ASHESI UNIVERSITY

TABLET-BASED MATH APPLICATION FOR PRIMARY SCHOOL PUPILS IN GRADE 1 AND GRADE 2

APPLIED PROJECT

B.Sc. Computer Science

Nanis Kanana

2019

ASHESI UNIVERSITY

Tablet-Based Math Application for Primary School Pupils in Grade 1 and Grade 2

Applied Project

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science

Nanis Kanana

April 2019

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date: .....

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date: .....

Acknowledgement

I would like to express my sincere gratitude to my supervisor, David Amatey Sampah for guiding me throughout the project period, for feedback and support. I am also thankful to Prof. Olaf for his invaluable contribution to this project.

To my family and friends, thank you for your encouragement and support.

Abstract

Education is one of the basic needs of a child. The knowledge imparted through education is key to the recipient's growth and skilfulness. In most cases, education is attained by attending a school (face to face) where several subjects are taught. Of all the subjects taught in schools, mathematics is a fundamental subject that cuts across all disciplines, hence a significant field for all school-going children.

For some children, they find it easy to understand basic mathematical concepts and excel tremendously in the subject. However, quite a number struggle to learn both deep mathematical concepts and the most essentials skills such as addition, subtraction, multiplication, fractions, and decimals.

As a result, this project seeks to develop a tablet-based math application for primary school pupils in grade 1 and grade 2, based on the Ghanaian education system. The app simplifies addition, subtraction and object grouping for pupils by making it fun through interactive game-like activities.

Table of Contents

Chapter 1: Introduction

1.1 Background

Mathematics is a compulsory subject from kindergarten, lower primary to upper primary, junior high school and senior high school of the Ghanaian education system. At each level, the syllabus builds on the knowledge and competencies developed in the previous level. This requires that a strong foundation is laid at the kindergarten level and lower primary so that in Senior high school students can solve real-life problems and are competent enough to venture into computing, e-commerce, science and other math-related fields at the tertiary level as stated in the Ghana teaching syllabus for core mathematics presented by UNESCO in 2010[12]. However, there has been a low performance in Mathematics over the years. According to the West African Examination Council (WAEC), in 2017, only 11 percent of primary six pupils and 22 percent of primary three excelled in Mathematics[13].

ICT (Information Communication Technology) was officially introduced into the education system in 2007 by the Ghanaian government as one of the strategies to improve learning[1]. The kind of ICT in schools however only introduces the pupils to basic computing skills such as typing and internet use. The system has not yet adopted ICT as a method of teaching and learning. Adoption of technology could also help solve the challenge of class overpopulation in most public schools. Currently, most schools use curriculum textbooks as the only material for teaching math, which provides limited knowledge and students are not able to explore activities. Most of the content is theoretical, and rarely, students do practical exercises in the classroom to

enhance the understanding of math concepts. A tablet-based math application will promote interactive, practical and fun learning for the pupils.

1.2 Problem

Some children in Ghanaian schools struggle to learn both deep mathematical concepts and essential skills such as addition, subtraction, multiplication, fractions, and decimals. The Chairperson of the West African Examination Council (WAEC), has previously admitted to poor performance in Mathematics in West African Senior School Certificate Examination (WASSCE). He referenced a 2013 survey that showed that lower primary students experienced challenges in mathematizing addition and subtraction[14].

1.3 Motivation/Why the problem is important

Mathematics is a foundation subject that cuts across all disciplines. It is one of the most important foundational subjects in any curriculum for primary education[7]. In this era of technological advancement, the usefulness of the subject in problem-solving, and decision-making is immeasurable.

Poor Teaching Methodology; Poor performance. Over the years, primary schools in Ghana have recorded poor performance in Mathematics, and tertiary institutions have experienced low enrolments of students pursuing Mathematics majors[14]. The poor performance is attributed to the fact that at the foundational levels, students have difficulties understanding the subject and with time, poor performance demotivates them from pursuing math at higher levels.

In a workshop held in Accra in February 2016 themed "Revamping Mathematics Education in Ghana through Transformation", the Minister of Education observed that "poor

teaching methodology, inadequate instructional materials, low understanding of test items, inadequate coverage of syllabus and poor communication skills are some of the reasons for low performance of students in subjects like Mathematics and Science"[2].

Understanding mathematics and developing a positive attitude in the subject helps students realize the importance of the subject in solving real-life problems and its application in pursuing future professions. This can be achieved by making the subject fun and loveable through technological activities at the early stages of school so that they can grow with it.

Integration of technology can advance the teaching and learning of primary mathematics. Research by Robert Powers and William Blubaugh both from the University of Northern Colorado, on the impact of technology on mathematics education, shows that proper use of technology in class encourages students to engage with mathematics actively and as a result, it helps in developing their understanding of various concepts[7]. Moreover, with the current technological advancements and the usefulness of the subject in computing, there is a need to emphasize mathematics in elementary school and create a strong foundation.

1.4 The benefit of the proposed solution

The proposed solution is a tablet-based math app designed to help grade 1 and grade 2 pupils learn mathematics through game-like activities (building animal towers). Albert Sangra's research on the role of ICT in improving teaching and learning proves that its integration in mathematics has a positive impact in teaching and learning of the subject as it enhances interaction between learners and the technology itself [8].

The application will particularly focus on developing a coherent understanding of addition and subtraction. The app will incorporate game-like mathematics activities to promote

interactive and fun learning, thus creating a positive attitude towards the subject which is more likely to impact good results.

1.5 Related work

1.5.1 Matific Digital Resources

Matific is a digital platform that provides free math learning resources such as games suited for children from kindergarten to grade 8 [15]. The games are presented in the form of episodes suited for different levels, and by completing various activities on each episode, teachers can track students' progress to ensure that they are performing well and support on challenging topics.

The platform is widely used with over 15,000 teachers in more than 20 countries as of 2015 [15]. Matific's limitation is that its content is based on the US and UK math curriculum which may not appeal to a school that uses a different education system.

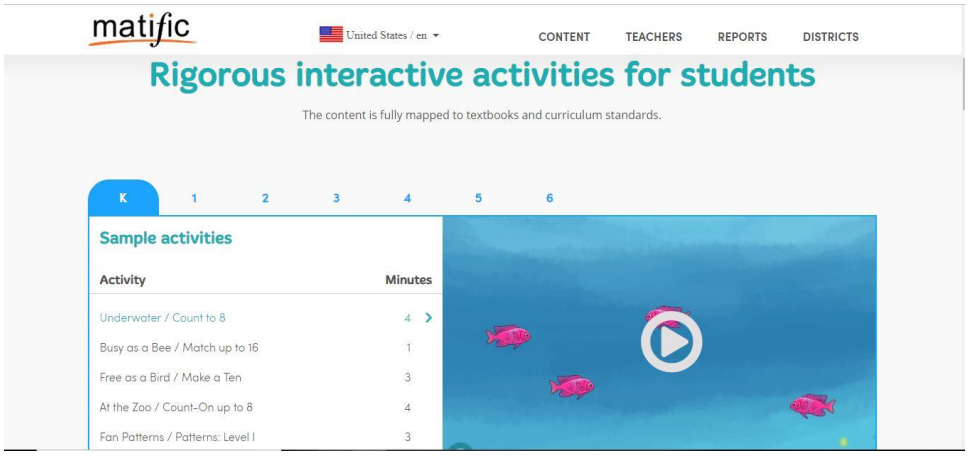

Figure 1.1: Matific website homepage

1.5.2 Early Math with Gracie & Friends

Early Math with Gracie & Friends is a math app designed to help preschool pupils master numbers and group objects through gaming activities. Early Math with Gracie and Friends was developed by Next Generation Preschool Math Project, to create and assess digital resources to support math learning[8]. The app focuses on only two concepts; subitizing (grouping of objects) and equipartitioning (grouping objects of the same classification)[11]. Most of the reviews about the app are positive because the app is accessible at first8studios.org at no cost. However, there have been a few cases whereby some users have raised a concern about the app focusing on only two math topics.

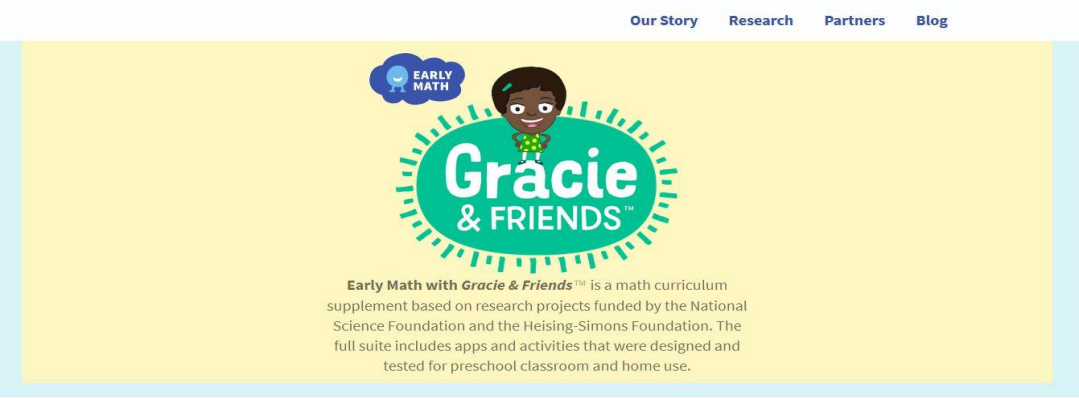

Figure 1.2: Gracie & Friends website homepage

1.6 Tablet-based math app

Tablet-based math app is designed to help children learn mathematics easily through the utilization of interactive activities like games and use of familiar objects for calculations as a complement to the standard textbooks. It aims to promote active learning of mathematics for children in grade 1 and grade 2.

1.6.1 Project Objectives

The project will research on the current computer/technology literacy levels among pupils in primary schools in Ghana. On-ground interviews will be conducted at Berekuso Basic School in Berekuso township which will be used as a case study.The project will first introduce students to learning mathematics using physical blocks for addition and subtraction before they can work with them on the app.

Research will be carried out to determine some of the materials or models that are used in Ghanaian classrooms to teach mathematics and how such can be integrated into the tablet math app.

The project will undertake the development of a tablet-based math app that will enhance teaching and learning to achieve improved academic performance in mathematics.

The app will teach students four main topics: concrete-fading (using objects to represent numbers), subitizing (grouping abjects of the same height), addition and subtraction

Upon completion and deployment of the app, it will be given to pupils for testing and how to use the app. Afterward, the app will be used to answer hypotheses questions for a thesis paper.

1.6.2 Functionalities

The app will be designed for pupils to learn mathematics. The pupils will have access to various game-like interactive activities for mathematics learning. The approach used to teach mathematics in the tablet-based math app is building towers to compute addition and subtraction. The towers are made of goats, spiders, and ants. The app will cover addition and subtraction of two digits and three digits. Each of the gaming activities has three levels. The

goats represent the hundreds place value; spiders are in the tens place value while ants are the ones in place value.

Login page. The login page will allow registered users to access the app content by providing a username and a password.

Concrete-fading. This refers to using goats, spiders or ants to represent a given number. Students will play games to build towers that reflect a given number.

Subitizing. Given a tower of a certain height, students will build towers of the same height to match it. This will help in the concept of carrying in addition.

Addition. The main goal of the project is that students will be able to do addition by building towers having learned concrete-fading and subitizing. To perform addition, students will build towers of the given values to produce a tower that represents the sum of the values.

Subtraction. Students will use the same concept of towers to do subtraction. The concept of finding a missing value given the first value and the answer will be used. Students will then be given a tower for the first value, a tower for the summation and will have to find the missing value by finding its tower height.

Feedback. Upon submission of the results, the user will get a notification for a correct or wrong answer submitted. If a wrong answer is submitted, the system will allow the user to redo the question until a correct answer is submitted or the user wishes to move to the next level

Chapter 2: Requirements Analysis

2.1 Users

The users of the system are pupils. Pupils will use the app to compute various concepts and play mathematics-oriented games for addition and subtraction.

2.2 Scope of the System

The tablet-based math app is designed to enhance a coherent, meaningful understanding and synthesis of mathematics knowledge and its concepts. Regarding that, the app will focus on the most basic mathematics fundamentals such as addition, subtraction, counting, and objects grouping. The app aims to promote a serene learning environment for math through a series of fun, interactive, and engaging game-like activities.

2.2.1 Use case Diagram

The use case diagram, figure 2.1 below is an overview of the users (pupils) of the system and the functionalities that the application would offer.

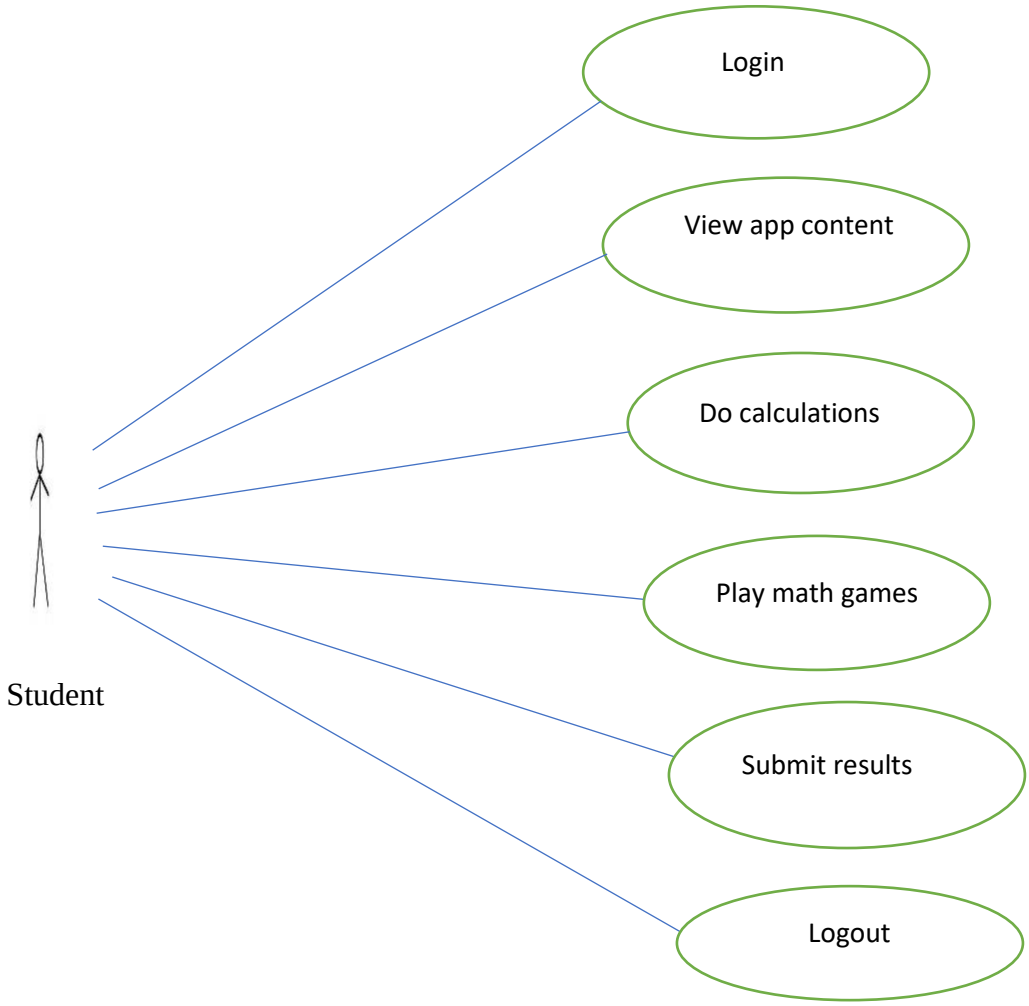

Figure 2.1: Use-case diagram

2.2.2 Sequence Diagram

The sequence diagram, figure 2.2 below shows the interaction between the system and the user' activities.

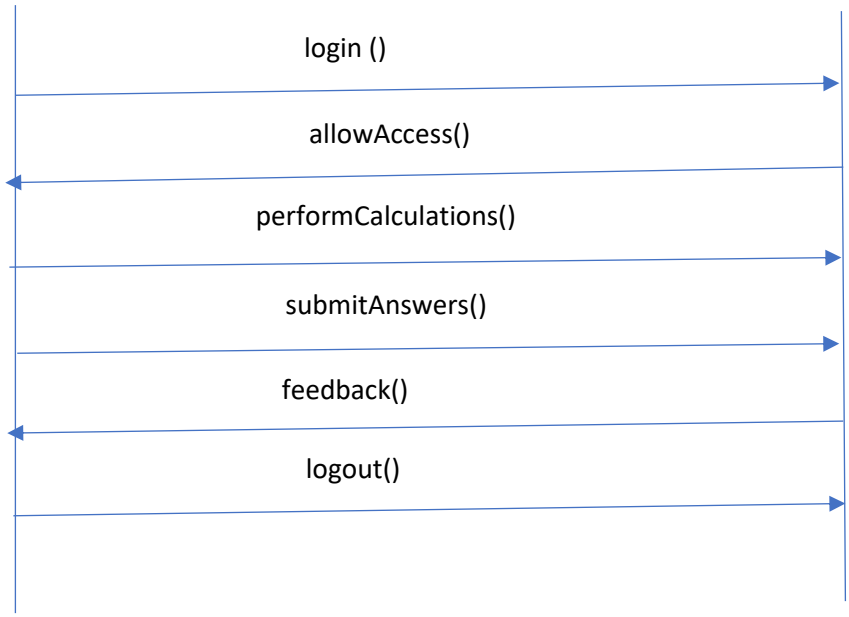

Figure 2.2: Sequence diagram

2.3 Functional Requirements

The requirements of the app were informed through a talk by Dr. Olaf, a Computer Science professor at St. Olaf College USA, themed "Building Educational Web Apps for Children" on 30th January 2018 at Ashesi University. Through his interaction with teachers and elementary school children from different schools in Northfield Minnesota, he was able to determine what features they needed in a math app designed to enhance conceptual understanding of mathematical concepts in both teaching and learning. He talked about the growth in the use of mobile devices in educational sectors and how this could transform the classrooms into digitalized learning.

Matific Digital Resources research report also addresses what most users stated as their requirements based on research conducted in 4 different schools; school A and B in Sydney

Australia, school C in Newcastle and school D in Central West (New South Wales). The researchers' main finding was that children wanted a learning tool that could guide them through the basic foundational concepts in Mathematics[16]. Further requirements analysis was undertaken at Berekuso Basic School in Ghana as well. This informed the researcher of similarities between requirements from various regions/continent. These functional and non-functional requirements were therefore extracted from these three sources: talk by Dr. Olaf, secondary research from Matific Digital Resources and Berekuso basic school students:

All users are registered by the administrator into the system manually for security purposes. After successful registration, the user can access the login page.

The system should let registered users login into their accounts. To log in, the user will provide a username and a password.

The system should enable users to perform concrete-fading activities (given a number, build towers to represent the number).

The system should enable users to perform subitizing (match towers of the same height).

The system should enable users to perform addition by building towers.

The system should enable users to perform subtraction by building towers.

The system should allow users to make changes by editing built towers before submission.

The system should allow users to submit the result for any activity performed on the site.

The system should provide feedback to the user on whether the submitted answer is correct or wrong.

The system should keep track of each user's activities by recording the logs into the server.

The system should provide more visual-based calculations, i.e., more images rather than theory to enhance better understanding of the concepts. Sean Whiteley emphasizes that it helps to think of math problems visually, using groups of objects as it improves memory [10].

The system should allow users to log out

2.4 Non-functional requirements

Quality content**:** the content provided on the app will be drawn from the Ghana curriculum recommended math textbooks to ensure that our material aligns with what is covered in the classrooms.

Accuracy: the app should provide accurate results, especially in calculations. Errors in calculations will be avoided by using mathematics libraries.

Usability: the app will have a user-friendly interface especially because the main target users are school children. Icons used will be easy to understand and should not require specific training; meaning navigation will be fairly easy for all users.

Security: the app will be well secured to ensure that user information is protected from unauthorized access. Only registered users can log in to the system. To submit an activity, you have to create an account.

Performance: the tablet-based math will be fast regarding performance when computing calculations; instant feedback will be provided.

Chapter 3: Design and Architecture

3.1 Introduction

The tablet-based math app is a mobile application focused on three main areas; login, math activities (concrete-fading, subitizing, addition and subtraction) and feedback.

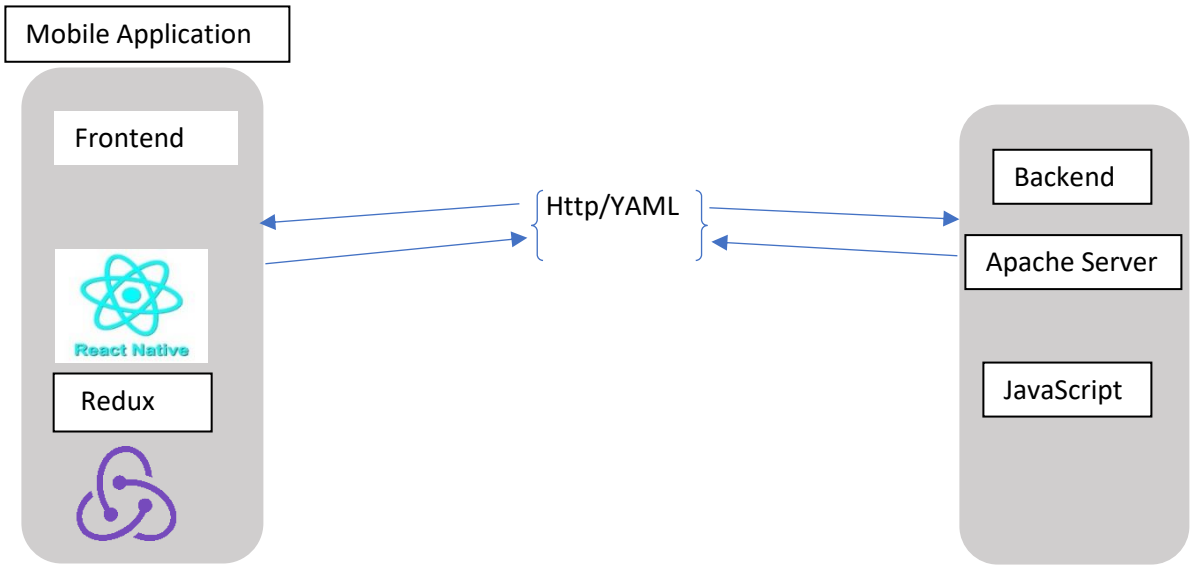

Figure 3.1: Programming Languages

The application will be built on Flux architecture. Flux is an application architecture for React and Redux utilizing a unidirectional data flow [17]. Flux when used with Redux, consists of a single object called a store that keeps the state or data of the application. The state is modified by dispatching user actions which are triggered by views. The reducer passes

current state to the store which updates the react native component for a new view to be rendered to the user, completing the cycle.

The figures below show the Flux design pattern:

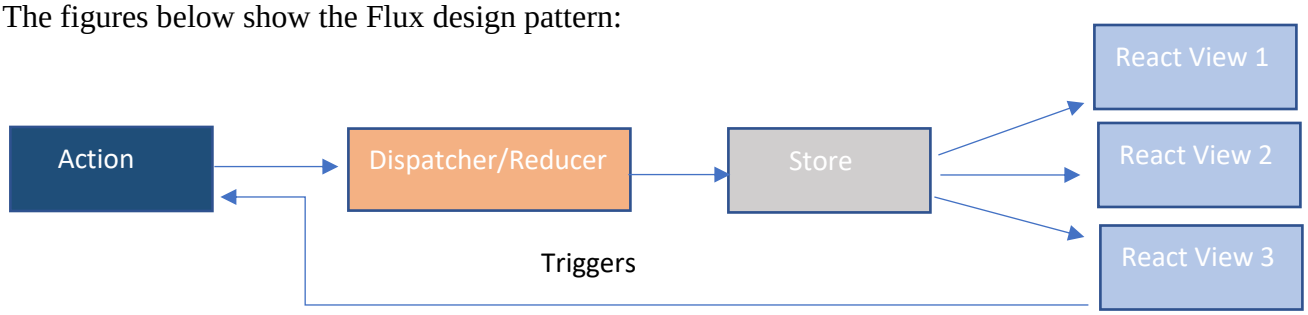

Figure 3.2: Flux Design

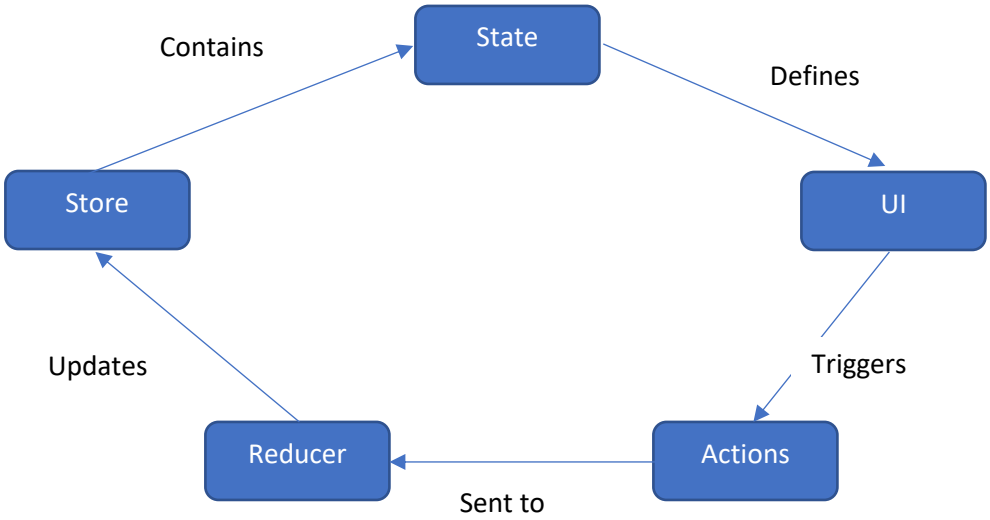

Figure 3.3: Unidirectional flow of data

3.2 Technologies Used for Development

3.2.1 React Native

The user interface will be developed using React Native. React Native is “a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android.”[3] React Native was born from React, “a JavaScript library for building user interfaces”[18]. The

developer decided to use React Native because it facilitates faster development of applications through code sharing for iOS and Android platforms.

3.2.2 Redux

Redux is “an open-source JavaScript library that controls an application’s state” [19]. It is also referred to as a data management library. In Redux, the direction of data flow is unidirectional, debugging is easy, and data is organized and managed in a simplified way[6].

There are three main pillars of Redux: a store which is a big object that keeps track of the application state, actions which are simple JavaScript functions that describe an activity as they transform the state of the application. Actions include a type property so that reducers know which action it is (once actions are dispatched, the reducers listen to them to know which state to modify)[20]. The third pillar is a reducer which is a function that updates the next state of an application based on the previous state and action as parameters.

Action ⟶ Reducer ⟶ Update Store

Figure 3.4 shows the interaction between React Native and Redux

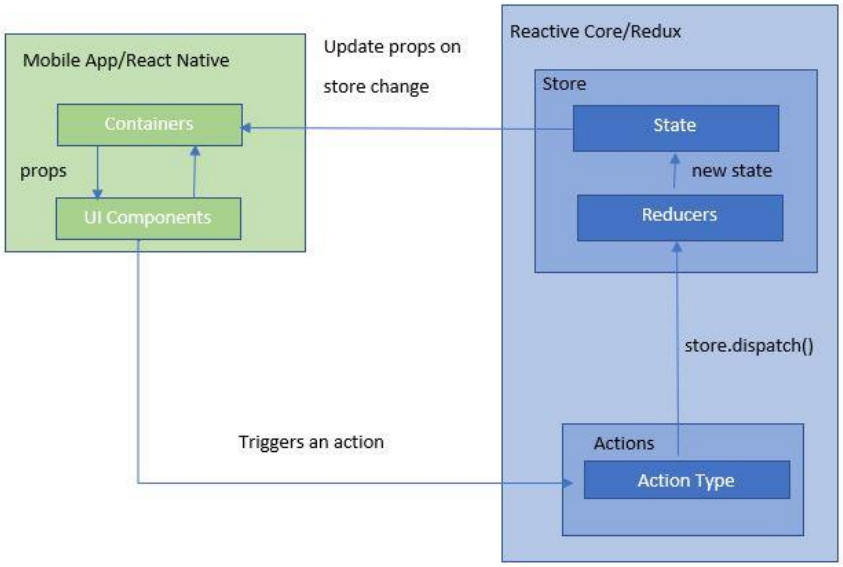

Figure 3.4: React Native & Redux Interaction

3.2.3 YAML

YAML is "a human-readable data serialization language" [21]. YAML is similar to JSON except that it includes indentation. The YAML file is read into a JavaScript data structure just like JSON. The developer is using YAML to display data on a browser or the application.

3.3 Activity Diagram

The activity diagram below shows an overview of all the activities that the user can do while interacting with the app.

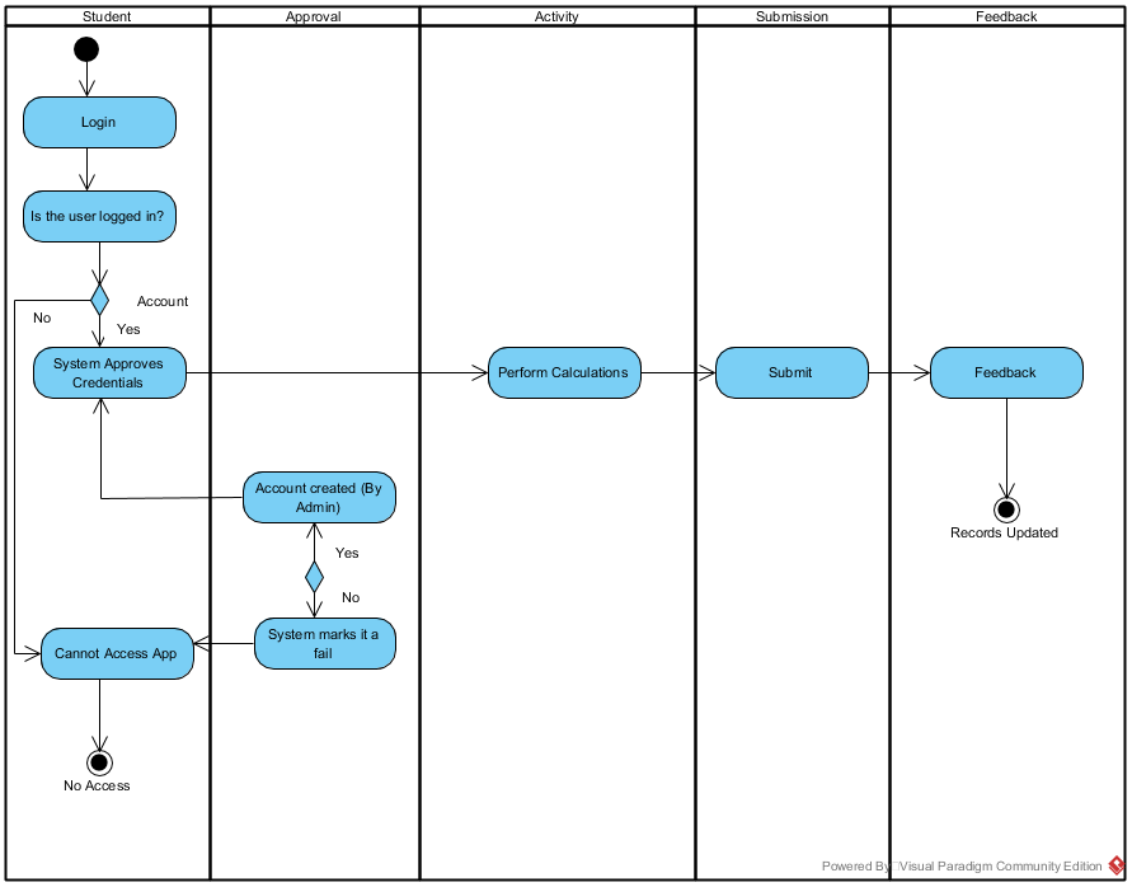

Figure 3.5: Activity Diagram

The user needs an account (created by the administrator) to access the app content. After login, the user can perform various math activities and submit answers for feedback. All submitted answers and calculations by each user are recorded in the server in a file system

3.4 System architecture

3.4.1 React + Redux: Architecture Overview [9]:

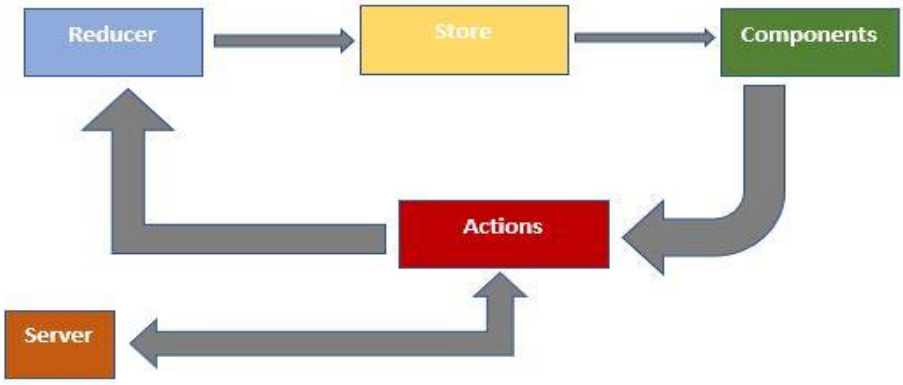

Figure 3.6: Architecture Overview

The components as shown in figure 3.6 above are React Native components or classes which render the application view. The actions make server requests and return responses from the server through AJAX. Actions communicate to the reducer which in turn updates the state of the store to be rendered for view via components.

To hook up React Native to Redux, react-redux library module is installed, and connection code written in the containers folder.

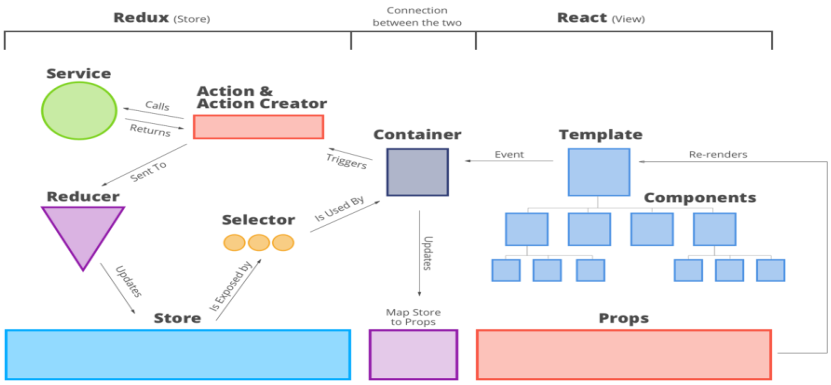

Figure 3.7: Comprehensive React Native-Redux Connection [9]

State is the data that keeps an application in a particular state or condition. In Redux, the store keeps the state while in React Native the state is in the component props. An event from the react native components triggers through the container and to the actions. The action gets data and sends it to the reducer which updates the store. The store is updated based on the data received.

The container, using the react-redux module, connects React Native with Redux. “It takes three arguments, an object that maps state to props - mapStateToProps, an object that maps actions to dispatch *(these are used to wire events to actions)* - mapActionsToDispatch, and *mergeProperties* merges all the properties and passes them to react for rendering” [9].

Chapter 4: Implementation

4.1 React Native

React Native is an open source JavaScript framework for front-end development[18]. React Native is recommended because of its cross-platform development advantages; an application is developed for several platforms (iOS and Android) at the same time [5]. This is made possible through platform code sharing or code reuse. It saves on development time, need for more development skills and the application is easy to maintain.

4.1.1 Installing Dependencies

The development Operating System (OS) is Windows, and the target OS is Android. However, the code can be reused and run on an iOS device. The dependencies required are "Node, a Java SE Development Kit (JDK), Python2, Android Studio and React Native command line interface" [22].

Node Package Manager (npm) which comes with Node enables one to install the React Native command line interface by running the command below:

npm install -g react-native-cli

To set up the Android development environment, Android Studio is installed, which by default comes with Android SDK. After all the dependencies are set, the developer is now ready to create a new React Native application via CLI (Command Line Interface) by running CLI commands.

The way React Native works is that the developer has a bunch of components and each component is described as a js (JavaScript) file.

The folder structure

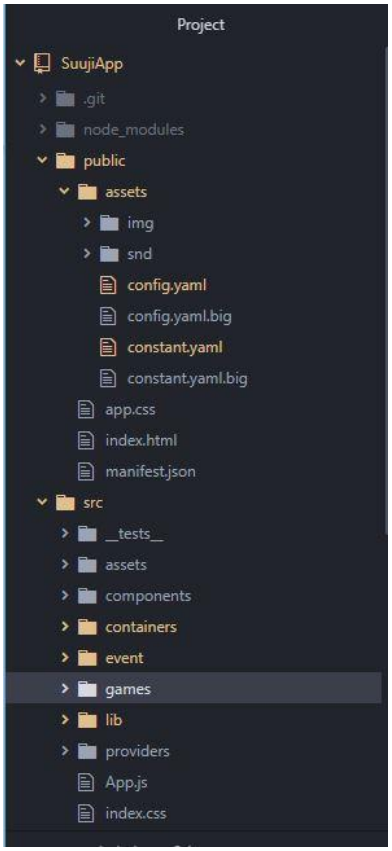

Figure 4.1: Folder structure

SuujiApp is the name of the application. The public folder has an assets folder which holds app (CSS), index(HTML), and manifest(JSON) files. Index(HTML) is the page template. Img folder contains all the images in the application while the snd folder contains the YAML files. Src (source) folder contains the application's code and also holds the index.js file which is the JavaScript entry file. Other folders inside src folder include _tests_, components, containers and providers. Tests folder contains all the tests written to test the application. Components folder has React Native components. A component makes a call to the container

which in turn makes a connection to the Redux store. Containers folder contains files that connect React Native components and Redux. Providers folder holds all the Redux files including, actions, reducers, and store.

Node modules folder is where all the libraries reside. Package(JSON) describes what the dependencies of the application are.

4.2 Installing Redux

Redux, a JavaScript library, is installed by running an npm package manager command.

npm install redux

The developer also needs React bindings to connect React Native and Redux. This is installed by running the command:

npm install react-redux

4.2.1 Redux

React components are connected to redux via containers. Redux can essentially push out props (property changes) to the container components thus giving access to the actions. Actions are simple functions that describe an activity, e.g., set the tower; they transform the state of the application. Components are centralized such that any component can trigger action but, once they send the action, they no longer have any responsibility for the resulting data. This is because the components are purely on the receiving end of any data transformations. The reducer is a function that updates the next state of an application by taking the previous state and an action.

4.3 Server

The app is hosted on Apache Server and all the data logs are also stored in the server.

4.4 Implementation Approach

To teach addition and subtraction, the concept of building towers is used; a tower height represents a number. To build the towers, three objects are used: goats, spiders, and ants. The three objects are of different sizes. The goat is bigger than a spider, and a spider is bigger than an ant, the ant being the smallest. The first step to multi-digit addition or subtraction is the place value knowledge. Therefore, each of the objects represents a place value. A goat being the biggest will occupy the hundreds place value, spiders will be the tens place value and ants will be the ones place value. At this stage, the students are learning addition of 2 digits and three digits only.

- Goats-represent hundreds
- Spiders -represent tens
- Ants -represent ones

To teach the concept of carrying:

- 10 spiders= 1 goat
- 10 ants = 1 spider

At times the user is presented with a fiver, which is a group of 5 spiders or 5 ants to ease the process of drag and drop.

The developer is using the Redux store for information about towers (a concept used in addition and subtraction) and all the other elements seen on the screen. In particular, there is a place in the redux store where the name of a tower is kept. A name might look like this: [2.0,

0.3]. The tower name is used by the num component to create everything that is needed on the screen for a tower with that name, including two goats and three spiders (items used to build the towers). To change the tower to have another spider, the value 0.3 increases to 0.4 and the tower name will now be [2.0, 0.4]. Changing the information in the redux store is accomplished via actions and action-creators. In this case, the setName action-creator is used to change the name of a given tower, like this: doAction.setName(id, name). When information in the redux store is changed, the components that are listening for changes in the redux store are notified, and they then re-render using the updated information.

Tower creation code(for display on the screen):

```
counting_up_sub:
  params:
    generate: # { tower_1_name: [pick_from_range, .1, 3.8, .1] }
      tower_1_height: [pick_from_range, 0.01, 1.995, 0.01]
      tower_2_height: [pick_from_range, 0.01, 2.995,0.01]
      total_height: tower_1_height + tower_2_height
    create:
      big_paren: true
      big_op: +
      keypad_kind: 'buildTower'
      tower_2_shadow: []
      tower_1_shadow: { name: tower_1_height }
```

Listing 4.1: Display tower code

Num component code(to name the component):

```
const hide_all_tower_names = true
        return (
                <View style={[styles.num, style, {left: position[0], bottom: position[1]}]}>
                        <Tower
                                anim_info={t_anim_info}
                                block_anim_info={block_anim_info}
```

```
                        block_opacity={block_opacity}
                        id={id}
                        just_grey={just_grey}
                        misc={misc}
                        name={name}
                        position={position}
                        scale_factor={scale_factor}
                        style={tower_style}
                    />
                    {hide_all_tower_names ? null : tn}
                    {tm}
                    {tvi}
                </View>
        )
```

Listing 4.2: React native component

Code snippet to set tower name:

```
export function setName(id, name) {
    return {type: AT.SET_NAME, id, name: fromJS(name)}
}
```

Listing 4.3: Set tower name code

Reducer updates the store function code(name of the tower):

```
function name(state = Map({}), action) {
    switch (action.type) {
        case AT.TOWER_CREATE:
        case AT.TILE_CREATE:
        case AT.DOOR_CREATE:
        case AT.PORTAL_CREATE:
        case AT.FIVE_FRAME_CREATE:
        case AT.BAR_CREATE:
        case AT.SET_NAME:
            return obj_add_remove_property(state, action.id, action.name)
        case AT.TOWER_DELETE:
        case AT.TILE_DELETE:
        case AT.DOOR_DELETE:
        case AT.PORTAL_DELETE:
        case AT.FIVE_FRAME_DELETE:
        case AT.BAR_DELETE:
            return obj_add_remove_property(state, action.id, null)
        case AT.TOWER_ADD_BLOCK:
            return state.set(
                action.id,
```

```
        add_block_to_name(action.size,action.is_fiver,state.get(action.id))
,)
                case AT.TOWER_REMOVE_BLOCK
                        return                                  state.set(action.id,
remove_block_from_name(state.get(action.id)))
                default:
                        return state
        }
}
```
Listing 4.4: Reducer function

4.4.1 Tower Implementation

To implement a tower, a component called the Tower is created. A class called Tower is also created. The tower component triggers an action function called towerCreate, and the towerCreate has an action name and position to the name and position reducer functions. And because Tower is the container component, the created tower props will be automatically updated when the reducer is modified, and Tower component will get these new props to be able to render an updated view, which is the creation of a new tower. All the asynchronous activity is happening in the actions, and everything else in the application is happening unidirectionally which means that, Tower component will behave predictably and that all of the actions or all of the UI will happen in a proper sequence.

4.5 App Implementation

4.5.1 Concreteness Fading

Concreteness fading is the introductory task into the entire addition concept. A student is given a number, and the task is to show animals that correspond to that number by building a tower as shown in the figures below. To complete the task, a student clicks the start button on the first screen. The number is then displayed and the items to choose from. The user selects the items by clicking any of the given items in the top left side. The tower is built from left to

right. When done, the user clicks the submit button for submission and can also click the delete button to undo a selection.

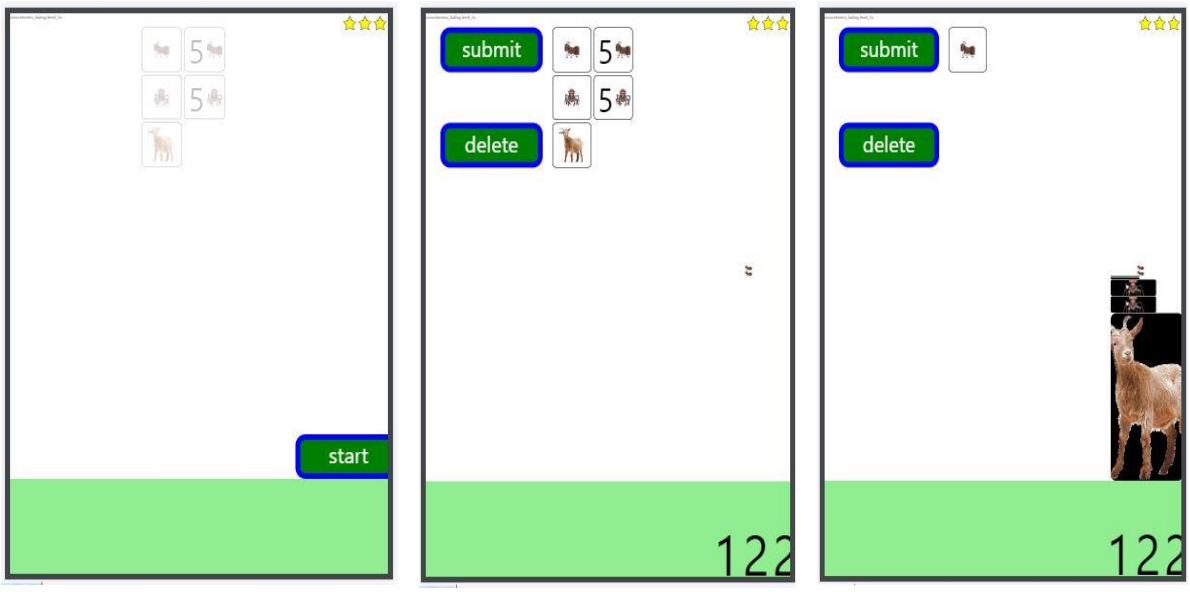

Figure 4.5: Concrete-fading

Figure 4.5 above shows a task; a student is given the number 122 and expected to represent the number with the animals from the choices given. This translates to one goat (hundreds place value), two spiders (tens place value) and two ants (ones place value).

A code snippet of the functionality is shown below:

```
concreteness_fading:
  params:
    generate:
      tower_1_height: [pick_from_range, 0.01, 3.505, 0.01]
    create:
      keypad_kind: 'buildTower'
      tower_1: { name: tower_1_height }
      tower_result: []
      tile_success: { name: emoji_smile }
      tile_fail: { name: emoji_frown }
    modify:
      tower_1:
        position: [595, 0]
        tower_style: { opacity: 0
```

```
 misc:
backgroundColor: 'black'
```

Listing 4.6: Concrete-fading code

4.5.2.Subitizing (Hidden Tower)

Subitizing is a way of grouping items. The students get to learn how to produce towers of the same or matching height as the one in question. As children learn to group items of a given height, then it will make it easy for them to measure heights for the next addition concept.

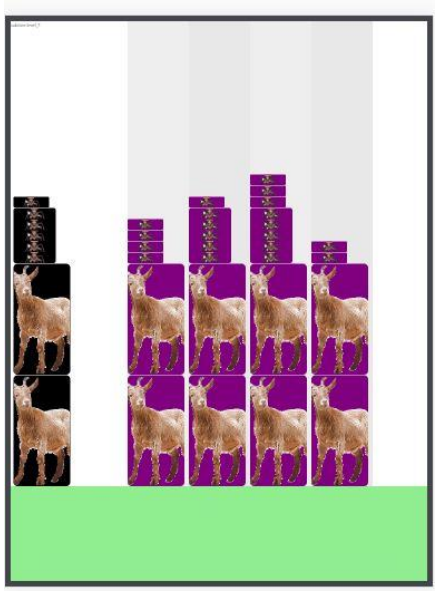

Figure 4.7: Subitizing

In this task, the student is expected to select a tower which has a matching height as the one given (black tower). The matching tower from the choices given (purple towers) would be the second tower. The student clicks on the matching height to make a selection.

Subitizing code snippet:

```
subitize:
  params:
    generate:
      num_animals: [pick_from_range, 0.3, 3.9, 0.1]
      option_value_delta: 0.
```

```
  option_value_seed: num_animals
  restriction_3: option > 0
  restriction_4: option < 4
  m1: num_animal
create:
  tower_1: { name: m1 }
  tower_2: { name: 1 }
modify:
  tower_1:
    position: [15, 0]
    misc:
      hide_tower_number: true
      backgroundColor: 'black'
  tower_2:
    position: [0, 0]
    misc:
      is_option: true
      option_offset: 220
      option_width: 110
      backgroundColor: 'purple'
event_handling:
  correctness: identical
  target: tower_2
  arg_1: tower_1
  comparison_source: tower_1
  disappearing_object: tower_1
misc:
  new_scale_factor: 200
  config_iteration: 5
```

Listing 4.8: Subitize code

4.5.3 Addition

To calculate the sum of 2 numbers, each of the given numbers is represented as tower height; the first number is represented in the blue tower while the second number is the red tower. The two towers combined form the mixed tower (faded). To get the results of the summation, the user is required to build a black tower of the same height as the mixed height tower. However, to build the black tower, the user is provided with a limited number of items which means he or she has to be creative about using some animals as a representation of the

other. For example, ten spiders can be represented by one goat, a concept that introduces the user to the carry operation in Maths. The black tower is, therefore, the result of the sum as shown in the figures below.

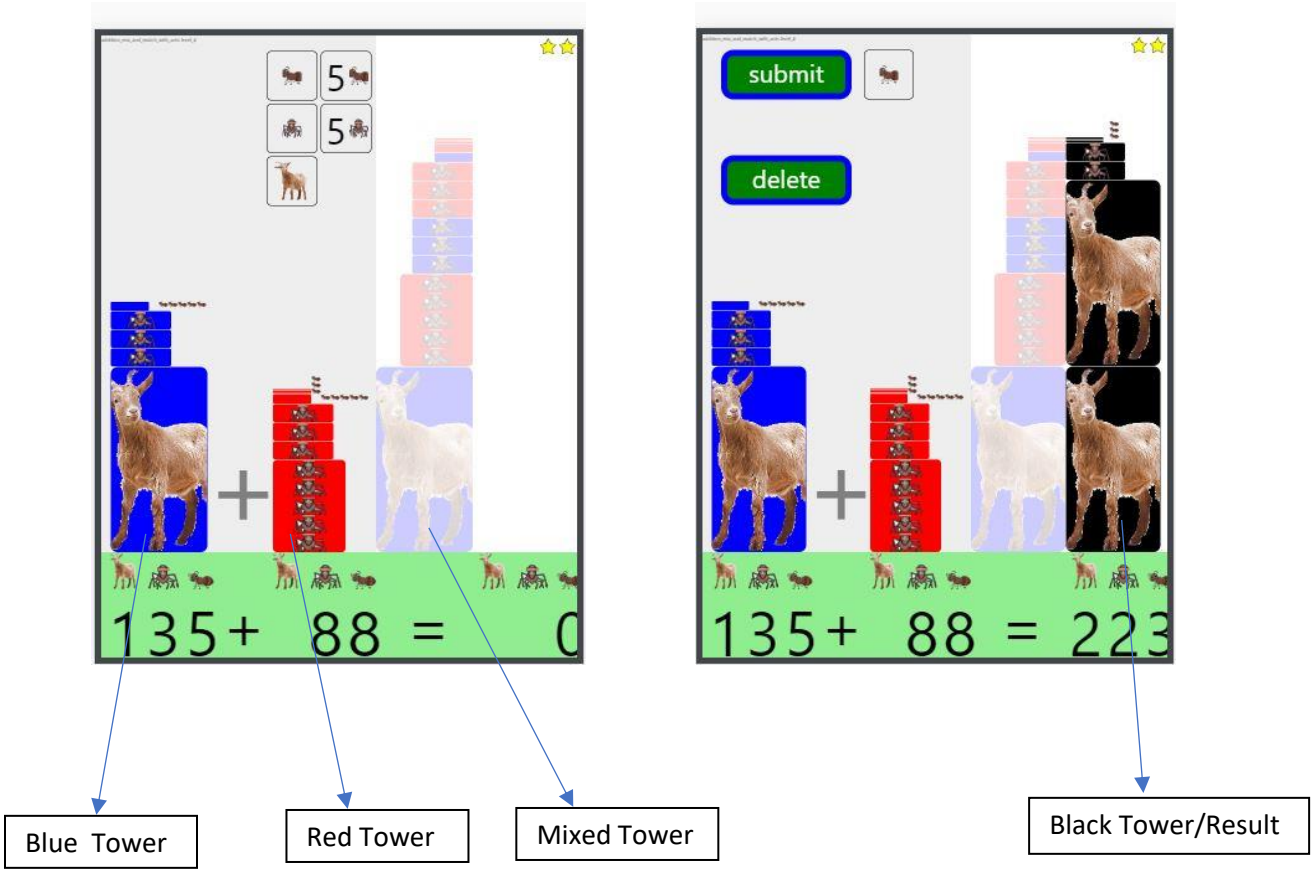

Figure 4.9: Addition

In figure 4.9, the first number is 135; in terms of place value, one is the hundreds, three is the tens and five is the ones. This translates to one goat, three spiders, and five ants respectively. The second number is 88; the first 8 is the tens, and the other 8 is ones. The mixed tower shows both the blue tower and the red tower stacked together. Having been given the mixed tower, the user has to build a black tower of the same height as the mixed tower given a limited number of the items. In this case, the mixed tower contains (1 goat, 11 spiders, and 13 ants). To get the black tower from the mixed tower, 1 goat can be aligned with another goat, of

the 11 spiders, 10 spiders can be represented by 1 goat, then add 1 more spider to meet the the total number, of the 13 ants, 10 ants can be represented by 1 spider and then add 3 more ants to meet the total number. The black tower thus represents the summation of the two numbers; 2 goats, two spiders, and three ants, which correspond to the number 223, the sum.

Code snippet:

```
addition_test:
  params:
    generate:
      tower_1_height: [pick_from_range, 0.01, 1.995, 0.01]
      tower_2_height: [pick_from_range, 0.01, 1.995, 0.01]
    create:
      tower_1: { name: tower_1_height }
      tower_2: { name: tower_2_height }
      tower_result: []
      arith_symbol: +
      equal_symbol: =
    modify:
      tower_1:
        position: [15, 0]
        tower_style: { opacity: 0 }
      tower_2:
        position: [230, 0]
        tower_style: { opacity: 0 }
      tower_result:
        position: [545, 0]
        tower_style: {opacity: 0}
        misc:
          backgroundColor: 'black'
```

Listing 4.9.1: Addition code

4.5.4 Subtraction (Missing Addend)

To perform the subtraction of numbers, the missing value concept is used with towers. The user is given the first value, the summation value of two numbers and the task is to find the missing second value.

The figure below shows the task in question:

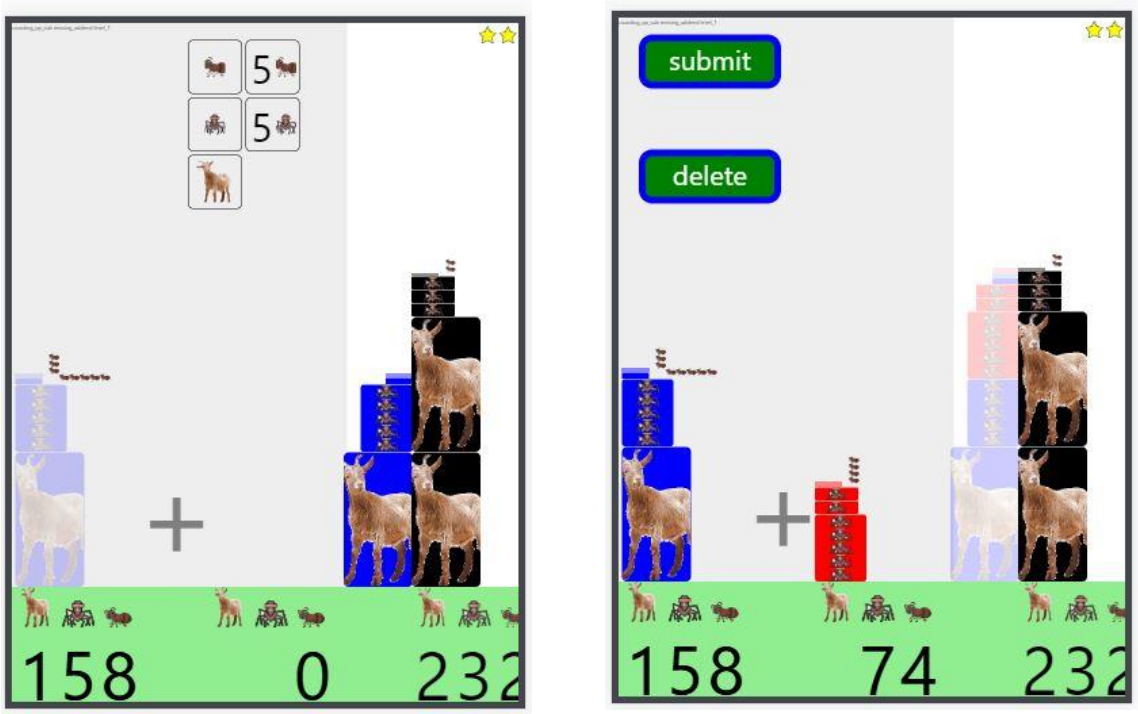

Figure 4.9.2: Subtraction

Given the first number (158) represented by the faded blue tower, the task is to find the second number which will be represented by a red tower. To find the red tower, the user has to build the missing middle tower by selecting items from the given choices. The goal is to create a mixed tower that is of the same height as the black tower which represents the summation of the two numbers. By creating the mixed tower (the blue tower and red tower stuck together), the user can get the red tower. This approach is the same as subtracting 158 from 232.

4.5 Login Module

The login page allows registered users (registration set up is done by the administrator from the server setup) to access the platform. When a user inputs the login details, the system performs a backend check to determine if the user is already registered. A user is registered if his or her details correspond to what exists in the database. After a successful login, the system displays the activities for the user to select what to do.

Chapter 5: Testing, Results, and Conclusion

5.1 Testing Overview

Testing is done to ensure that the written program meets all the requirements. Four levels of testing are carried out: unit testing, component/integration testing, system testing, and user testing. Unit testing is done to verify that each functionality or unit of code works as expected. Component testing, also called integration testing, is done to verify that a function that relies on another function or several individual units that are integrated meet the expectations. In system testing, the entire system is tested as a whole. User testing involves getting the actual users of the system use the system to ensure that they can navigate through and it meets all their requirements.

The developer decided to use Jest, a JavaScript testing framework used to test React Native and React applications [23]. Jest can also be used to test other JavaScript frameworks and libraries such as Vue, Angular, AngularJS, MobX, Express, and Redux [24]. Jest takes the test files, runs the test functions and then based on the assertions if they are true or correct, the test is a pass. The test does not necessarily have to be true; it could also be false. The assertion is basically that something is one way and if it is not that way it is going to fail the test.

Jest setup is automatically installed by running the command *react-native init* (to create a project) [4].

5.1.1 Unit Testing

To perform unit testing, test functions are implemented to test individual components. Figure 5.1 below tests whether the submit button renders correctly without an error while Figure 5.2 shows the results of the test which returns a pass.

```
16  describe('SubmitButton', () => {
17    it('renders correctly', () => {
18      const wrapper = shallow(<Button />);
19      expect(wrapper).toMatchSnapshot();
20      // On the first run of this test, Jest will generate a snapshot file automatica
21    });
22  });
```

Figure 5.1: Unit test for submit button

```
Study@Nanis MINGW64 ~/OneDrive - Ashesi University/4th Year-Sem 2/Capstone/Su
App (master)
$ npm test test.js

> Suuji_app@0.1.0 test C:\Users\Study\OneDrive - Ashesi University\4th Year-S
2\Capstone\SuujiApp
> jest "test.js"

PASS src/__tests__/test.js (23.393s)
  ○ skipped 1 test
  SubmitButton
    √ submit button renders correctly (4ms)

Test Suites: 1 passed, 1 total
Tests:       1 skipped, 1 passed, 2 total
Snapshots:   0 total
Time:        59.911s
Ran all test suites matching /test.js/i.
```

Figure 5.2: Unit test results

5.1.2 Component/Integration Testing

In component testing, functions that depend on each other are tested. In this case, the developer will test the integration of Redux store functions, the reducers, actions and React Native components. For instance, the developer will test the redux store function that creates a new tower and renders it to the tower component. Changing the information in the Redux store is accomplished via actions and action-creators. In this case, the towerCreate action-creator is used to create a new tower given the new tower parameters such as id, name, and position. When

information is changed in the store, the tower components that is listening for changes in the Redux store via the reducer function TOWER_CREATE is notified, and then re-renders the updated information. The communication between all these components will be tested to ensure that it outputs a new tower. Figure 5.3 below shows the component test for different functions called to build a tower to perform a calculation. The towerCreate function takes the new tower id, name, and the position.

```
describe('towerCreate', () => {
  it('Tower created', () => {
    const id = 1,
    name: fromJS(name),
    position: fromJS(position),
    const expectedAction = {
      type: types.TOWER_CREATE,
      id, name, position
    }
    expect(actions.towerCreate(id, name, position)).toEqual(expectedAction)
  })
```

Figure 5.3: Component test for creating a tower

```
Study@Nanis MINGW64 ~/OneDrive - Ashesi University/4th Year-Sem 2/Capstone/Suu
App (master)
$ npm test test.js

> Suuji_app@0.1.0 test C:\Users\Study\OneDrive - Ashesi University\4th Year-S
2\Capstone\SuujiApp
> jest "test.js"

PASS src/__tests__/test.js (15.703s)
  ○ skipped 1 test
  towerCreate
    √ Tower created (19ms)

Test Suites: 1 passed, 1 total
Tests:       1 skipped, 1 passed, 2 total
Snapshots:   0 total
Time:        19.986s
Ran all test suites matching /test.js/i.
```

Figure 5.4: Component test results

Appendix A also shows component testing of adding an animal item when creating a tower.

5.1.3 System Testing

In system testing, the entire system is tested to ensure that it is fully functional and that all the requirements are met. The main requirements tested include the ability of the system to let a user log in, perform calculations for various gaming activities, submit the answer and get feedback for the submitted results.The system successfully allows registered users to log in. After logging in, a user can do a range of calculations ranging from concrete-fading at various levels, subitizing levels, addition, and subtraction. Upon submission of an answer, the system responds accordingly on whether the submitted answer is correct or not. For a correct answer, the system allows the user to move on to the next level. If a user submits a wrong answer, a frowning emoji appears, and the system gives the user to redo the activity.

Table 1.1: Table of tested requirements

Requirement	**Implemented or Not Implemented**	**Test Result**
Concrete-fading activities (given a number, build towers to represent the number)	Implemented	Passed
Subitizing (match towers of the same height).	Implemented	Passed
Addition by building towers	Implemented	Passed
Subtraction by building towers	Implemented	Passed
Edit/adjust tower height	Implemented	Passed
Submit completed activity	Implemented	Passed
Provide feedback to the user on whether the submitted answer is correct or wrong.	Implemented	Passed

5.1.4 User Testing

The users of the application are grade 1 and grade 2 students. Students from Berekuso Basic school in the classes above used the app to do various math problems; subitizing, addition and subtraction. They were being observed to ensure that the system fully works.

Figure 5.5: Children using the app on tablets

5.2 Project Overview

This project sought to develop a tablet-based math learning application for primary school children in grade 1 and grade 2. The app introduces children to basic Mathematical concepts: addition, subtraction, and grouping of objects by building towers. The towers represent the values to be added or subtracted, and students can perform various computations. The most difficult aspect of numbers for grade one and two students is mainly carrying for the addition of numbers.

The developed system simplifies this aspect through towers; using objects to represent another (e.g., ten spiders are represented by one goat). The gaming approach makes Maths learning fun and engaging especially for children who love to be involved and enjoy the process.

Familiar objects (goats, spiders, and ants) are used to build the towers, and this makes the concepts simpler, and the children can easily relate to the concepts of counting using objects as well.

5.3 Limitations

When a user submits an incorrect answer for a sum, the application does not give the user the option to continue to the next level. It requires the user to perform the same calculation until a correct answer is submitted.

The system does not also provide the user with a summary of performed tasks as they are all saved in the server. Moreover, a user cannot register into the system; users can only be registered by the administrator, which limits access to the system.

5.4 Challenges

I started developing this application using React Native and Laravel; a PHP framework. However, I changed to using React Native and Redux because I had challenges connecting React Native and Laravel. It took me some time to switch and catch up because this also meant making changes to my write-up.

5.5 Future Work

Based on the limitations mentioned above, the application could be improved to accommodate more functionalities. The application can allow users to create their accounts

without the administrator setting it up. The system could also accommodate teachers as users so that they can keep track of students performance. Having teachers as users would also make the application a teaching tool that can use to supplement the textbooks. The system could also separate the content for different literacy levels. The user can be classified at various levels to enhance productivity and facilitate performance tracking.

5.6 Conclusion

Mathematics is a compulsory subject in pre-university levels of education in the Ghanaian system. It is a foundational subject that cuts across all disciplines, and in this era of technological advancement, the usefulness of the subject in computing, problem-solving and decision-making is immeasurable. However, it is also one of the subjects that most students tend to perform poorly at and end up not pursuing math-related courses in higher levels of education. Poor performance and declining engagement of the subject to students discourages some of them, and as a result, they go for theoretical courses such as sociology and psychology. Poor performance is attributed to the approach of teaching Mathematics. Due to insufficient teaching resources, in most public schools, the curriculum textbook is usually the only material that teachers have access to, and, in most cases, the number of textbooks is also less than the number of students who need them. Lack of supplementary materials to enhance the learning of Mathematics and students not being able to explore the practicality of the subject are the main causes of declining performance and interest in Mathematics.

The tablet-based math application contributes to the improvement of Mathematics education by providing a game-like technological approach to learning basic Maths concepts at the foundation levels (grade 1 and grade 2) of schooling.

References

[1] Douglas Agyei. 2013. Analysis of Technology Integration in Teacher Education in Ghana. *Journal of Global Initiatives: Policy, Pedagogy, Perspective* 8, (January 2013), 69-86.

[2] Godwin Akweiteh Allotey. 2016. Improving maths and science education in Ghana [Article]. *citifmonline.com*. Retrieved October 26, 2018 from http://citifmonline.com/2016/02/16/improving-maths-and-science-education-in-ghana-article/

[3] Bonnie Eisenman. 1. What Is React Native? - Learning React Native [Book]. Retrieved February 2, 2019 from https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html

[4] Jason Gaare. 2017. Learning to test React Native with Jest — part 1. *React Native Training*. Retrieved March 30, 2019 from https://medium.com/react-native-training/learning-to-test-react-native-with-jest-part-1-f782c4e30101

[5] Niclas Hansson and Tomas Vidhall. 2016. *Effects on performance and usability for cross-platform application development using React Native*. Retrieved February 13, 2019 from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-130022

[6] Abhinav Jain. 2017. React + Redux Architecture: Separation of Concerns. *freeCodeCamp.org*. Retrieved March 6, 2019 from https://medium.freecodecamp.org/react-redux-architecture-part-1-separation-of-concerns-812da3b08b46

[7] D.K Mereku. 2010. Five Decades of School Mathematics in Ghana. 9, 1 (2010). DOI:https://doi.org/10.4314/mc.v9i1.61558

[8] Jillian Orr, Louise Flannery, Ashley Lewis Presser, Phil Vahey, and Sonja Latimore. 2015. Early Math with Gracie & Friends™ Demo: App-infused Curriculum and Teacher Support for Preschool. In *Proceedings of the 14th International Conference on Interaction Design and Children* (IDC '15), 458–461. DOI:https://doi.org/10.1145/2771839.2771885

[9] Colt Pini. 2016. React + Redux: Architecture Overview. *MoFed*. Retrieved March 12, 2019 from https://medium.com/mofed/react-redux-architecture-overview-7b3e52004b6e

[10] Sean Whiteley. Using Visualization for Learning - by Sean Whiteley. Retrieved January 22, 2019 from https://trans4mind.com/counterpoint/index-creativity-career/whiteley.shtml

[11] Christine Zanchi, Ashley Lewis Presser, and Phil Vahey. 2013. Next Generation Preschool Math Demo: Tablet Games for Preschool Classrooms. In *Proceedings of the 12th International Conference on Interaction Design and Children* (IDC '13), 527–530. DOI:https://doi.org/10.1145/2485760.2485857

[12] International Bureau of Education. *International Bureau of Education*. Retrieved October 24, 2018 from http://www.ibe.unesco.org/en

[13] WAEC Ghana. Retrieved October 8, 2018 from https://www.waecgh.org/

[14] WAEC bemoans poor performance in Maths - Ghanaian Times. Retrieved September 30, 2018 from http://www.ghanaiantimes.com.gh/waec-bemoans-poor-performance-in-maths/

[15] Matific. Retrieved October 26, 2018 from https://www.matific.com/us/en-us/home/

[16] Matific. Retrieved January 15, 2019 from https://www.matific.com/us/en-us/home/

[17] Flux | Application Architecture for Building User Interfaces. Retrieved March 18, 2019 from http://facebook.github.io/flux/index.html

[18] React – A JavaScript library for building user interfaces. Retrieved March 29, 2019 from https://reactjs.org/

[19] Redux · A Predictable State Container for JS Apps. Retrieved March 6, 2019 from https://redux.js.org/index.html

[20] Actions · Redux. Retrieved March 16, 2019 from https://redux.js.org/index.html

[21] The Official YAML Web Site. Retrieved March 6, 2019 from https://yaml.org/

[22] Getting Started · React Native. Retrieved February 12, 2019 from https://facebook.github.io/react-native/index.html

[23] Testing React Native Apps · Jest. Retrieved March 11, 2019 from https://jestjs.io/index.html

[24] Testing Web Frameworks · Jest. Retrieved March 11, 2019 from https://jestjs.io/index.html

Appendices

Appendix A: Component test to add block animal info for a tower

```
describe('addBlockAnimInfo'),() => {
  it('should return the initial state of the tower created'),() => {
    expect(reducer.addAnimInfo(id, anim_info),{}).toEqual(null)
  }),
  it('should handle ADD_BLOCK_ANIM_INFO'),()=>{
    expect(
      reducer.addAnimInfo(
        undefined,
        {
          addAnimInfo:addAnimInfo(id, anim_info),{}),
          type:addBlockAnimInfo.ADD_BLOCK_ANIM_INFO
        }
      )
    ).toEqual(addBlockAnimInfo)
  })
})
```

Appendix B: Add block animal test results

```
Study@Nanis MINGW64 ~/OneDrive - Ashesi University/4th Year-Sem
2/Capstone/SuujiApp (master)
$ npm test test.js

> Suuji_app@0.1.0 test C:\Users\Study\OneDrive - Ashesi University\4th Year-Sem
2\Capstone\SuujiApp
> jest "test.js"

PASS src/__tests__/test.js (18.246s)
  ○ skipped 1 test
  addBlockAnimInfo
```

```
  √ should return the initial state of the tower created (9ms)
Test Suites: 1 passed, 1 total
Tests:       1 skipped, 1 passed, 2 total
Snapshots:   0 total
Time:        43.07s
Ran all test suites matching /test.js/i.
```