Verb Ventures

Software Requirements Specifications

Nathan Glover
Drew Nelson
Jacob Pohl
Ben Traffanstedt

Revision History

Version	Summary	Author	Date
0.1	Title Page, Introduction, and Outline	Everyone	9/21/17
0.2	Formatting, Functional Requirements	Nathan Glover	9/22/17
0.3	Formatting, Overall Description, Non-functional Requirements	Jacob Pohl	9/24/17
0.4	Completed Functional Requirements	Nathan / Drew	10/2/17
0.5	Formatting, Completed Overall Description	Jacob Pohl	10/2/17
0.6	Diagrams, Diagrams, Diagrams	Ben/Drew/Jacob	10/2/17
0.7	Expanded goals and scope, added features	Drew	10/3/17
0.8	Defined new features, updated contents	Jacob/Nathan	10/3/17
0.9	Create new activity diagrams for features	Ben	10/3/17
1.0	Updated use case overview, format finalized	Drew	10/3/17
1.1	Added Design Diagrams to Table of Contents	Nathan	10/26/17
1.2	Added Database Design Diagram	Nathan	10/26/17
1.3	Added Class Design Diagram	Drew	10/28/17
1.4	Added Sequence Diagrams 6.3.7 - 6.3.13	Jacob	10/30/17
1.5	Added Database Diagram Description	Nathan	10/30/17
1.6	Added Class Diagram Description	Drew	10/30/17
1.7	Added Sequence Diagrams 6.3.1 - 6.3.6	Ben	10/30/17

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this project is to create an app to help teach the meanings of verbs with animations for children with learning/speech disabilities. Speech and language therapists, special education teachers, and caregivers of those with language deficits will be able to use the app to work with their clients. Children will also be able to use the app on their own to practice learning verbs. Currently, only three therapeutic tools work on aiding with the comprehension of verbs. Only two of these three use animations. These competitors are costly and only work on desktops. We have the desire to make the app cost less than competitors, have better teaching methods than competitors by focusing on animations, and be able to be used on a mobile device.

1.2 Scope

This project is set out to be a base implementation and proof of concept for what will potentially be a published Android application. We plan to build the features that are necessary to create and play verb animation packs, and will try to add in other features as time allows. Our contacts and the owners of this IP, Vanessa Parkinson and Jessie Shares, have already shown that there is a need and want us to show it can be done.

Sequentially, our scope could reach into the following areas:

- Show initial basic functionality to provide a base for further expansion of functions.
- Show that the UI can easily be used by the target users and that it tailors to their specific needs.
- Show further implementation of more complex functions in an attempt to reach the requirements of our contacts for the initial published version of the application.
- Show that the application actually improves learning, provide examples of use from real world situations.

1.3 Goals

Our goals for this project include testing and teaching the comprehension of appropriate verbs to users, presenting verbs in a way that enables accurate understanding by using animations and multiple representations, providing a way for tutors and therapists to track a student's progress, and creating an interface that can be easily used by children with learning disabilities.

Our specific detailed goals are:

- Create the initial basic functionality of the app including:
  - Admin ability to create and manage users
  - Admin ability to create and manage verb packs
  - Admin ability to see results and statistics for all students
  - Student ability to play a verb pack
  - Student ability to see personal results
- Create a UI that caters to the needs of our audience, including:
  - Appealing colors for children
  - Easy navigation, large buttons and text
  - Design that fosters intuitive learning
- Ensure security of sensitive data including:
  - Names of students created by an admin
  - Scores and results of students
- Expand functionality into lower priority features including:
  - Admin ability to create their own verb-animation pairs
  - Admin ability to view the ongoing progress of students in verb packs
  - Admin ability to set target goals for students
  - Student ability to see completion percentage of individual verb packs
  - Automatic creation of verb pack with words answered incorrectly
  - Feedback given to the student on what to work on
  - Allow students to play at home on personal devices
  - Offline function capabilities.

1.4 Definitions

Business Domain:
Verb-animations pairs- One verb a set of animations depicting the action.

Verb Pack - A set of verb-animation pairs compiled together by some common characteristic.

Student - The person who will use the app to learn verbs to help with that person's speech / learning disabilities.

Admin - The speech therapist, teacher, or caregiver, who has the ability to add students, create verb packs, and view the progress of a student.

2. Overall Description

2.1 General Process Flow

The general process flow of our application is fairly straightforward. Added students will be able to find and click their name, select a verb pack to play/test, and then view their results. The student can then go back to select another verb pack or sign out and let another student go through the same process. Admins will be able to see the results of all students in a similarly straightforward fashion.

2.2 Admin Creation Activities

Admins in charge of students will have the ability to add and remove students as members of the app and assign verb packs to students individually. Admins will also be able to create verb packs for students to test with and add individual verb-animation pairs. Admins will also be able to add students to the session before starting the play action.

2.3 Admin Progress Tracking Activities

Admins will have the ability to look at student's statistics so that they can track an individual student's progress. This will allow the admin to determine how well the student is learning. Admins will also be able to set target goals for individual students.

2.4 Student Testing Activities

The student will be able to select their name from a list of students, see verb packs that they can choose to test from, and play/test a verb pack. After the student has tested with the verb pack, their results will be shown.

2.5 Settings/Accessibility

Users will have the ability to modify settings in accordance to their needs. Some of these modifiable settings include adjusting field size and determining if old scores or other aspects are displayed to the screen.

3. Functional Requirements

3.1 Admin Functional Requirements

3.1.1 Add Student - *High Priority*

Admin's need to be able to add their students to the app by inputting the students names into a form. This function will add the student to the database, and create a unique id for the student to be used within the app.

3.1.2 Create Verb Pack - *High Priority*

Admin's need the ability to create Verb Pack's from a list of default verb-animation pairs that are preloaded into the app. Additionally any verb-animation pairs that they have added themselves will be selectable as well.

3.1.3 Create Verb-Animation Pair - *Medium Priority*

Admin's can add their own verb-animation pairs, if they want more options than the defaults. This will be done through a from where the admin inputs the verb, the definition, and uploads the animation files. Medium priority for the initial release, because admins will always have the ability to select from the default verb-animation pairs.

3.1.4 View Progress of Students - *Low Priority*

Admin's should have the ability to view the progress of their students. This will include seeing statistics such as how many verbs the student has successfully learned, most missed verbs, total number of verb packs completed, among others. Low priority for our initial release, because it is not a core feature to the playing / testing of verb-animation pairs.

3.1.5 Set Target Goals for Students - *Low Priority*

Admin's can set individual target goals for their students, such as total number of verbs learned. Low priority because it is not a core feature needed to demonstrate the playing / testing of verb-animation packs.

3.1.6 Create Group Sessions - *Medium Priority*

Admins will be able to select students from their student list view to add to a group session. Once the students are selected, the admin will select a single Verb Pack for the group to play. How the Verb Pack is played will be the same except that the application will display whose turn it is. Medium priority, will be one of the first features added after the initial Verb Pack playing is implemented.

3.2 Student Functional Requirements

3.2.1 Select a Verb Pack to play - *High Priority*

The first screen a student should be presented with is a list of available Verb Packs that are available to play. When a verb pack is touched it should start the testing of that Verb Pack.

3.2.2 View Completion % of Verb Pack - *Low Priority*

Allow students to stop and restart the testing of a Verb Pack in the middle of testing. The completion percentage (completed words in verb pack / total words in word pack) will be shown on the Verb Pack list screen. The completion percentage is a measure of how many words a student has seen in a Verb-Pack, it is not necessarily a measure of how many words that student got right. Low priority because the feature is not necessary to demonstrate the playing / testing of a verb pack.

3.2.3 Creating Missed Verb Packs for each Student - *Low Priority*

When a student mis-identifies a verb-animation pair during the course of going through a verb pack, that verb should be automatically added to a personal "Missed Verb Pack" that the student can use to review verbs he may of missed. When a verb-animation pair has been successfully completed in the "Missed Verb Pack" it will be removed. Low priority because it is not necessary to demonstrate the playing / testing of a verb pack.

3.2.4 Playing / Testing Verb Packs - *High Priority*

Once a verb pack has been selected to play, a student will be presented with a verb and a series of animations. The student's job is to select the animation that corresponds to the given verb. This process will repeat until the verb pack has been completed.

3.2.5 Giving Student Feedback - *Low Priority*

Once a student has completed a verb pack give them feedback about which verbs they missed and how these verbs are related. These statistics could also be displayed alongside completed Verb Packs. Low priority because it is not essential in order to demonstrate the playing / testing of verb packs.

3.2.6 Allow Students to Play on their Own Time- *Low Priority*

Allow students to create their own accounts linked to their teachers and play Verb Packs while not in a session with their teacher. If possible, allow students to download verb packs to the phone locally to allow them to be played while not connected to the internet. Low priority because it is not necessary to demonstrate the playing / testing of a verb pack.

4. Non-functional Requirements

4.1 Simple easy-to-use user interface - *High Priority*
The user interface of our application should be easy to use uncongested. Our target audience includes children with learning disabilities, so we feel as though we have extra motivation to create a very user friendly interface.

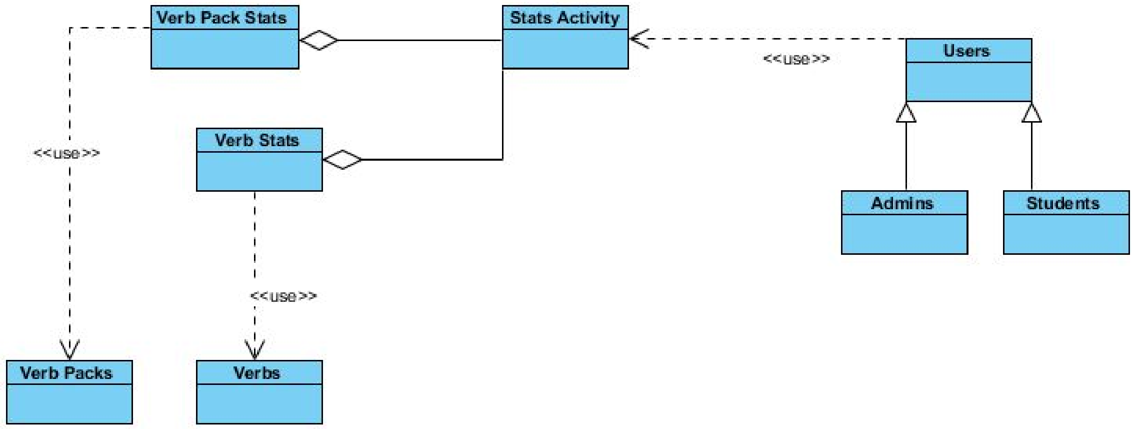
4.2 Friendly error handling- *Medium Priority*
The error handling of our application should be meticulously handled so that we do not present errors that a user does not recognize. Error messages to our users should be presented in a way that does not cause confusion.

4.3 Stable and frustration free - *Low Priority*
Creating a stable and frustration free app can take some time. Due to certain time constraints, such as the relatively short duration of our class term, making the app as stable as possible may only be possible if work is put into the app once our term is over. Therefore, in order to have something to present at the end of our class term, working out certain frustrations may need to be saved until after our term is over.

4.4 Offline Data Storage - *Low Priority*
Allowing admins to select verb packs or individual verbs that are considered “default” to be downloaded locally on their device so that they can be accessed without a connection to the internet is something that we plan to add once the base implementation of our app is complete.

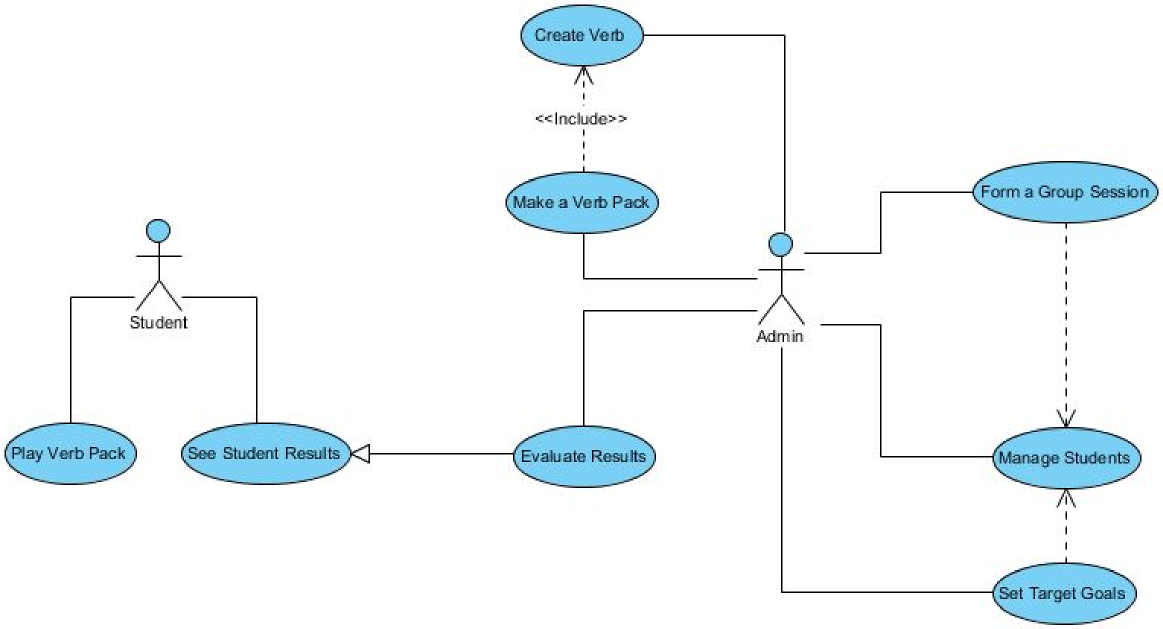
4.5 User Data Security - *Medium Priority*
Protecting the data of our users should be handled accordingly. Building a working base implementation of our app is the main goal, and user security will come once our app is in a fairly stable state and is ready to handle real users.

5. Requirements Diagrams



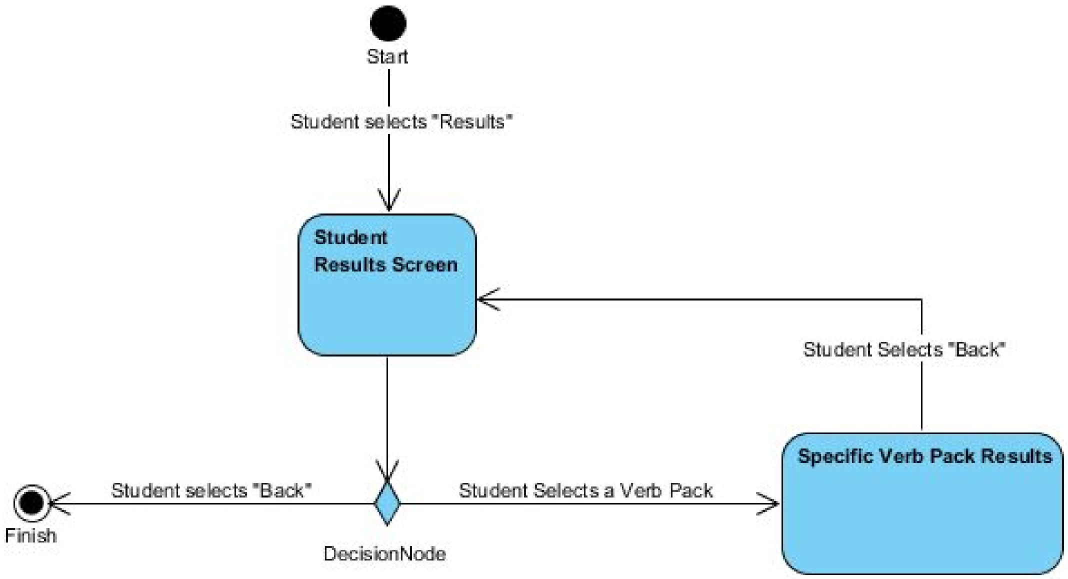
5.1 High Level Class Diagram

This is an overview of the classes we expect to implement in the finished application. The Users are broken down into Admins and Students, while the Stats Activity will be divided into Verb Pack Stats and specific Verb Stats, which depend on Verb Packs and Verbs, respectively.



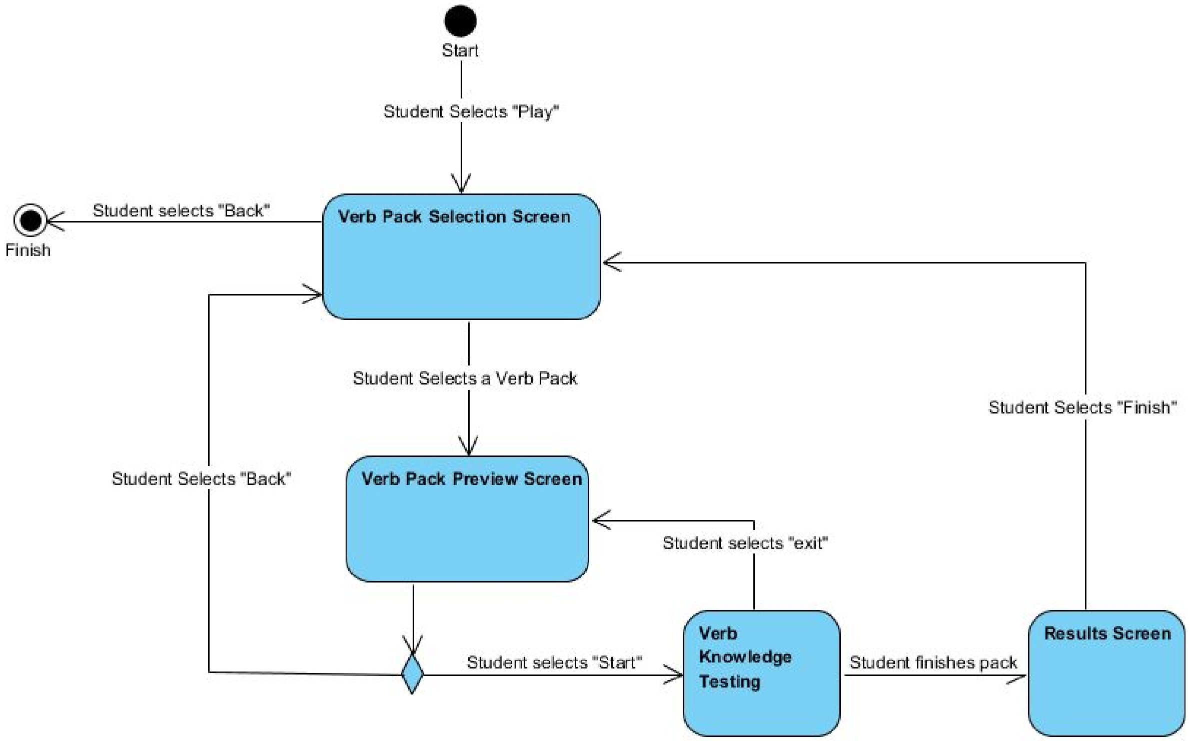
5.2 Use Case Diagram

This is a representation of the different cases in which our users might use our application. Each user has a specific list of actions they are allowed to perform. Forming a group session depends on the students available, which is affected by Manage Students. Evaluate Results includes the individual results seen by students.



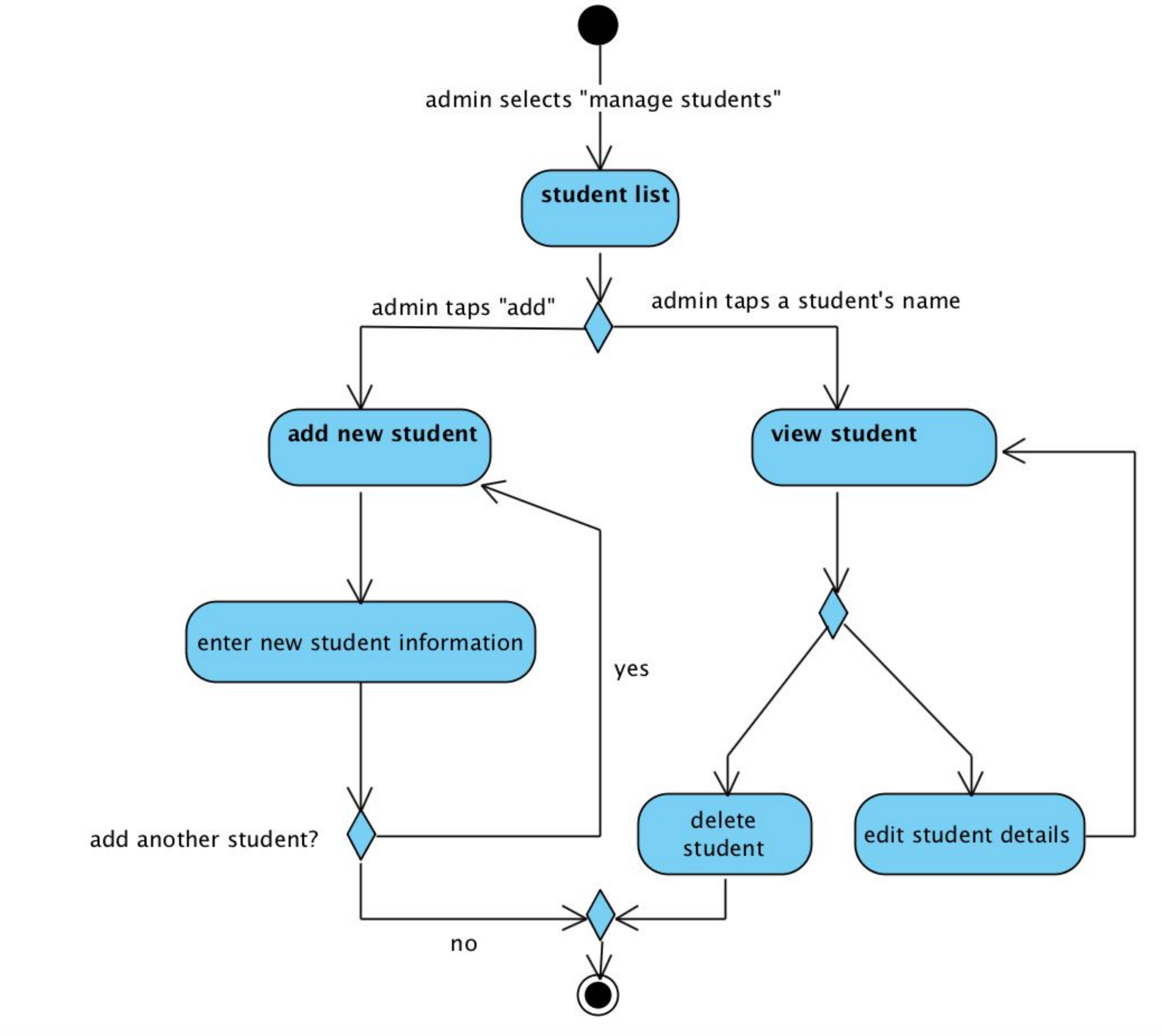
5.3 Results Activity Diagram

The Student starts by selecting Results, then a screen is displayed with a summary of their results information. They can drill down into the results of a specific Verb Pack by selecting a Verb Pack to view.



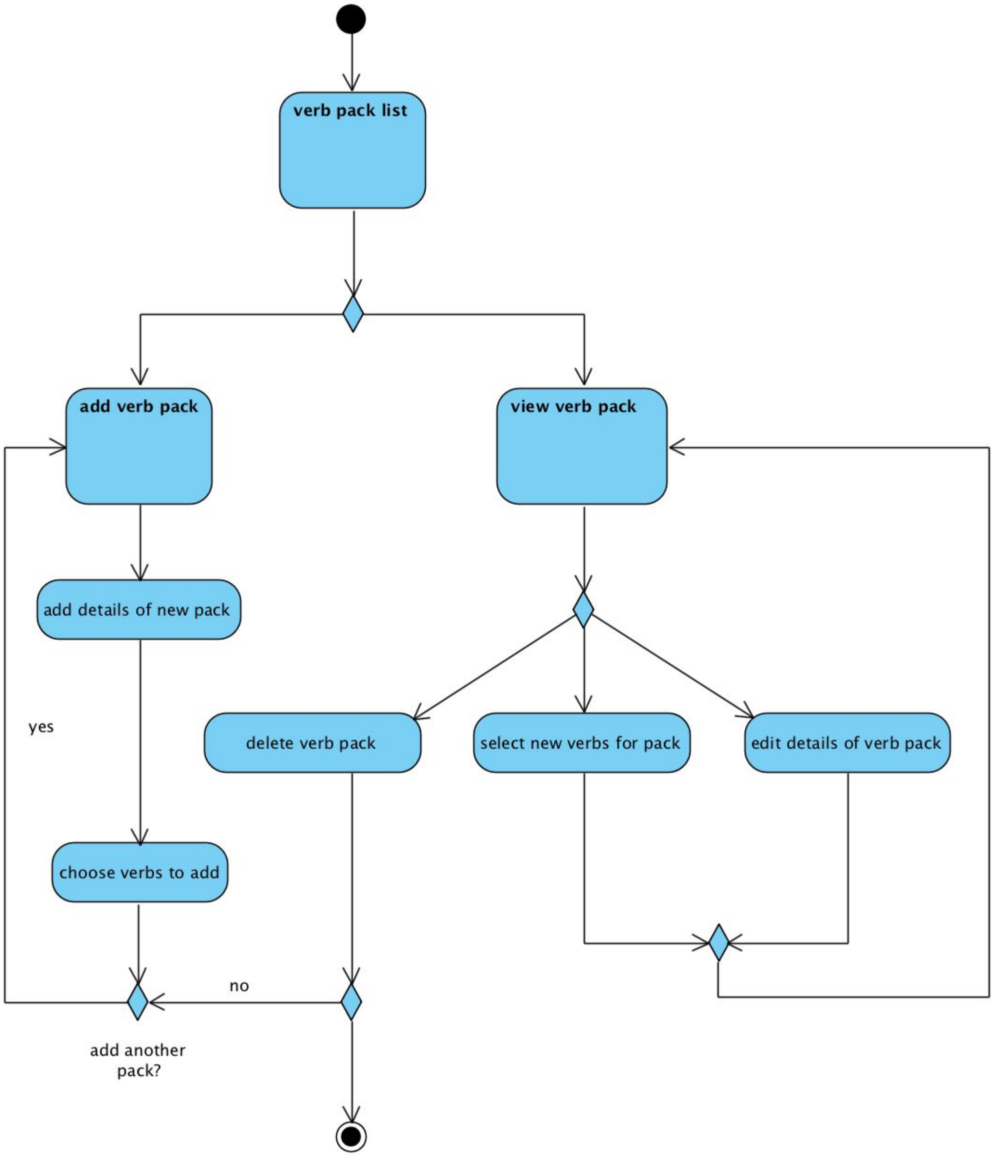
5.4 Play Activity Diagram

The Student selects play and is then presented with a screen that lists all Verb Packs available to them. From there, they can go back to the main screen or select a Verb Pack. Upon selecting a verb pack, the Student is presented with a preview screen of the pack that will show their current progress, if any. Upon selecting start, the Student will be presented with the play screen that contains 4 animations and one verb, the Student is tasked with selecting the animation that matches the verb. Once an animation is selected, the play screen continues to the next verb until it reaches the end of the list, where it proceeds to the results screen that presents the information on how the Student performed. From there they can select finish and are placed back at the Verb Packs selection screen.



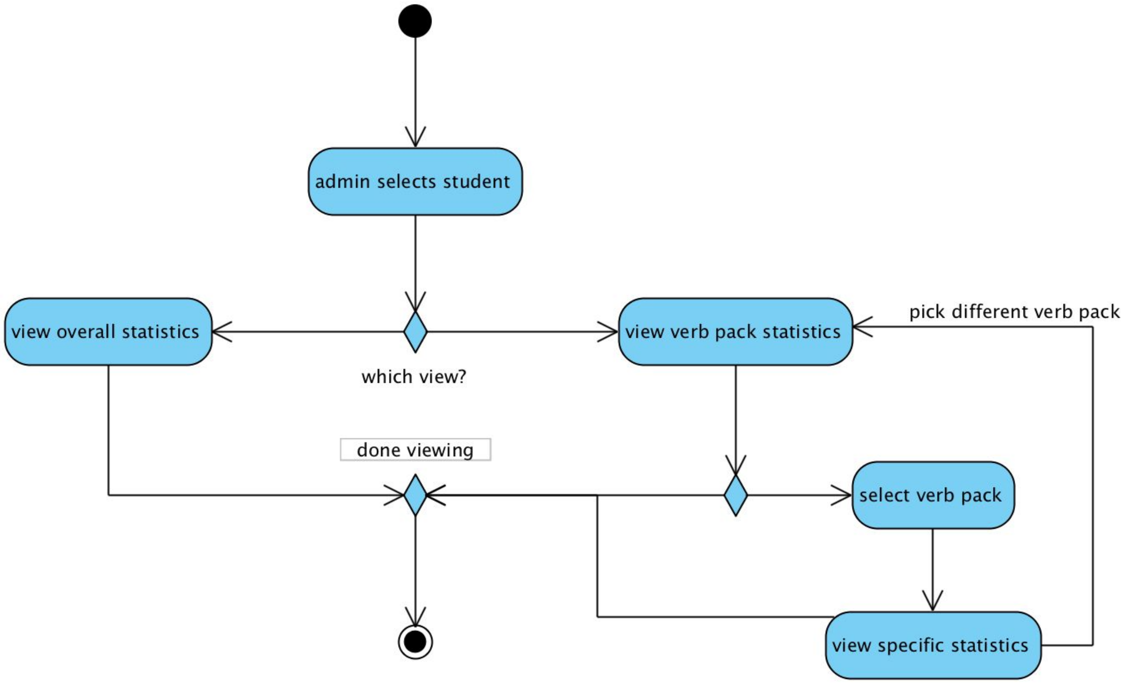
5.5 Manage Students Activity Diagram

The admin starts at a list of students, from there they can either add a new student or select an existing student for further options. In the case of adding a new student, the admin will enter the information about the new student. When they are done they will be asked if they wish to add another student, in which case they will be presented another new student form to fill out. When the admin selects an existing student, they will have the option to either delete the student or edit the details.



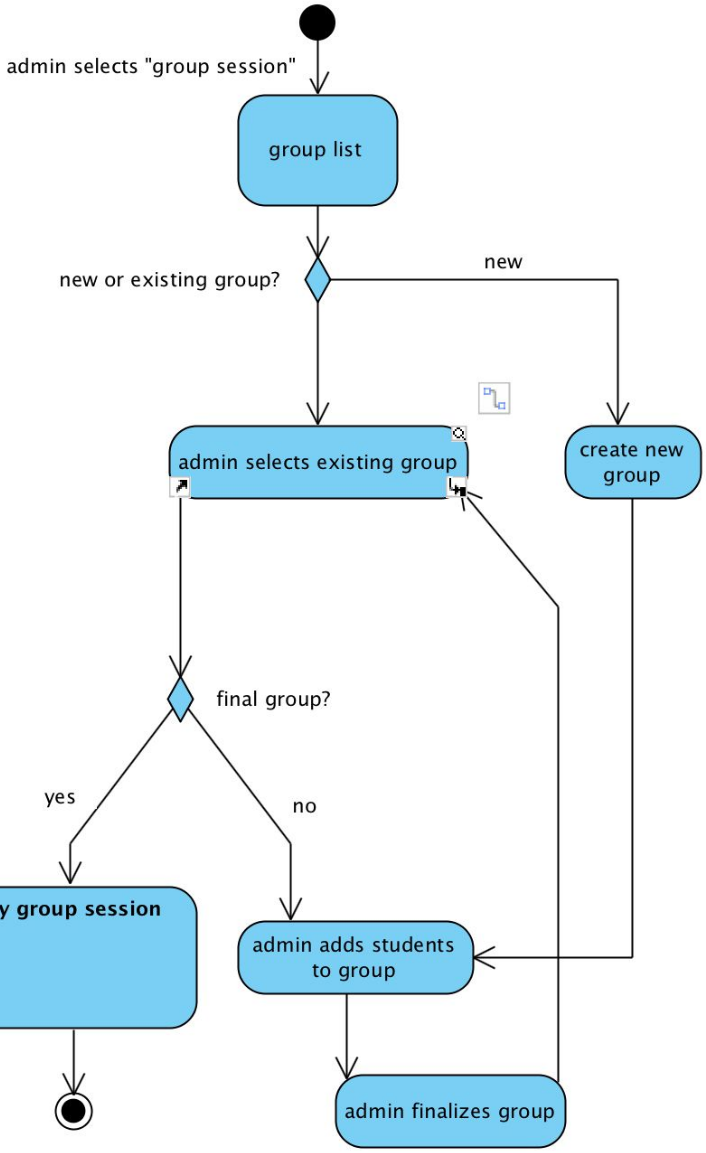
5.6 Manage Verb Packs Activity Diagram

To start, the admin will be presented with a list of existing verb packs. They can either create a new verb pack or choose an existing verb pack. In the case of creating a new verb pack, the admin will enter some basic information and then choose which verbs are to be included in the pack. They will then be asked if they wish to create another pack, in which case they will be presented with the original verb pack creation form. When the admin selects an existing verb pack, they are presented with several options: They can delete the verb pack, add new verbs to the verb pack, or change some data about the verb pack.



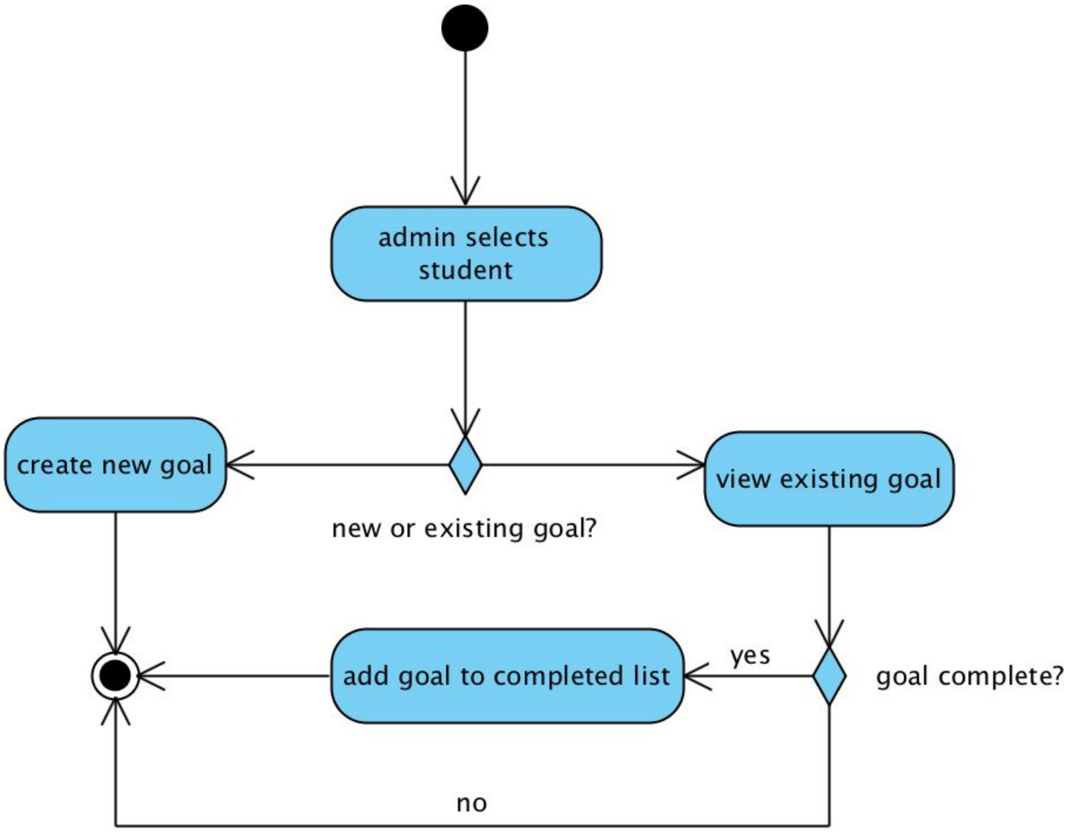
5.7 Evaluate Student Activity Diagram

At the student list, an admin can choose a student to view statistics. When a student is selected, the admin has the option to view overall statistics, or to view statistics per verb pack. In the case of overall statistics, they are presented with a list of the student's stats. If "verb pack stats" are selected, then they are presented with a list of verb packs. Upon selecting one of these verb packs, the admin is presented with statistics about the pack specific to the student (i.e. completion percentage, correct verbs, etc.). After viewing the statistics, the admin will return to the student details screen.



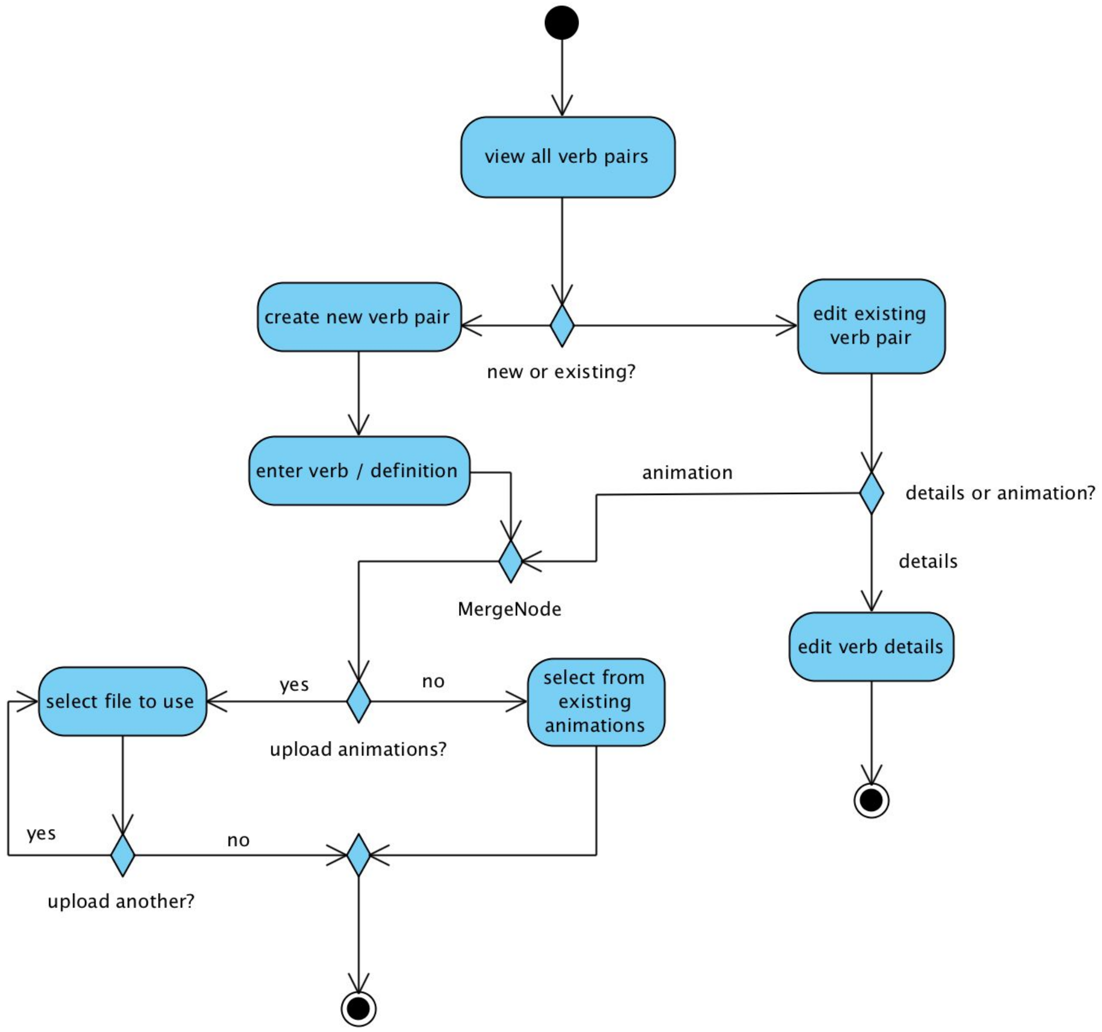
5.8 Create Group Session Activity Diagram

At the list of existing group sessions, the admin can either select one of those, or create a new one. In the case of creating a new one, they are prompted to add students to the group. Once they finalize the group, it is added to the group list. They are redirected to the group details screen, where they can edit the group or play the session. If the admin does not wish to create a new group, then they can select an existing one and be directed to the group details page described above.



5.9 Create Target Goal Activity Diagram

The admin starts by selecting a student from the student list. From there they are presented with a list of the student’s goals. They have the option to create a new goal, which will take them to a form to enter information about the goal. They can also view existing goals, and progress of those goals. If a goal is completed, then it is added to a “completed goals” list. These goals may be a target percentage for a certain verb pack or verbs, or completion of a certain verb pack.

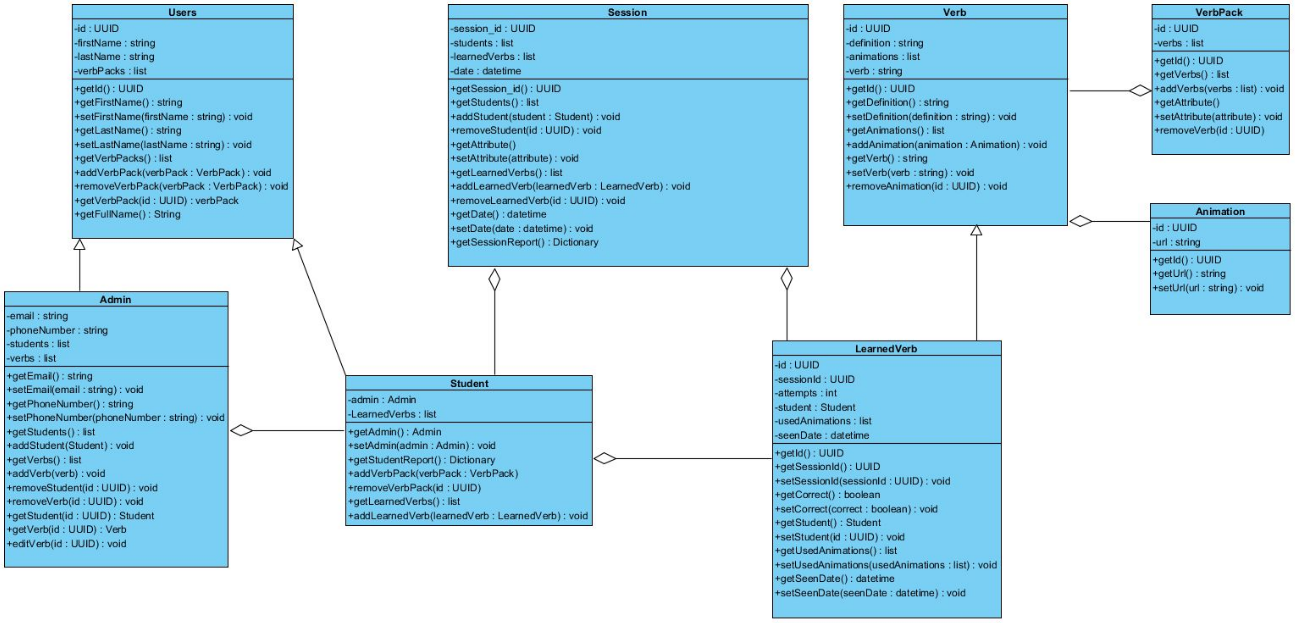


5.10 Create Verb-animation Set Activity Diagram

The admin is presented with a list of the verb-animation pairs on their device. They can either select one of the existing pairs, or create a new one. In the case of creating a new one, they are prompted to enter the verb and its definition. After that they are asked if they want to upload new animations or use existing ones. If they select "upload," they then select a file to upload. They can upload multiple files and then finish the action. If they decide to use existing animations, then they are presented with a list of existing animations that they can select. If the admin selects an existing pair from the main list, then they can either edit the details or the animations of it. If they select to edit the animations, then they are presented with a similar process described above. If they select to edit details, then they are presented with a form to enter the new details.

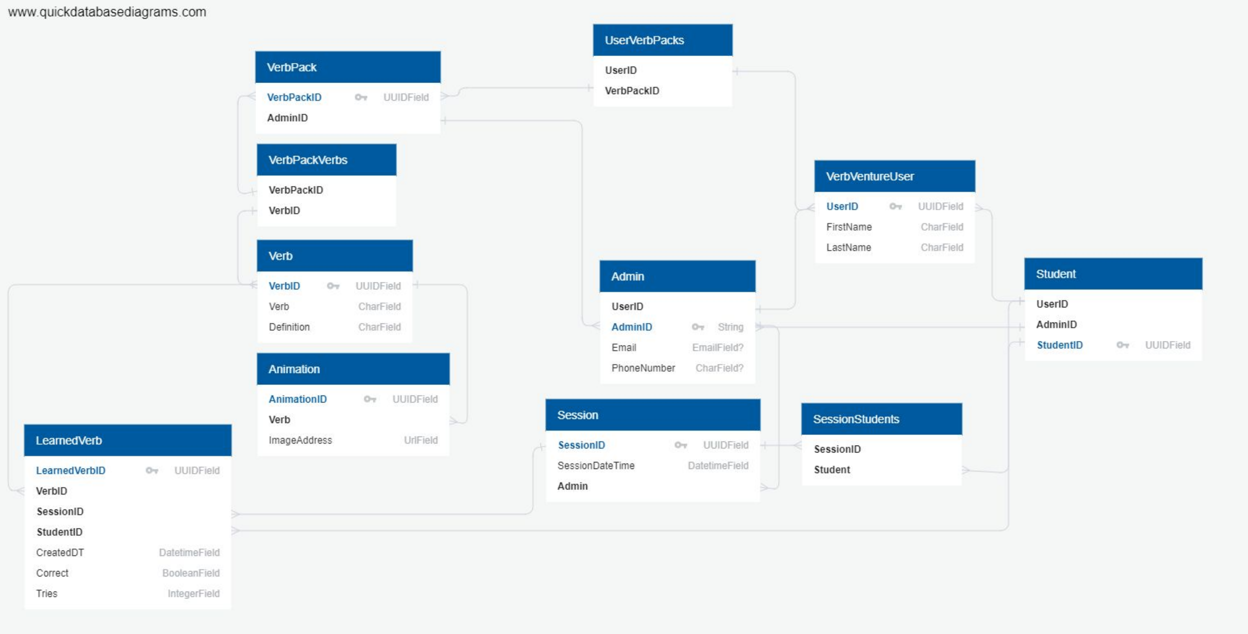
6. Design Diagrams

6.1 Class Diagram


Users
-id : UUID
-firstName : string
-lastName : string
-verbPacks : list
+getId() : UUID
+getFirstName() : string
+setFirstName(firstName : string) : void
+getLastName() : string
+setLastName(lastName : string) : void
+getVerbPacks() : list
+addVerbPack(verbPack : VerbPack) : void
+removeVerbPack(verbPack : VerbPack) : void
+getVerbPack(id : UUID) : verbPack
+getFullName() : String
Session
-session_id : UUID
-students : list
-learnedVerbs : list
-date : datetime
+getSession_id() : UUID
+getStudents() : list
+addStudent(student : Student) : void
+removeStudent(id : UUID) : void
+getAttribute()
+setAttribute(attribute) : void
+getLearnedVerbs() : list
+addLearnedVerb(learnedVerb : LearnedVerb) : void
+removeLearnedVerb(id : UUID) : void
+getDate() : datetime
+setDate(date : datetime) : void
+getSessionReport() : Dictionary
Verb
-id : UUID
-definition : string
-animations : list
-verb : string
+getId() : UUID
+getDefinition() : string
+setDefinition(definition : string) : void
+getAnimations() : list
+addAnimation(animation : Animation) : void
+getVerb() : string
+setVerb(verb : string) : void
+removeAnimation(id : UUID) : void
VerbPack
-id : UUID
-verbs : list
+getId() : UUID
+getVerbs() : list
+addVerbs(verbs : list) : void
+getAttribute()
+setAttribute(attribute) : void
+removeVerb(id : UUID)
Animation
-id : UUID
-url : string
+getId() : UUID
+getUrl() : string
+setUrl(url : string) : void
Admin
-email : string
-phoneNumber : string
-students : list
-verbs : list
+getEmail() : string
+setEmail(email : string) : void
+getPhoneNumber() : string
+setPhoneNumber(phoneNumber : string) : void
+getStudents() : list
+addStudent(Student) : void
+getVerbs() : list
+addVerb(verb) : void
+removeStudent(id : UUID) : void
+removeVerb(id : UUID) : void
+getStudent(id : UUID) : Student
+getVerb(id : UUID) : Verb
+editVerb(id : UUID) : void
Student
-admin : Admin
-LearnedVerbs : list
+getAdmin() : Admin
+setAdmin(admin : Admin) : void
+getStudentReport() : Dictionary
+addVerbPack(verbPack : VerbPack)
+removeVerbPack(id : UUID)
+getLearnedVerbs() : list
+addLearnedVerb(learnedVerb : LearnedVerb) : void
LearnedVerb
-id : UUID
-sessionId : UUID
-attempts : int
-student : Student
-usedAnimations : list
-seenDate : datetime
+getId() : UUID
+getSessionId() : UUID
+setSessionId(sessionId : UUID) : void
+getCorrect() : boolean
+setCorrect(correct : boolean) : void
+getStudent() : Student
+setStudent(id : UUID) : void
+getUsedAnimations() : list
+setUsedAnimations(usedAnimations : list) : void
+getSeenDate() : datetime
+setSeenDate(seenDate : datetime) : void

Our Admin and Student classes are our main actors, they inherit from the Users class. Admins can create Verbs, Verb Packs, and Sessions. Each Verb is comprised of an ID, verb, definition, and series of animations. Each animation has a URL. VerbPacks are just collections of verbs. LearnedVerb inherits from Verb and adds some information that allows us to determine when the verb was seen and how many tries it took the student to get the correct answer. Sessions are created with a list of Students, they then use those Students to create a VerbPack for that Session by placing the students in an order and selecting a verb from one of their packs in that order.

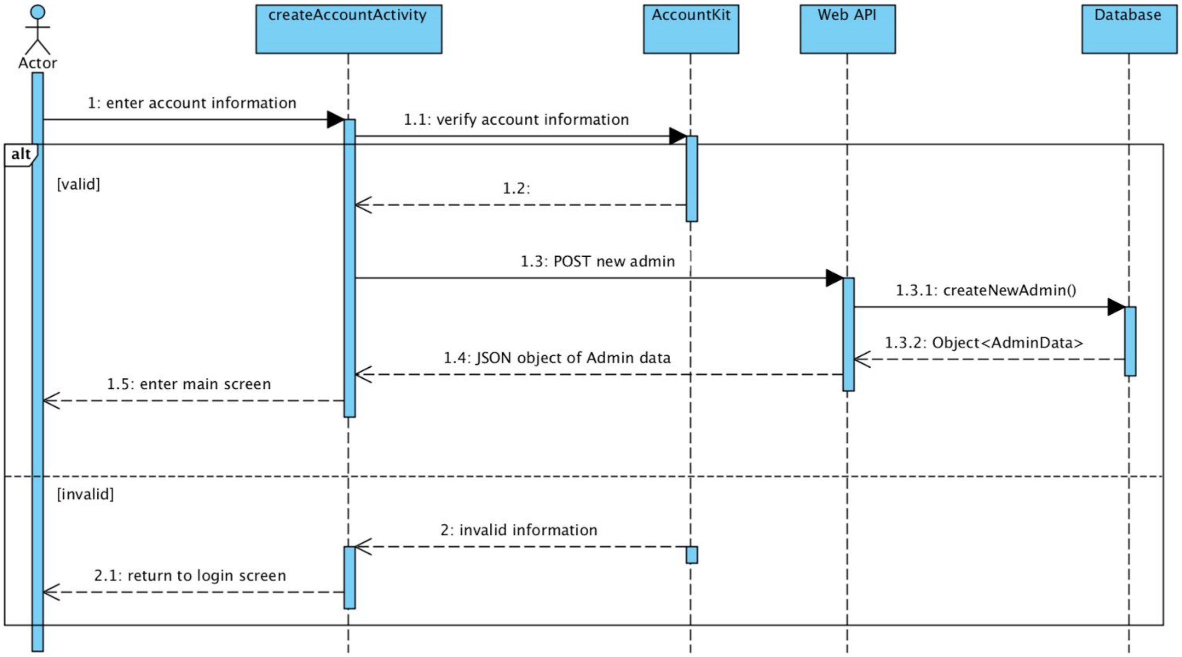
6.2 Database Diagram



A PostgresQL relational database hosted on AWS will be used as the backend data storage for the application. The key tables that relate to the user classes include VerbVentureUser, Admin, and Student. UserVerbPacks is a intermdiary table for the many to many relationship between Users and VerbPacks. One VerbPack object can have many users, and one user can have many VerbPacks. Likewise, VerbPackVerbs and SessionStudents are also many to many intermediary tables. Session contains information about a single learning session. Finally, the LearnedVerb table will be the largest table. This table contains all the verbs students have seen while learning in a session and weather that student was successful in learning that verb. This table will be queried in order to create the individual student and session reports. SessionID and StudentID will be indexes in order to allow for fast querying.

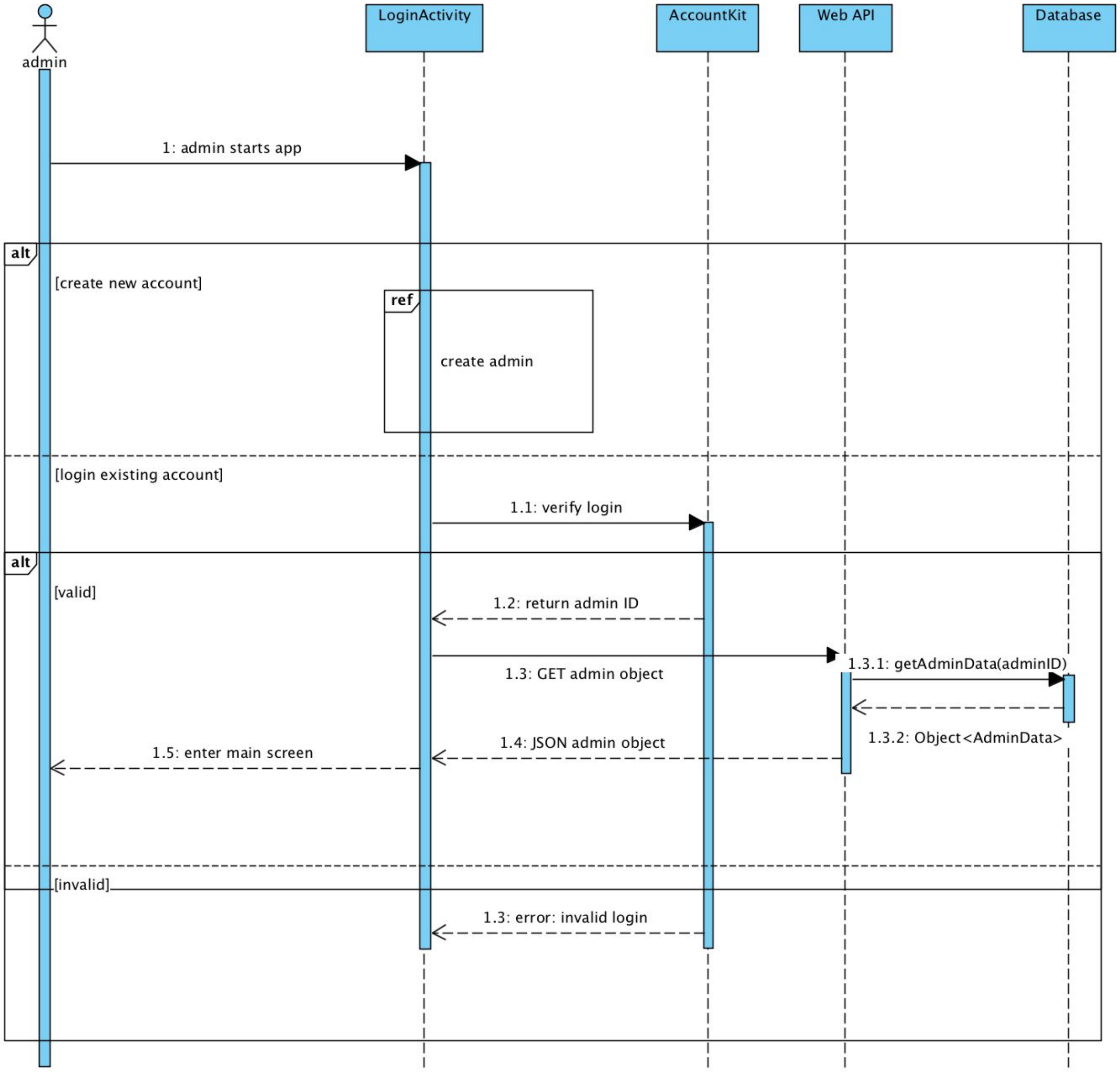
6.3 Sequence Diagram

6.3.1 Create Admin



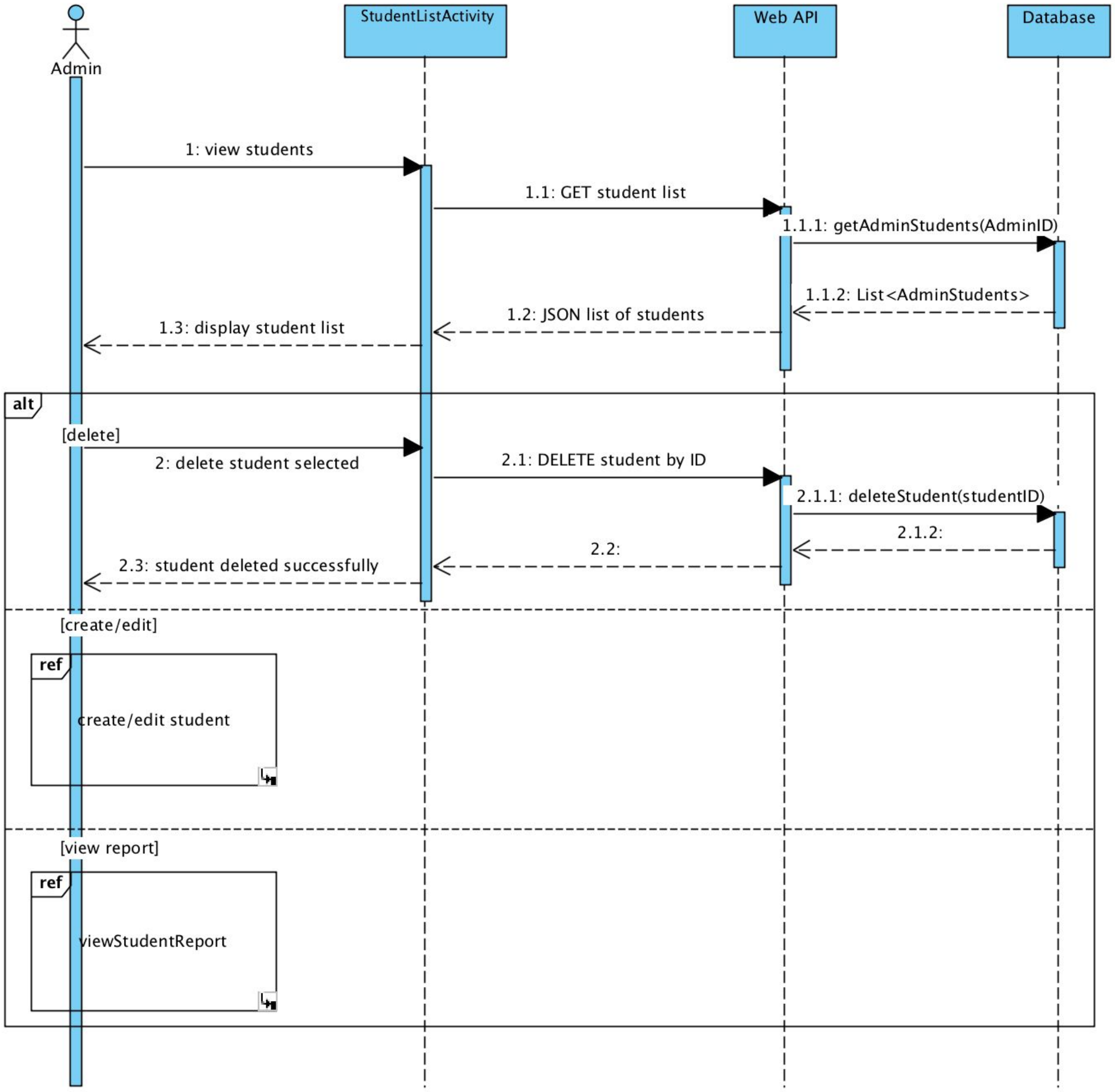
Upon logging in for the first time, the Admin will be prompted to create an account. They will enter some basic information like their name, and then either their phone number or email which will be passed to Facebook's Account Kit framework for verification. Upon verification of their email or phone number the Admin will be logged in and will stay logged in on the device until they select sign out. If their information is not verified, then they will be returned to the login screen.

6.3.2 Admin Login



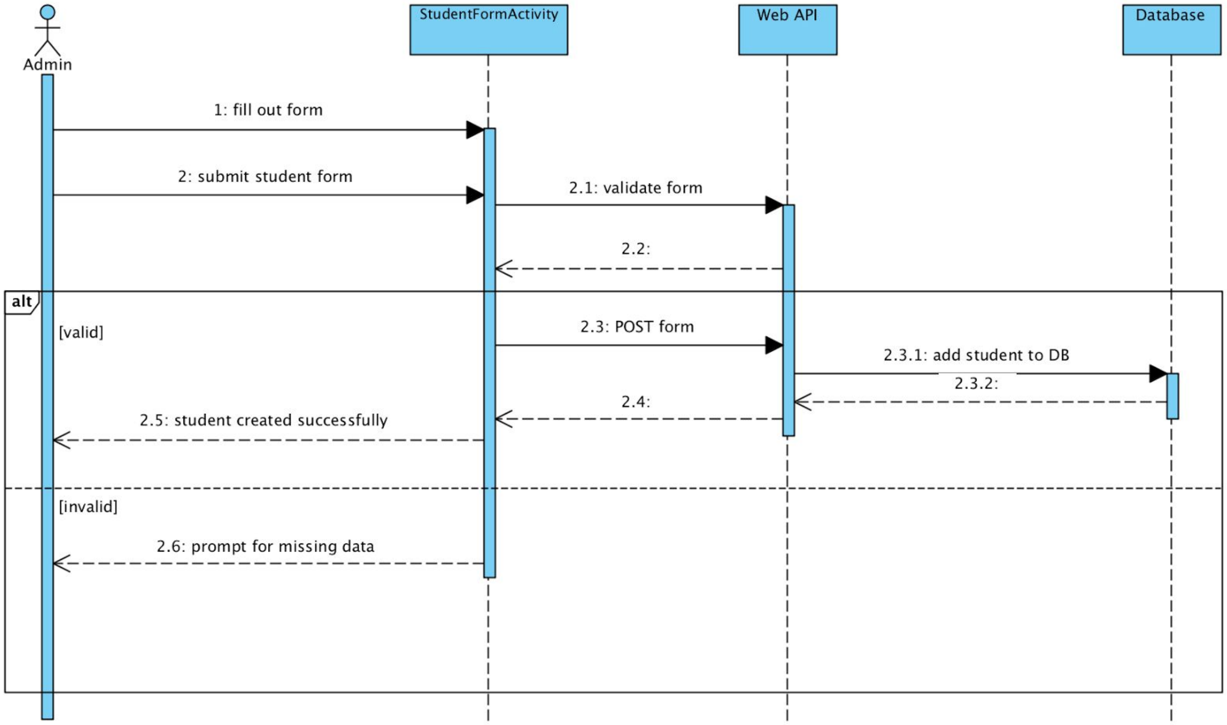
If the Admin already has an account, they will be asked to input their email address or phone number. Once entered, this will be passed to Account Kit for verification and the Admin will be logged in once verification is complete.

6.3.3 Manage Students


Admin
StudentListActivity
Web API
Database
1: view students
1.1: GET student list
1.1.1: getAdminStudents(AdminID)
1.1.2: List<AdminStudents>
1.2: JSON list of students
1.3: display student list
alt
[delete]
2: delete student selected
2.1: DELETE student by ID
2.1.1: deleteStudent(studentID)
2.1.2:
2.2:
2.3: student deleted successfully
[create/edit]
ref
create/edit student
[view report]
ref
viewStudentReport

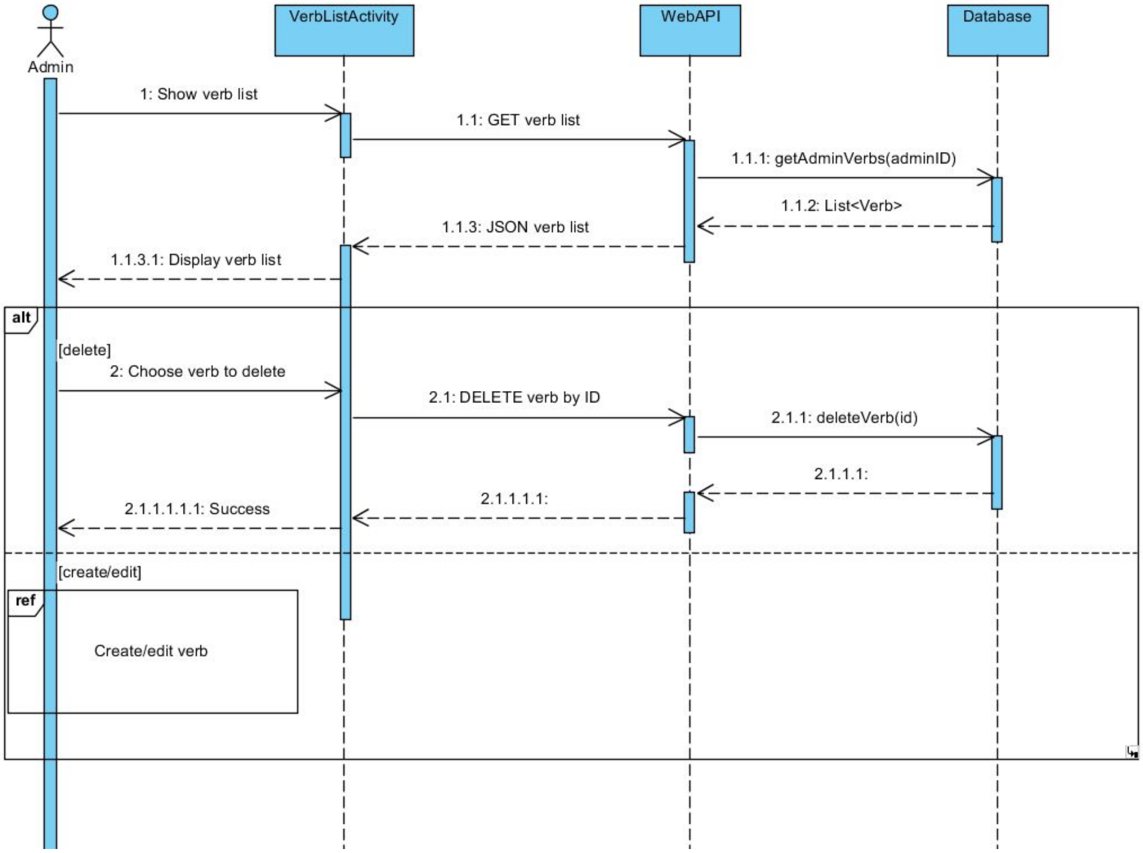
When an admin wishes to manage a student, whether that be to add, delete, or edit a student or view a student report, they will request to view a list of their students. A list of students will eventually be displayed back to the admin once the web API works with the database to retrieve the list of students. If a student is to be deleted, the delete option beside a student will be chosen. The student will then be deleted from the database and the admin will be notified of a successful deletion. If an admin wishes to create or edit a student, they will begin the “Create/edit student” sequence. If an admin wishes to view a student report, they will start the “View student report” sequence.

6.3.4 Create / Edit Student



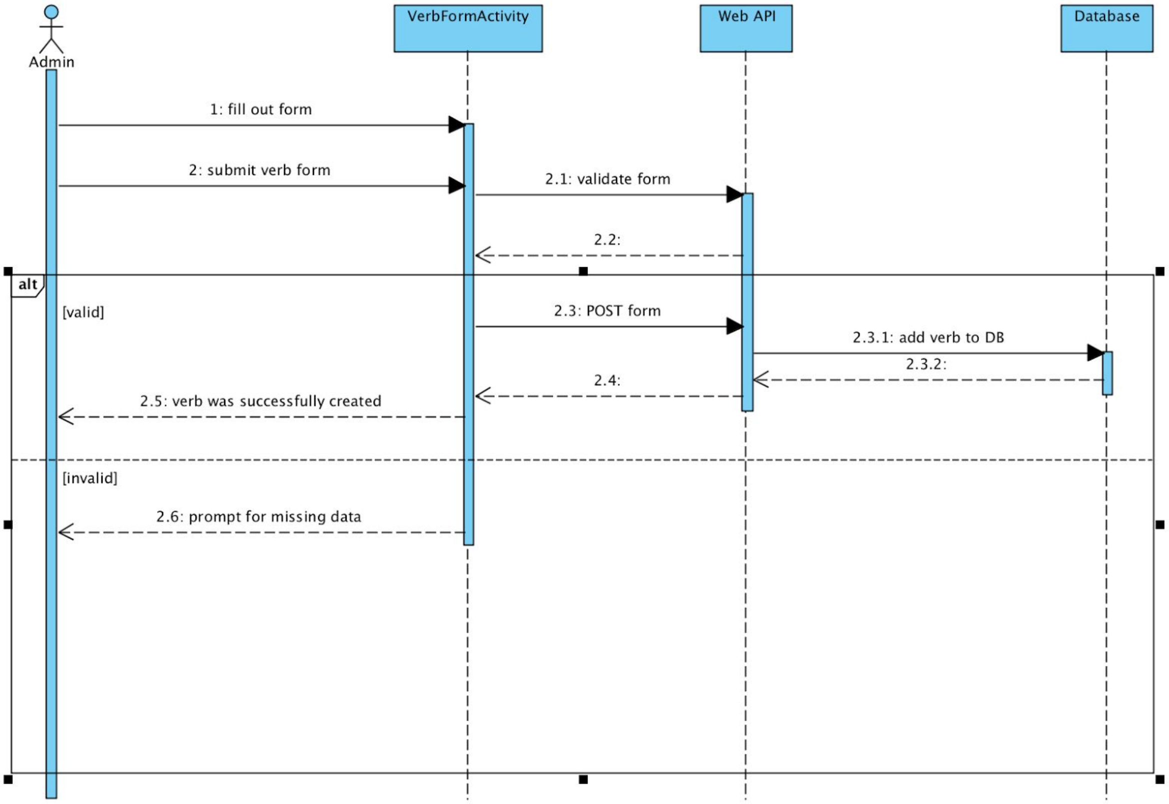
When an admin needs to add or edit a student to their list they will be presented with a form. If the admin is creating a student the form will be empty, if the admin is editing a student the form will be prepopulated with the student's info. The admin can then add or edit the student's info to the form, and submit. The form will be validated, if the form is valid a POST request will be sent to the Web API and the student's info will be saved. If the from is invalid the app will prompt for the missing information.

6.3.5 Manage Verbs



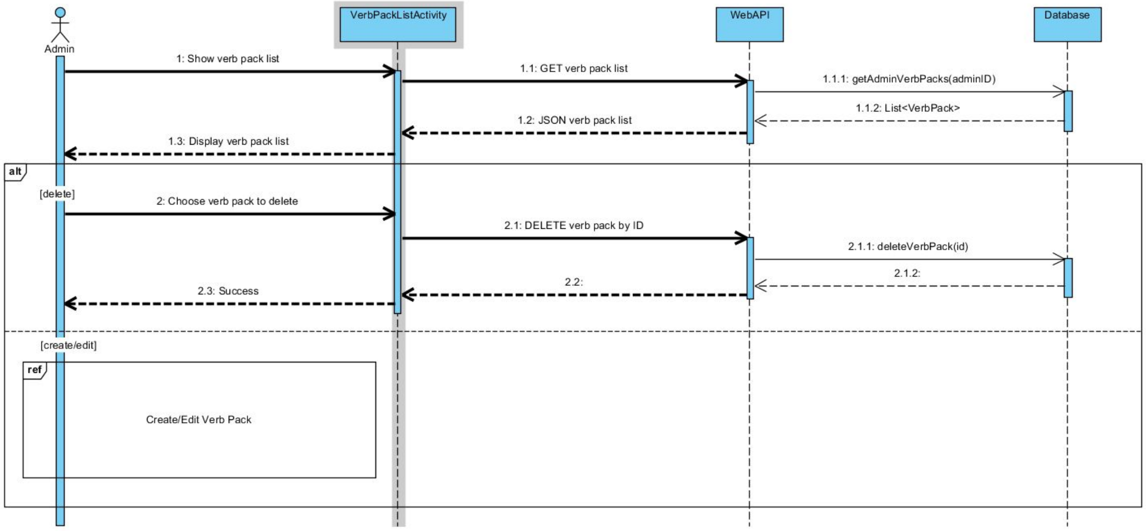
When an admin wishes to create, edit, or delete a verb, the admin selects the option to view a list of verbs associated with their admin ID. The web API and database returns a list of verbs for the admin to see. If an admin wishes to delete a verb, they choose the delete option for the verb and the database deletes the verb and the user is informed of successful deletion. If the admin wishes to create or edit a verb, they begin the “Create/edit verb” sequence.

6.3.6 Create / Edit Verb



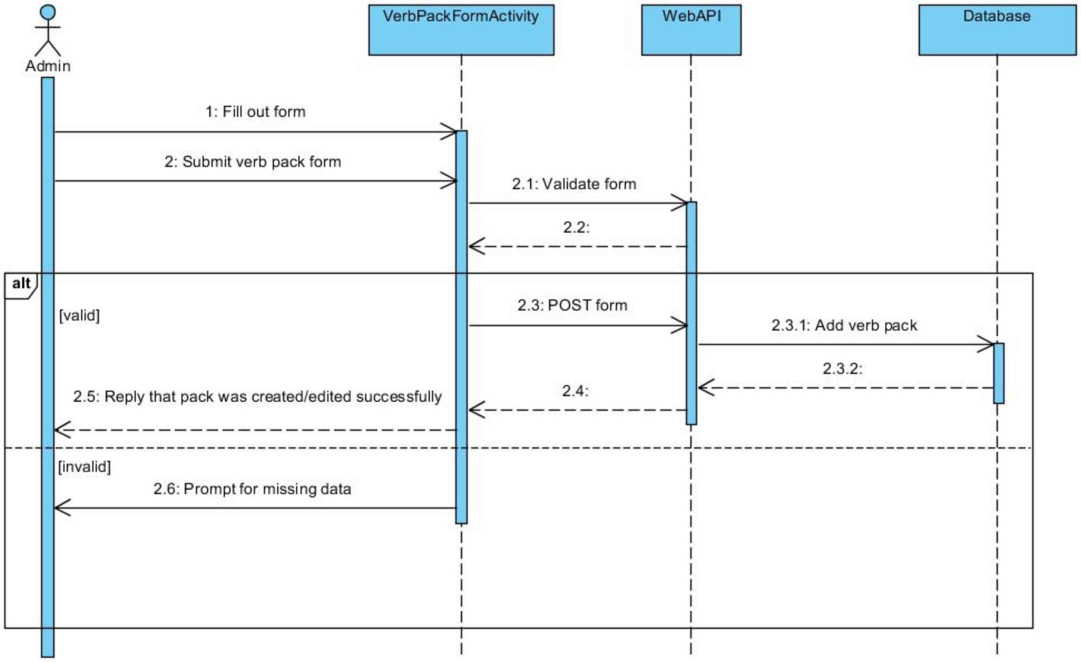
When an admin chooses to edit or create a verb they are presented with a form. If the admin is creating the verb the form will be blank; if the admin is editing a verb the form will be prefilled with the verb's data. The user will then populate the form with their changes, and then this data will be submitted to Web API for validation. If the form is valid, then a POST request will be sent to the API and the verb will be added to the database. If the form is not valid, the user will be presented with a prompt to add in the missing fields.

6.3.7 Manage Verb Pack



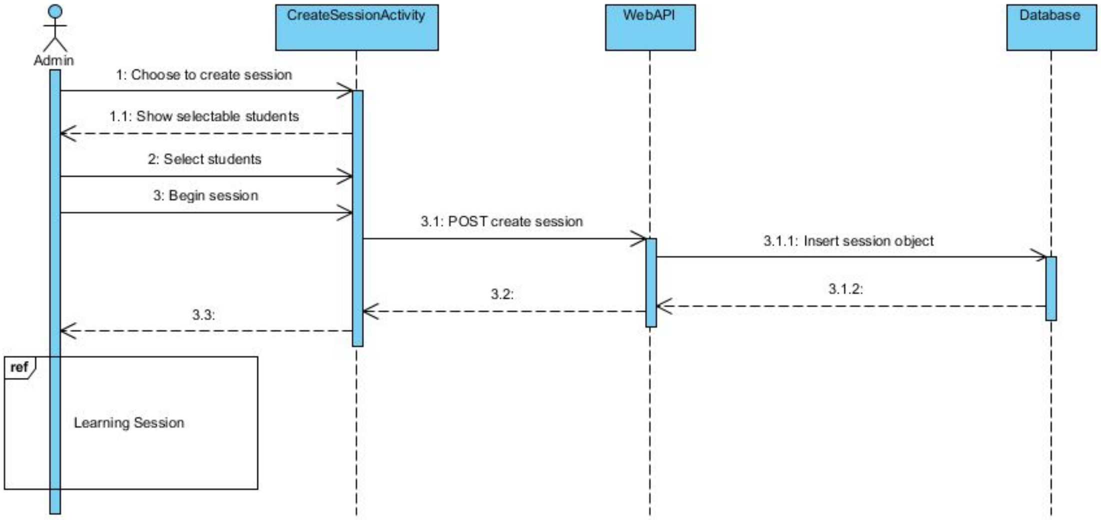
If an admin wishes to create, edit, or delete a verb pack, they must choose to view a list of their verb packs. Once the web API and database communicate to gather the verb packs, the activity will display the list of verb packs to the admin. Once the list is displayed, if the admin chooses to delete a particular verb pack, it is deleted from the database and the admin is informed of successful deletion. If they choose to create or edit a verb pack, they begin the “Create/edit verb pack” sequence.

6.3.8 Create / Edit Verb Pack



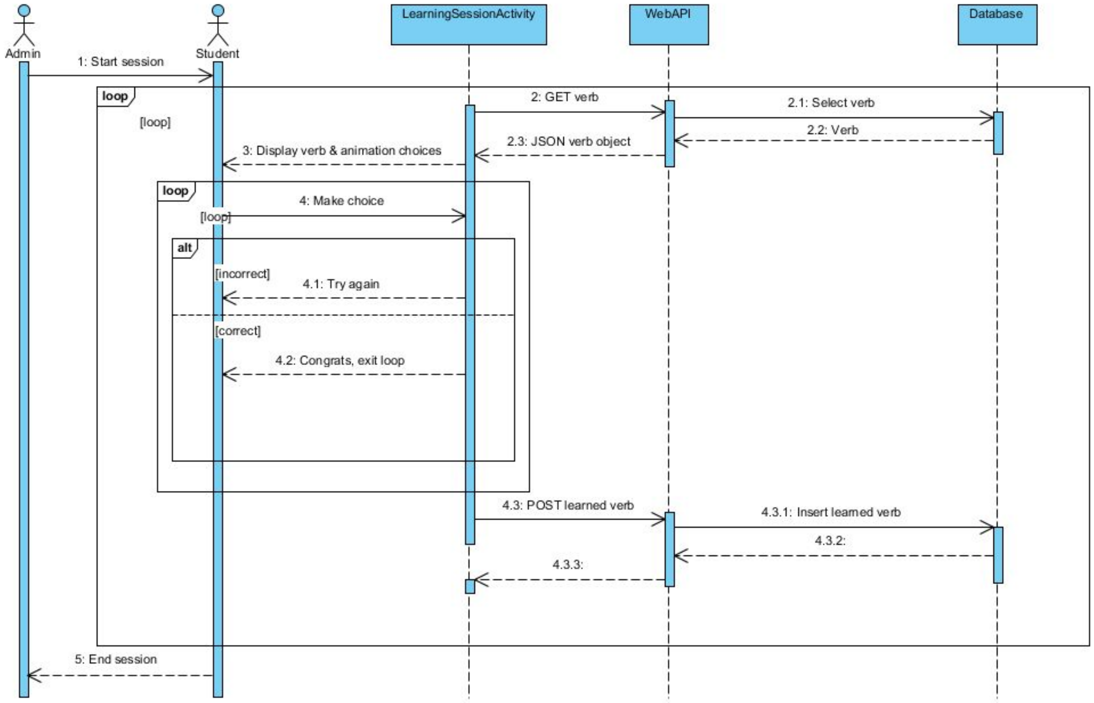
When an admin chooses to create a new verb pack or edit an already created verb pack, they are presented with a screen that contains a form. If the admin is creating a verb pack, the form will be blank. If they are editing a verb pack, the form will be populated with the current data for the pack which the admin can edit. If the form is valid and has data for all of the required fields, it is added to the database and the user is informed that the pack was created. If the data in the form is somehow incomplete or invalid, the admin is informed that data is missing.

6.3.9 Create Session



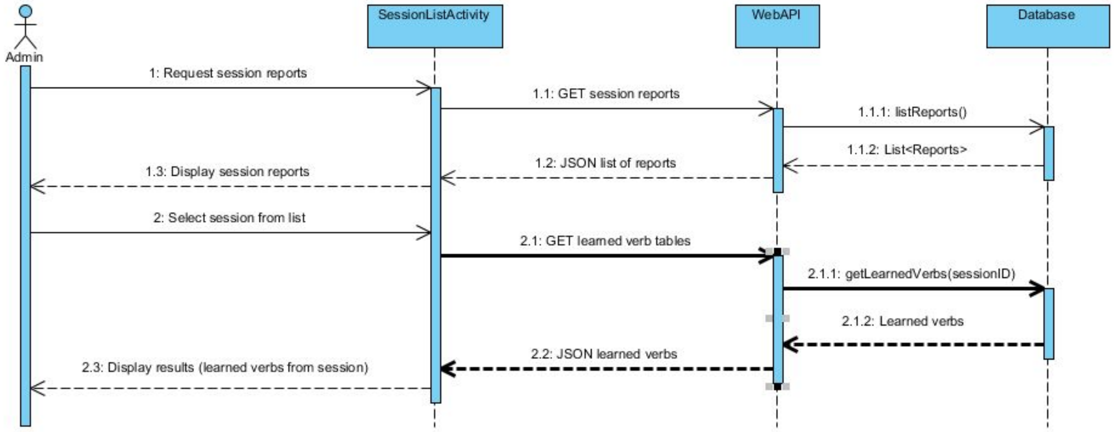
When an admin chooses to create a learning session, they are presented with a screen that shows the list of students they can select to add to a learning session. The admin will then select the students they wish to participate in the learning session, whether that be only one student or multiple students. The admin will then choose to begin the session. Once a session has been started, a learning session object will be stored in the database, followed by a student beginning a learning session.

6.3.10 Learn Session



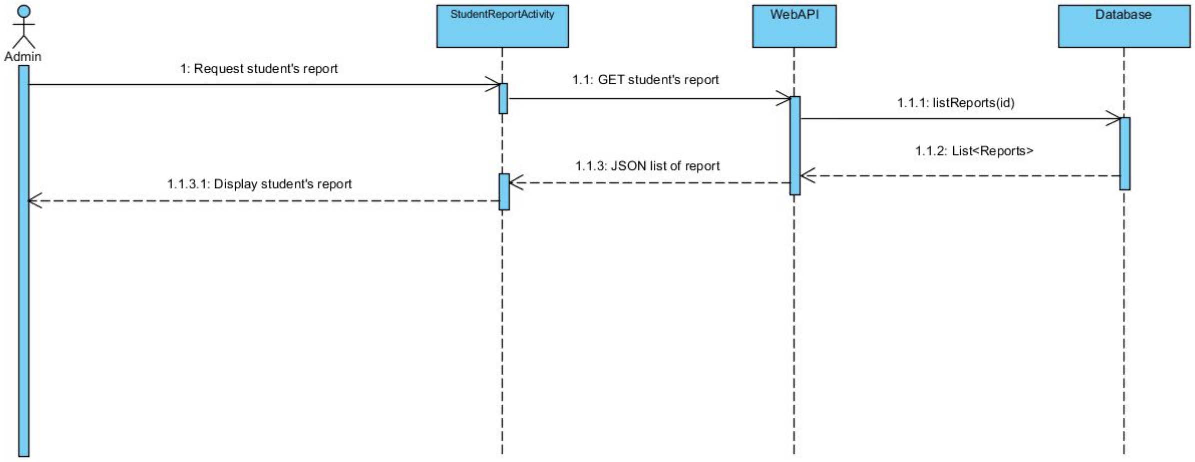
Once an admin has created a learning session and it is started, the learning session activity will fetch a verb from the database to display to the student. The student will be shown the verb along with one correct animation that depicts the verb and three incorrect animations. The student will choose the animation that they think best depicts the verb. If an incorrect animation is chosen, the student is instructed to try again, with their incorrect animation choice disappearing from the screen so it can no longer be accepted. If the correct animation is chosen, the student is notified they were correct and the fact that they learned the verb is noted in the database. This process of presenting a verb to a student is repeated until all students in the session have been presented all of their verbs.

6.3.12 Get Session Report


Admin
SessionListActivity
WebAPI
Database
1: Request session reports
1.1: GET session reports
1.1.1: listReports()
1.1.2: List<Reports>
1.2: JSON list of reports
1.3: Display session reports
2: Select session from list
2.1: GET learned verb tables
2.1.1: getLearnedVerbs(sessionID)
2.1.2: Learned verbs
2.2: JSON learned verbs
2.3: Display results (learned verbs from session)

When an admin wishes to view statistics from a particular learning session, they will select the option to view session reports. The web API and database will work together to return a list of all sessions that have been played and the session reports will be shown to the admin. The admin can then select a specific session from the list and the web API will retrieve the learned verbs from that session for all students that participated. The learned verbs from the session will then be displayed to the admin.

6.3.13 Get Student Report


Admin
StudentReportActivity
WebAPI
Database
1: Request student's report
1.1: GET student's report
1.1.1: listReports(id)
1.1.2: List<Reports>
1.1.3: JSON list of report
1.1.3.1: Display student's report

If an admin wishes to view statistics for a particular student, they will request a student's report by selecting the student from a list of students associated with them. The web API and database will work to return a report of statistics for all sessions the student has participated in. The student's statistics report will then be displayed back to the admin.