
Design Document for CyPlan

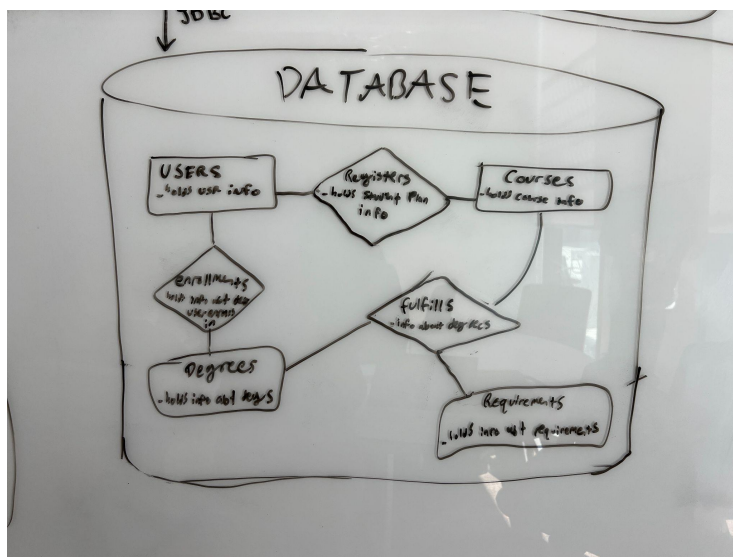
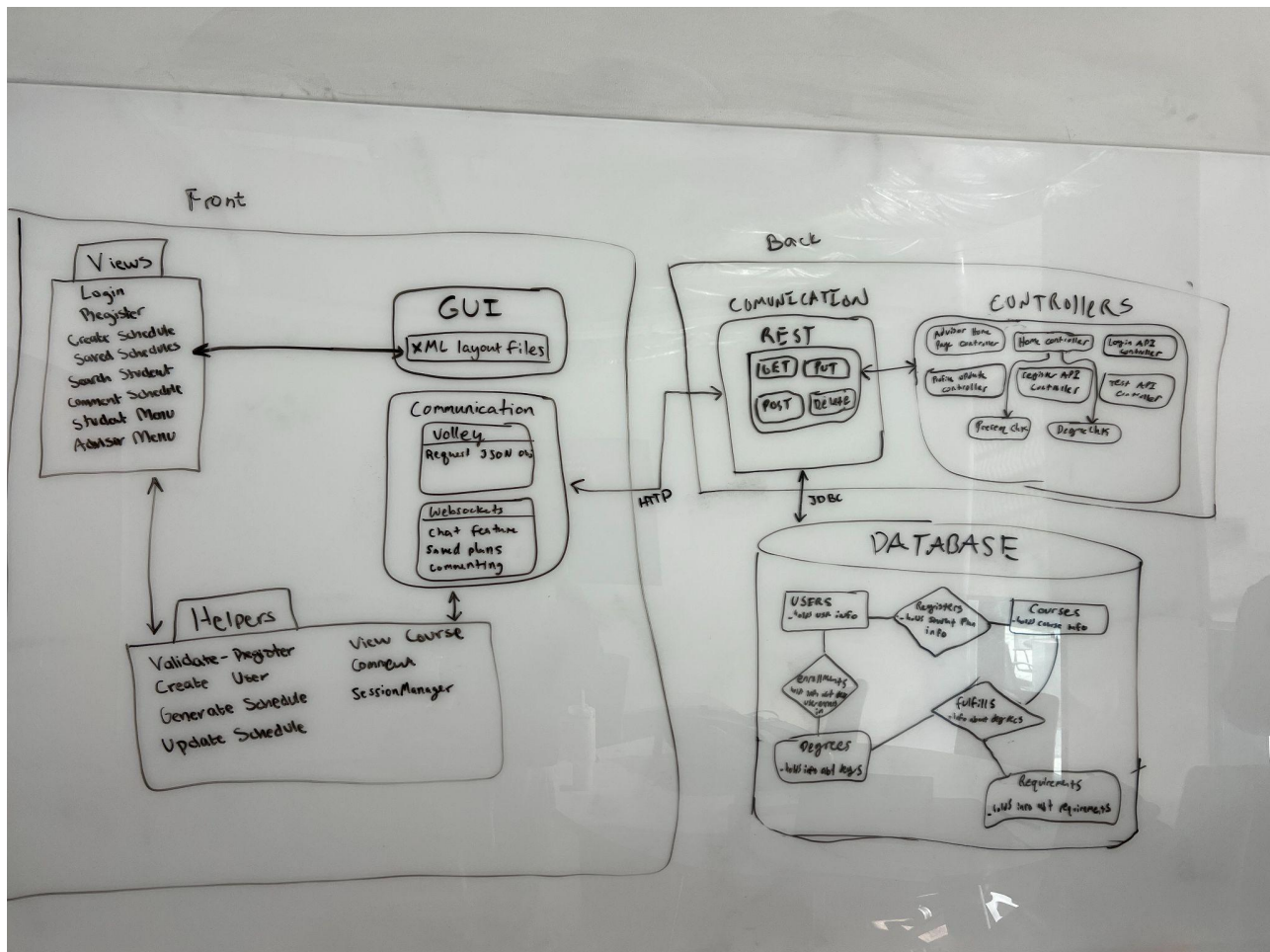
Group KM_207

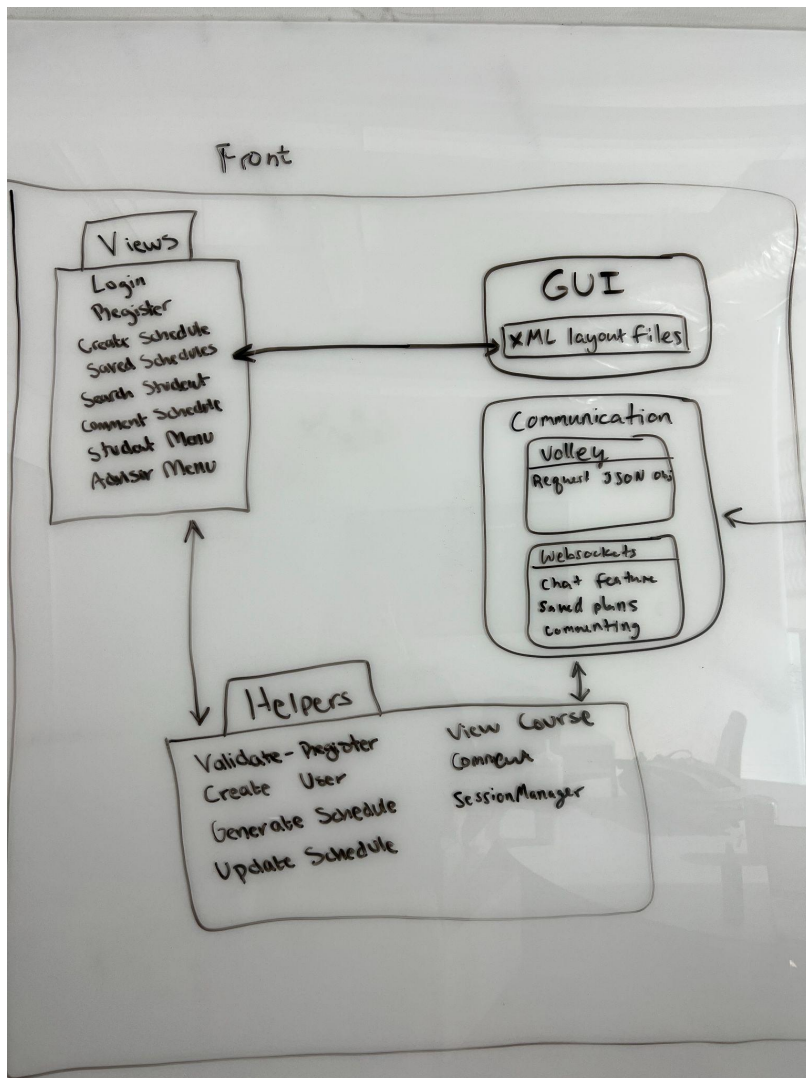
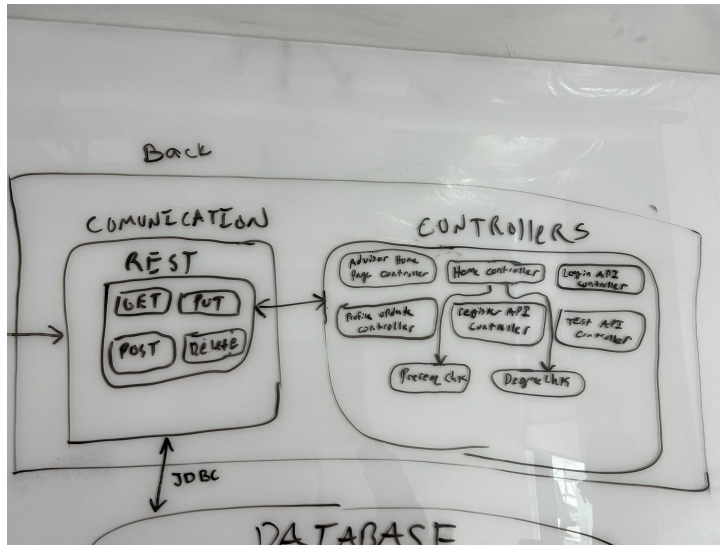
Member1: Paige Rolling: 25 % Front end

Member2: Kayley Clark: 25% Back end

Member3: Logan Ellsworth: 25% Front end

Member4: Tanner Smith: 25% Back end





Frontend

CreateAccount

- Create account generates a page with the following elements:
 - EditText: email
 - EditText: name
 - EditText: Password
 - Spinner: Major
 - Checkbox: isAdvisor
 - Button: CreateAccount
 - Button: goToLogin
- Upon clicking the button 'CreateAccount' the values of the UserID, Password, and Email are sent as a POST request to the server.

Login

- Create account generates a page with the following elements:
 - EditText: isuEmail
 - EditText: Password
 - Button: Login
 - Button: SignUp
- Login Screen takes user input of their username and password, and if that username and password is valid, it will take the user to the main menu, where they are logged in as that user. If they are an advisor, it will take them to the advisor view and if they are a student, it will take them to the student view.

Profile (Student)

- Profile generates a page with the following elements:
 - Button: Update Profile/Save Changes
 - EditText: Name
 - Spinner: Major
- Clicking the Update Profile button will take the user into 'edit mode' in which they are able to make changes to their information. The same button will change to say 'Save Changes', and on click that sends a PUT request to to update the user information.

Create Plan (Student)

- Create Plan generates a page with the following elements:
 - Scroll View: Schedule
- A GET request to the db generates the base schedule for a student based on their chosen major. On click and drag of a class, a PUT request is sent to the server to move what semester they are taking that course. On click of a course, a pop up of information of that course will appear and the student can see information about that course and comments that other students have made about that course. They can also click 'Remove Course' which will remove it from their schedule if they have transfer credit for that course.

Saved Plans (Student)

- Saved Plans generates a page with the following elements:
 - Plans: buttons to each of their saved plans
 - Button: Delete Plans
- The Plans buttons are generated from the student's saved plans. On click of one of the plans, the student is able to go to that plan in the Create Plan mode. On click of the delete plans button, the student has the ability to mark courses to delete.

Search Student (Advisor)

- Search Student generates a page with the following elements:
 - Search: searchStudent
- When a student is searched, if their isu email exists in the database, their saved plans will be generated on the screen. On click, the advisor can go to those plans.

Advise Plan (Advisor)

- Advise Plan generates a page with the following elements:
 - Scroll View: Schedule
- On click of a schedule in Student Search, the schedule will be generated. On click, the advisor can make comments on when students should take certain courses and if they need courses that they don't have.

Backend

Communication

The backend uses mappings to update the database based on information sent to the given mappings' URLs. These include:

- Post: send information on an item to be added to the database.
- Get: request information, often with an identifier for the specific item requested from the database
- Put: send information to update a specific item in the database
- Delete: send an identifier to delete a specific item from the database

Controllers

The controllers contain the mappings for communication between frontend and the database. These include:

- AdvisorHomeController: This controller puts and receives advisor information into the database.
- HomeApiController: This controller deals with most things related to student plans including getting and posting information into the plan database. It also deals with checking whether a plan's prerequisites are in order and all degree requirements are met.
- LoginApiController: This controller deals with the login page. It gets information from the database and compares it to the information that the user enters to see if they are a valid user.
- ProfileUpdateController: This controller deals with updates to user profiles including name changes or other personal information changes.
- RegisterApiController: This controller creates a user and their information. It also checks that the user does not already exist
- TestApiController: This controller is one we use when we are having problems connecting to the application. It is a super simple controller to test if our connection is correct.

Database Diagram:

