# 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. グループボックスの情報
4. 有向グラフの情報

　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 | 型 | 詳細 | 制約など |
| title | String | 部屋のタイトル |  |
| creator | String | 作成者 |  |
| detail | String | 詳細 | 空文字OK |
| created | Date | 作成日 | 自動作成 |
| password | String | 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 | 型 | 詳細 | 制約など |
| roomId | String | 部屋のid  （room collectionの\_id） |  |
| name | String | 投稿者 |  |
| text | String | チャット内容 |  |
| created | Date | 投稿日 |  |
| isVisible | Boolean | チャットボードに表示するか否か |  |
| position | Object | xとy座標を保存 |  |

## グループボックスのDB設計（groupBox collection）

　グループボックスの情報についてはDBコレクション名をgroupBoxとし，構成を表 3のようにする．

表 3 groupBox collection

|  |  |  |  |
| --- | --- | --- | --- |
| 名前 | 型 | 詳細 | 制約など |
| roomId | String | 部屋のid  （room collectionの\_id） |  |
| title | String | ラベルのタイトル名 |  |
| isExpand | Boolean | 中身を展開するか否か |  |
| position | Object | xとy座標を保存  （chatboardからの相対座標を保存する） |  |
| childs | Array | グルーピングしているlabelのidをリスト形式で持つ |  |

~~ラベルとグルーピングラベルにおいては，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．~~（使い道が浮かばないため現状は使わない方向とする）

# グラフの作成（graph collection）

　グラフの情報についてはDBコレクション名をgraphとし，構成を表 4のようにする．

表 4 graph collection

|  |  |  |  |
| --- | --- | --- | --- |
| 名前 | 型 | 詳細 | 制約など |
| roomId | String | 部屋のid  （room collectionの\_id） |  |
| isDirection | Boolean | 方向があるか否か |  |
| base | String | 起点となるlabelかgroupBoxのid |  |
| target | String | 終点となるlabelかgroupBoxのid |  |

　それぞれidはMongoDBで自動生成されるObjectIdを使用している．このidは|4byte（Unix時間）|3byte（マシンID）|2byte（プロセスID）|3byte（カウンタ）|で構成されており，他のコレクションともユニークであるためlabelとgroupBoxでidがかぶることはない．

　現時点では有向グラフのみ使用する予定だが，拡張性を考慮して無効グラフも作成しやすいような構成にしている．