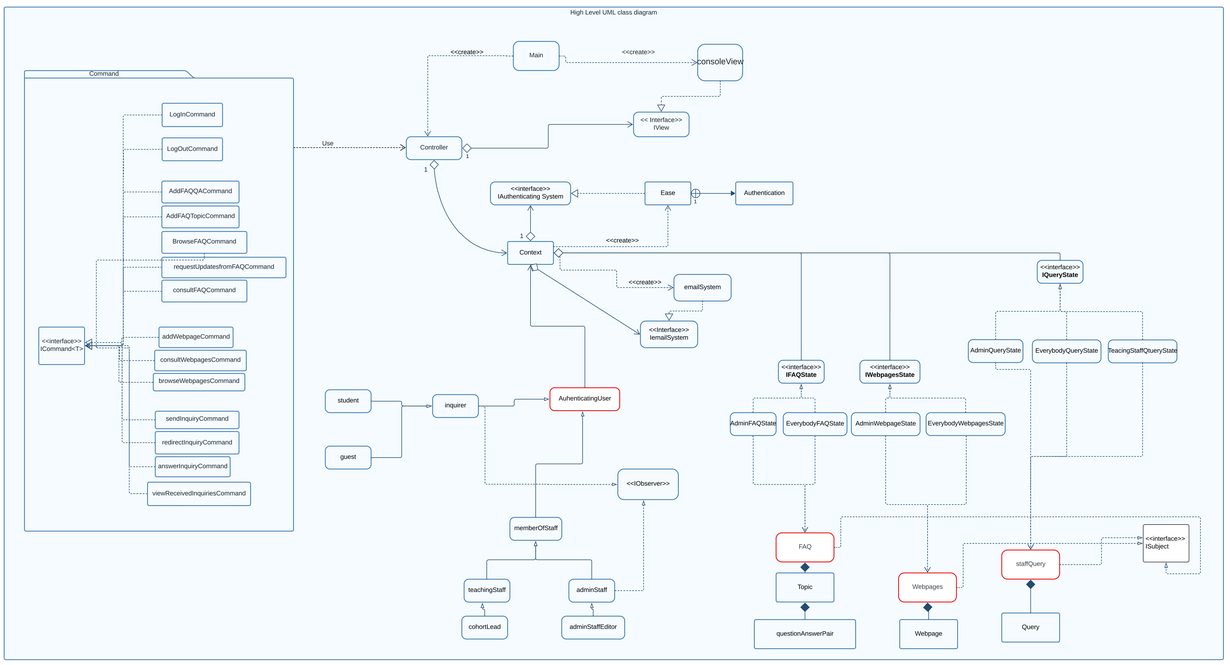

Detailed UML Class Diagram

View package

<<interface>>
IView

+ displayMainMenu(): void

consoleView

// all the main menu console prompts
+ MAIN_MENU_LOGIN: int
+ MAIN_MENU_LOGOUT: int
+ MAIN_MENU_LIST_WEBPAGES: int
+ MAIN_MENU_LIST_FAQS: int
+ MAIN_MENU_QUERY: int

// all the console prompts for the FAQ list.
+ GUEST_NAVIGATE_FAQ: int
+ STUDENT_NAVIGATE_FAQ: int
+ ADMIN_STAFF_NAVIGATE_FAQ: int
+ TEACHING_STAFF_NAVIGATE_FAQ: int
+ GUEST_PROVIDE_EMAIL_FAQ: int
+ ADMIN_STAFF_ADD_TOPIC_QA_FAQ: int
+ GUEST_PROVIDE_QA: int
+ STUDENT_PROVIDE_QA: int
+ ADMIN_STAFF_PROVIDE_QA: int
+ TEACHING_STAFF_PROVIDE_QA: int

// all the console prompts for the Webpages
+ GUEST_LUCENE_SEARCH_WEBPAGES: int
+ GUEST_VIEW_WEBPAGE: int
+ STUDENT_LUCENE_SEARCH_WEBPAGES: int
+ STUDENT_VIEW_WEBPAGE: int
+ TEACHING_STAFF_LUCENE_SEARCH_WEBPAGES: int
+ TEACHING_STAFF_VIEW_WEBPAGES: int
+ ADMIN_STAFF_LUCENE_SEARCH_WEBPAGES: int
+ ADMIN_STAFF_VIEW_WEBPAGES: int
+ ADMIN_STAFF_BROWSE_WEBPAGES: int // TAKES A NUM OR A RETURN
+ ADMIN_STAFF_ADD_WEBPAGE: int

// all the console prompts for queries
+ STUDENT_WRITE_QUERY: int
+ GUEST_WRITE_QUERY: int
+ TEACHING_STAFF_CHOOSE_QUERY: int
+ TEACHING_STAFF_WRITE_ANSWER: int
+ ADMIN_STAFF_CHOOSE_QUERY: int
+ ADMIN_STAFF_REDIRECT_QUERY: int
+ ADMIN_STAFF_WRITE_ANSWER: int

+ ConsoleView()
+ displayFAQ(topicstring: List<String>): void
+ displayQuery(): void
+ displayGetEmail(): void
+ displayWebpages(): void
+ displayMainMenu(): void
+ displaySuccessStatus(): void
+ displayFailureOutStatus(): void

Controller package

Main

+Main()

+main(String[]): void

controller

- context: Context
- view: IView

+ Controller(Context, IView)
+ changeDisplay(List<String>): void
+ confirmLoggingOut(): boolean
+ displayFAQList(topic: Topic): void
+ displayQueryList(query: Query): void
+ displayWebpagesgpage: Webpage): void
+ displaySearch(content: String): void
+ runCommand(ICommand<?>): void

Context

- user: AuthenticatingUser
- emailSystem: IemailSystem
- authenticationSystem: IauthenticatingSystem
- Qstate: IQueryState
- Fstate: IFAQState
- Wstate: IWebpageState

+ Context(): void
+ Context(username: String, password: String)
+ Authentication(): boolean

Model

Model package

HumanUsers

AuthenticatingUser

- nameOfAuthUser:String
- emailOfAuthUser: String
- roleOfAuthUser: String
- cohortLeadName: String
- cohortLeadEmail: String
- loggedIn: Boolean
- userInput: String

+ authenticatingUser(username: String, password: String): [String]
+ receiveConfirmation(loggedIn: Boolean)
+getInput(): String

inquirer

- sendTo: String
- sendQueryTo: String
- queriesMade: List<faqQuery>

+ Inquirer(authenticationUser)

memberOfstaff

teachingStaff

cohortLead

<<Interface>>
IObserver

+ update(TopicChanged)
+ update(WebpageChanged)
+ update(QuerySent)

<<enum>>
Role

Student
Guest
AdminStaff
TeachingStaff
CohortLead

guest

- TopicChanged: Topic

+ update()

student

- TopicChanged: Topic

+ update()

editorAdminStaff

adminStaff

- generalAdminStaffEmail:String
- TopicChanged: Topic
- QuerySent: Query
- WebpageChanged: Webpage

+update()

services

Webpages

Webpages

- staffObservers: List<adminStaff>
+ webpageChanged: Webpage
- publicWebpages: List<Webpage>
- privateWebpages: List<Webpage>
- fileList: List<String>
- commandObject: ICommand

+ Webpages()
+ searchWebpages(query: String): List<Webpage>
+ addWebpage(webpage: Webpage, public: boolean): boolean
+ searchWebpage(webpage: Webpage, query: String): String
+ addObserver(updateSubscriber: IObserver): void
+ removeObserver(updateUnsubscriber: IObserver): void
+notifyStaffObservers(editorStaffEmail: String);

Webpage

- fileName: String
- pageName: String
- content: String
- public: boolean

+ Webpage(file:String, name: String, content: String): void
+ fileData(): void

FAQs

FAQ

- publicTopicHead: Topic
- privateTopicHead: Topic
- commandObject: ICommand

+ FAQlist(topic: Topic, public: Boolean): List<String>
+ AddQAs(topic: Topic, qaPair: questionAnswerPair): boolean
+AddTopic(parentTopic: Topic, topicToAdd:Topic): boolean

Topic

- staffObservers: List<adminStaff>
- inquirerObservers: List<inquirer>
- topicChanged: Topic
- topicName: String
- QAPairs: List<questionAnswerPair>
- subTopics: List<Topic>
- greatParentTopic: Topic
- commandObject: ICommand

+ Topic(name: String, qaPair: questionAnswerPair): void
+ addSubTopic(subtopic: Topic): void
+ TopicString(): String
+ addObserver(updateSubscriber: IObserver): void
+ removeObserver(updateUnsubscriber: IObserver): void
+ notifyObservers(): void
+ notifyStaffObservers(editorStaffEmail: String): boolean
+ notifyStudentObservers(): boolean

questionAnswerPair

- question: String
- answer: String
- visibility: VisibilityStatus
- parentTopic: String

+ QuestionAnswerPair (String, String, VisibilityStatus)

Query

staffQuery

- Queries: List<Query>
- commandObject: ICommand

+ viewQueries(): void
+ sendQuery(senderEmail:String, queryTitle:String, queryContent:String, receiverEmail:String): boolean
+ redirectQuery(senderEmail:String, query: Query): boolean
+ sendQueryAnswer(query: Query, title: String, content: String): boolean
+ getQueryList(email: String): List<Query>

Query

- staffObservers: List<adminStaff>
+ querySent: Query
- senderEmail: String
- receiverEmail: String
- title: String
- content: String
- answered: boolean
- commandObject: ICommand

+ Query(senderEmail:String, receiverEmail: String, title:String, content: String): void
+ addObserver(): void
+ removeObserver(): void
+notifyStaffObservers(editorStaffEmail: String): boolean

<<interface>>
ISubject

+ AddObserver(ob: IObserver): void
+ RemoveObserver(ob: IObserver): void
+ notifyObservers(ob: IObserver): void

<<enum>>
contentChanged

UPDATED
UNCHANGED

<<enum>>
QueryStatus

PENDING
RESOLVED

<<enum>>
updateRequested

Activated
Deactivated

state

<<interface>>
IFAQState

- FAQaccess: FAQ

+ FAQlistState(topic: Topic, public: Boolean)
+ AddQAsState(topic: Topic, qaPair: questionAnswerPair): boolean
+ AddTopicState(parentTopic: Topic, topicToAdd:Topic): boolean
+ AddObserverState(ob: IObserver): void
+ RemoveObserverState(ob: IObserver): void
+notifyStaffObserversState(editorStaffEmail: String): boolean
+notifyStudentObserversState(): boolean
+ runCommand(ICommand<?>): void

AdminFAQState

+ addFAQstate(): void

EverybodyFAQState

+FAQlistState(): void

AdminWebpageState

+ addWebpageState(): void
+ BrowseWebpages(): void

<<interface>>
IWebpageState

- WebpageAccess: Webpages

+ searchWebpagesState(query: String): List<Webpage>
+ addWebpageState(webpage: Webpage, public: boolean): boolean
+ searchWebpageState(webpage: Webpage, query: String): String
+BrowseWebpages(): void
+ AddObserverState(ob: IObserver): void
+ RemoveObserverState(ob: IObserver): void
+notifyStaffObserversState(editorStaffEmail: String): boolean
+notifyStudentObserversState(): boolean
+ConsultWebpages(query: String): List<Webpages>

EverybodyWebpageState

+ ConsultWebpages(query: String): List<Webpages>

AdminQueryState

+RedirectQueryState(email: String): void
+sendQueryAnswerState(title: String, content: String): boolean

TeachingstuffQueryState

+sendQueryAnswerState(query: Query, title: String, content:String): boolean

EverybodyQueryState

+ sendQueryState(senderEmail:String, queryTitle:String, queryContent:String, receiverEmail:String): boolean

<<interface>>
IQueryState

- QueryAccess: staffQuery

+ viewQueriesState(): void
+ sendQueryState(senderEmail:String, queryTitle:String, queryContent:String, receiverEmail:String): boolean
+ redirectQueryState(senderEmail:String, query: Query): boolean
+ sendQueryAnswerState(query: Query, title: String, content: String): boolean
+ AddObserverState(ob: IObserver): void
+ RemoveObserverState(ob: IObserver): void
+notifyStaffObserversState(editorStaffEmail: String): boolean
+ notifyStudentObservers(): boolean
+ getQueryListState(email: String): List<Query>

command package

LoginCommand

- AuthenticatingUserResult: AuthenticatingUser
- username: String
- password: String

+ execute(Context, IView): void
+ getResult(): AuthenticatingUser

LogOutCommand

+ execute(Context, IView): void
+ getResult(): void

AddQAsCommand

- QAsResult: questionAnswerPair
- question: String
- answer: String
- parentTopic: Topic

+ execute(Context, IView): void
+ getResult(): questionAnswerPair

BrowseFAQsCommand

- topicResult: Topic

+ execute(Context, IView): void
+ getResult(): Topic

AddTopicCommand

- topicResult: Topic
- parentTopic: Topic

+ execute(Context, IView): void
+ getResult(): Topic

requestUpdatesfromFAQCommand

- requestUpdatesFromFAQResult: AuthenticatingUser
- email: String
- currentTopic: Topic

+ execute(Context, IView): void
+ getResult(): AuthenticatingUser

<<interface>>
ICommand<T>

+execute(Context, IView): void
+getResult(): T

addWebpageCommand

- webpageResult: webpage
- title: String
- filename: String
- content: String

+ execute(Context, IView): void
+ getResult(): webpage

listWebpageCommand

- listWebpageResult: List<webpage>
- userWebpagesOnly: boolean

+ execute(Context, IView): void
+ getResult(): List<webpage>

showWebpageContentCommand

- showWebpageContentResult: String[]
- webpageFileName: String
- webpageTitle: String
- webpageContent: String

+ execute(Context, IView): void
+ getResult(): String[]

searchKeywordsCommand

- searchKeywordsResult: String[]
- searchQuery: String
- listOfWebpages: List<webpage>

+ execute(Context, IView): void
+ getResult(): String[]

sendQueryCommand

- searchQueryResult: staffQuery
- queryTitle: String
- queryContent: String
- querySenderEmail: String
- queryReceiverEmail: String

+ execute(Context, IView): void
+ getResult(): staffQuery

redirectQueryCommand

- redirectQueryResult: boolean
- query: staffQuery

+ execute(Context, IView): void
+ getResult(): boolean

answerQueryCommand

- answerQueryResult: String
- query: staffQuery

+ execute(Context, IView): void
+ getResult(): String

listQueriesCommand

- listQueryResult: List<staffQuery>
- userQueriesOnly: boolean

+ execute(Context, IView): void
+ getResult(): List<staffQuery>

External Package

<<interface>>
IemailSystem

+ sendEmail:
+ receiveEmail:

<<interface>>
IauthenticatingSystem

+ authenticateUser(): boolean

raises an error if username is not present or if password is wrong

mockEmailSystem

+sendEmail(toEmail: String, title: String, content: String): boolean

ease

+ authenticateUser(username: String, password: String): [String]