Describe a couple more use cases

Post the facilitated Sport/Event

Name	Post the facilitated Sport/Event
Description	Students can post an event or view events
Actor	Student
Entry Condition	Login, Student access to the facilitated Sport/Event page
Exit Condition	Post an event, View the list of events
Flow of the event	1. Students can click the "event " button on the homepage after they login to open the event page 2. They can click the "post event" button to create an event 3. They will need to fill in a form for this event about the location ,time, content ,aiming group, and so on. 4. At the bottom part of this page, student can choose to add this event into a public calendar for future reminding
Exception	None
Special Requirement	None

Entity Object	- Event Information
Boundary Object	- View Button for the list of events - Events List View page - Post button to post an event
Control Object	- Post an event - Show the list of events

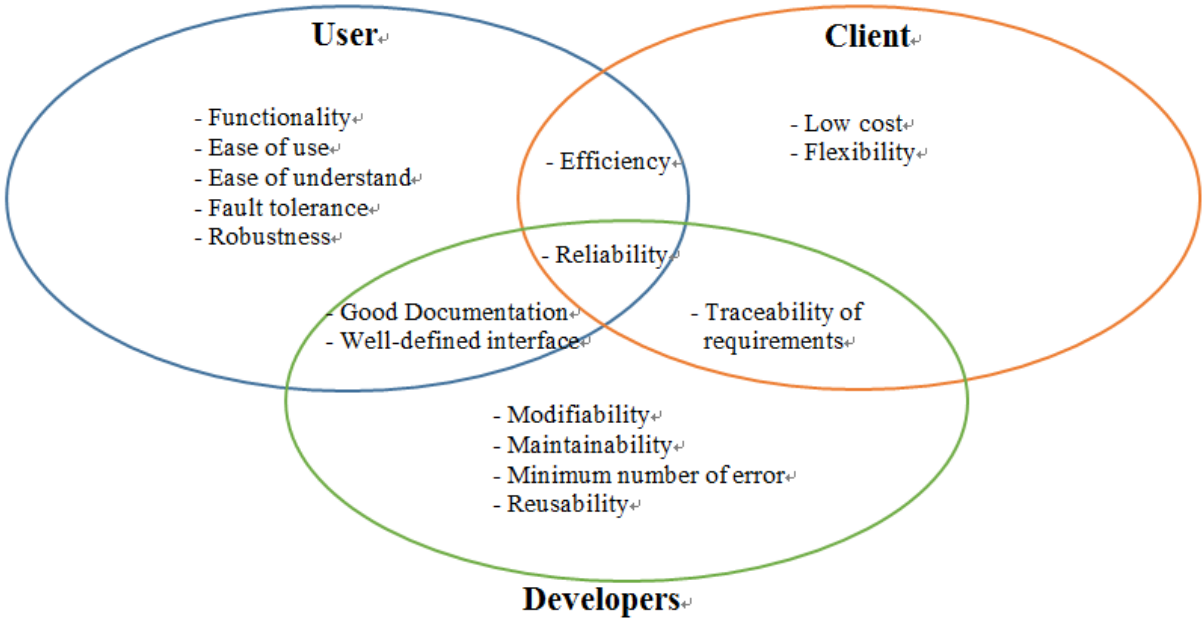
Communication Message Board

Name	Communication Messages Board
Description	Students can communicate through leaving messages on Board
Actor	Student
Entry Condition	Login, User access message board page
Exit Condition	Message Board page
Flow of the event	1. Students click "message board" button on the homepage after they login to open the message board page. 2. The messages left on this page will be organized by the order of time and the titles. 3. Students can click the titles of message to review the detail. If they are interested in the message they can directly reply. 4. Students can also choose to leave a message by click the button "leave a message" on the top of the page . 5. They will go in to the page for creating a message, here body part for the message and provide their information on information part(optional)
Exception	None
Special Requirement	None

Entity Object	- Message Board
Boundary Object	- Message Board page
Control Object	- Message Board view

Identify and address design goals

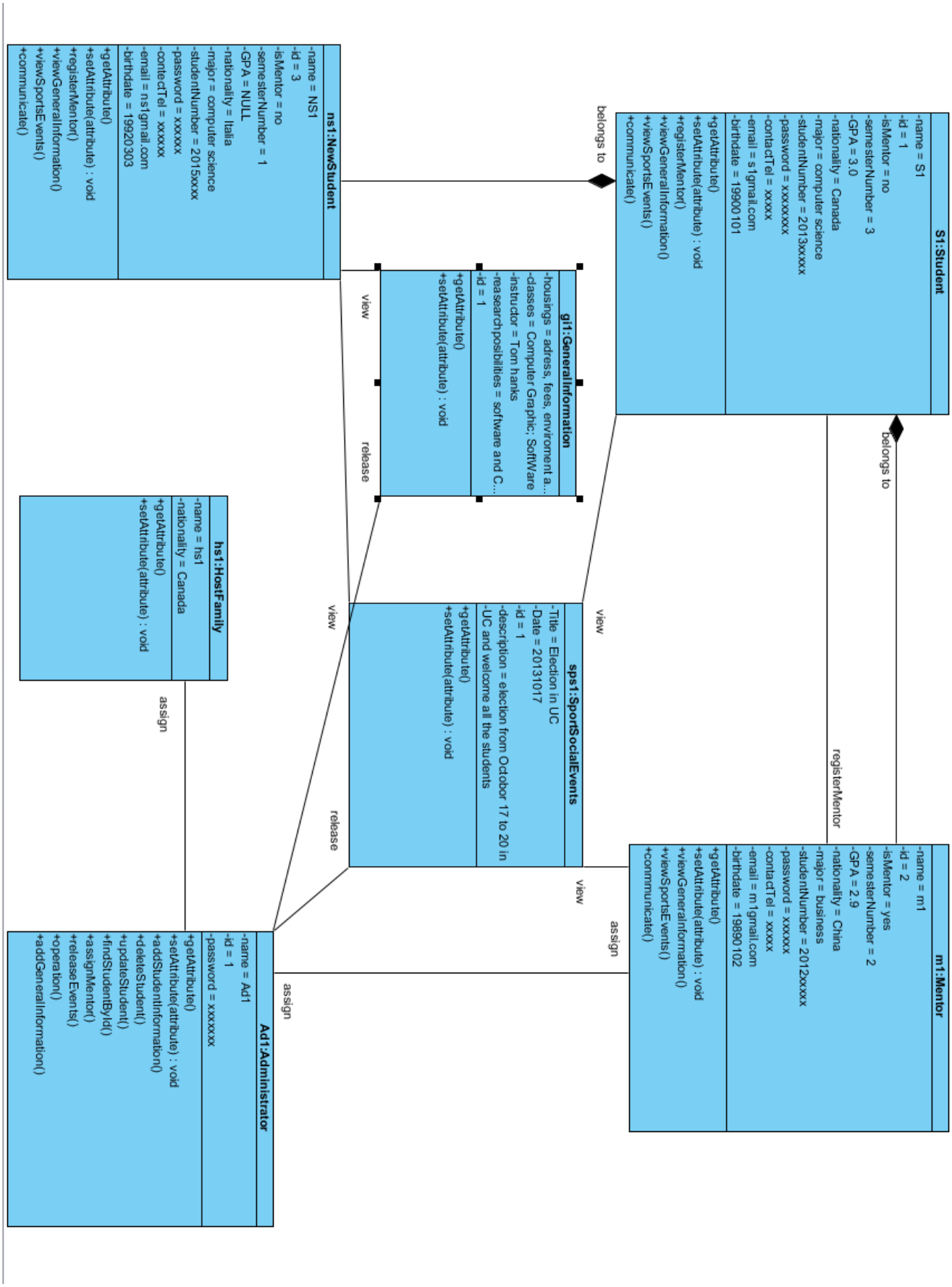
Design goals



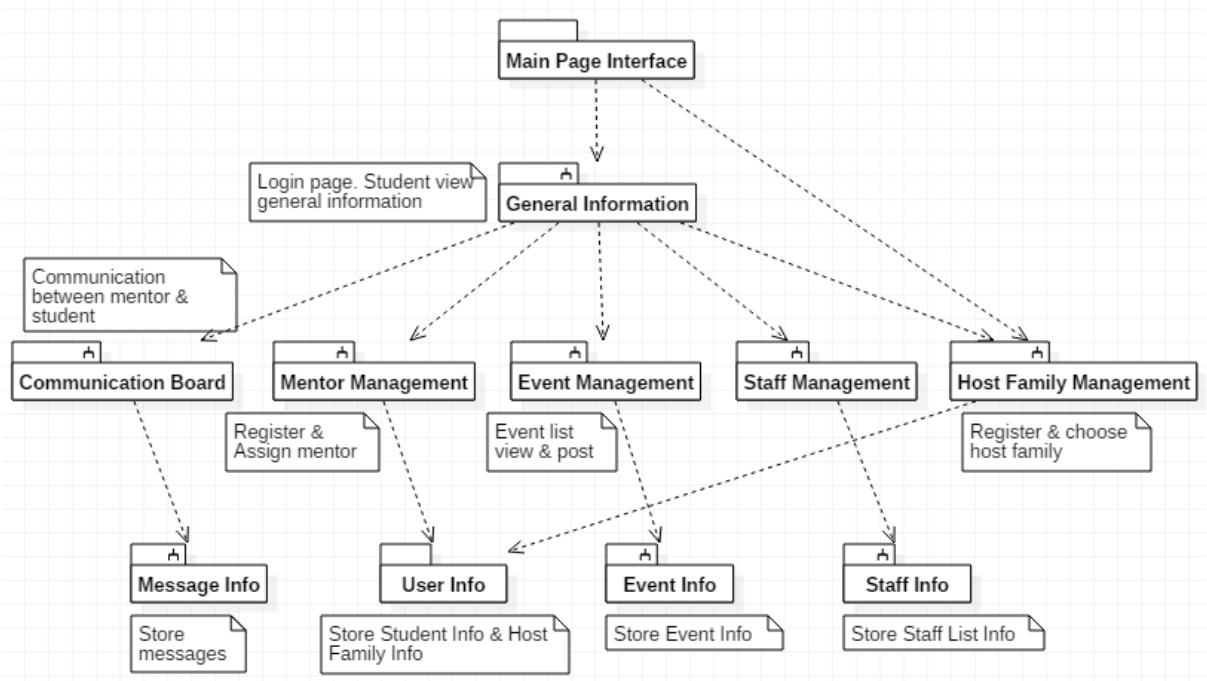
Addressing the goals

- **Functionality, Modifiability, Maintainability, Flexibility:**
  We can achieve them through a good design of the system structure including the layers and the classes, and the design patterns.
- **User-Friendliness, Ease of Use, Ease of Understand:**
  We can achieve them through a good design on the webpage or give some hint on some functions on the webpage (optional)
- **Robustness, Reliability, Efficiency, Low-cost:**
  We can achieve them through optimizing on code and multi-test on the runtime on the code, we also need to set up criteria on the response time to avoid interrupting the user's flow of thought
- **Fault Tolerance, Minimum Number of Error:**
  We can optimize the phase of system testing.
- **Good Documentation, Traceability of Requirements:**
  We can achieve them through the frequent update and backup on the rationale decisions and documenting all the original requirements.

Perform (essential) object modeling

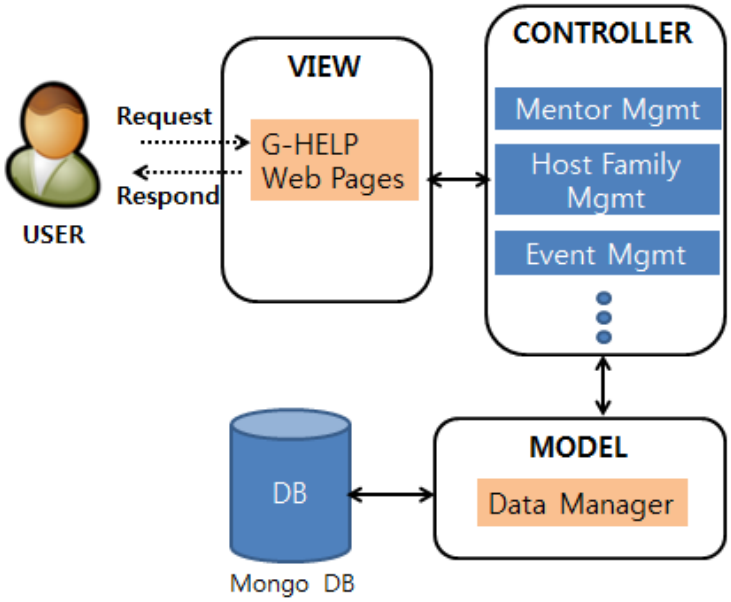


Decompose the system



Decide about the logical architecture (e.g. layering)

MVC Architectural Style



Implement a first prototype.

LINK: https://github.com/KwangsubAhn/G_Help