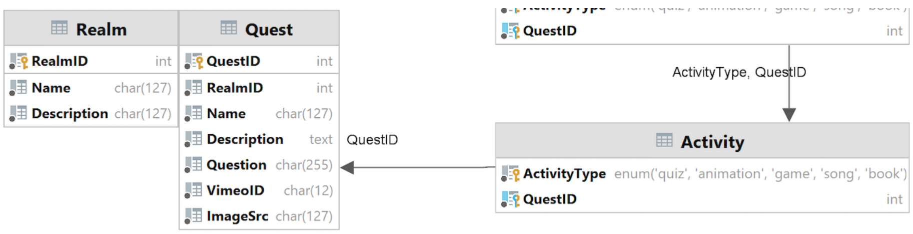
I. Database Sharing

Discussion Topic 1

For the system integration, one of the keys to combine two systems are the database co-usage. Each system (Teacher Portal and Mapping System) should retrieve data from the same database for their own use, since Teacher Portal is using the mapping results from Mapping System whereas the Mapping system needs some resource from the Teacher Portal such as the preview image of each activity card.

Therefore, we have found some tables can be reused for database integration (see questions below).

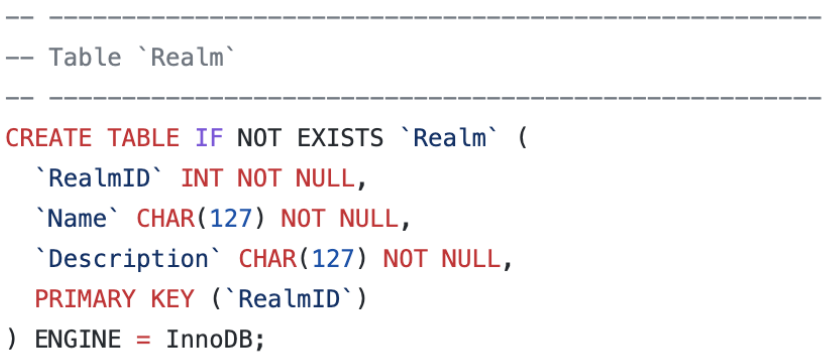
Database schema in Teacher Portal



Questions

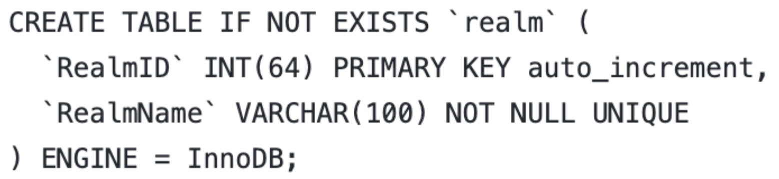
1. **Can we reuse the Realm table in Teacher Portal for new Mapping System?**
   Seems two tables content are similar, potentially to use only one of them

   Realm table in Teacher Portal
   https://github.com/Science-Island/Platform/blob/main/database/schema.sql



```
-- ------------------------------------------------------
-- Table `Realm`
-- ------------------------------------------------------
CREATE TABLE IF NOT EXISTS `Realm` (
  `RealmID` INT NOT NULL,
  `Name` CHAR(127) NOT NULL,
  `Description` CHAR(127) NOT NULL,
  PRIMARY KEY (`RealmID`)
) ENGINE = InnoDB;
```

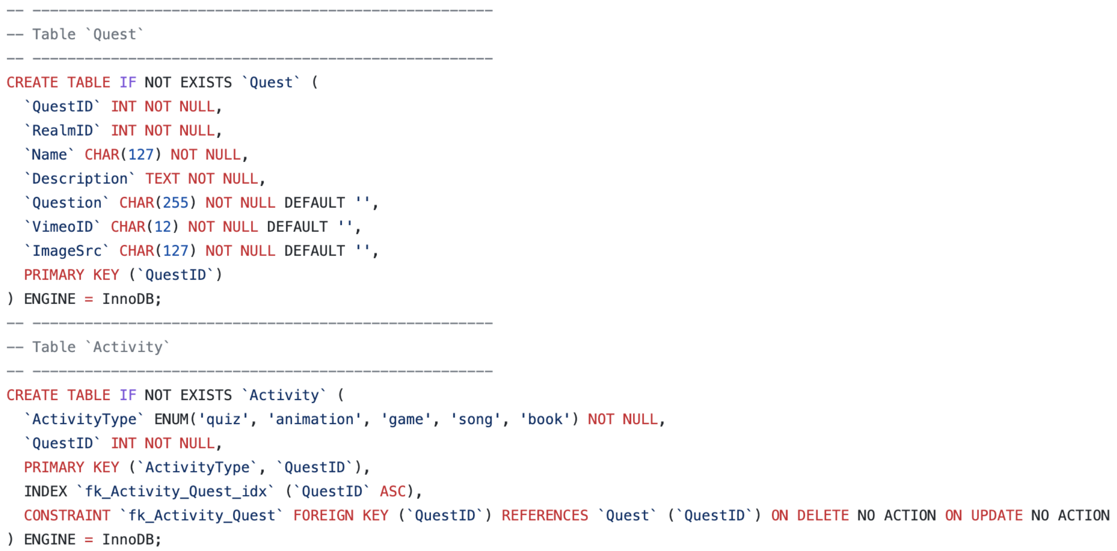
   Realm table in Mapping system
   https://github.com/COMP90082SM12022/SC-Redback/blob/dev/src/backend/database/tableSQL.txt



```
CREATE TABLE IF NOT EXISTS `realm` (
  `RealmID` INT(64) PRIMARY KEY auto_increment,
  `RealmName` VARCHAR(100) NOT NULL UNIQUE
) ENGINE = InnoDB;
```

2. **Can we modify other tables in Teacher Portal for integrating Activity table from Mapping System?**
   We found the Activity table in Mapping System is the expansion of the Activity table in Teacher Portal, so two tables could be combined but involving modify other tables such as Quest table - i.e. inserting a new unique column to Quest table storing the Activity ID so it can be referred.

   Quest & Activity table in Teacher Portal
   https://github.com/Science-Island/Platform/blob/main/database/schema.sql



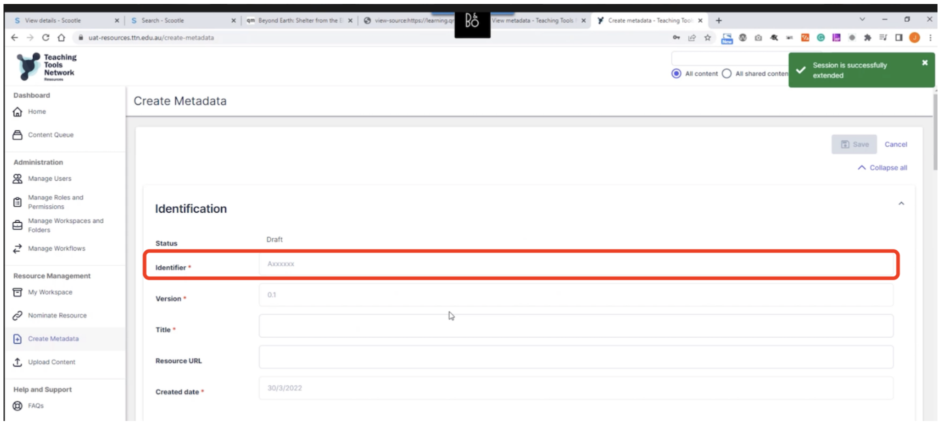
```
-- -----------------------------------------------------
-- Table `Quest`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Quest` (
  `QuestID` INT NOT NULL,
  `RealmID` INT NOT NULL,
  `Name` CHAR(127) NOT NULL,
  `Description` TEXT NOT NULL,
  `Question` CHAR(255) NOT NULL DEFAULT '',
  `VimeoID` CHAR(12) NOT NULL DEFAULT '',
  `ImageSrc` CHAR(127) NOT NULL DEFAULT '',
  PRIMARY KEY (`QuestID`)
) ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `Activity`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Activity` (
  `ActivityType` ENUM('quiz', 'animation', 'game', 'song', 'book') NOT NULL,
  `QuestID` INT NOT NULL,
  PRIMARY KEY (`ActivityType`, `QuestID`),
  INDEX `fk_Activity_Quest_idx` (`QuestID` ASC),
  CONSTRAINT `fk_Activity_Quest` FOREIGN KEY (`QuestID`) REFERENCES `Quest` (`QuestID`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB;
```

   Activity table in Mapping System
   https://github.com/COMP90082SM12022/SC-Redback/blob/dev/src/backend/database/tableSQL.txt

```
-- Table `Activity`
CREATE TABLE IF NOT EXISTS `activity` (
  `ActivityID` INT(64) PRIMARY KEY auto_increment,
  `QuestName` VARCHAR(100),
  `QuestType` ENUM('Quiz', 'Animation', 'Game', 'Song', 'Book', 'Other'),
  `ResourceURL` VARCHAR(200),
  `Link` VARCHAR(200),
  `CreateDate` DATETIME DEFAULT CURRENT_TIMESTAMP,
  `QuestionSpoken` VARCHAR(100) ,
  `RealWorldConnection` VARCHAR(100),
  `ContentDescription` VARCHAR(500),
  `Outcomes` VARCHAR(500),

  `AuthorPhone` VARCHAR(60),
  `AuthorTitle` VARCHAR(60),
  `AuthorEmail` VARCHAR(60),

  `FolderID`  INT(64),
  `RealmID` INT(64),
  `TopicID` INT(64),
  `KeyConceptID` INT(64),
  `PurposeID` INT(64),
  `AgeRangeID` INT(64),
  `ScientistID` INT(64),
  `InstitutionID` INT(64),
  `FieldID` INT(64),
  `CountryID` INT(64),

  `MappingPersonPhone` VARCHAR(60),
  `MappingPersonName` VARCHAR(60),
  `MappingPersonEmail` VARCHAR(60),
  `MapDate` DATETIME DEFAULT CURRENT_TIMESTAMP,
  `MapStatus` BOOL DEFAULT FALSE NOT NULL,
```

3. Can we make the Activity ID in Mapping System auto generated?

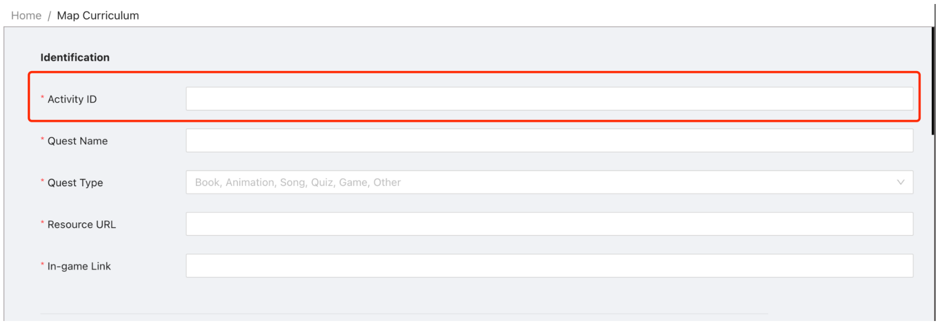
The reference from ESA is user defined, which may have the risk that different users put same ID for different activities, causing the system conflict or system couldn’t identify the activity


View details - Scootle
Search - Scootle
Beyond Earth: Shelter from the E
view-source:https://learning.q
View metadata - Teaching Tools
Create metadata - Teaching Tool
uat-resources.ttn.edu.au/create-metadata
Teaching Tools Network
● All content ○ All shared conten
Session is successfully extended
Dashboard
Home
Content Queue
Administration
Manage Users
Manage Roles and Permissions
Manage Workspaces and Folders
Manage Workflows
Resource Management
My Workspace
Nominate Resource
Create Metadata
Upload Content
Help and Support
FAQs
Save
Cancel
Collapse all

Create Metadata

Identification

Status	Draft
Identifier *	Axxxxxx
Version *	0.1
Title *	
Resource URL	
Created date *	30/3/2022

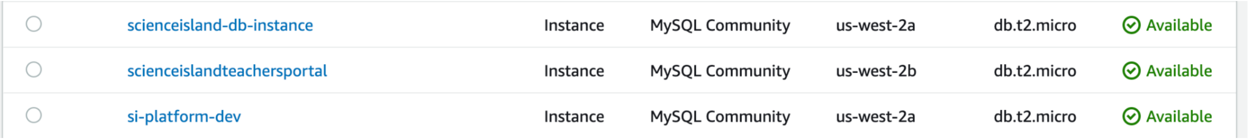

Home / Map Curriculum

Identification

* Activity ID	
* Quest Name	
* Quest Type	Book, Animation, Song, Quiz, Game, Other
* Resource URL	
* In-game Link	

Discussion Topic 2

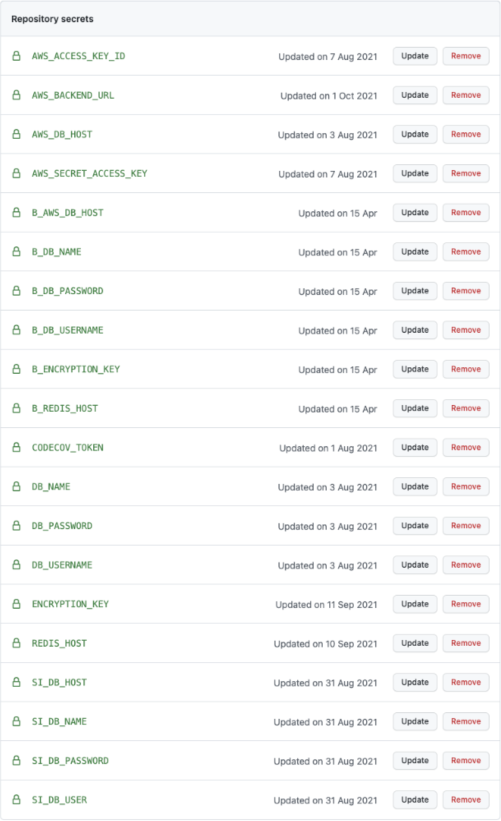
Since the database of Teacher Portal is held in AWS RDS, we need the username and password to access it for schema update, otherwise the system integration could not be done currently.



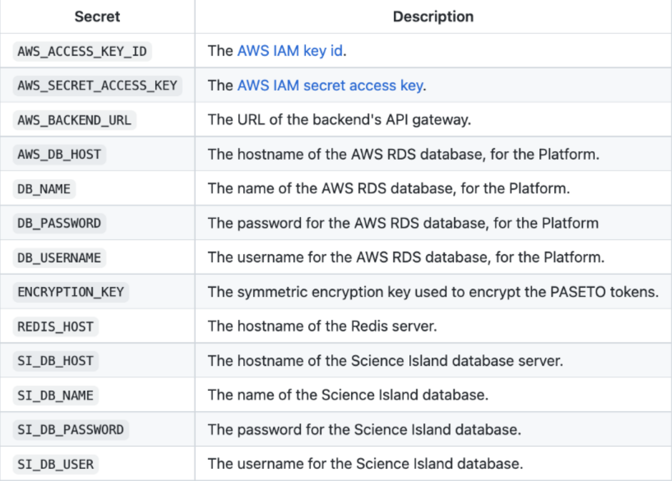
○	scienceisland-db-instance	Instance	MySQL Community	us-west-2a	db.t2.micro	✓ Available
○	scienceislandteachersportal	Instance	MySQL Community	us-west-2b	db.t2.micro	✓ Available
○	si-platform-dev	Instance	MySQL Community	us-west-2a	db.t2.micro	✓ Available

Meanwhile, we also need other usernames and passwords shown on Github, they are all embedded in the repository so we can’t see it. This is also involving the system deployment to be discussed below.

https://github.com/Science-Island/Platform/settings/secrets/actions



Repository secrets			
AWS_ACCESS_KEY_ID	Updated on 7 Aug 2021	Update	Remove
AWS_BACKEND_URL	Updated on 1 Oct 2021	Update	Remove
AWS_DB_HOST	Updated on 3 Aug 2021	Update	Remove
AWS_SECRET_ACCESS_KEY	Updated on 7 Aug 2021	Update	Remove
B_AWS_DB_HOST	Updated on 15 Apr	Update	Remove
B_DB_NAME	Updated on 15 Apr	Update	Remove
B_DB_PASSWORD	Updated on 15 Apr	Update	Remove
B_DB_USERNAME	Updated on 15 Apr	Update	Remove
B_ENCRYPTION_KEY	Updated on 15 Apr	Update	Remove
B_REDIS_HOST	Updated on 15 Apr	Update	Remove
CODECOV_TOKEN	Updated on 1 Aug 2021	Update	Remove
DB_NAME	Updated on 3 Aug 2021	Update	Remove
DB_PASSWORD	Updated on 3 Aug 2021	Update	Remove
DB_USERNAME	Updated on 3 Aug 2021	Update	Remove
ENCRYPTION_KEY	Updated on 11 Sep 2021	Update	Remove
REDIS_HOST	Updated on 10 Sep 2021	Update	Remove
SI_DB_HOST	Updated on 31 Aug 2021	Update	Remove
SI_DB_NAME	Updated on 31 Aug 2021	Update	Remove
SI_DB_PASSWORD	Updated on 31 Aug 2021	Update	Remove
SI_DB_USER	Updated on 31 Aug 2021	Update	Remove

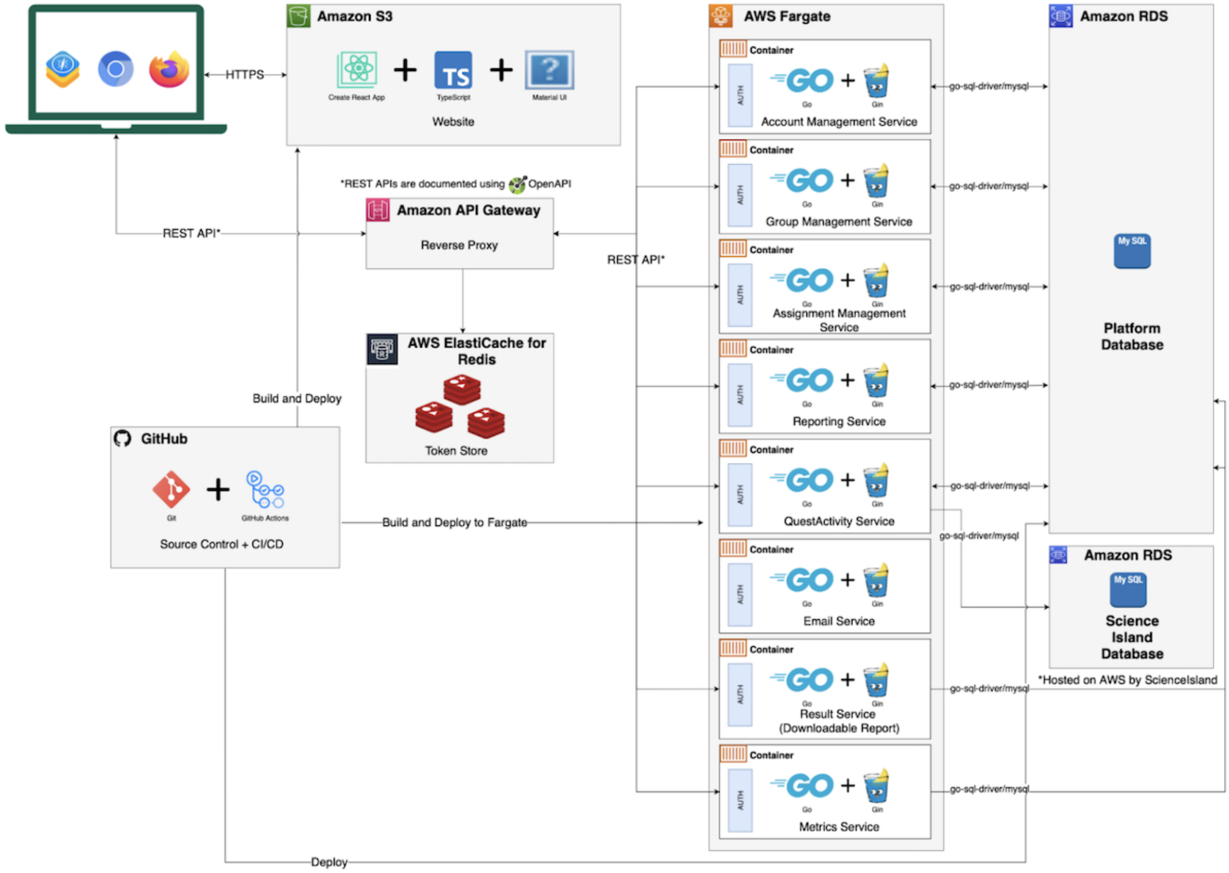


Secret	Description
AWS_ACCESS_KEY_ID	The AWS IAM key id.
AWS_SECRET_ACCESS_KEY	The AWS IAM secret access key.
AWS_BACKEND_URL	The URL of the backend's API gateway.
AWS_DB_HOST	The hostname of the AWS RDS database, for the Platform.
DB_NAME	The name of the AWS RDS database, for the Platform.
DB_PASSWORD	The password for the AWS RDS database, for the Platform
DB_USERNAME	The username for the AWS RDS database, for the Platform.
ENCRYPTION_KEY	The symmetric encryption key used to encrypt the PASETO tokens.
REDIS_HOST	The hostname of the Redis server.
SI_DB_HOST	The hostname of the Science Island database server.
SI_DB_NAME	The name of the Science Island database.
SI_DB_PASSWORD	The password for the Science Island database.
SI_DB_USER	The username for the Science Island database.

II. Deployment

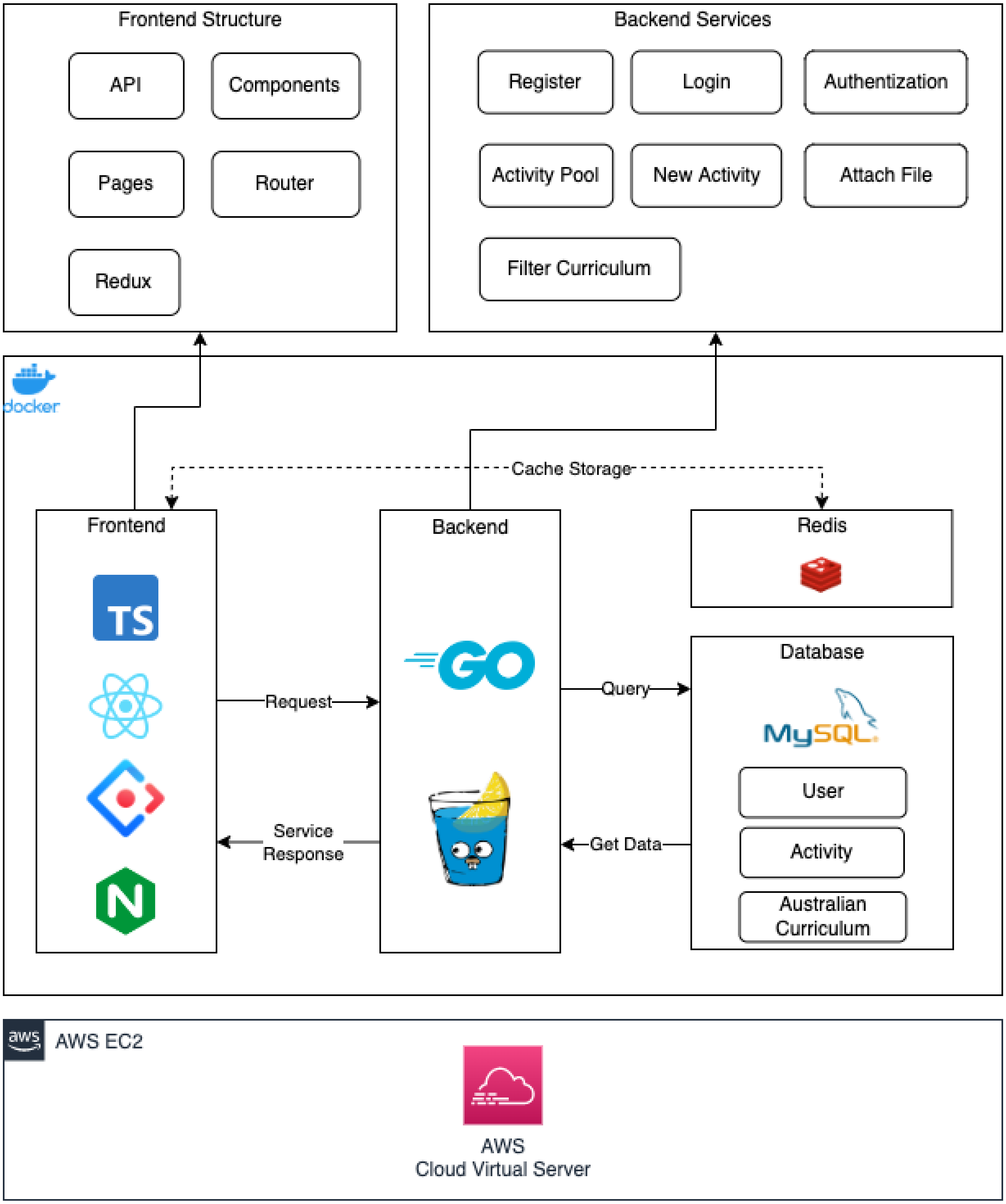
The previous deployment method is quite important for us because, as mentioned before, we have to have all the usernames and passwords listed above to make our new deployment consistent with the previous team. **What we found is that they set up an very comprehensive CI/CD workflow from Github to AWS services. If we don't have a very detailed instruction telling us how to do it step by step, we may not replicate it by ourselves.**

Teacher Portal is using microservice architecture (see picture below). They deployed their backend APIs separately in AWS Fargate which is a container service run by AWS, and using API Gateway to connect frontend and backend. Even though they were using AWS Amplify to deploy their frontend, we don't know how they stored it in AWS S3 and how they made the automatic deployment through Github. And also we don't know how they made the Farget interact with AWS RDS database.



The architecture of the Mapping System is quite different from Teacher Portal. We use an AWS virtual machine EC2 (currently in our own personal account) to deploy the entire project by Docker which is also a type of container service. We packaged whole frontend and backend APIs into two containers respectively then make them interact with Redis

cache storage and MySQL database. Therefore, if there is no detailed instruction from previous team, we may need a new virtual machine launched in your AWS console to hold the new mapping system.



III. Teacher Portal Extension

The extension in Teacher Portal, no matter for “Assignment” section or “Metrics” & “Report” sections, is depending on database co-usage, systems integration and deployment. Especially Teacher Portal is mainly using the data from Mapping System, therefore if the above problems could not be solved, the extension would be hard to continue.

Here are the extension potentially for future work:

Assignment

Filtering/Sorting activities based on the mapping results from Mapping System.

Metrics & report

Display metrics based on mapping results from Mapping System.
Including completed Australian Curriculum in report based on mapping results from Mapping System.