**Santa Clara University**

16

**EventShare**

**Final Project Submission**

**Anusha Govindan-W1167283**

**Pragati Shrivastava-W1129451**

**Shilpa Shetty-W1111484**

# Project Goal

Application to manage daily events and task by sending event reminders to the friends, colleagues, clients and yourself (without the participants requiring to install the eventshare app).

# Team Members

* Pragati Shrivastava-00001129451-pshrivastava@scu.edu
* Shilpa Shetty-00001111484-s1shetty@scu.edu
* Anusha Govindan- 00001167283-agovindan@scu.edu

# Major functionalities since milestone 2

* Edit/Delete: This class provides the functionality to update event with events details like  Event description, location, attendees, time before which a reminder has to be sent, start date and end date, attachments.
* Reminder(Email/Sms): Email/SMS is sent as reminder to the attendees
* Attachment: Files/images can be added to the event.
* Group Chat: Event attendees can participate in a group chat.

# Future Prospects

* Share images via chat
* Adding Camera intent to chat
* Customize view based on Event Type
* Voice Call
* Integrating with social networking sites

# Labor Division

## Create Events - Anusha

* Creating an event, sending notification through Google calendar
* Storing SMS details in the event using Extended Properties
* Configuring app with chips to fetch Email/SMS contact details from the phone.
* Performed POC for AutocompleteTextview to include email and SMS contacts.
* Search location using Google places.

## Editing/Delete events Event Details-Anusha

* Editing attendees based on notification mode (SMS/Email)
* Google Maps to Navigate in view event.
* Deleting Events.

## View Events-Anusha, Pragati

* View and navigate to location
* Configuration for viewing attachment
* Sync with Google Calendar for different type of events (including all day events)

## Reminder-Pragati

Reminding the attendees through Email/SMS before the event using Alarm manager and Notification Manager

## Attachment-Pragati

Attaching files/images to events while create/edit events thro Google drive

## API-Pragati

Performed POC to configure, read and access files using Google Drive API

## Chat-Shilpa

Event Specific Group Chat Functionality for the participants using Dynamo DB. Service is used to start a thread which polls the DB for any change in the data.

## API-Shilpa

Performed POC to configure ,write and read google API data and send email via API.

## SMS-Shilpa

Configured SMS manager to send SMS.

## List View- Pragati,Shilpa

Upcoming List of Events for the User

**Integration, Testing and UI Pages:** All three members

# Prerequisites for the Developer to run the Code in Android Studio

## Configuration for setting up project and enabling the Google Calendar API

1. In a terminal, run the following key utility command to get the SHA1 fingerprint you will use to enable the API.
   1. Keytool -exportcert -alias androiddebugkey -keystore ~/.android/debug.keystore -list -v
   2. Enter the keystore password as android
2. Login to your Google account and go to the below link

[https://console.developers.google.com/flows/enableapi?apiid=calendar](https://console.developers.google.com/flows/enableapi?apiid=calendar" \t "_blank)

* 1. Select Create a new project
  2. Click Continue, Go to credentials

1. Add credentials to your project page
   1. Select calendar api and android setting. Select data accessing as user data.
   2. Click on what credentials do I need button.
2. Copy SHA1 fingerprint from the first step into the fingerprint field
   1. Add package name as edu.scu.eventshare
   2. Click on create button
3. Set up Auth 2.0
   1. Select an email address
   2. Give a product name
   3. Click continue and then click on done.
4. Android device must have ‘Google Calendar’ installed and ‘Google account’ associated with it.

## Configuration for Google Places:

1. Enable Google Places in your Google Developer Console for you project
2. Generate the API key for the project package using the SHA1 Key
3. Add the API key in your manifest like the following

<**meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="A43k34FEFfaDFADFSFSD34343DFSFS"** />

Demo Instruction

* Copy the eventshare folder into the project folder
* Open the project in Android Studio and run in on an android device
* The android device should have a ‘Google Calendar’ and ‘Google account’ associated with it so that the app directly inserts event into the Google calendar as we are using Google API to achieve this.
* In order to run the app, please follow the steps in the Youtube link.
* Part 1: <https://www.youtube.com/watch?v=g8I_k7oqQ7M&feature=youtu.be>
* Part 2: <https://www.youtube.com/watch?v=QVt-0BXovBU&feature=youtu.be>

Points to Note:

* The android device in which the app is running must have a Google account associated with it.