

COP290: User Registration App

Sai Deep (2012CS10223)
Mano Teja (2010CS50286)
TRK Saran (2010BB50042)

February 26, 2016

Contents

1	Introduction	1
2	User Interface	2
2.1	Primary Screen	2
2.2	Display of Errors	3
2.3	On Pressing Register Button	4
2.4	Long Press Register Button to Clear All the Fields	5
2.5	Press Back Button Twice to Exit the App	6
2.6	Progress Bar	7
3	Implementation Details	7
3.1	Validation of Text Field Inputs	8
3.1.1	ValidateName	8
3.1.2	ValidateEntry	8
3.2	Methods for network communication	9
4	VCS	9

1 Introduction

Our task was to design an Android application to register teams for COP290 course.

2 User Interface

2.1 Primary Screen

A scrollable layout with one field for entering team name, 3 fields each for names and entry numbers respectively followed by a **Register Now** button.

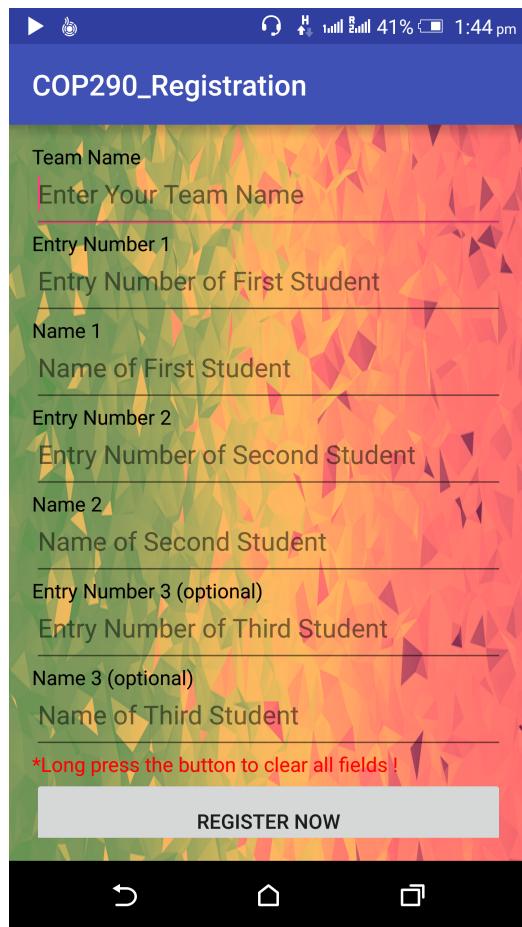


Figure 1: Primary Screen

2.2 Display of Errors

The fields are processed one by one. The first invalid text field is highlighted. Subsequent invalid fields are highlighted one by one only after the previous invalid one is corrected.

This screenshot shows the COP290_Registration application interface. At the top, it says "COP290_Registration". Below that is a form with several input fields. The first field, "Team Name", is empty and has a red border, indicating it is required. A tooltip box is overlaid on this field with the text "This is a required field!". The other fields in the form are empty and have standard black borders. At the bottom of the form is a button labeled "REGISTER NOW".

Figure 2: Empty field (Applicable for first 5 fields)

This screenshot shows the COP290_Registration application interface. At the top, it says "COP290_Registration". Below that is a form with several input fields. The first field, "Entry Number 1", contains the value "2012CS10234" and has a red border, indicating it is required. The second field, "Name 1", contains the value "Darth Vader" and has a standard black border. The third field, "Entry Number 2", contains the value "2011CS10245" and has a standard black border. The fourth field, "Name 2", contains the value "Leader Snake" and has a standard black border. The fifth field, "Entry Number 3 (optional)", contains the value "Sai" and has a standard black border. A tooltip box is overlaid on the "Entry Number 3 (optional)" field with the text "If you are submitting the third member's name, you should fill his entry number too!". At the bottom of the form is a button labeled "REGISTER NOW".

Figure 3: When only one of the two optional fields is filled

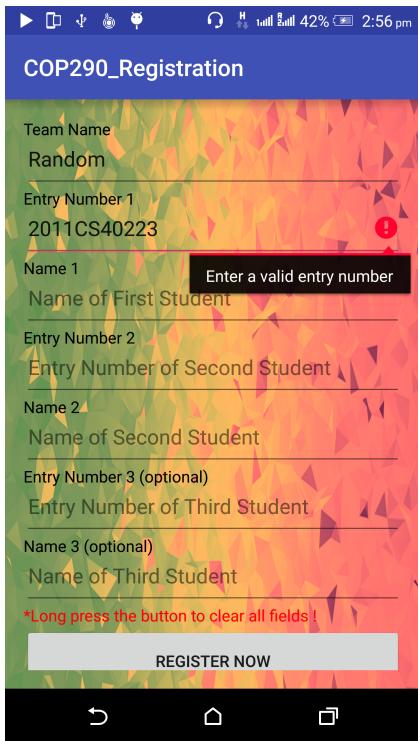


Figure 4: Invalid Entry

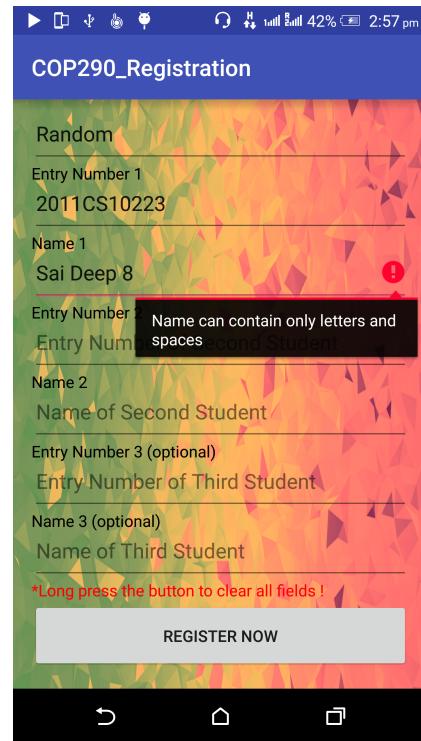


Figure 5: Invalid Name

2.3 On Pressing Register Button

Register button sends the data to server only if all the entered fields are valid. Below are the screens illustrating possible responses from the server which is shown as a toast message.

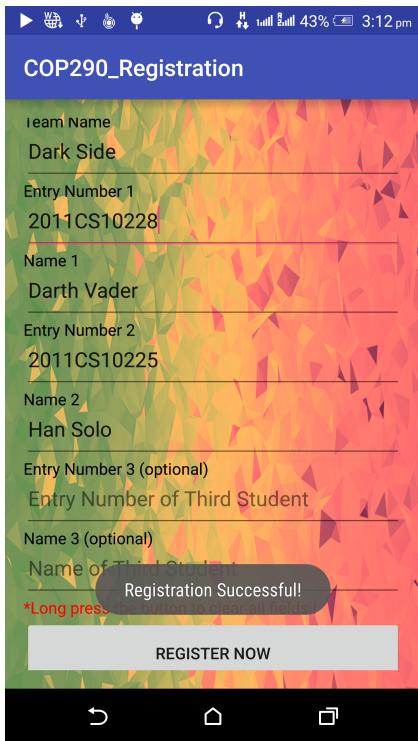


Figure 6: Successful Registration

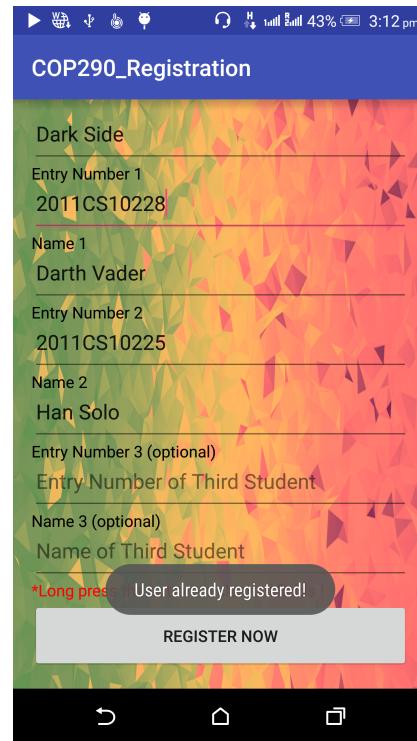


Figure 7: User already registered

2.4 Long Press Register Button to Clear All the Fields

Deleting text in each field can be tedious. So a new feature has been added by which all the text fields can be cleared by long pressing the register button[2].

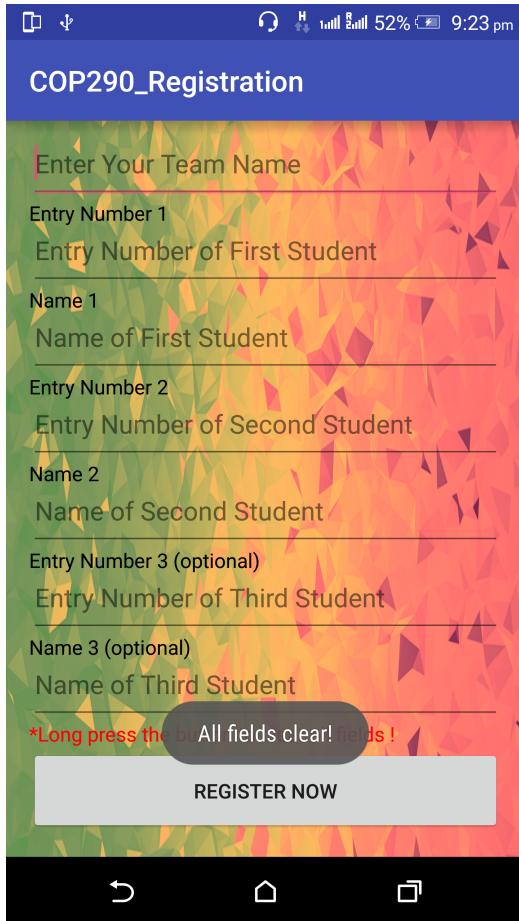


Figure 8: Clear Fields

2.5 Press Back Button Twice to Exit the App

When the back button is clicked to exit the application, a Toast comes up with a message **Press again to exit[3]**.

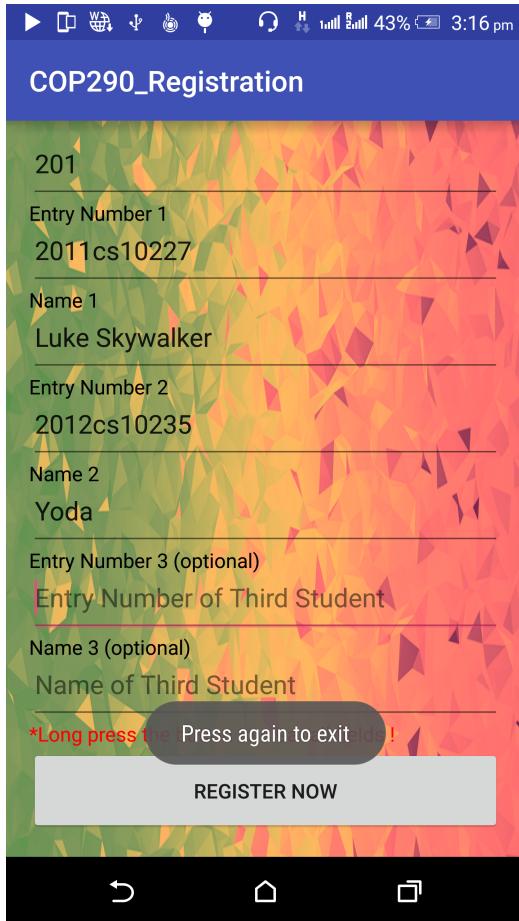


Figure 9: Back Button Twice

2.6 Progress Bar

A progress bar appears on clicking register button when all the text inputs are valid and keeps spinning till response is received from server after which it disappears.

3 Implementation Details

There are two classes/modules : **Main Activity & Validate** .

Main Activity consists of methods to describe what actions are to be performed on Click, longClick etc. It also contains methods for the network communications where input data is put into a hash map as (key,value) pairs and sent to the server as a string request.

3.1 Validation of Text Field Inputs

Validate Class contains two methods - **ValidateName** & **ValidateEntry** for validating name & entry Number fields respectively in order to make sure that information entry is not ambiguous.

3.1.1 ValidateName

To make sure that the user is entering a valid entry number, **only alphabet and spaces are allowed**. To check whether the entered text is a valid name it is matched with the following regular expression

[a-zA-Z]+(+[a-zA-Z]+)*

[4]

3.1.2 ValidateEntry

This method is to make sure that the user is entering a valid entry number.

- Since COP290 is an undergraduate course, we have assumed that **PG entry numbers are invalid**. This restricts the length of entry numbers to **11 characters**.
- Also we have restricted the domain of students registering this course to be between **2005 & 2014** years of entry (*2005 in the worst case).
- 2015 entry students are also ineligible to register for COP290.
- We took into consideration the curriculum changes that took place in the years 2013 & 2014. Following is the domain of valid branches in an entry number
 - **UG Programs that are part of 2012 or prior Curriculum**
BB5, CH1, CH7, CS1, CS5, CE1, EE1, EE2, EE5, ME1, ME2, MT5, PH1, TT1;

- **UG Programs that are part of 2013 Curriculum**
BB5, CH1, CH7, CS1, CS5, CE1, EE1, EE3, ME1, ME2, MT6, PH1, TT1;
- **UG Programs that are part of 2014 Curriculum**
BB1, BB5, CH1, CH7, CS1, CS5, CE1, EE1, EE3, ME1, ME2, MT1, MT6, PH1, TT1;
- Summing up all the above points , our definition of a valid entry number is the one which obeys the following rules :
 - Length of the entry number is 11
 - First 4 characters (**YEAR**) has to be a Valid Year (i.e 2005-2014)
 - Next 3 characters (**BRANCH**) has to be a Valid UG Programme (i.e **contains(BRANCH, YEAR) == true**)
 - *Next 4 characters has to be a Valid Number (All digits)*

3.2 Methods for network communication

In this assignment, we used Volley, an HTTP library that makes networking for Android applications easier [1].

4 VCS

The code for the project is being maintained in this repository:
https://saideepbsd@bitbucket.org/saideepbsd/cop290_a1.git

References

- [1] Android volley post tutorial. <https://www.simplifiedcoding.net/android-volley-post-request-tutorial/>.
- [2] Clear all fields. <http://stackoverflow.com/questions/5740708/android-clearing-all-edittext-fields-with-clear-button>.
- [3] Press back button twice to exit. <http://stackoverflow.com/questions/8430805/android-clicking-twice-the-back-button-to-exit-activity>.

- [4] Regex for valid names. <http://stackoverflow.com/questions/27126455/java-regex-to-validate-full-name-allow-only-spaces-letters-commas-dots-in-java>.