MDroid

Android App for Moodle
Software Requirements Specification

Version 1.0

Revision History

Date	Version	Description	Author
07/07/2014	1.0	Software requirements specification	G.G.S.M Nandathilaka

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the "Android application for Moodle". It will describe the purpose and features of the system, the user interfaces, dependencies and constraints under which it must operate. This document is intended for both the developers and stakeholders of the system.

1.2 Scope

As students we all access to Moodle every day. It is essential for us to deal with the Moodle to view the content of courses which we have registered. As the Moodle is viewed frequently by the students, it would be beneficial if we can access this stuff from a single mobile application. I have planned to do this project because of that. I think that, it is very important to develop an android application for the Moodle system.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
USER	Student who interact with the mobile application
STAKEHOLDERS	Any person who has interactions with the system who is not a developer
DEFINED	The official definition of a term contained in a Programming Language statement
MUST	The minimum level required to avoid failure contained in a Programming Language statement

1.4 References

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[2] Davis M A, "Just Enough Requirements Management: Where Software Development Meets Marketing", New York, Dorset House Publishing, 2005.

1.5 Overview

The remainder of this document includes two chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the table of contents, index and appendixes. Appendixes in the end of the document include all the results of the requirement prioritization and a Release plan based on them.

2. Overall Description

This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand.

2.1 Product perspective

This system will consist of a mobile application. The mobile application will be used to find courses for which users have registered themselves and view information about them. Also users can search about their current grades of continuous assessments of the course modules. Mainly this application is developed to get notifications from the Moodle. Most users have to face many issues when they submit assignments. Most of them forget their deadlines of relevant course modules. So this is a very important android application for users

2.2 Product functions

- Users should be able to check their courses which they have registered through this application.
- Users should be able to check forums.
- Users should be able to check grades of courses.
- Users should be able to download course notes and get notifications for assignment submissions.

2.3 User characteristics

There are two types of users that interact with the system: users of the mobile application and students. Each of these two types of users has different use of the system so each of them has their own requirements.

The mobile application users can search details about the course modules which they have registered and get notifications of those modules.

2.4 Constraints

The mobile application is constrained by the Moodle web site. Therefore the developer has to comply with the architecture of the Moodle system and can't violate it. For example, data retrieval and storing should be done using the existing classes built into Moodle system.

The Internet connection is also a constraint for the application. Since the application fetches data from the database over the internet, it is crucial that there is an internet connection for the application to function.

2.5 Assumptions and dependencies

One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or even at all.

3. Specific Requirements

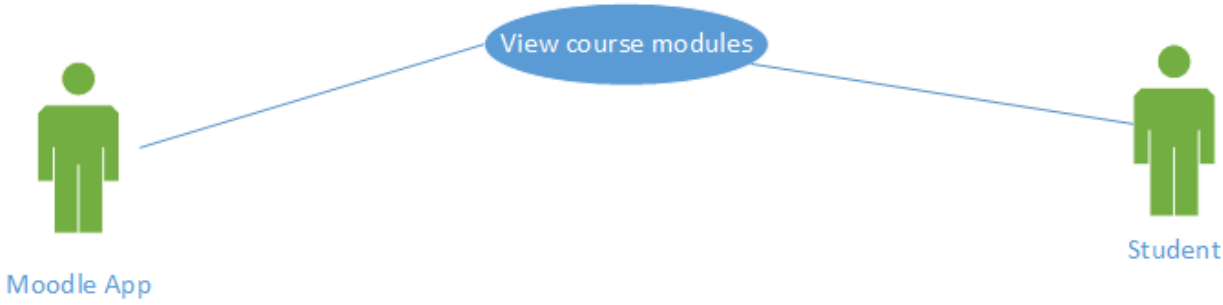
This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

3.1 Functionality

This section defines all the functional requirements in detail.

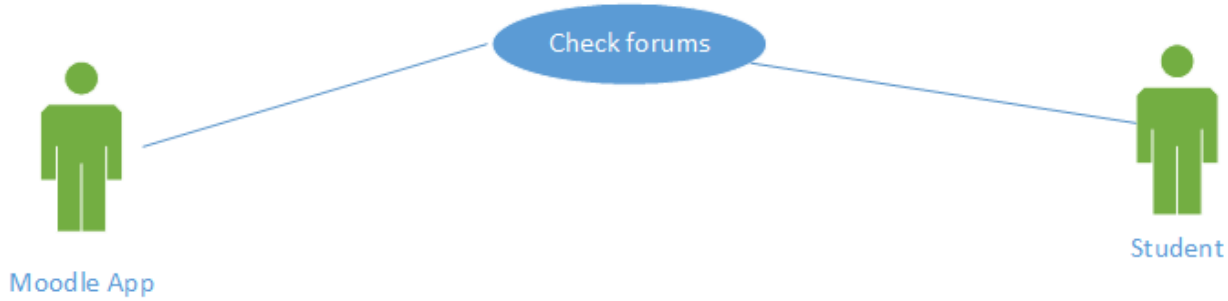
3.1.1 Check the courses which students have registered.

This mobile application should be able to view the details of the registered courses. This view will be a summarized one.



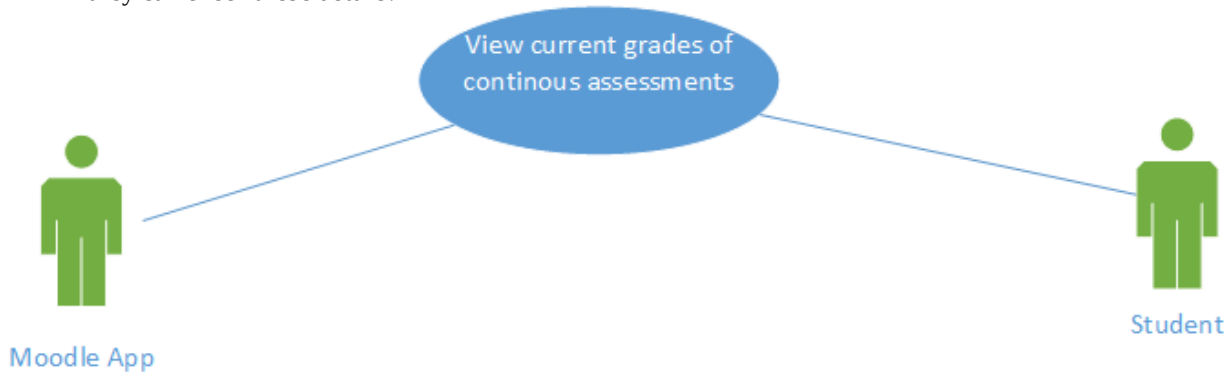
3.1.2 Check forums

Students can check forums through this application. Also they can post to forums through this application same as the Moodle website.



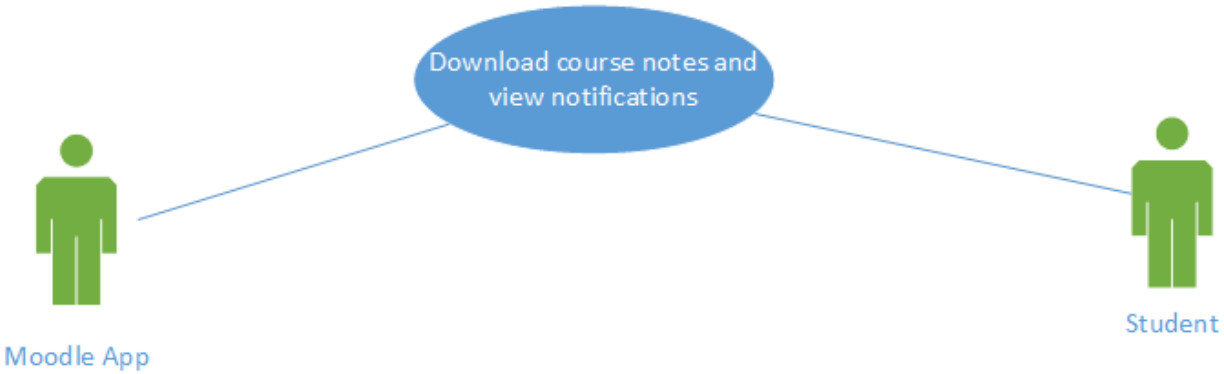
3.1.3 Check grades of the courses

Students can check their current grades for the course modules through this application. Straightforwardly they can check these details.



3.1.4 Download course notes and get notifications

Students can download the course notes related to the course modules which they have registered. And also they can refer to those course notes through their mobile phones.



3.2 Usability

In most cases, this systems will be used by users already familiar with the Moodle. So it would take no time to learn new features added by this system. Even for a completely new user it wouldn't take a considerable extra time to learn the features added by this application in addition to the time to learn default features in Moodle.

3.2.1 Intuitive user interfaces

User could be able to adapt to this Moodle application easily. And also users are attracted with the user interfaces and they can easily handle the application. Actually these user interfaces have to be user friendly.

3.2.2 Time to learn should be less amount

User should understand what the new feature will do in no time. This could be enabled by viewing place holders when user touch the screen of mobile phone.

3.2.3 User should clearly understand what new feature does

User should clearly know what the new feature does. We can make sure this by showing a little description along with these new features.

3.3 Reliability

3.3.1 The system should not contain any bugs

Bugs or defect rate should be less. The application will not be containing any "critical" bugs. For example, complete loss of data or a complete inability to use certain parts of the system's functionality.

3.3.2 This mobile application should be available 100%

This application will be able to download from the Google play anytime. And also after installed to the mobile platform it must work correctly.

3.3.3 Mean time to repair should be less

Mean time to repair means how long is the system allowed to be out of operation after it has failed. This time will be less.

3.3.4 Accuracy of the system should be high

System precision (resolution) and accuracy (by some known standard) should be high. Output of the system will directly depend on these features.

3.4 Performance

3.4.1 Response time for a transaction

This Android application enabled system should not increase the time taken to load a course more than 20ms compared to default Moodle system.

3.4.2 Throughput of the system

Throughput of the system means transactions per second. This app will be able to give outputs in a less amount of time.

3.4.3 Resource utilization

This system utilize the memory and disk spaces. From that it can provide a better throughput.

3.5 Supportability

This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.

A complete online documentation for the system should make publically available so that anyone interested in the project can contribute in evolving and improving the product.

3.6 Design Constraints

There is no API for Moodle of the University of Moratuwa. So I have to use web scraping technology to do this project. I am developing this application using android. Therefore I have to use JSoup jar library to implement the web scraping. Here I will get details from the Moodle web site and display those details through an android application. Here I have to use web view method to display the content of the web site. First I have to create a login window. For that we have to connect to the internet using our android application. After getting connected to the internet we can load contents of the Moodle web site and display that information using the android application. Also I have a plan to enable viewing the site while offline. Then students are able to view their courses which they have registered even while they are offline.

3.7 Purchased Components

The entire Android app for Moodle project is open source, and hence no component of it will be purchased.

3.8 Interfaces

3.8.1 User Interfaces

There will be only one user type. Therefore there will be only one kind of user interfaces.

3.8.2 Hardware Interfaces

There is no hardware interfaces for the system.

3.8.3 Software Interfaces

This app will be directly associated with the users and it scrapes details from the Moodle web site.

3.9 Licensing Requirements

The system will be licensed under GPL so that other developers will be able to contribute to the system and it will be free