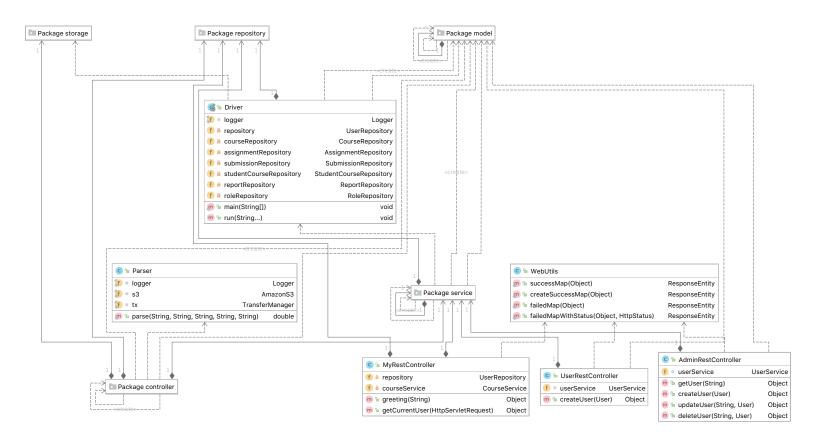
## All Packages UML

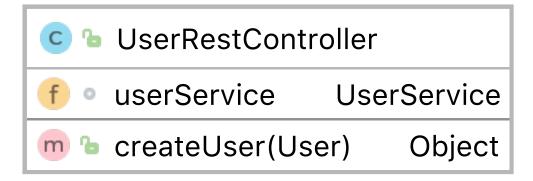


Package security

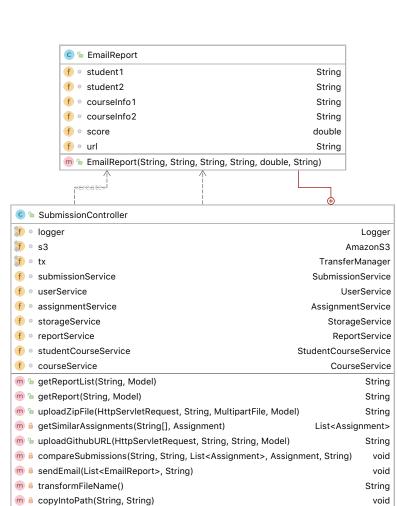
**API UML** 

C • AdminRestController	
f • userService U	serService
m = getUser(String)	Object
m 🕒 createUser(User)	Object
m 🕒 updateUser(String, User)	Object
m 🕒 deleteUser(String, User)	Object

C MyRestController	
f a repository	UserRepository
f a courseService	CourseService
m • greeting(String)	Object
m = getCurrentUser(HttpServletReg	juest) Object



#### Controller UML



void

String

String

void

© % KittyRestController	
f & repository	UserRepository
f a courseService	CourseService
m & getRelatedCourses(String, String)	String
m & sendEmailStudent(String, String, String	, String) void
m & sendFeedBack(String)	void
m & sendQuery(String)	void
m & sendMessage(String, String, String)	void
m & sendEmail(String, String, String, String)	void

m & deleteTargetDirectory(String)

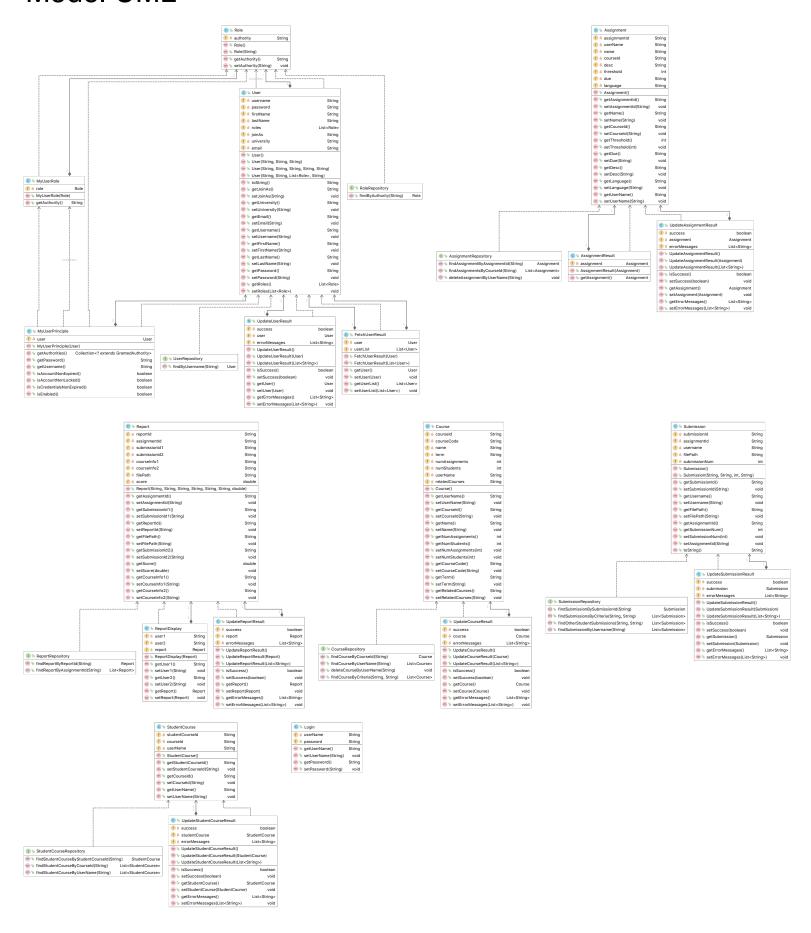
extractFolder(String)

m & getAllSubmissionList(String, Model)

m & getStudentSubmissionList(String, Model)

© % KittyController	
🤢 • logger	Logger
f o courseService	CourseService
f assignmentService As	signmentService
f • userService	UserService
f • studentCourseService Stude	entCourseService
f o terms	String[]
f o currentSemester	String
⊕ greeting(Model)	String
⊕ admin(Model)	String
⊕ adminDashboard(HttpServletRequest, Login, Model)	String
m 😘 editUser(User, Model)	String
⊕ deleteUser(User, Model)	String
	String
⊕ goCourses(HttpServletRequest, Model)	String
@ @ getLeftCourses(List <studentcourse>)</studentcourse>	List <course></course>
m a getCourses(List <studentcourse>)</studentcourse>	List <course></course>
m & editCourse(HttpServletRequest, Course, Model)	String
m 😘 deleteCourse(HttpServletRequest, Course, Model)	String
m & deleteCourseStudent(HttpServletRequest, StudentCourse, I	Model) String
m 🖫 addCourseStudent(HttpServletRequest, StudentCourse, Mo	del) String
	String
m 🖫 addCourse(HttpServletRequest, Course, Model)	String
m 😘 assignmentsPage(HttpServletRequest, String, Model)	String
m 🖫 addAssignmentsPage(HttpServletRequest, Assignment, Mo	del) String
m updateAssignmentsPage(Assignment, Model)	String
m & deleteAssignmentsPage(Assignment, Model)	String
□ addUser(User, Model)	String
	String

### Model UML



# Repository UML

■ SubmissionRepository	
m handSubmissionBySubmissionId(String)	Submission
m 庙 findSubmissionsByCriteria(String, String)	List <submission></submission>
m • findOtherStudentSubmissions(String, String)	List <submission></submission>
m findSubmissionByUsername(String)	List <submission></submission>

■ CourseRepository	
m h findCourseByCourseId(String)	Course
m findCourseByUserName(String)	List <course></course>
m b deleteCourseByUserName(String)	void
m 🔓 findCourseByCriteria(String, String)	List <course></course>

■ StudentCourseRepository	
m findStudentCourseByStudentCourseId(Str	ing) StudentCourse
m & findStudentCourseByCourseId(String)	List <studentcourse></studentcourse>
m findStudentCourseByUserName(String)	List <studentcourse></studentcourse>

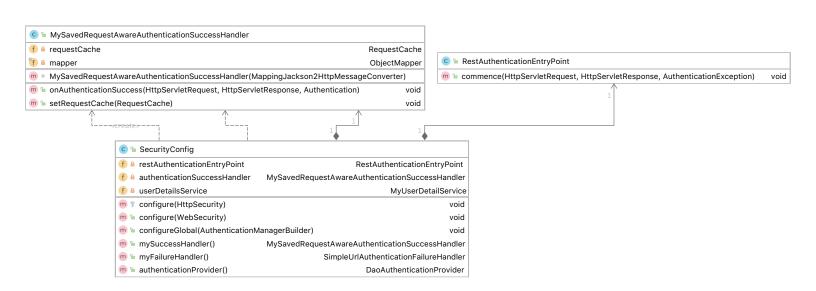
AssignmentRepository	
m • findAssignmentByAssignmentId(String)	Assignment
m • findAssignmentsByCourseld(String)	List <assignment></assignment>
m b deleteAssignmentByUserName(String)	void

■ ReportRepository	
m h findReportByReportId(String)	Report
m • findReportByAssignmentId(String)	List <report></report>



RoleRepositoryfindByAuthority(String)Role

## Security UML



# Service UML

© b CourseService	
f o courseRepository Co	urseRepository
m 'a createCourse(Course) Upda	teCourseResult
m & deleteCourse(String)	void
m & deleteCourseByUserName(String)	void
m 🖥 findCourseByCriteria(String, String)	List <course></course>
m = getCourseByCourseId(String)	Course
m b getAllCourses()	List <course></course>
m = getAllCourses(String)	List <course></course>
m & validateCourse(Course)	List <string></string>
m & checkNull(List <string>, String, String)</string>	void

© 🖫 SubmissionService	
f • submissionRepository	SubmissionRepository
m 🖥 createSubmission(Submission)	UpdateSubmissionResult
m b getSubmissionByld(String)	Submission
m 🖥 addFileToSubmission(String, Submission)	${\bf Update Submission Result}$
m b getSubmissionsForAssignment(String, String	) List <submission></submission>
m b getOtherStudentSubmissions(String, String)	List <submission></submission>
m b getSubmissionByUsername(String)	List <submission></submission>
m & validateSubmission(Submission)	List <string></string>
m & checkNull(List <string>, String, String)</string>	void

C & AssignmentService	
f • assignmentRepository	AssignmentRepository
	Logger
m 🖥 createAssignment(Assignment)	UpdateAssignmentResult
m 🖥 getAssignmentById(String)	Assignment
m b getAssignmentsForCourse(String)	List <assignment></assignment>
m 🖥 deleteAssignment(String)	void
m 🖥 deleteAssignmentByUserName(Stri	ing) void
m & validateAssignment(Assignment)	List <string></string>
m & checkNull(List <string>, String, Stri</string>	ng) void

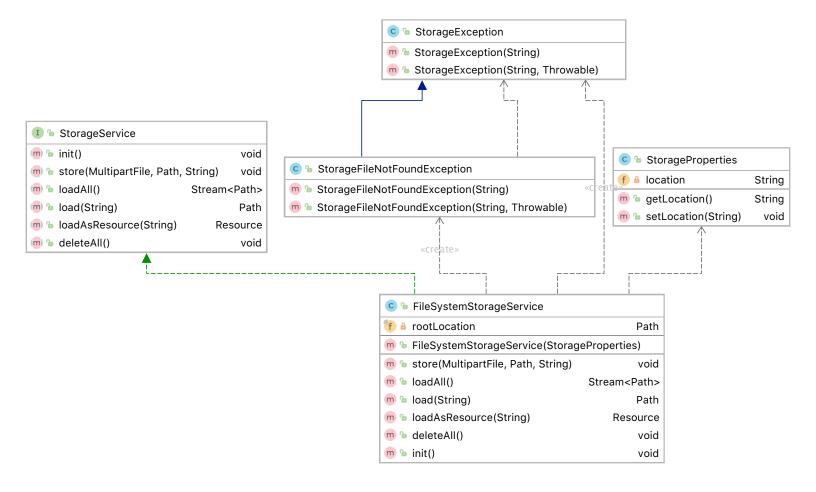
C • UserService	
f • userRepository	UserRepository
f • roleRepository	RoleRepository
m 🖥 createUser(User)	UpdateUserResult
m 🖥 deleteUser(String)	void
m b getUserByUsername(String)	FetchUserResult
m 🖥 getAllUsers()	List <user></user>

•	□ StudentCourseService	
•	studentCourseRepository	StudentCourseRepository
r	n 🖫 createStudentCourse(Studen	tCourse) void
r	deleteStudentCourse(String)	void
r	n 🖫 getCourseByCourseld(String)	List <studentcourse></studentcourse>
r	🥦 🍗 getAllCourses(String)	List <studentcourse></studentcourse>

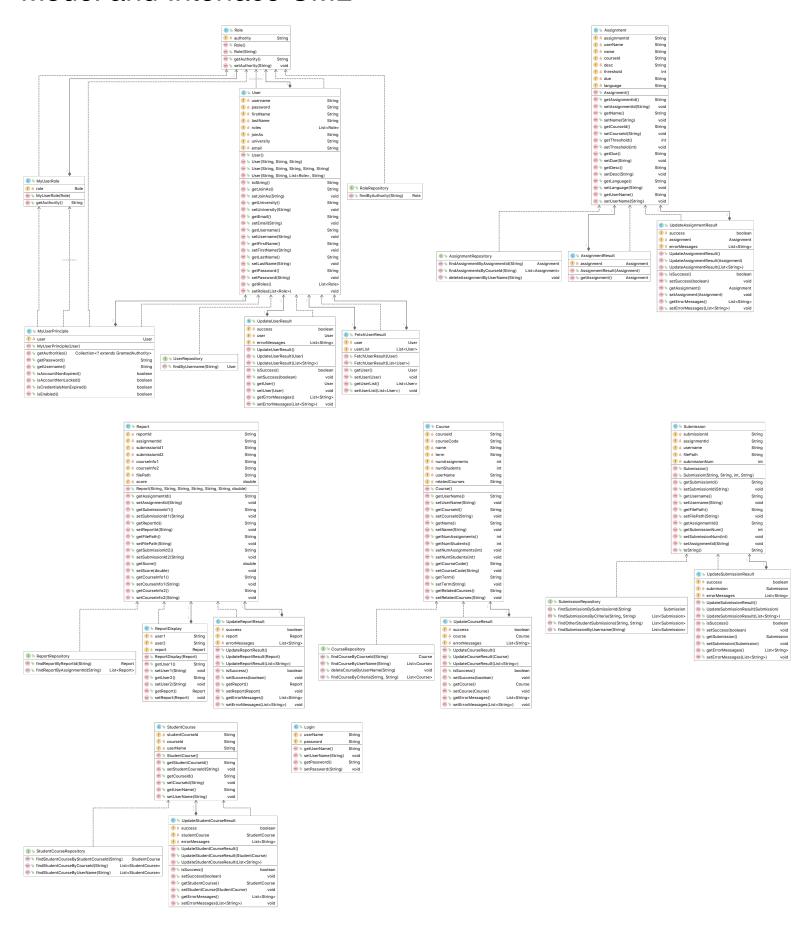
© • ReportService		
f • reportRepository	ReportRepositor	у
m 🖆 createReport(Report)	UpdateReportResu	lt
m = getReportsForAssignment(St	tring) List <report< th=""><th>&gt;</th></report<>	>
m • getReportByld(String)	Repor	t

© ७ MyUserDetailService	
f & userRepository	UserRepository
m 🖥 loadUserByUsername(String)	UserDetails

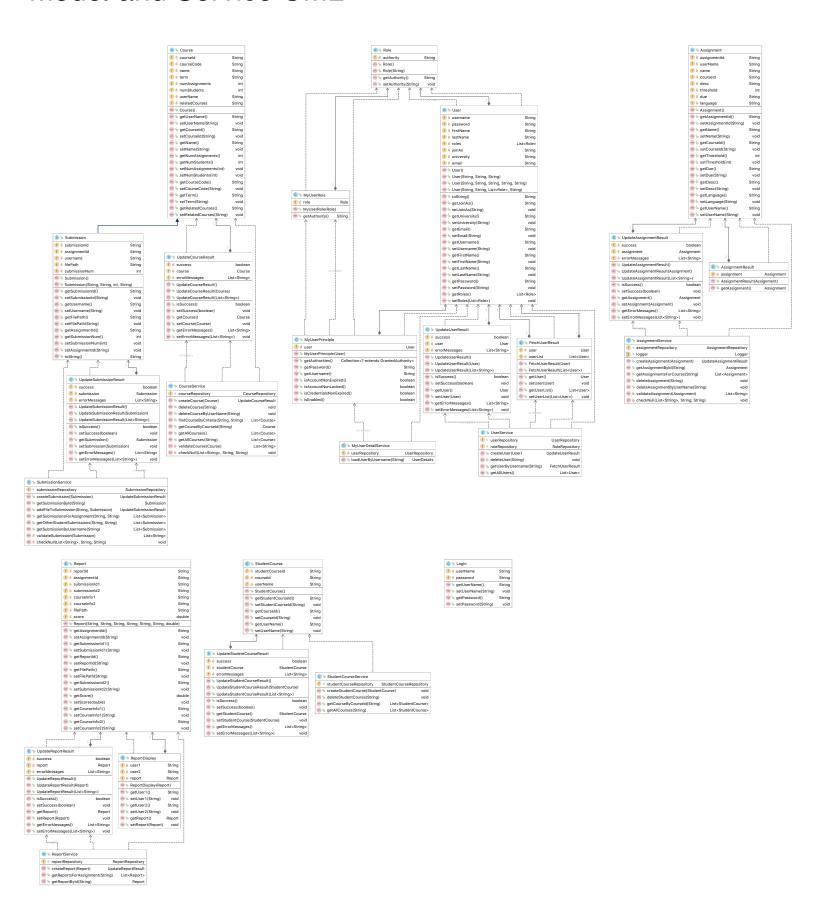
#### Storage UML



#### Model and Interface UML



# Model and Service UML



# Service API Controller UML

