Requirements Analysis Document
FindMeTutor

August 11, 2016

Contents

1 Introduction

1.1 Purpose of this system

The purpose of this system is to provide a safe environment to link students to tutors in the prospect of tutors providing students with academic help.

2 Overview

2.1 Functional requirements

Requirement	Functional Requirement	Use Case
RQ1.1	The system shall allow a student to register	UC-CS
RQ1.2	The system shall allow a student to update their account	UC-US
RQ1.3	The system shall allow a student to view their account details	UC-VS
RQ1.4	The system shall allow a student to mark their account as deleted	UC-DS
RQ2.1	The system shall allow a tutor to register	UC-CT
RQ2.2	The system shall allow a tutor to update their account	UC-UT
RQ2.3	The system shall allow a tutor to view their account details	UC-VT
RQ2.4	The system shall allow a tutor to mark their account as deleted	UC-DT
RQ3.1	The system shall allow a administrator to update a student account	UC-US
RQ3.2	The system shall allow a administrator to view a student account details	UC-VS
RQ3.3	The system shall allow a administrator to mark a student as deleted	UC-DS
RQ3.4	The system shall allow a administrator to update a tutor account	UC-UT
RQ3.5	The system shall allow a administrator to view a tutor account details	UC-VT
RQ3.6	The system shall allow a administrator to mark a tutor account as deleted	UC-DT

3 System Models

3.1 Use cases

Create Student UC-CS
Update Student UC-US
View Student UC-VS
Archive Student UC-DS

Create Tutor UC-CT

Update Tutor UC-UT

View Tutor UC-VT

Archive Tutor UC-DT

3.2 Use case models

Use Case UC-CS: Create Student	
Related Requirements:	RQ1.1
Initiating actor:	Student
Actor goal:	To register on FindMeTutor
Participating Actors:	External Database System
Preconditions:	N/A
Postconditions:	Student is created
Flow of activities:	
1. Student indicates sign up 2. System displays student sign up form 3. Student enters demographic data, faculty registration details, security answer and password 4. System sends demographic data, faculty registration details, security answer and password to external database system 5. Student is created	

Use Case UC-US: Update Student	
Related Requirements:	RQ1.2, RQ3.1
Initiating actor:	Student or Administrator
Actor goal:	To update student demographic data, faculty registration details, security answer or password
Participating Actors:	External Database System
Preconditions:	Student exists
Postconditions:	Student is updated
Flow of activities:	
1. Student/Administrator requests to update Student 2. System displays form to update Student 3. Student/Administrator enters demographic data, faculty registration details or security answer 4. System sends demographic data, faculty registration details, security answer or password to external database system 5. Student is updated	

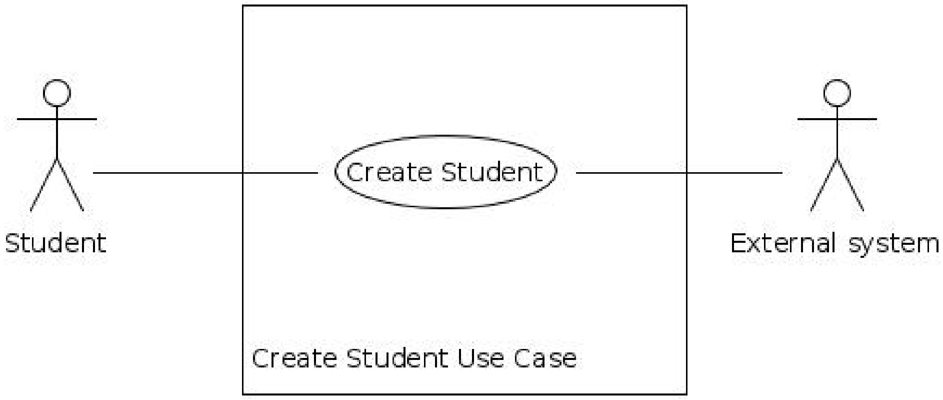
Use Case UC-CT: Create Tutor	
Related Requirements:	RQ2.1
Initiating actor:	Tutor
Actor goal:	To register on FindMeTutor
Participating Actors:	External Database System
Preconditions:	N/A
Postconditions:	Tutor is created
Flow of activities:	
1. Tutor indicates sign up 2. System displays tutor sign up form 3. Tutor enters demographic data, courses tutored, security answer and password 4. System sends demographic data, courses tutored details, security answer and to external database system 5. Tutor is created	

Use Case UC-UT: Update Tutor	
Related Requirements:	RQ2.2, RQ3.4
Initiating actor:	Tutor or Administrator
Actor goal:	To update Tutor demographic data, courses tutored, security answer or password
Participating Actors:	External Database System
Preconditions: Tutor exists	
Postconditions:	Tutor is updated
Flow of activities:	
1. Tutor/Administrator requests to update Tutor 2. System displays form to update Tutor 3. Tutor/Administrator enters demographic data, faculty registration details or security answer 4. System sends demographic data, faculty registration details, security answer or password to external database system 5. Tutor is updated	

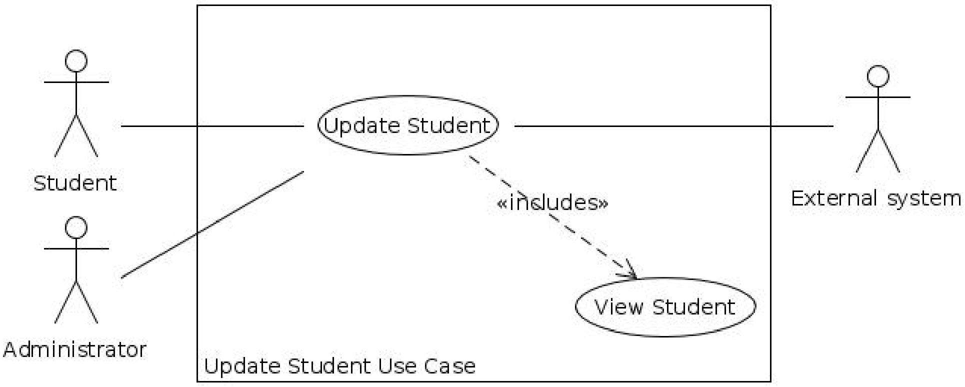
Use Case UC-DS: Archive Student	
Related Requirements:	RQ1.4, RQ3.3
Initiating actor:	Student or Administrator
Actor goal:	To delete Student
Participating Actors:	External Database System
Preconditions: Student exists	
Postconditions:	Student is Archived
Flow of activities:	
1. Student/Administrator requests to delete Student 2. System displays confirmation message 3. Student/Administrator enters confirmation 4. System sends archive Student signal to external database system 5. Student is archived	

Use Case UC-DT: Archive Tutor	
Related Requirements:	RQ2.4, RQ3.6
Initiating actor:	Tutor or Administrator
Actor goal:	To delete Tutor
Participating Actors:	External Database System
Preconditions: Tutor exists	
Postconditions:	Tutor is Archived
Flow of activities:	
1. Tutor/Administrator requests to delete Tutor 2. System displays confirmation message 3. Tutor/Administrator enters confirmation 4. System sends archive Tutor signal to external database system 5. Tutor is archived	

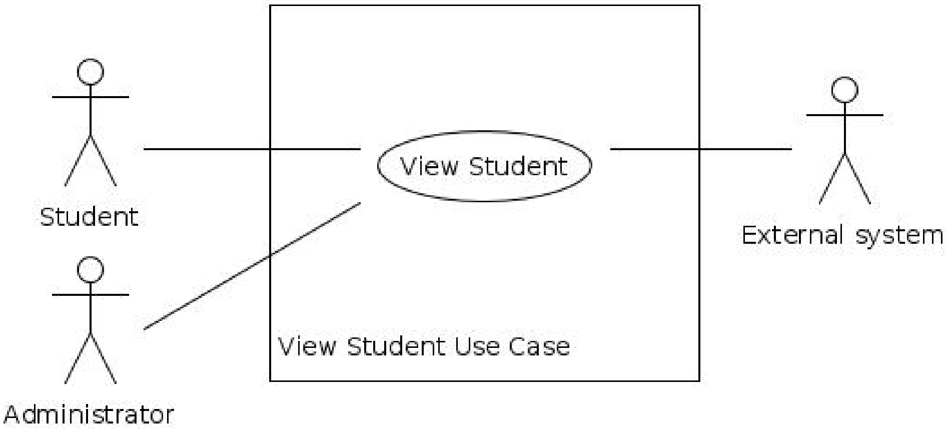
3.3 Use case Diagrams



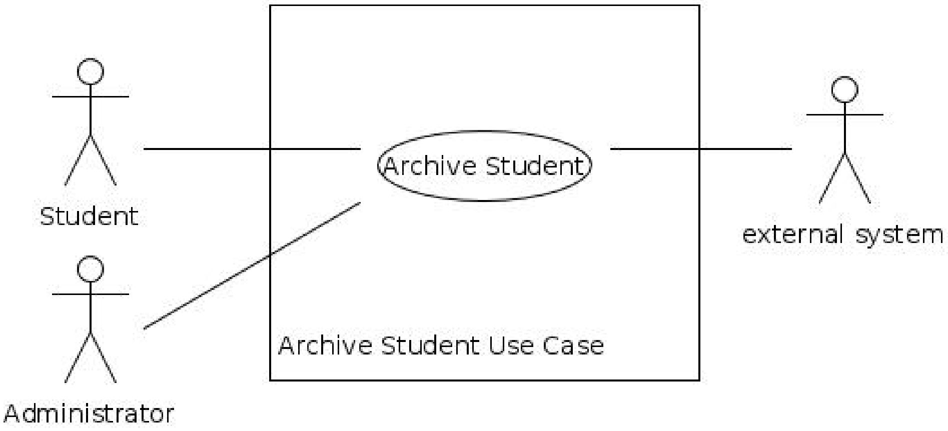
Use case diagram: Update Student



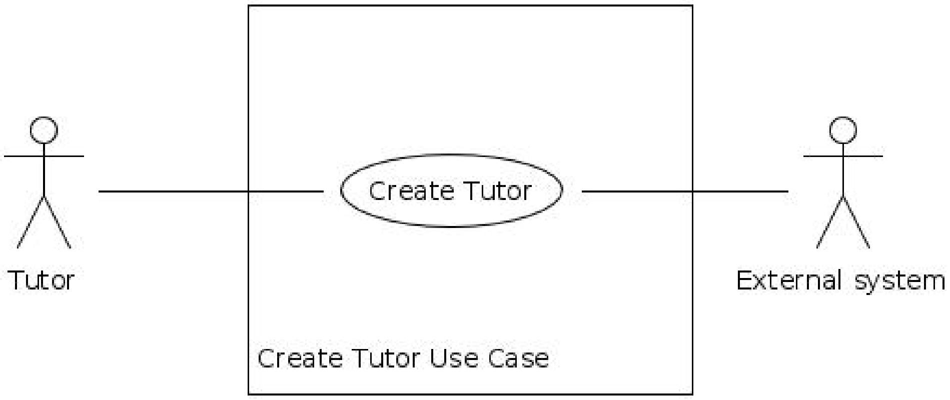
Use case diagram: Update Student



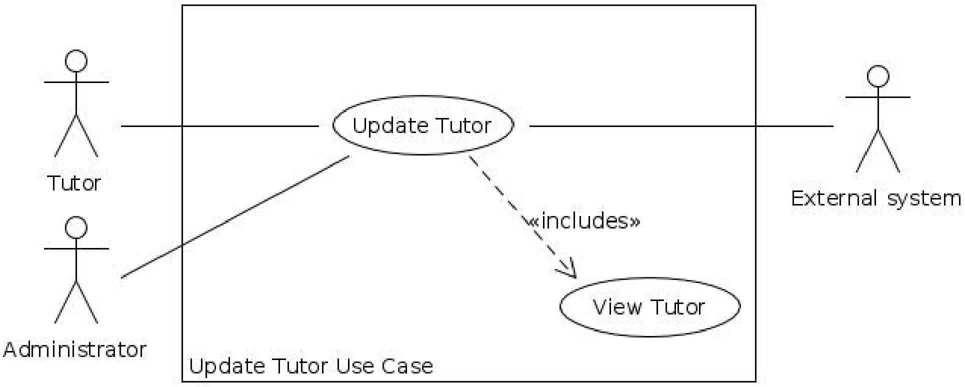
Use case diagram: View Student



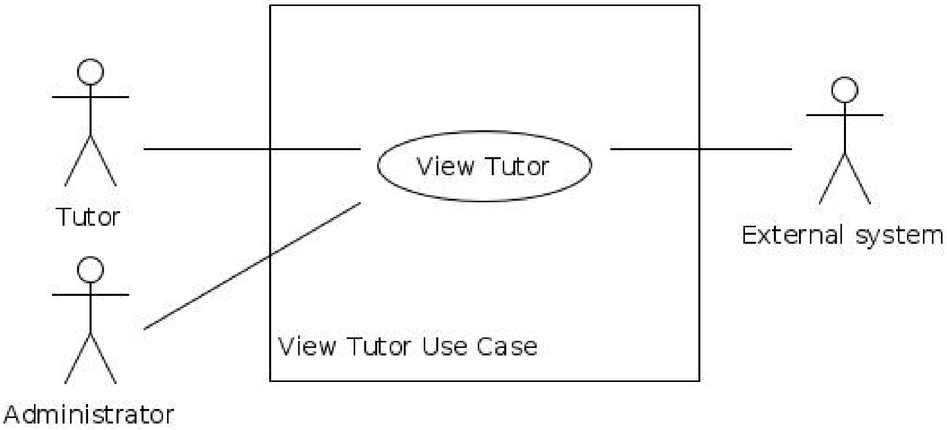
Use case diagram: Archive Student



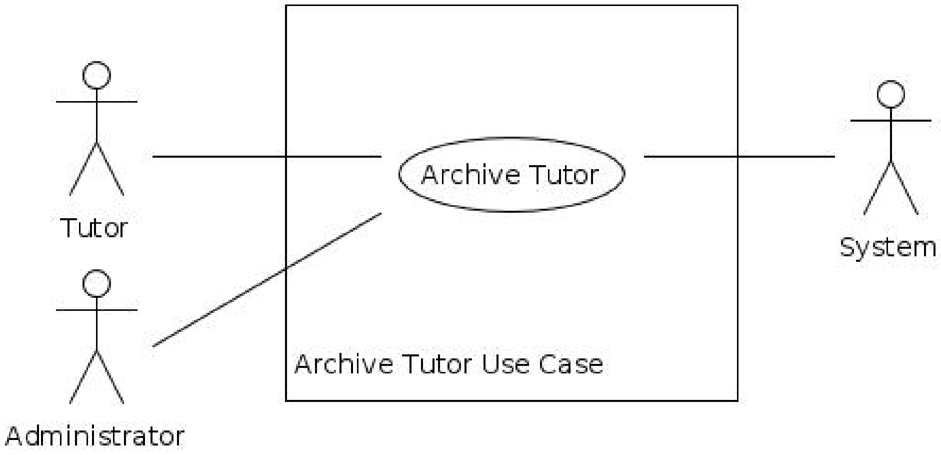
Use case diagram: Create Tutor



Use case diagram: Update Tutor



Use case diagram: View Tutor



Use case diagram: Archive Tutor