

Database Schema

This project is using MongoDB - a Document-Oriented database system, which is a non-relational database.

Because MongoDB allows flexible dynamic schema, the schema might keep changing while more features are developed. We will try to keep the schema updated as quickly as possible.

For a quick understanding of how the database is structured, the diagrams below show the documents in the database and illustrate their relationships to help development.

Database Schema

The diagram below is the structure of our database design and it can be modified [here](#).

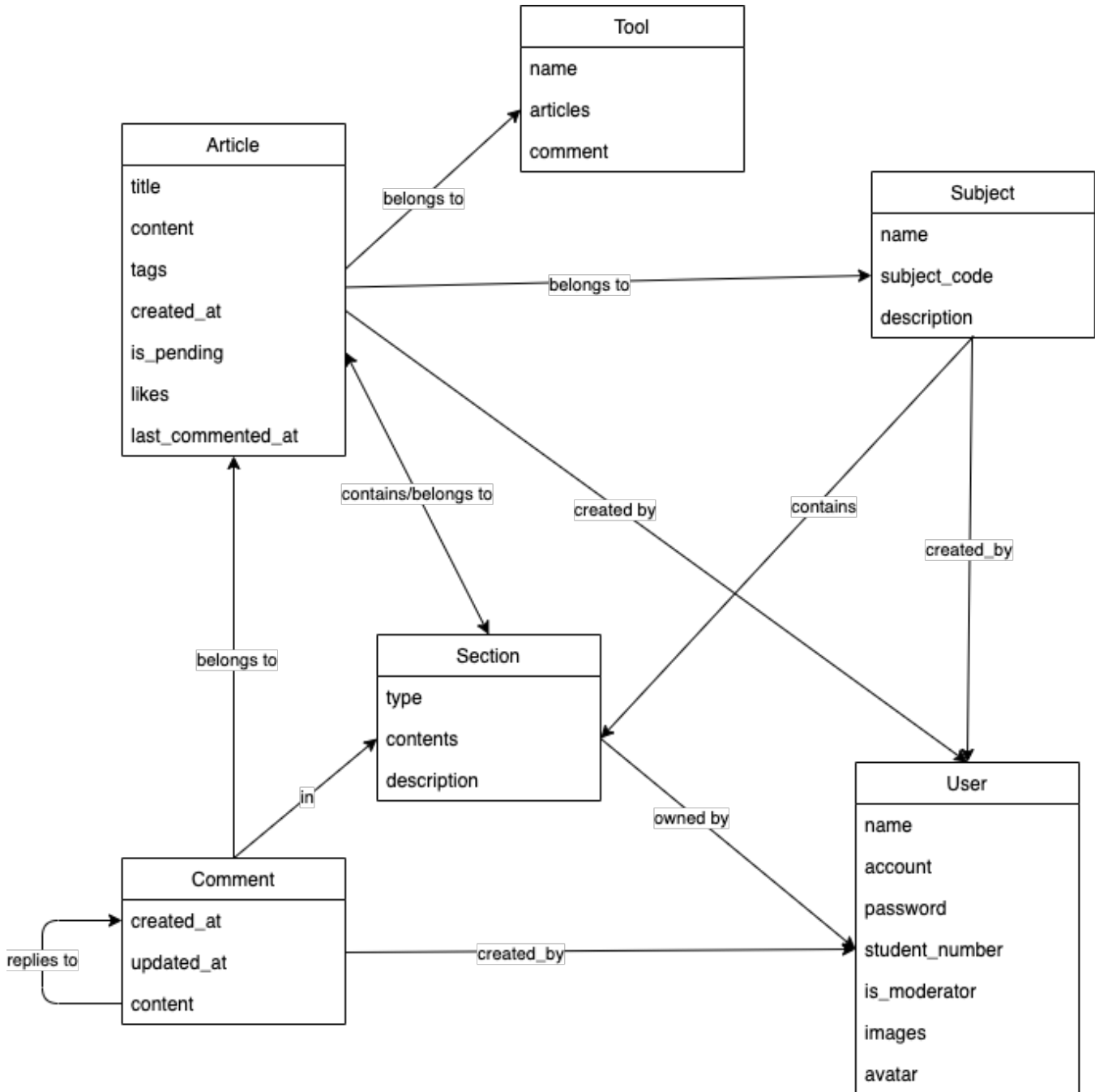
RS Database Schema

<table><tr><th>subjects</th></tr><tr><td><code>_id: Objectid, auto</code> <code>sections: array</code> <code>name: string</code> <code>subject_code: string</code> <code>description: string</code> <code>created_by: Objectid</code> <code>__v: int32, auto</code></td></tr></table>	subjects	<code>_id: Objectid, auto</code> <code>sections: array</code> <code>name: string</code> <code>subject_code: string</code> <code>description: string</code> <code>created_by: Objectid</code> <code>__v: int32, auto</code>	<table><tr><th>sections</th></tr><tr><td><code>_id: Objectid, auto</code> <code>tools: array</code> <code>articles: array</code> <code>comments: array</code> <code>name: string</code> <code>owner: Objectid</code> <code>subject_code: string</code> <code>type: string</code> <code>__v: int32, auto</code></td></tr></table>	sections	<code>_id: Objectid, auto</code> <code>tools: array</code> <code>articles: array</code> <code>comments: array</code> <code>name: string</code> <code>owner: Objectid</code> <code>subject_code: string</code> <code>type: string</code> <code>__v: int32, auto</code>	<table><tr><th>pendingarticles</th></tr><tr><td><code>_id: Objectid, auto</code> <code>subjects: array</code> <code>tools: array</code> <code>tags: array</code> <code>edited_at: date</code> <code>title: string</code> <code>editor_id: Objectid</code> <code>content: string</code> <code>__v: int32, auto</code></td></tr></table>	pendingarticles	<code>_id: Objectid, auto</code> <code>subjects: array</code> <code>tools: array</code> <code>tags: array</code> <code>edited_at: date</code> <code>title: string</code> <code>editor_id: Objectid</code> <code>content: string</code> <code>__v: int32, auto</code>	<table><tr><th>comments</th></tr><tr><td><code>_id: Objectid, auto</code> <code>leaf_comments: array</code> <code>likes: array</code> <code>author_id: Objectid</code> <code>section_id: Objectid</code> <code>reply_to_username: string</code> <code>reply_to_id: Objectid</code> <code>content: string</code> <code>create_at: date</code> <code>update_at: date</code> <code>author_name: string</code> <code>user_avatat: string</code> <code>root_comment_id: Objectid</code> <code>__v: int32, auto</code></td></tr></table>	comments	<code>_id: Objectid, auto</code> <code>leaf_comments: array</code> <code>likes: array</code> <code>author_id: Objectid</code> <code>section_id: Objectid</code> <code>reply_to_username: string</code> <code>reply_to_id: Objectid</code> <code>content: string</code> <code>create_at: date</code> <code>update_at: date</code> <code>author_name: string</code> <code>user_avatat: string</code> <code>root_comment_id: Objectid</code> <code>__v: int32, auto</code>
subjects											
<code>_id: Objectid, auto</code> <code>sections: array</code> <code>name: string</code> <code>subject_code: string</code> <code>description: string</code> <code>created_by: Objectid</code> <code>__v: int32, auto</code>											
sections											
<code>_id: Objectid, auto</code> <code>tools: array</code> <code>articles: array</code> <code>comments: array</code> <code>name: string</code> <code>owner: Objectid</code> <code>subject_code: string</code> <code>type: string</code> <code>__v: int32, auto</code>											
pendingarticles											
<code>_id: Objectid, auto</code> <code>subjects: array</code> <code>tools: array</code> <code>tags: array</code> <code>edited_at: date</code> <code>title: string</code> <code>editor_id: Objectid</code> <code>content: string</code> <code>__v: int32, auto</code>											
comments											
<code>_id: Objectid, auto</code> <code>leaf_comments: array</code> <code>likes: array</code> <code>author_id: Objectid</code> <code>section_id: Objectid</code> <code>reply_to_username: string</code> <code>reply_to_id: Objectid</code> <code>content: string</code> <code>create_at: date</code> <code>update_at: date</code> <code>author_name: string</code> <code>user_avatat: string</code> <code>root_comment_id: Objectid</code> <code>__v: int32, auto</code>											
<table><tr><th>users</th></tr><tr><td><code>_id: Objectid, auto</code> <code>is_moderator: boolean</code> <code>is_admin: boolean</code> <code>subscribed_tools: array</code> <code>subscribed_subjects: array</code> <code>moderated_subjects: array</code> <code>articles: array</code> <code>images: array</code> <code>account: string</code> <code>password: string</code> <code>name: String</code> <code>student_number: string</code> <code>__v: int32, auto</code></td></tr></table>	users	<code>_id: Objectid, auto</code> <code>is_moderator: boolean</code> <code>is_admin: boolean</code> <code>subscribed_tools: array</code> <code>subscribed_subjects: array</code> <code>moderated_subjects: array</code> <code>articles: array</code> <code>images: array</code> <code>account: string</code> <code>password: string</code> <code>name: String</code> <code>student_number: string</code> <code>__v: int32, auto</code>	<table><tr><th>articles</th></tr><tr><td><code>_id: Objectid, auto</code> <code>subjects: array</code> <code>tools: array</code> <code>tags: array</code> <code>is_pending: boolean</code> <code>likes: array</code> <code>create_at: date</code> <code>title: string</code> <code>author_id: Objectid</code> <code>content: String</code> <code>comment_section: Objectid</code> <code>__v: int32, auto</code></td></tr></table>	articles	<code>_id: Objectid, auto</code> <code>subjects: array</code> <code>tools: array</code> <code>tags: array</code> <code>is_pending: boolean</code> <code>likes: array</code> <code>create_at: date</code> <code>title: string</code> <code>author_id: Objectid</code> <code>content: String</code> <code>comment_section: Objectid</code> <code>__v: int32, auto</code>	<table><tr><th>bookmarks</th></tr><tr><td><code>_id: Objectid, auto</code> <code>articles: array</code> <code>__v: int32, auto</code></td></tr></table> <table><tr><th>tools</th></tr><tr><td><code>_id: Objectid, auto</code> <code>articles: array</code> <code>name: string</code> <code>__v: int32, auto</code></td></tr></table>	bookmarks	<code>_id: Objectid, auto</code> <code>articles: array</code> <code>__v: int32, auto</code>	tools	<code>_id: Objectid, auto</code> <code>articles: array</code> <code>name: string</code> <code>__v: int32, auto</code>	
users											
<code>_id: Objectid, auto</code> <code>is_moderator: boolean</code> <code>is_admin: boolean</code> <code>subscribed_tools: array</code> <code>subscribed_subjects: array</code> <code>moderated_subjects: array</code> <code>articles: array</code> <code>images: array</code> <code>account: string</code> <code>password: string</code> <code>name: String</code> <code>student_number: string</code> <code>__v: int32, auto</code>											
articles											
<code>_id: Objectid, auto</code> <code>subjects: array</code> <code>tools: array</code> <code>tags: array</code> <code>is_pending: boolean</code> <code>likes: array</code> <code>create_at: date</code> <code>title: string</code> <code>author_id: Objectid</code> <code>content: String</code> <code>comment_section: Objectid</code> <code>__v: int32, auto</code>											
bookmarks											
<code>_id: Objectid, auto</code> <code>articles: array</code> <code>__v: int32, auto</code>											
tools											
<code>_id: Objectid, auto</code> <code>articles: array</code> <code>name: string</code> <code>__v: int32, auto</code>											

Domain-Layer Design

This section refers to previous project and it can be found at [Back-end Design](#).

The mongoose framework allows us to treat the data in an object-oriented way. Therefore, we used the **Domain Model** pattern in our domain layer design. The domain model constructs a model of the business domain. It has the advantage of extensibility and reusability and thus matches our needs in the project. Our domain design is shown in the domain diagram below:



We categorized the objects in our system into four classes:

- **User**: A user can be a student, a moderator or a administrator. Students are normal users of the website. Moderators have the additional responsibility to manage the website (e.g. create subject, approve articles ,etc.). And administrators are able to manage the accounts of students and moderators.

- **Subject:** Each subject has its own page which contain different sections. Moderators can create and manage subject. And student can enroll into a subject.
- **Tools:** Each tool has its own page which contain different sections.
- **Article:** An article can be an article or a pending article. A pending article is an article created by a student and has not been approved by the moderator to publish to the website. An article can be related to a subject or a tool. Tags can be attached to the articles.
- **Section:** Sections can be created by moderators under a subject/tool/article page. A section may be a comment section, an article section, a tool section, etc.
- **Comment:** Comments are contained in sections. They are made by student to comment on articles/subjects/tools.