Module registration – DB – RESTful interface.

* Module ID (PK), Module Title, Running Year, Staff Id (FK)

User and staff registration should already be created on LDAP, confirm file to match the creation (we shouldn’t need a database for this), this should also mean that Users have a foreign key for module ID.

Message store – DB – RESTful interface.

* MessageID (PK), User owner (FK) (PK created by LDAP), Message body

(Some kind of enum FK for a type of message? (Private, Group, Open) If so then a connected table containing enum)

Notification centre – DB – RESTful interface.

* AutoGen(PK)?, UserID(FK UNIQUE), MessageID(FK), ReceiveNotifsFromSpecificUser(UserID FK UNIQUE), ReiceveNotifsAtAll

Message Groups – DB

* Group Title (PK UNIQUE), Module ID (FK)
* Connected DB -> Group title (FK), UserID(FK)

Front-end server – presentation tier used as the primary centre of the application.

Authentication and authorisation provided by universities LDAP server, file host intermediary for testing.

SignalR – Real time message relay, notifications.

Docker – assembly of containers, images and stacks to host and run the application on designated ports.

Apache Solr – Used to handle the search functionality on a database.

**When user sends a message:**

User sends a message to another user (client tier) -> HTTP post for message is sent (web tier) -> Message is sent for validation check, prohibited words are blocked out (business tier) -> Message is relayed to the database (data tier) -> Database accepts the message and stores it (data tier) -> Response code is sent back as accepted (business tier) -> Message is sent to SignalR for notification relay (business tier) -> SignalR checks for users that should be notified with the message location (business tier) -> notifications are distributed to users (web tier) -> User receives notification pointing to the message location (client tier).

**When user follows a notification:**

User clicks on notification link (client tier) -> HTTP get method is called for the message ID (web tier) -> Relayed to or skips business tier? -> Database accepts the request and returns the data (data tier) -> Data is filled in to a page to be displayed (business tier) -> Data is displayed to the user (web tier).