Stupidoc

Executive abstract

Stupidoc is a lightweight, Real-time and collaborative office software which supports Markdown and LaTeX.

In Stupidoc, Users can create a file and share it with others. The creator can modify the information of the file and the certificates of the participants. And the participants can read or join writing. When writing a document, users can see the participants' edits synchronously and click the compile to save it.

Teams

11710231	Wu Qiuqi (back-end, testing)
11712406	Yao Xinghe (front-end)
11712122	Zhang Shuyu (front-end)
11710603	Chen Linyao (back-end, database)

Description: Motivation

In SUSTech, lots of students write documents using Markdown and LaTeX. But when they collaborate to edit the one document, they often troubled by network fluctuation. So, we want to design a lightweight, Real-time and collaborative office software Stupidoc, which is running on the internal network in our school to make synchronization more stable and faster. Also, Stupidoc supports both LaTeX and Markdown to make the document management more convenient. It implements version control and lets users restore the historic version.

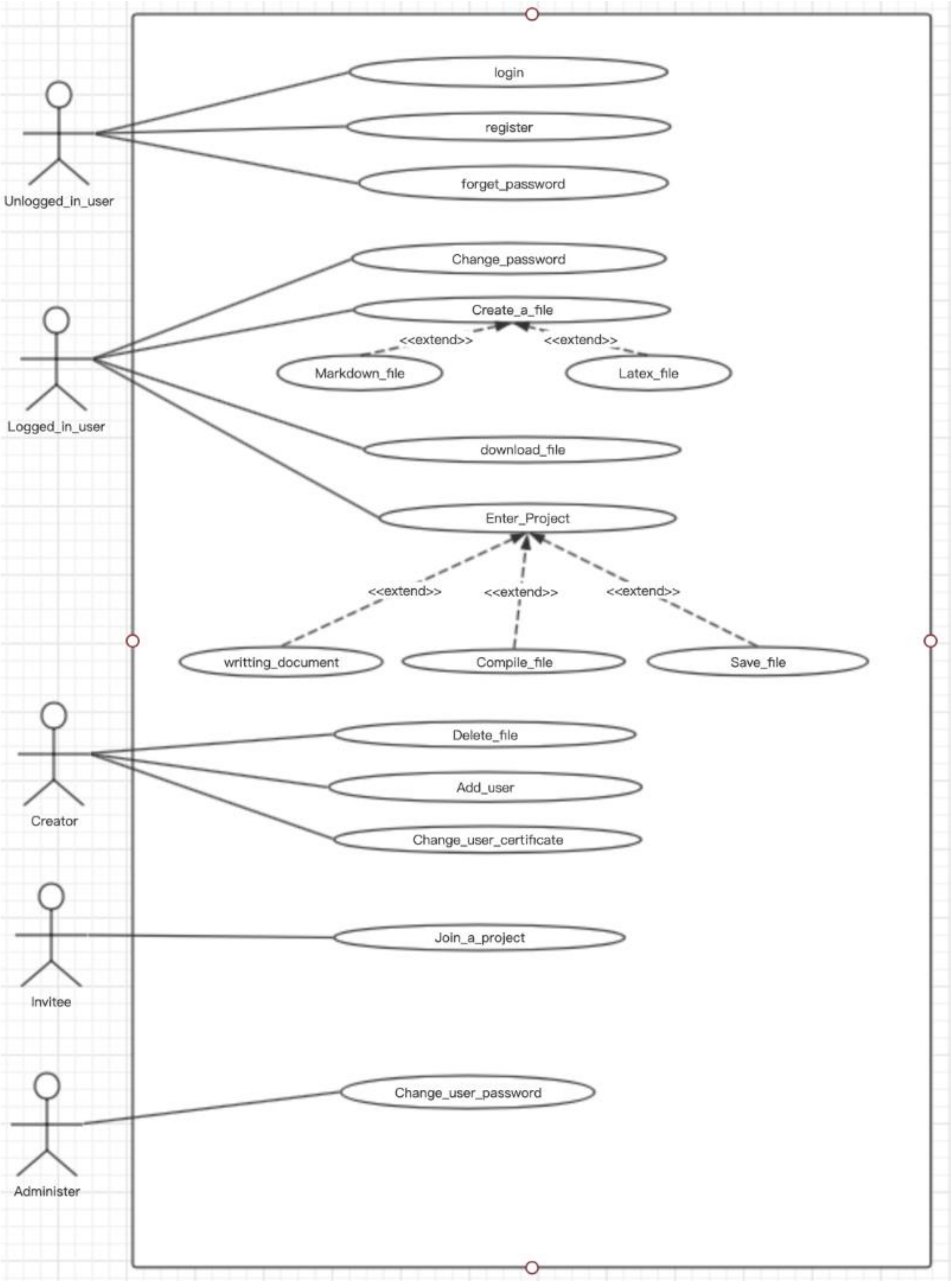
Description: Feature Description

User "stories"

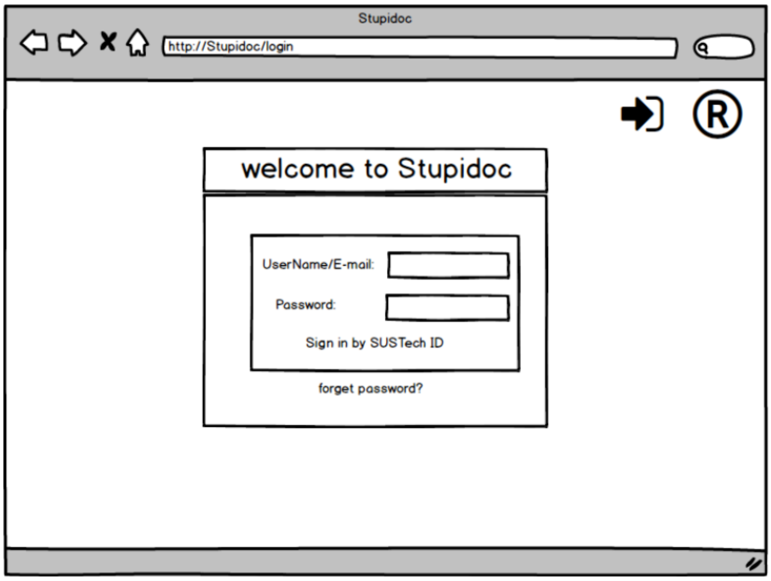
One day, Shuyu wanted to write a story about his OOAD project. So, he opened the Stupidoc. But he found he forgot his password. So, after retrieving his password and logging in. Shuyu creates a new markdown file and invites Xinghe, Qiuqi, and Linyao. Then they began to write the story. When Shuyu was going to write, he surprisingly found that Xinghe was writing "One day, Shuyu wanted to write a story ...".

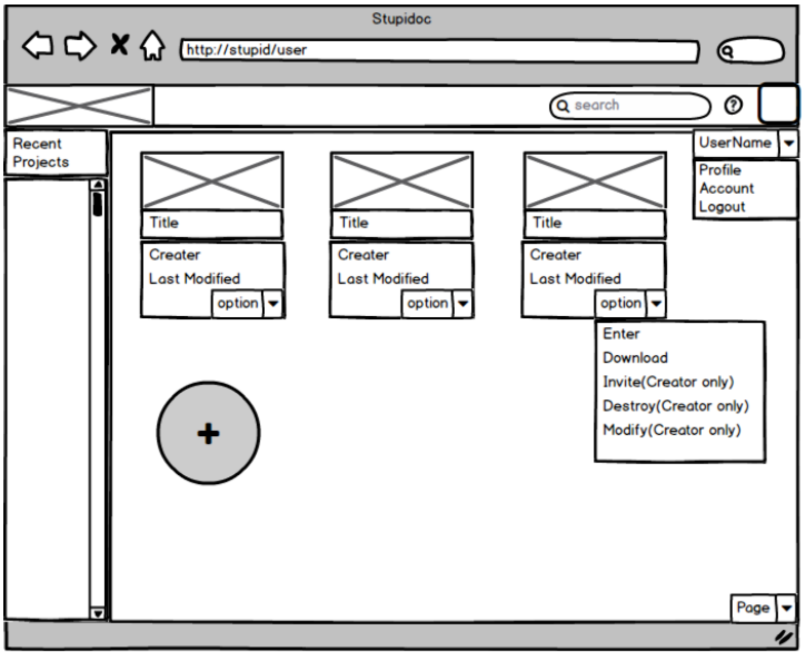
A week later, they finally completed the beginning of the story. At this time, Linyao and Xinghe quarreled with a plot in the story. Linyao directly restored to the historical version he wants. Xinghe told Shuyu, and Shuyu shut down Linyao's writing license. So Qiuqi and Linyao argued with Shuyu, but Shuyu removed Linyao and Qiuqi from the project. Finally, Shuyu and Xinghe found they could not finish it, so Shuyu deleted the file.

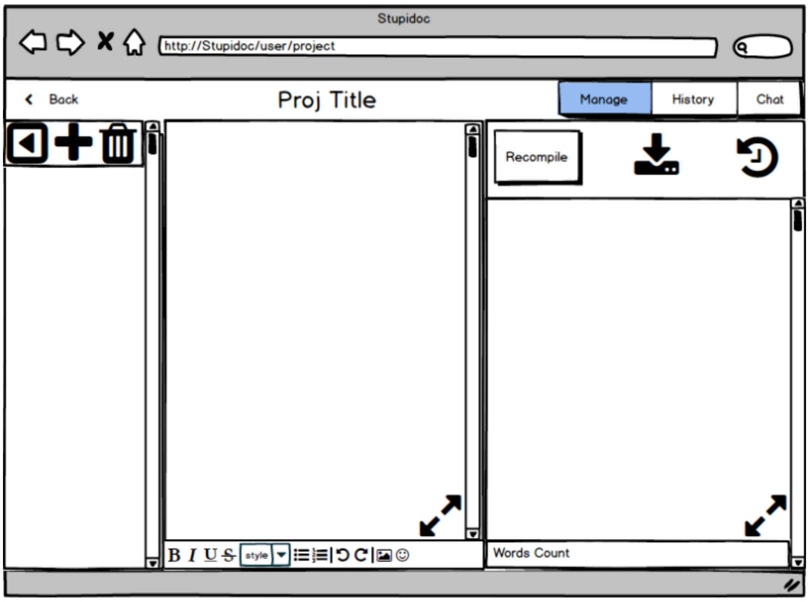
User case diagram



Mockups


Stupidoc
http://Stupidoc/login
welcome to Stupidoc
UserName/E-mail:
Password:
Sign in by SUSTech ID
forget password?


Stupidoc
http://stupid/user
search
UserName
Profile
Account
Logout
Recent Projects
Title
Creater
Last Modified
option
Title
Creater
Last Modified
option
Title
Creater
Last Modified
option
Enter
Download
Invite(Creator only)
Destroy(Creator only)
Modify(Creator only)
+
Page


Stupidoc
http://Stupidoc/user/project
Back
Proj Title
Manage
History
Chat
Recompile
B I U S style
Words Count

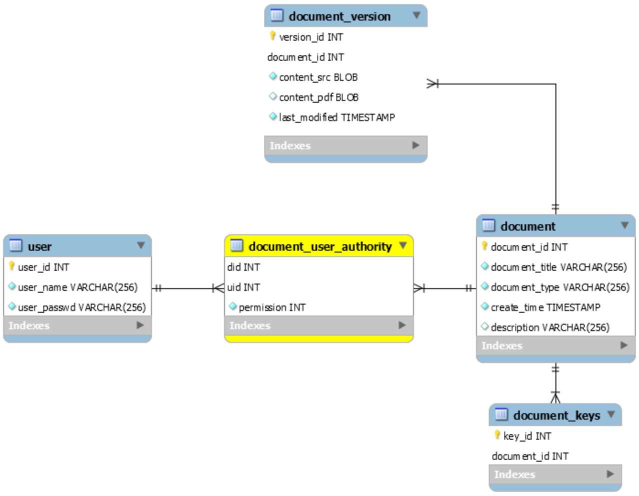
Description: Requirements

The stupidoc has some functional requirements.

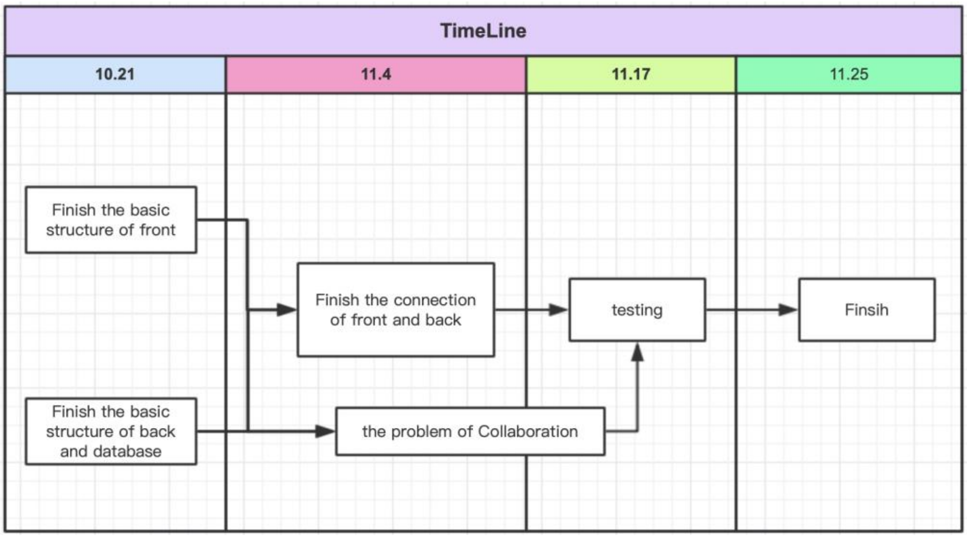
- you can register an account on the register page. You can solve it by just Using an E-mail. You can log in by the account or by SUSTech ID.
- The system can create a project by any user. In a project, the creator can invite others to join the project in two states: participant or read-only. The participant and the creator can create a new file or change the existing file. All the joiners can download the file.
- The cost of delete a user from the project is quit cheap.
- The system allows some users to work in the same file at the same time.
- For the user to use markdown, the change would be shown immediately due to the support of inner network of SUSTech.
- For the user to use latex, the system would show the last modified version. We also allow the system to go back to the history version by preserving the history version snapshot. And the user can compile his version to modify the file. While a participant is compiling, the other can break his compiling.
- Since the system save just the snapshot, each snapshot would only need some kilobytes to store it. So, the storage space would not need to much space.
- Also, to ensure the security of inviting operation, we make sure that the invite link would not be effective after being open once.
- The cost to ensure the security is also cheap since we just do it by add and delete a key to the invitation link.
- To make the communication between users more convenient, we allow the users to send a message on the edit page.

Description: Design Document

Database schema:



Timeline:


TimeLine
10.21
11.4
11.17
11.25
Finish the basic structure of front
Finish the connection of front and back
testing
Finsih
Finish the basic structure of back and database
the problem of Collaboration

Description: Feasibility

For this project, there exist some applications that already implement some functions. And most of the technologies used in these functions are basic and have some related material on Github. What's more, we have tried the front and rear connection of the document interface.

Technologies

On the server-side, to simplify the developing process, we will use log4J to record log, and Springboot to build server. As for programming languages, we use js + java. Maybe some third-part script will be invoked such as python script.

When Choosing IDE, we share a good experience on the JetBrains IDE suite, so we will use webstorm and IDEA to code this project.

On the connection method, we prefer a long-time connection, so websocket is involved.