COMP 3059 – Capstone Project I

Software Requirements Analysis and Design Assignment

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:
http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf

http://gmitweb.gmit.ie/pdunne/sweng/03-Requirements.pdf

www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc

https://nces.ed.gov/pubs2005/tech_suite/part_2.asp

Reference template:
www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc

1. 1.0 Introduction

The Introduction section provides an overview of the system using software requirements analysis and design for the scope of the system.

Literacy Uplift is

1.1. Purpose

The purpose of Literacy Uplift will be to improve the reading and writing skills of students enrolled in literacy courses and classes with the goal of improving student's workplace efficiency. Students improvement is expected to reach a literacy level of 3 based on the Programme for the International Assessment of Adult Competencies (PIAAC) test. Teachers leading a class will be able to monitor the progress of their students, who will be organized into classes just like those they are enrolled in in real life.

1.2. Scope

Literacy Uplift will be the development of the web extension of the initial mobile application, and the development of the backend service for the application. The mobile application has already been completed and will not be developed alongside the web extension. The web extension will require users to log in before interacting with the service. Once logged in they can take part in activities designed to improve their reading comprehension and literacy skills. Results of the user's activities will be stored on the server and will generate a progression system to show the users improvement over time using the system. Users will enroll in classes with their teacher and will be able to communicate with their teacher and classmates through a chat system.

Teachers will be able to monitor the progress of their students who will be grouped together with the teacher as the head of the class. Activities that students complete will be viewable by the teacher who can monitor their performance in each activity they complete. Teachers will be able to offer assistance to the student through a chat system.

2. System Overview

The System Overview section introduces the system context and design.

2.1. Project Perspective

The Project Perspective describes the context and origin of the system by defining whether the system is:

- a follow-on member of a system family
- a replacement for existing systems, or
- a new self-contained system.

Literacy Uplift is a follow-on member of the already developed Literacy Uplift mobile application. It will be the web version of the application. It was created to expand the accessibility of the system to Desktop and Laptops. As only the front-end of the mobile application has been create this document will also cover the development of the back-end service for Literacy Uplift.

2.2. System Context

The System Context describes the resulting software within the business case, including strategic issues in which the system is involved or which it specifically addresses.

According to the Canadian Learning and Literacy Network (CLLN) statistically around 42% of Canadians between ages 16 and 65 have low literacy skills. Among new immigrants to Canada this is even higher. As the Canadian job market requires more and more job seekers to be sufficient in their reading as a basic and essential requirement for work this excludes . The goal of Literacy Uplift is to raise the percentage among Canadians regardless of status to adapt to economical changes in the digital age through mobile learning in both classrooms and online.

Literacy Uplift will be designed as a way to compliment classroom learning through an online structure focused on task repetition and activity practice. User interaction will focus on reading comprehension and oral reading fluency, as well as typing and sentence processing. Targeted at both Canadian citizens and immigrants, Literacy Uplift will focus on English communication skills, not English as a Second Language (ESL).

2.3. General Constraints

General Constraints identify any business or system constraints that will impact the manner in which the software is to be:

- specified
- designed
- implemented, or
- tested.

When designing the web app, the styling of pages will be constrained to what the mobile app looks like currently or what the stakeholders decide, if the stakeholders decide to adopt a different style.

- Web application will receive data through web API. This will require the proper coding of the system to follow rules of the API in terms of data retrieval and data storage.

- Use of Laravel framework. Using a framework require that the system follow the rules set down by the framework. With Laravel the database must be designed inside Laravel itself.

- Activities creation. Because the system is being designed for use not by the developers or any customers, but instead by the teachers and students a business constraint will be how the website content is created.

- An outside designer will need to be brought into designed the front end; because this project is being designed primarily to implement backend service for both web and mobile there will need to be extra work performed to complete the project before release.

2.4 Assumptions and Dependencies

List any assumptions that have been made during the initiation of the project. In addition, list any dependencies that may impact its success or the desired result.

DEPENDENCIES:

- People enrolling in the Literacy Uplift program.
- Students having a connection to the internet

ASSUMPTIONS:

- Server for testing will be available
- Test data will be provided by stakeholder
- Additional documents, if required, will be provided by project-manager/ stakeholder.

3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

Features of Literacy Uplift are activities and the live chat system. Activities are the main focus of the website for user interaction, and how they will practice their literacy skills outside of lessons. The chat system will be how the students and their teacher will communicate over activity problems or difficulties with lessons. They will be

Functional Req. ID	Title	Description
F1	Authentication	System should provide authentication
F2	Registration	System should allow users to register

F3	Tracking	System should provide ability to track user actions
F4	Progress presentation - Individual	System should provide UI for a user (learner) to see individual progress based on tracking data
F5	Progress presentation - group	System should provide UI for a user (teacher) to see and drill the progress of learners
F6	Add Activities	System should provide functionality to add new activities
F7	Add Units	System should provide functionality to add units and group activities
F8	Add Objectives	System should allow teachers to add objectives
F9	Tag Activity	System should allow teachers to tag activities with Objectives (one or more)
F10	Nested Objectives	System should allow users to nest Objectives (objective that is a parent to other objectives)
F11	Edit and version activities	Uses should be able to edit activity and save new version without destroying old one
F12	Add activity to the unit	User should be able to add activities to units (1 activity can be in many units)
F13	API	System should provide web API (restful webservice) to access the data
F14	UI	System should provide web UI to access selected pieces of the system (stats, user management)
F15	Administrator	Administrators can control the system and make modifications to help teachers and students

3.1 <Functional Requirement or Feature #1>

- Introduction, Inputs, Processing, Output

Authentication

- The system should be able to identify and verify that the user has an account, and that their identity matches the store credentials. It will take a username and password, and match those against the stored unique username and password. The user will then be login.

Registration

- The system will allow a user to register an account with Literacy Uplift. It will ask for a username, and a password. The username and password will be stored on the server and used for future logins. The user will then be able to log into the system.

Tracking

- The system will be able to track a user's actions. It can track the performance of a user in an activity and keep scores of their attempts. The system will also track their communications with their teacher and other students.

Progress presentation - Individual

- Users will be able to visually see individual progress based on tracking data. The tracked data will be applied to test performance, activity scores, and a total improvement progress shown using a mathematical formula.

Progress presentation - group

- Teachers will be able to see a group performance in the UI. This will reflect the total progress of each student in the class through their tracked data, and allow for comparisons between each individual student.

Add Activities

- The system will provide functionality to add new activities. Teachers will be able to create new activities for students for different learning tasks. Activities would be typing, verbal reading, static reading, and reading comprehension.

Add Units

- The system should provide functionality to add units and group activities. This allows students to interact in a cohesive manner when practicing on materials learned in class. It will assist the teacher in organizing course materials to better suit students base on levels of skill.

Add Objectives

- The system should allow teachers to add objectives. This will allow for homework to be assigned, or for the teacher to create learning goals for students to follow as they practice through the application.

Tag Activity

- System should allow teachers to tag activities with Objectives. Tagging activities allows for the teacher to highlight for students the activities that they wish to see them complete. Tagged activities may also be used to highlight individual tasks that an individual student should focus on.

Nested Objectives

- System should allow users to nest Objectives. With nested objectives the teacher can specify a global objective with multiple smaller objectives within for students to achieve. This can form a complete portfolio of the skills that make up a goal for the class, or student.

Edit and version activities

- Uses should be able to edit activity and save new version without destroying old one. Multiple versions being created, but saved, will allow for the the teacher to create several versions that may be dropped later, but can be used if the teacher deems them to be useful for other classes or lessons in the future.

Add activity to the unit

- User should be able to add activities to units. A unit can have multiple activities stored in it, but activities in there may be used across multiple units. The allows for the pairing of activities across units that may be different but require same skills from those units. It will also save on time spent create units and activities for the classes.

API

- System should provide web API (restful webservice) to access the data. Using the xAPI will allow the system to be coded in accordance with the data request rules. Using a RESTful API allows for data to not be flowing freely back and forth, but to be access only when needed by the system.

UI

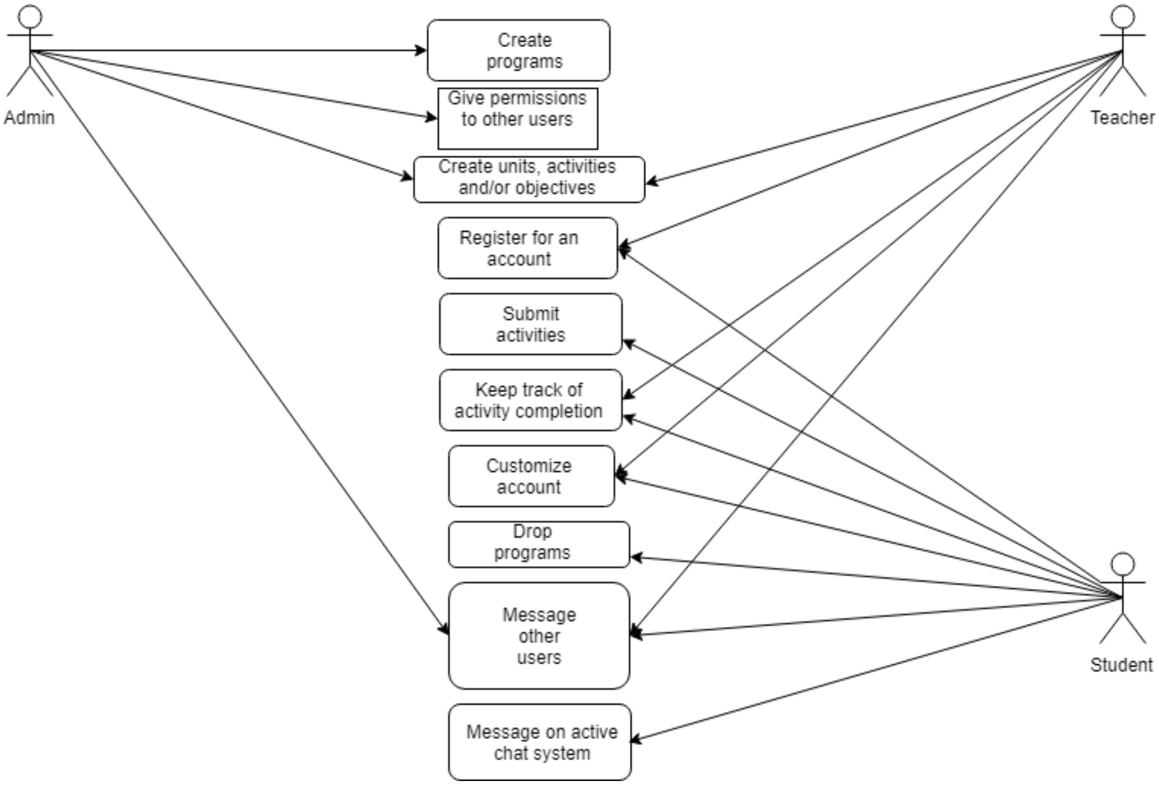
- System should provide web UI to access selected pieces of the system (stats, user management). Creating UI elements allow for administrators and teachers to access data from the server related to their tasks; administrators can access users logs and teachers can access student activity progress. With students, UI elements accessing data can show their overall work progress, not in terms of grades but in a comprehensive skill breakdown to show they truly are improving themselves. Accessing complex Nested Objectives will need a UI carefully designed for efficiency, but also intuitive exploration.

Administrator

- Administrators can control the system and make modifications as needed, or as request by teachers. Teachers can have other teachers added by the administrator. Administrators can also remove accounts from classes, and delete old units no longer needed from the version control.

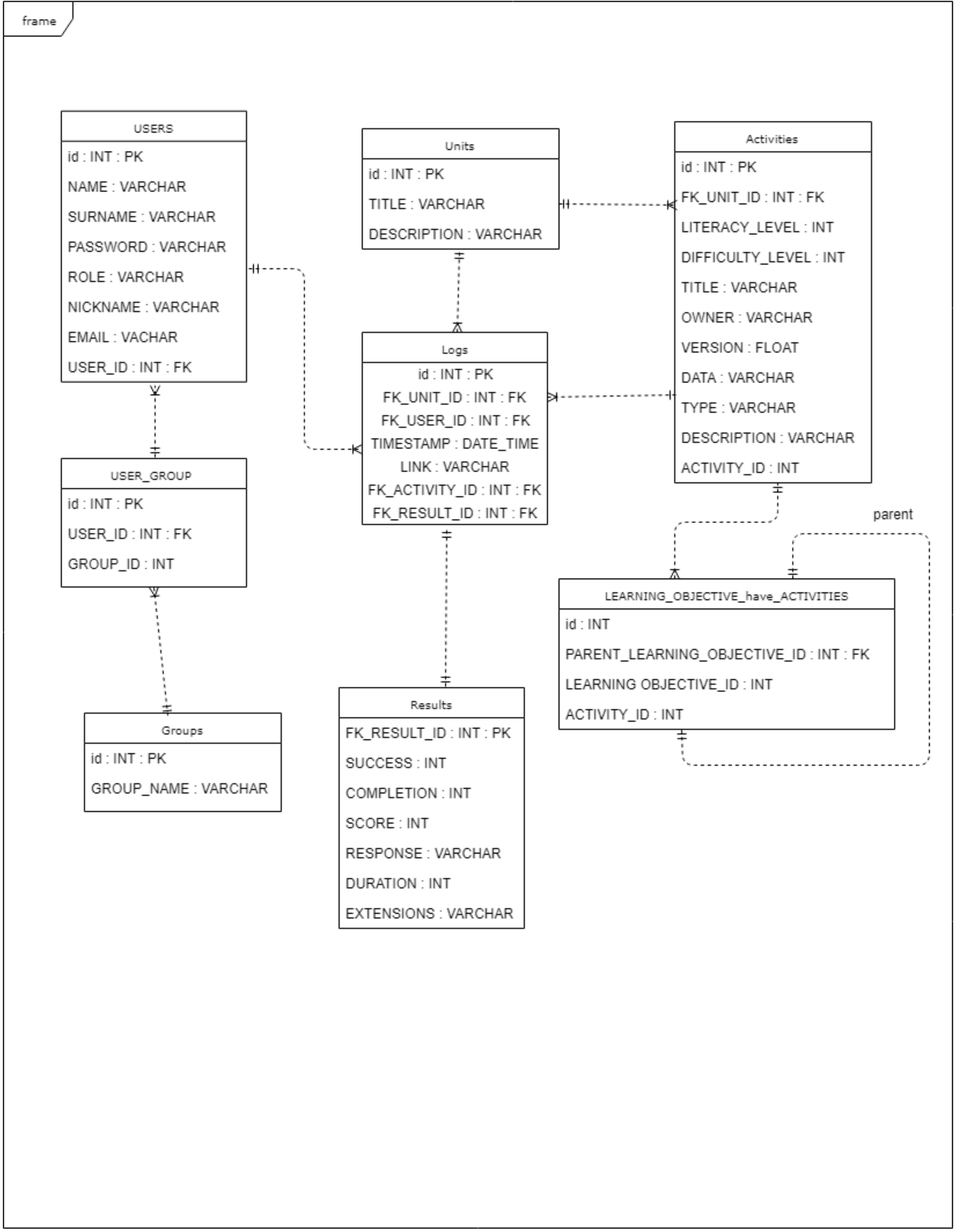
3.2 Use Cases

3.2.1 Use Case #1 …

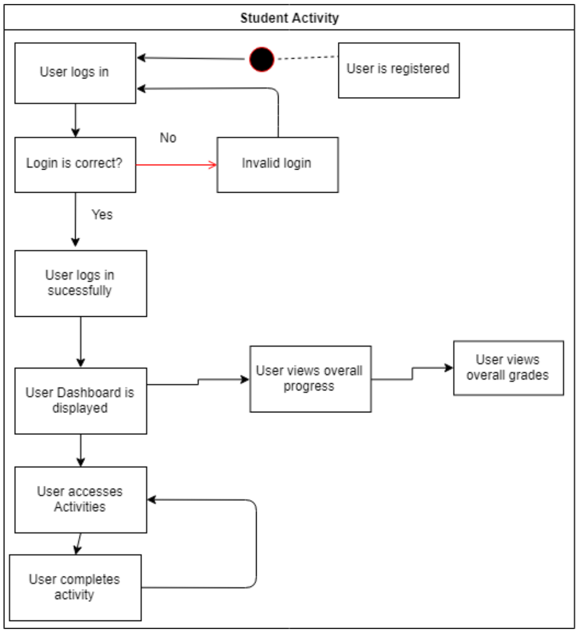

Admin
Teacher
Student
Create programs
Give permissions to other users
Create units, activities and/or objectives
Register for an account
Submit activities
Keep track of activity completion
Customize account
Drop programs
Message other users
Message on active chat system

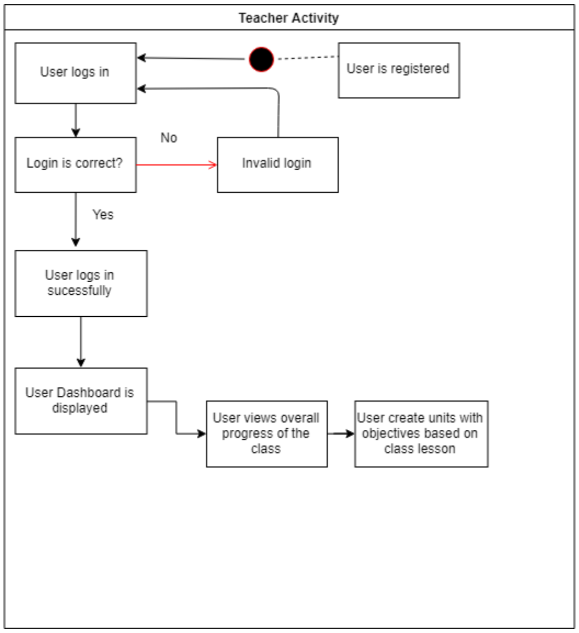
3.3 Data Modelling and Analysis

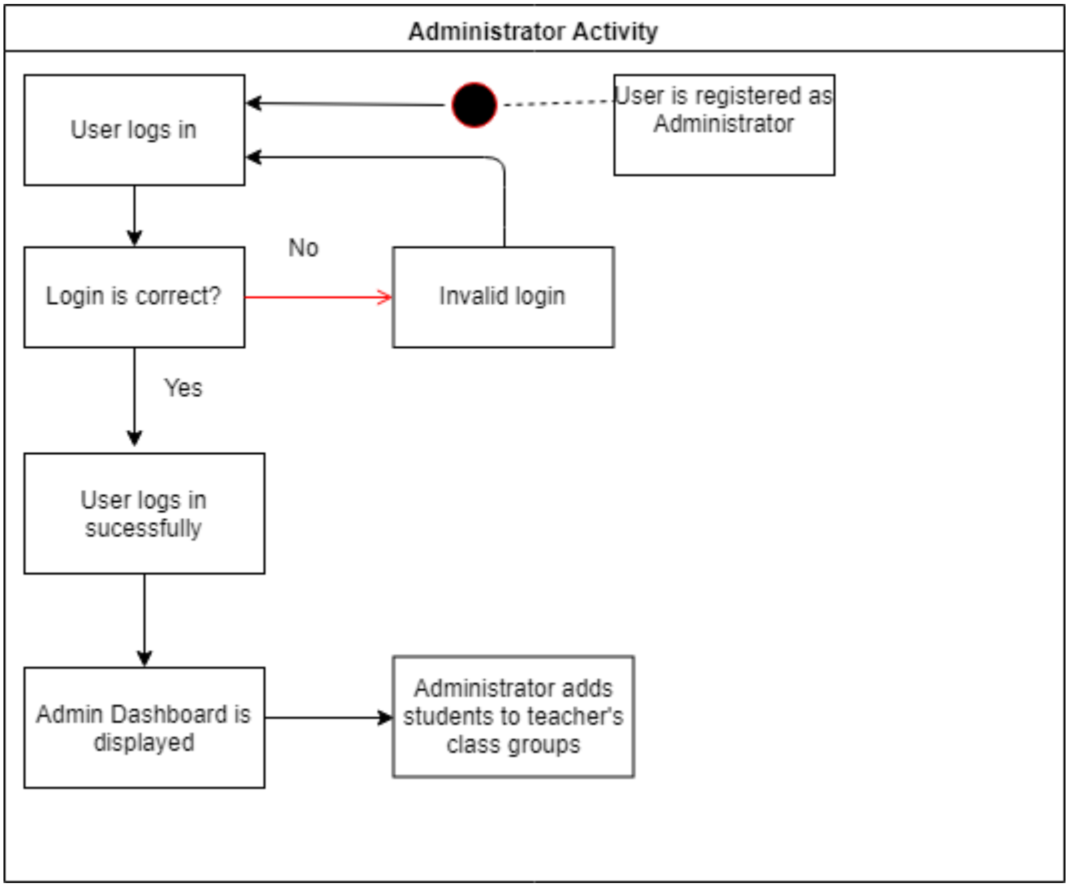
- Normalized Data Model Diagram



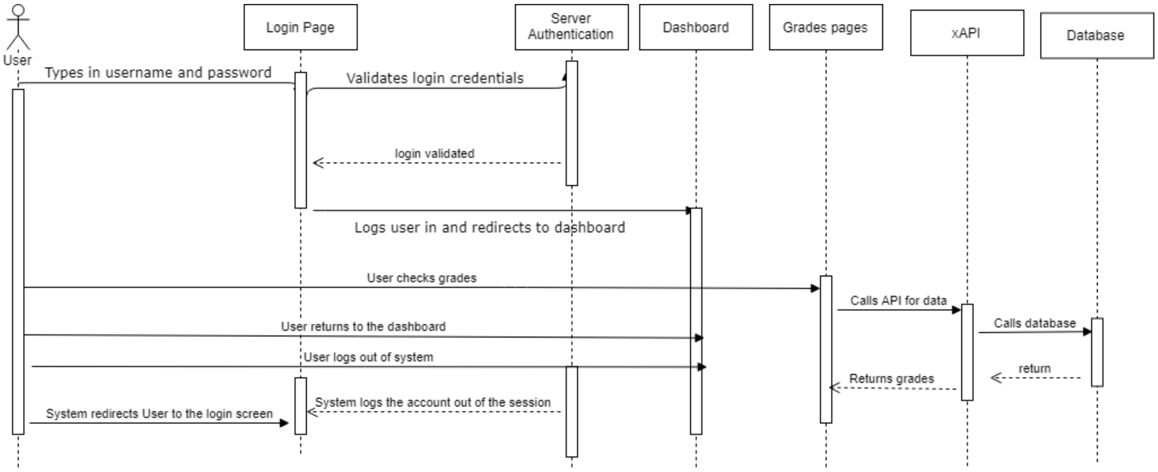
- Activity Diagrams


Student Activity
User logs in
User is registered
No
Login is correct?
Invalid login
Yes
User logs in sucessfully
User Dashboard is displayed
User views overall progress
User views overall grades
User accesses Activities
User completes activity

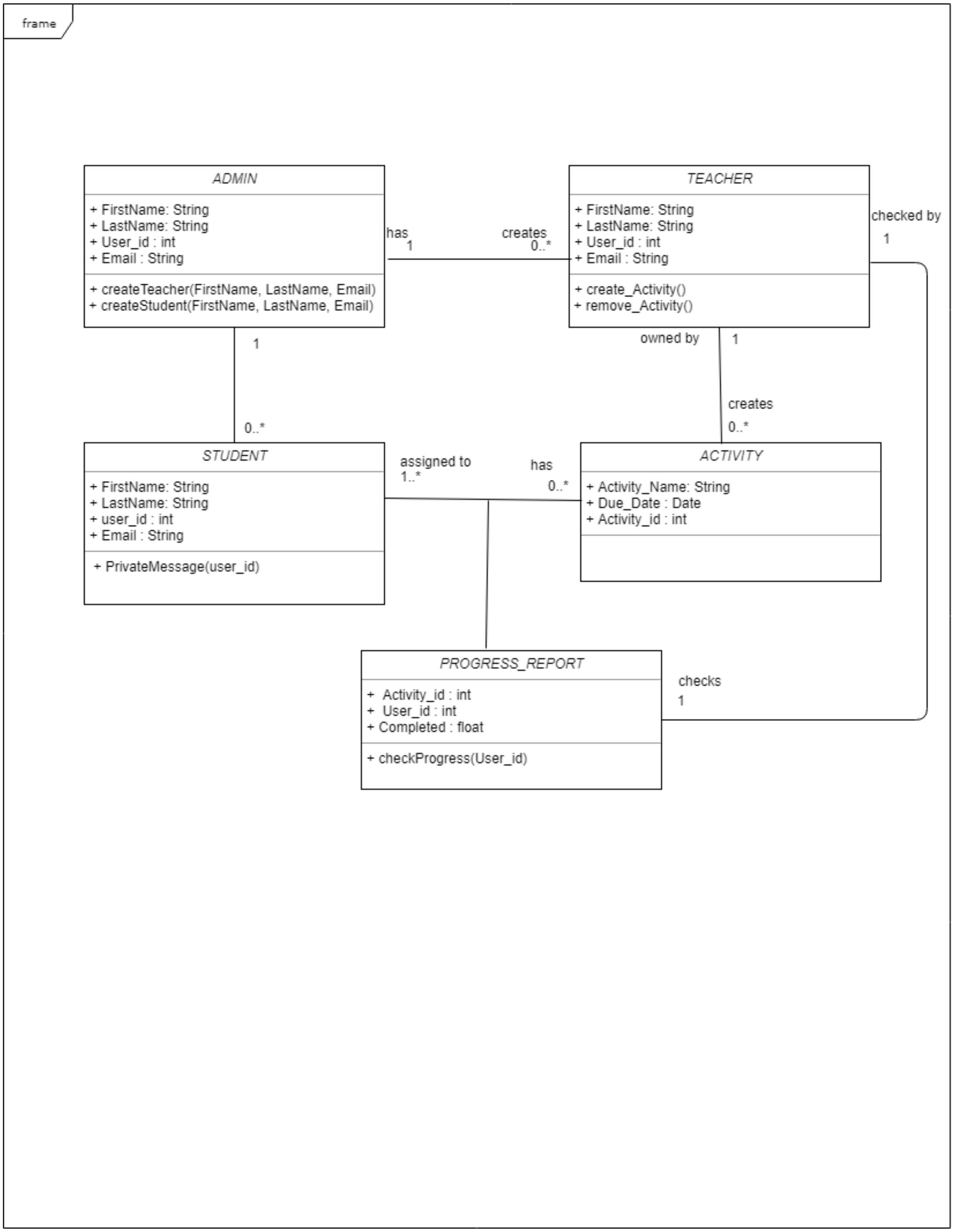

Teacher Activity
User logs in
User is registered
No
Login is correct?
Invalid login
Yes
User logs in sucessfully
User Dashboard is displayed
User views overall progress of the class
User create units with objectives based on class lesson


Administrator Activity
User logs in
User is registered as Administrator
No
Login is correct?
Invalid login
Yes
User logs in sucessfully
Admin Dashboard is displayed
Administrator adds students to teacher's class groups

- Sequence Diagrams

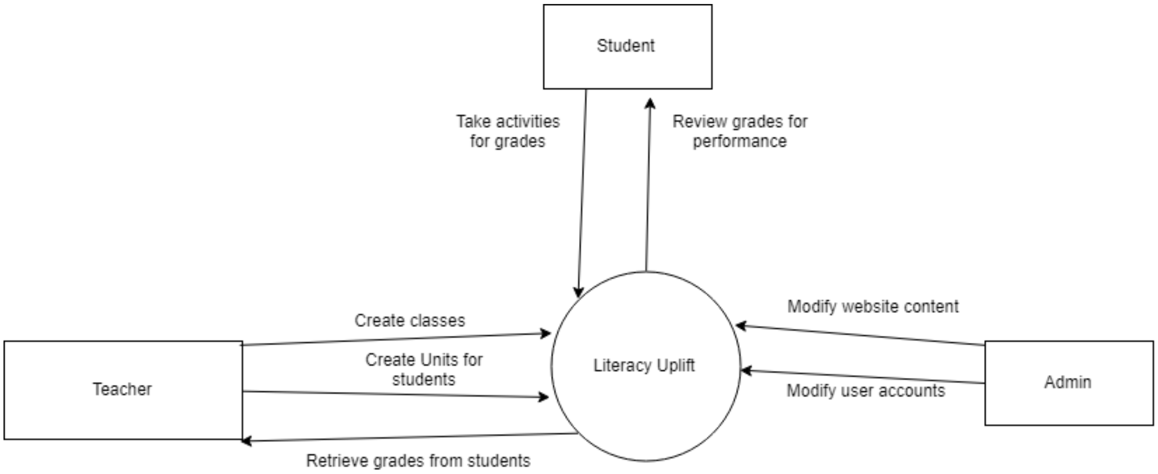

User
Login Page
Server Authentication
Dashboard
Grades pages
xAPI
Database
Types in username and password
Validates login credentials
login validated
Logs user in and redirects to dashboard
User checks grades
Calls API for data
Calls database
return
Returns grades
User returns to the dashboard
User logs out of system
System logs the account out of the session
System redirects User to the login screen

- UML Class Diagram



3.4 Process Modelling

- Data Flow Diagram


Student
Take activities for grades
Review grades for performance
Create classes
Create Units for students
Literacy Uplift
Modify website content
Modify user accounts
Teacher
Admin
Retrieve grades from students

4.0 Non-Functional Requirements

The non-functional requirements for a system are typically constraints on the functional requirements – that is, not what the system does, but how it does it (e.g. how quickly, how efficiently, how easily from the user's perspective, etc.).

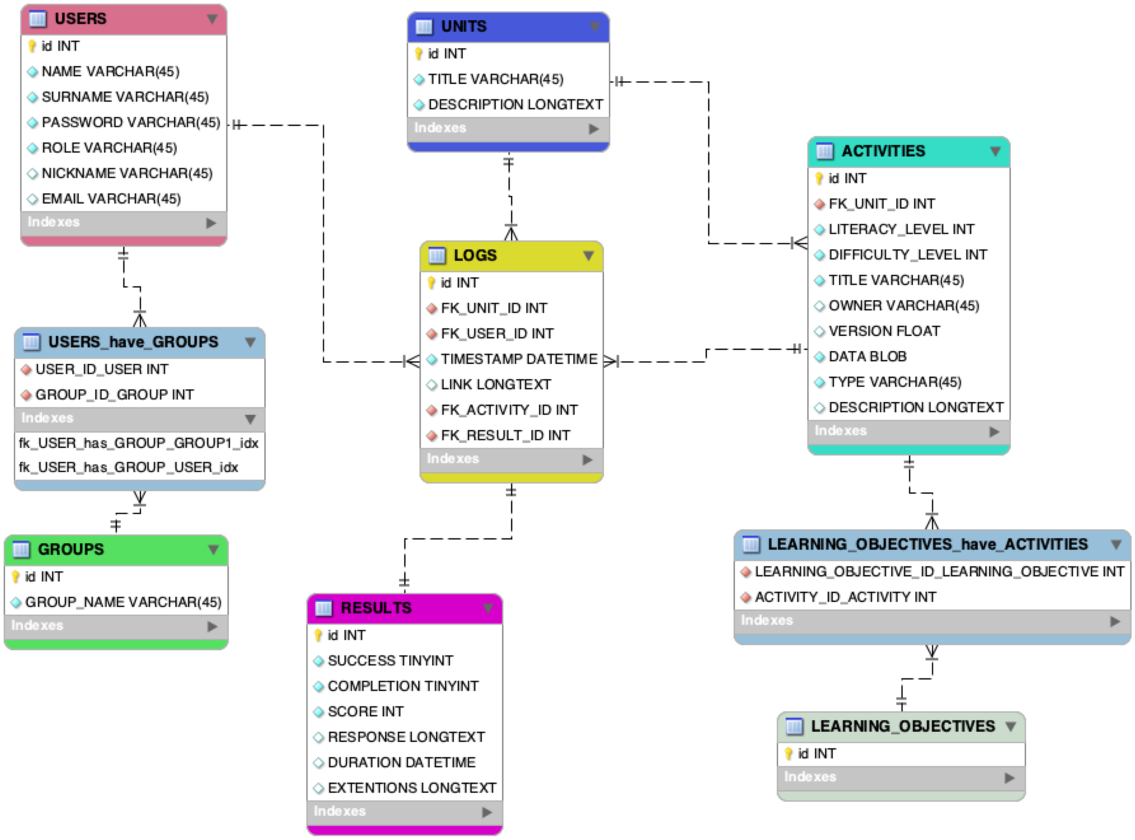
Non-functional requirements may exist for any of the following attributes – Performance, Reliability, Availability, Security, Maintainability, Portability.

Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, etc).

- Progress presentation (i.e progress bar), will be updated as soon as a progress change has been tracked.
- Teachers can add activities, units or objectives at any time, as long as administrator has given the teacher's account privileges.
- Students can leave in the middle of an activity and the progress will save. They can resume from where they left off, when the student returns

5.0 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc?



6.0 Other Requirements

Additional requirements, if any.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Developer	Nicholas Entecott	nentecott	06/11/2018
Developer	Maxim Paxton	maximthomaspasxton	06/11/2018
Developer	Nehaal Shaikh	nshaikh	06/11/2018
Developer	Patrick Murphy	pmurphy	06/11/2018