

Plan: FIT Schedule Planner

Semester 2



Pedro Moura, Jordan Synodis

Project Goal and Motivation



- **Minimize confusion and misunderstandings in the class registration process.**
- **Help students avoid unnecessary classes and scheduling difficulties.**
- **Enhance efficiency by offering a comprehensive tool for academic planning.**

Approach



- **Develop a user-friendly interface that streamlines the class registration process for students at FIT.**
- **Create a tool that aims to combine access to class schedules, degree evaluations, and program requirements in one convenient location.**
- **Implement features such as class search filters, schedule visualization, and progress checklists are key components of the development approach.**
- **The user will be able to add classes to their schedule, check prerequisites, identify overlapping classes, and utilize filters for a tailored search experience.**
- **Integrate personal time blocks to prevent class schedule conflicts with extracurricular activities.**

Algorithms and Tools



- **MongoDB**
- **Express.js (Backend)**
- **React (Frontend)**
- **Node.js (Server-side JavaScript execution)**
- **Axios (HTTP requests)**
- **RateMyProfessor API**
- **FIT's Online Resources:**
 - Fall Schedule
 - Spring Schedule
 - Degree Programs

Novel Features/Functionalities

01

Streamlined Web Interface

- Combines class schedules, degree evaluations, and program requirements
- Enables seamless class registration

02

Class Search Filters

- Find classes by subject, professor, times, course number, credit range, or RateMyProfessor score

03

Schedule Visualization

- View personalized weekly schedules as classes are added or removed

04

Progress Checklist

- Generated from CAPP Degree Evaluation
- Tracks program completion progress

05

User-Friendly Interface

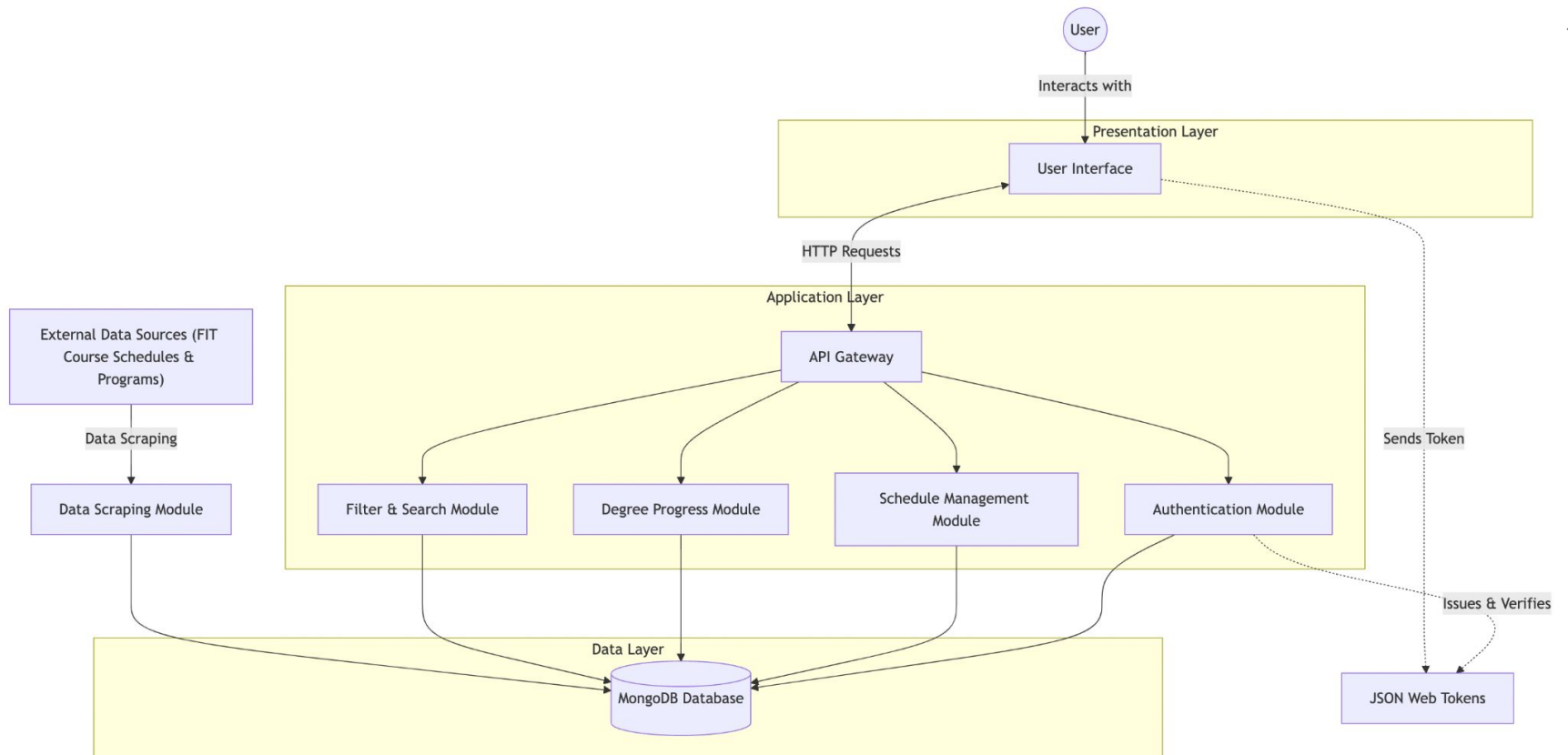
- Consistent layout, navigation bar, and FAQs tab
- Pop-ups and loading bars for real-time feedback and status updates

Technical Challenges



- **Getting Real-Time Data**
 - We can't access data directly so we have to scrape data
- **Web Application Development**
 - We only have experience with building simple GUIs
- **Dynamic Data Handling**
 - Efficient querying and data processing is required for real-time features

Design



What We're Looking For



- **Usability:** Measured ease of navigation, and clarity of features, among others, through user surveys.
- **Reliability:** Make sure that the system functions correctly 90% of the time when testing.
- **Accuracy:** Compare/Match/Validate course filtering, prerequisite checks, and schedule conflict detection.
- **Performance:** Measure response times for class searches and checklist generation. We are looking for an average of 2 and 5 seconds, respectively.

What We Have So Far



| Feature | Completion % | To-dos |
|---------------------------------|--------------|--|
| Degree Checklist | 60% | Fix client-side deployment bugs, incorporate CAPP data |
| Schedule Conflict Detection | 70% | Debug and enhance time conflict resolution |
| Advanced Filtering | 90% | Finalize RateMyProfessor filter integration |
| Prerequisite Tree Visualization | 90% | Improve course selection flow |
| Weekly Schedule Grid | 100% | Completed |

Milestone 4



| Task | Pedro | Jordan |
|---|-------|--------|
| 1. Implement, test, and demo loading the CAPP Degree Evaluation | 50% | 50% |
| 2. Implement, test, and demo accessing the program checklist | 50% | 50% |
| 3. Implement test, and demo additional filtering option, address bugs with formatting | 0% | 100% |
| 4. Fix user context | 100% | 0% |

Milestone 5



- **Implement, test, and demo prerequisite checking system**
- **Implement, test, and demo schedule updates**
- **Conduct evaluation and analyze results**

Milestone 6



- **Implement, test, and demo which features/modules**
- **Test/demo of the entire system**
- **Conduct evaluation and analyze results**
- **Create user/developer manual**
- **Create demo video**