

### 首页（未登录）：

填写登录名

### 首页（已登录）：

创建新故事（按钮）

展示现有的故事（自带一个房间号），

包含故事名称、作者名称、创建时间

展示图片

一段文字，介绍故事内容

### 点击图片跳转至聊天室：

上方：展示故事的图片，并可以在图片上画画

中间：展示聊天信息与进入房间提示

下方：发送聊天信息

备注：用户绘制的信息可被房间内所有人实时看见

### 创建新故事：一个表格

图片

故事名称

一段文字

作者名称（即用户名称）

创建时间（当时时间）

### 离线：

该程序必须能够访问图片，显示绘图，并允许在线和离线的绘图。

图片列表要缓存，内容也要缓存。

1. 当用户联机时，要执行的第一个操作是刷新报表数据库。
2. 操作必须是非阻塞的，即浏览器必须在图片刷新时工作，
3. 图片不能被修改或删除，与 **Mongo** 数据库同步意味着只下载自上次同步操作以来出现的新图片
4. 当上传一个新故事时，如果设备处于离线状态，该故事必须与所有元数据（标题、文本、日期等）一起缓存到 **IndexedDB** 中。
5. 当设备再次上线时，该故事必须通过 **Axios** 上传到服务器的数据库中。
6. 以前聊天的注释保存在 **IndexedDB** 中，并在请求特定故事时，甚至在离线时显示。
7. 可以在离线时进行注释，但注释不会与任何人共享。

### 其他要求：

1. 必须使用 **axios** 发送到数据库
2. 必须使用 **JSON** 进行数据交换
3. 图像要转为 **Base 64**，并被储存在 **MongoDB** 中。
4. 图像的整个信息必须保存在本地，并且在图像上绘制的线条要求本地保存。
5. 当用户返回至之前的聊天室，之前绘制的线条会被显示。但如果有新用户加入，新用户

不会看到之前的聊天信息以及绘制的线条。

6. 当选中图片中某个部分时，右侧弹出框：允许用户搜索相关信息（搜索框+结果展示）
7. 服务器端必须使用 **NodeJS + Express** 来实现
8. 必须支持聊天系统和与 **MongoDb** 的连接。
9. 服务器文档必须通过 **Swagger** 提供。

#### **允许：**

1. Bootstrap
2. Npm 库
3. Socket.io、Axio、JQuery

#### **不被允许：**

1. 不能使用 Angular、React、Vue
2. 任何构建在 Javascript 之上的语言