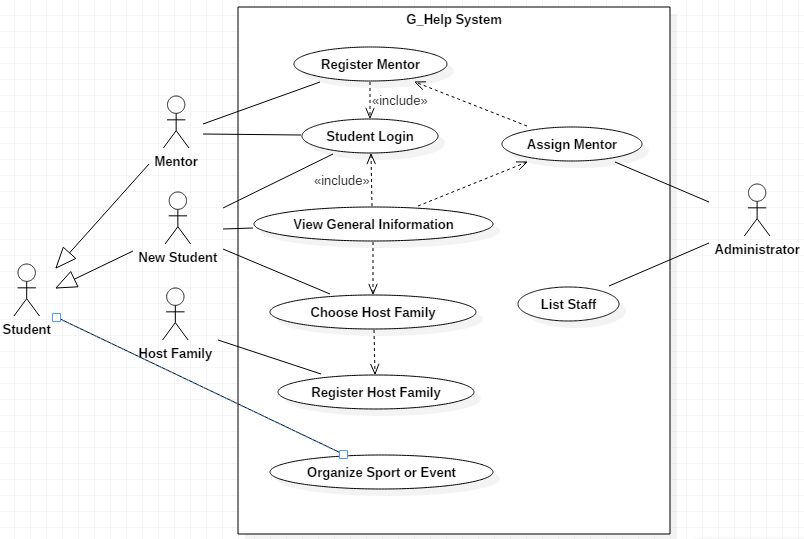
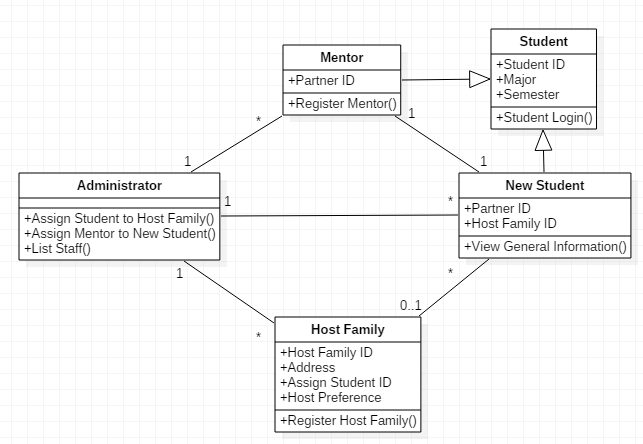
**System Description:**

Through this system, all students will be able to view their information such as personal information, academic grade and the events they like. They can choose their favorite events and put them into calendar; they can browse the information of staff of their departments and other research or any academic information to help them get familiar with school life better. Based on the criteria given by system, students reached the requirements could choose to register as mentors, and the administrator could assign those mentors to students, those mentors will also guide these new students through the whole semester. Also, the interested host family could register in system, and our system will list some host families to students by some rules, then students choose the host family, meanwhile the host family can choose to accept the students or not.

**Use Case diagram**



**Domain Modeling**



**Detail description for Use Cases**

1. Student Login

|  |  |
| --- | --- |
| Name | Student Login |
| Description | A student uses their account and password to login. The account and password will have been given from the existing system |
| Actor | New Student |
| Entry Condition | The student is a MUN student |
| Exit Condition | The student can login to the system |
| Flow of the event |  |
| Exception | The student is notified in case of the system failure |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - Student account name and password |
| Boundary Object | - Login Interface (Text boxes and buttons) |
| Control Object | - Verification |

2. View General Information

|  |  |
| --- | --- |
| Name | View General Information |
| Description | After new students login, the students can view the general information. The information will be housing, classes, instructors and research possibilities |
| Actor | New Student |
| Entry Condition | The student is a new student |
| Exit Condition | The student can view the general information |
| Flow of the event |  |
| Exception | The student is notified in case of the system failure or the student is not a new student |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - Information of housing, classes, instructors and  research possibilities |
| Boundary Object | - The general information interface page for new students |
| Control Object | - Display the general information |

3. Register Mentor **(Essential Use Case)**

|  |  |
| --- | --- |
| Name | Register Mentor |
| Description | A student apply to mentor so the student can be a mentor |
| Actor | Mentor |
| Entry Condition | Login |
| Exit Condition | A student is registered in the list of mentor |
| Flow of the event | 1. A student wants to be a mentor  2. The student click the button for register  3. Moves to the page that display guideline for mentor and \*criteria  to be a mentor  4. Click the register button  5. Verify the information if the student meet the criteria  if yes, 6-1  Otherwise, 6-2  6-1. The system add the student to the list of mentor and notice to the student that the student is successfully added to the list  6-2. Notice to the student that the student doesn’t meet the criteria |
| Exception | The student is notified if they are not satisfied with predefined criteria |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - Student account name |
| Boundary Object | - View page of guide line for mentor  - Register button |
| Control Object | - Verification  - Record the information |

\* Criteria: Students must have finished at least one year of their program

4. Assign Mentor **(Essential Use Case)**

|  |  |
| --- | --- |
| Name | Assign Mentor |
| Description | Administrator assign mentor to new student. Mentors must be registered in advance |
| Actor | Administrator |
| Entry Condition | Mentors must be registered |
| Exit Condition | New students are assigned to mentors |
| Flow of the event | 1. In certain period of time (early of each semester), the system  starts to match mentor and new student  2. Order the mentors by criteria, for example GPA, and select  mentors with only the number of new students.  3. The system match one student to one mentor for every new  students  4. The system notify them they are matched through e-Mail or can be seen from their login page |
| Exception | New students would be notified who are not assigned to a mentor. New students would not able to be assigned if the number of mentors are not enough than new students |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - Information of students such as name, program, GPA, address |
| Boundary Object | - The general information interface page for students |
| Control Object | - Select mentors by criteria  - Match students to mentors  - Record the information |

5. Register Host Family **(Essential Use Case)**

|  |  |
| --- | --- |
| Name | Register Host Family |
| Description | A family apply to host family so the family can host students |
| Actor | Host Family |
| Entry Condition | None |
| Exit Condition | A host family is registered in the list of host family |
| Flow of the event | 1. A family wants to be a host family  2. The family user click the button for application  3. The page moves to registration form page  4. The user fills out the form with \*the user information  5. Click the submit button  6. Verify the information,  if it is correctly filled, 7-1  Otherwise, 7-2  7-1. The system add the user to the list of host families and notice to the user that they are successfully added to the list  7-2. Notice to the user that their information is not proper |
| Exception | The user is alarmed if the form is not completely filled out |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - \*User provide information |
| Boundary Object | - Registration Form  - Submit Button |
| Control Object | - Verification  - Record the information |

\* User provide information: 'Date', 'Applying Semester', 'Name', 'Address', 'e-Mail', 'Phone #',

'Occupation', '\*Preference', 'Message'

\* Preference: Type of student host family is looking for. Host family list the 3 most important

Characteristics such as country, language, sport/hobbies, religion, gender etc.

6. Choose Host Family **(Essential Use Case)**

|  |  |
| --- | --- |
| Name | Choose Host Family |
| Description | A new student choose a host family from the list of host families |
| Actor | New Student |
| Entry Condition | Login |
| Exit Condition | A new student is assigned to the selected host family |
| Flow of the event | 1. A new student wants a host family  2. The student click the button for dialog of the list of host families  3. The system find the families from the list based on predefined criteria so student only can see the satisfying families  4. The student click the select button for one family  5. The system assigned the student to the selected family as a host  6. The system send an e-Mail to the family to notify the family is  select for host family  7. The host family is shown in the student’s login page |
| Exception | None |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - \* Student information |
| Boundary Object | - View Button for the list of host families  - A dialog of the list of host families  - Select Button |
| Control Object | - Assign the student to selected host family  - Show the list of host families |

\* Student information: ‘Name’, 'Major', 'Nationality', 'Language', 'Address', 'Phone #', ‘e-Mail’

7. Organize Sport/Event

|  |  |
| --- | --- |
| Name | Organize Sport/Event |
| Description | Students can post an event or view events |
| Actor | New Student, Mentor |
| Entry Condition | Login |
| Exit Condition | Post an event, View the list of events |
| Flow of the event |  |
| Exception | None |
| Special Requirement | None |

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

8. List Staff

|  |  |
| --- | --- |
| Name | List Staff |
| Description | List \*staff information so students can see from their login page |
| Actor | Student |
| Entry Condition | Login |
| Exit Condition | View the information of staff |
| Flow of the event |  |
| Exception | None |
| Special Requirement | None |

|  |  |
| --- | --- |
| Entity Object | - \*Staff information |
| Boundary Object | - View Button for staff information |
| Control Object | - Show the staff information |

\* Staff information: name, department, position, e-Mail, room and telephone