CSE251 - Software Engineering

University system project

Project Manager:

Adham Ayman Farouk Ibrahim   21100782

Team Members:

Abdelrahman Mohamed Abdelaziz   21100799

Ahmed Mohamed Mahdy   21100822

Mazen Mohamed Elsayed   21100912

Ahmed Mokhtar Mokhtar   21100781

Supervised By :

Dr. Islam Elgedawy

Eng. Ahmed Shoukry

Eng. Ahmed Samir

Eng. Amal Hafny

System and User Requirements:

- **User-Friendly Interface:** The system should be easy to use and navigate for all users, including students, faculty, staff, and administrators. The interface should be intuitive and visually appealing, with clear labeling and instructions.

- **Integration with Existing Systems:** The system should be able to integrate with the university's existing systems, such as the student information system, the learning management system, and the financial management system. This will enable seamless data transfer and avoid duplication of effort.

- Security and Privacy: The system should be secure and protect sensitive data, including student records,

- financial information, and personal data. The system should comply with applicable data protection regulations, such as GDPR and HIPAA

- Scalability: The system should be scalable to accommodate the growing needs of the university, such as an increase in the number of students, faculty, or programs.

- Analytics and Reporting: The system should provide robust analytics and reporting capabilities to enable administrators to track key metrics, such as enrollment, retention, and graduation rates.

- Support and Training: The system should come with comprehensive support and training resources, including user manuals, video tutorials, and online

forums, to ensure that users can use the system effectively.

- Multi-Language Support: The system should support multiple languages, to cater to users from diverse linguistic backgrounds. This will ensure that all users can access the system and use it effectively.

- Audit Trails: The system should provide audit trails, to enable administrators to track user activity and identify any unauthorized or malicious actions. This will enhance the system's security and accountability.

- Flexibility: The system should be flexible enough to accommodate different types of programs, such as traditional classroom-based courses, online courses, and hybrid courses.

- This will enable the university to offer a variety of programs to meet the needs of its students.

- Feedback and Evaluation: The system should provide tools for feedback and evaluation, to enable users to provide feedback on courses, instructors, and the system itself. This will enable the university to improve its programs and services based on user feedback.

Subsystems:

- ## Student Information System Interface:

  - List<Course> getCourses();

  - registerForCourse(Course course, Student student);

  - dropCourse(Course course, Student student);

  - Student getStudent(String studentId);

  - List<Student> getStudentsByCourse(Course course);

  - List<Grade> getGradesByStudent(Student student);

- ## Course Management System Interface:

  - List<Course> getCourses();

  - addCourse(Course course);

  - removeCourse(Course course);

  - assignFaculty(Course course, Faculty faculty);

  - List<Attendance> getAttendanceByCourse(Course course);

  - RecordGrade(Grade grade, Student student, Course course);

- **Faculty Information System Interface:**

  o public List<Faculty>getFaculty();

  o addFaculty (Faculty faculty);

  o removeFaculty (Faculty faculty);

  o List<Course> getCourses ByFaculty(Faculty faculty);

  o List<Faculty>getFacultyByCourse(Course course);

  o List<Salary>getSalaryByFaculty(Faculty faculty);

- **Financial Management System Interface**

  o List<Transaction> getTransactions();

  o void add Transaction(Transaction transaction);

  o remove Transaction(Transaction transaction);

  o List<StudentBill> getStudentBills();

  o List<FacultyPayroll>getFacultyPayrolls();

  o List<Grant> getGrants();

- **Research Management System Interface:**

  o `List<Grant> getGrants();`

  o `addGrant(Grant grant);`

  o `removeGrant(Grant grant);`

  o `List<ResearchPublication> getPublications();`

  o `addPublication(Research Publication publication);`

  o `removePublication(Research Publication publication);`

- **Campus Security System Interface**

  o `grantAccess(Student student; AccessLevel accessLevel);`

  o `revokeAccess(Student student, AccessLevel accessLevel);`

  o `List<AccessLog> getAccessLogs();`

  o `handleEmergency (Emergency emergency);`

  o `List<Student> getStudentsInBuilding(Building building);`

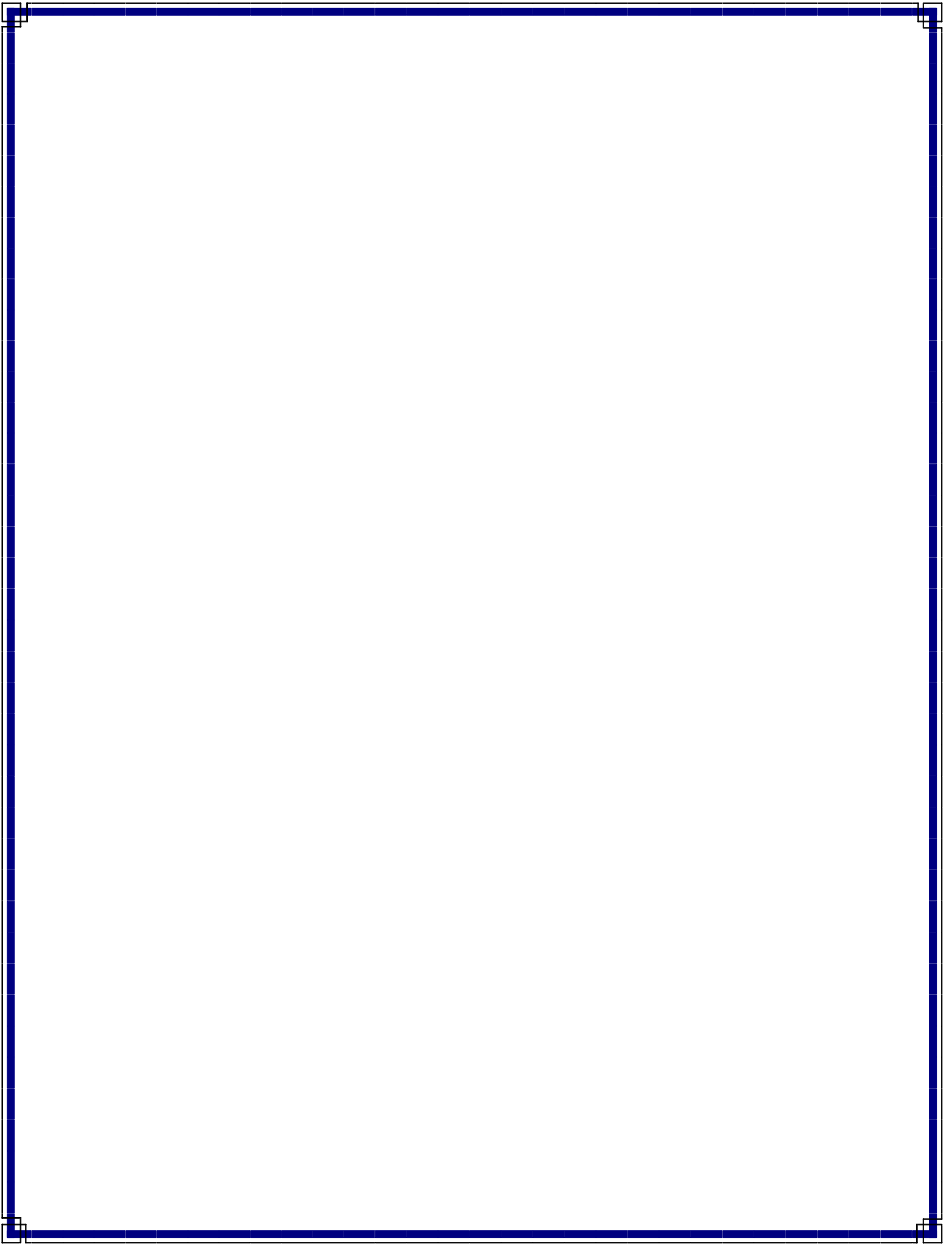- **Event Management System Interface:**

  o `List<Event> getEvents();`

- addEvent(Event event); removeEvent(Event event);

- updateEvent(Event event);

- List<Event> getEventsByDate(Date date);

- List<Event> getEventsByLocation(String location); List<Event>

  getEventsByType(String eventType);

- List<EventAttendee> getAttendeesByEvent(Event event);

- addAttendee(EventAttendee attendee, Event event);

- removeAttendee(EventAttendee attendee, Event event);

- List<Event> getEventsByAttendee (Student student);

- List<Event> getUpcoming Events();

- List<Event> getPastEvents();

▪ Human Resources Management System

- List<Employee> getEmployees(); addEmployee(Employee employee);

- removeEmployee (Employee employee);

- updateEmployee (Employee employee); List<Employee>

  getEmployeesByDepartment(String department);

- List<Employee> getEmployeesByPosition(String position);

  List<Employee> getEmployeesBySalary Range(double minSalary,

- double maxSalary);

- assignEmployee To Department(Employee employee, String

  department);

- assignEmployee To Position (Employee employee, String

- position);

- increaseEmployee Salary (Employee employee, double amount);

  decreaseEmployeeSalary(Employee employee, double amount);

- List<JobOpening> getJobOpenings();

- addJobOpening(JobOpening jobOpening);

  removeJobOpening(JobOpening jobOpening);

- updateJobOpening (JobOpening jobOpening);

- ·· ··. . . addJobApplication (JobApplication jobApplication);

- List<JobOpening> getJobOpeningsByDepartment(String

- department);

- List<JobOpening> getJobOpeningsByPosition(String position);

- List<JobOpening> getJobOpeningsBySalaryRange(double minSalary, double maxSalary);

- List<JobApplication> getJobApplications();

- removeJobApplication (JobApplication jobApplication);

  updateJobApplication(JobApplication jobApplication);

  List<JobApplication>

- getJobApplicationsByJobOpening(JobOpening jobOpening);

- List<JobApplication> getJobApplicationsByApplicant(Student student);

- hireEmployee (Employee employee, JobApplication jobApplication);

REQUIREMENTS TRACEABILITY MATRIX

Project Name	University System		Date	2/26/2023			

Requirements	Student Information System Interface	Course Management System Interface	Faculty Information System Interface	Financial Management System Interface	Research Management System Interface	Campus Security System Interface	Event Management System Interface	Human Resources Management System
User-Friendly Interface	1	1	0	0	0	0	0	0
Integration with Existing Systems	1	0	0	1	0	0	0	0
Security and Privacy	1	1	1	1	1	1	1	0
Scalability	1	1	1	0	0	0	0	0
Analytics and Reporting	0	1	0	1	0	0	0	1
Support and Training	1	0	0	0	0	0	0	1
Multi-Language Support	1	1	0	1	1	0	1	0
Audit Trails	0	0	1	0	0	1	0	1
Flexibility	0	1	0	0	0	0	0	0
Feedback and Evaluation	1	0	0	0	0	0	0	1

User manual

Introducution

Welcome to the user manual for this product! This manual has been created to help you understand how to use the features of the product effectively and safely.

Before using the product, we highly recommend that you read through this manual in its entirety. This will ensure that you are able to use the product efficiently, and also help prevent any potential accidents or damage to the product.

This manual is designed to be easy to read and understand. We have included step-by-step instructions, diagrams, and other visual aids to make it as user-friendly as possible.

In addition to this manual, there may be other materials included with the product such as warranty information, safety guidelines, or troubleshooting instructions. Be sure to review all of these materials thoroughly before using the product.

If you have any questions or concerns while using the product, please consult this manual first. If you are unable to find the answers you need, do not hesitate to contact our customer support team for assistance.

Thank you for choosing our product, and we hope that this manual will help you enjoy using it to its fullest potential.

Log in interface

Home Page

This is log in interface.

Email

Email

[   ]

Here write your acadmic email.

Password

password

[   ]

Here write your acadmic password.

Your password should not be less than 8 words and must contain captial letter.
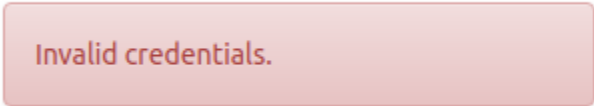
Role of user

◯ Admin   ◯ Instructor   ◯ Student

Please select your role in university.

Sign in

After entering your data click *sign in*

Cant sign in

This massage will apper if you put your data wrong.

Please revise your input data or contact our IT.

Student Information System

Manage student enrollment, course registration, and scheduling

Maintain student records, including personal information, academic records, and financial aid details

Generate reports and dashboards for student performance, enrollment trends, and financial aid disbursement

Interface with other subsystems, such as the course management system and the financial management system

back

student photo

student infotmation

View Grades

details about student grades

Course Management System

Manage course information, including course descriptions, prerequisites, schedules, and availability

Assign faculty members and course assistants to courses

Maintain course records, including student enrollment, attendance, and grades

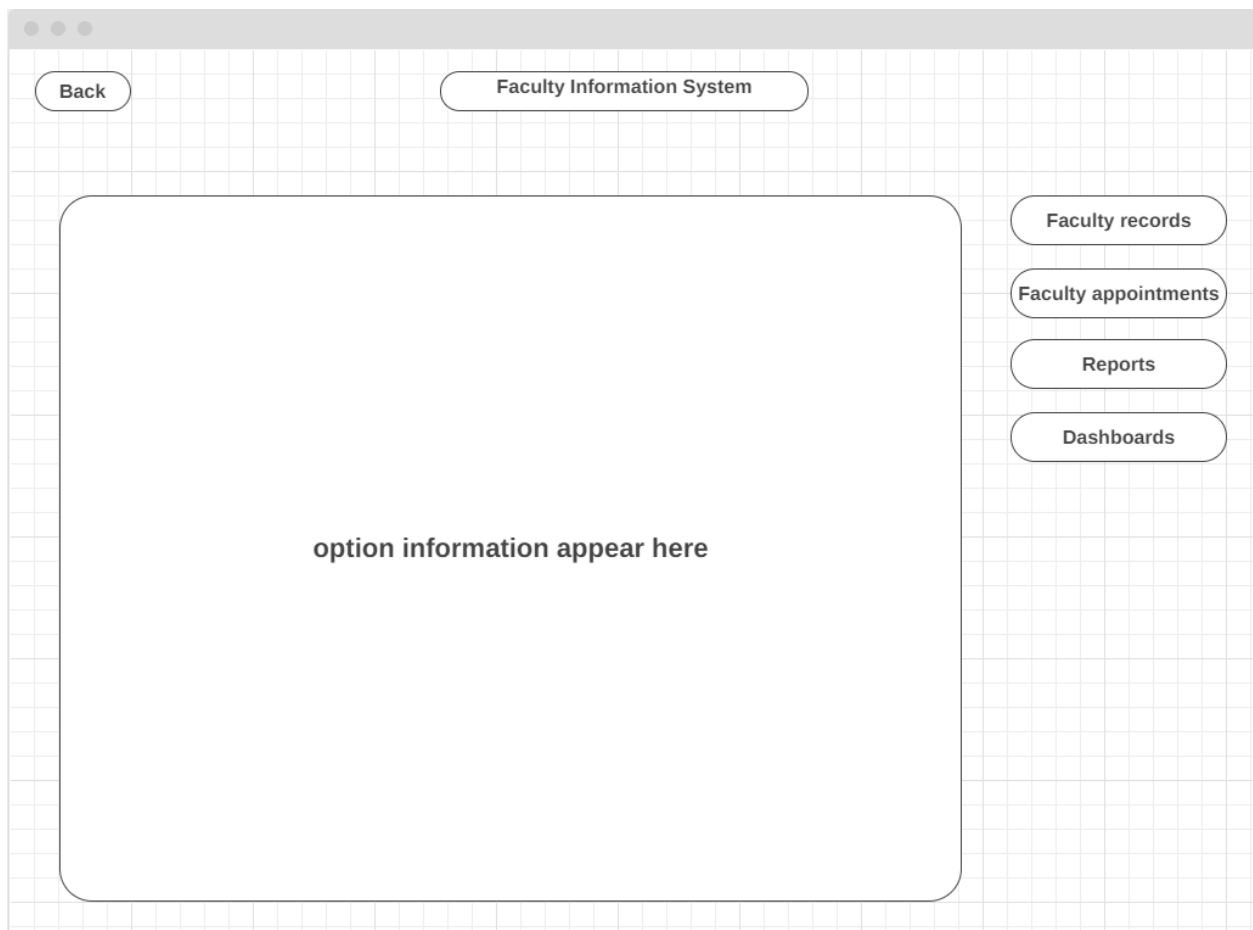Generate reports and dashboards for course performance, faculty workload, and student evaluation
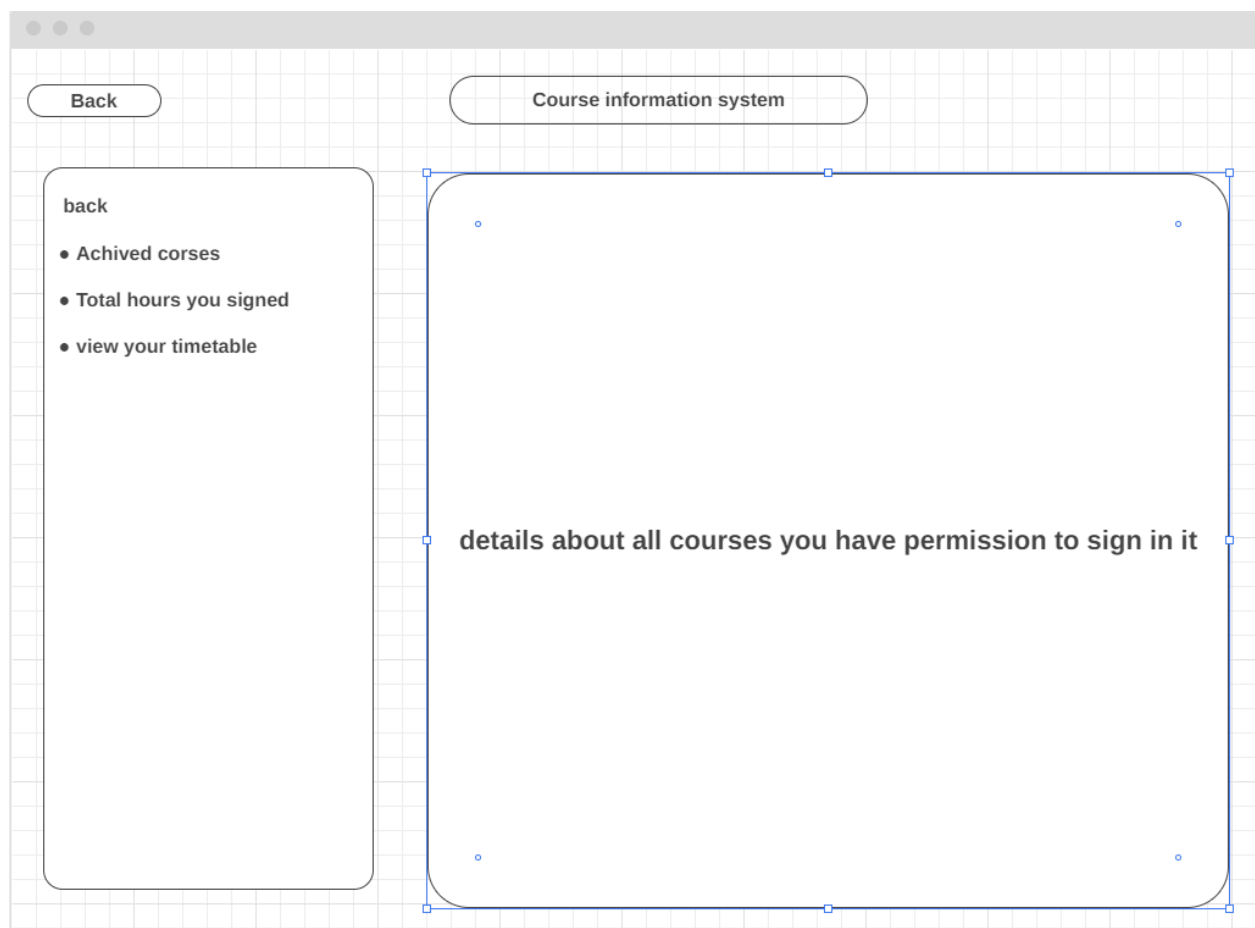
Faculty Information System

Maintain faculty records, including personal information, academic records, and teaching assignments

Manage faculty appointments, scheduling, and evaluation

Generate reports and dashboards for faculty performance, research output, and professional development

Back

Faculty Information System

Faculty records

Faculty appointments

Reports

Dashboards

option information appear here

Financial Management System

Manage financial information, including tuition fees, student billing, payroll, and budgeting

Generate financial reports and dashboards for budget tracking, revenue projections, and expense analysis

Interface with other subsystems, such as the student information system and the housing management system, to handle financial transactions and account reconciliation

Back	Financial Management System

| Tuition fees | option details appear here |
Student billing	
payroll and budgeting	
Reports	
Dashboard	
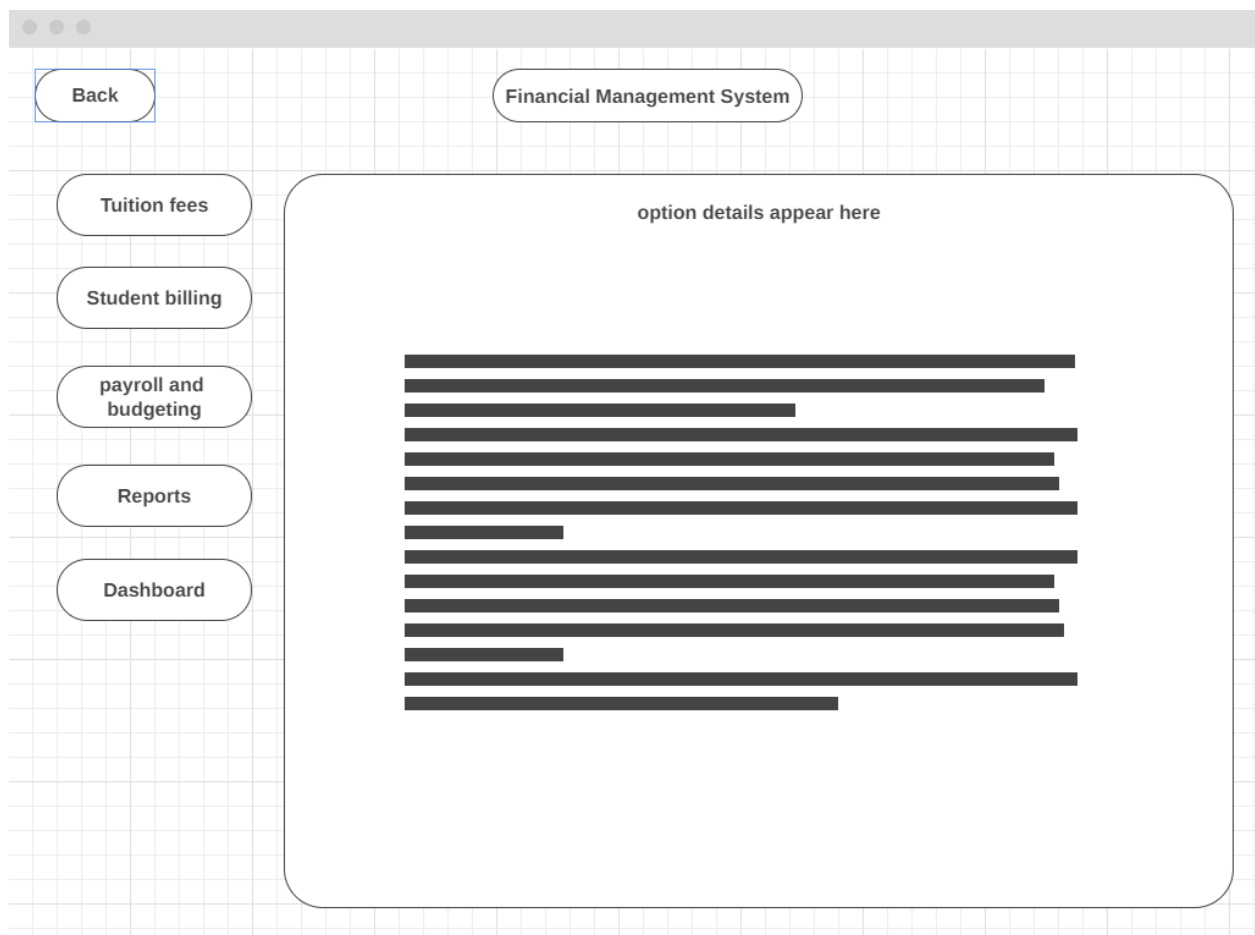
Research Management System

Manage research information, including grant applications, funding sources, and research outcomes

Maintain research records, including intellectual property rights, collaborations, and publications

Generate reports and dashboards for research output, funding utilization, and citation impact

Back

Research Management System

Research information

Research records

Reports

option details appear here
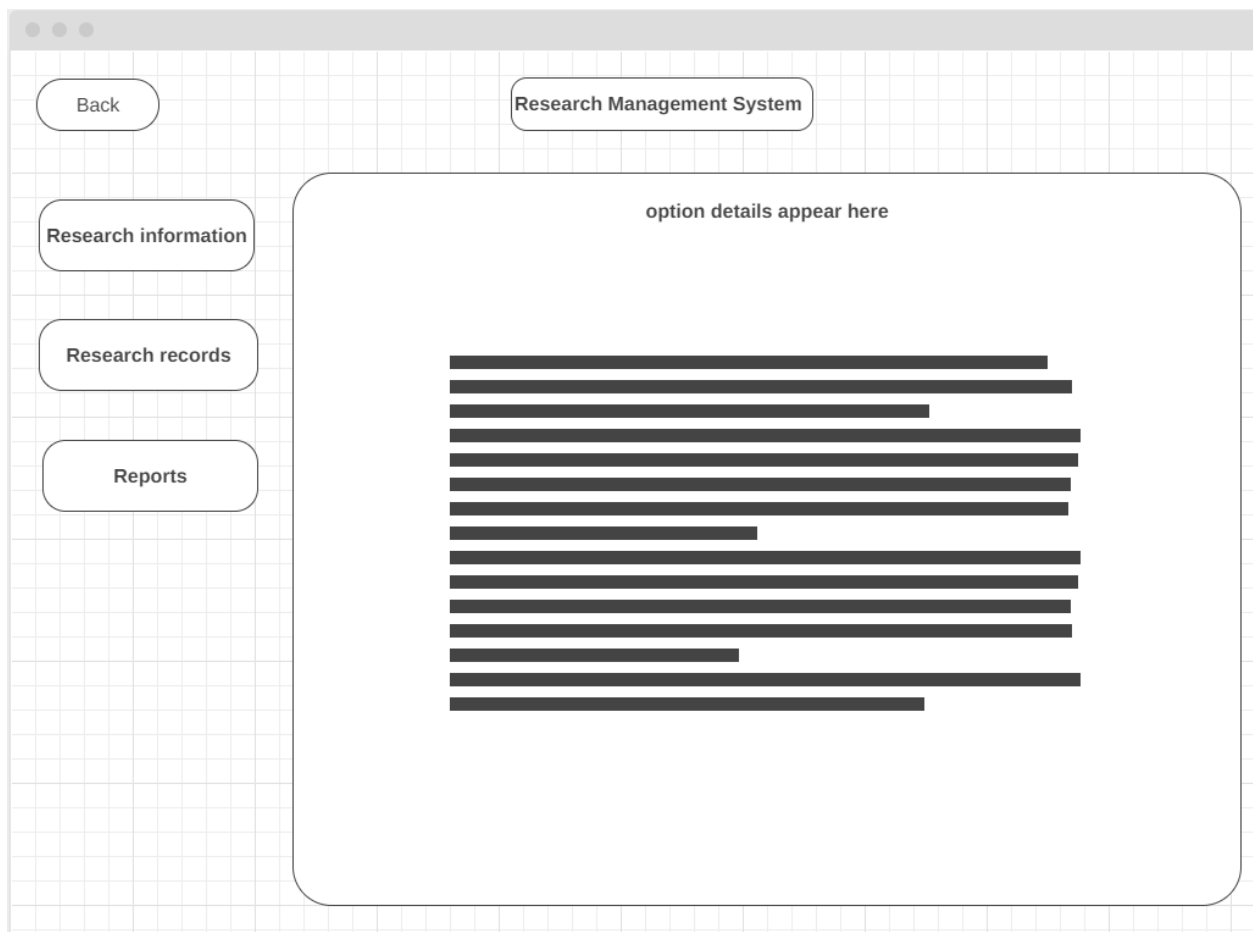
Campus Security System

Manage campus security information, including access control, surveillance, and emergency response

Maintain security records, including incident reporting, risk assessment, and training

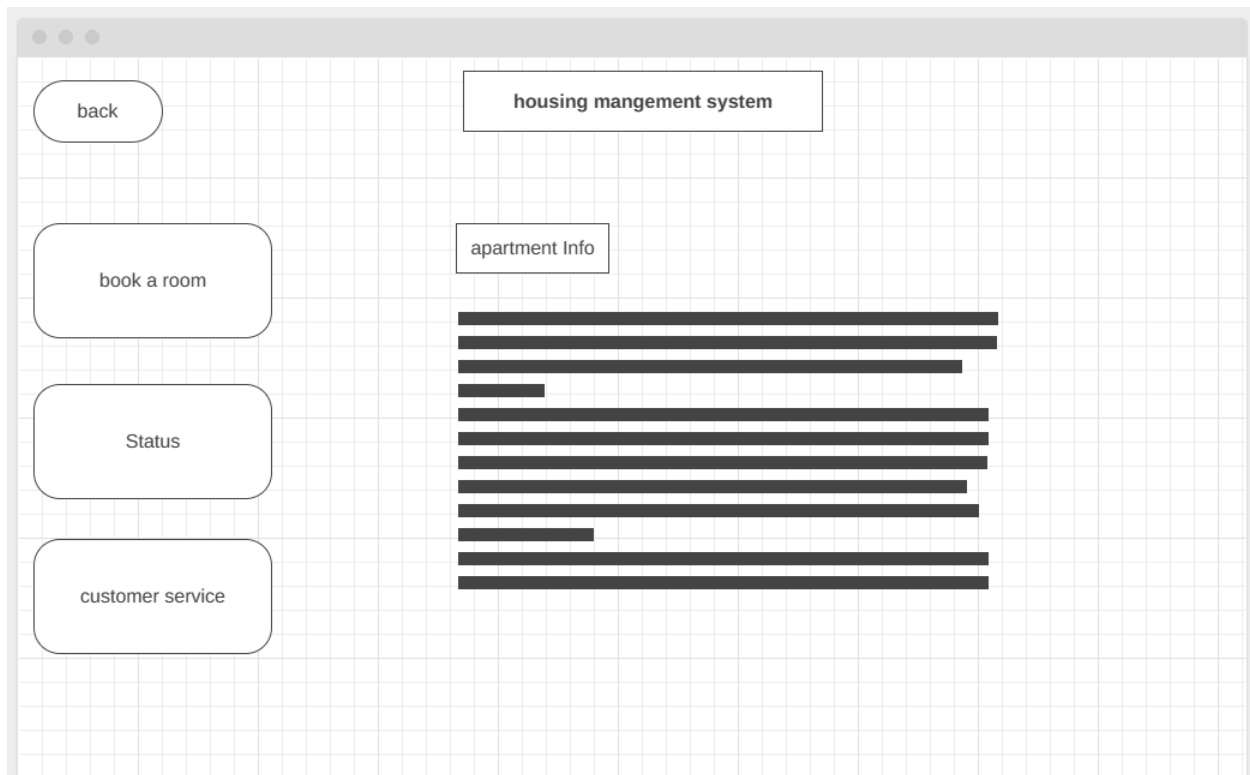Generate reports and dashboards for security incidents, safety compliance, and risk mitigation
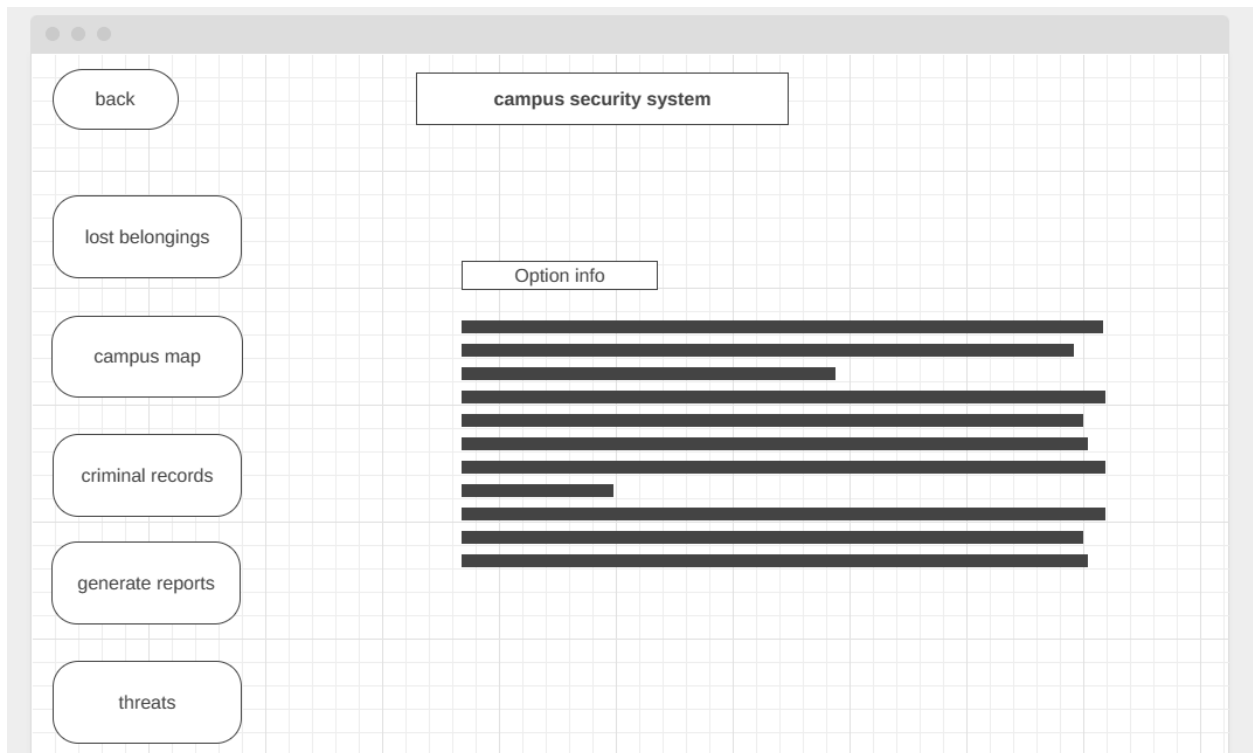
Housing Management System

Manage housing information, including room assignments, roommate matching, and maintenance requests

Maintain housing records, including occupancy rates, rental agreements, and housing policies

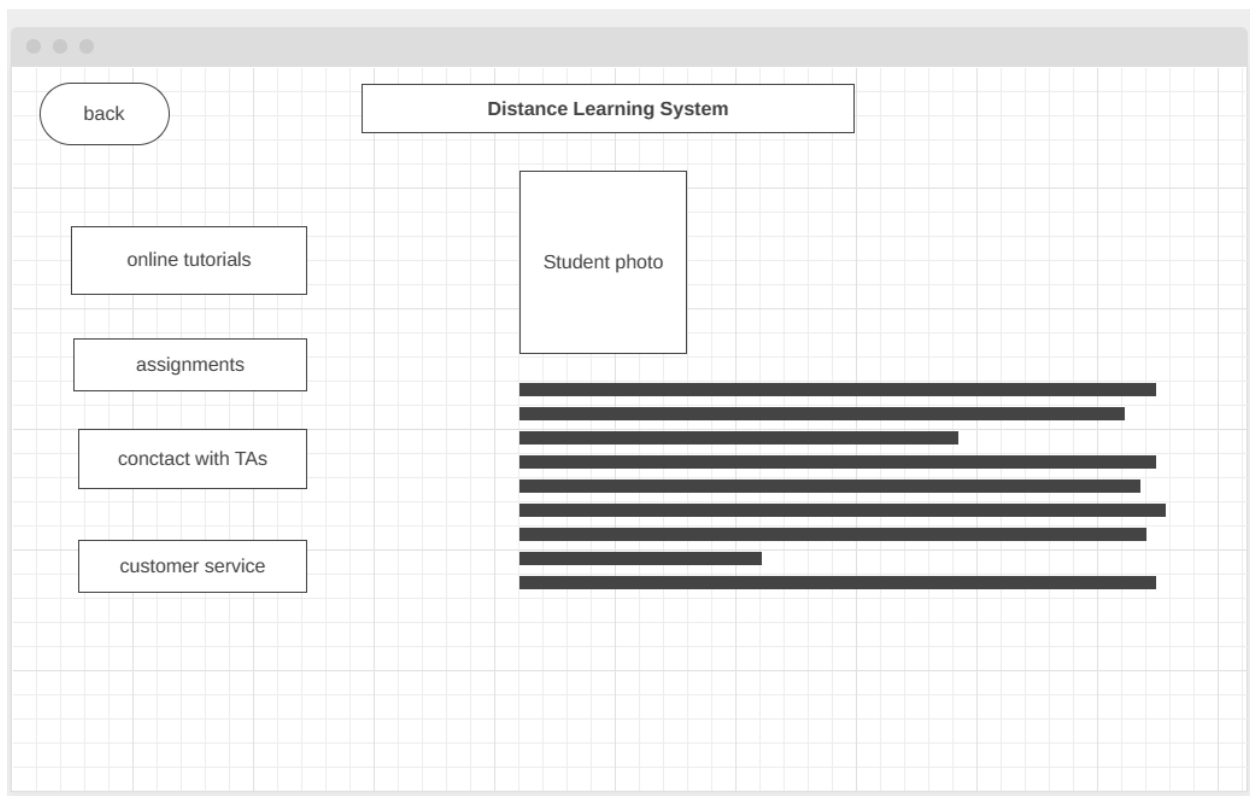Generate reports and dashboards for housing occupancy, rent collection, and maintenance tracking

Distance Learning System

Manage online course information, including course materials, assignments, and assessments

Maintain online course records, including student enrollment, attendance, and grades

Generate reports and dashboards for online course performance, student engagement, and retention

Human Resources Management System

Manage employee information, including job

Maintain employee records, including payroll, tax withholdings, and time-off requests

Generate reports and dashboards for employee performance, turnover, and workforce demographics
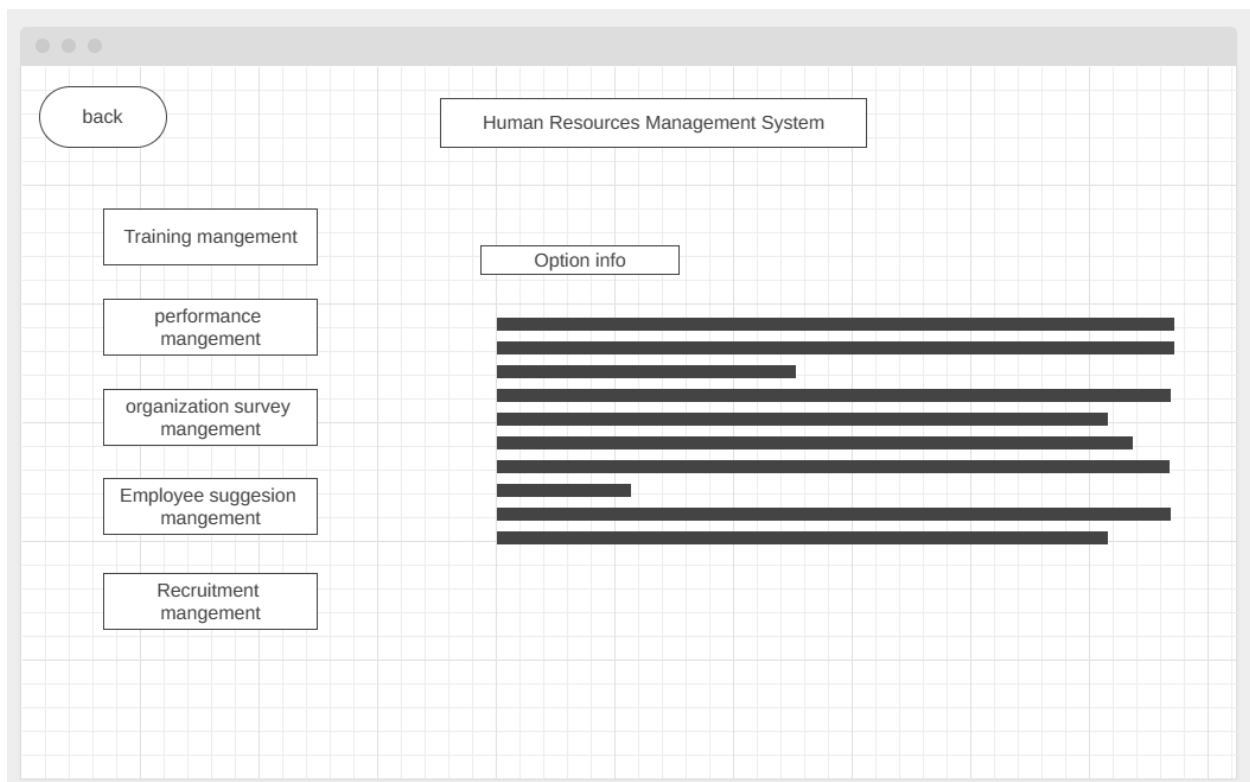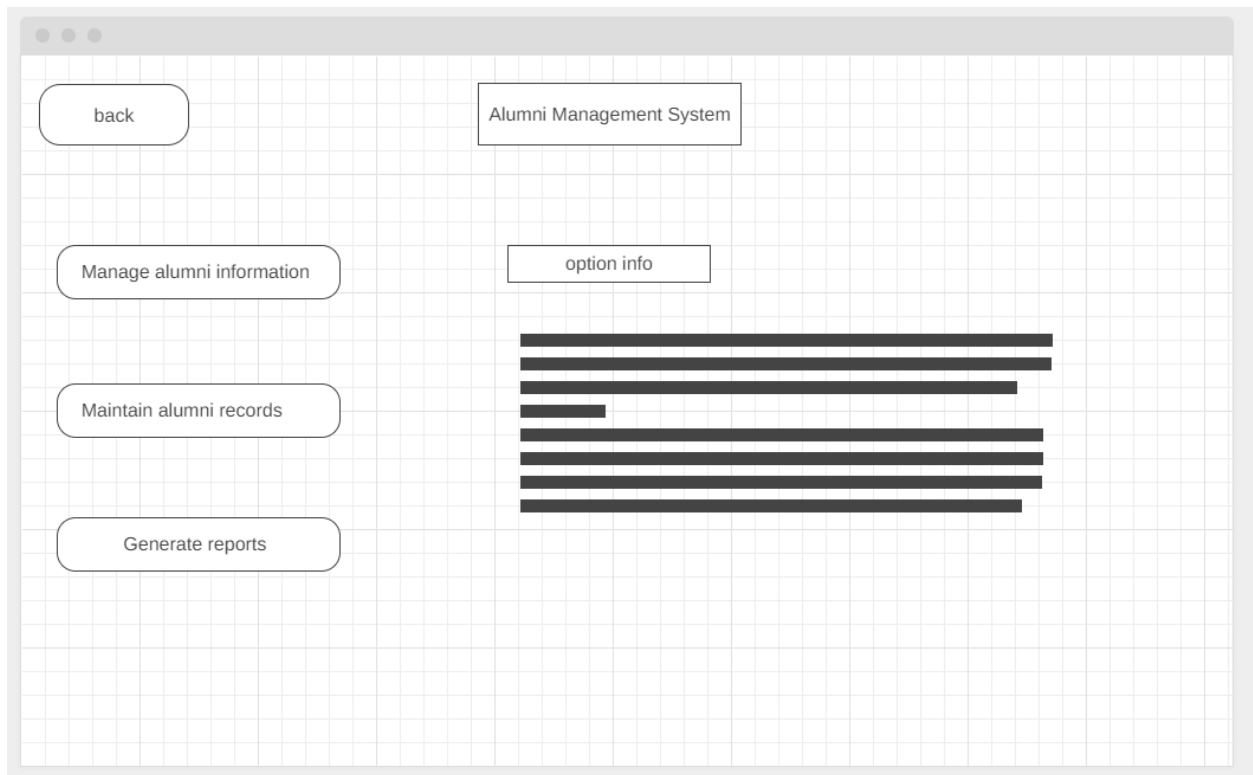
Alumni Management System

Manage alumni information, including contact information, employment history, and donation history

Maintain alumni records, including event attendance, mentoring activities, and career development

Generate reports and dashboards for alumni engagement, job placement, and fundraising

Marketing and Communications System

Manage marketing and communications information, including branding, advertising, and public relations

Maintain marketing and communications records, including social media accounts, email lists, and website content

Generate reports and dashboards for marketing and communications campaigns, web analytics, and audience engagement
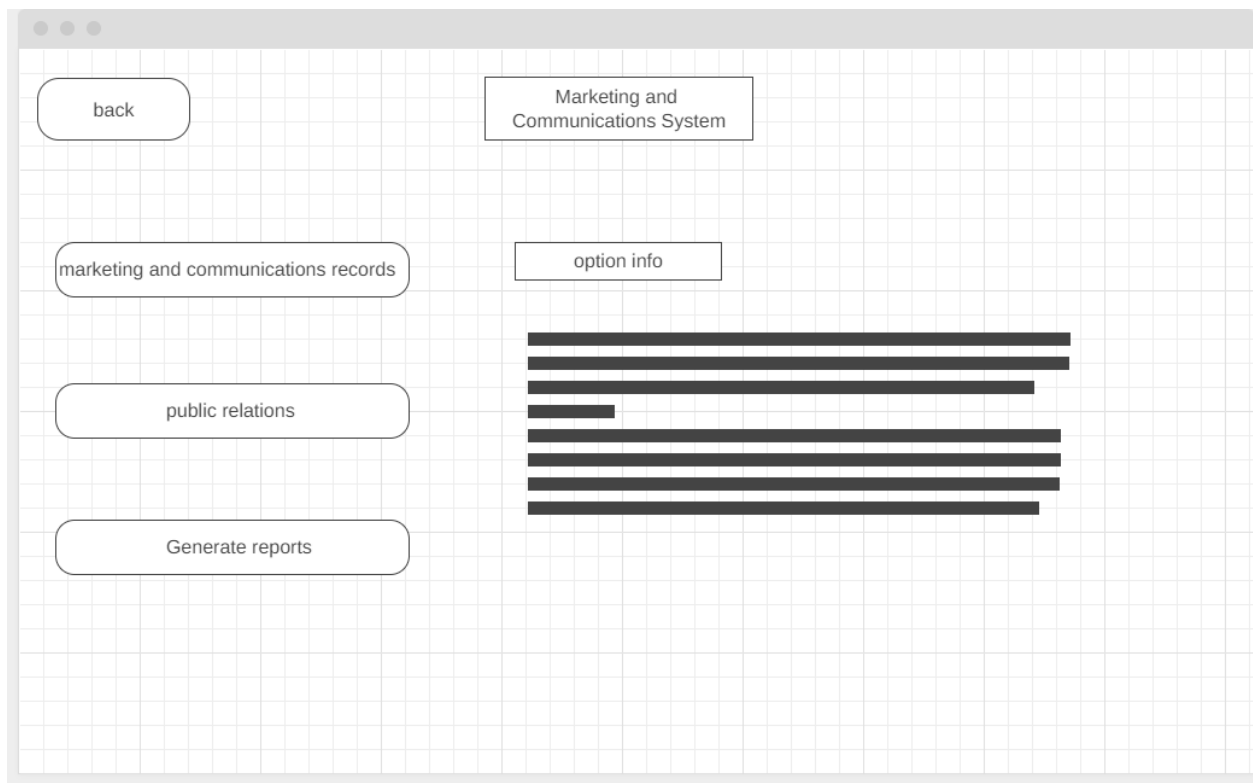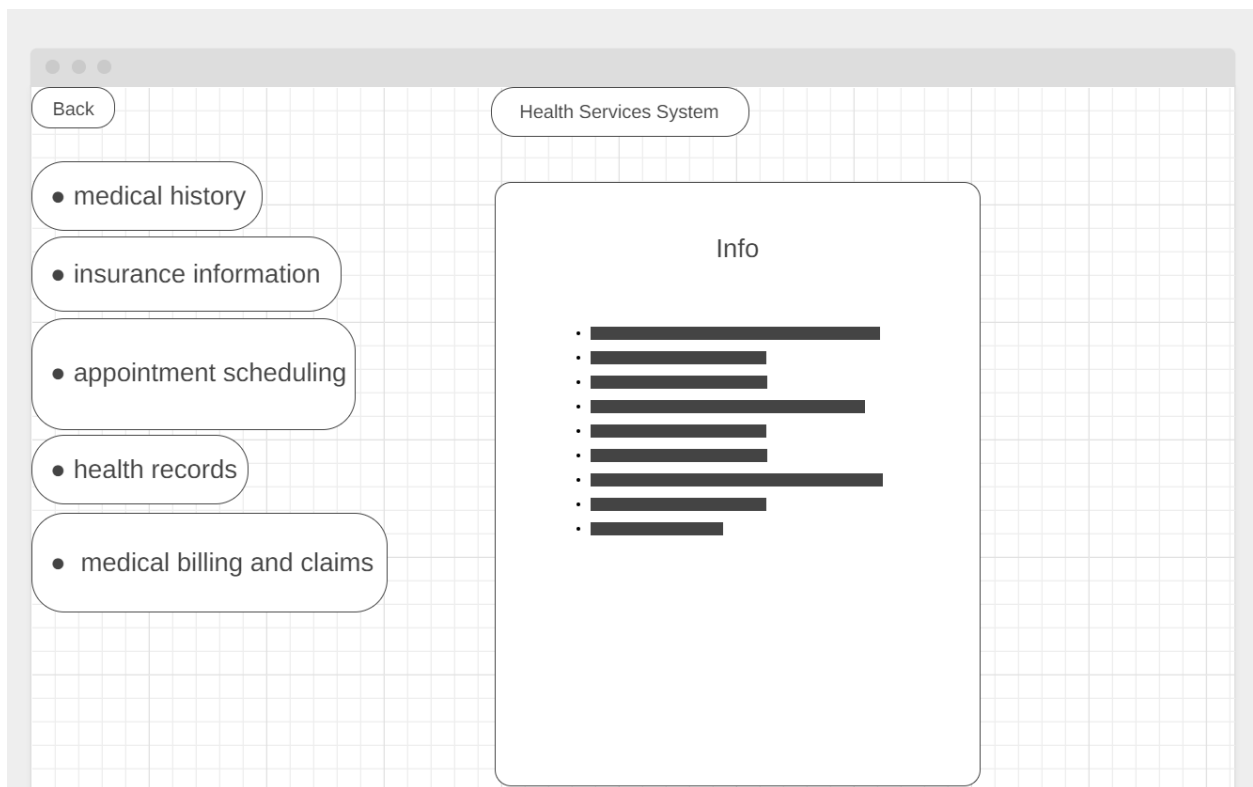
Health Services System

Manage health information, including medical history, insurance information, and appointment scheduling

Maintain health records, including medical billing and claims, and patient communication

Generate reports and dashboards for health outcomes, insurance utilization, and patient satisfaction
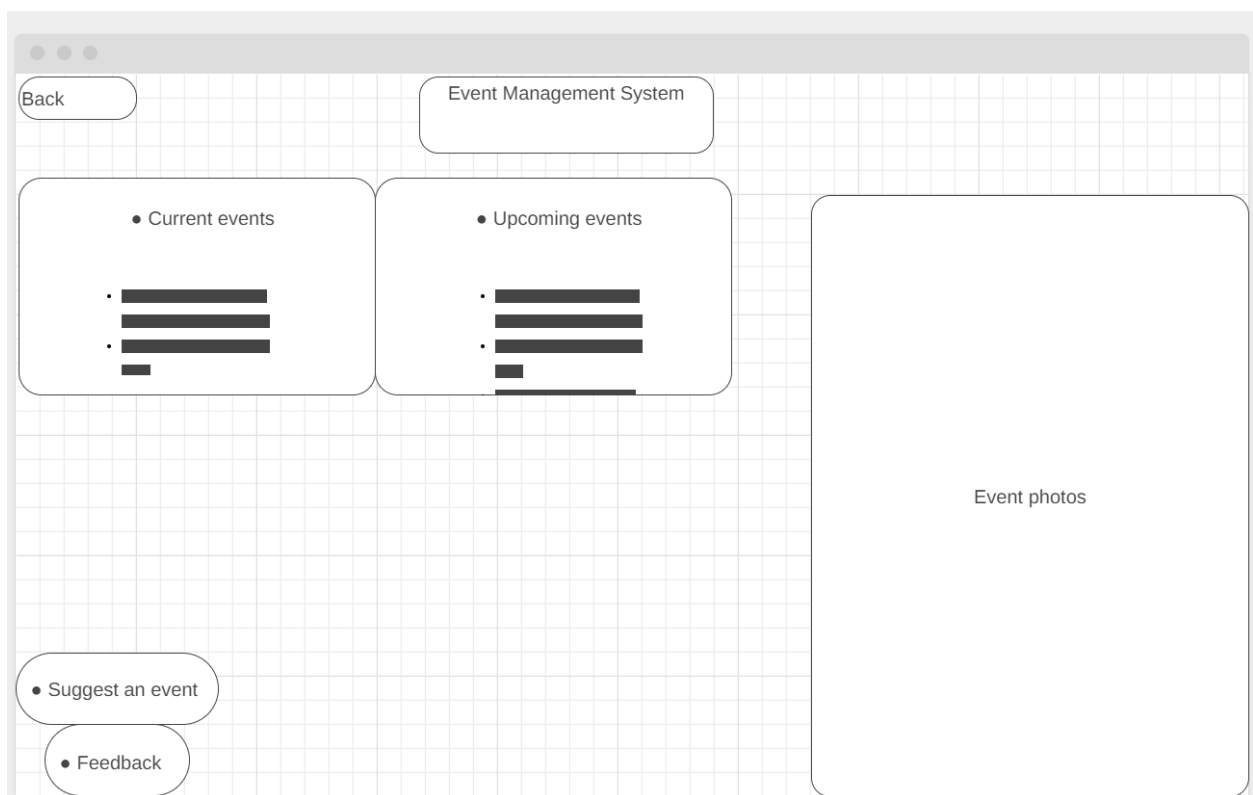
Event Management System

Manage campus event information, including scheduling, registration, and logistics

Maintain event records, including attendance, feedback, and evaluation

Generate reports and dashboards for event attendance, satisfaction, and impact

Back

Event Management System

● Current events

● Upcoming events

Event photos

● Suggest an event

● Feedback

Class Diagram