SOFTWARE ARCHITECTURE DOCUMENT

October 16, 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 **3**

2 Architecture **4**

3 Architectural Goals and Constraints **6**

4 Stakeholders **7**

5 Concern and Stakeholder Traceability **8**
5.1 Concerns 8
5.2 Traceability Matrix 9

6 Views **10**
6.1 Logical View 10
6.1.1 Class Diagram 11
6.1.2 Sequence Diagram 12
6.2 Development View 14
6.2.1 Package Diagram 14
6.2.2 Component Diagram 16
6.3 Process View 17
6.3.1 Activity Diagrams 17
6.3.2 Communication Diagrams 20
6.4 Physical View 22
6.4.1 Deployment Diagram 22
6.5 Scenarios 23
6.5.1 Use Case Diagram 23

1 Introduction

- Purpose
  The Software Architecture Document (SAD) provides a comprehensive architectural overview of the FindMeTutor system. It presents a number of different architectural views to depict different aspects of the system. It is intended to capture and depict the different architectural decisions which have been made on the system.

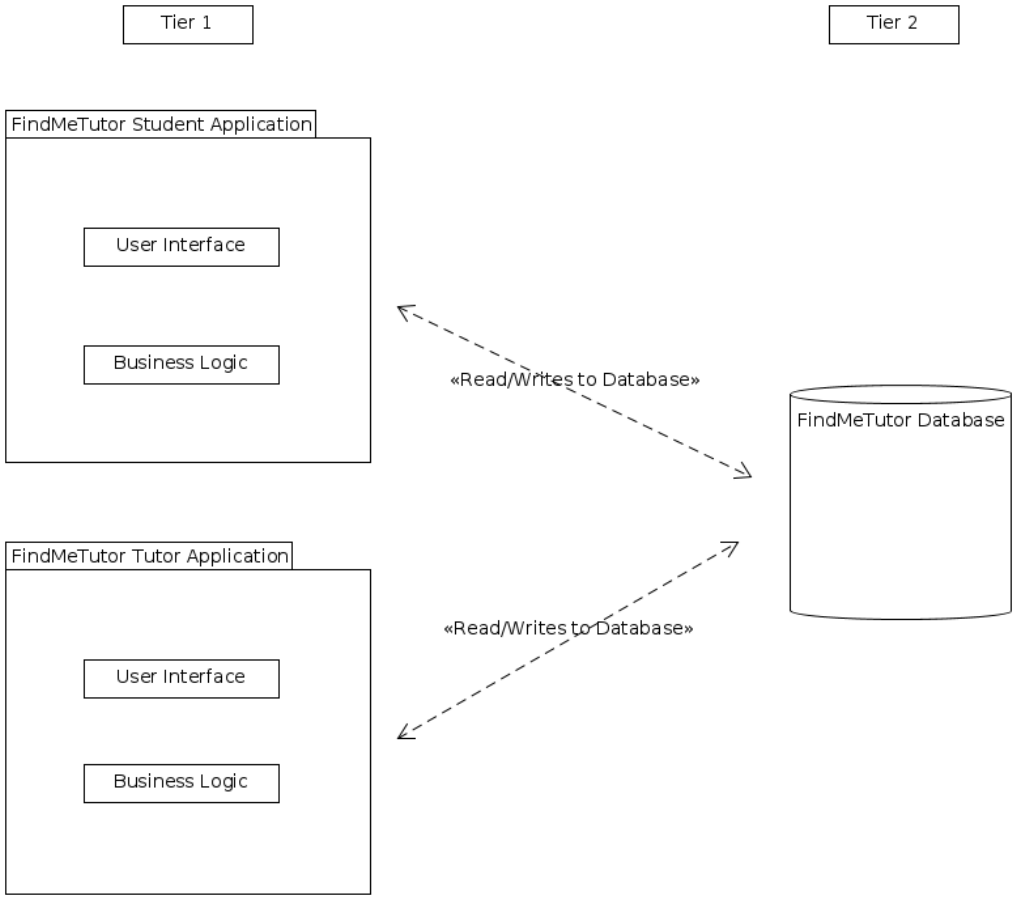
- Scope
  The scope of the SAD is to depict the architecture of the FindMeTutor. This includes the FindMeTutor tutor application, FindMeTutor student application and the database (backend systems) which allow FindMeTutor to operate.

- Definitions, Acronyms and Abbreviations
  UML: Unified Modeling Language
  SAD: Software Architecture Document

2 Architecture

This section contains a description pertaining to the architecture used in the FindMeTutor system.

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



- Advantages
  - Allows the application to be easily devloped due to simplicity.

  - Maximum user satisfaction can be ensured due to accurate and fast prototyping of applications
  - Database logic and business logic is physically close, which offers higher performance.

- Responsibilities of Layer 1:
  - Contains the businnes logic and user interface all built into the application. The first layer is responsible for all actions with the client. Contains the administration layer used to add funds to a student/tutor account as well as general configuration.
- Responsibilities of Layer 2:
  - Contains the data for the system.Allows multiple users to access the same data simultaneously.
- Systems of interest
  The main systems of interest are the FindMeTutor which contain the following two subsystems:
  - FindMeTutor Student Application
  - FindMeTutor Tutor Application
- 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 are made using the tutor application.

3 Architectural Goals and Constraints

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

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

- 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 this 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.

- 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 until they communicate directly with FindMeTutor.

- 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 made significant network improvements so as to anticipate an uptake in system usage and as a result ensure a stable system.

4 Stakeholders

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

- Development team (Stakeholder 1)
  The Development team are concerned with the implementation of the system, they want to develop the designed system.

- Analysts (Stakeholder 2)
  The analysts are concerned with the design of the system, and the proper functionality of the system.

- Lecturer (Stakeholder 3)
  The Software Design lecturer is concerned with the progress of the designing and implementation of the system.

- Students (Stakeholder 4)
  Students want a system which address their needs as well as a proper functioning system which dis favourable towards them.

- Tutors (Stakeholder 5)
  Tutors want a system which address their needs as well as a proper functioning system which is favourable towards them.

5 Concern and Stakeholder Traceability

5.1 Concerns

This section identifies concers 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 looking for a tutor and a tutor looking to earn money by tutoring a student.A potential concern here would be a slow uptake or worse students not interested in using the system

- Suitability of the architecture (Concern 2).
  The three 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 businness logic should change, we can simply modify the logic layer of the system while not affecting the user interface or database layers of the system.

  A potential concern here would be that the architecture above does not full 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. Scability 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 progress, 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 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.

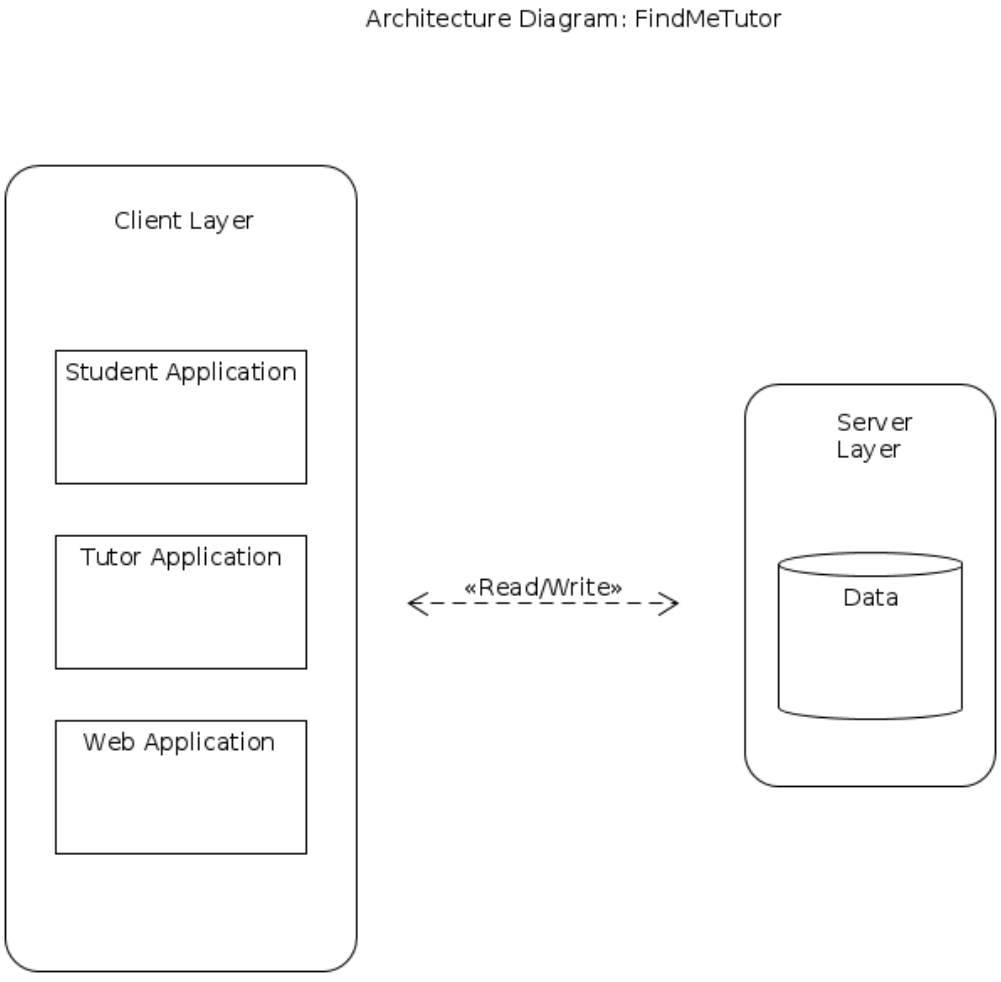
Table 5.1.1					
	Stakeholder 1	Stakeholder 2	Stakeholder 3	Stakeholder 4	Stakeholder 5
Concern 1		X	X	X	X
Concern 2	X	X			
Concern 3		X			
Concern 4	X	X		X	X

6 Views

6.1 Logical View

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

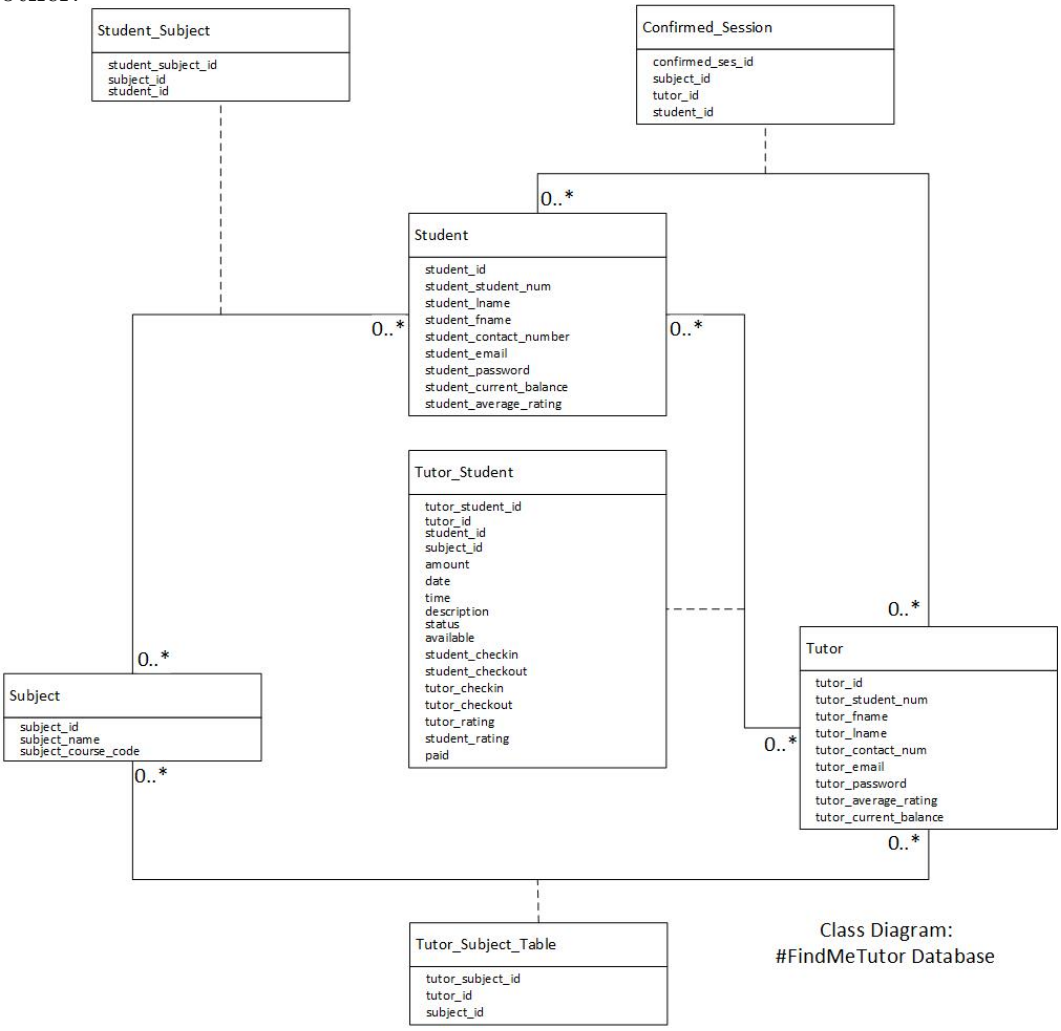
The FindMeTutor application uses a 2-Tier architecture.



Put in diagram similar to above of sequence of how to request a tutor.

6.1.1 Class Diagram

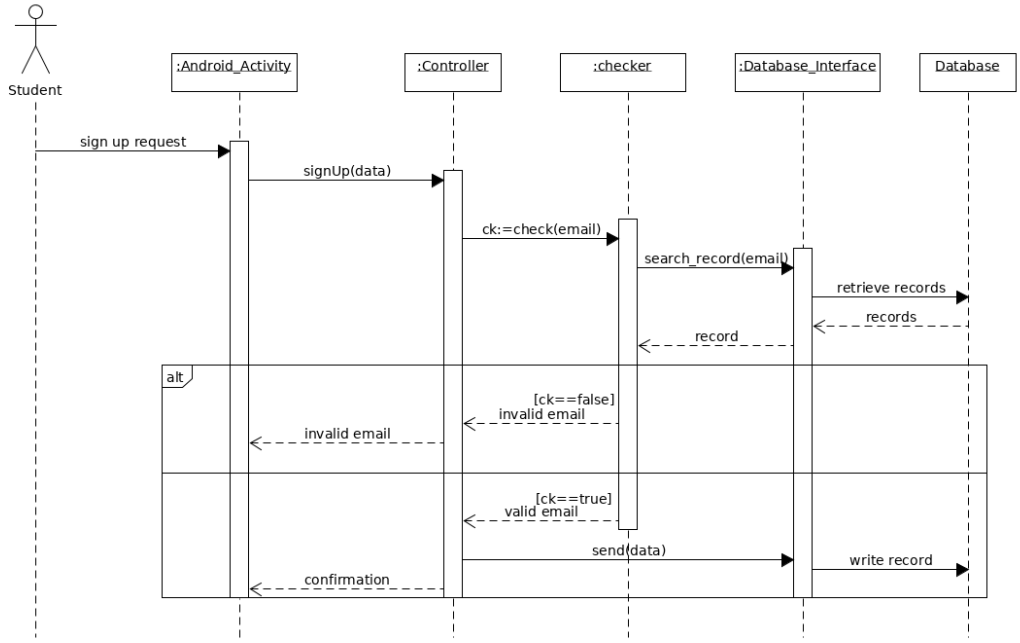
The class diagram below describes the structure of the FindMeTutor system ny showing the system classes, their attributes and relationships to one another.



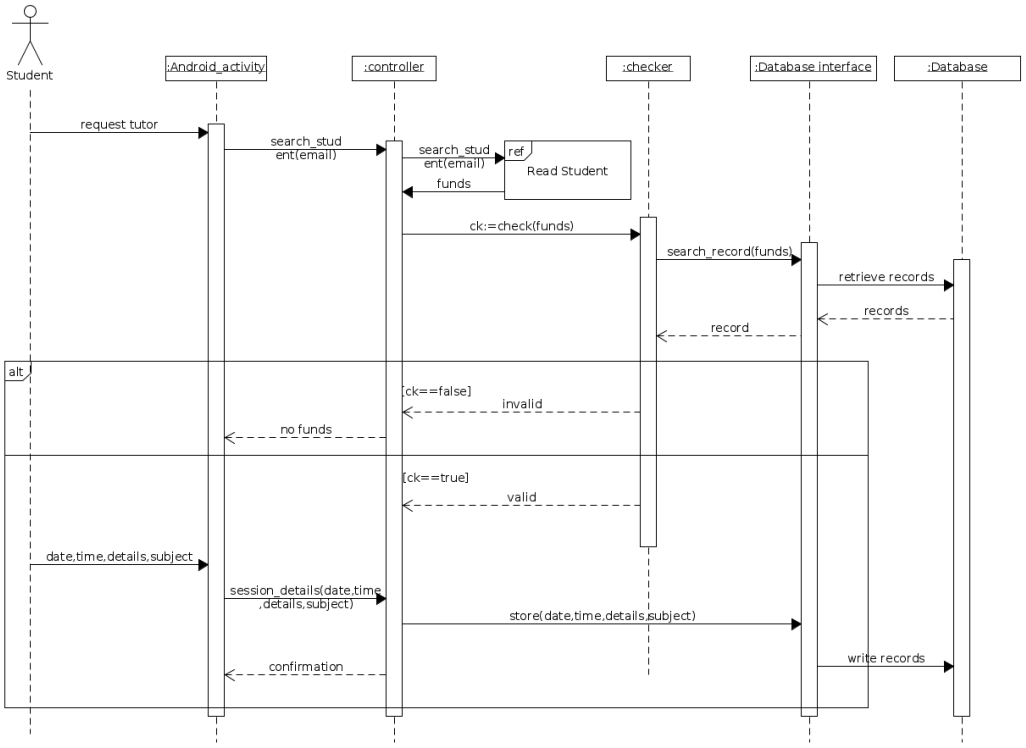
6.1.2 Sequence Diagram

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

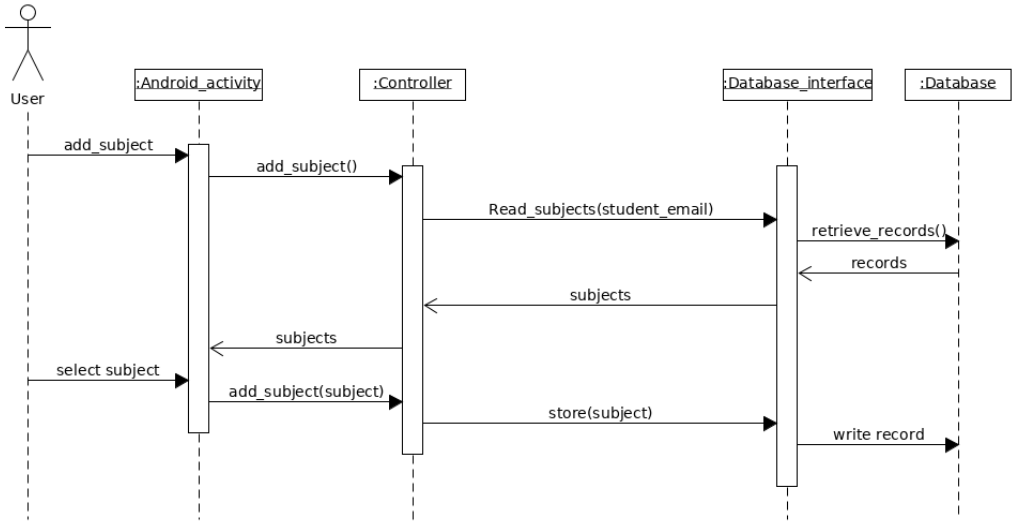
Create Student Sequence Diagram:



Request Tutor Sequence Diagram:



Add Subject Sequence Diagram:



6.2 Development View

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

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:


com
example
jared
findmetutor
AccountSettingsActivity
addSubject
AddSubsActivity
addtotutor_student
AsyncResponse
CardViewAdapter
FragmentDrawer
FriendsFragment
getallsubjects
GetLocation2
getSessions
getsubject
getTutors
HomeActivity
HomeFragment
JSSEProvider
ListViewAdapter
login
LoginActivity
NavDraweritem
NavigationDrawerAdapter
register
RegisterActivity
RequestActivity
requestSession
selectTutor
Session
Settings
SettingsFragment
student_checkin
student_checkout
Subjects
SubjectsAddViewAdapter
SubjectSettingsActivity
SubjectsViewAdapter
TutorListFragment
Tutors
TutorStudentFragment
TutorsViewAdapter
updateProfile
UploadToServer
tutor
AccountSettingsActivity2
addSubject
AddSubjectActivity
addtotutorstudent
CardViewAdapter
CardViewNotificationsAdapter
Event
Events_notifications
FragmentDrawer
FriendsFragment
GetLocation
GetLocation2
getsubject
GmailSender
HomeActivity
HomeFragment
JSSEProvider
ListViewAdapter
login
LoginActivity
NavDrawerItem
NavigationDrawerAdapter
Notifications
NotificationsActivity
NotificationsFragment
Pending
PendingAdapter
PendingSessionsActivity
register
RegisterActivity
RequestActuvuty
SendMailTask
Settings
SettingsFragment
Subjects
SubjectsAddViewAdapter
SubjectSettingsActivity
SubjectsViewAdapter
tutor_addsubject
tutor_AsyncRespnse
tutor_cancel
tutor_checkin
tutor_checkout
tutor_checkRegister
tutor_confirm
tutor_decline
tutor_getallsubjects
tutor_getnotification
tutor_getPending
tutor_getSessions
tutor_getsubject
tutorNotificationsAsync
tutor_paid
tutor_rateStudent
tutor_removeSubject
tutor_Sessions
tutor_updateAccount
tutor_updateStudentRating
UploadToServer

6.2.2 Component Diagram

A component diagram describes the components used to achieve the functionality of the system.
ADD COMPONENT DIAGRAM

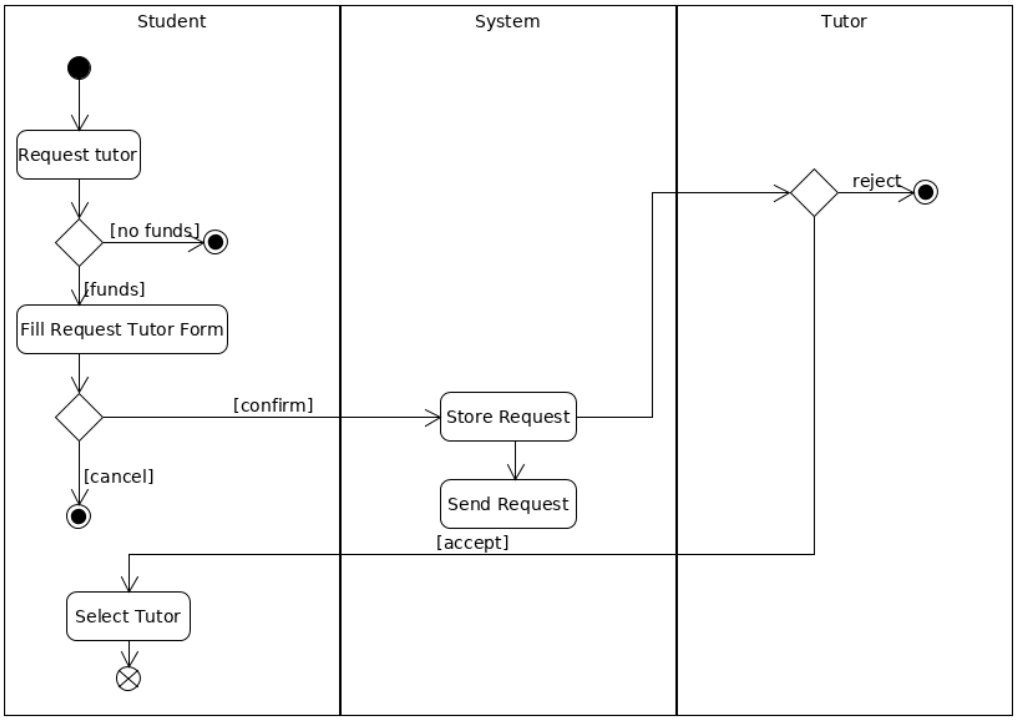
6.3 Process View

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

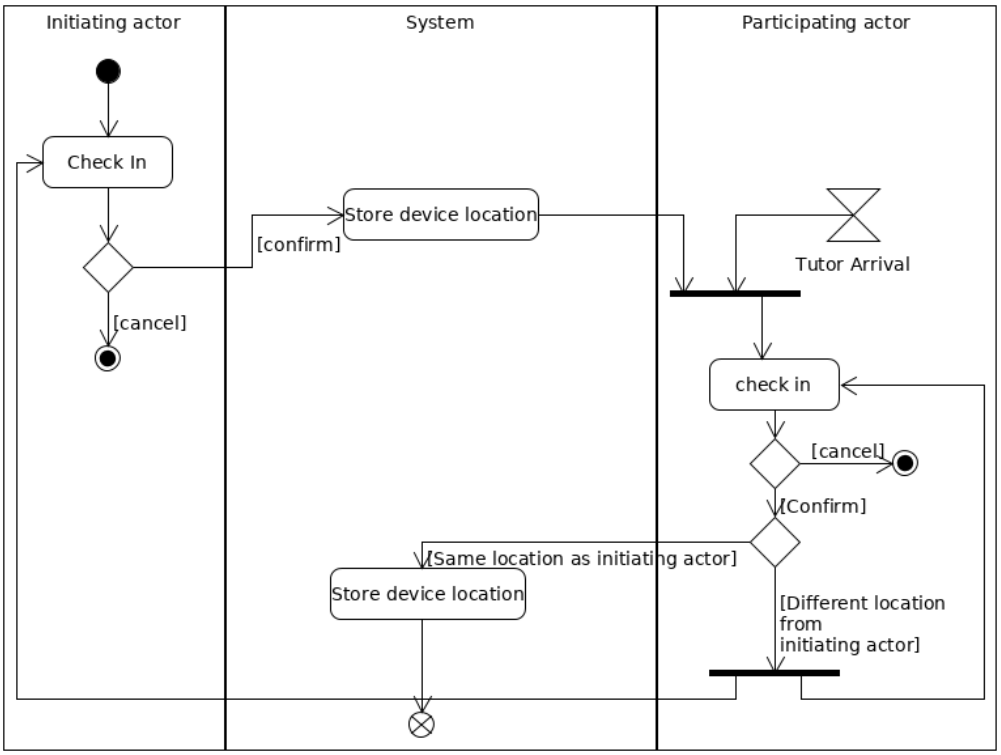
6.3.1 Activity Diagrams

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

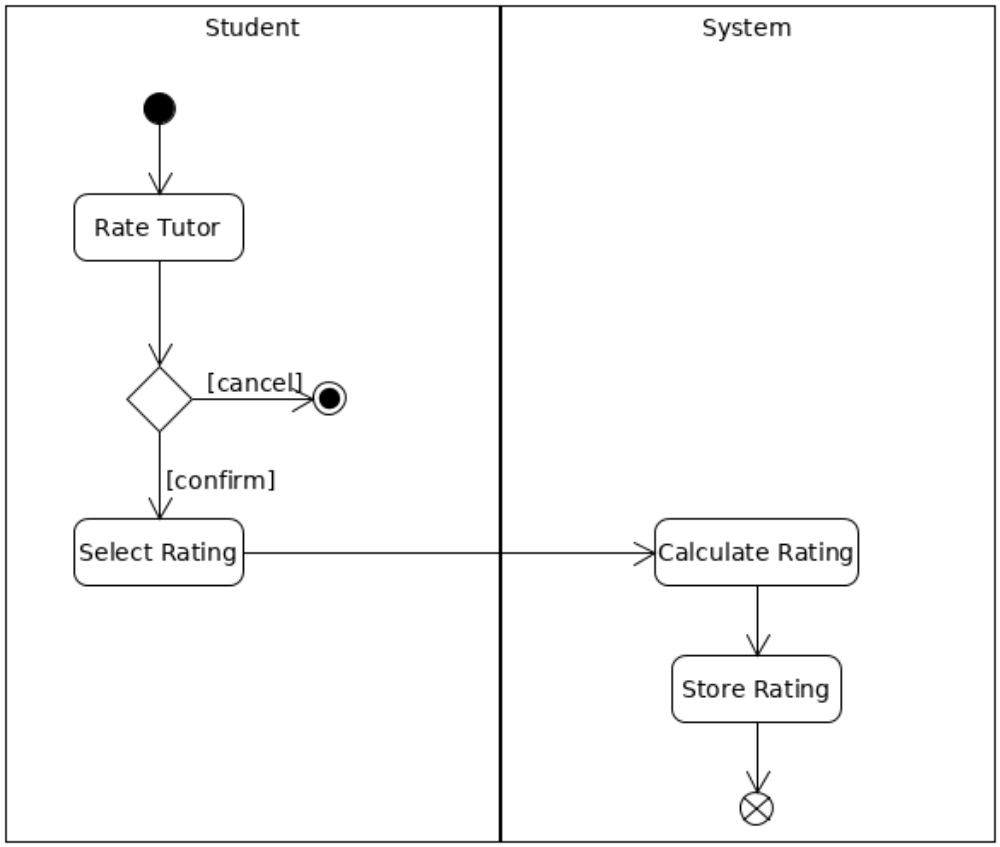
Request Tutor:



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


Initiating actor
System
Participating actor
Check In
[confirm]
Store device location
[cancel]
Tutor Arrival
check in
[cancel]
[Confirm]
[Same location as initiating actor]
Store device location
[Different location from initiating actor]

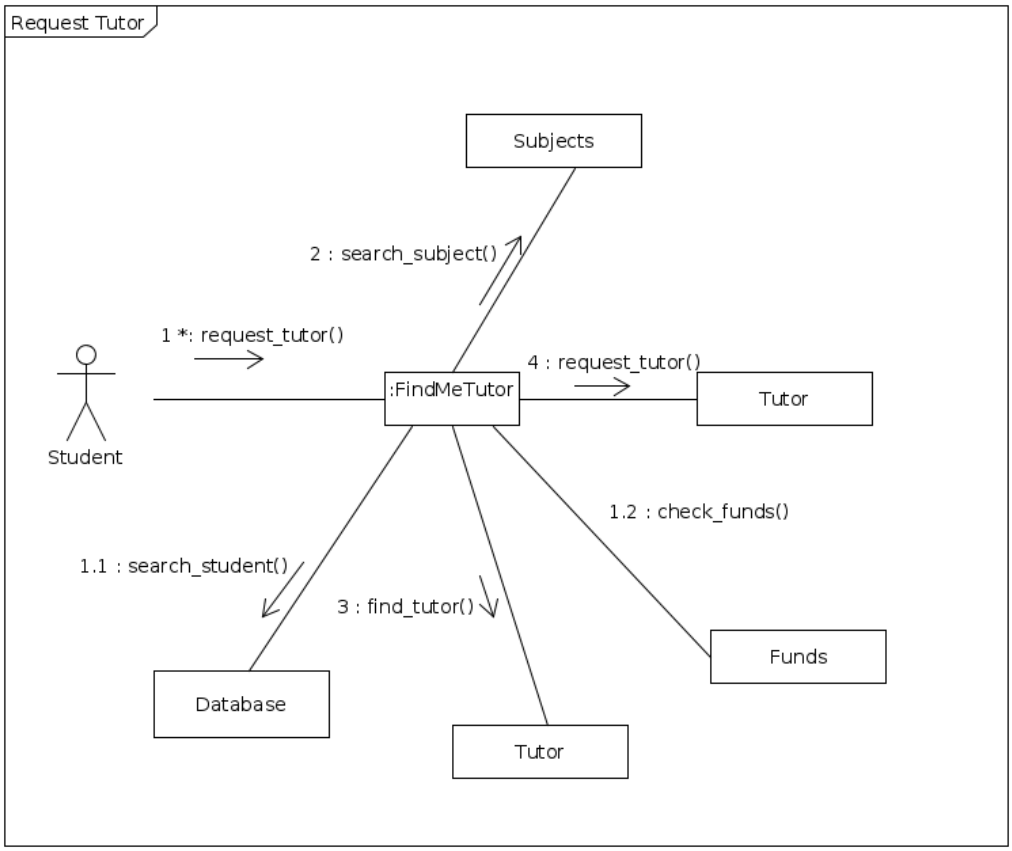
Rate Tutor:


Student
System
Rate Tutor
[cancel]
[confirm]
Select Rating
Calculate Rating
Store Rating

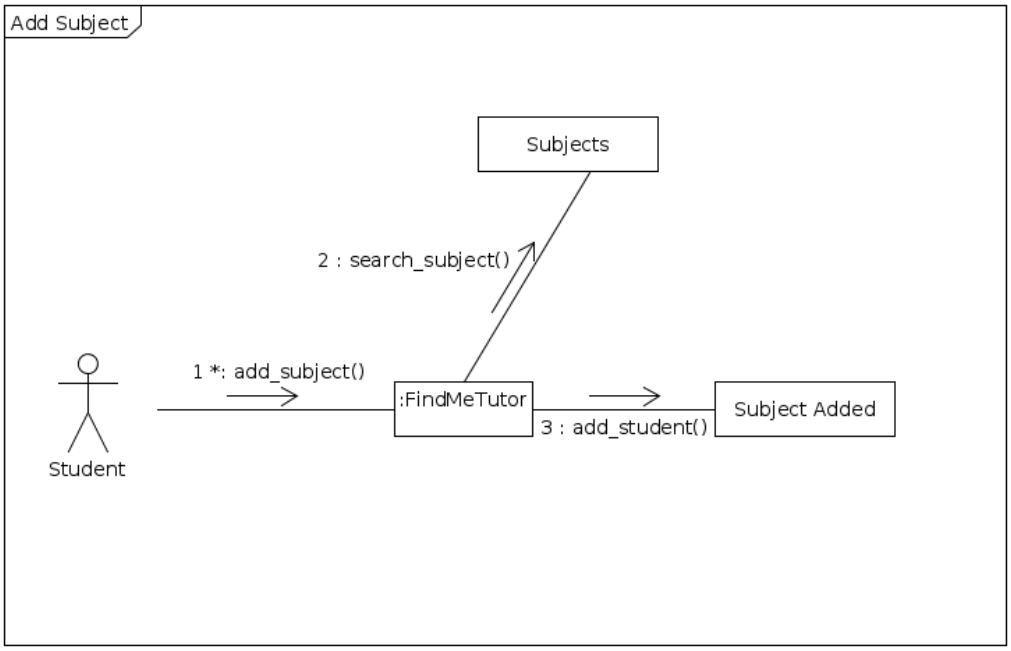
6.3.2 Communication Diagrams

The communication diagrams to Reqeust a Tutor and to Create a Student Account on the system are below:

Request Tutor



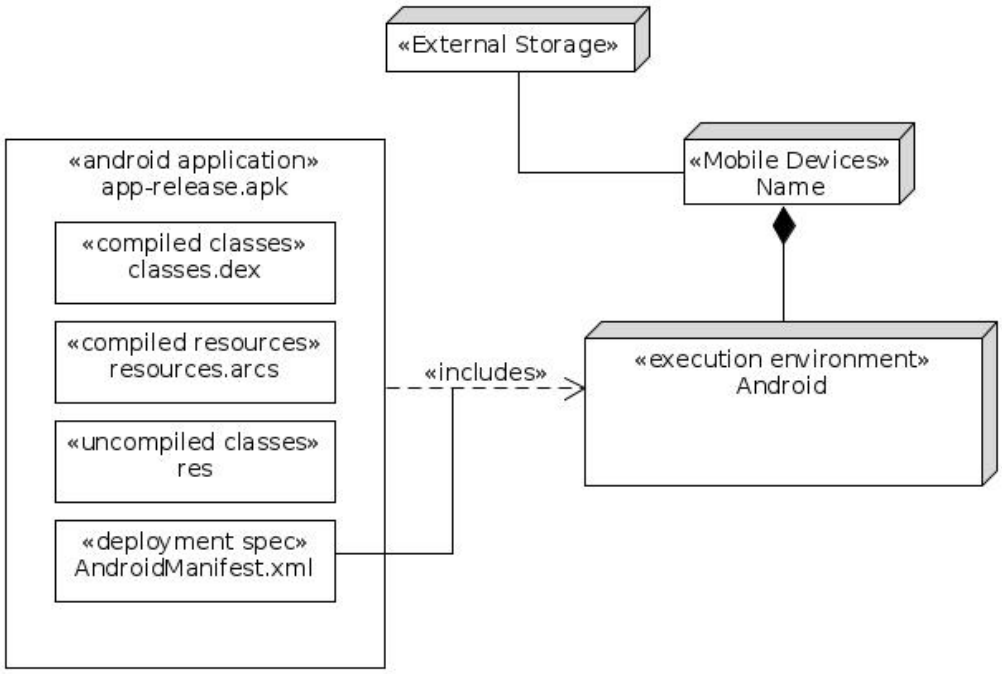
Add Subject


Add Subject
Subjects
2 : search_subject()
1 *: add_subject()
:FindMeTutor
Subject Added
3 : add_student()
Student

6.4 Physical View

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

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.