

SOFTWARE ARCHITECTURE DOCUMENT

October 30, 2016

Software Design COMS3009

FindMeTutor Android Application

Proposed idea by:

Shaneel James-718840

Jadon Manilal-815050

Jared Naidoo - 719238

Krupa Prag - 782681

Nivek Ranjith - 802119

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms and Abbreviations	4
1.4	Overview	4
2	Architecture	5
2.1	Type of Architecture	5
2.2	Advantages of 2-Tier Architecture	6
2.3	Responsibilities of Layer 1 (Application Level):	6
2.4	Responsibilities of Layer 2:	6
2.5	Systems of interest	6
2.6	Supplementary Information	7
3	Architectural Goals and Constraints	8
3.1	Technical Platform	8
3.2	Transactions	8
3.3	Security	8
3.4	Reliability/Availability (failover)	9
4	Stakeholders	10
4.1	List of Stakeholders	10
5	Concern and Stakeholder Traceability	11
5.1	List of Concerns	11
5.2	Traceability Matrix	12
6	Views	12
6.1	Logical View	13
6.1.1	Class Diagram	14
6.1.2	Sequence Diagram	16
6.2	Development View	21
6.2.1	Package Diagram	21
6.2.2	Component Diagram	23
6.3	Process View	25
6.3.1	Activity Diagrams	25
6.3.2	Communication Diagrams	29

6.3.3	State Diagrams	32
6.4	Physical View	34
6.4.1	Deployment Diagram	34
6.5	Scenarios	35
6.5.1	Use Case Diagram	35

1 Introduction

1.1 Purpose

The purpose of a Software Architecture Document(SAD) is to provide a comprehensive architectural overview of the FindMeTutor system. It presents a number of different architectural views which depict different aspects of the system. It is intended to capture and depict the different architectural decisions that have been made by the development team.

1.2 Scope

The scope of the SAD is to depict the architecture of the FindMeTutor system. This includes the FindMeTutor Tutor Application, FindMeTutor Student Application and the backend system that allows FindMeTutor to operate.

1.3 Definitions, Acronyms and Abbreviations

UML: Unified Modeling Language

SAD: Software Architecture Document

1.4 Overview

This document provides the reader with a deeper understanding of the FindMeTutor system. It is organised into several views of the system. The document makes use of the Krutchten 4+1 model which includes the following views:

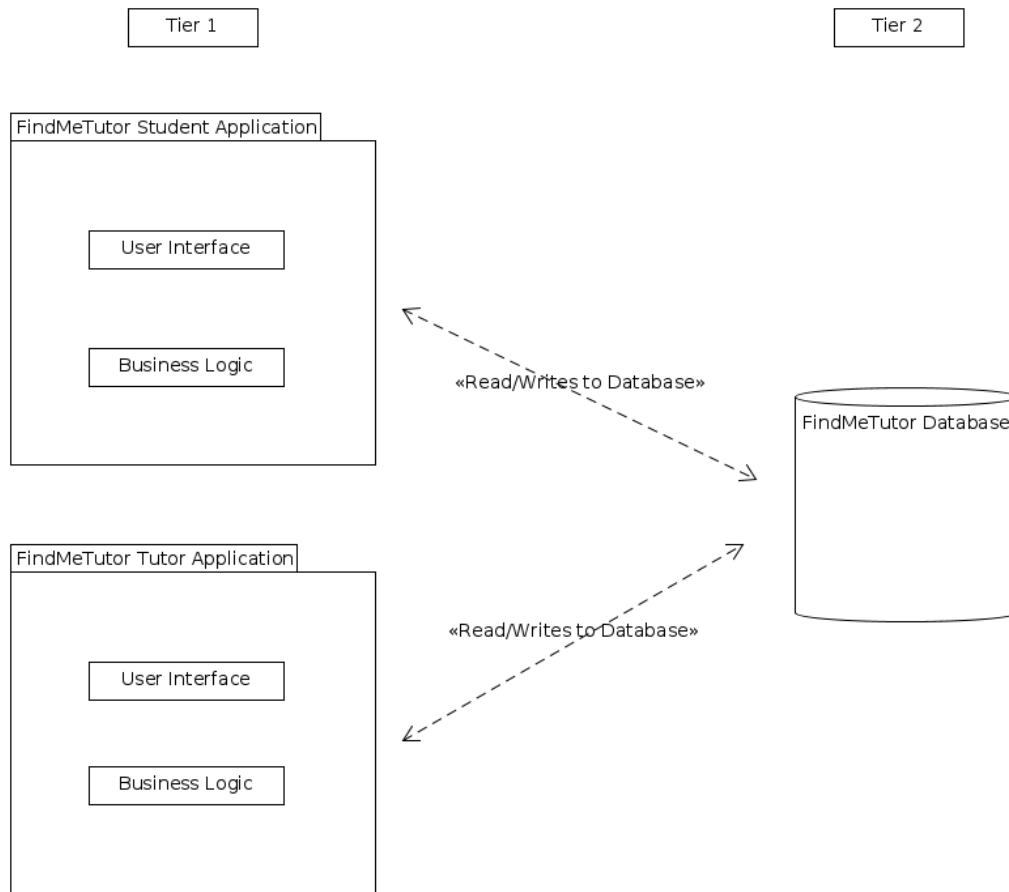
- Logical View
- Deployment View
- Process View
- Physical View
- Scenarios

2 Architecture

This section contains a information pertaining to the 2+1 architecture selected for the FindMeTutor system.

2.1 Type of Architecture

The FindMeTutor system makes use of 2-Tier architecture. The application layer contains the user-interface and the business logic. The second layer contains the database ,which stores the user data for the FindMeTutor system. Below is a diagram which depicts the architecture of the system:



2.2 Advantages of 2-Tier Architecture

- Allows the application to be easily developed due to the architectures simplicity.
- Maximum user satisfaction can be ensured due to accurate and fast prototyping of applications.
- Fast development time due to fast prototyping and testing.
- Database logic and business logic are physically close, which offers higher performance, (i.e.no intermediate layers between user interface layer and database layer).

2.3 Responsibilities of Layer 1 (Application Level):

- The application layer contains the business logic and user interface components of the system. Layer 1 is responsible for conveying FindMeTutor services to the client. Layer 1 also contains the administration layer used to add funds to a student/tutor account as well as general configuration by an approved administrator.

2.4 Responsibilities of Layer 2:

- Contains the data for the system. Allows multiple users to access the same data simultaneously.
- Primarily regarded as the backbone of the system.

2.5 Systems of interest

The main systems of interest is FindMeTutor which contain the following two subsystems:

- FindMeTutor Student Application
- FindMeTutor Tutor Application
- FindMeTutor SQL Server

2.6 Supplementary Information

The FindMeTutor system comprises of two main components: The FindMeTutor Student Application and the FindMeTutor Tutor Application. The student application is used by the students to manage anything related to a student account and the tutor application is used by the tutors to manage anything related to the tutor account. Requests for a tutor are made using the student application and responses from tutors as to whether to accept or reject the requests are made using the tutor application.

3 Architectural Goals and Constraints

The following lists the goals and constraints of the FindMeTutor system.

3.1 Technical Platform

The FindMeTutor application will be deployed on the Android mobile platform. The application is made up of two parts the user interface and the business logic. The application then communicates with a server of which the entire systems data is stored. The database system consists of an Ubuntu server running a MySql database.

3.2 Transcations

- Student Transcation
The student would pay funds into the FindMeTutor bank account. The students email address would be used as a reference. As soon as the funds have cleared, the student is provided with a credit on the FindMeTutor system. The student can then use the credit to book and pay for tutoring sessions.
- Tutor Transaction
Upon a successful tutoring session. The FindMeTutor system credits the tutor with the amount agreed upon. All of the tutors credits are added up at the end of the month and then paid out to the tutor by means of a bank transfer.

3.3 Security

The FindMeTutor system takes security very seriously. During a tutoring session the systems logs the exact GPS location of both the tutor and student at all times. Should a location not be logged by both the student and tutor. The the tutor will not receive payment and the student will receive a poor rating. Further communication with FindMeTutor will be required by the tutor to receive payment.

3.4 Reliability/Availability (failover)

The FindMeTutor system makes use of a failover or backup server in the event that something should go wrong. Furthermore FindMeTutor has acquired additional capacity so as to anticipate an uptake in system usage and as a result ensure the system is reliable at all times.

4 Stakeholders

This section lists the various organisations who are concerned with the project.

4.1 List of Stakeholders

- Development Team (Stakeholder 1)
The Development team is concerned with the implementation of the system, they want to develop the system in the best possible way in the shortest time.
- Analysts (Stakeholder 2)
The analysts are concerned with the design of the system, and the functionality of the system once implemented.
- Lecturer (Stakeholder 3)
The Software Design lecturer is concerned with the progress of the design and implementation of the system.
- Students (Stakeholder 4)
Students want a system which will address their needs as well as a functioning system which is favourable towards them finding tutors.
- Tutors (Stakeholder 5)
Tutors want a system which addresses their needs as well as a functioning system which is favourable towards them as they wish to earn an income from FindMeTutor.

5 Concern and Stakeholder Traceability

5.1 List of Concerns

This section identifies concerns relating to the architecture of the FindMeTutor system.

- Purpose of the system (Concern 1)
The main purpose of the system is to provide a connection between a student and a tutor. A potential concern here would be a slow uptake or worse students are not interested in using the application.
- Suitability of the architecture (Concern 2).
The 2-tier architecture selected allows the developers to modify different aspects of the system without affecting other important aspects of the system. For example: if the business logic should change, we can simply modify the logic layer of the application while not affecting the user interface or database layers of the system.

A potential concern here would be that the architecture above does not fully support certain aspects of expansion should the system experience growth and need to be expanded

- Feasibility (Concern 3)
Should the system not be feasible. i.e. High costs of keeping the system running. Developers and server costs for the system would be high and FindMeTutor would need some sort of revenue to keep the system running. Scalability is another issue, we need to scale in order to create revenue. A concern would be that the system does not create revenue for the upkeep of the system and hence the system is no longer feasible.
- Evolution of the FindMeTutor system (Concern 4)
The system will change as time progresses, the system is largely based on the users of the system. Thus as the users change, the system will have to change to match the users. A concern here would be that the system does not meet the users' demands/preferences and ultimately fails.

5.2 Traceability Matrix

The following table depicts the relationship between the concerns listed above and the various stakeholders of the FindMeTutor system listed in the previous section.

Stakeholder 1 - Development Team

Stakeholder 2 - Analysts

Stakeholder 3 - Lecturer

Stakeholder 4 - Students

Stakeholder 5 - Tutors

Table 5.1.1					
	Stakeholder 1	Stakeholder 2	Stakeholder 3	Stakeholder 4	Stakeholder 5
Concern 1		X	X	X	X
Concern 2	X	X			
Concern 3		X			
Concern 4	X	X		X	X

6 Views

Description of the different views:

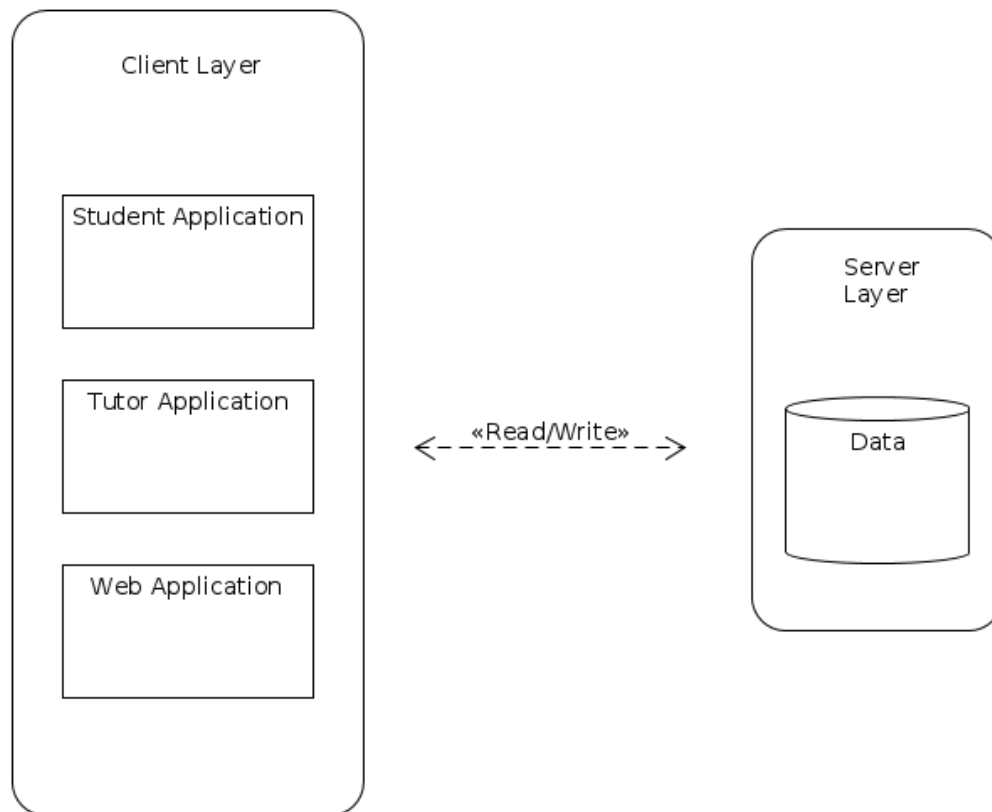
- Logical View The logical view is concerned with the functionality that the system provides to end users. The view includes diagrams such as a class diagram and sequence diagrams.
- Development View The development view illustrates a system from a programmers perspective and is concerned with the software management. Diagrams include a package diagram and component diagram.
- Process View The process view deals with the dynamic aspects of the system, explains the systems processes and how they communicate, and focusses on the runtime behaviour of the system. This view addresses concurrency, distribution, integrators, performance, and scalability.
- Physical View The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between theses componenets.

6.1 Logical View

The Logical View focuses on realising the functionality of the FindMeTutor system. This view addresses the concerns of the end-user by realising the functionality of the system.

The FindMeTutor application uses a 2-Tier architecture.

Architecture Diagram: FindMeTutor



Responsibilities of Layer 1 (Application Level):

- The application layer contains the business logic and user interface. The first layer is responsible for conveying FindMeTutor services to

the user. It also contains the administration layer used to add funds to a student/tutor account as well as general configuration by an approved administrator. This layer runs on the Android operating system, the application is installed on the users mobile device.

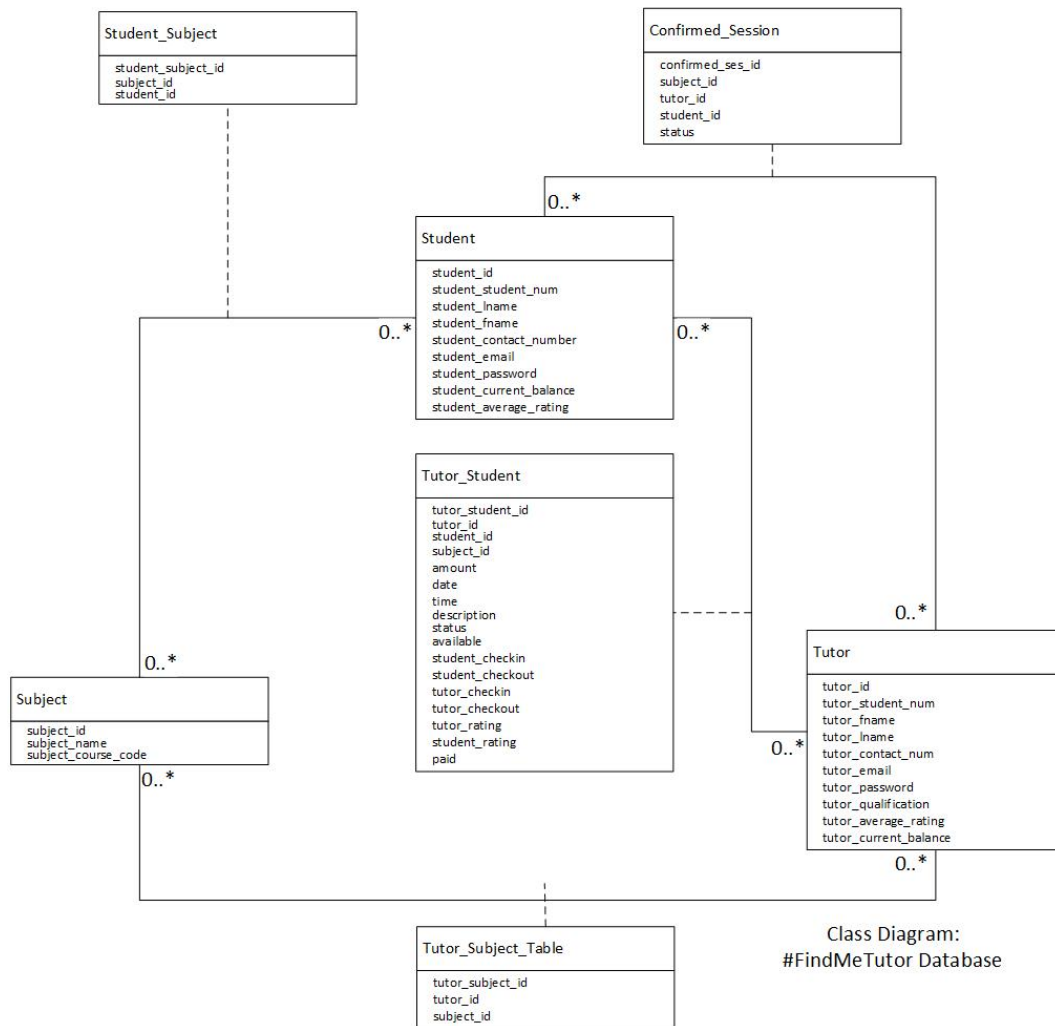
Responsibilities of Layer 2:

- Stores the data for the system. Layer 2 allows multiple users to access the systems data simultaneously. A backup/failover server for Layer 2 is available at all times.

6.1.1 Class Diagram

The class diagram below describes the structure of the FindMeTutor system. Showing the system classes, their attributes and relationships to one another.

The SQL database is stored on a linux based server hosted by Amazon Web Services apart from Amazon, FindMeTutor has ensured that we have a backup server hosted on the Microsoft Azure platform.

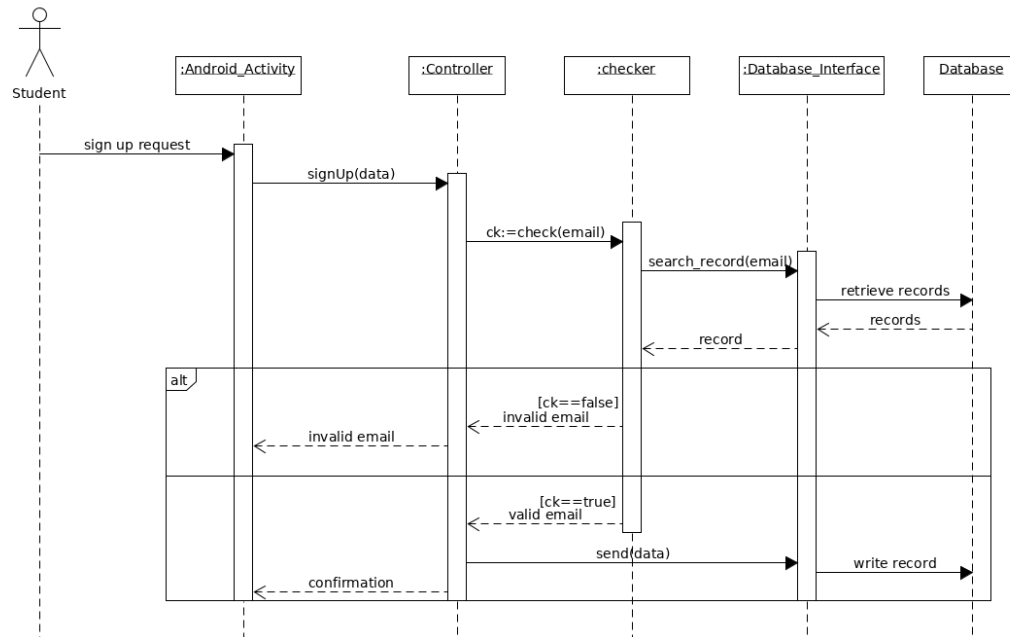


6.1.2 Sequence Diagram

Sequence Diagrams show the sequence of messages passed between objects of the FindMeTutor system over time.

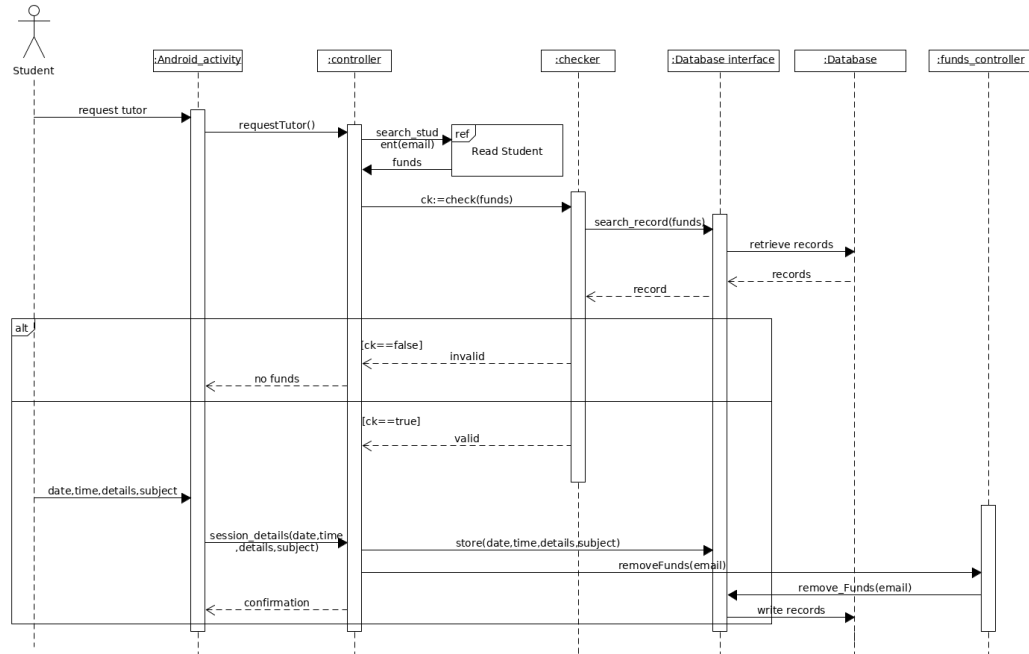
Create Student Sequence Diagram:

The process of registering/creating a new student account on FindMeTutor using the student application.



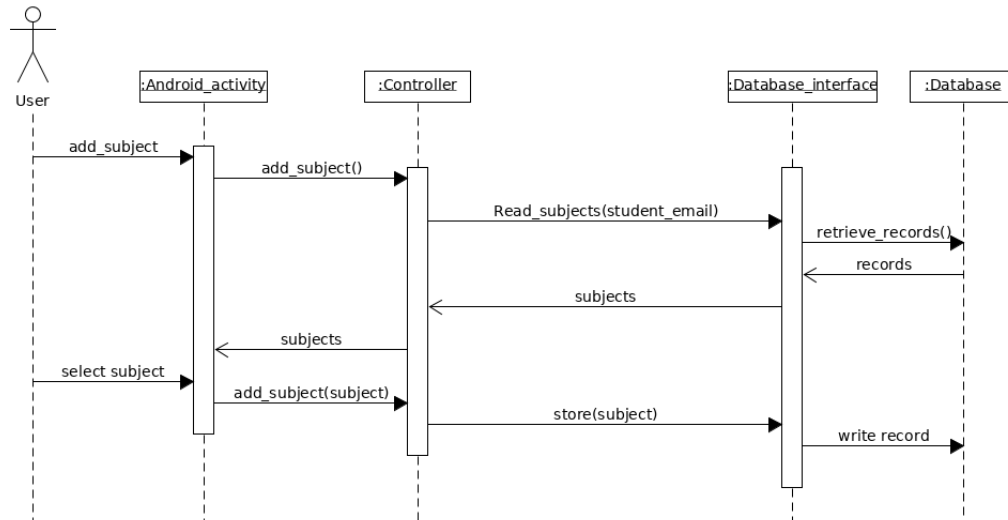
Request Tutor Sequence Diagram:

The process of requesting a tutor on FindMeTutor using the student application.



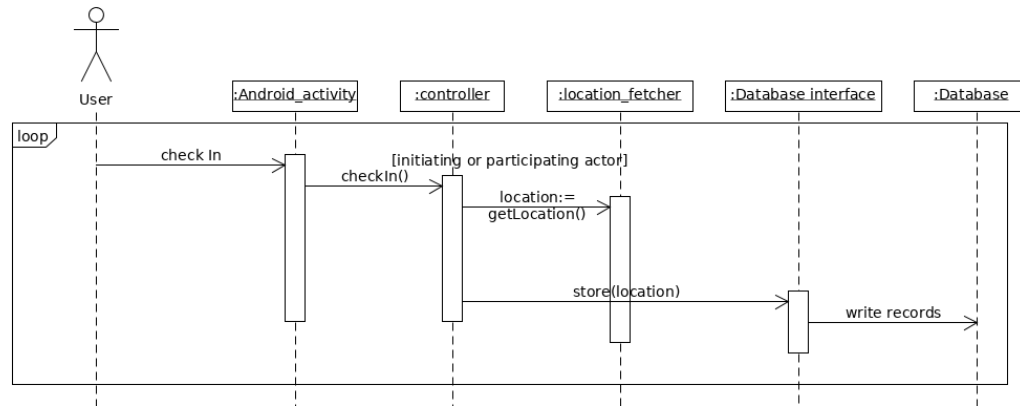
Add Subject Sequence Diagram:

The process of adding a new subject on FindMeTutor using the student application.



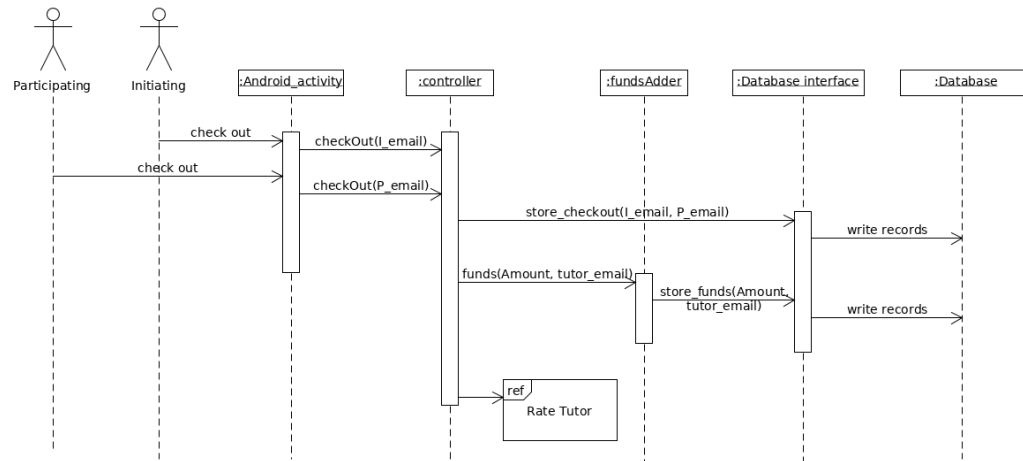
Check In Sequence Diagram:

The process of checking in a tutorial session.



Check Out Sequence Diagram:

The process of checking out of a tutorial session.



6.2 Development View

The Development view outlines the components that are used to assemble the physical system. The view addresses the concerns of stakeholders concerned with the development of system such as developers.

Find Me Tutor is a mobile application built to run on Google's Android mobile operating system. Android is the currently the biggest mobile operating system and we believe that it is the perfect platform for FindMeTutor.

There are two mobile applications that make up FindMeTutor. The first being the student application and the second being the tutor application. Students make use of the student application while Tutors make use of the tutor application.

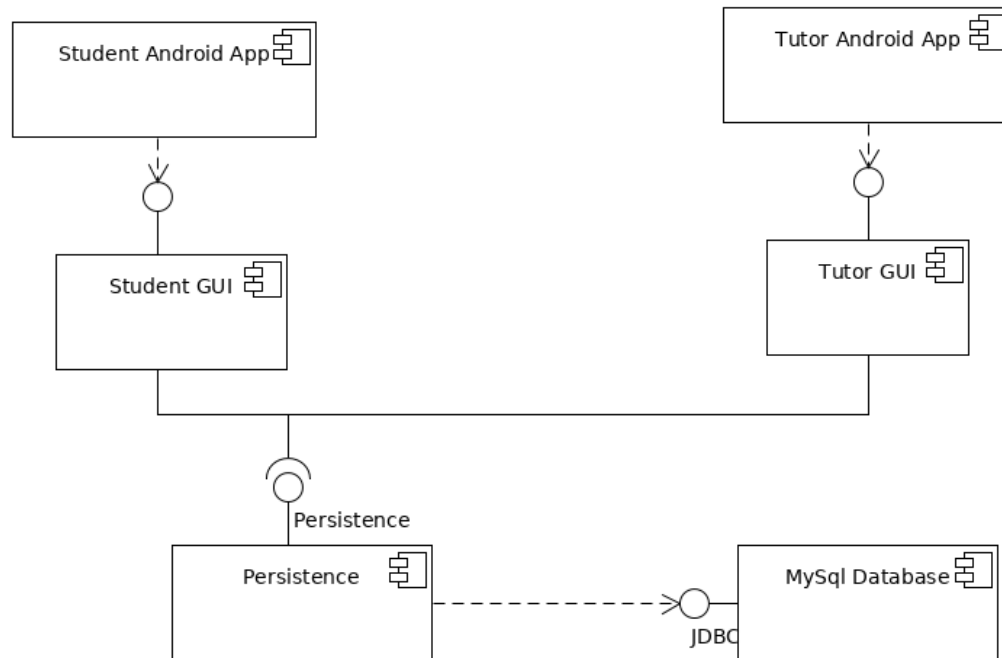
6.2.1 Package Diagram

A package diagram depicts the dependencies the packages of the FindMeTutor system. Below is package diagrams of the FindMeTutor system:



6.2.2 Component Diagram

A component diagram describes the components used to achieve the functionality of the FindMeTutor system.



Description of components above:

- **Student Android App**
The application is installed on a students mobile device. The application is used for requesting tutor services, payment of tutoring services, the addition of subjects, to decline/accept tutoring sessions and other student related requirements.
- **Tutor Android App**
The application is installed on a students mobile device. The application is used for confirming/declining tutoring sessions, creating a tutor account and general tutor related requirements.
- **Student and Tutor GUI**
This is the user interface designed for the two applications. The two user interfaces look similar and are designed in Android for a mobile phone.

- MySQL Database
The storage system used to store the data of the FindMeTutor system.
Utilizing the latest in cloud storage technology.

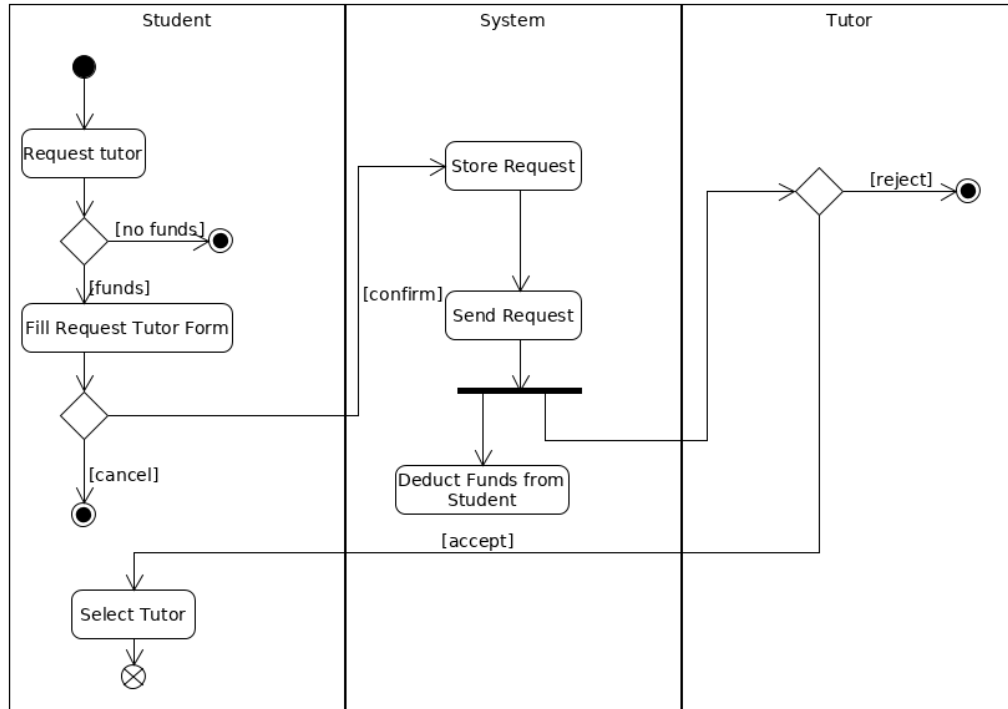
6.3 Process View

The process view considers the non-functional aspects of the FindMeTutor system. It addresses the concerns of the stakeholders concerned with the design of the system.

6.3.1 Activity Diagrams

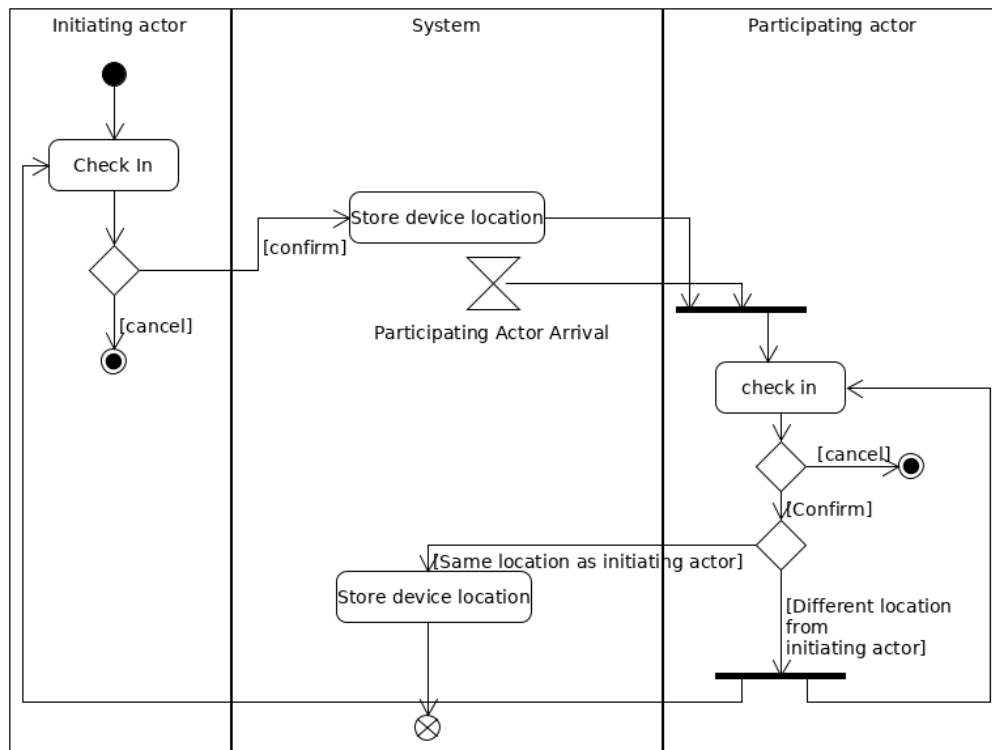
An activity diagram depicts the the flows of the system and business logic with actions.

Request Tutor:

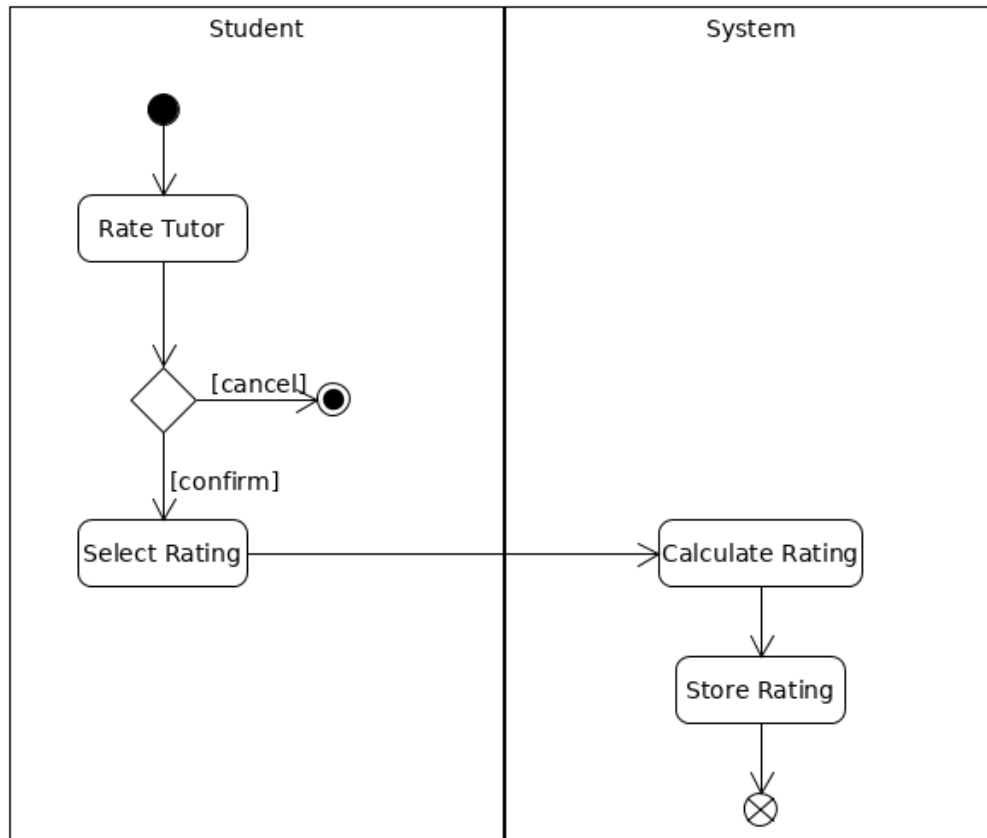


Check In:

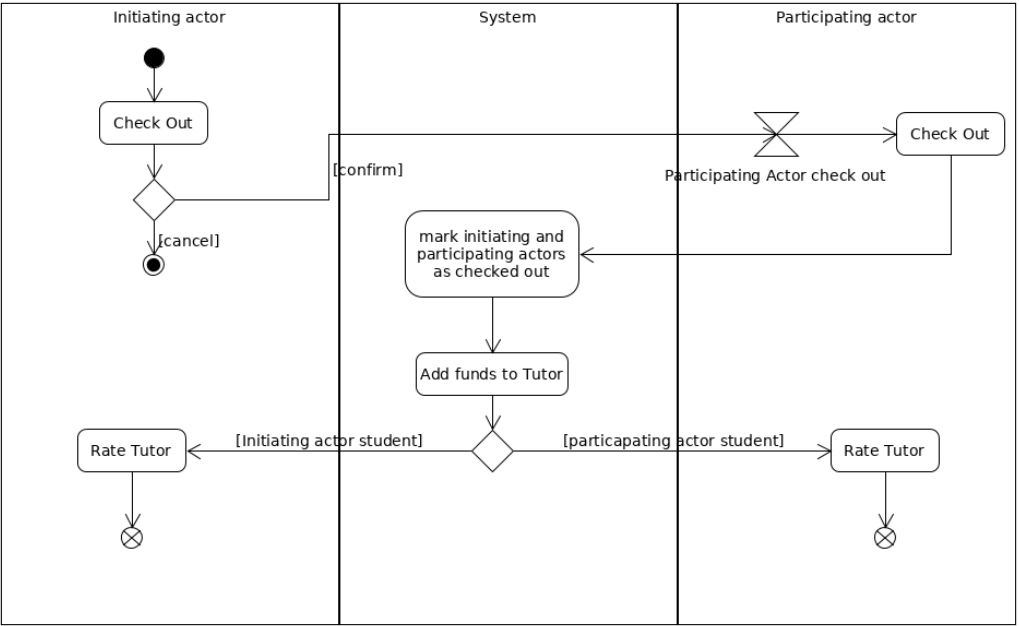
The initiating actor is either a student or a tutor where a participating actor refers to a tutor or student.



Rate Tutor:



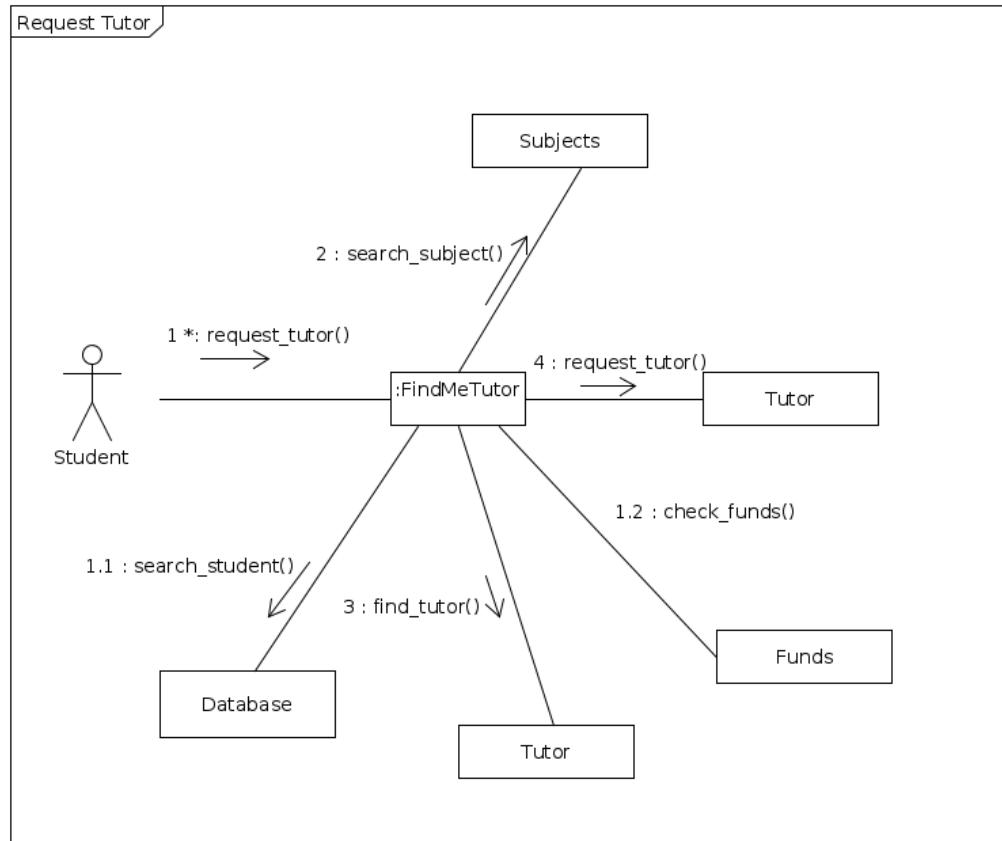
Check Out:



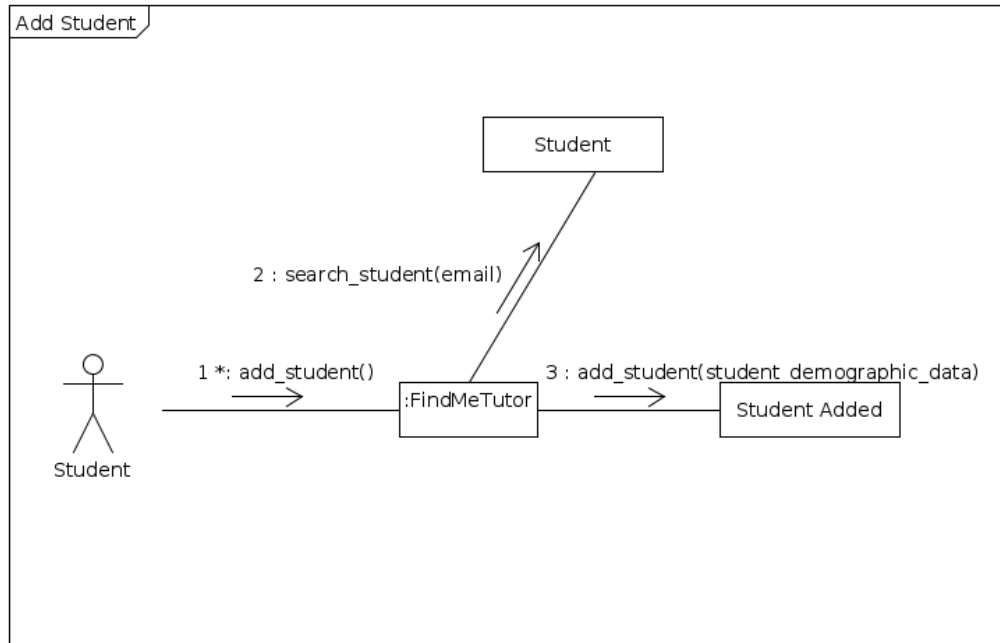
6.3.2 Communication Diagrams

The communication diagrams for Request a Tutor, Create a Student Account and to add a Subject on the system are below:

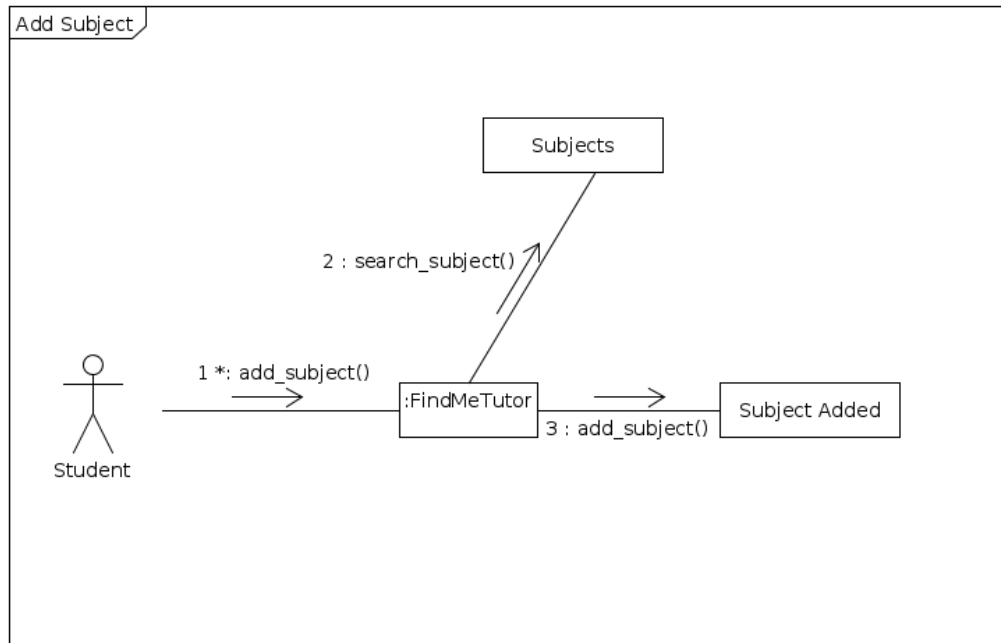
Request Tutor



Create Student



Add Subject



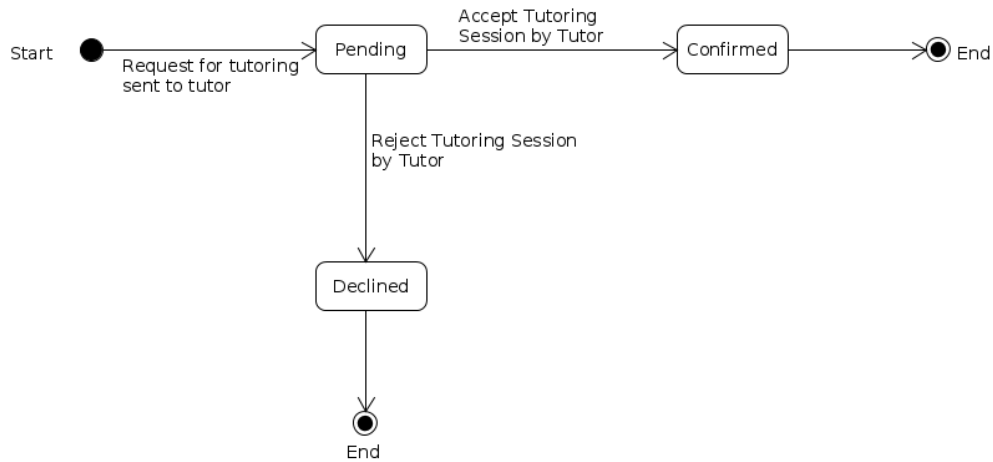
6.3.3 State Diagrams

The most important state diagrams for the FindMeTutor system are listed below, they depict the states in which various attributes of the database change.

State Diagram: "Status" in the Confirmed Session Table

The state is changed when the tutor has accepted the tutoring session.

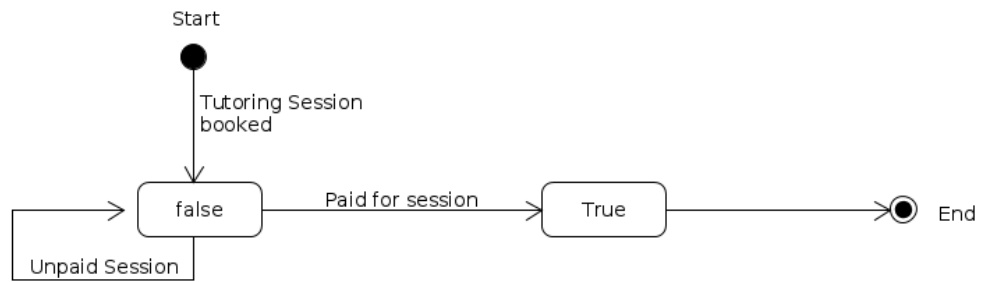
State Diagram: "Status" in Confirmed_Session



State Diagram: "Paid" in the Tutor Student Table

The state is changed when the student has paid for the tutoring session.

State Diagram: "Paid" in Tutor_Student Table



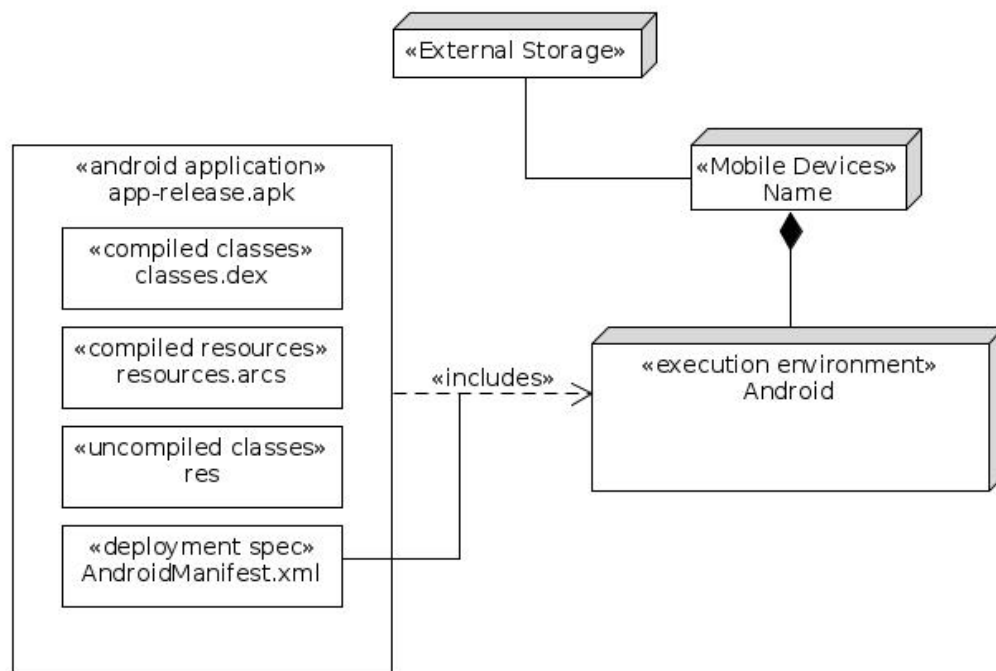
6.4 Physical View

The physical view describes the physical locations of software, the scalability of the system, the deployment and installation.

The server side of FindMetutor makes use of an SQL server, we have selected Amazon Web Services as our primary provider. We also have a backup server on the Microsoft Azure platform.

On the user side we have the mobile device running the latest version of Google's Android operating system. The application(s) are then installed on Android.

6.4.1 Deployment Diagram



6.5 Scenarios

The Scenarios view address concerns of all stakeholders, this view helps show how the system is to be used.

6.5.1 Use Case Diagram

The use case diagram describes the behaviour of the system from the view of it's users.

