**UC Schedule**

**Requirements Document**

**Ver. 1.3**

**Table of Contents**

1. Application Overview
2. [Objective](#Objective)
3. [Business Process](#BusinessProcess)
4. [User Roles and Responsibilities](#UserResponsibilities)
5. [Interaction with other systems](#Interaction)
6. [Replacement of Legacy Systems](#Replacement)
7. [Terminology](#Terminology)
8. Functional Requirements
9. [Statement of Functionality](#Functionality)
   1. [Security](#Security)
   2. [Auditing](#Auditing)
   3. [Administration](#Admin)
10. [Scope](#Scope)
11. [Performance](#Performance)
12. [Usability](#Usability)
13. [Concurrency](#Concurrency)
14. Appendices
15. [Author(s) background and expertise](#AuthorBackground)
16. [Resources](#Resources)
17. [Contacts](#Contacts)

**Application Overview**

Objective:

The purpose of this application is to make a portable UC schedule including the course name, course time, course duration, building location, room number, call number, professor name and room for notes like homework dates. The current calendar does not have the options for notes or anything of the sort so this would be quite convenient for both students and staff.

Business Process:

With the help of UCIT, we will be able to access an API to interface with the UC schedule.

User Roles and responsibilities:

* UC students and faculty – These users will use the application to import and locally edit their UC class schedule. The system will be responsible for properly importing the user’s Class schedule. Once the schedule has been imported, the user has the responsibility to edit their local schedule to suit their needs.

Interaction with other systems:

This application will have to interact with two other larger systems. One system is the default Android calendar and the other is the UC database, specifically the UC scheduling database.

Replacement for legacy systems:

This app will possibly be integrated into the UC Mobile app. The UC Mobile app is a mobile version of the University of Cincinnati’s website. With application you can complete popular tasks such as:

* Use blackboard (a site used by university professors to upload hw, post notes, post exam dates, etc.
* Check the student/faculty directory (get email info,etc.) by searching the person’s full name.
* Check UC related news
* Check UC calendar events(non-individual)
* Check all Campus maps
* Get UC Emergency Contact info
* Check the entire database of courses
* Check OneStop information (view schedule)
* Check UC athletic information (like news, scores and schedule info)
* Check social media related to UC
* Watch videos about UC
* Look at UC photos
* Lookup UC library information
* Apply to UC
* Lookup academic programs
* Lookup Scholarship & Aid information

Specifically we would to replace the current OneStop schedule viewing. The current method took some digging to find and you can only view your information one at a time and you can’t edit anything. The system we plan to use will not only be a local backup of a student’s/faculty member’s schedule but the user will be able to edit it.

Terminology:

* UC – University of Cincinnati
* UCIT – University of Cincinnati Information Technologies department
* Database – An organized collection of data.
* Local – Pertaining to the mobile device itself without any outside interactions.
* Concurrency – Simultaneously executing multiple processes.**Functional Requirements**

State of Functionality:

* Import your UC class schedule into your calendar. Imported fields will include the following:
  + Time of day of the class.
  + Days the class is scheduled for.
  + Building the class is located in.
  + Room number of the class.
  + Call number of the class.
  + Professor’s name.
* The import of schedule data will be done through the app and will populate the standard Android calendar.
* In each event, the user can add notes (hw/quiz dates, faculty stuff, etc.)

**Security**

Only UC students/faculty with valid credentials will be able to import their schedule.

**Auditing**

A release notes document will be released along with each version of our application. The release notes will indicate what has changed since the previous version of the application. This includes things like new features or bug fixes.

**Administration/Customization of the Application**

The user’s imported schedule will act as their default schedule and they will be able to edit the schedule locally however they see fit.

Scope:

The development cycle will consist of multiple phases:

Phases already complete:

Pre-Alpha Release 0.0:

* Created Software Requirements document

Pre-Alpha Release 0.1:

* Added login page(MainActivity)
* Added specification document
* Updated Software Requirements document

Pre-Alpha Release 0.2:

* Added calendar intent
* Added changed MainActivity into navigation page
* Added LoginActivity(old MainActivity)
* Added “add event”(for test purposes) functionality
* Updated specification document
* Updated Software Requirements document

Alpha Release 0.1:

* Fixed MainActivity layout problems
* Added Test Cases
* Replace Add Event intent with manual add event functionality
* Updated Specifications document
* Updated Software Requirements document

Alpha Release 0.2

* Added JSON file usability
* Added Test Cases
* Updated Software Requirements document

Beta Release 0.1

* Added “Download Schedule” Functionality
* Added Test Cases
* Updated Software Requirements document

Projected Phases:

Beta Release 0.2

* Clean up code (particularly Json parsing)
* Completely implement “Download Schedule” (include multiple days of week, complete semester, etc)
* Add more comments
* Organize code by breaking up into classes
* Added Test Cases
* Updated Software Requirements document

Performance:

There is no performance requirement in the design of this application at this time.

Usability:

The minimum Android SDK is 4.0 (Ice Cream Sandwich).

The targeted Android SDK is 4.2 (Jelly Bean).

Concurrency:

There is no concurrency in the design of this application at this time.

**Appendices**

Author’s background and expertise:

Matthew Scurry:

1. Currently a pre-Junior at the University of Cincinnati, majoring in computer engineering.
2. Coding expertise includes: Visual Basic, HTML, C++, Objective C, Python, PySide and Visual Basic Application.

Arthur Johnson:

1. Currently a pre-junior in Computer Engineering at the university of Cincinnati.
2. C#, C++, Visual Basic, Perl, Visual Studio Application, Silverlight Application, XML, Microsoft Office, IBM Rational Tool’s.

Resources

* <UC schedule API>
* <Calendar API>

Contacts

Greg Benjamin Phone: (513) – 558 – 8570

Email: Benjamgs@uc.edu