SOEN 490 - Capstone Software Engineering Design Project
Version 4

Lock & Learn Milestone Document

Fall 2023

Instructor: Dr. Peter Rigby
Date: 14/11/23

Student ID	Student Name
40176279	Alice Chen
40177866	Fatema Akther
40175616	Leon Zhang
40176360	William Chong
40175335	Julie Trinh
40174210	Hao Yi Liu
40178711	Anna Hsu
40175423	Ryan Kim
40175034	Georgia Pitic
26665691	Costa Papadakos

Lock & Learn

Team Members

Name and student ID	GitHub id	Number of story points that member was an **author** on.
Fatema Akther (40177866)	fatema-a	26
Leon Zhang (40175616)	LeonZhang7	21
Julie Trinh (40175335)	julietrinh1	16
William Chong (40176360)	Chovv	16
Georgia Pitic (40175034)	georgia1515	14
Ryan Kim (40175423)	RIGNITE	16
Alice Chen (40176279)	Aykchen	22
Hao Yi Liu (40174210)	Niniishere	18
Anna Hsu (40178711)	AnnaHsu1	18
Costa Papadakos (26665691)	Cotso	

Project Summary

Lock & Learn provides an efficient tool for parents to have control over their child's digital life/device usage by implementing a lock mechanism on their device. Once the child completes and successfully passes the lesson selected by the parent, the system will unlock to allow the child to continue using their device. Our application can easily be tailored to help kids focus on their studies without having any digital distractions through their tablets, laptops, and phones. This platform gives peace of mind to parents knowing that their child can maximize their academic potential while not needing to monitor them regularly.

Risk

The following are the highest risks:

- Locking the device: We want the locking system to work properly otherwise the app will not work as intended. We came up with multiple solutions and will be primarily focusing on encrypting sensitive data related to the locking mechanism to prevent tampering and password protection on changing the locking mechanism. See Issue 17 for more information.
- Unlocking the device: We do not want permanent or unplanned locking as it would cause serious problems. To mitigate this, we will have an emergency code to unlock the device. See Issue 20 for more information.
- Security for payment methods: Payment processes are sensitive to security issues since the user's personal information will be found there. We will be using Stripe SDK to ensure a secure transaction for the users. See Issue 4 for more information.
- Uploading study material: There is a high risk due to the specific way implementation is wanted by our stakeholders and the unfamiliarity of the task to our group. Since this is needed right away we cannot afford to delay it. See Issue 1 and #71 for more information.

- Work Package Creation: This feature is high risk due to the nature of the story requiring a good level of understanding from the way both study material files and quizzes are created and handled. Work packages affect the study material and quiz story, which proposes a high amount of effort needed to carefully connect these components. See Issue 49
- Sign up and Log in: This is a high risk since it is related to security and has multiple layers with parents and children accounts and we need to ensure security on each level. See #26 for more information.

Legal and Ethical Issues

Privacy Concerns: Locking and monitoring a child's device usage may raise privacy concerns, as it involves tracking and controlling their digital activities. This needs to be managed carefully to ensure compliance with privacy regulations, we need consent from children and parents.

Ethical Considerations: Locking children out of their devices for extended periods may lead to questions about the ethical implications of such control. Striking a balance between parental guidance and personal freedom is crucial. We need to make sure there is a limit for the study time so parents won't abuse the system.

Environmental Impact: No large component, as of now, can have an environmental impact due to increased energy consumption. This should be taken into account and minimized where possible.

Discrimination Concerns: The application's usage should be fair and not discriminate against any minority groups. It should not be used to target or discriminate based on race, gender, or other characteristics. We should monitor content to make sure there is nothing discriminatory.

If using libraries, the team has to ensure to use open source libraries in case the application is published on the marketplace.

Impact on Jobs/Careers: Depending on the extent of adoption, Lock & Learn could potentially disrupt existing careers related to child education, as it provides an alternative means for parents to manage their children's digital activities. Tutoring services might be less needed. However, this impact is likely to vary based on the business model given to us by the stakeholders.

Economic

Parents purchasing study material will allow instructors to get paid and we will get a percentage of this amount.

Contractor Estimate

No charge will be applied for release/iteration work. We will be working free of charge and in exchange, our group will have a 50% stake in the project.

Velocity

Project Total: 32 stories, 144 points over 26 weeks

Iteration 1 (2 stories, 13 points)
The main achievements are regarding setting up Github and the base of our application as well as installing Docker, CI/CD, and its emulators. Stories were listed with their labels, risks, task distribution, and points (SP, priority, risk). Documentations such as the project proposal, project milestones, software testing, product vision document, SAD (overview of architecture and design were established), and MOU were completed. We also progressed in the features of signing up/logging in and uploading study material which will be carried onto iteration 2 for further development.

Iteration 2 (3 stories, 23 points)
The main achievement from this iteration will be the database was connected to the application based on the stakeholder preference, since the decision on the database was made later on delays were made in doing so and we had to do it within this iteration instead of iteration 1. Now that it is made users will be able to sign in based on their user type in a secure manner. Instructors will be able to upload material and choose their formats such as pdf or txt. Locking was also worked on but due to its complexity, it will not be able to be completed for this iteration.

Iteration 3 (3 stories, 21 points)
For this iteration, the main goal will be: to allow parents to create profiles for their children, to have the upload feature accommodate various formats of learning, to be able to download and delete the study material for the instructor, and to continue the work on the app-locking mechanism of the application. This iteration also included providing the Tutors with different formats of questions for the creation of quizzing material. Now that parents have their accounts, they will have to create profiles for their children so they can use the application as well. The different formats of learning material will offer diverse study material to the children. The different types of quizzing questions will provide diverse methods for the tutor to quiz its students which will help quiz different competencies.

Release 1 (Iteration 4) (6 stories, 23 points)
For this iteration, the main goal is to: allow tutors to tag their study material with keyboards that are pertinent to the content they upload, allow tutors to edit the quizzes they create, allow tutors to create work packages to group their study material, and quizzes, and allow them to preview the study material within the same page. There are also a few bug fixes and enhancements planned for this iteration such as app refresh not working to automatically update the data shown to the user and deleting an uploaded document.
Release 1 Video (click on view raw to download the video)

Iteration 5 (3 stories, 16 points)

For this iteration, the main goal is to figure out the locking device mechanism so the children cannot get out of the application, and to allow students to review their answers to quizzes they have done so they have an idea of how well they are doing, and to allow the parents to configure a passing grade for a subject their child is working on so they are only allowed a pass in that quiz if their grade is higher than the set passing grade.

The rest of the iterations will be discussed as we go further within this project based on the priorities of our stakeholders.

Overall Architecture and Class Diagram

In our GitHub stories, we have started to include more in-depth diagrams to see their architecture more in-depth. We will create these diagrams as we go. In Issue 26 we have a use case diagram for how each type of user will be able to log in. Issue 1 we have a state diagram and a use case diagram for how uploading logic will work and how users will interact with the system. Issue 17 we have a flowchart diagram of how the locking mechanism works and also a sequence diagram for how the components interact.

In Figure 1, we have a component diagram illustrating the main components of our system. It shows how the users, system, and backend server connect.

In Figure 2, we have a use case diagram displaying how users will use the application and components it interacts with, payment, and AI correction tool (potential view from our client, which is not yet concrete).

In Figure 3, we have the domain model to help our stakeholders understand the structure and dynamics of the system we will create.

In Figure 4, we have a class diagram that illustrates the system's structure. It shows our different classes, their attributes, their methods, and the different relations the classes have between them. It will help us understand the different components and how they interact together. More information will be added regarding the diagram concerning the transaction after a discussion with the stakeholders.

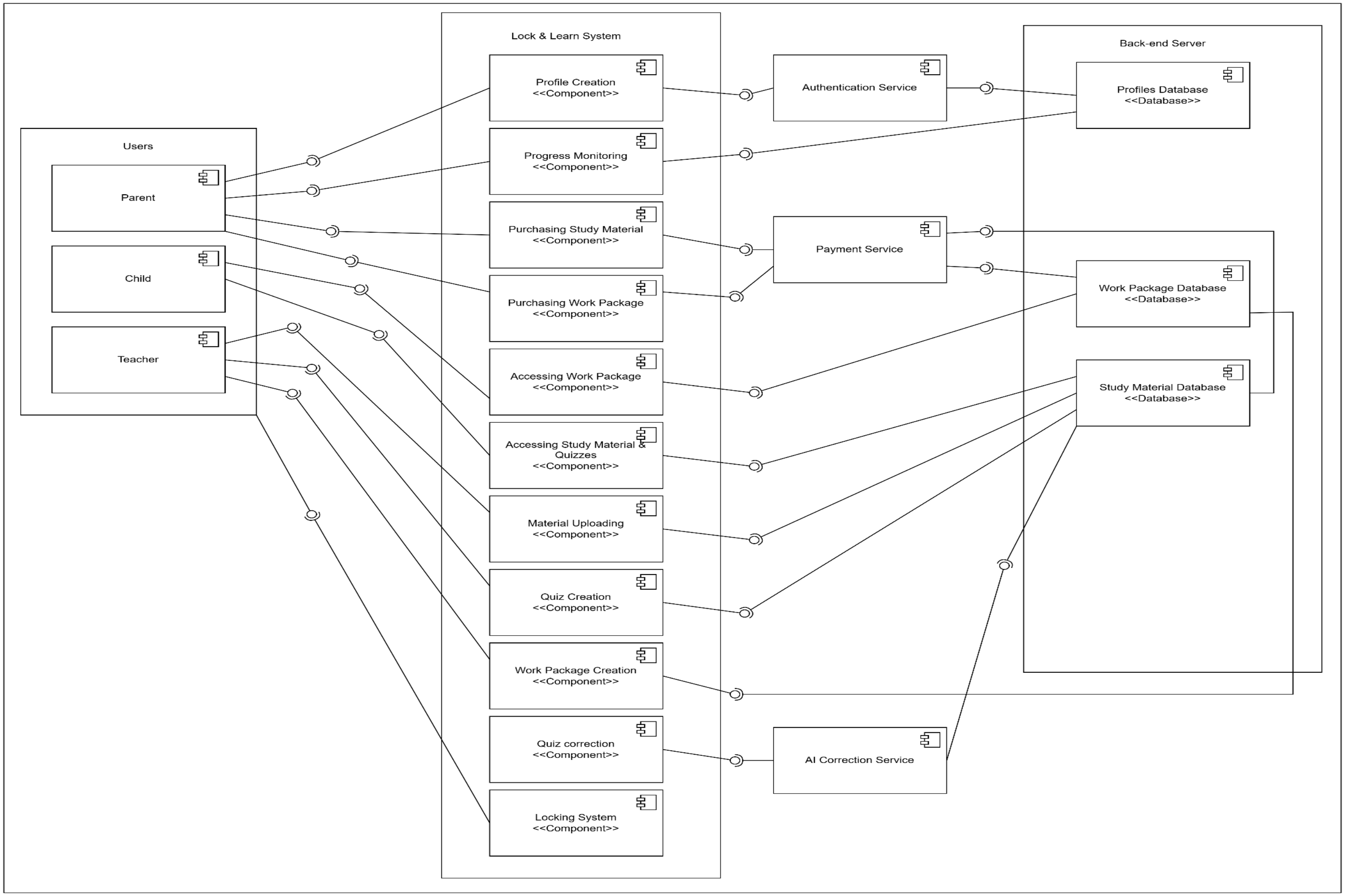

Figure 1: Component Diagram

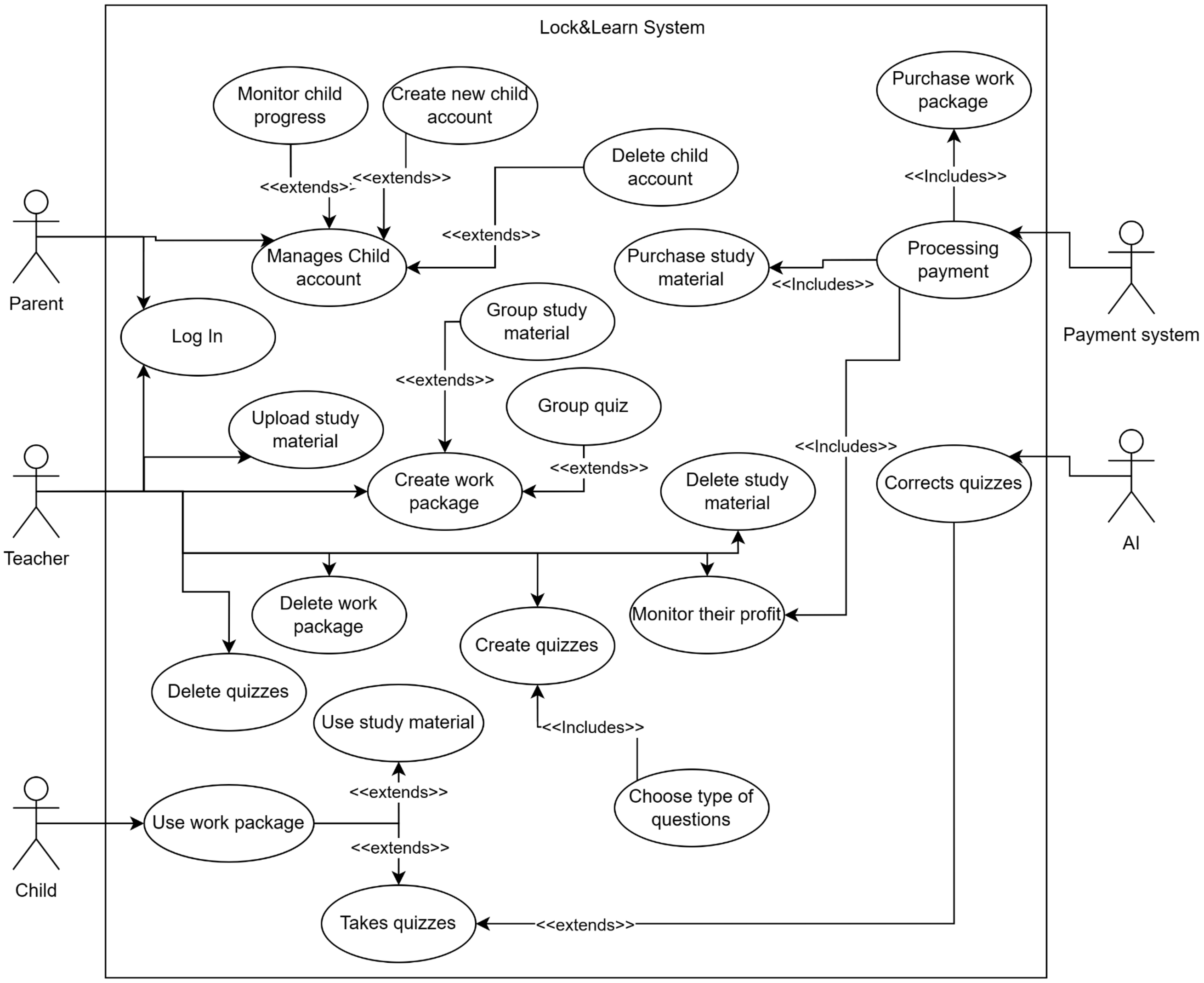

Figure 2: Use Case Diagram

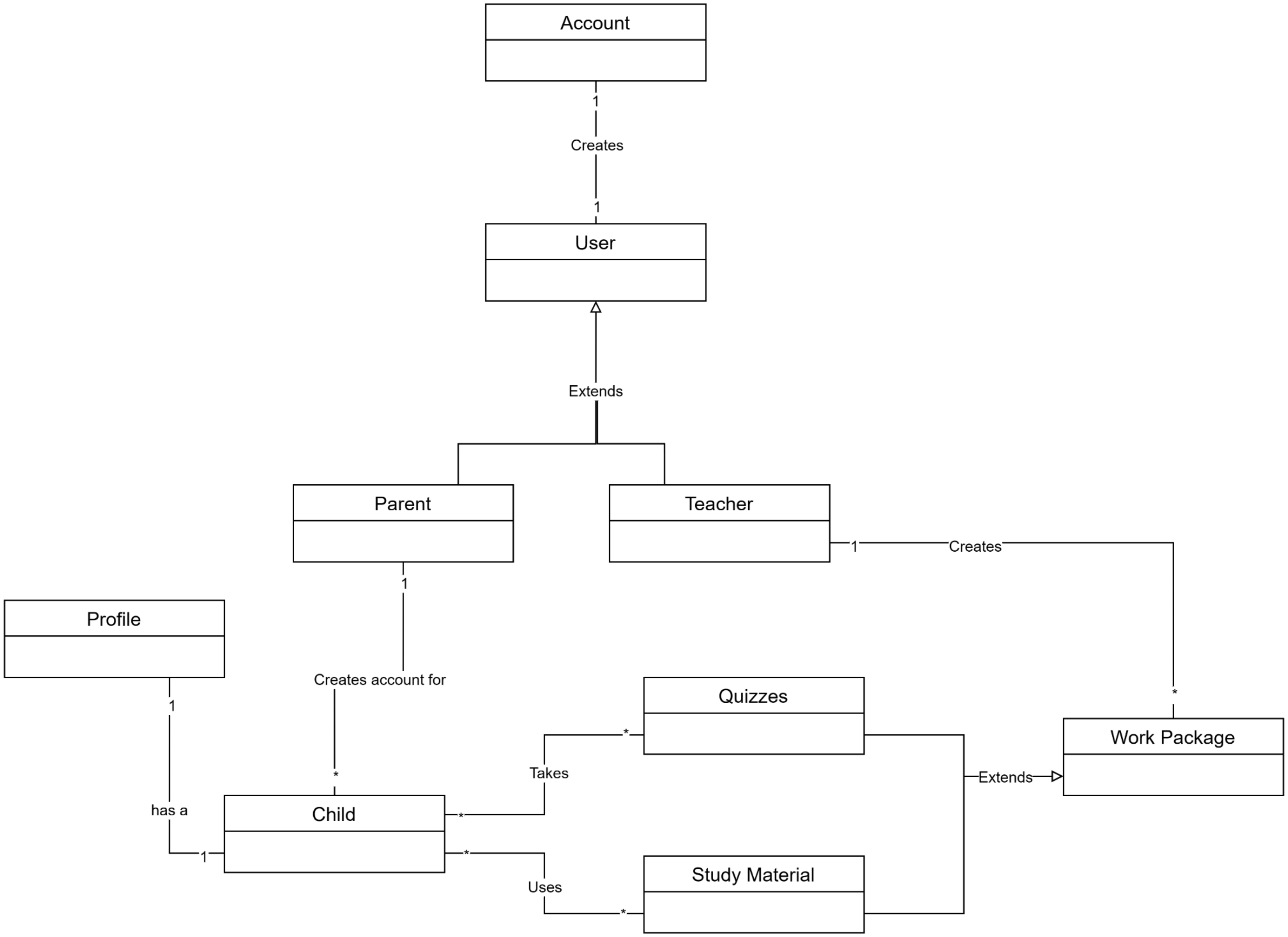

Figure 3: Domain Model Diagram

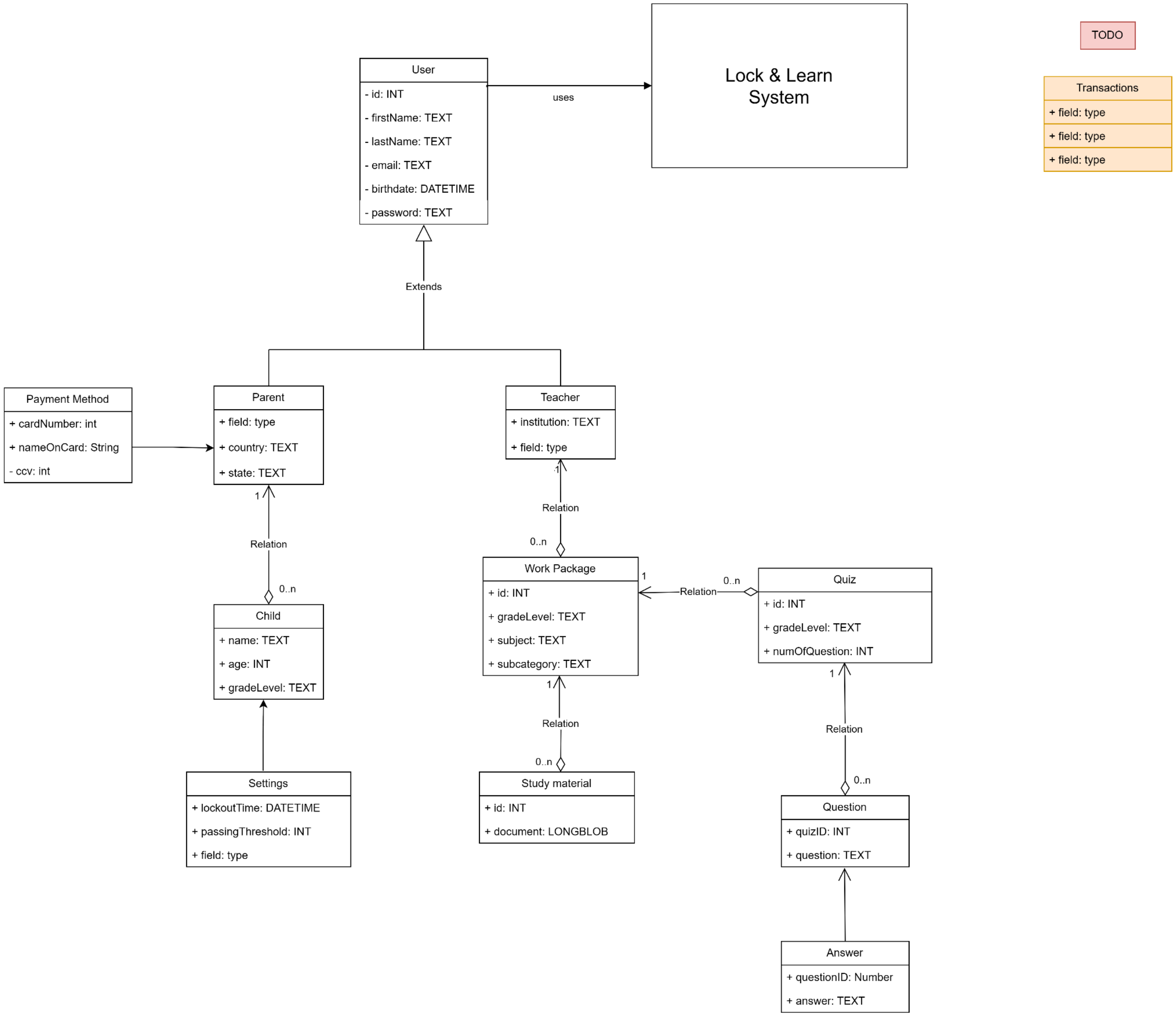

Figure 4: Class diagram

Infrastructure

We are using different frameworks, which suit best for the team and for developing a mobile and web application:

- React Native for the UI
- Javascript for the backend
- MongoDB for the database
- Docker for the container
- Jest for the unit and integration testing
- Detox for end-to-end testing

Name Conventions

We will be using the PascalCase naming convention. To see how it is used refer to the following link:
https://www.theserverside.com/definition/Pascal-case

Code

File path with clickable GitHub link	Purpose
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/screens/StudyMaterial/UploadScreen.js	The screen where tutors will upload their education material
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/screens/User/Child/AddChildScreen.js	Screen to add a child concerning the parent
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/screens/QuizMaterial/EditQuestion.js	The screen where tutors edit the questions in their quizzes
LockAndLearn/LockAndLearn/screens/StudyMaterial/ViewUploadedFilesScreen.js at main · RIGNITE/LockAndLearn	Screen for tutors to view their uploaded study material
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/screens/WorkPackage/DisplayWorkPackageContent.js	Screen for tutors to add and view study material and/or quizzes in their work package

Testing and Continuous Integration

Test File path with clickable GitHub link	What is it testing
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/tests/_test_studyMaterial/uploadScreen.test.js	These tests verify the functionality of the component, including the ability to select files using a file picker, trigger file uploads, and ensure that the uploadFilesHandler function correctly communicates with a server for file uploads.

https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/tests/_test_user/_test_child/AddChildScreen.test.js	This test file is for a React Native component called AddChildScreen. The first test ensures that the component allows the user to enter first name, last name, and grade, while the second test verifies that the component correctly submits child data when the "add child" button is pressed, including making an expected API call using the fetch function and checking the interactions with AsyncStorage and navigation.
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/tests/_test_quizMaterial/EditQuestion.test.js	These Jest tests verify the React Native EditQuestion component's functionality, including loading and populating fields with mock question data, updating questions on save button press, rendering and interacting with multiple-choice options, and ensuring that changes to inputs update state for short answer questions. The tests also cover scenarios such as handling non-200 response status and network errors during question fetching.
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/tests/_test_studyMaterial/viewUploadedFilesScreenFE.test.js	These Jest tests for the ViewUploadedFilesScreen component check various aspects of its functionality. The tests confirm the presence and clickability of elements such as the "Select files" button, the filter button, and checkboxes. Additionally, they validate the successful fetching and rendering of uploaded files, as well as the ability to delete a file, ensuring the correct API call and the disappearance of the deleted file from the screen.
https://github.com/RIGNITE/LockAndLearn/blob/main/LockAndLearn/tests/_test_workPackage/CreateWorkPackage.test.js	These Jest tests for the CreateWorkPackage component verify its rendering, interaction, and functionality. The tests include checks for the correct rendering of the create button, enabling the button when subject and grade are selected, updating of subcategories based on subject and grade selection, and successful

	navigation upon creating a work package. Additionally, the tests incorporate mocks for asynchronous operations, such as fetching data and navigation, to ensure the expected behavior of the component in various scenarios.

Description of continuous integration environment:
Create node.js.yml template · RIGNITE/LockAndLearn@f815c4c (github.com)

For the continuous integration, we are using GitHub's actions. Every merged pull request runs through continuous integration. This latter builds the code with the required dependencies and tests the added code/commits using Jest.

Release 1 Video

https://github.com/RIGNITE/LockAndLearn/blob/main/CourseAdmin/release1.mp4