

# **REQUIREMENTS ANALYSIS DOCUMENT**

August 9, 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

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
1.0.1	Purpose of the system . . . . .	3
1.0.2	Scope of the system . . . . .	3
1.0.3	Objective and success criteria of the project . . . . .	3
1.0.4	Definitions, acronyms, and abbreviations . . . . .	3
1.0.5	References . . . . .	4
<b>2</b>	<b>CURRENT SYSTEM</b>	<b>4</b>
2.0.1	Overview . . . . .	4
<b>3</b>	<b>PROPOSED SYSTEM</b>	<b>4</b>
3.0.1	Overview . . . . .	4
3.0.2	Functional requirements - "Shall lists" . . . . .	5
3.0.3	Non-functional requirements . . . . .	5
3.0.4	System models . . . . .	6
3.0.5	Operational requirements . . . . .	8

# **1 INTRODUCTION**

## **1.0.1 Purpose of the system**

The purpose of the FindMeTutor application is to provide a convenient means for tutors and students who are looking for tutors to be able to connect within a particular tertiary institute.

## **1.0.2 Scope of the system**

Our team, working on the FindMeTutor application, envisions a successful product to be an Android Application which will be at students's disposal in order to improve their grades and achieve their academic dreams. With limited resources a stringent budget and capped time, we aim to execute this task in an economical fashion.

This goal will be achievable by implementing an agile methodology strategy within our team. We will be able to set short term targets to show case our deliverables at particular sprints, with a long term goal being to present the FindMeTutor Android Application.

## **1.0.3 Objective and success criteria of the project**

The FindMeTutor Android application will be seen as successful if it facilitates a platform for tutors and students can meet. We have great hope that the result of this would mean better results obtained by the students, and a manner in which tutors can generate some income and gain some job experience.

## **1.0.4 Definitions, acronyms, and abbreviations**

1. App - abbreviation for application.
2. Application - is a software program that runs on your computer.
3. Android - is a mobile operating sysytem developed by Google.
4. OS - abbreviation for operating system.
5. Operating system - is a software that communicates with the hardware and allows other programs to run. It compromises of system software, or the fundemental files your computer needs to boot up and function.
6. Boot up - turning on the computer.
7. Java - is a high-level programming language

8. UI - abbreviation for User interface
9. User interface - is the means in which a person controls a software application or hardware device.
10. ID - abbreviation for identity
11. User ID - the identity that uniquely identifies someone on a computer system.
12. Sign in - when asked to enter username and password information. A sign in/ login is a combination of information that authenticates a user's identity.

### **1.0.5 References**

1. <http://techterms.com/definition> (2016-08-08)

## **2 CURRENT SYSTEM**

### **2.0.1 Overview**

Currently, there are many students in search of tutors to help them with particular courses with which they require some support, as well as fellow students or tutors who are available to tutor particular courses of study. However, the problem that is faced on hand is that either pool (students and tutors) are struggling to find each other.

## **3 PROPOSED SYSTEM**

### **3.0.1 Overview**

FindMeTutor app will be a platform through which students and tutors can meet in order to resolve the current situation.

FindMeTutor app will facilitate the following two registration categories:

1. Student looking for tutors - they are able to register to the app with merely some personal details (Name, email address, Skype name (if applicable) and courses enrolled in).
2. Tutor - those who would like to tutor can register on the app by simply filling in some details with respect to the fields of study they are particularly comfortable to tutor.

### **3.0.2 Functional requirements - "Shall lists"**

Describes the high-level functionality of the system

3.2.1 The database has the ability to store data captured of the users of the application (student and tutors data). More specifically, the database will be able to accept numeric data entry and will store the courses under the University of Wthe Witwatersrand's School of Compute science and Applied mathematics undergraduate topics which will be displayed in the drop-down menu when students and tutors will be required to select the courses respectively to which they are enrolled in or able to tutor.

3.2.2 The app will provide an effective means through which tutors and students can communicate. The messenger facility will facilitate communication between students and tutors - when students request a tutor, the tutor and student will be able to use this facility to arrange a convenient time, place and tutition fee for the tutor service.

3.2.3 Request appropriate tutors for students who send a request on the app for a tutor eligible to tutuor a specific course (aid to pair up appropriate tutors and students).

3.2.4 In the light of security, the database shall be able store a corresponding password for each user of the application, which will allow for each users account to be protected, as each user's user ID and entered password will be compared to that of which was entered into the database at sign-up, hence the system resticting access to authorized users.

3.2.5 FindMeTutor Android application will be able to be utilised on any Android OS of (some version), which is able easily accessable,as many students can download this application for use on their smartphones and/or tablets which shall makes the application easily accessible.

### **3.0.3 Non-functional requirements**

Describes the user-level requirements that are not directly realted to the functionality.

#### **3.3.1 Usability**

The application will be user friendly as will be pitched as an Android application which is suportred by multiple devices (smartphones and tablets) with an Android OS. This will allow for the application to be easily accessible to students and tutors as majority of the students have access to these devices.

#### **3.3.2 Reliability**

The probability that the system will be able to process work correctly and completely without being aborted.

In the case of system failure, the damage that could be caused could be such where the personalised accounts of the application user will lose their preference choices of subjects selected on sign-up. 3.3.3 Performance

The response time between the UI and the server will be(?). The expected volume of data (?). The expected volume of user activity will peak at the end of each academic term within the tertiary institute when examinations / tests will be approaching, while on a regular basis the application will be utilised when students who feel the need to get assistance immediately when they encounter a topic they require assistance. 3.3.4 Supportability

#### 3.3.5 Implementation

Our team has implemented the agile methodology in order to obtain our final goal of building a find me tutor application. For each sprint we will set targets of what we would like to achieve, with the objective of using these milestones to be building blocks towards our final goal. 3.3.6 Interface

The UI will be made in Android studio. The set up will be simple and neat. The app will be used by students who will be using the app in order to search for a tutor which is suitable to tutor, hence, with this intention, to prevent furthering the overwhelmed feeling, the app will not be cluttered and simple to use. The 'user-friendly' experience provided by the UI, will allow the user to interact with the app in a natural and intuitive way.

Each user's home page will be customized to the particular topics in which they are enrolled in or signed up to tutor, with respect to whether they are students or tutors.

#### 3.3.7 Packaging

#### 3.3.8 Legal

### 3.0.4 System models

#### 3.4.1 Scenario

For instance, there is a student - Joe Soap - who is currently doing his 3rd year of study in computer science. Joe would like to generate some income from tutoring first and second year mathematics modules. However, Joe is also looking for a 3rd year Mathematics tutor to help him with some of his mathematics courses. We also know that the student, Mary Smith, is a first

year astronomy student who is looking for a mathematics tutor. The FindMeTutor app will be ideal to resolve the problems faced in this particular scenario. Joe will register on the application as a student and as a tutor, on registering, he will select what he is capable and willing to tutor - first and second year mathematics. On the other hand, we will have Mary register as a student. Mary will then be able to search for the course she needs assistance in, for example Calculus I; any tutors who have uploaded material on this particular topic (Calculus I) will come up. Mary may find these resources useful, but if Mary does not get the assistance she requires from the material available, and would like to seek a personal tutor, Mary will click the 'Request tutor' button and this will send a request to all those who have registered to tutor Calculus I. Joe Soap will be part of the list of tutors approached. Joe accepts the request. Mary is notified of this and is now able to communicate with Joe to make further arrangements of time of meeting, venue and also to flag him of what sections she needs assistance in. On the other hand, Joe requires an 'Emergency lesson' this is a last minute lesson before his test which is scheduled to take place in a day's time. Tutors eligible to tutor Joe's course will be sent an 'Urgent request'. A tutor that accepts this request will communicate with Joe through the messaging facility which is made available. If the only possible means of having a lesson before the test is to Skype each other, then the tutor and Joe will be redirected to a skype page which will facilitate this conference.

#### 3.4.2 Use case model

#### 3.4.3 Analysis object model

#### 3.4.4 Dynamic model

A dynamic model represents the behaviour of an object over time. It is used where the object's behaviour is best described as a set of states that occur in a defined sequence. The components of the dynamic model are:

States. The purchase order (PO) is modeled as passing through a set of states. The operations that can be performed on the PO are dependent on the state it is in. For example, information cannot be entered after it has reached the approved state.

State transitions. When purchase order data entry has been completed it transitions from the entering PO state to the pending approval state. State transitions are modeled as being instantaneous.

Events. Events trigger state transitions. For example, the order mailed event triggers a state transition from the approved to the placed state.

Actions. Actions occur on state transitions. For example, on a goods received event the action of notify purchaser is performed. Actions are modeled as instantaneous occurrences (contrast with activities).

Activities. An activity is performed while an object is in a specific state. For example, while in the entering PO state data is entered into the PO. Activities are modeled as occurring over a period of time (contrast with actions).

(Will add diagram) 3.4.5 User interface navigational paths and screen mock-ups

### **3.0.5 Operational requirements**

Operational requirements describe the non-business characteristics of an application.

3.5.1 Azure.com - Web server to host the database

3.5.2 Android studio to design UI

3.5.3 GitHub to facilitate the build of the project among team members

3.5.4