0	UD ELECTIVE	4.0

2022-2023 annual units 40.0

Fall (term units 16.0)		Winter (term units 12.0)		Spring (term units 12.0)	
COGS UD ELECTIVE	4.0	COGS UD ELECTIVE	4.0	COGS UD ELECTIVE	4.0
COGS UD ELECTIVE	4.0	COGS UD ELECTIVE	4.0	COGS UD ELECTIVE	4.0
GE	4.0	ELECTIVE	4.0	ELECTIVE	4.0
ELECTIVE	4.0				



GENERAL CATALOG 2020–21 CATALOG OF RECORD UC San Diego

Home | Courses/Curricula/Faculty | Search | Go

COGS 14B. Introduction to Statistical Analysis (4)
Introduction to descriptive and inferential statistics. Tables, graphs, measures of central tendency and variability. Distributions, Z-scores, correlation, regression. Probability, sampling, logic of inferential statistics, hypothesis testing, decision theory. T-test, one and two-way Anova, nonparametric tests (Chi-square). ***Prerequisites:*** COGS 14A.

COGS 15. What the *#!?: An Uncensored Introduction to Language (4)
This course uses the study of swearing to introduce topics in language: how children learn it, why it changes over time, and how people pronounce and understand it. Students who believe they could be offended by the study of swearing and other taboo language might not find this course appropriate for them.

COGS 17. Neurobiology of Cognition (4)
Introduction to the organization and functions of the nervous system. Topics include molecular, cellular, developmental, systems, and behavioral neurobiology. Specifically, structure and function of neurons, peripheral and central nervous systems, sensory, motor, and control systems, learning and memory mechanisms. (Students may not receive credit for both BILD12 and COGS 17. This course fulfills general-education requirements for Marshall and Roosevelt Colleges as well as Warren by petition.)

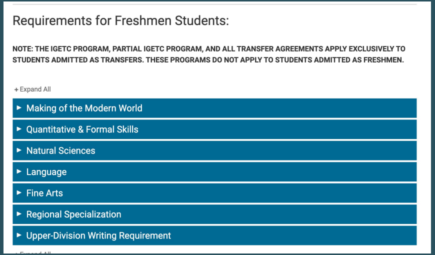
COGS 18. Introduction to Python (4)
This class will teach fundamental Python programming skills and practices, including the "Zen of Python." Students will focus on scientific computing and learn to write functions and tests, as well as how to debug code using the Jupyter Notebook programming environment. Students with limited computing experience may take COGS 3 for preparation.

COGS 20. Exploring the Musical Mind (4)
(Cross-listed with MUS 20.) How do we transform complex sounds into comprehensible and meaningful music? What physiological, neurological, cognitive, and cultural systems are involved? Why do we make music in such diverse ways around the globe? Does music have evolutionary or ecological significance? What is the relationship between music, motion, and emotions? This course explores contemporary understandings of how we hear and how we become musical and invites students to listen to new music in new ways. Students may not receive credit for both COGS 20 and MUS 20.

COGS 87. First-year Student Seminar (1)
The First-year Student Seminar Program is designed to provide new students with the opportunity to explore an intellectual topic with a faculty member in a small seminar setting. First-year Student seminars are offered in all campus departments and undergraduate colleges, and topics vary from quarter to quarter. Enrollment is limited to fifteen to twenty students, with preference given to entering first-year students.

COGS 90. Undergraduate Seminar (1)
Special topics in cognitive science are discussed. P/NP grades only. (May be repeated when topics vary.)

COGS 91. SCANS Presents (1)
The department faculty and the Students for Cognitive and Neurosciences (SCANS) offer this seminar exploring issues in cognitive science. It includes informal faculty research presentations, investigations of topics not covered in the curriculum, and discussions on graduate school and careers. (May be repeated when

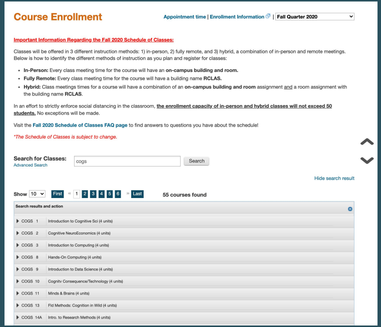


Requirements for Freshmen Students:

NOTE: THE IGETC PROGRAM, PARTIAL IGETC PROGRAM, AND ALL TRANSFER AGREEMENTS APPLY EXCLUSIVELY TO STUDENTS ADMITTED AS TRANSFERS. THESE PROGRAMS DO NOT APPLY TO STUDENTS ADMITTED AS FRESHMEN.

+ Expand All

- ▸ Making of the Modern World
- ▸ Quantitative & Formal Skills
- ▸ Natural Sciences
- ▸ Language
- ▸ Fine Arts
- ▸ Regional Specialization
- ▸ Upper-Division Writing Requirement



Course Enrollment

Appointment time | Enrollment Information | Fall Quarter 2020

Important Information Regarding the Fall 2020 Schedule of Classes:

Classes will be offered in 3 different instruction methods: 1) in-person, 2) fully remote, and 3) hybrid, a combination of in-person and remote meetings. Below is how to identify the different methods of instruction as you plan and register for classes:

- **In-Person:** Every class meeting time for the course will have an **on-campus building and room.**
- **Fully Remote:** Every class meeting time for the course will have a building name **RCLAS.**
- **Hybrid:** Class meetings times for a course will have a combination of an **on-campus building and room** assignment and a room assignment with the building name **RCLAS**.

In an effort to strictly enforce social distancing in the classroom, **the enrollment capacity of in-person and hybrid classes will not exceed 50 students.** No exceptions will be made.

Visit the **Fall 2020 Schedule of Classes FAQ page** to find answers to questions you have about the schedule!

**The Schedule of Classes is subject to change.*

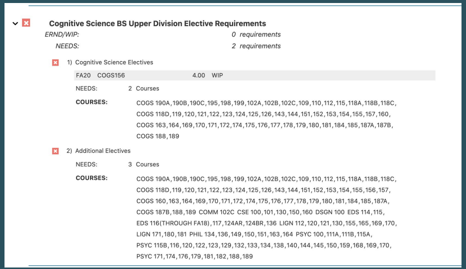
Search for Classes: cogs Search
Advanced Search

Hide search result

Show 10 First « 1 2 3 4 5 6 » Last **55 courses found**

Search results and action

COGS	1	Introduction to Cognitive Sci (4 units)
COGS	2	Cognitive NeuroEconomics (4 units)
COGS	3	Introduction to Computing (4 units)
COGS	8	Hands-On Computing (4 units)
COGS	9	Introduction to Data Science (4 units)
COGS	10	Cognitv Consequence/Technology (4 units)
COGS	11	Minds & Brains (4 units)
COGS	13	Fld Methods: Cognition in Wild (4 units)
COGS	14A	Intro. to Research Methods (4 units)



Cognitive Science BS Upper Division Elective Requirements
ERND/WIP: 0 requirements
NEEDS: 2 requirements

1) Cognitive Science Electives

FA20 COGS156 4.00 WIP

NEEDS: 2 Courses

COURSES: COGS 190A,190B,190C,195,198,199,102A,102B,102C,109,110,112,115,118A,118B,118C, COGS 118D,119,120,121,122,123,124,125,126,143,144,151,152,153,154,155,157,160, COGS 163,164,169,170,171,172,174,175,176,177,178,179,180,181,184,185,187A,187B, COGS 188,189

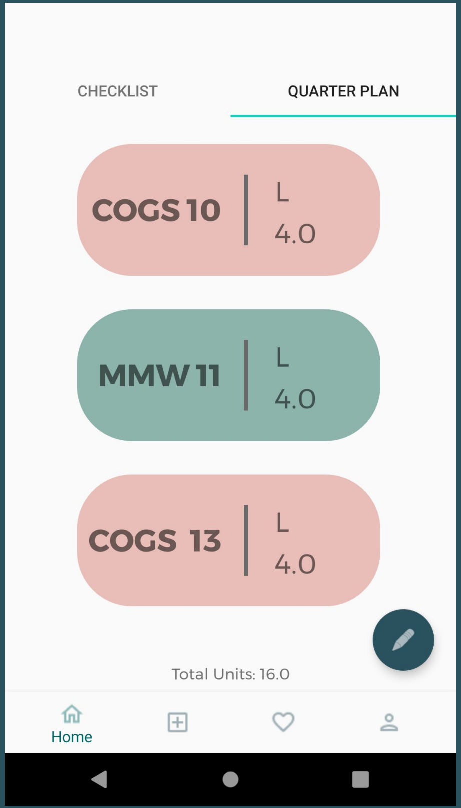
2) Additional Electives

NEEDS: 3 Courses

COURSES: COGS 190A,190B,190C,195,198,199,102A,102B,102C,109,110,112,115,118A,118B,118C, COGS 118D,119,120,121,122,123,124,125,126,143,144,151,152,153,154,155,156,157, COGS 160,163,164,169,170,171,172,174,175,176,177,178,179,180,181,184,185,187A, COGS 187B,188,189 COMM 102C CSE 100,101,130,150,160 DSGN 100 EDS 114,115, EDS 116(THROUGH FA18),117,124AR,124BR,136 LIGN 112,120,121,130,155,165,169,170, LIGN 171,180,181 PHIL 134,136,149,150,151,163,164 PSYC 100,111A,111B,115A, PSYC 115B,116,120,122,123,129,132,133,134,138,140,144,145,150,159,168,169,170, PSYC 171,174,176,179,181,182,188,189

Our solution

- Degree Designer is a mobile app that helps UCSD undergraduates plan and visualize their classes easily!
- It also streamlines the process of planning classes by offering suggestions on future classes and helping students graduate on time.



Project Details: design, workflow, responsibilities

Technologies used

Who did what roles

Everyone can probably talk about what they did, might need to be multiple slides

Our roles

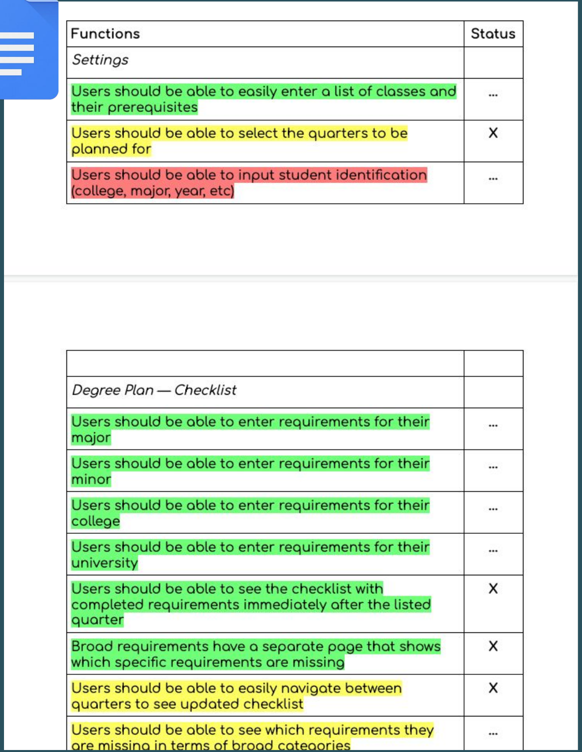
P.S.: everyone pitched in for brainstorming and innovating ideas, needfinding, and app features!

- Aneesha Ramaswamy: Front End Developer
  - Implementing the front end designs for user interfaces.
  - Implementing the storing of custom objects using SharedPreferences and Gson
  - Creating demo video
- Aven Huang: Back End Developer
  - Creating back end custom classes
  - Testing existing classes for improved functionality
- SK Bost: Full Stack Developer
  - Designing, creating, and testing back end classes
  - Developing app concept and use case design
  - Consulting on integration of user interfaces and functions
- Zoe Wong: Designer
  - Designing wireframes and mockups using Figma
  - Designing the logo for the app

Process

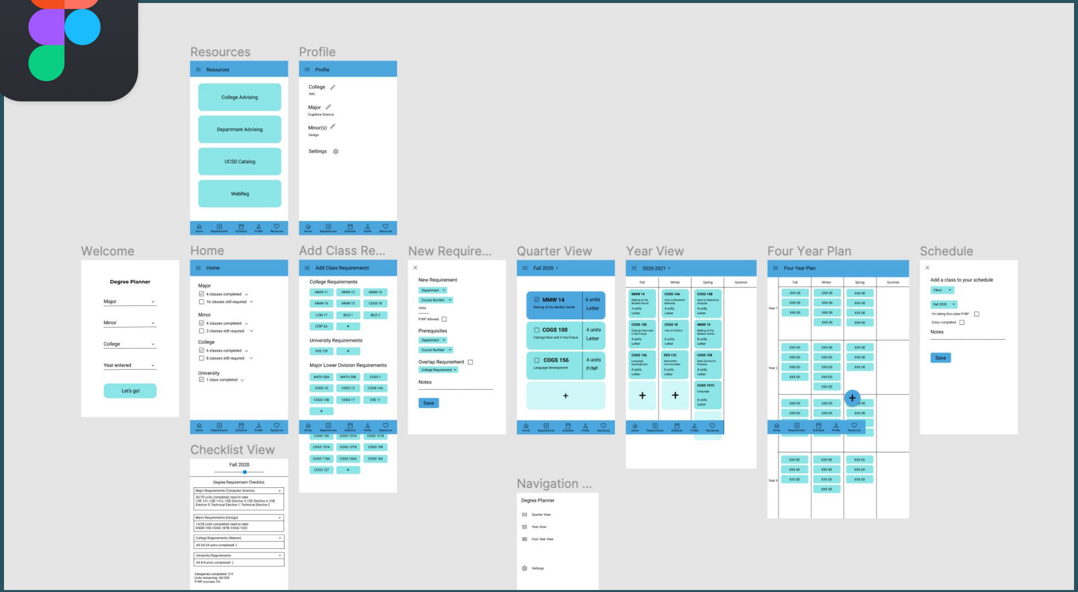
- Brainstorming
- Wireframing
- Start developing!
  - Adding and testing functionality (back end)
  - Implementing designs with code (front end)
- Connecting front and back end
- More tests and finishing touches

Brainstorming and wireframing



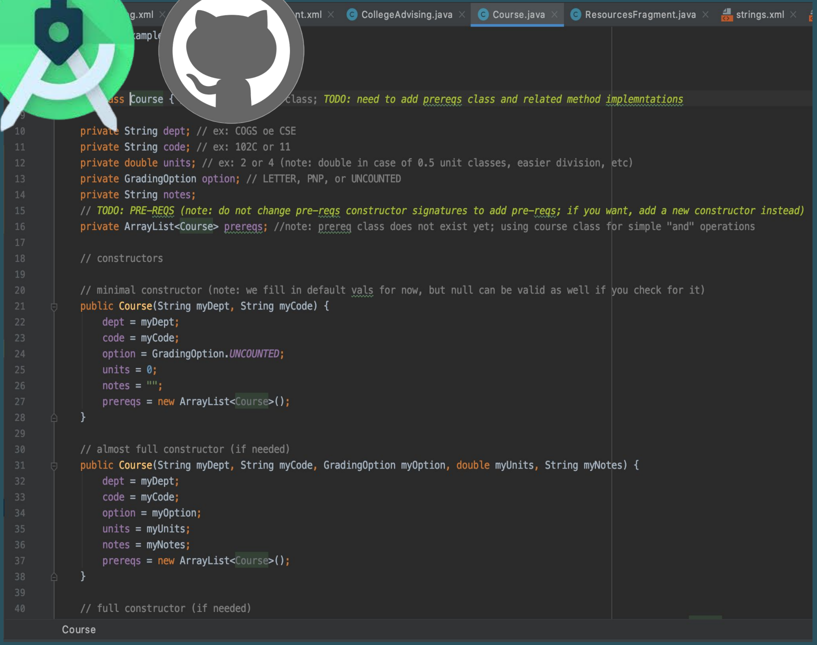
Functions	Status
Settings	
Users should be able to easily enter a list of classes and their prerequisites	...
Users should be able to select the quarters to be planned for	X
Users should be able to input student identification (college, major, year, etc)	...

Degree Plan — Checklist	
Users should be able to enter requirements for their major	...
Users should be able to enter requirements for their minor	...
Users should be able to enter requirements for their college	...
Users should be able to enter requirements for their university	...
Users should be able to see the checklist with completed requirements immediately after the listed quarter	X
Broad requirements have a separate page that shows which specific requirements are missing	X
Users should be able to easily navigate between quarters to see updated checklist	X
Users should be able to see which requirements they are missing in terms of broad categories	...



Developing!

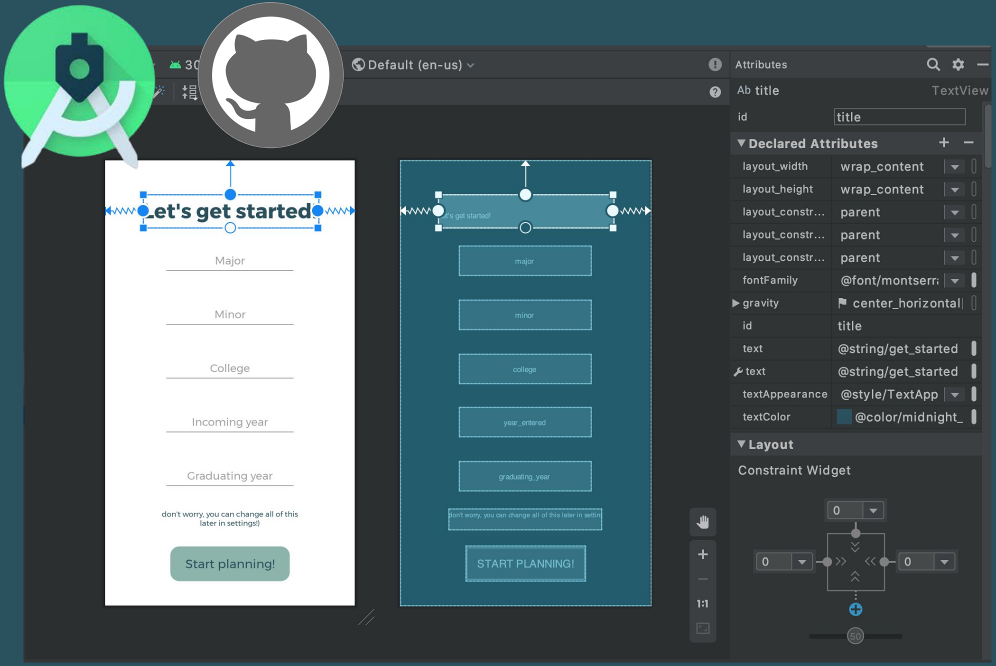
Back end



```
kample

ass Course {                class; TODO: need to add prereqs class and related method implemntations

    private String dept; // ex: COGS oe CSE
    private String code; // ex: 102C or 11
    private double units; // ex: 2 or 4 (note: double in case of 0.5 unit classes, easier division, etc)
    private GradingOption option; // LETTER, PNP, or UNCOUNTED
    private String notes;
    // TODO: PRE-REQS (note: do not change pre-reqs constructor signatures to add pre-reqs; if you want, add a new constructor instead)
    private ArrayList<Course> prereqs; //note: prereg class does not exist yet; using course class for simple "and" operations

    // constructors

    // minimal constructor (note: we fill in default vals for now, but null can be valid as well if you check for it)
    public Course(String myDept, String myCode) {
        dept = myDept;
        code = myCode;
        option = GradingOption.UNCOUNTED;
        units = 0;
        notes = "";
        prereqs = new ArrayList<Course>();
    }

    // almost full constructor (if needed)
    public Course(String myDept, String myCode, GradingOption myOption, double myUnits, String myNotes) {
        dept = myDept;
        code = myCode;
        option = myOption;
        units = myUnits;
        notes = myNotes;
        prereqs = new ArrayList<Course>();
    }

    // full constructor (if needed)
```

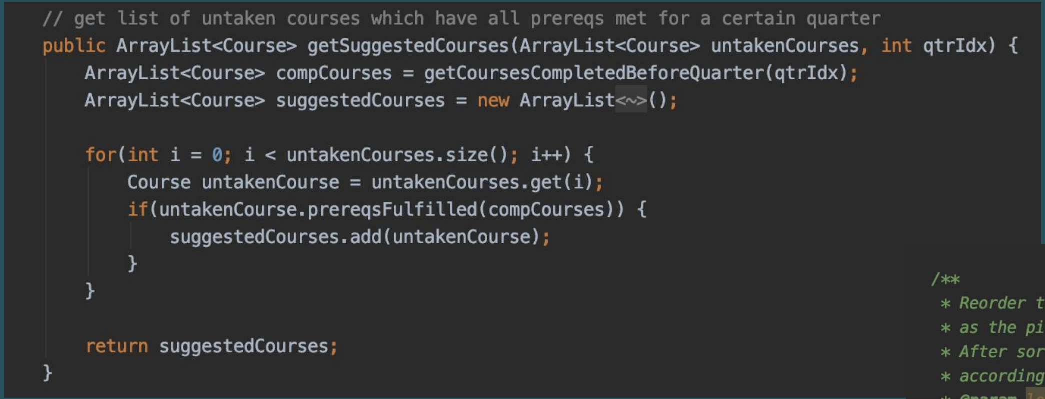
Course

Front end



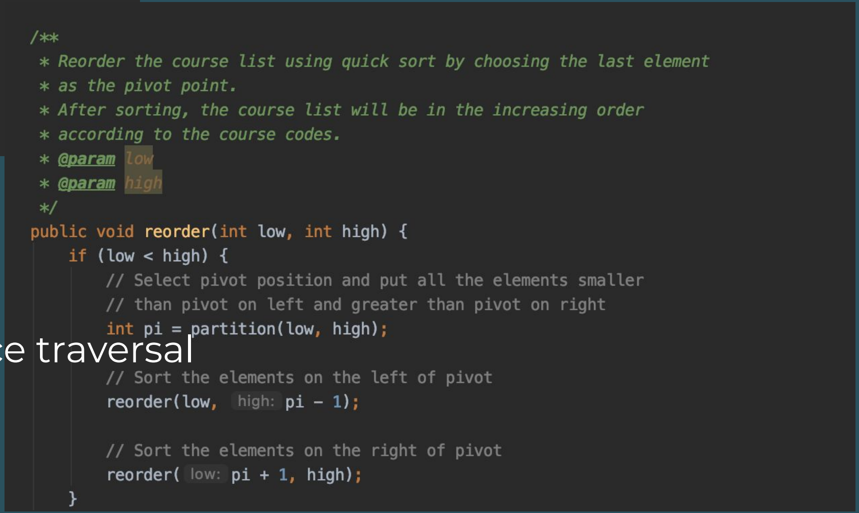
Attributes	
Ab title	TextView
id	title
Declared Attributes	
layout_width	wrap_content
layout_height	wrap_content
layout_constr...	parent
layout_constr...	parent
layout_constr...	parent
fontFamily	@font/montserr
gravity	center_horizontal|
id	title
text	@string/get_started
text	@string/get_started
textAppearance	@style/TextApp
textColor	@color/midnight_
Layout	
Constraint Widget	

Adding functionalities in the back end



```java
// get list of untaken courses which have all prereqs met for a certain quarter
public ArrayList<Course> getSuggestedCourses(ArrayList<Course> untakenCourses, int qtrIdx) {
    ArrayList<Course> compCourses = getCoursesCompletedBeforeQuarter(qtrIdx);
    ArrayList<Course> suggestedCourses = new ArrayList<~>();

    for(int i = 0; i < untakenCourses.size(); i++) {
        Course untakenCourse = untakenCourses.get(i);
        if(untakenCourse.prereqsFulfilled(compCourses)) {
            suggestedCourses.add(untakenCourse);
        }
    }

    return suggestedCourses;
}
```

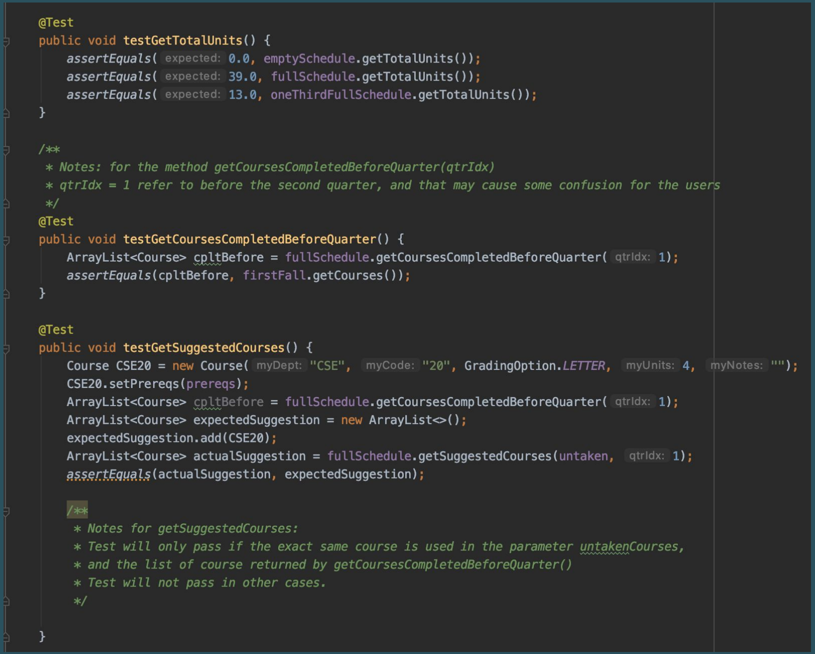
Suggesting what courses to take next



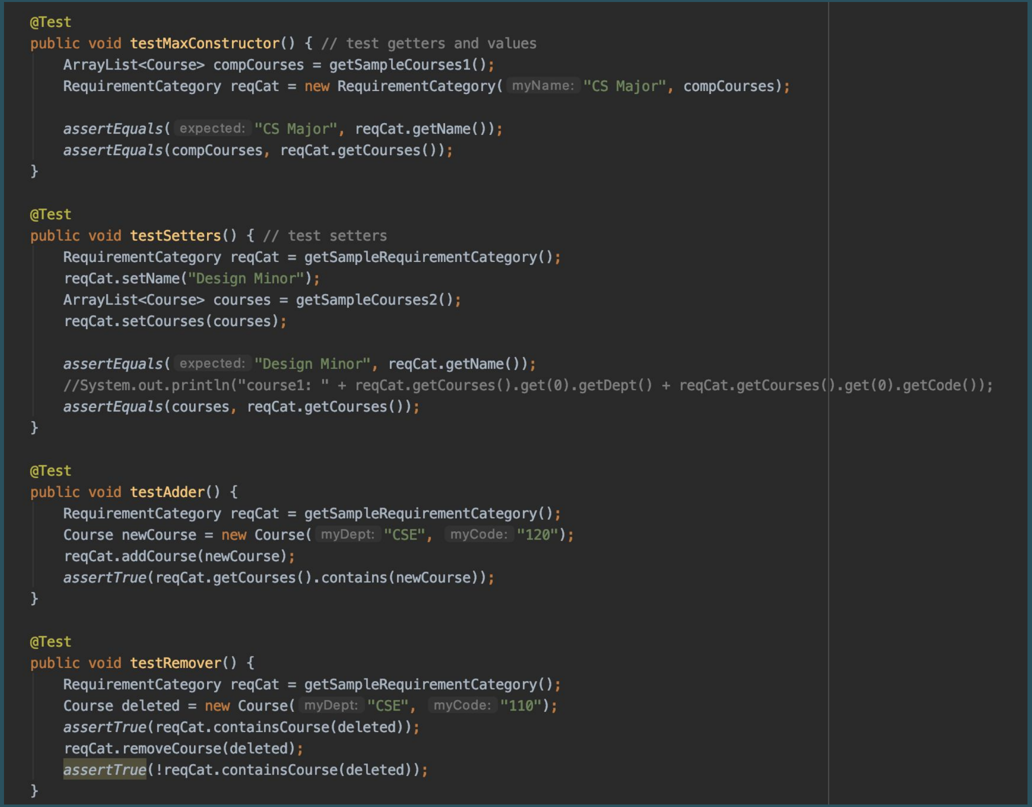
```java
/**
 * Reorder the course list using quick sort by choosing the last element
 * as the pivot point.
 * After sorting, the course list will be in the increasing order
 * according to the course codes.
 * @param low
 * @param high
 */
public void reorder(int low, int high) {
    if (low < high) {
        // Select pivot position and put all the elements smaller
        // than pivot on left and greater than pivot on right
        int pi = partition(low, high);

        // Sort the elements on the left of pivot
        reorder(low, high: pi - 1);

        // Sort the elements on the right of pivot
        reorder( low: pi + 1, high);
    }
```

Reordering the course list for convenience traversal

Testing back end functions

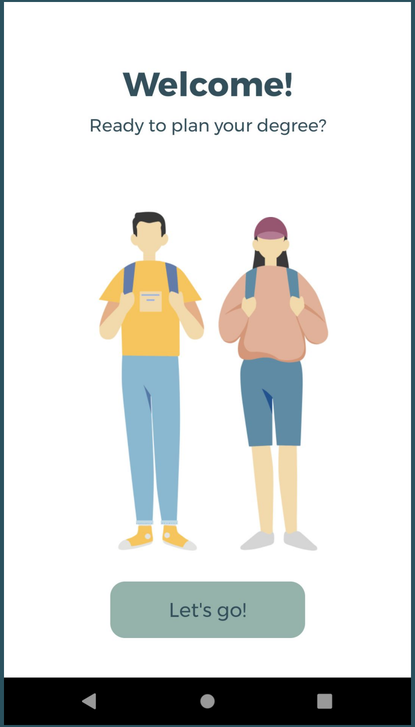


```java
@Test
public void testGetTotalUnits() {
    assertEquals( expected: 0.0, emptySchedule.getTotalUnits());
    assertEquals( expected: 39.0, fullSchedule.getTotalUnits());
    assertEquals( expected: 13.0, oneThirdFullSchedule.getTotalUnits());
}

/**
 * Notes: for the method getCoursesCompletedBeforeQuarter(qtrIdx)
 * qtrIdx = 1 refer to before the second quarter, and that may cause some confusion for the users
 */
@Test
public void testGetCoursesCompletedBeforeQuarter() {
    ArrayList<Course> cpltBefore = fullSchedule.getCoursesCompletedBeforeQuarter( qtrIdx: 1);
    assertEquals(cpltBefore, firstFall.getCourses());
}

@Test
public void testGetSuggestedCourses() {
    Course CSE20 = new Course( myDept: "CSE",  myCode: "20", GradingOption.LETTER,  myUnits: 4,  myNotes: "");
    CSE20.setPrereqs(prereqs);
    ArrayList<Course> cpltBefore = fullSchedule.getCoursesCompletedBeforeQuarter( qtrIdx: 1);
    ArrayList<Course> expectedSuggestion = new ArrayList<>();
    expectedSuggestion.add(CSE20);
    ArrayList<Course> actualSuggestion = fullSchedule.getSuggestedCourses(untaken,  qtrIdx: 1);
    assertEquals(actualSuggestion, expectedSuggestion);

    /**
     * Notes for getSuggestedCourses:
     * Test will only pass if the exact same course is used in the parameter untakenCourses,
     * and the list of course returned by getCoursesCompletedBeforeQuarter()
     * Test will not pass in other cases.
     */

}
```



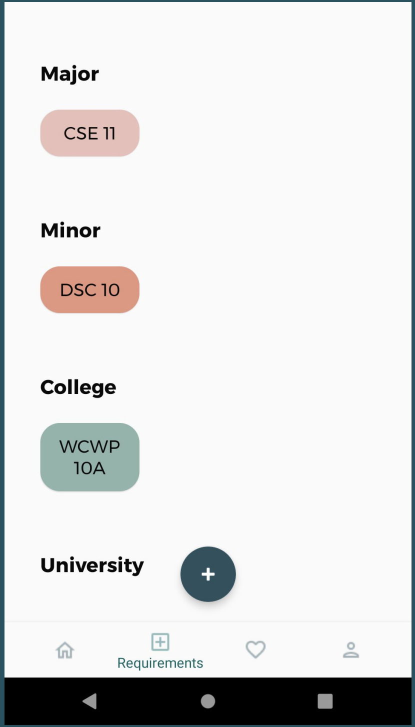
```java
@Test
public void testMaxConstructor() { // test getters and values
    ArrayList<Course> compCourses = getSampleCourses1();
    RequirementCategory reqCat = new RequirementCategory( myName: "CS Major", compCourses);

    assertEquals( expected: "CS Major", reqCat.getName());
    assertEquals(compCourses, reqCat.getCourses());
}

@Test
public void testSetters() { // test setters
    RequirementCategory reqCat = getSampleRequirementCategory();
    reqCat.setName("Design Minor");
    ArrayList<Course> courses = getSampleCourses2();
    reqCat.setCourses(courses);

    assertEquals( expected: "Design Minor", reqCat.getName());
    //System.out.println("course1: " + reqCat.getCourses().get(0).getDept() + reqCat.getCourses().get(0).getCode());
    assertEquals(courses, reqCat.getCourses());
}

@Test
public void testAdder() {
    RequirementCategory reqCat = getSampleRequirementCategory();
    Course newCourse = new Course( myDept: "CSE",  myCode: "120");
    reqCat.addCourse(newCourse);
    assertTrue(reqCat.getCourses().contains(newCourse));
}

@Test
public void testRemover() {
    RequirementCategory reqCat = getSampleRequirementCategory();
    Course deleted = new Course( myDept: "CSE",  myCode: "110");
    assertTrue(reqCat.containsCourse(deleted));
    reqCat.removeCourse(deleted);
    assertTrue(!reqCat.containsCourse(deleted));
}
```

Creating high-fidelity mockups


Welcome!

Ready to plan your degree?

Let's go!


Major

CSE 11

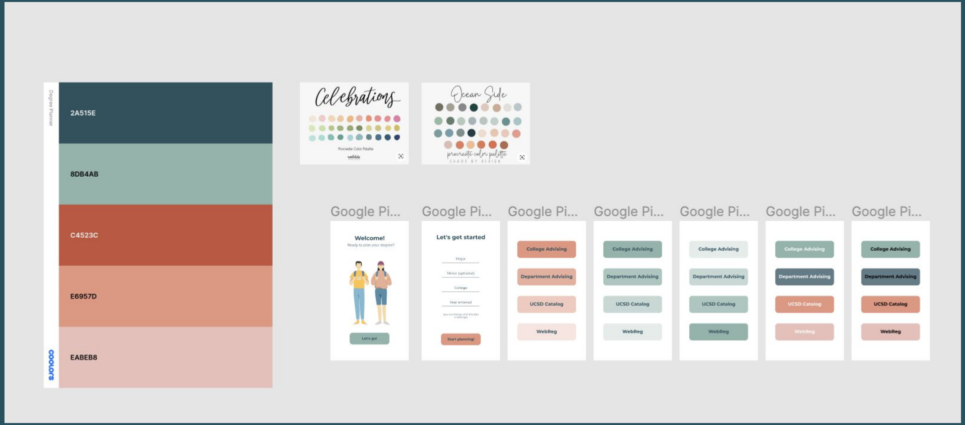
Minor

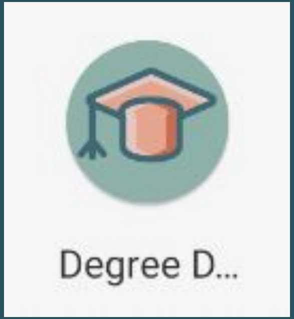
DSC 10

College

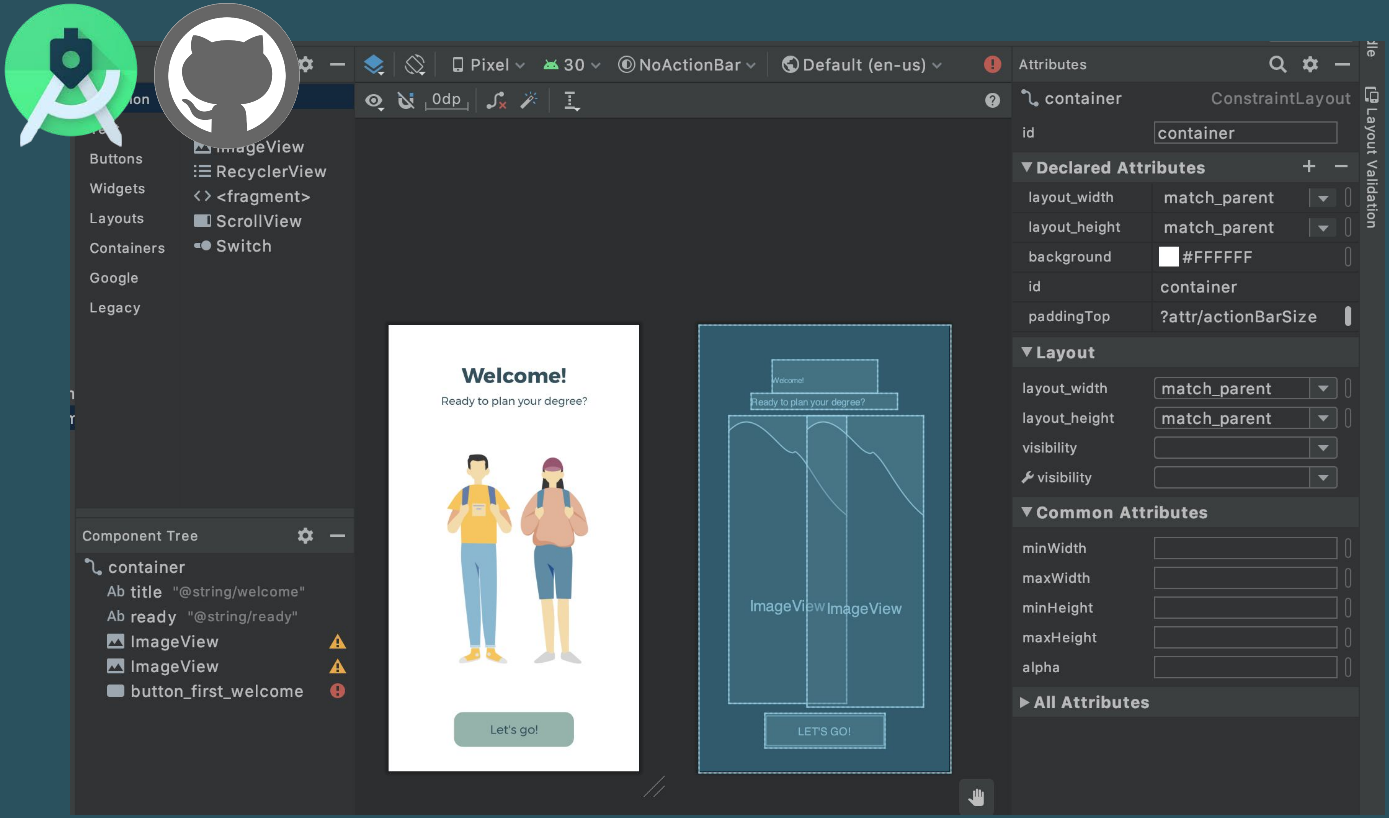
WCWP 10A

University





Implementing UI with code



Component Tree

- container
  - Ab title "@string/welcome"
  - Ab ready "@string/ready"
  - ImageView
  - ImageView
  - button_first_welcome

Attributes

container ConstraintLayout

id: container

Declared Attributes

layout_width	match_parent
layout_height	match_parent
background	#FFFFFF
id	container
paddingTop	?attr/actionBarSize

Layout

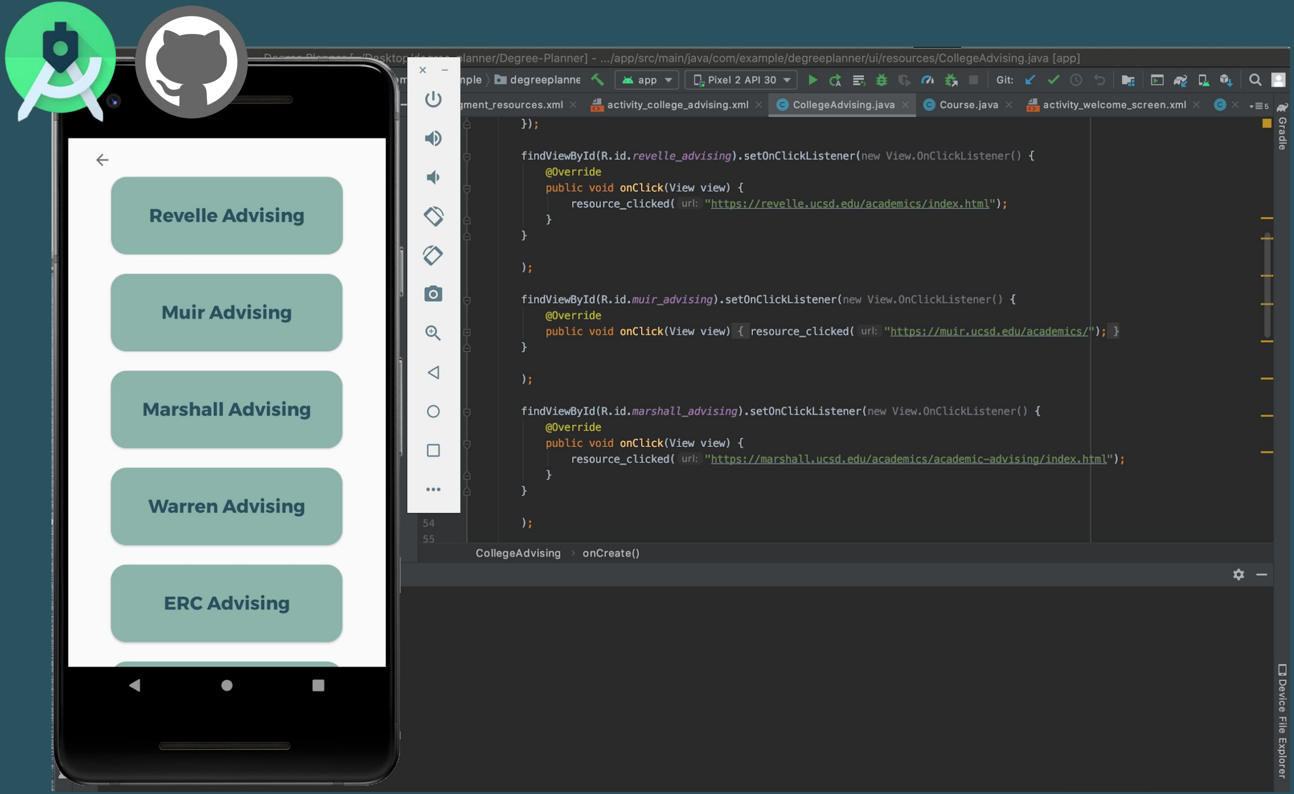
layout_width	match_parent
layout_height	match_parent
visibility	
visibility	

Common Attributes

minWidth	
maxWidth	
minHeight	
maxHeight	
alpha	

All Attributes

Connecting back end and front end

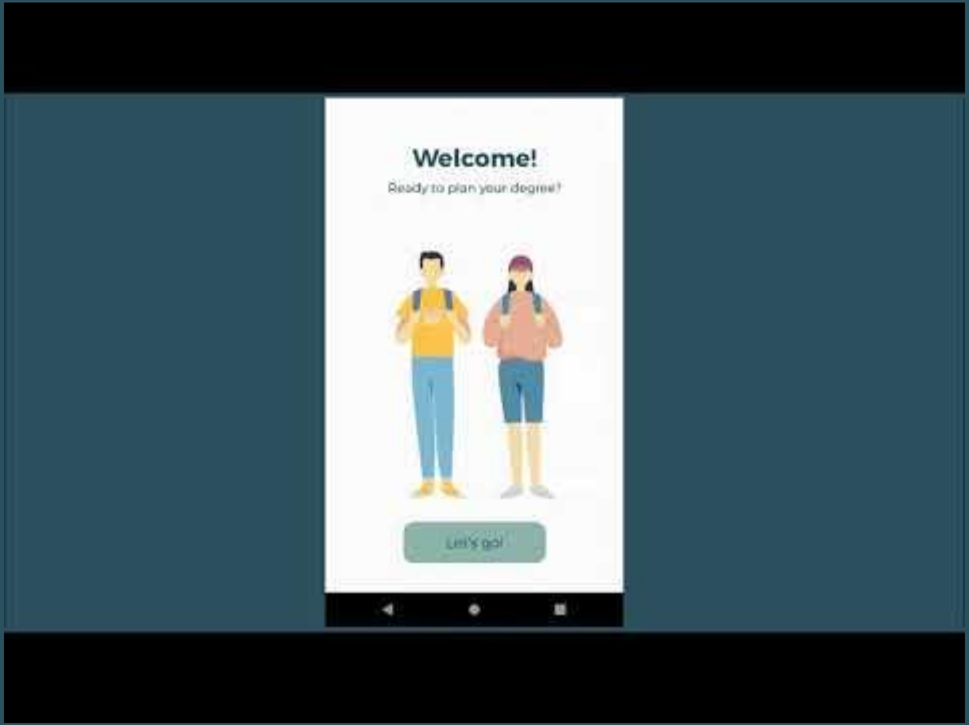

Revelle Advising
Muir Advising
Marshall Advising
Warren Advising
ERC Advising
…lanner/Degree-Planner] - .../app/src/main/java/com/example/degreeplanner/ui/resources/CollegeAdvising.java [app]
degreeplanne
app
Pixel 2 API 30
Git:
…gment_resources.xml
activity_college_advising.xml
CollegeAdvising.java
Course.java
activity_welcome_screen.xml
Gradle
CollegeAdvising › onCreate()
Device File Explorer

```java
        });

        findViewById(R.id.revelle_advising).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                resource_clicked( url: "https://revelle.ucsd.edu/academics/index.html");
            }
        }

        );

        findViewById(R.id.muir_advising).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { resource_clicked( url: "https://muir.ucsd.edu/academics/"); }
        }

        );

        findViewById(R.id.marshall_advising).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                resource_clicked( url: "https://marshall.ucsd.edu/academics/academic-advising/index.html");
            }
        }

        );
```

User flow

- Enter major, minor, colleges, and incoming year
- Enter course requirements
    - Under each category of course requirement, user needs to enter Department, Course Number, Course Units, P/NP, and its prerequisites. User could also add any overlapping requirements for the courses.
- After adding courses, users can view them in Quarter view, Year view, or Four Year Plan
- Users will be able to see their major and minor requirements, university requirements, and college requirements in the Checklist view. Users can also get a list of suggested courses to take.

Demo


Welcome!

Ready to plan your degree?

Lessons learned

- Time management is super important!
  - Balancing schoolwork with the project (Summer Session II)
- Need for planning and communication between front and back end
  - Communicating what’s finished and how to use things across different roles, paces, and skill levels
- The connection between a function and its implementation isn't always intuitive
  - Storing app information
  - Planning degree quarter by quarter and view unmet prerequisites

What's next?

- What we would do differently if we had more time
  - Add more detailed resources for each college and department
  - Add preloaded setups for majors/minors, classes, and prerequisites
- What still needs to be improved
  - Improve profile section so user input ties more directly into app functions
  - Include the ability to edit class information after it has been planned
  - Make it clearer what quarter is currently being viewed on the home page

Thanks for listening!