# 内部仕様書の概要

　プログラム作成に当たってどう作っていくかを設計しておくが，基本的にはDBのテーブルの内容を決めることのみ行う．今回使用するDBはMongo DBとする．

# Mongo DBの設計

　実際に作成する画面は画面設計書を参照するとし，DBに保存する内容は以下の3種類となる．

1. 部屋の情報
2. チャットの情報
3. リーフ，グルーピングリーフの情報

　これらの情報はお互いに関連しあうため，取り扱いに注意しなければならない．

## 部屋情報のDB設計（room collection）

　部屋情報についてのDBのコレクション名をroomとし，構成を表 1のようにする．

表 1 room collection

|  |  |  |
| --- | --- | --- |
| 名前 | 詳細 | 制約など |
| title | 部屋のタイトル |  |
| creator | 作成者 |  |
| detail | 詳細 | 空文字OK |
| created | 作成日 | 自動作成 |
| password | パスワード | 空文字OK |

## チャット情報のDB設計（chat collection）

　チャット情報についてはDBのコレクション名をchatとし，構成を表 2のようにする．

表 2 chat collection

|  |  |  |
| --- | --- | --- |
| 名前 | 詳細 | 制約など |
| roomId | 部屋のid  （room collectionの\_id） |  |
| name | 投稿者 |  |
| text | チャット内容 |  |
| created | 投稿日 |  |

## リーフ，グルーピングリーフのDB設計（leaf collection）

　リーフ，グルーピングリーフの情報についてはDBコレクション名をleafとし，構成を表 3のようにする．

表 3 leaf collection

|  |  |  |
| --- | --- | --- |
| 名前 | 詳細 | 制約など |
| chatId | チャットのid  （chat collectionの\_id） |  |
| title | リーフのタイトル名 |  |
| isVisible | 表示するか否か |  |
| position | xとy座標を保存  （chatboardからの相対座標を保存する） |  |
| childs | グルーピングしているleafのidをリスト形式で持つ  leafは子を持たないためnullに，groupingLeafで子が無いときは空リストにする |  |

　リーフとグルーピングリーフにおいては，再帰的な構成を取ることも可能としているため，デザインパターンCompositeを用いて実装する．