

Delgado-UNO Course Transfer Application

Technical Documentation & User Guide

Development Team

June 19, 2025

Contents

1	Executive Summary	4
2	The Problem We're Solving	4
3	System Architecture Overview	5
3.1	Why These Technologies?	5
3.2	The Big Picture	5
4	Database Design Philosophy	5
4.1	Core Database Tables	6
4.2	Performance Considerations	6
5	User Interface Design	6
5.1	Search-First Design	6
5.2	Clear Information Hierarchy	7
5.3	Responsive Design	7
6	Search Functionality	7
6.1	Multi-Field Search	7
6.2	Intelligent Filtering	8
6.3	Real-Time Results	8
7	Transfer Planning Features	8
7.1	Personalized Transfer Plans	8
7.2	Progress Tracking	8
7.3	Degree Audit Integration	9
8	Administrative Features	9
8.1	Equivalency Management	9
8.2	User Management	9
8.3	Data Analytics	9
9	Security and Privacy	9
9.1	Authentication and Authorization	10
9.2	Data Protection	10
9.3	Privacy Compliance	10
10	Performance and Scalability	10
10.1	Database Optimization	10
10.2	Frontend Performance	10
10.3	Scalability Architecture	11
11	Deployment and Infrastructure	11
11.1	Server Requirements	11
11.2	High Availability Setup	11
11.3	Security Hardening	12

12 Development Workflow	12
12.1 Technology Stack	12
12.2 Development Environment	12
12.3 Code Organization	12
13 Testing Strategy	13
13.1 Frontend Testing	13
13.2 Backend Testing	13
13.3 Integration Testing	13
14 Monitoring and Maintenance	14
14.1 Performance Monitoring	14
14.2 Usage Analytics	14
14.3 Maintenance Procedures	14
15 Future Enhancements	14
15.1 Advanced Search Features	15
15.2 Mobile Application	15
15.3 Institutional Expansion	15
15.4 Advanced Analytics	15
16 Conclusion	16

Executive Summary

The Delgado-UNO Course Transfer Application represents a significant step forward in making college transfers smoother and less stressful for students. If you've ever tried to figure out which of your community college courses will count toward your bachelor's degree, you know how frustrating and confusing the process can be. This application solves that problem by providing a comprehensive, user-friendly platform that helps students navigate their transfer journey with confidence.

At its core, this isn't just another academic website—it's a tool designed by people who understand the real challenges students face when transferring between institutions. The application connects Delgado Community College students with University of New Orleans, providing clear answers to questions like "Will my biology course transfer?" and "How many credits do I need to graduate?"

The system handles the complex task of course equivalency mapping, which is essentially a sophisticated matching system that determines how courses from one institution align with requirements at another. But instead of buried in confusing catalogs and outdated PDFs, this information is presented in an intuitive, searchable format that actually makes sense to students.

What makes this application particularly valuable is its focus on the student experience. Rather than forcing students to decode academic jargon or navigate multiple systems, everything they need is consolidated into one place. Students can search for courses, see exactly how their credits will transfer, and even plan out their entire academic path before making the transfer decision.

The Problem We're Solving

College transfer processes are notoriously complicated, and for good reason. Educational institutions have different course structures, numbering systems, and degree requirements. When a student wants to transfer from Delgado Community College to UNO, they're essentially asking two institutions to agree on what knowledge and skills are equivalent between their respective programs.

Traditionally, this process involves:

- Manually comparing course descriptions from different catalogs
- Submitting transcripts and waiting for evaluation
- Meeting with multiple advisors across different institutions
- Dealing with incomplete or outdated equivalency information
- Making transfer decisions without complete information about credit acceptance

The result is often frustrated students who discover after transferring that some of their hard-earned credits don't count toward their degree, extending their time to graduation and increasing costs.

Our application addresses these pain points by providing transparent, up-to-date information about course equivalencies and transfer requirements. Students can make informed decisions about their academic path before committing to a transfer, reducing surprises and improving their overall educational experience.

System Architecture Overview

The application is built using modern web technologies that prioritize both user experience and system reliability. We've chosen a traditional but robust approach that many successful web applications use: a React frontend communicating with a Node.js backend, supported by a MySQL database.

Why These Technologies?

React for the Frontend: React has become the gold standard for building interactive user interfaces, and for good reason. It allows us to create a responsive, dynamic experience where students can search for courses, filter results, and manage their transfer plans without constant page refreshes. The component-based architecture means we can build reusable interface elements that work consistently across the entire application.

Node.js and Express for the Backend: Node.js excels at handling the kind of data-intensive operations this application requires—searching through thousands of courses, managing user sessions, and processing complex equivalency relationships. Express provides a clean, minimal framework for building our API endpoints without unnecessary complexity.

MySQL for Data Storage: Course equivalencies are inherently relational data. When we say "Delgado's BIOL 1001 equals UNO's BIOL 1010," we're describing a relationship between two courses, two institutions, and potentially additional metadata like prerequisites or transfer conditions. MySQL handles these complex relationships naturally and provides the reliability that academic data requires.

The Big Picture

From a student's perspective, the application flow is straightforward:

1. Search for courses using familiar terms (course codes, titles, or descriptions)
2. View detailed information about how courses transfer
3. Create and manage a personalized transfer plan
4. Track progress toward degree completion

Behind the scenes, the system is handling complex queries across multiple database tables, managing user authentication, and providing real-time search results. But the complexity is hidden from users, who see only a clean, intuitive interface.

Database Design Philosophy

The database design reflects how academic institutions actually organize their information. We have a hierarchical structure that mirrors real-world academic organization:

Institutions → Departments → Courses → Equivalencies

This structure makes sense because it matches how people think about courses. A course isn't just "Biology 101"—it's "Biology 101 as taught by the Biology Department at Delgado Community College." This specificity is crucial for accurate transfer credit evaluation.

Core Database Tables

Institutions Table: Stores basic information about Delgado and UNO, including their official names, codes, and contact information.

Departments Table: Contains all academic departments (Biology, Mathematics, English, etc.) with their official codes and names. This helps organize courses logically and enables department-level filtering.

Courses Table: The heart of the system, containing detailed information about every course offered by both institutions. This includes course codes, titles, descriptions, credit hours, and prerequisites.

Course Equivalencies Table: This is where the magic happens—the table that defines which Delgado courses are equivalent to which UNO courses. Each equivalency record includes transfer type (direct equivalent, elective credit, etc.) and any special conditions or notes.

Users and Transfer Plans: These tables support the personalized planning features, allowing students to create accounts, save their transfer plans, and track their progress over time.

Performance Considerations

Academic databases can be large—UNO alone offers thousands of courses across dozens of departments. To ensure the search functionality remains fast and responsive, we've implemented strategic database indexing on the fields students search most frequently: course codes, titles, and departments.

We also use connection pooling, which is a technical optimization that dramatically improves database performance. Instead of creating a new database connection for each search (which is slow), the application maintains a pool of ready-to-use connections, reducing response times from several hundred milliseconds to just a few milliseconds.

User Interface Design

The user interface is designed around the principle that academic information should be accessible, not intimidating. We've prioritized clarity and usability over flashy graphics or complex navigation structures.

Search-First Design

The primary interface element is a search box that works the way students expect it to. Students can search for courses using:

- Course codes (e.g., "MATH 1550")
- Course titles (e.g., "Calculus I")
- Partial information (e.g., "calc" to find calculus courses)
- Department names (e.g., "biology" to see all biology courses)

The search includes intelligent features like fuzzy matching, which means if a student types "MATH 15" looking for "MATH 1550," the system will still find the right course. This reduces frustration and makes the search process more forgiving of typos or incomplete information.

Clear Information Hierarchy

When students view course information, they see the most important details first:

1. Course title and code
2. Credit hours and transfer status
3. UNO equivalent (if applicable)
4. Prerequisites and additional notes

This hierarchy matches how students actually think about courses—they want to know what the course is, whether it transfers, and what they need to take it.

Responsive Design

Many students access academic information on their phones, especially when they're meeting with advisors or comparing options between classes. The application works seamlessly across all device sizes, from large desktop monitors to small smartphone screens.

Search Functionality

The search system is one of the most technically complex parts of the application, but it's designed to feel simple and intuitive to users. Here's how it works:

Multi-Field Search

When a student enters a search term, the system simultaneously searches across multiple fields:

- Course codes (BIOL, MATH, ENGL, etc.)
- Course numbers (1001, 1550, 2000, etc.)
- Course titles (Calculus, Biology, Composition, etc.)
- Course descriptions (full text search)

This comprehensive approach means students can find courses using whatever information they remember or have available.

Intelligent Filtering

Beyond basic text search, students can filter results by:

- Department (to see only courses in their major)
- Credit hours (to find courses that fit their schedule)
- Transfer status (to see only courses that transfer to UNO)
- Course level (introductory, intermediate, advanced)

These filters can be combined, so a student could search for "3-credit biology courses that transfer to UNO" and get exactly the results they need.

Real-Time Results

The search updates in real-time as students type, providing immediate feedback. To balance responsiveness with performance, we use a technique called "debouncing"—the system waits 300 milliseconds after the user stops typing before executing the search. This prevents the system from being overwhelmed by searches for every keystroke while still feeling instantaneous to users.

Transfer Planning Features

Beyond just searching for courses, the application provides comprehensive transfer planning tools that help students map out their entire academic journey.

Personalized Transfer Plans

Students can create custom transfer plans that include:

- Courses they've already completed at Delgado
- Courses they plan to take at Delgado
- Their intended UNO major and degree requirements
- A semester-by-semester timeline for completion

These plans are saved to their account and can be modified as their academic goals evolve.

Progress Tracking

The application automatically calculates:

- Total credits earned and remaining
- Percentage of degree requirements completed
- Estimated graduation timeline
- Prerequisites still needed for planned courses

This information helps students stay on track and make informed decisions about course selection and timing.

Degree Audit Integration

The system can integrate with existing degree audit systems to provide comprehensive academic advising support. Students can see not just how their courses transfer, but how they fit into their overall degree plan.

Administrative Features

While students are the primary users, the application also includes robust administrative tools for academic advisors and institutional administrators.

Equivalency Management

Administrators can:

- Add new course equivalencies as curriculum changes
- Modify existing equivalencies when transfer policies update
- Review and approve proposed equivalencies
- Generate reports on transfer patterns and success rates

User Management

Administrative users can manage student accounts, reset passwords, and provide technical support. The system maintains detailed logs of all administrative actions for audit purposes.

Data Analytics

The application provides insights into:

- Most frequently searched courses
- Common transfer patterns
- Student success rates after transfer
- System usage statistics

This data helps institutions improve their transfer processes and identify areas where additional support might be needed.

Security and Privacy

Academic information is sensitive, and the application implements comprehensive security measures to protect student data and system integrity.

Authentication and Authorization

The system uses industry-standard JWT (JSON Web Token) authentication, which provides secure, stateless user sessions. Students log in once and can access their personalized features without repeatedly entering credentials.

User passwords are protected using bcrypt hashing with 10 rounds of salt, making them extremely difficult to compromise even if the database were breached.

Data Protection

All user data is protected by:

- Encryption in transit (HTTPS)
- Secure database storage with access controls
- Regular security updates and patches
- Comprehensive backup and disaster recovery procedures

Privacy Compliance

The application is designed to comply with FERPA (Family Educational Rights and Privacy Act) and other relevant privacy regulations. Student data is used only for the purposes of academic advising and transfer planning, and is never shared with unauthorized parties.

Performance and Scalability

The application is designed to handle the demands of a large student population while maintaining fast response times.

Database Optimization

Strategic database indexing ensures that even complex searches across thousands of courses return results in under 200 milliseconds. The system uses connection pooling to efficiently manage database resources and can handle hundreds of concurrent users.

Frontend Performance

The React frontend is optimized for fast loading and smooth interaction:

- Code splitting reduces initial load times
- Efficient rendering prevents unnecessary screen updates
- Caching strategies minimize redundant data requests
- Responsive design ensures good performance on all devices

Scalability Architecture

The system is designed to scale horizontally as usage grows:

- Stateless backend servers can be load-balanced
- Database read replicas can handle increased search volume
- CDN integration provides fast static asset delivery
- Monitoring and alerting systems track performance metrics

Deployment and Infrastructure

The application is designed for deployment in a self-hosted datacenter environment, providing institutions with full control over their data and systems.

Server Requirements

Application Server:

- 4 CPU cores for handling concurrent user requests
- 8GB RAM for application caching and session management
- 100GB SSD storage for fast application startup and file operations

Database Server:

- 4 CPU cores for query processing
- 16GB RAM for database caching and buffer pools
- 500GB SSD storage with RAID 1 for data redundancy

Load Balancer:

- 2 CPU cores for request routing
- 4GB RAM for connection management
- 50GB storage for logs and configuration

High Availability Setup

The system includes comprehensive high availability features:

- Application clustering with automatic failover
- Database replication for redundancy
- Load balancing across multiple servers
- Automated backup and recovery procedures

Security Hardening

Production deployments include:

- Firewall configuration with minimal open ports
- SSL certificate management with automatic renewal
- Regular security updates and patches
- Intrusion detection and monitoring systems

Development Workflow

The application is built using modern development practices that ensure code quality, maintainability, and reliable deployment.

Technology Stack

Backend Dependencies:

```
npm install express mysql2 cors bcrypt jsonwebtoken dotenv
```

Frontend Dependencies:

```
npx create-react-app delgado-uno-transfer  
npm install axios
```

Development Environment

Developers can quickly set up a local development environment:

```
# Start backend server  
npm run dev  
  
# Start frontend development server  
npm start  
  
# Access application  
# Frontend: http://localhost:3000  
# Backend API: http://localhost:5000
```

Code Organization

The codebase is organized into logical modules:

- Backend API routes and middleware
- Frontend React components and styling
- Database schema and migration scripts
- Configuration and environment management

Testing Strategy

Comprehensive testing ensures the application works reliably across different scenarios and use cases.

Frontend Testing

React components are tested using Jest and React Testing Library:

```
describe('CourseSearch Component', () => {
  test('renders search interface correctly', () => {
    // Test component rendering
  });

  test('handles user input and search', () => {
    // Test search functionality
  });

  test('displays course results properly', () => {
    // Test result display
  });
});
```

Backend Testing

API endpoints are tested for functionality, security, and performance:

```
describe('Course Search API', () => {
  test('returns correct search results', async () => {
    // Test search logic
  });

  test('handles authentication properly', async () => {
    // Test security measures
  });

  test('validates input parameters', async () => {
    // Test input validation
  });
});
```

Integration Testing

End-to-end tests verify complete user workflows:

- User registration and login
- Course search and filtering
- Transfer plan creation and management
- Administrative functions

Monitoring and Maintenance

Ongoing monitoring ensures the application continues to perform well and serve students effectively.

Performance Monitoring

The system tracks key performance metrics:

- Search response times
- Database query performance
- User session duration
- Error rates and types

Usage Analytics

Understanding how students use the application helps guide future improvements:

- Most popular courses and searches
- Common user paths through the application
- Feature adoption rates
- Student success metrics

Maintenance Procedures

Regular maintenance ensures continued reliability:

- Weekly database optimization
- Monthly security updates
- Quarterly backup testing
- Annual security audits

Future Enhancements

The current application provides a solid foundation, but there are many opportunities for enhancement and expansion.

Advanced Search Features

Future versions could include:

- AI-powered course recommendations
- Natural language search ("find me calculus courses")
- Visual course pathway mapping
- Integration with academic planning tools

Mobile Application

A dedicated mobile app could provide:

- Offline access to course information
- Push notifications for important deadlines
- Barcode scanning for textbook integration
- Calendar integration for academic planning

Institutional Expansion

The platform could be expanded to include:

- Additional community colleges in the region
- Four-year universities beyond UNO
- Graduate program transfer pathways
- Professional certification programs

Advanced Analytics

Enhanced data analysis could provide:

- Predictive modeling for student success
- Automated equivalency suggestions
- Transfer trend analysis
- Institutional benchmarking

Conclusion

The Delgado-UNO Course Transfer Application represents a significant improvement in how students navigate the college transfer process. By combining modern web technologies with thoughtful user experience design, the application makes complex academic information accessible and actionable.

The system's strength lies in its focus on real student needs rather than administrative convenience. Every feature, from the intelligent search functionality to the personalized transfer planning tools, is designed to help students make informed decisions about their academic future.

From a technical perspective, the application demonstrates that robust, scalable educational technology doesn't require cutting-edge complexity. By choosing proven technologies and implementing them thoughtfully, we've created a system that is both powerful and maintainable.

The comprehensive security measures, performance optimizations, and deployment strategies ensure that the application can serve students reliably while protecting their sensitive academic information.

Perhaps most importantly, the application is designed to evolve. The modular architecture, comprehensive testing, and monitoring systems provide a foundation for continuous improvement based on student feedback and changing institutional needs.

As educational institutions continue to recognize the importance of seamless transfer processes, applications like this one will play an increasingly vital role in student success. By removing barriers to information and providing tools for academic planning, we can help more students achieve their educational goals efficiently and effectively.

The future of academic advising lies in systems that combine institutional knowledge with student-centered design. This application represents a step toward that future—one where students have the information and tools they need to make confident decisions about their academic journey.