Acceptance Test Document

CITS3200 Group R

Objectives

This document is designed to describe the conditions under which we can consider the project completed. It cannot be fully comprehensive because we anticipate the final product will contain features beyond those described here, but it describes the essential features for the sleep diary to be considered complete. Due to this the document describes the minimum final product to be delivered, with the expectation that future documentation will supersede it.

Test Summary

The system is a sleep diary app for iOS and Android devices. The app takes user input about their sleep patterns and uploads it to a SQL database where it is converted to an excel spreadsheet and sent to their clinician upon request. Further details are available in the General Scope of Work document.

Testing Strategy

For the system to be considered functional there are several subsystems that must be functional and correctly linked:

- iOS app
- Android App
- Server database
- Webservice

The individual subsystems will be tested based on function, performance and user interactability. Function can be tested through interaction with the subsystems and attempting many types of inputs. Performance will be tested by calculating how quickly the subsystem can perform tasks and how many users it can handle. User interactibility will need to be tested by external people for ease of use and verified by the client.

Subsystems to be tested

iOS App

The iOS app must be able to:

- Recieve user input correctly
  - Ensure input is valid
  - Ensure all info is filled in
- Differentiate users
- Send user data to the server
- Save user data locally
- Visually display user data

Android App

The Android app must be able to:

- Recieve user input correctly
  - Ensure input is valid
  - Ensure all info is filled in
- Differentiate users
- Send user data to the server
- Save user data locally
- Visually display user data

Server database

The server database must be able to:

- Recieve user data
  - Must be resistent to SQL injection attacks
- Store user data in a SQL table
  - Must differentiate between users
  - Must only update once per day
- Convert SQL data into excel spreadsheet
- Verify and send authorised data to clinicians/researchers

Webservice [I really don't know what this does]

The webservice links the apps to the server database. To do this it must:

- Form a connection between mobile device and server