# Courseworks App Development Documentation
# Pre-Alpha Build

Alexander Roth
UNI: air2112

July 7th, 2014

## 1 Overview

The goal of mobile application is to compliment the features of Columbia's Courseworks web site and increase accessibility for students.

As of July 7, 2014, this document outlines the development process and future iterations of the Columbia Courseworks app. **Note:** This application is still in development in a *pre-alpha build* and is currently being built for Android only. Currently looking into development for iOS devices.

## 2 Features

Again, this application is in the pre-alpha development cycle. Thus, this section is subject to heavy change depending on justification in further design and development cycles.

### 2.1 Current Features

Features currently implemented within the application include:

1. "Remember Me!" functionality
2. REST API calls to CAS server for Log In procedures.
3. Dynamic Notification center.

### 2.2 Future & Proposed Features

1. Current Semester Homepage
2. Course Names instead of Course numbers
3. Seamless calendar integration with native mobile calendar applications.

4. A viewable roster
5. Easy downloads under "Files & Resources"

# 3 Implementation & Efficiency

## 3.1 Workflow

The current design of the app works in a three-step process: the preamble, the authenticat ion, and the main process.

1. **Preamble Process**
   The preamble process involves the use of the Splash screen. To the user, a splash screen with Columbia's logo will be displayed. On the back-end, the application will be checking if the "auth.xml" exists. If this file is present, then a login attempt will begin. If this file is not present, then the login activity will be invoked.

2. **Authentication Process**
   After the preamble process, the application moves into authentication. If the auth file exists, the system goes to auto-login using the credentials of the last user. If this is not the case, the user will be redirected to the login screen, where he or she will provide credentials for the system to use.
   Once credentials are provided, the application launches a series of RESTful calls to CAS servers, which then authenticates the user. Once the user has been authenticated, the application moves on towards the information gathering process.

3. **Information Process**
   The main process of the application. Once the user is authenticated, we implement some method that will recreate the user's identity. After the reconstruction of the user's identity, we send out a number of GET requests to the Courseworks API, which will allow the application to parse the information in as JSON or XML.

## 3.2 Resource Allocation

Since this application is native to the mobile device, there will be very little overhead, except for the HTTP requests and responses when communicating with the servers. As of July 7, 2014, there are only two requests being sent to the Courseworks servers: The POST and GET requests from the Authentication process.

# 4 Class Design

## 4.1 AnnnouncementView

A fragment activity that acts within the Main activity. It is the first activity that is launched after authentication. It will take the parsed JSON and display it dynamically to the user in a list format.

## 4.2 AuthPreferences

A helper class for the Login activity. The class contains a number of methods that either write to or read from the SharedPreference file "auth.xml".

## 4.3 CalendarView

Currently a work in progress: the CalendarView will act as a Calendar showing upcoming events for a user within a given time frame. Possibly considering updating that for just one week display. Also, implementing a date picker feature to add events.

## 4.4 CourseView

Currently a work in progress: the CourseView will show all the current classes in a user's current semester. Need to implement a method that launches an activity that will display information relating to that class from that view.

## 4.5 InformationRequester

The backend of the application. Handles all the GET and POST calls to the Courseworks API.

## 4.6 Main

The main hub of the application. It holds the fragments that act as different acts for each section.

## 4.7 Login

This activity controls the Login screen. View the source and javadocs for specific functionality for each method.

## 4.8 RestGrant

An alternative to both OAuth implementations mentioned beforehand. Implements CAS RESTful API to make a series of calls to the CAS Authentication servers.

### 4.9 Splash

The Splash activity acts as a buffer between the Login activity (on first access of the application) or the Main activity on the application.

## 5 Risks

### 5.1 Security

Since the application is native, it is more conducive to store the user's credentials on the mobile device; thus allowing the application to hold the user's credentials in a file system. However, this ease opens up a large security risk. The file holding the user's sensitive information (e.g. uni and password) is not secure at the present moment. There are three workarounds to this issue:

1. We encrypt the SharedPreference file holding the user's information.
2. We remove the "Remember Me!" feature.
3. We move to web authentication through OAuth's *Authorization Code* protocol.
4. We move to web authentication through OAuth's *Implicit Grant* protocol.

As of July 7, 2014, I have implemented an encryption method using Advanced Encryption Standard (AES). However, the SharedPreference file is not guaranteed completely secure; I just increased the difficulty for someone to access the information.

## 6 ToDo

1. Test Authentication Process.
2. ~~Secure private information of user.~~
3. Update feature list based on student response.
4. Move to alpha build once the Courseworks servers are prepared.
5. Implement methods to the Courseworks API.