

# UniPubSub

## Mobile Computing: App Development Project

### Milestone 5 User Documentation

By

Sandesh Pardeshi [ssp9237@rit.edu](mailto:ssp9237@rit.edu), Koustubh Kulkarni [krk9372@rit.edu](mailto:krk9372@rit.edu)

Date: November 13<sup>th</sup>, 2012

## Introduction

'UniPubSub' is an Android-based mobile app which helps students of various universities to share information with each other. We wanted to introduce a dedicated platform for university students and that's why we decided to build this app. This app provides the students to share information by creating various topics such as university events, course information or any other information desired. The students seeking the information for a certain topic can be subscribe to those topics and receive information in the form of emails.

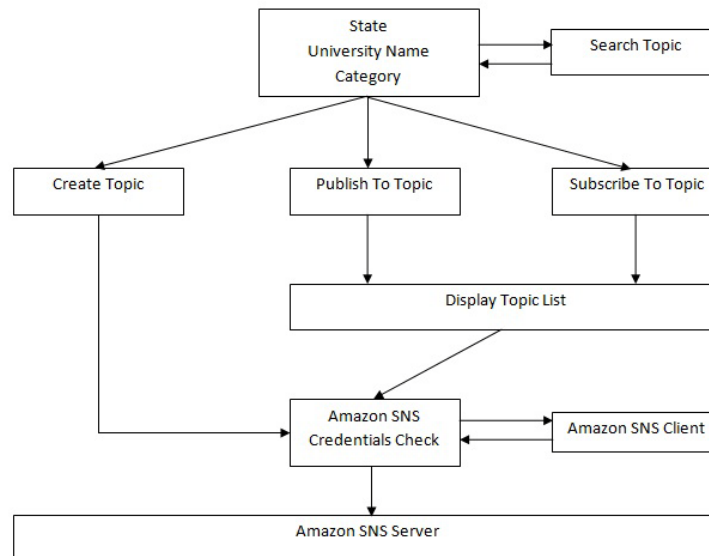
This app can also be used to communicate the buying and selling the items amongst students. Since, we provide this information to the students of same university; it becomes easy, simple and feasible to carry out these transactions.

The students can search through the existing topics before creating new topics and publish or subscribe to the existing topics, if they are present. We have maintained separate categories and topic lists for each university and students from all university will not have access to all the topics. The students will have access to the topics associated with university they select.

In this user documentation, we provide the description of our work in this project. The pictorial representation of our app is shown below. We also have described the architecture and design of our project along with our Java classes. We have provided the detailed step by step procedure to make efficient use of our app with the screenshots of our app. This makes easy and simple to understand how to use our app. We also have provided the area in which this app can be extended further to achieve better functionality and expand the scope of this app so that it can be used more efficiently.

## App Architecture and Design

While the development of this app, we have made use of Amazon Simple Notification Service. We also have implemented the pub-sub architecture since the users of our app need not be coupled in space, time or synchronization. Following is the architecture and description of our architecture.



Following are the java classes we built while achieving the goals of our app.

### CreateTopic.java

This class is used to create new topic on Amazon server. This class takes the topic name to be created. By using the Amazon SNS client, this class creates the topic on Amazon server.

### CredentialsCheck.java

This class is used to check the authenticity of the credentials used to communicate with Amazon Web Services.

### MainActivity.java

This is the MainActivity class. This class provides the main screen of our app and provides access to other operations such as creating topic, publishing to a topic, subscribing to a topic and searching a topic.

### PublishToTopic.java

This class displays the topics present in specific university under specific category. The user can publish information in the selected topics using this class.

PubSubOperations.java

This class implements the important operations to carry out publisher subscriber functionalities. This class checks the authenticity of the provided Amazon Web Services credentials. It gains access to Amazon SNS client to communicate with the Amazon server. And eventually performs the appropriate action by creating, publishing or subscribing to a topic.

SearchResults.java

This class provides the ability to search through the existing topics. It takes the topic name to be searched. Depending upon the selected university and category, this class displays the list of topics which match the search topic criteria.

SNSClient.java

This class is used to gain access to the Amazon SNS client. Once the user credentials are authenticated for Amazon Web Services, this class is used to create the object of Amazon SNS client. This object is used to perform further pub-sub operations.

SubscribeToTopic.java

This class is used to subscribe to a specific topic. Once the user subscribes to a topic, a confirmation email is sent to the email address provided. Whenever new information is added to this topic, the subscribed users receive an email notification with the new information in this topic.

UniversitySelection.java

This class provides the list of states and universities in USA. This class also provides the ability to search the existing topics. Once the state and appropriate university is selected, the user is transited to the next view where he/she can proceed with other actions such as creating topic, publishing information to a topic and subscribing to a topic.

AwsCredentials.properties

This is the properties file which stores the credentials for Amazon Web Services. The CredentialsCheck class reads the credentials from this file and validates them against Amazon Web Services.

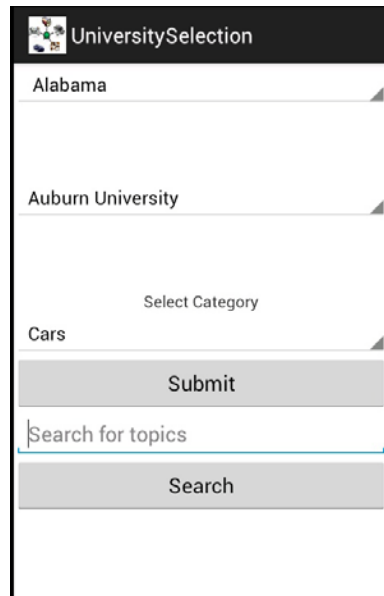
## UniPubSub User Guide

Installation

1. Minimum Android OS version required for this app is 2.3.6 GingerBread
2. Copy UniPubSub.apk file to your mobile device.
3. Enable the 'Allow installation of non-market applications' on your mobile device.
4. Open the APK file to install the app.

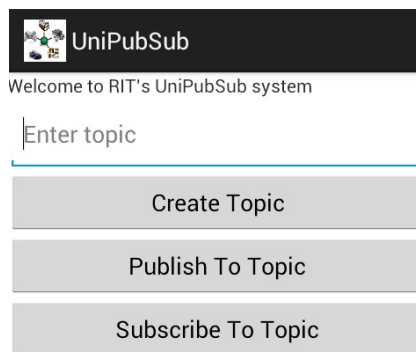
### How To Use UniPubSub

1. Run the app.
2. On the main screen, select the state. This will populate the university list in that state. Select the desired university.
3. Select the category which needs to be browsed for creation, publishing or subscribing to topics.



The screenshot shows a mobile application titled "UniversitySelection". It features three dropdown menus. The first dropdown is labeled "Alabama". The second dropdown is labeled "Auburn University". The third dropdown is labeled "Select Category" and has "Cars" selected. Below the dropdowns is a "Submit" button. At the bottom, there is a text input field labeled "Search for topics" and a "Search" button.

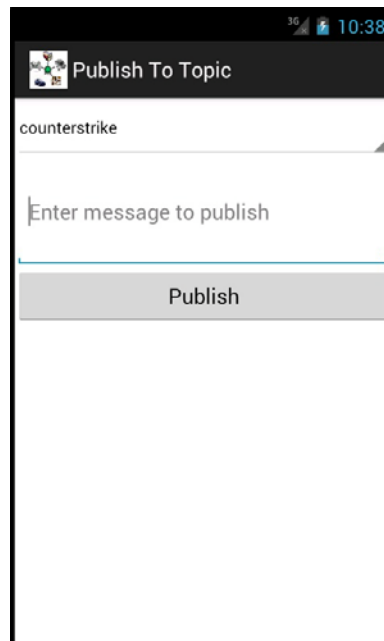
4. Click on Submit button. This will open the next view in which the user is provided with access to the main pub-sub actions.



The screenshot shows a mobile application titled "UniPubSub". It displays a welcome message: "Welcome to RIT's UniPubSub system". Below the message is a text input field labeled "Enter topic". At the bottom, there are three buttons: "Create Topic", "Publish To Topic", and "Subscribe To Topic".

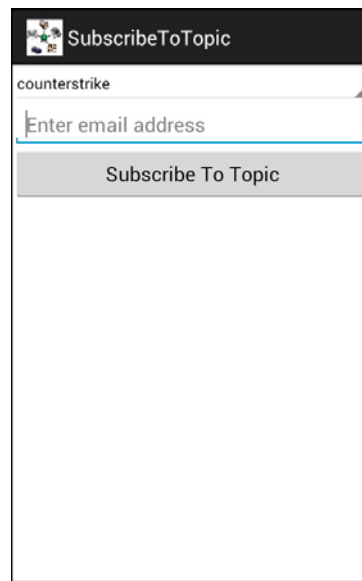
5. To create a topic, enter the name of the topic in empty text box and click on 'Create Topic'.

6. To publish to a topic click on 'Publish To Topic'. This opens a new view which displays the list of topics for selected university and category. Enter the information in this topic and click on 'Publish'. This will publish the entered information in that topic.



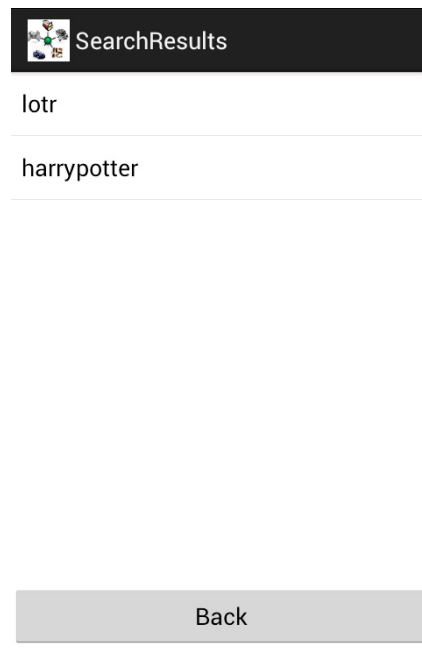
The screenshot shows a mobile application interface titled "Publish To Topic". At the top, there is a status bar with the time "10:38" and battery level "35%". Below the title bar, there is a dropdown menu currently showing "counterstrike". Underneath the dropdown is a text input field with the placeholder text "Enter message to publish". At the bottom of the form is a grey button labeled "Publish".

7. To subscribe to a topic click on 'Subscribe To Topic'. This opens a new view which displays the list of topics for selected university and category. Select the desired topic and enter the email address at which you would like to receive email notifications. An email will be sent to confirm your email address. Click on the URL provided in the email to confirm this email address. Once the email address is confirmed, the user will receive the email notifications for new information published in this topic.



The screenshot shows a mobile application interface titled "SubscribeToTopic". At the top, there is a status bar with the time "10:38" and battery level "35%". Below the title bar, there is a dropdown menu currently showing "counterstrike". Underneath the dropdown is a text input field with the placeholder text "Enter email address". At the bottom of the form is a grey button labeled "Subscribe To Topic".

8. Users can unsubscribe from receiving emails from the emails they receive for the new information publishing.
9. On the main screen where list of state and universities is displayed, the option for searching through existing topics is also provided.
10. To search through existing topics, select state, university and category as mentioned in steps 2 and 3.
11. Enter the topic to be searched and click on 'Search' button. If the search is found, this will display the topics present in that category corresponding to that university.



## Future Work

Since, this is the first version of our app development, there are opportunities to extend this work and improve this app for better functionality and usage. Following are some of the areas in which this app can be improved.

1. Improving app security by adding spam filters
2. Displaying published information on mobile devices and allowing to insert images in information publishing
3. Provide personalized usage of UniPubSub (Student Account Management)
4. Implement permissions for publishers and subscribers to restrict their access to all topics, if required.
5. Support for other platforms such as iOS, Windows.

## References

1. Amazon Simple Notification Service  
<http://aws.amazon.com/sns/>
2. Craigslist Mobile App  
<http://www.craigslist.org/>
3. Android App Development  
<http://developer.android.com/develop/index.html>
4. Stack Overflow  
<http://stackoverflow.com/>