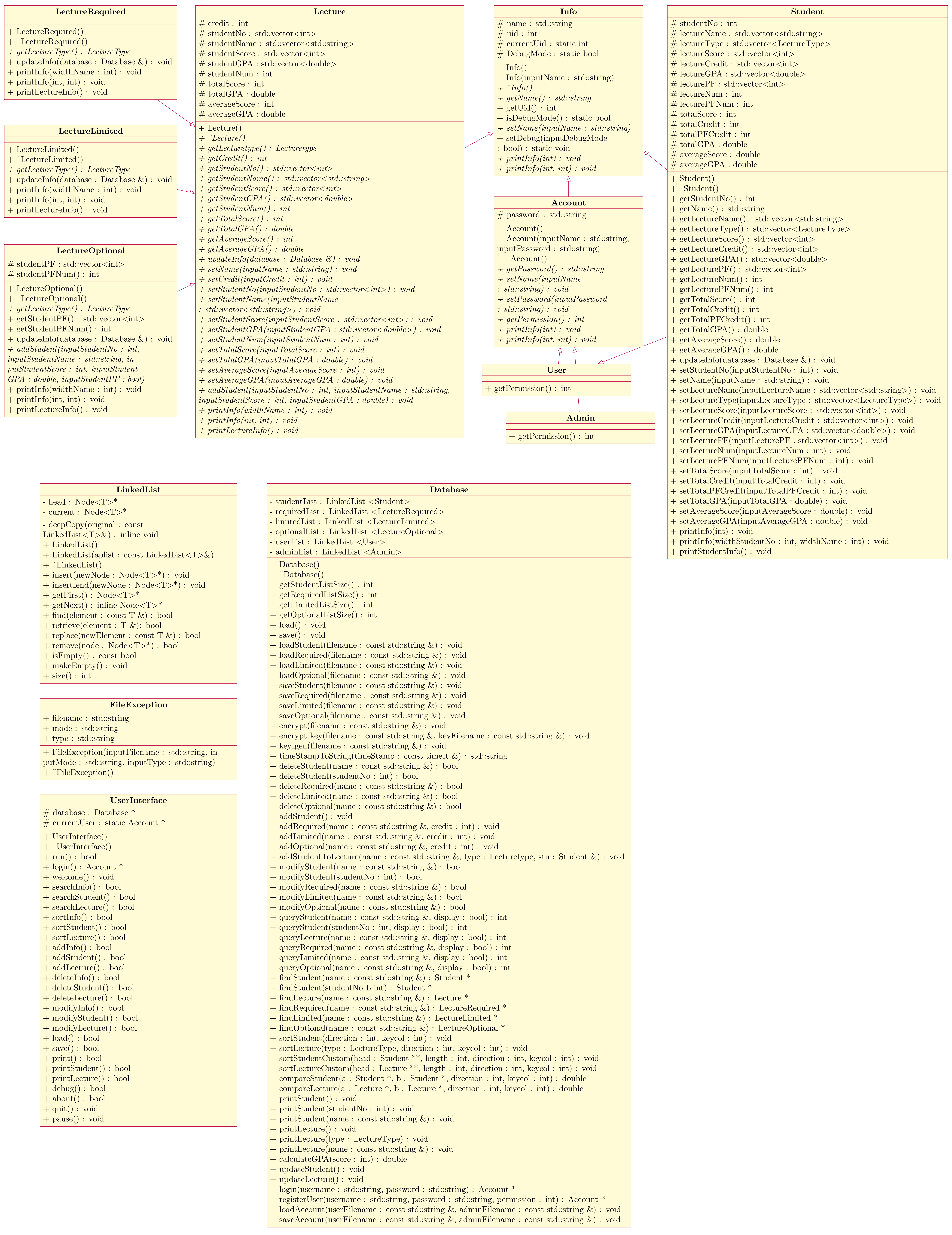
LectureRequired
+ LectureRequired()
+ ˜LectureRequired()
+ getLectureType() : LectureType
+ updateInfo(database : Database &) : void
+ printInfo(widthName : int) : void
+ printInfo(int, int) : void
+ printLectureInfo() : void
LectureLimited
+ LectureLimited()
+ ˜LectureLimited()
+ getLectureType() : LectureType
+ updateInfo(database : Database &) : void
+ printInfo(widthName : int) : void
+ printInfo(int, int) : void
+ printLectureInfo() : void
LectureOptional
# studentPF : std::vector<int>
# studentPFNum() : int
+ LectureOptional()
+ ˜LectureOptional()
+ getLectureType() : LectureType
+ getStudentPF() : std::vector<int>
+ getStudentPFNum() : int
+ updateInfo(database : Database &) : void
+ addStudent(inputStudentNo : int, inputStudentName : std::string, inputStudentScore : int, inputStudentGPA : double, inputStudentPF : bool)
+ printInfo(widthName : int) : void
+ printInfo(int, int) : void
+ printLectureInfo() : void
Lecture
# credit : int
# studentNo : std::vector<int>
# studentName : std::vector<std::string>
# studentScore : std::vector<int>
# studentGPA : std::vector<double>
# studentNum : int
# totalScore : int
# totalGPA : double
# averageScore : int
# averageGPA : double
+ Lecture()
+ ˜Lecture()
+ getLecturetype() : Lecturetype
+ getCredit() : int
+ getStudentNo() : std::vector<int>
+ getStudentName() : std::vector<std::string>
+ getStudentScore() : std::vector<int>
+ getStudentGPA() : std::vector<double>
+ getStudentNum() : int
+ getTotalScore() : int
+ getTotalGPA() : double
+ getAverageScore() : int
+ getAverageGPA() : double
+ updateInfo(database : Database &) : void
+ setName(inputName : std::string) : void
+ setCredit(inputCredit : int) : void
+ setStudentNo(inputStudentNo : std::vector<int>) : void
+ setStudentName(inputStudentName : std::vector<std::string>) : void
+ setStudentScore(inputStudentScore : std::vector<int>) : void
+ setStudentGPA(inputStudentGPA : std::vector<double>) : void
+ setStudentNum(inputStudentNum : int) : void
+ setTotalScore(inputTotalScore : int) : void
+ setTotalGPA(inputTotalGPA : double) : void
+ setAverageScore(inputAverageScore : int) : void
+ setAverageGPA(inputAverageGPA : double) : void
+ addStudent(inputStudentNo : int, inputStudentName : std::string, inputStudentScore : int, inputStudentGPA : double) : void
+ printInfo(widthName : int) : void
+ printInfo(int, int) : void
+ printLectureInfo() : void
Info
# name : std::string
# uid : int
# currentUid : static int
# DebugMode : static bool
+ Info()
+ Info(inputName : std::string)
+ ˜Info()
+ getName() : std::string
+ getUid() : int
+ isDebugMode() : static bool
+ setName(inputName : std::string)
+ setDebug(inputDebugMode : bool) : static void
+ printInfo(int) : void
+ printInfo(int, int) : void
Account
# password : std::string
+ Account()
+ Account(inputName : std::string, inputPassword : std::string)
+ ˜Account()
+ getPassword() : std::string
+ setName(inputName : std::string) : void
+ setPassword(inputPassword : std::string) : void
+ getPermission() : int
+ printInfo(int) : void
+ printInfo(int, int) : void
User
+ getPermission() : int
Admin
+ getPermission() : int
Student
# studentNo : int
# lectureName : std::vector<std::string>
# lectureType : std::vector<LectureType>
# lectureScore : std::vector<int>
# lectureCredit : std::vector<int>
# lectureGPA : std::vector<double>
# lecturePF : std::vector<int>
# lectureNum : int
# lecturePFNum : int
# totalScore : int
# totalCredit : int
# totalPFCredit : int
# totalGPA : double
# averageScore : double
# averageGPA : double
+ Student()
+ ˜Student()
+ getStudentNo() : int
+ getName() : std::string
+ getLectureName() : std::vector<std::string>
+ getLectureType() : std::vector<LectureType>
+ getLectureScore() : std::vector<int>
+ getLectureCredit() : std::vector<int>
+ getLectureGPA() : std::vector<double>
+ getLecturePF() : std::vector<int>
+ getLectureNum() : int
+ getLecturePFNum() : int
+ getTotalScore() : int
+ getTotalCredit() : int
+ getTotalPFCredit() : int
+ getTotalGPA() : double
+ getAverageScore() : double
+ getAverageGPA() : double
+ updateInfo(database : Database &) : void
+ setStudentNo(inputStudentNo : int) : void
+ setName(inputName : std::string) : void
+ setLectureName(inputLectureName : std::vector<std::string>) : void
+ setLectureType(inputLectureType : std::vector<LectureType>) : void
+ setLectureScore(inputLectureScore : std::vector<int>) : void
+ setLectureCredit(inputLectureCredit : std::vector<int>) : void
+ setLectureGPA(inputLectureGPA : std::vector<double>) : void
+ setLecturePF(inputLecturePF : std::vector<int>) : void
+ setLectureNum(inputLectureNum : int) : void
+ setLecturePFNum(inputLecturePFNum : int) : void
+ setTotalScore(inputTotalScore : int) : void
+ setTotalCredit(inputTotalCredit : int) : void
+ setTotalPFCredit(inputTotalPFCredit : int) : void
+ setTotalGPA(inputTotalGPA : double) : void
+ setAverageScore(inputAverageScore : double) : void
+ setAverageGPA(inputAverageGPA : double) : void
+ printInfo(int) : void
+ printInfo(widthStudentNo : int, widthName : int) : void
+ printStudentInfo() : void
LinkedList
- head : Node<T>*
- current : Node<T>*
- deepCopy(original : const LinkedList<T>&) : inline void
+ LinkedList()
+ LinkedList(aplist : const LinkedList<T>&)
+ ˜LinkedList()
+ insert(newNode : Node<T>*) : void
+ insert_end(newNode : Node<T>*) : void
+ getFirst() : Node<T>*
+ getNext() : inline Node<T>*
+ find(element : const T &) : bool
+ retrieve(element : T &): bool
+ replace(newElement : const T &) : bool
+ remove(node : Node<T>*) : bool
+ isEmpty() : const bool
+ makeEmpty() : void
+ size() : int
FileException
+ filename : std::string
+ mode : std::string
+ type : std::string
+ FileException(inputFilename : std::string, inputMode : std::string, inputType : std::string)
+ ˜FileException()
UserInterface
# database : Database *
# currentUser : static Account *
+ UserInterface()
+ ˜UserInterface()
+ run() : bool
+ login() : Account *
+ welcome() : void
+ searchInfo() : bool
+ searchStudent() : bool
+ searchLecture() : bool
+ sortInfo() : bool
+ sortStudent() : bool
+ sortLecture() : bool
+ addInfo() : bool
+ addStudent() : bool
+ addLecture() : bool
+ deleteInfo() : bool
+ deleteStudent() : bool
+ deleteLecture() : bool
+ modifyInfo() : bool
+ modifyStudent() : bool
+ modifyLecture() : bool
+ load() : bool
+ save() : bool
+ print() : bool
+ printStudent() : bool
+ printLecture() : bool
+ debug() : bool
+ about() : bool
+ quit() : void
+ pause() : void
Database
- studentList : LinkedList <Student>
- requiredList : LinkedList <LectureRequired>
- limitedList : LinkedList <LectureLimited>
- optionalList : LinkedList <LectureOptional>
- userList : LinkedList <User>
- adminList : LinkedList <Admin>
+ Database()
+ ˜Database()
+ getStudentListSize() : int
+ getRequiredListSize() : int
+ getLimitedListSize() : int
+ getOptionalListSize() : int
+ load() : void
+ save() : void
+ loadStudent(filename : const std::string &) : void
+ loadRequired(filename : const std::string &) : void
+ loadLimited(filename : const std::string &) : void
+ loadOptional(filename : const std::string &) : void
+ saveStudent(filename : const std::string &) : void
+ saveRequired(filename : const std::string &) : void
+ saveLimited(filename : const std::string &) : void
+ saveOptional(filename : const std::string &) : void
+ encrypt(filename : const std::string &) : void
+ encrypt_key(filename : const std::string &, keyFilename : const std::string &) : void
+ key_gen(filename : const std::string &) : void
+ timeStampToString(timeStamp : const time_t &) : std::string
+ deleteStudent(name : const std::string &) : bool
+ deleteStudent(studentNo : int) : bool
+ deleteRequired(name : const std::string &) : bool
+ deleteLimited(name : const std::string &) : bool
+ deleteOptional(name : const std::string &) : bool
+ addStudent() : void
+ addRequired(name : const std::string &, credit : int) : void
+ addLimited(name : const std::string &, credit : int) : void
+ addOptional(name : const std::string &, credit : int) : void
+ addStudentToLecture(name : const std::string &, type : Lecturetype, stu : Student &) : void
+ modifyStudent(name : const std::string &) : bool
+ modifyStudent(studentNo : int) : bool
+ modifyRequired(name : const std::string &) : bool
+ modifyLimited(name : const std::string &) : bool
+ modifyOptional(name : const std::string &) : bool
+ queryStudent(name : const std::string &, display : bool) : int
+ queryStudent(studentNo : int, display : bool) : int
+ queryLecture(name : const std::string &, display : bool) : int
+ queryRequired(name : const std::string &, display : bool) : int
+ queryLimited(name : const std::string &, display : bool) : int
+ queryOptional(name : const std::string &, display : bool) : int
+ findStudent(name : const std::string &) : Student *
+ findStudent(studentNo L int) : Student *
+ findLecture(name : const std::string &) : Lecture *
+ findRequired(name : const std::string &) : LectureRequired *
+ findLimited(name : const std::string &) : LectureLimited *
+ findOptional(name : const std::string &) : LectureOptional *
+ sortStudent(direction : int, keycol : int) : void
+ sortLecture(type : LectureType, direction : int, keycol : int) : void
+ sortStudentCustom(head : Student **, length : int, direction : int, keycol : int) : void
+ sortLectureCustom(head : Lecture **, length : int, direction : int, keycol : int) : void
+ compareStudent(a : Student *, b : Student *, direction : int, keycol : int) : double
+ compareLecture(a : Lecture *, b : Lecture *, direction : int, keycol : int) : double
+ printStudent() : void
+ printStudent(studentNo : int) : void
+ printStudent(name : const std::string &) : void
+ printLecture() : void
+ printLecture(type : LectureType) : void
+ printLecture(name : const std::string &) : void
+ calculateGPA(score : int) : double
+ updateStudent() : void
+ updateLecture() : void
+ login(username : std::string, password : std::string) : Account *
+ registerUser(username : std::string, password : std::string, permission : int) : Account *
+ loadAccount(userFilename : const std::string &, adminFilename : const std::string &) : void
+ saveAccount(userFilename : const std::string &, adminFilename : const std::string &) : void