



SGRANC2020 - REST APIs



Project: SGRANC IT7501 Trimester: 1 Year: 2020
By: Gellio Gao & Louise Zhang

BACKGROUND

The Supreme Grand Royal Arch Chapter of New Zealand (SGRANCNZ) Compendium mobile App was originally created by previous teams. Its goal was to provide authorised members of SGRANCNZ with offline access to the resources of the SGRANCNZ Compendium. However, this app could not read data from database.

Therefore, this project was developed for Representational State Transfer (RESTful) APIs using a (Linux, Apache, MySQL, and PHP) LAMP stack that accepts requests (calls) from the mobile app, authenticates a user, and provides a JSON response to the request. The APIs were built in Lumen framework which is a PHP micro-framework for building lightweight APIs that use HTTP requests when handling data.

OBJECTIVE

The project's goal is to ensure SGRANCNZ Compendium App can be synced with the most recent data from database and all functionalities can be accessed by current SGRANCNZ members, while being offline, to be reconciled when going online.

METHODOLOGY

The project team used the Scrum Methodology throughout the project life cycle. Each sprint had an analysis, design, development, testing and delivery phases.

SCOPE

Scope Description:

The scope of this Project is to plan, design, build, implement and test REST APIs for SGRANCNZ Compendium App in order to allow current SGRANCNZ members to access data.

High Level Requirements:

In order to create RESTful APIs to support the SGRANCNZ Compendium App, there are several requirements which needed to be met as part of successful execution of this project:

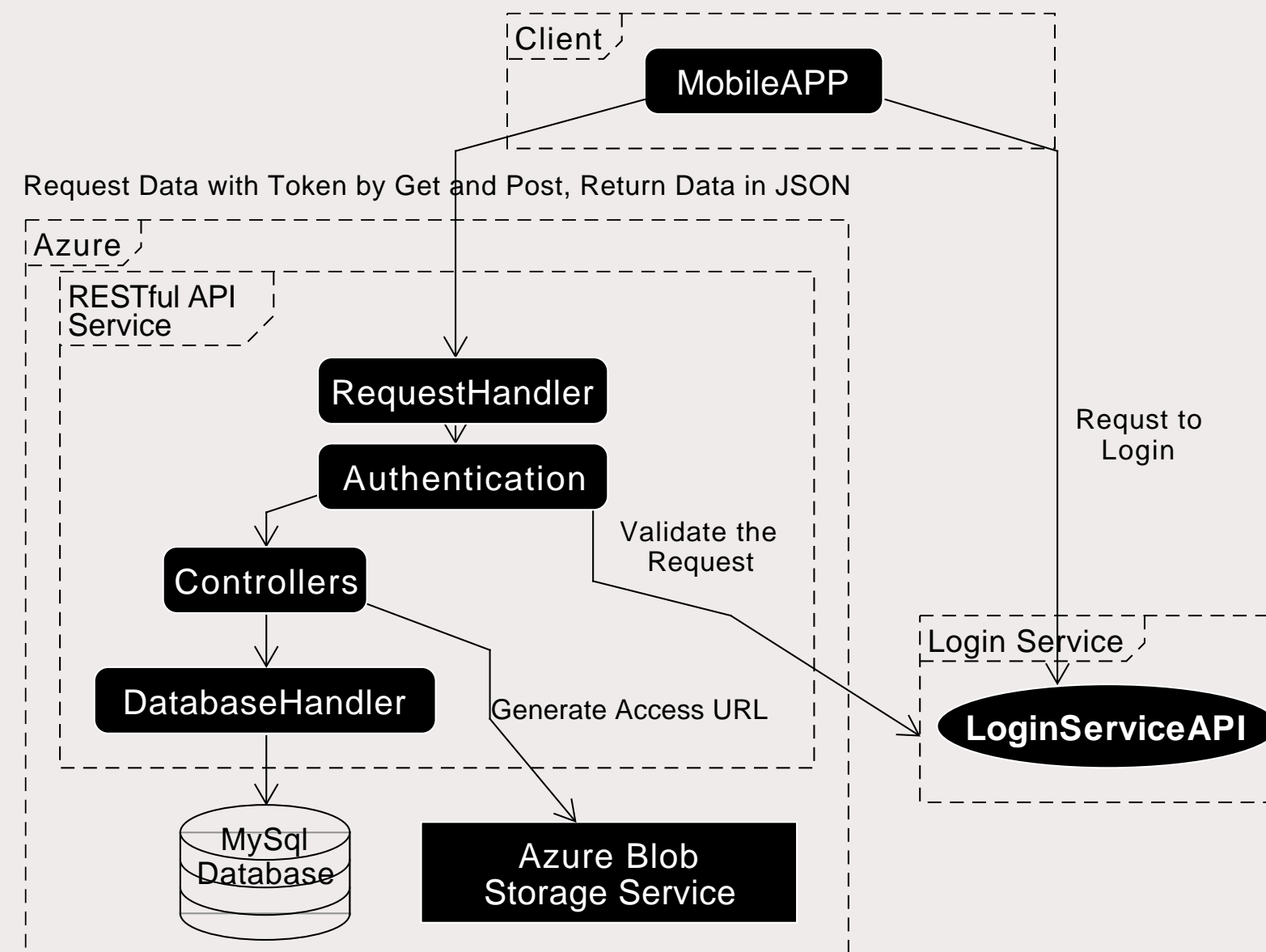
- Users are allowed to access data via RESTful APIs and aggregate information (PDF files, images, audio, etc.) from the database.
- Connection to the client's Database Infrastructure for the application.
- Testing: API test.

Project Boundaries:

Not included in the scope of this project:

- Ongoing maintenance of database.
- Login and registration.
- Upgrade mobile app to communicate with APIs.

SYSTEM ARCHITECTURE FLOWCHART



API ENDPOINTS

| API | Method | Action | Description |
|--------------------------------------|--------|---------------------------------------------|-----------------------------------------------------------|
| /Version/Member | GET | MemberController@getAll | Get all members profile details. |
| /Version/Member/{ID} | GET | MemberController@getMember | Get the specific member profile detail by given the id. |
| /Version/Ceremonial | GET | CeremonialController@getAll | Get all ceremonials data. |
| /Version/Ceremonial/{OrderID} | GET | CeremonialController@getCeremonialSByOrder | Get the specific ceremonials data by given the order id. |
| /Version/CeremonialMember/{MemberID} | GET | CeremonialController@getCeremonialSByMember | Get the specific ceremonials data by given the member id. |
| /Version/Regalia | GET | RegaliaController@getAll | Get all regalia data. |
| /Version/Regalia/{OrderID} | GET | RegaliaController@getRegaliaByOrder | Get the specific regalia data by given the order id. |
| /Version/RegaliaType/{Type} | GET | RegaliaController@getRegaliaByType | Get the specific regalia data by given the regalia type. |
| /Version/Degree | GET | DegreeController@getAll | Get all degrees data. |
| /Version/Degree/{ID} | GET | DegreeController@getDegree | Get the specific degree data by given the id. |
| /Version/DegreeMember/{ID} | GET | DegreeController@getDegreesByMember | Get the specific degrees data by given the member id. |
| /Version/Division | GET | DivisionController@getAll | Get all divisions data. |
| /Version/Division/{ID} | GET | DivisionController@getDivision | Get the specific division data by given the id. |
| /Version/District | GET | DistrictController@getAll | Get all districts data. |
| /Version/District/{ID} | GET | DistrictController@getDistrict | Get the specific district data by given the id. |
| /Version/Group | GET | GroupController@getAll | Get all groups data. |
| /Version/Group/{ID} | GET | GroupController@getGroup | Get the specific group data by given the id. |
| /Version/Chronology | GET | ChronologyController@getAll | Get all chronology data. |
| /Version/Chronology/{ID} | GET | ChronologyController@getChronology | Get the specific chronology data by given the id. |
| /Version/History | GET | HistoryController@getAll | Get all history data. |
| /Version/History/{ID} | GET | HistoryController@getHistory | Get the specific history data by given the id. |
| /Version/Glossary | GET | GlossaryController@getAll | Get all glossaries data. |
| /Version/Glossary/{ID} | GET | GlossaryController@getGlossary | Get the specific glossary data by given the id. |

DEVELOPMENT

The project was started by gathering requirements and client expectations. As a result of this, the team submitted a proposal with a plan of action to follow SCRUM and organise the development sprints and tasks.

After the proposal gained approval, the team started system analysis and researching development techniques that could be used. The research helped to determine that the Lumen Framework and a LAMP stack would be the best fit options for building lightweight APIs.

After setting up the development environment, the team created system architecture and API list. This was an important step as it helped to ensure that the client expectations were met.

During the development stage, when a milestone was reached, testing was also done in parallel using Postman to keep a consistent workflow. As a team all code was merged using Azure DevOps and then progressed to the next sprint.

Lastly, the final documents were finished and compiled in technical documents format which could be used for future developments.

CONCLUSION & RECOMMENDATIONS

The team members used the essential skills of project management to maintain the relationship with the client and to carry out the process within the team.

The team compiled all recommendations and heavily documented the source code to make sure any party who may continue with the development have a good knowledge base to work with.

The team has performed adequately in order to complete the project and gained a good learning experience and the skills from this project.

PROJECT TEAM

Project Client: **Ken Burt**

Academic Advisor: **Dr. Waqar Khan**

Project Coordinator: **Dr. Scott Morton**

Team member: **Gellio Gao**

Team member: **Louise Zhang**

TOOLS USED

Visual Studio Code (Source-code Editor)

Lumen by Laravel (The php framework)

XAMPP (For dev environment)

POSTMAN (Rest client for testing the API)

apiDoc (Generate the API documentation)

PHPUnit (Testing framework for unit testing)

Basecamp (Project Management)

Azure DevOps (Source Control and Project Management)

