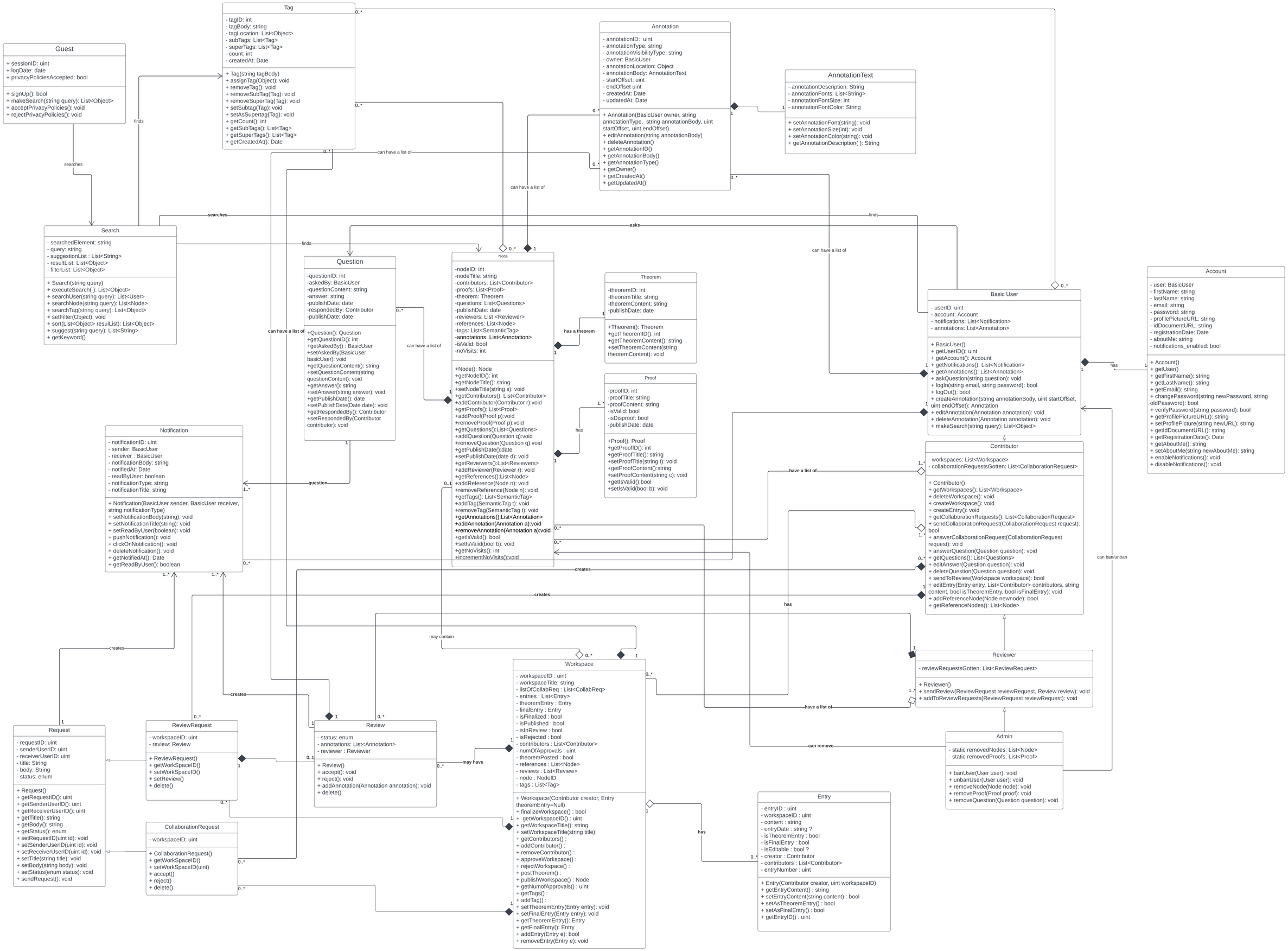
Tag
- tagID: int
- tagBody: string
- tagLocation: List<Object>
- subTags: List<Tag>
- superTags: List<Tag>
- count: int
- createdAt: Date
+ Tag(string tagBody)
+ assignTag(Object): void
+ removeTag(): void
+ removeSubTag(Tag): void
+ removeSuperTag(Tag): void
+ setSubtag(Tag): void
+ setAsSupertag(Tag): void
+ getCount(): int
+ getSubTags(): List<Tag>
+ getSuperTags(): List<Tag>
+ getCreatedAt(): Date
Guest
+ sessionID: uint
+ logDate: date
+ privacyPoliciesAccepted: bool
+ signUp(): bool
+ makeSearch(string query): List<Object>
+ acceptPrivacyPolicies(): void
+ rejectPrivacyPolicies(): void
Annotation
- annotationID: uint
- annotationType: string
- annotationVisibilityType: string
- owner: BasicUser
- annotationLocation: Object
- annotationBody: AnnotationText
- startOffset: uint
- endOffset uint
- createdAt: Date
- updatedAt: Date
+ Annotation(BasicUser owner, string annotationType, string annotationBody, uint startOffset, uint endOffset)
+ editAnnotation(string annotationBody)
+ deleteAnnotation()
+ getAnnotationID()
+ getAnnotationBody()
+ getAnnotationType()
+ getOwner()
+ getCreatedAt()
+ getUpdatedAt()
AnnotationText
- annotationDescription: String
- annotationFonts: List<String>
- annotationFontSize: int
- annotationFontColor: String
+ setAnnotationFont(string): void
+ setAnnotationSize(int): void
+ setAnnotationColor(string): void
+ getAnnotationDescription( ): String
Search
- searchedElement: string
- query: string
- suggestionList : List<String>
- resultList: List<Object>
- filterList: List<Object>
+ Search(string query)
+ executeSearch( ): List<Object>
+ searchUser(string query): List<User>
+ searchNode(string query): List<Node>
+ searchTag(string query): List<Object>
+ setFilter(Object): void
+ sort(List<Object> resulList): List<Object>
+ suggest(string query): List<String>
+ getKeyword()
Question
-questionID: int
-askedBy: BasicUser
-questionContent: string
-answer: string
-publishDate: date
-respondedBy: Contributor
-publishDate: date
+Question(): Question
+getQuestionID(): int
+getAskedBy() : BasicUser
+setAskedBy(BasicUser basicUser): void
+getQuestionContent(): string
+setQuestionContent(string questionContent): void
+getAnswer(): string
+setAnswer(string answer): void
+getPublishDate(): date
+setPublishDate(Date date): void
+getRespondedBy(): Contributor
+setRespondedBy(Contributor contributor): void
Node
-nodeID: int
-nodeTitle: string
-contributors: List<Contributor>
-proofs: List<Proof>
-theorem: Theorem
-questions: List<Questions>
-publishDate: date
-reviewers: List <Reviewer>
-references: List<Node>
-tags: List<SemanticTag>
-annotations: List<Annotation>
-isValid: bool
-noVisits: int
+Node(): Node
+getNodeID(): int
+getNodeTitle(): string
+setNodeTitle(string s): void
+getContributors(): List<Contributor>
+addContributor(Contributor r):void
+getProofs(): List<Proof>
+addProof(Proof p):void
+removeProof(Proof p):void
+getQuestions():List<Questions>
+addQuestion(Question q):void
+removeQuestion(Question q):void
+getPublishDate():date
+setPublishDate(date d): void
+getReviewers():List<Reviewers>
+addReviewer(Reviewer r): void
+getReferences():List<Node>
+addReference(Node n): void
+removeReference(Node n): void
+getTags(): List<SemanticTag>
+addTag(SemanticTag t): void
+removeTag(SemanticTag t): void
+getAnnotations():List<Annotation>
+addAnnotation(Annotation a):void
+removeAnnotation(Annotation a):void
+getIsValid(): bool
+setIsValid(bool b): void
+getNoVisits(): int
+incrementNoVisits():void
Theorem
-theoremID: int
-theoremTitle: string
-theoremContent: string
-publishDate: date
+Theorem(): Theorem
+getTheoremID(): int
+getTheoremContent(): string
+setTheoremContent(string theoremContent): void
Proof
-proofID: int
-proofTitle: string
-proofContent: string
-isValid: bool
-isDisproof: bool
-publishDate: date
+Proof(): Proof
+getProofID(): int
+getProofTitle(): string
+setProofTitle(string t): void
+getProofContent():string
+setProofContent(string c): void
+getIsValid():bool
+setIsValid(bool b): void
Basic User
- userID: uint
- account: Account
- notifications: List<Notification>
- annotations: List<Annotation>
+ BasicUser()
+ getUserID(): uint
+ getAccount(): Account
+ getNotifications(): List<Notification>
+ getAnnotations(): List<Annotation>
+ askQuestion(string question): void
+ logIn(string email, string password): bool
+ logOut(): bool
+ createAnnotation(string annotationBody, uint startOffset, uint endOffset): Annotation
+ editAnnotation(Annotation annotation): void
+ deleteAnnotation(Annotation annotation): void
+ makeSearch(string query): List<Object>
Account
- user: BasicUser
- firstName: string
- lastName: string
- email: string
- password: string
- profilePictureURL: string
- idDocumentURL: string
- registrationDate: Date
- aboutMe: string
- notifications_enabled: bool
+ Account()
+ getUser()
+ getFirstName(): string
+ getLastName(): string
+ getEmail(): string
+ changePassword(string newPassword, string oldPassword): bool
+ verifyPassword(string password): bool
+ getProfilePictureURL(): string
+ setProfilePicture(string newURL): string
+ getIdDocumentURL(): string
+ getRegistrationDate(): Date
+ getAboutMe(): string
+ setAboutMe(string newAboutMe): string
+ enableNotifications(): void
+ disableNotifications(): void
Notification
- notificationID: uint
- sender: BasicUser
- receiver : BasicUser
- notificationBody: string
- notifiedAt: Date
- readByUser: boolean
- notificationType: string
- notificationTitle: string
+ Notification(BasicUser sender, BasicUser receiver, string notificationType)
+ setNotificationBody(string): void
+ setNotificationTitle(string): void
+ setReadByUser(boolean): void
+ pushNotification(): void
+ clickOnNotification(): void
+ deleteNotification(): void
+ getNotifiedAt(): Date
+ getReadByUser(): boolean
Contributor
- workspaces: List<Workspace>
- collaborationRequestsGotten: List<CollaborationRequest>
+ Contributor()
+ getWorkspaces(): List<Workspace>
+ deleteWorkspace(): void
+ createWorkspace(): void
+ createEntry(): void
+ getCollaborationRequests(): List<CollaborationRequest>
+ sendCollaborationRequest(CollaborationRequest request): bool
+ answerCollaborationRequest(CollaborationRequest request): void
+ answerQuestion(Question question): void
+ getQuestions(): List<Questions>
+ editAnswer(Question question): void
+ deleteQuestion(Question question): void
+ sendToReview(Workspace workspace): bool
+ editEntry(Entry entry, List<Contributor> contributors, string content, bool isTheoremEntry, bool isFinalEntry): void
+ addReferenceNode(Node newnode): bool
+ getReferenceNodes(): List<Node>
Reviewer
- reviewRequestsGotten: List<ReviewRequest>
+ Reviewer()
+ sendReview(ReviewRequest reviewRequest, Review review): void
+ addToReviewRequests(ReviewRequest reviewRequest): void
Admin
- static removedNodes: List<Node>
- static removedProofs: List<Proof>
+ banUser(User user): void
+ unbanUser(User user): void
+ removeNode(Node node): void
+ removeProof(Proof proof): void
+ removeQuestion(Question question): void
Request
- requestID: uint
- senderUserID: uint
- receiverUserID: uint
- title: String
- body: String
- status: enum
+ Request()
+ getRequestID(): uint
+ getSenderUserID(): uint
+ getReceiverUserID(): uint
+ getTitle(): string
+ getBody(): string
+ getStatus(): enum
+ setRequestID(uint id): void
+ setSenderUserID(uint id): void
+ setReceiverUserID(uint id): void
+ setTitle(string title): void
+ setBody(string body): void
+ setStatus(enum status): void
+ sendRequest(): void
ReviewRequest
- workspaceID: uint
- review: Review
+ ReviewRequest()
+ getWorkSpaceID()
+ setWorkSpaceID()
+ setReview()
+ delete()
CollaborationRequest
- workspaceID: uint
+ CollaborationRequest()
+ getWorkSpaceID()
+ setWorkSpaceID(uint)
+ accept()
+ reject()
+ delete()
Review
- status: enum
- annotations: List<Annotation>
- reviewer : Reviewer
+ Review()
+ accept(): void
+ reject(): void
+ addAnnotation(Annotation annotation): void
+ delete()
Workspace
- workspaceID : uint
- workspaceTitle: string
- listOfCollabReq : List<CollabReq>
- entries : List<Entry>
- theoremEntry : Entry
- finalEntry : Entry
- isFinalized : bool
- isPublished : bool
- isInReview : bool
- isRejected : bool
- contributors : List<Contributor>
- numOfApprovals : uint
- theoremPosted : bool
- references : List<Node>
- reviews : List<Review>
- node : NodeID
- tags : List<Tag>
+ Workspace(Contributor creator, Entry theoremEntry=Null)
+ finalizeWorkspace() : bool
+ getWorkspaceID() : uint
+ getWorkspaceTitle(): string
+ setWorkspaceTitle(string title):
+ getContributors() :
+ addContributor() :
+ removeContributor() :
+ approveWorkspace() :
+ rejectWorkspace() :
+ postTheorem() :
+ publishWorkspace() : Node
+ getNumofApprovals() : uint
+ getTags() :
+ addTag() :
+ setTheoremEntry(Entry entry): void
+ setFinalEntry(Entry entry): void
+ getTheoremEntry(): Entry
+ getFinalEntry(): Entry
+ addEntry(Entry e): bool
+ removeEntry(Entry e): void
Entry
- entryID : uint
- workspaceID : uint
- content : string
- entryDate : string ?
- isTheoremEntry : bool
- isFinalEntry : bool
- isEditable : bool ?
- creator : Contributor
- contributors : List<Contributor>
- entryNumber : uint
+ Entry(Contributor creator, uint workspaceID)
+ getEntryContent() : string
+ setEntryContent(string content) : bool
+ setAsTheoremEntry() : bool
+ setAsFinalEntry() : bool
+ getEntryID() : uint
finds
searches
can have a list of
asks
has a theorem
has
question
creates
may contain
may have
have a list of
can remove
can ban/unban