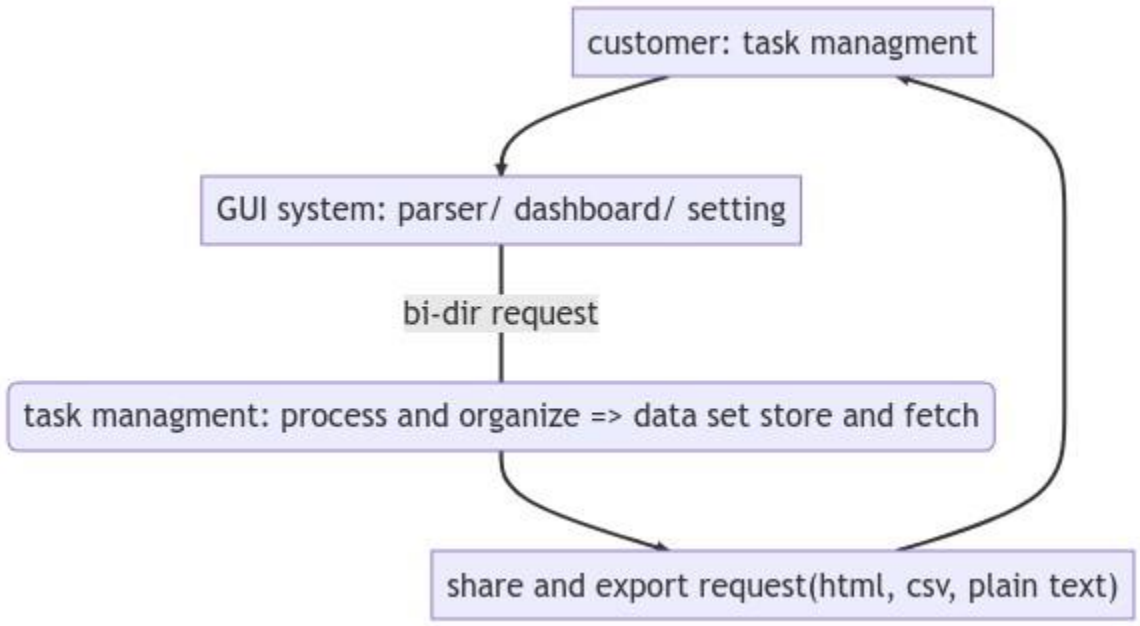
Problem statement:

As a university student, we are often overwhelmed by enormous amounts of upcoming tasks and deadlines. Therefore, it is important for us to develop a system that help university students to get more organized and get the most out of learn and achieve the target marks. More importantly, university students might want to share information between friends. The system must be capable of share the information among friends.

The major stakeholder in the system is student interact with the system.

The context of the system can be characterized as the following diagram.



https://mermaid.ink/img/pako:eNpVkDFvwjAQhf_KyRNIidgjFQmIVHXo1HaKGQ77kliN7dS-tFDEf--RsNSTdc_-7r13VSZaUpVqh_hjekwM77UOIGfXmClz9JQqYMyf4DFg5ynwEcpyC_vm-eMF8iUz-QpGTJnSBizm_hQx2Q1kYnahOy64vXyCkyutS5Doa6LMMinhsPoPF1SKhnIGDBZi6jC4X4KnraAZ71AQV4lmuSU2_XpZcJhd1Y1WWXIsOp3HKJEe-1Y9-6EAk78LGAd0AZjOvNbqYbGeCTsdVKEktkdnpZnrXdSKe_KkVSVXSy1OA2ulw02e4sTx7RKMqjhNVKhpFKdUO-wSelW1OGSZknVi-3Vpey799gdp3H5i?type=png)

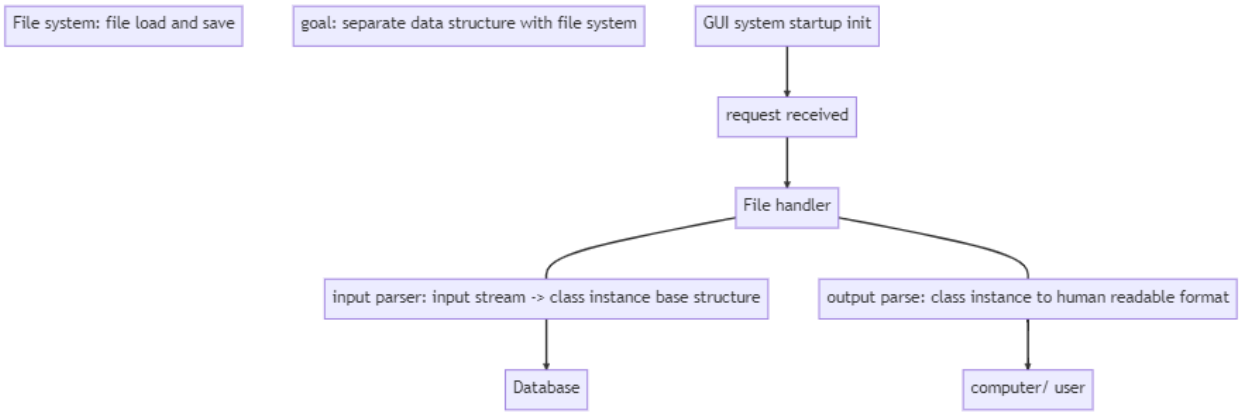
Note: the front end, GUI system ➔ task management system (request to add or fetch a value from the data base)

Task management system -> GUI system (request to update a timer event or a timeout input)

Share and export system also capable of parsing the data into a database file for next time startup

Container diagram:

File System



https://mermaid.ink/img/pako:eNpdkbtOAzEQRX9l5JqI3gUSYUlIQQVU3hQTezryY_Fj0RRlH9nNl6BhCvP49459lyFjoaEFIOLZz1iKvDZ9QH4PKuNdQT5kgt5CcMcuIgGMBjIeKJ963tVx4hOQgYJExYCgwUhl1R1qYngbMvY1M1qke3U9mu3pLibJ9cJbLBlD6vVE2xUou9KuUAiTfZEpuXXTb1ucCOzOEpzaQUvyoapFmCMTElCixiE0ANLtcOcOcvDgiY4YKY_zOb-pjqGnysL5fru3KlYy6-1_O9UIozVY2BUNHhgriEmj8tLtkpHz2JKj1DzDNsH8SA8cYs1_PXXeVQvykieeiH5amjA6kov-nDjVqwlflyCFpJp6UHUib-YOovHhF7IAV3mLBlbYnpv67xv9fYD1meimw?type=png)

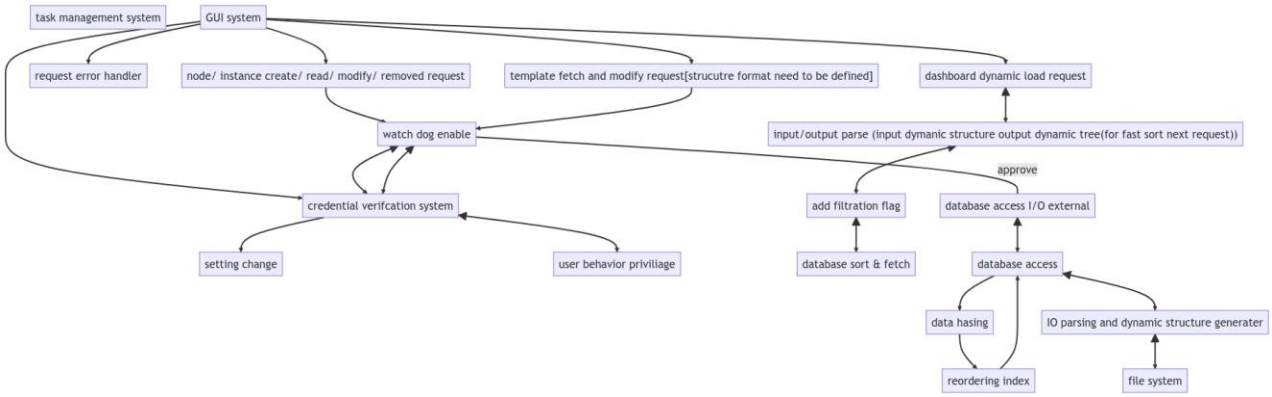
This diagram characterized the overall behavior of the file system. Turn data structure during execution phase into storage/ export file or vice versa.

Note: The Database refer to runtime advanced data structure RAM

The system needs to be able to customize the data structure being used in the system.

The system also needs to be capable of turning the runtime data structure into a readable file or a file that will be fetched at the beginning of the next time program startup.

Task management system



[https://mermaid.ink/img/pako:eNptk01v2zAMhv8KocPQAg1yD4YBW9MG2UeTfR06OQfGomNhsuRJdNqg6H8fbdkIMiw-xCapl-JD8kWVwZBaqMqFp7LGyPBjWXiQ33vNmH5Dgx4P1JBnSKfE1Oyy-4Ne_VxPJpjN3sGtNpjqfcBowJw8NrYEF9BApD8dJZ4ODsFLPVqBYgwRavTGUbyIudOFEvXWIRNUxGUNEgVNMLY6Tao6cezKjqOEhNgggycywAH2BIYq68nsCnUhfK-9VD0H6xOjLwnKSJJjLppo5mOC_qsJR_r__VdazhihYtHBkaKtSmQb_AWRWiditv4AgtYfaAdZYgVve_dn3SWKcs8aj1YYtNEerbOCe0x1m-M-Cgfr247noWP5gxZjIrgabIJaWiSoBw7cCQcYw6YmCBu6EjhQofBOIfaMnnmq6_pa8ORMnzQaA5V1HHM1lcPD6FtLexn3KJkHiTe5JeNV74eCH_QT9m0y4QDkce-mSh6yxmoEcJejL33T12wG2LZR0MvrDDbnvFiWlBKs5xuQ-1P06Eb9TdbY_hs7urdDvq-DV0YtSUtyh77JGIZopH_SJOsNPWf7tvDTyaz8qNebgXsf2E_hBPeM_UCeBJsMcX_ilxaMdN6Z_PSS6kY1JJNqjezdS2-TMa9lxQq1kFeZWewcF6rwrxKKHYfvJ1-qhSSiG9W1UgQtZUoiNmpRoUtiJWM5xC95l4eVfv0LURJEUw?type=png)

Notes: the database are consisting of two parts:

part 1. User login and privilege part 2. Data instance storage and sorting(optimization) 3. Preference

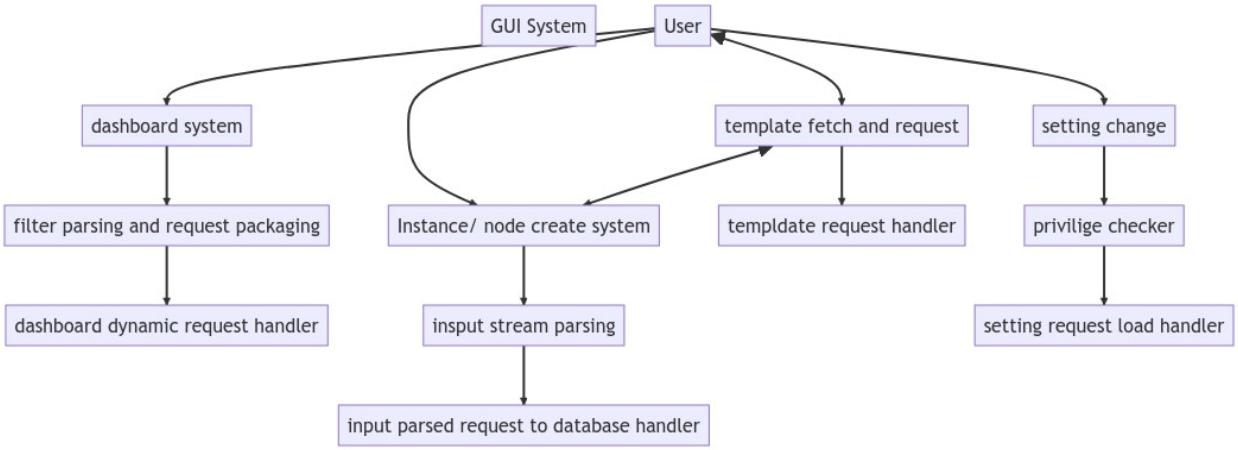
Request error handler is a handler deal with all system error situation from front end back end or from the internal system.

A credential management system is the entry point for all database modification request. (Implement watch dog for data recovery and access management)

Templates simplify the node/ instance field creating process. (Template can be saved upon request or latest store habit, provide build in template)

Every change in database will trigger a hashing (quick sort/ fetch i.e., library) update the system can be accessed only by File system or I/O access management system.

Dashboard load request able to dynamically request database set update and filtration important information for front end to use. (Granted with read-only access)



https://mermaid.ink/img/pako:eNp9UstOwzAQ_BXLZyruEUICSkt5XUpPTg9be5NYdZxgb0BV1X9nHdNShIRP1szszuzae6k7g7KQles-dQOBxNu09ILPjZqvFmK5i4TtOkO3ahUxrMYbDoIRsQsGvGpqqqwjDKKHEK2vBXgjAr4PGIkxvYWa0ay9P-thdh5aq0_Shuscm307j_qZWvhI4DVeCs-5hQ4IhL_8F8r62A8kIjHZHmNk8pHJxCUQf2JRJwwQbCDiH9-rVDdX3L93yatC0s35UOssmeWC-ejzkPUmFfw30JOKSJS2xLv3NeaUz6oP9sM6W_OEDertcefi9aQ_dnUdmFNreSFbDC1Ywy-6T0alpAZbLGXBV4MVDI5KWfoDS2GgbrnzWhYUBryQQ5_yTi3UAVpZVOAio2gsdeEl_5Lxsxy-ALLnvkM?type=png)](https://mermaid.live/edit#pako:eNp9UstOwzAQ_BXLZyruEUICSkt5XUpPTg9be5NYdZxgb0BV1X9nHdNShIRP1szszuzae6k7g7KQles-dQOBxNu09ILPjZqvFmK5i4TtOkO3ahUxrMYbDoIRsQsGvGpqqqwjDKKHEK2vBXgjAr4PGIkxvYWa0ay9P-thdh5aq0_Shuscm307j_qZWvhI4DVeCs-5hQ4IhL_8F8r62A8kIjHZHmNk8pHJxCUQf2JRJwwQbCDiH9-rVDdX3L93yatC0s35UOssmeWC-ejzkPUmFfw30JOKSJS2xLv3NeaUz6oP9sM6W_OEDertcefi9aQ_dnUdmFNreSFbDC1Ywy-6T0alpAZbLGXBV4MVDI5KWfoDS2GgbrnzWhYUBryQQ5_yTi3UAVpZVOAio2gsdeEl_5Lxsxy-ALLnvkM

notes: the front end just handler the input and forward the request to backend → no advanced operation is done on front end.

All the component in the front-end has a corresponding handler at the back end to deal with request.

application notes: front end and back end does not need to be one the same device; all the data transfer is done upon request and database is stored at back-end side.