Crowd Control

Senior Design Final Documentation

Bowtaps

Johnathan Ackerman Daniel Andrus Charles Bonn Evan Hammer
Joseph Mowry

May 1, 2016

Contents

List of Figures

List of Tables

List of Algorithms

Overview Statements

0.1 Mission Statement

Our mission at Bowtaps is to develop innovative mobile software applications to provide solutions to inconveniences that trouble the everyday user. With our software, we plan on changing the mobile environment by creating applications that are easy to use with intuitive interfaces and reliable services for everyday use.

0.2 Elevator Pitch

Our company, Bowtaps, is developing an iPhone/Android app to help young adults and event-goers stay in contact with friends while in loud and crowded places using group messaging and GPS features.

Our product, Crowd Control, is designed to become an essential element for groups looking to go out together by providing both powerful group-management tools and interesting nearby outing suggestions, such as local events, concerts, and pub crawls.

We will work with local businesses and event planners to sponsor these suggestions. This will generate content for our users, visibility for our sponsors, and revenue for ourselves.

We plan to release the app for free in early-to-mid summer of 2016.

Document Preparation and Updates

Current Version [1.0.0]

Prepared By:
Johnathan Ackerman
Daniel Andrus
Charles Bonn
Evan Hammer
Joseph Mowry

Revision History

Date	***Author***	***Version***	***Comments***
1/8/14	*Charles Bonn*	*1.0.0*	*Refactor to new design document*

1

Overview and concept of operations

1.1 Bowtaps and its team

Bowtaps is a start up company out of SDSM&T created by the team members of Bowtaps. Our goal is to create easy to use software applications that help ease the everyday life of the user. Bowtaps currently consists of the members Charles Bonn, Johnathan Ackerman, Daniel Andrus, Evan Hammer, and Joesph Mowry.

1.2 Crowd Control

Our flag-ship product(Crowd Control) is to create a mobile application that combines GPS tracking, group messaging and group management features into one easy to use application.

1.2.1 Purpose of the System

Crowd Control is a mobile application designed to ease the experience of going out though the implementation of integrated group messaging, GPS tracking and group management features. Along with the features to manage your group at the event Crowd Control also gives suggestions of local events, restaurants and attraction. This allows the group to continue even when the next item on the agenda is a mystery.

Even though Crowd Control is designed for the party scene, and people going out to events; it's uses can be expanded to fit more purposes. Crowd Control can be used to help manage any kind of group at an event such as church groups, tour groups, or school field trips.

1.3 Business Need

(TODO)

1.4 Deliverables

(TODO)

1.5 System Description

Behind the UI, Crowd Control is written using an MVC architecture. IOS is written natively in Swift under the IDE XCode. On the Android side, the code is written in Native Mobile Java under Android Studio.

The code for Android uses xml files for layouts which act as the view in MVC. Each activity acts as a controller. Each one of the models has an interface, which is how the controllers get access to the functions and data provided by the model. Each interface is set up in such a way that it uses OOP inheritance to generalize the models, thus abstracting our third party software.

(TODO) IOS??

1.5.1 Integrated Group Messaging

Integrated group messaging is an important feature of Crowd Control. It allows for communication between cross platform, different phone brands, and different carriers. This allows for seamless communication between users with out the issues associated with messaging such as messages not using the same format, messages not going to all recipients, and messages with users in the group that you do not want to have your personal information.

Currently this is handled by a third party called Sinch. Sinch messaging handles the encryption of messages for the security of our users. Also, Sinch uses app to app messaging. In this way, any device with Crowd Control can send a message to other group members. However, since our app is currently only implemented on Android, messages can only be passed between Android users. It only needs either an internet connection or cell service to function.

1.5.2 GPS Location services

GPS allows for tracking of members in the group on a local map of the area. With this feature you will be able to keep track of anyone in the group off of their last GPS check in. This is useful to help locate members of the group that maybe lost or unable to be located. This feature will have the option of being able to opt out when the user does not want to have their location known to the group. When the users battery is low it will allow for the check in period to be extended or turned off to save battery life.

(TODO)

1.5.3 Group Management Features

The group management features allow for information to be shared with the group. A group management menu will allow for a group agenda to be posted as well as updates when the agenda changes. With the GPS features it will allow for the group leader to set way-points for the group.

1.5.4 Suggestions

Suggestions are both a plus for the user and our way of making revenue. Suggestions are sponsored by local businesses in the form of an ad. Although these are not traditional ads, they are in the form of local points of interest such as restaurants, bars, amusement parks, or bowling allies. The possibilities are endless. With the suggestion method it will allow for our users to have helpful suggestions of places for their group to attend as well as exposure for the local businesses that are sponsoring Crowd Control.

1.6 System Overview and Diagram

The basic overview of Crowd Control can be seen in the diagram below. See Figure 1.1. Crowd Control will be using a model-view-controller design structure. With the model view controller design method we are able to abstract the user interface from the control structures that will communicate with the third party services such as Parse, Google play services, or Sinch. The model of each respective operating system (Android or iOS) will be able to communicate with the respective mapping feature (Google Play Services or Apple Map Features). While both models will be able to communicate with Parse, our back end server. Though Parse, using their features, will be able to connect user profiles to their Facebook and twitter accounts for faster log in.

1.7 Technologies Overview

Some technologies used in the creation of Crowd Control are Google Play Services, Apple Map Features, Parse, Sinch, and Android Studio.

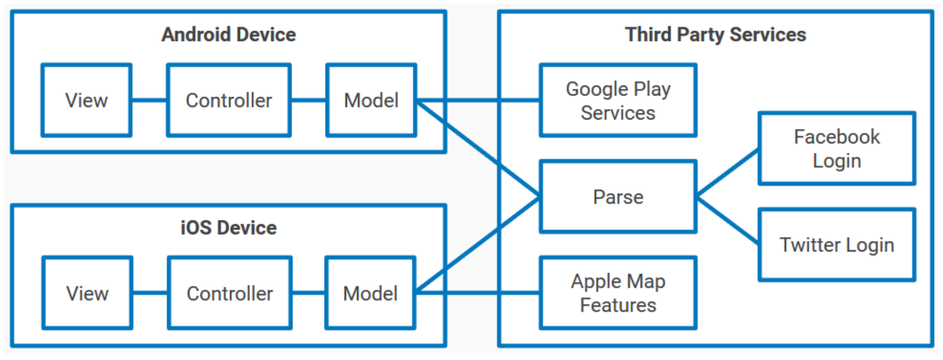


Figure 1.1: Basic System Flow Diagram

1.7.1 Google Play Services

1.7.1.a Description

Google Play Services contains the native android API for mapping features. With this it allows for communication between a map and your GPS location along with other mapping features.
REFERENCE LINK: `https://developers.google.com/android/guides/setup`

1.7.1.b Usage

Google Play Services will be used on the Android device as the default map. We chose to go with Google Play services to give android users a more native feel when it comes to using the mapping features. This allows for a less intrusive feel when it comes to using Crowd Control, will be used for displaying your location on a map, displaying other users in your group on a map, and displaying event suggestions on the map.

1.7.2 Apple Map Features

1.7.2.a Description

Apple Map Features is the native iOS API for mapping features. With this it allows for commiseration between a map and your gps location along with other mapping features.
REFERENCE LINK: `https://developer.apple.com/maps/`

1.7.2.b Usage

Apple Map Features will be used on the iOS device as default. We chose to go with Apple Map Features to give iOS users a more native, less intrusive feel. This will be used for displaying your location, displaying other users from your group, and displaying local event suggestions on the map.

1.7.3 Parse

1.7.3.a Description

Parse is abstracted behind all of our models. Our usage of it is restricted to the functions we provided ourselves though the implementations of our models, keeping all of the actual parse code out of the controllers. Parse gives us access to a web-based database that is fully protected by an experienced third party.
REFERENCE LINK: `http://parse.com/`

1.7.3.b Usage

Parse is abstracted behind all of our models. Our usage of it is restricted to the functions we provided ourselves though the implementations of our models, keeping all of the actual parse code out of the controllers. Parse is our back-end database. It saves all the group information into a web accessible parse database, as well as storing personal log in information to another Parse database local to the phone. The globally stored information is paced among members of the group, were as the locally stored information is used for automatic log in at the convenience of our users. The web-based storage will also hold all of the location values(though encrypted), and by using a service, will keep those value up-to-date on any given device.

1.7.4 Sinch

1.7.4.a Description

Sinch is a third party, device to device, communication API. We have selected it for its encryption, and ready to use app-to-app messaging platform. This platform will work with either wi-fi connection, or cell service.
REFERENCE LINK: `https://www.sinch.com/`

1.7.4.b Usage

Sinch has its own service to handle the sending and receiving of messages. We have constructed a fragment(with the help of some Sinch code) to control a user interface to grab messages passed by the user. We have also modified the basic one-to-one message sending to send the message to the entire group.

2

User Stories, Requirements, and Product Backlog

2.1 Overview

This document contains the features, creation and develpment of crowd control. It covers prerequsit user stories, to the design and implimentation of the application its self.

2.2 User Stories

2.2.1 User Story #1

As a user i want to be able to join a group.

2.2.1.a User Story #1 Breakdown

As a user i want the ability to join a group. Group joining options would be from a list or from an invite from a user.

2.2.2 User Story #2

As a user i want the abilitiy to track locations of other members in the group.

2.2.2.a User Story #2 Breakdown

2.2.3 User Story #3

As a user i want post agenda for the group.

2.2.4 User Story #4

As a user i want to i want the abilitiy to look for local groups

2.2.5 User Story #5

As a user i want the ability to have suggestions of local activities.

2.2.6 User Story #6

As a user i want the ability to leave a group.

2.2.7 User Story #7

As a user i want the ability to have a list of local groups.

2.2.8 User Story #8

As a user i want the abilitiy to login.

2.2.9 User Story #9

As a user i would like to message other members of the group.

2.2.10 User Story #10

As a user i would like my information protected.

2.3 Requirements and Design Constraints

This section will cover the main design requirement in all aspects of crowd control.

2.3.1 System Requirements

Sense there we are creating Crowd Control to run on two different platforms, both iOS and Android, there are two sets of requirements that will be similar between both platforms. Even though they are both similar, implimentation between both will be differnet. With them both being different they are split into two sections as listed below.

2.3.1.a iOS Requirements

- Use Apple Mapping Features
- Access Parse as the Database

2.3.1.b Android Requirements

- Use Google Maps
- Access Parse as the Database

2.3.1.c Parse Requirements

- Delete groups when group is not in use

2.3.2 Network Requirements

Network requrements are mobile networks as this is a mobile applications. The requirement on our part is making sure that the application is able to reach the server and use at little data as possible when connected to the network. Making sure we use as little data as possible will help our users not use all of their data.

2.3.3 Development Environment Requirements

The development enviroment requirement is that Crowd Control be avalabe on both iOS and Android platforms. Being cross platform allows for us to reach as many users as possible. Android development will be handled with Android Studio and iOS will be developed with xCode.

2.3.4 Project Management Methodology

We have set restrictions on the developemnt of Crowd Control and are listed as follows:

- GitHub issues will be used to keep track of current status as well as backlogs for the product.
- There will be 6 total sprints over 2 scimesters for this products.
- The sprint cycles are 3 weeks long.
- Progress reports will be summited to Dr. McGough and Brian Butterfeild at the end of each sprint.
- Github will be used for source control.

2.4 Specifications

2.5 Product Backlog

T

- What system will be used to keep track of the backlogs and sprint status?
- Will all parties have access to the Sprint and Product Backlogs?
- How many Sprints will encompass this particular project?
- How long are the Sprint Cycles?
- Are there restrictions on source control?

2.6 Research or Proof of Concept Results

The Proof of conecpt is a rough design that impliments basic features of Crowd Control. Basic features are currently under construction. This is currently a functional prototype with improvements in the future.

Below are screen shots of both android and iOS proof of concepts. (current formatting issues need to fix)

2.6.1 iOS Proof of Concept Screen Shots

Below are screen shots from the iOS version of CrowdControl.

2.6.2 Android Proof of Concept Screen Shots

Below are screen shots from the Android version of CrowdControl.

2.7 Supporting Material

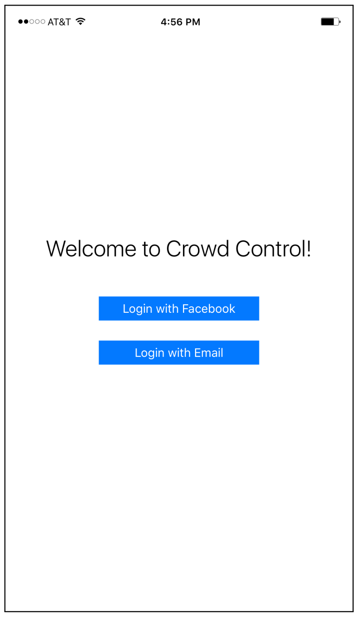


Figure 2.1: iOS login select screen

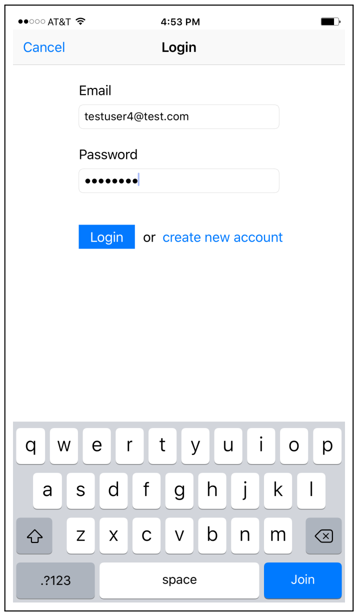
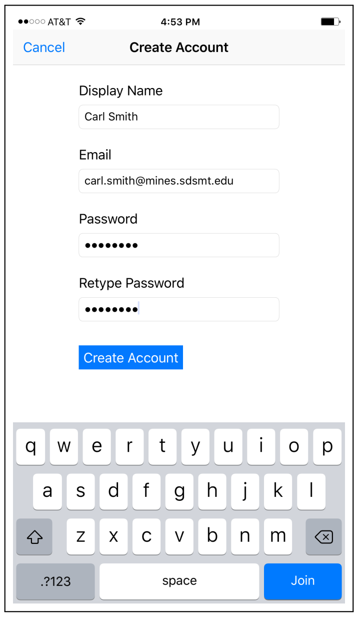


Figure 2.2: iOS email login screen



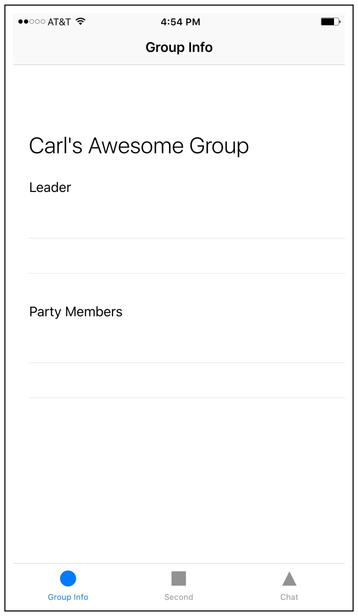
Display Name
Carl Smith

Email
carl.smith@mines.sdsmt.edu

Password
••••••••

Retype Password
••••••••

Figure 2.3: iOS create account screen



Carl's Awesome Group

Leader

Party Members

Figure 2.4: iOS group infomation screen

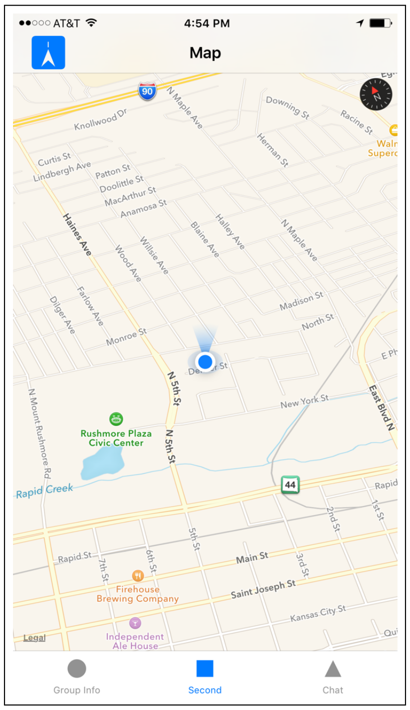
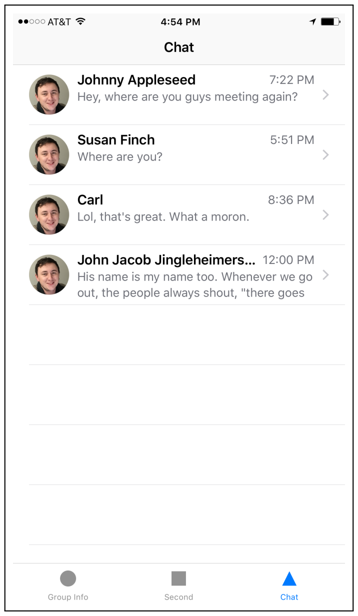


Figure 2.5: iOS map view screen



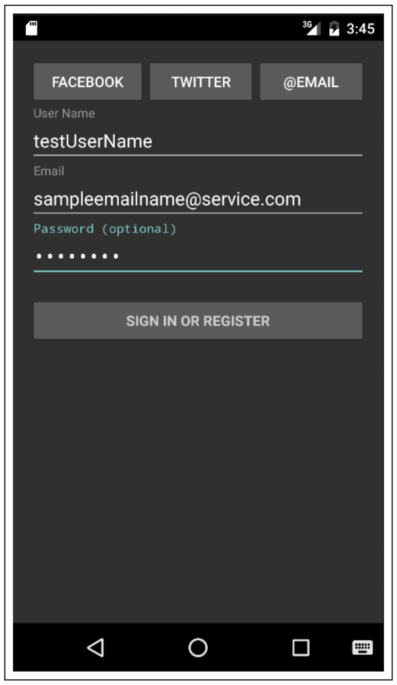
Johnny Appleseed 7:22 PM
Hey, where are you guys meeting again?

Susan Finch 5:51 PM
Where are you?

Carl 8:36 PM
Lol, that's great. What a moron.

John Jacob Jingleheimers... 12:00 PM
His name is my name too. Whenever we go out, the people always shout, "there goes

Figure 2.6: iOS messaging main screen



User Name
testUserName
Email
sampleemailname@service.com
Password (optional)
••••••••

Figure 2.7: Android login screen

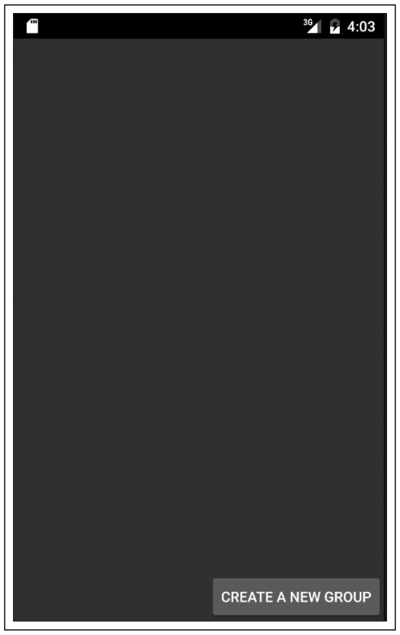
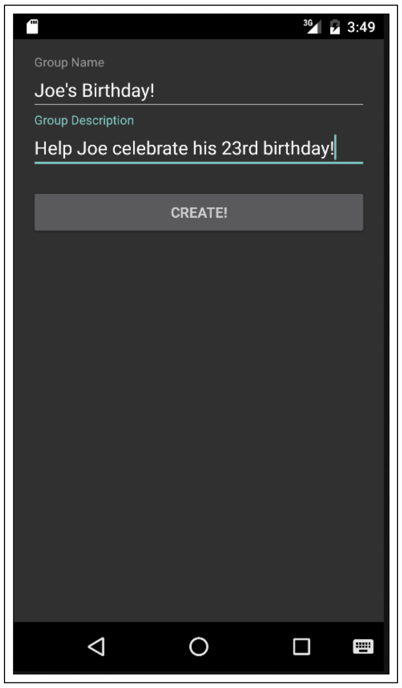


Figure 2.8: Android create group screen



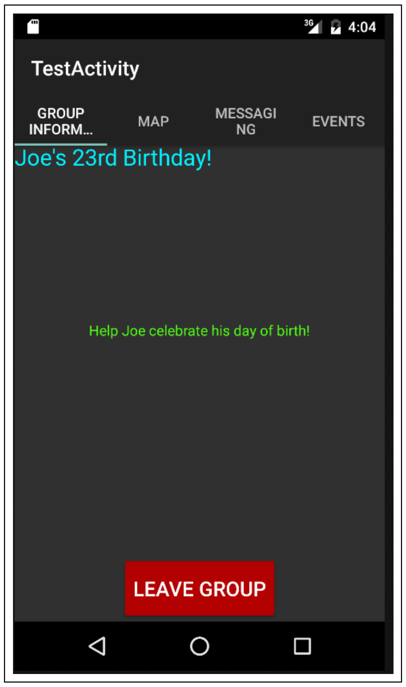
Group Name

Joe's Birthday!

Group Description

Help Joe celebrate his 23rd birthday!

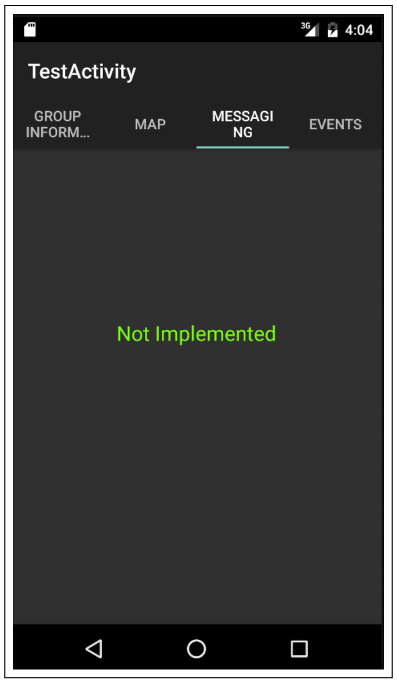
Figure 2.9: Android group information screen



Joe's 23rd Birthday!

Help Joe celebrate his day of birth!

Figure 2.10: Android group join screen



Not Implemented

Figure 2.11: Android messaging main screen

3

Project Overview

This section provides some housekeeping type of information with regard to the team, project, environment, etc.

3.1 Team Member's Roles

Johnnathon Ackerman - Johnnathon is leading the GUI design and implimentation side for the android version of Crowd Control. This entails:

1. Graphical Design
2. Smooth Moving Interfaces
3. Easy to Use and learn layout

Daniel Andrus - Daniel is leading the Gui design ad implimentation for the IoS version of Crowd Control. This entails:

1. Graphical Design
2. Smooth Moving Interfaces
3. Easy to Use and learn layout

Charles Bonn - Charles is leading the database side of Crowd Control. This database is for both IoS and andriod versions. This entails:

1. Creating and managing database qurries
2. Creating Cloud Code to manage database information
3. Database load testing

Charles is also working on future encryption of data going to and from the database.

Evan Hammer - Evan is leading the backend side for the IoS version of Crowd Control. This entails:

1. Creating links from the database to the mobile application
   (a) Login link
   (b) Group Join Link
   (c) Group Member
2. Creating links to Apple maps to the mobile application

Joseph Mowry - Joseph is leading the backend side for the android version of Crowd Control. This endtails:

1. Creating links from the database to the mobile application

   (a) Login link
   (b) Group Join Link
   (c) Group Member

2. Creating links to Apple maps to the mobile application

3.2 Project Management Approach

This section will provide an explanation of the basic approach to managing the project. Typically, this would detail how the project will be managed through a given Agile methodology. The sprint length (i.e. 2 weeks) and product backlog ownership and location (ex. Trello) are examples of what will be discussed. An overview of the system used to track sprint tasks, bug or trouble tickets, and user stories would be warranted.

3.3 Stakeholder Information

This section would provide the basic description of all of the stakeholders for the project. Who has an interest in the successful and/or unsuccessful completion of this project?

3.3.1 Customer or End User (Product Owner)

Who? What role will they play in the project? Will this person or group manage and prioritize the product backlog? Who will they interact with on the team to drive product backlog priorities if not done directly?

3.3.2 Management or Instructor (Scrum Master)

Who? What role will they play in the project? Will the Scrum Master drive the Sprint Meetings?

3.3.3 Investors

Are there any? Who? What role will they play?

3.3.4 Developers –Testers

Who? Is there a defined project manager, developer, tester, designer, architect, etc.?

3.4 Budget

Describe the budget for the project including gifted equipment and salaries for people on the project.

3.5 Intellectual Property and Licensing

Describe the IP ownership and issues surrounding IP.

3.6 Sprint Overview

If the system will be implemented in phases, describe those phases/sub-phases (design, implementation, testing, delivery) and the various milestones in this section. This section should also contain a correlation between the phases of development and the associated versioning of the system, i.e. major version, minor version, revision.

All of the Agile decisions are listed here. For example, how do you order your backlog? Did you use planning poker?

3.7 Terminology and Acronyms

Provide a list of terms used in the document that warrant definition. Consider industry or domain specific terms and acronyms as well as system specific.

3.8 Sprint Schedule

The sprint schedule. Can be tables or graphs. This can be a list of dates with the visual representation given below.

3.9 Timeline

Gantt chart or other type of visual representation of the project timeline.

3.10 Backlogs

Place the sprint backlogs here. The product backlog will be in the chapter with the user stories.

3.11 Burndown Charts

Place your burndown charts, team velocity information, etc here.

3.12 Development Environment

The basic purpose for this section is to give a developer all of the necessary information to setup their development environment to run, test, and/or develop.

3.13 Development IDE and Tools

Describe which IDE and provide links to installs and/or reference material.

3.14 Source Control

Which source control system is/was used? How was it setup? How does a developer connect to it?

3.15 Dependencies

Describe all dependencies associated with developing the system.

3.16 Build Environment

How are the packages built? Are there build scripts?

3.17 Development Machine Setup

If warranted, provide a list of steps and details associated with setting up a machine for use by a developer.

4

Design and Implementation

This section is used to describe the design details for each of the major components in the system. Note that this chapter is critical for all tracks. Research tracks would do experimental design here where other tracks would include the engineering design aspects. This section is not brief and requires the necessary detail that can be used by the reader to truly understand the architecture and implementation details without having to dig into the code. Sample algorithm: Algorithm 1. This algorithm environment is automatically placed - meaning it floats. You don't have to worry about placement or numbering.

Algorithm 1 Calculate $y = x^n$

```
Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
  $y \Leftarrow 1$
  if $n < 0$ then
    $X \Leftarrow 1/x$
    $N \Leftarrow -n$
  else
    $X \Leftarrow x$
    $N \Leftarrow n$
  end if
  while $N \neq 0$ do
    if $N$ is even then
      $X \Leftarrow X \times X$
      $N \Leftarrow N/2$
    else {$N$ is odd}
      $y \Leftarrow y \times X$
      $N \Leftarrow N - 1$
    end if
  end while
```

Citations look like [**?**, **?**, **?**] and [**?**, **?**, **?**]. These are done automatically. Just fill in the database `designrefs.bib` using the same field structure as the other entries. Then pdflatex the document, bibtex the document and pdflatex twice again. The first pdflatex creates requests for bibliography entries. The bibtex extracts and formats the requested entries. The next pdflatex puts them in order and assigns labels. The final pdflatex replaces references in the text with the assigned labels. The bibliography is automatically constructed.

4.1 Architecture and System Design

This is where you will place the overall system design or the architecture. This section should be image rich. There is the old phrase *a picture is worth a thousand words*, in this class it could be worth a hundred points (well if you sum up over the entire team). One needs to enter the design and why a particular design has been done.

4.1.1 Design Selection

Failed designs, design ideas, rejected designs here.

4.1.2 Data Structures and Algorithms

Describe the special data structures and any special algorithms.

4.1.3 Data Flow

4.1.4 Communications

4.1.5 Classes

4.1.6 UML

4.1.7 GUI

4.1.8 MVVM, etc

4.2 Major Component #1

4.2.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.2.2 Component Overview

This section can take the form of a list of features.

4.2.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.2.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.2.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.2.6 Design Details

This is where the details are presented and may contain subsections. Here is an example code listing:

```c
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;
    // Line comment.
    puts("Hello world!");

    for (i = 0; i < N; i++)
    {
        puts("LaTeX is also great for programmers!");
    }

    return 0;
}
```

This code listing is not floating or automatically numbered. If you want auto-numbering, but it in the algorithm environment (not algorithmic however) shown above.

4.3 Major Component #2

4.3.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.3.2 Component Overview

This section can take the form of a list of features.

4.3.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.3.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.3.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.3.6 Design Details

This is where the details are presented and may contain subsections.

4.4 Major Component #3

4.4.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.4.2 Component Overview

This section can take the form of a list of features.

4.4.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.4.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.4.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.4.6 Design Details

This is where the details are presented and may contain subsections.

5 System and Unit Testing

This section describes the approach taken with regard to system and unit testing.

5.1 Overview

Provides a brief overview of the testing approach, testing frameworks, and general how testing is/will be done to provide a measure of success for the system.

Each requirement (user story component) should be tested. A review of objectives and constraints might be needed here.

5.2 Dependencies

Describe the basic dependencies which should include unit testing frameworks and reference material.

5.3 Test Setup and Execution

Describe how test cases were developed, setup, and executed. This section can be extremely involved if a complete list of test cases was warranted for the system. One approach is to list each requirement, module, or component and describe the test.

The unit tests are described here.

5.4 System Testing

5.5 System Integration Analysis

5.6 Risk Analysis

5.6.1 Risk Mitigation

5.7 Successes, Issues and Problems

5.7.1 Changes to the Backlog

6

Prototypes

This chapter is for recording each prototype developed. It is a historical record of what you accomplished in 464/465. This should be organized according to Sprints. It should have the basic description of the sprint deliverable and what was accomplished. Screen shots, photos, captures from video, etc should be used.

6.1 Sprint 1 Prototype

6.1.1 Deliverable

6.1.2 Backlog

6.1.3 Success/Fail

6.2 Sprint 2 Prototype

6.2.1 Deliverable

6.2.2 Backlog

6.2.3 Success/Fail

6.3 Sprint 3 Prototype

6.3.1 Deliverable

6.3.2 Backlog

6.3.3 Success/Fail

6.4 Sprint 4 Prototype

6.4.1 Deliverable

6.4.2 Backlog

6.4.3 Success/Fail

6.5 Sprint 5 Prototype

6.5.1 Deliverable

6.5.2 Backlog

6.5.3 Success/Fail

7

Release – Setup – Deployment

This section should contain any specific subsection regarding specifics in releasing, setup, and/or deployment of the system.

7.1 Deployment Information and Dependencies

Are there dependencies that are not embedded into the system install?

7.2 Setup Information

How is a setup/install built?

7.3 System Versioning Information

How is the system versioned?

8

User Documentation

8.1 User Guide

8.1.1 Registering a new User

When you First open the app you will be required to register a user or you can login via Facebook or twitter. To create a Crowd Control account fill out the required page and click register.

8.1.2 Login

8.1.2.a Crowd Control User

From the main page click the email button. This will allow you to login in via a Crowd Control account.

8.1.2.b Facebook Login

From the main page click the Facebook button. This will allow you to login via a Facebook account that will link to a Crowd Control user account.

8.1.2.c Twitter Login

From the main page click the twitter button. This will allow you to login via a twitter account that will link to a Crowd Control user account..

8.1.3 Creating a Group

From the main page:

1. Click "CREATE A NEW GROUP" Button at the bottom of the page.
2. Fill out Group Name and Group Description
3. Click "create group"

8.1.4 Joining a Group

8.1.4.a From Main List

1. Find the group from the list that you wish to join.
2. Click on the group to join.

8.1.4.b From Invite

1. Go to notifications.
2. Click on the invitation to join the group that invited you.

8.1.5 Leaving a Group

Click the leave group button at the bottom of the main group page.

8.1.6 Group Main Page

This page shows the main information of the group including:

- Group Name.
- Description of the Group.
- List of Group Members.
- Leave Button

8.1.7 Map Page

This page displays a map with pins of the other members in your group

8.1.7.a Sync Map

The sync map button will sync the map with the current locations of other members if your group. This feature is for if your map becomes desynced.

8.1.7.b Center Map

Clicking on the center map button will center the map on your location.

8.1.8 Messaging Page

This is the messaging feature. It allows you to send messages to the rest of the group. To bring up the key board click the text feild at the bottom of the page.

8.1.9 Options Menu

8.1.9.a Notifications

1. Opens the notifications page.

8.1.9.b Settings

1. Opens the settings page for CrowdControl.

8.1.10 Group Leader Options

8.1.10.a Invite

1. Opens the Invite page

8.1.10.b Notifications

1. Opens the notifications page.

8.1.10.c Change Group Name

1. Opens a window to change the name of the group

8.1.10.d Settings

1. Opens the settings page for CrowdControl.

8.1.11 Inviting Group Members

From the invite page:

1. Select users from the list of users, or you can search for there user name using the search bar.
2. When you have selected the users you would like to invite click on the next tab for the comfermation screen.
3. If these are the people you would like to invite click confirm and invites will be sent to the user.

8.1.12 Settings

8.1.12.a Changing Profile Name

1. To change your profile name click the text feild and change the name to the name you wish to be displayed to the rest of the group.
2. click change name.

8.1.12.b Search Distance

8.1.12.c Current Group

8.1.12.d Logout

8.1.12.e Finished

This will bring you back to the home page or the group main page if you are in a group.

8.2 Installation Guide

8.2.1 Android Install

8.2.1.a AppStore

Android Appstore version is not currently avalable.

8.2.1.b Android Studio

Set up your Device

1. Plug in your device to your development machine with a USB cable. If you're developing on Windows, you might need to install the appropriate USB driver for your device.
2. Enable USB debugging on your device. On Android 4.0 and newer, go to Settings > Developer options.

Run the app from Android Studio

1. Select one of your project's files and click Run from the toolbar.
2. In the Choose Device window that appears, select the Choose a running device radio button, select your device, and click OK .

Android Studio installs the app on your connected device and starts it.

Taken from http://developer.android.com.

8.2.2 iOS

8.2.2.a AppStore

Apple Appstore version is not currently avalable.

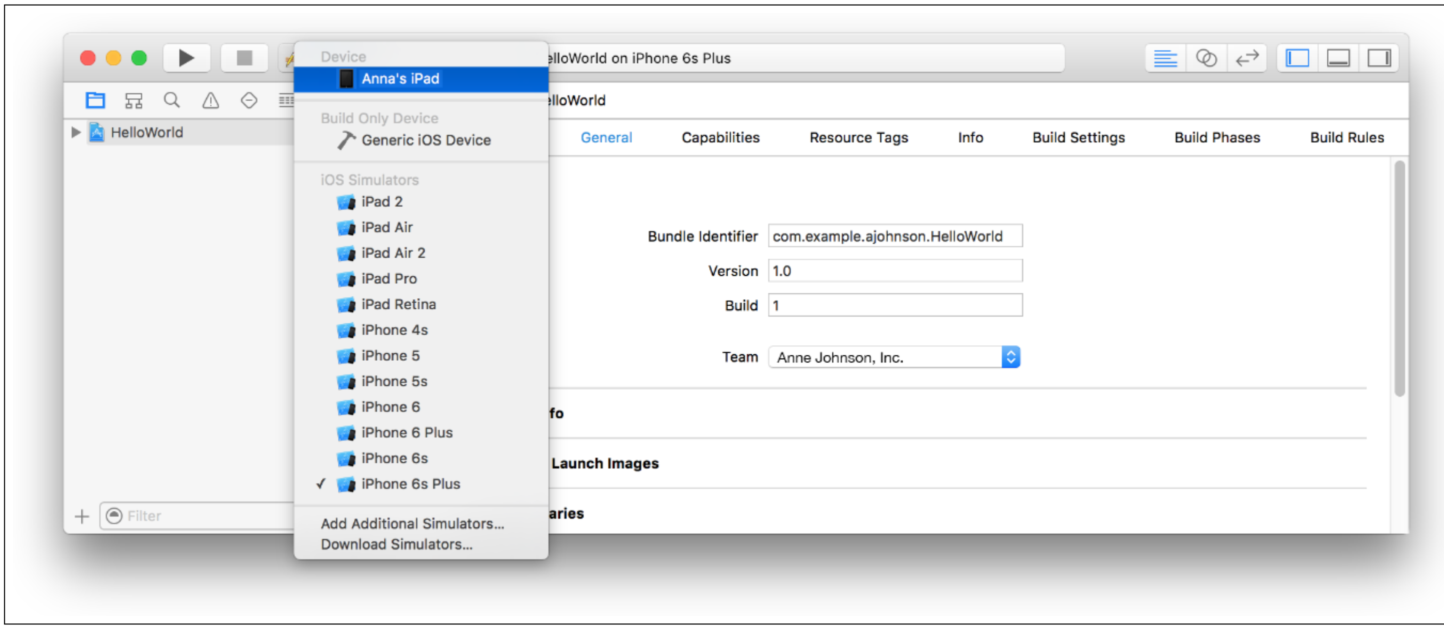
8.2.2.b XCode

Launching Your App on a Device

It takes just a few steps to launch your app on a device if you previously created your code signing identity and team provisioning profile, as described in Creating the Team Provisioning Profile. Otherwise, a series of dialogs and warnings may appear as Xcode resolves the code signing issues in the process of launching your app.

To launch an app on a device

1. Connect the device to your Mac.
2. In the project navigator, choose your device from the Scheme toolbar menu.
   Xcode assumes you intend to use the selected device for development and automatically registers it for you.



Bundle Identifier com.example.ajohnson.HelloWorld
Version 1.0
Build 1
Team Anne Johnson, Inc.

Figure 8.1: iOS Device selection

   If your device is disabled in the Scheme toolbar menu because it is ineligible, fix the issue before continuing. Under Ineligible Devices, hover the mouse over the device to read the reason why it's ineligible. For example, if the OS version is lower than the deployment target, upgrade the OS version on the device or choose the version you want to target from the Deployment Target pop-up menu (located in the Deployment Info section). Then select the device from the Scheme toolbar menu.

3. If a prompt appears asking whether codesign can sign the app using a key in your keychain, click Always Allow.

4. Click the Run button.

   Xcode installs the app on the device before launching the app.

Taken from https://developer.apple.com.

8.3 Programmer Manual

9

Class Index

9.1 Class List

9.1.1 Andriod

10

Class Documentation

10.1 ApplicationTest Class Reference

Public Member Functions

- ApplicationTest ()
- ∼ApplicationTest ()

10.1.1 Constructor & Destructor Documentation

10.1.1.a ApplicationTest::ApplicationTest ( )

Applicationtest Constructor

10.1.1.b ApplicationTest::∼ApplicationTest ( )

ApplicationTest Deconstructor

10.1.2 Method Summary

10.1.2.a Methods inherited from class android.test.Application TestCase ( )

getApplication, getSystemContext, testApplicationTestCaseSetUpProperly

10.1.2.b Methods inherited from class android.test.AndroidTestCase ( )

assertActivityRequiresPermission, assertReadingContentUriRequiresPermission, assertWritingContentUriRequires-Permission, getContext, getTestContext, setContext, setTestContext, testAndroidTestCaseSetupProperly

10.1.2.c Methods inherited from class java.lang.Object ( )

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

10.1.3 Generation

The documentation for this class was generated from the following file:

- ApplicationTest

10.2 BaseModel Class Reference

Interface BaseModel

public interface ***BaseModel***

The base model interface, providing access to core model functionality, including: - unique object identifier - the initial object creation - the last updated date - model saving and loading.

All Known Subinterfaces:
GroupModel, UserModel, UserProfileModel

All Known Implementing Classes:
ParseBaseModel, ParseGroupModel, ParseUserModel, ParseUserProfileModel

Member Functions

- getCreated ()
- getId ()
- getUpdated ()
- load ()
- loadInBackground ()
- save ()
- saveInBackground ()
- wasModified ()

10.2.1 Method Detail

10.2.1.a getCreated ( )

java.lang.Date getCreated()

Gets the creation date and time for the model. This value is automatically generated and assigned when the object is first added to storage.

Modifier and Type:
java.util.Date

Returns:
Date object representing the date and time when the model was first introduced into storage.

10.2.1.b getId ( )

java.lang.String getId()

Gets the unique object identifier for the model. This value is usually determined by the storage medium, such as the database where the model is stored.

Modifier and Type:
java.util.String

Returns:
String representation of the unique identifier that can be used to reference this object.

10.2.1.c getUpdated ( )

java.lang.Date getId()

Gets the date and time that the model was last updated in storage. This timestamp is automatically assigned by storage and represents when the storage-side model was last changed.

Modifier and Type:
java.util.Date

Returns:
Date object representing the date and time when the model was last updated in storage.

10.2.1.d load ( )

void getId()

Loads this object from storage. This is a blocking function, so care should be taken to not call this function on the main thread.

Modifier and Type:
void

Throws:
java.lang.Exception

10.2.1.e loadInBackground (BaseModel.LoadCallback *callback*)

void loadInBackground(BaseModel.LoadCallback callback)

Loads this object from storage asynchronously. Spawns a separate thread so that this function can be called from the main thread without blocking the UI. Upon completion, whether successful or unsuccessful, returns control to the main thread by calling the.

Modifier and Type:
void

Parameters:

- callback - The callback object to pass control to once the operation is completed. If no object is provided (or null is given), then nothing will happen after the object has been loaded.

10.2.1.f load ( )

void save()

Saves this object to storage. This is a blocking function, so care should be taken to not call this function on the main thread

Modifier and Type:
void

Throws:
java.lang.Exception

10.2.1.g loadInBackground (BaseModel.SaveCallback *callback*)

void loadInBackground(BaseModel.saveCallback callback)

Saves this object to storage asynchronously. Spawns a separate thread so that this function can be called from the main thread without blocking the UI. Upon completion, whether successful or unsuccessful, returns control to the main thread by calling the.

Modifier and Type:
void

Parameters:

- callback - The callback object to pass control to once the operation is complete. If no object is provided (or null is given), then nothing will happen after the object has been saved.

10.2.1.h wasModified ( )

void wasModified()

Gets a flag indicating that the model has new changes that haven't been saved in storage. This value will change whenever a property is set in the model but hasn't been saved.

Modifier and Type:
Boolean

Returns:
Boolean flag indicating whether or not the model contains unsaved changes.

10.2.2 Nested Class Summary

BaseModel.LoadCallback	40
BaseModel.SaveCallback	41

10.2.3 Generation

The documentation for this class was generated from the following file:

- BaseModel

10.3 BaseModel.LoadCallback Class Reference

Interface BaseMode.LoadCallback

public static interface ***BaseModel.LoadCallback***
The callback interface that should be used for asynchronous loading operations.

Enclosing interfaces:
BaseModel

Member Functions

- doneLoadingModel ()

10.3.1 Method Detail

10.3.1.a doneLoadingModel (BaseModel *object* java.lang.Exception *ex*)

void doneLoadingModel(BaseModel object, java.lang.Exception ex)

This method is called after completion of an asynchronous background load on a model. It accepts the loaded model and an exception object should something go wrong.

Modifier and Type:
void

Parameters:

- object - The object that attempted to be loaded. This parameter cannot be null and will be valid whether or not the operation was successful.
- ex - The exception that occurred, if any. This value will be null if the operation was successful and will be a valid exception object if an error occurred.

10.3.2 Generation

The documentation for this class was generated from the following file:

- BaseModel.LoadCallBack

10.4 BaseModel.SaveCallBack Class Reference

Interface BaseMode.SaveCallback

public static interface ***BaseModel.SaveCallback***
The callback interface that should be used for asynchronous loading operations.

Enclosing interfaces:
BaseModel

Member Functions

- doneLoadingModel ()

10.4.1 Method Summary

10.4.1.a doneSavingModel (BaseModel *object* java.lang.Exception *ex*)

void doneLoadingModel(BaseModel object, java.lang.Exception ex)

This method is called after completion of an asynchronous background save on a model. It accepts the saved model and an exception object should something go wrong.

Modifier and Type:
void

Parameters:

- object - The object that attempted to be saved. This parameter cannot be null and will be valid whether or not the operation was successful.
- ex - The exception that occurred, if any. This value will be null if the operation was successful and will be a valid exception object if an error occurred.

10.4.2 Generation

The documentation for this class was generated from the following file:

- BaseModel.SaveCallBack

10.5 BuildConfig Class Reference

Class BuildConfig

*public final class **BuildConfig***
extends:
java.lang.Object

Feild Summary

- APPLICATION_ID ()
- BUILD_TYPE ()
- DEBUG ()
- FLAVOR ()
- VERSION_CODE ()
- VERSION_NAME ()

10.5.1 Feild Detail

10.5.1.a APPLICATION_ID

public static final java.lang.String APPLICATION_ID

Modifier and Type:
public static final

10.5.1.b BUILD_TYPE

public static final java.lang.String BUILD_TYPE

Modifier and Type:
public static final

10.5.1.c DEBUG

public static final boolean DEBUG

Modifier and Type:
public static final

10.5.1.d FLAVOR

public static final java.lang.String FLAVOR

Modifier and Type:
public static final

10.5.1.e VERSION_CODE

public static final int VERSION_CODE

Modifier and Type:
public static final int

10.5.1.f VERSION_NAME

public static final java.lang.String VERSION_NAME

Modifier and Type:
public static final java.lang.String

10.5.2 Method Summary

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.5.3 Generation

The documentation for this class was generated from the following file:

- BuildConfig

10.6 CreateAccountActivity Class Reference

Class CreateAccountActivity

public class ***CreateAccountActivity*** extends android.support.v7.app.AppCompatActivity
implements android.view.View.OnClickListener

All Implemented Interfaces:
android.content.ComponentCallbacks, android.content.ComponentCallbacks2,
android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback,
android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator,
android.support.v4.app.TaskStackBuilder.SupportParentable,
android.support.v7.app.ActionBarDrawerToggle.DelegateProvider, android.support.v7.app.AppCompatCallback,
android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2,
android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener, android.view.Window.Callback,
android.view.Window.OnWindowDismissedCallback

Member Functions

- onClick ()
- onCreateOptionsMenu ()
- onCreateView ()
- onCreateView ()
- onOptionItemSelected ()

10.6.1 Method Summary

10.6.1.a onClick (android.view.View *view*)

public void onClick(android.view.View view)

Called when a view has been clicked.

Modifier and Type:
public void

Speciified by:
onClick in interface android.view.View.OnClickListener

Parameters':

- view - The view that was clicked.

10.6.1.b onCreateOptionsMenu (android.view.Menu *menu*)

public boolean onCreateOptionsMenu(android.view.Menu menu)
Initialize the contents of the Activity's standard options menu. You should place your menu items in to menu.

This is only called once, the first time the options menu is displayed. To update the menu every time it is displayed, see Activity.onPrepareOptionsMenu(android.view.Menu).

The default implementation populates the menu with standard system menu items. These are placed in the Menu.CATEGORY_SYSTEM group so that they will be correctly ordered with application-defined menu items. Deriving classes should always call through to the base implementation.

You can safely hold on to menu (and any items created from it), making modifications to it as desired, until the next time onCreateOptionsMenu() is called.

When you add items to the menu, you can implement the Activity's Activity.onOptionsItemSelected(android.view.MenuItem) method to handle them there.

Modifier and Type:
public boolean

Overrides:
onCreateOptionsMenu in class android.app.Activity
Parameters:

- menu - The options menu in which you place your items.

Returns:

- You must return true for the menu to be displayed; if you return false it will not be shown.

10.6.1.c onCreateView (android.view.View *parent* java.lang.String *name* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(android.view.View parent,
java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
public

Specified by:
onCreateView in interface android.view.LayoutInflater.Factory2

Overrides:
onCreateView in class android.app.Activity

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns:

- View Newly created view. Return null for the default behavior.

10.6.1.d onCreateView (java.lang.String *name* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
public

Specified by:
onCreateView in interface android.view.LayoutInflater.Factory

Overrides:
onCreateView in class android.app.Activity

Parameters:

- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns:

- View Newly created view. Return null for the default behavior.

10.6.1.e onOptionsItemSelect (android.view.MenuItem *item*)

public boolean onOptionsItemSelected(android.view.MenuItem item)

This hook is called whenever an item in your options menu is selected. The default implementation simply returns false to have the normal processing happen (calling the item's Runnable or sending a message to its Handler as appropriate). You can use this method for any items for which you would like to do processing without those other facilities.

Derived classes should call through to the base class for it to perform the default menu handling.

Modifier and Type:
public boolean

Overrides:
onOptionsItemSelected in class android.app.Activity

Parameters:

- item - The menu item that was selected.

Returns:

- boolean Return false to allow normal menu processing to proceed, true to consume it here.

10.6.2 Method Summary

10.6.2.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.6.2.b Methods inherited from class android.support.v4.app.FragmentActivity

ump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRequestPermissionsResult, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.6.2.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild, finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot, isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemSelected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState, onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOutside, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.6.2.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.6.2.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClass-

Loader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.6.2.f Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.6.2.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.6.3 Nested Class Summary

10.6.3.a Nested classes/interfaces inherited from class android.app.Activity

android.app.Activity.TranslucentConversionListener

10.6.3.b Nested classes/interfaces inherited from class android.content.Context

android.content.Context.BindServiceFlags, android.content.Context.CreatePackageOptions
, android.content.Context.ServiceName

10.6.4 Feild Summary

10.6.4.a Fields inherited from class android.app.Activity

```
DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL,
    DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, RESULT_CANCELED, RESULT_FIRST_USER,
    RESULT_OK
```

10.6.4.b Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY, CONTEXT_INCLUDE_CODE
    , CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR, DEVICE_IDLE_CONTROLLER,
     DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE, DROPBOX_SERVICE,
    ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE, INPUT_METHOD_SERVICE,
    INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE, LAUNCHER_APPS_SERVICE,
    LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE, MEDIA_PROJECTION_SERVICE,
    MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE, MODE_APPEND,
    MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE, MODE_WORLD_READABLE,
    MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE, NETWORK_SCORE_SERVICE,
    NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE, NOTIFICATION_SERVICE,
    NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE, PRINT_SERVICE, RADIO_SERVICE,
    RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE, SERIAL_SERVICE, SIP_SERVICE,
    STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE, TELEPHONY_SERVICE,
    TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE, TRUST_SERVICE,
    TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE, USAGE_STATS_SERVICE, USB_SERVICE,
     USER_SERVICE, VIBRATOR_SERVICE, VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE,
    WIFI_P2P_SERVICE, WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE,
    WIFI_SERVICE, WINDOW_SERVICE
```

10.6.5 Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.6.6 Generation

The documentation for this class was generated from the following file:

- CrowdControlApplication

10.7 CrowdControlApplication Class Reference

Class CrowdControlApplication

public class ***CrowdControlApplication*** extends android.app.Application

The official singleton object for the application.
All Implemented Interfaces:
ndroid.content.ComponentCallbacks, android.content.ComponentCallbacks2

Member Functions

- onCreate ()

10.7.1 Method Detail

10.7.1.a onCreate ( )

public void onCreate()

Called when the application is starting, before any activity, service, or receiver objects (excluding content providers) have been created. Implementations should be as quick as possible (for example using lazy initialization of state) since the time spent in this function directly impacts the performance of starting the first activity, service, or receiver in a process. If you override this method, be sure to call super.onCreate().

Modifier and Type:
public void

Overrides:
onCreate in class android.app.Application

10.7.2 Method Summary

10.7.2.a Methods inherited from class android.app.Application

onConfigurationChanged, onLowMemory, onTerminate, onTrimMemory, registerActivityLifecycleCallbacks, registerComponentCallbacks, registerOnProvideAssistDataListener, unregisterActivityLifecycleCallbacks, unregisterComponentCallbacks, unregisterOnProvideAssistDataListener

10.7.3 Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, canStartActivityForResult, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getResources, getSharedPreferences, getSharedPrefsFile, getSystemService, getSystemServiceName, getTheme, getThemeResId, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setTheme, setWallpaper, setWallpaper, startActivities, startActivities, startActivitiesAsUser, startActivity, startActivity, startActivityAsUser, startActivityAsUser, startActivityForResult, startInstrumentation, startIntentSender, startIntentSender, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.7.4 Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes

10.7.5 Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.7.6 Nested Class Summary

10.7.6.a Nested classes/interfaces inherited from class android.app.Activity

android.app.Activity.TranslucentConversionListener

10.7.6.b Nested classes/interfaces inherited from class android.content.Context

android.content.Context.BindServiceFlags, android.content.Context.CreatePackageOptions
, android.content.Context.ServiceName

Field Functions

- aGroup ()
- aUser ()

10.7.7 Field Detail

10.7.7.a aGroup ( )

aGroup
public static com.parse.ParseObject aGroup

10.7.7.b aUser ( )

aUser
public static com.parse.ParseUser aUser

10.7.8 Feild Summary

10.7.8.a Fields inherited from class android.app.Application

mLoadedApk

10.7.8.b Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
```

CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE, CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY, CONTEXT_INCLUDE_CODE , CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR, DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE, DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE, INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE, LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE, MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE, MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE, MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE, NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE, NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE, PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE, SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE, TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE, TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE, USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE, VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE, WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE, WINDOW_SERVICE

10.7.8.c Fields inherited from interface android.content.ComponentCallbacks2

TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE, TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE, TRIM_MEMORY_UI_HIDDEN

10.7.9 Generation

The documentation for this class was generated from the following file:

- CrowdControlApplication

10.8 EventFragment Class Reference

Class EventFragment

public interface ***EventFragment*** **extends:**
ndroid.support.v4.app.Fragment

A simple Fragment subclass. Handles showing the user potential Events they could attend

All Known Implementing Classes:
android.content.ComponentCallbacks, android.view.View.OnCreateContextMenuListener

Member Functions

- **newInstance ()**
- **onButtonPressed ()**
- **onCreate ()**
- **onCreateView ()**
- **onDetach ()**

10.8.1 Method Detail

10.8.1.a newInstance (java.lang.String *text*)

public void onCreate(android.os.Bundle savedInstanceState)

Use this factory method to create a new instance of this fragment using the provided parameters.

Modifier and Type:
public void

Parameters:

- text - Test text 1

Returns:
A new instance of fragment EventFragment.

10.8.1.b onButtonPressed (android.net.Uri *uri*)

10.8.1.c onCreate (android.os.Bundle *savedInstanceState*)

public void onCreate(android.os.Bundle savedInstanceState)

Called to do initial creation of a fragment. This is called after Fragment.onAttach(Activity) and before Fragment.onCreateView(LayoutInflater, ViewGroup, Bundle).

Note that this can be called while the fragment's activity is still in the process of being created. As such, you can not rely on things like the activity's content view hierarchy being initialized at this point. If you want to do work once the activity itself is created, see Fragment.onActivityCreated(Bundle).

Modifier and Type:
public void

Overrides:
onCreate in class android.support.v4.app.Fragment

Parameters:

- savedInstanceState - If the fragment is being re-created from a previous saved state, this is the state.

10.8.1.d onCreateView (android.view.LayoutInflater *inflater* android.view.ViewGroup *container* android.os.Bundle *avedInstanceState*)

public android.view.View onCreateView(android.view.LayoutInflater inflater, android.view.ViewGroup container, android.os.Bundle savedInstanceState)

alled to have the fragment instantiate its user interface view. This is optional, and non-graphical fragments can return null (which is the default implementation). This will be called between Fragment.onCreate(Bundle) and Fragment.onActivityCreated(Bundle).

If you return a View from here, you will later be called in Fragment.onDestroyView() when the view is being released.

Modifier and Type:
 public android.view.View

Overrides:
 onCreateView in class android.support.v4.app.Fragment

Parameters:

- inflater - The LayoutInflater object that can be used to inflate any views in the fragment,
- container - If non-null, this is the parent view that the fragment's UI should be attached to. The fragment should not add the view itself, but this can be used to generate the LayoutParams of the view.
- savedInstanceState - If non-null, this fragment is being re-constructed from a previous saved state as given here.

Overrides:
 Return the View for the fragment's UI, or null.

10.8.1.e onDetach (android.view.LayoutInflater *inflater* android.view.ViewGroup *container* android.os.Bundle *avedInstanceState*)

public void onDetach())

Called when the fragment is no longer attached to its activity. This is called after Fragment.onDestroy().

Overrides:
 onDetach in class android.support.v4.app.Fragment

10.8.2 Method Summary

10.8.2.a Methods inherited from class android.support.v4.app.Fragment

```
dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap,
    getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition
    , getFragmentManager, getHost, getId, getLayoutInflater, getLoaderManager,
    getParentFragment, getReenterTransition, getResources, getRetainInstance,
    getReturnTransition, getSharedElementEnterTransition, getSharedElementReturnTransition,
     getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText,
    getUserVisibleHint, getView, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded,
    isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isVisible,
    onActivityCreated, onActivityResult, onAttach, onAttach, onConfigurationChanged,
    onContextItemSelected, onCreateAnimation, onCreateContextMenu, onCreateOptionsMenu,
    onDestroy, onDestroyOptionsMenu, onDestroyView, onHiddenChanged, onInflate, onInflate,
    onLowMemory, onOptionsItemSelected, onOptionsMenuClosed, onPause, onPrepareOptionsMenu,
    onRequestPermissionsResult, onResume, onSaveInstanceState, onStart, onStop, onViewCreated
    , onViewStateRestored, registerForContextMenu, requestPermissions,
    setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap, setArguments,
    setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback,
    setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility,
    setReenterTransition, setRetainInstance, setReturnTransition,
    setSharedElementEnterTransition, setSharedElementReturnTransition, setTargetFragment,
```

setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivityForResult, toString, unregisterForContextMenu

10.8.2.b Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

10.8.3 Nested Class Summary

BaseModel.LoadCallback	40
BaseModel.SaveCallback	41

10.8.4 Generation

The documentation for this class was generated from the following file:

- BaseModel

10.9 GroupCreateActivityl Class Reference

Class GroupCreateActivity

public class GroupCreateActivity extends android.support.v7.app.AppCompatActivity
implements android.view.View.OnClickListener

The base model interface, providing access to core model functionality, including: - unique object identifier - the initial object creation - the last updated date - model saving and loading.

All Known Implementing Classes:
android.content.ComponentCallbacks, android.content.ComponentCallbacks2,
android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback,
android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator,
android.support.v4.app.TaskStackBuilder.SupportParentable, android.support.v7.app.ActionBarDrawerToggle.Delegatel
android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory,
android.view.LayoutInflater.Factory2, android.view.View.OnClickListener, android.view.View.OnCreateContextMenuList
android.view.Window.Callback, android.view.Window.OnWindowDismissedCallback

Member Functions

- onClick ()
- onCreateView ()
- onCreateView ()

10.9.1 Method Detail

10.9.1.a onClick (android.view.View *view*)

public void onClick(android.view.View view)

Called when a view has been clicked.

Modifier and Type:
public android.view.View

Parameters:

- view - The view that was clicked.

10.9.1.b onCreateView (android.view.View *parent* java.lang.String *name* android.content.Context *context* ndroid.util.AttributeSet *attrs*)

public android.view.View onCreateView(android.view.View parent,
java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
public android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory2

Overrides: onCreateView in class android.app.Activity

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns:

- View Newly created view. Return null for the default behavior.

10.9.1.c onCreateView (java.lang.String *name* android.content.Context *context* ndroid.util.AttributeSet *attrs*)

public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

tandard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.St
This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
public android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Overrides: onCreateView in class android.app.Activity

Parameters:

- name - Tag name to be inflated.

- **context - The context the view is being created in.**
- **attrs - Inflation attributes as specified in XML file.**

Returns:

- **View Newly created view. Return null for the default behavior.**

10.9.2 Method Summary

10.9.2.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.9.2.b Methods inherited from class android.support.v4.app.FragmentActivity

dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRequestPermissionsResult, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.9.2.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild, finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot, isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemSelected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreateOptionsMenu, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsItemSelected, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState,

onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOutside, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.9.2.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.9.2.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.9.2.f Methods inherited from class android.content.Context

etColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.9.2.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.9.3 Feild Summary

10.9.3.a Fields inherited from class android.app.Activity

```
DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL,
    DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, RESULT_CANCELED, RESULT_FIRST_USER,
    RESULT_OK
```

10.9.3.b Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.9.3.c Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.9.4 Generation

The documentation for this class was generated from the following file:

- GroupCreateActivity

10.10 GroupInfoFragment Class Reference

Class GroupInfoFragment

public class GroupInfoFragment extends android.support.v4.app.Fragment

A simple Fragment subclass. Activities that contain this fragment must implement the to handle interaction events. Use the newInstance(java.lang.String) factory method to create an instance of this fragment. Will display information of the current group and allow the user to leave the group

All Known Implementing Classes:
android.content.ComponentCallbacks, android.content.ComponentCallbacks2, android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback, android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator, android.support.v4.app.TaskStackBuilder.SupportParentable, android.support.v7.app.ActionBarDrawerToggle.DelegateProvider, android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2, android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener, android.view.Window.Callback, android.view.Window.OnWindowDismissedCallback

Member Functions

- newInstance ()
- onCreate ()
- onCreateView ()

10.10.1 Method Detail

10.10.1.a newInstance (java.lang.String *text*)

newInstance(java.lang.String text))

Use this factory method to create a new instance of this fragment using the provided parameters.

Modifier and Type:
static GroupInfoFragment

Parameters:

- text - Test text 1

Returns: A new instance of fragment GroupInfoFragment.

10.10.1.b onCreate (java.lang.String *text*)

public void onCreate(android.os.Bundle savedInstanceState)

Called to do initial creation of a fragment. This is called after Fragment.onAttach(Activity) and before Fragment.onCreateView(LayoutInflater, ViewGroup, Bundle).

Note that this can be called while the fragment's activity is still in the process of being created. As such, you can not rely on things like the activity's content view hierarchy being initialized at this point. If you want to do work once the activity itself is created, see Fragment.onActivityCreated(Bundle).

Modifier and Type:
void

Overrides: onCreate in class android.support.v4.app.Fragment.

Parameters:

- savedInstanceState - If the fragment is being re-created from a previous saved state, this is the state.

10.10.1.c onCreateView (android.view.LayoutInflater *inflater* android.view.ViewGroup *container* android.os.Bundle *savedInstanceState*)

public android.view.View onCreateView(android.view.LayoutInflater inflater,
android.view.ViewGroup container,
android.os.Bundle savedInstanceState)

Called to have the fragment instantiate its user interface view. This is optional, and non-graphical fragments can return null (which is the default implementation). This will be called between Fragment.onCreate(Bundle) and Fragment.onActivityCreated(Bundle).

If you return a View from here, you will later be called in Fragment.onDestroyView() when the view is being released.

Modifier and Type:
void

Overrides: onCreate in class android.support.v4.app.Fragment.

Parameters:

- inflater - The LayoutInflater object that can be used to inflate any views in the fragment,

- container - If non-null, this is the parent view that the fragment's UI should be attached to. The fragment should not add the view itself, but this can be used to generate the LayoutParams of the view.

- savedInstanceState - If non-null, this fragment is being re-constructed from a previous saved state as given here.

Returns: Return the View for the fragment's UI, or null.

10.10.2 Method Summary

10.10.2.a Methods inherited from class android.support.v4.app.Fragment

dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap, getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition, getFragmentManager, getHost, getId, getLayoutInflater, getLoaderManager, getParentFragment, getReenterTransition, getResources, getRetainInstance, getReturnTransition, getSharedElementEnterTransition, getSharedElementReturnTransition, getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText, getUserVisibleHint, getView, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded, isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isVisible, onActivityCreated, onActivityResult, onAttach, onAttach, onConfigurationChanged, onContextItemSelected, onCreateAnimation, onCreateContextMenu, onCreateOptionsMenu, onDestroy, onDestroyOptionsMenu, onDestroyView, onDetach, onHiddenChanged, onInflate, onInflate, onLowMemory, onOptionsItemSelected, onOptionsMenuClosed, onPause, onPrepareOptionsMenu, onRequestPermissionsResult, onResume, onSaveInstanceState, onStart, onStop, onViewCreated, onViewStateRestored, registerForContextMenu, requestPermissions, setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap, setArguments, setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback, setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility, setReenterTransition, setRetainInstance, setReturnTransition, setSharedElementEnterTransition, setSharedElementReturnTransition, setTargetFragment, setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivityForResult, toString, unregisterForContextMenu

10.10.2.b Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

10.10.3 Nested Class Summary

10.10.3.a Nested classes/interfaces inherited from class android.support.v4.app.Fragment

- android.support.v4.app.Fragment.InstantiationException
- android.support.v4.app.Fragment.SavedState

10.10.4 Generation

The documentation for this class was generated from the following file:

- GroupCreateActivity

10.11 GroupJoinActivity Class Reference

Class GroupJoinActivity

public class GroupJoinActivity
extendsandroid.support.v7.app.AppCompatActivity
implements android.view.View.OnClickListener

A simple Fragment subclass. Activities that contain this fragment must implement the to handle interaction events. Use the newInstance(java.lang.String) factory method to create an instance of this fragment. Will display information of the current group and allow the user to leave the group

All Known Implementing Classes:
android.content.ComponentCallbacks, android.content.ComponentCallbacks2,

android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback, android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator, android.support.v4.app.TaskStackBuilder.SupportParentable, android.support.v7.app.ActionBarDrawerToggle.DelegateProvider, android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2, android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener, android.view.Window.Callback, android.view.Window.OnWindowDismissedCallback

Member Functions

- onClick ()
- onCreateOptionsMenu ()
- onCreateView ()
- onCreateView ()
- onOptionsItemSelected ()

10.11.1 Method Detail

10.11.1.a onClick (java.lang.String *text*)

public void onClick(android.view.View view)

Called when a view has been clicked.

Modifier and Type:
void

Specified by: onClick in interface android.view.View.OnClickListener

Parameters:

- view - The view that was clicked.

10.11.2 Method Detail

10.11.2.a onCreateOptionsMenu (android.view.Menu *menu*)

public boolean onCreateOptionsMenu(android.view.Menu menu)

nitialize the contents of the Activity's standard options menu. You should place your menu items in to menu.

This is only called once, the first time the options menu is displayed. To update the menu every time it is displayed, see Activity.onPrepareOptionsMenu(android.view.Menu).

The default implementation populates the menu with standard system menu items. These are placed in the Menu.CATEGORY_SYSTEM group so that they will be correctly ordered with application-defined menu items. Deriving classes should always call through to the base implementation.

You can safely hold on to menu (and any items created from it), making modifications to it as desired, until the next time onCreateOptionsMenu() is called.

When you add items to the menu, you can implement the Activity's Activity.onOptionsItemSelected(android.view.Mer
method to handle them there.

Modifier and Type:
boolean

Overrides: onCreateOptionsMenu in class android.app.Activity

Parameters:

- menu - The options menu in which you place your items.

Returns: You must return true for the menu to be displayed; if you return false it will not be shown.

10.11.2.b onCreateView (java.lang.String *name* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(android.view.View parent,
java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Overrides: onCreateView in class android.app.Activity

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.11.2.c onCreateView (java.lang.String *name* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.St
This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Overrides: onCreateView in class android.app.Activity

Parameters:

- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.11.2.d onOptionsItemSelected (android.view.MenuItem *item*)

public boolean onOptionsItemSelected(android.view.MenuItem item)

This hook is called whenever an item in your options menu is selected. The default implementation simply returns false to have the normal processing happen (calling the item's Runnable or sending a message to its Handler as appropriate). You can use this method for any items for which you would like to do processing without those other facilities.

Derived classes should call through to the base class for it to perform the default menu handling.

Modifier and Type:
boolean

Overrides: onOptionsItemSelected in class android.app.Activity

Parameters:

- item - The menu item that was selected.

Returns: boolean Return false to allow normal menu processing to proceed, true to consume it here.

10.11.3 Method Summary

10.11.3.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.11.3.b Methods inherited from class android.support.v4.app.FragmentActivity

dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRequestPermissionsResult, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.11.3.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild, finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot, isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemSelected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState, onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOutside, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.11.3.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.11.3.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.11.3.f Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.11.3.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.11.4 Nested Class Summary

10.11.4.a Nested classes/interfaces inherited from class android.app.Activity

- android.app.Activity.TranslucentConversionListener

10.11.4.b Nested classes/interfaces inherited from class android.content.Context

- android.content.Context.BindServiceFlags
- android.content.Context.CreatePackageOptions
- android.content.Context.ServiceName

10.11.5 Feild Summary

10.11.6 Fields inherited from class android.app.Activity

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.11.7 Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.11.8 Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.11.9 Generation

The documentation for this class was generated from the following file:

- GroupCreateActivity

10.12 GroupModel Interface Reference

Interface GroupModel

Interface GroupModel
extends BaseModel

The interface for group models.

All Known Implementing Classes:

Member Functions

- getGeneralLocation ()
- getGroupDescription ()
- getGroupMembers ()

10.12.1 Method Detail

10.12.1.a getGeneralLocation ( )

com.parse.ParseGeoPoint getGeneralLocation()

Gets the general location of the group.

Modifier and Type:
com.parse.ParseGeoPoint

Returns: The location of the group in the form of a ParseGeoPoint object.

10.12.1.b getGroupDescription ( )

java.lang.String getGroupDescription()

Gets the description of the current group.

Modifier and Type:
java.lang.String

Returns: The description string attached to the group.

10.12.1.c getGroupMembers ( )

com.parse.ParseUser getGroupMembers()

Gets the list of users associated with the current group.

Modifier and Type:
com.parse.ParseUser

Returns: The list of users as ParseUserModel objects that belong to the group.

10.12.2 Method Summary

10.12.2.a Methods inherited from interface com.bowtaps.crowdcontrol.model.BaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.12.3 Nested Class Summary

10.12.3.a Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Base

- BaseModel.LoadCallback ()
- BaseModel.SaveCallback ()

10.12.4 Generation

The documentation for this class was generated from the following file:

- GroupModel

10.13 GroupNavigationActivity Class Reference

Class GroupNavigationActivity

class GroupNavigationActivity
extendsandroid.support.v7.app.AppCompatActivity

This Activity will manage all the tabs related to the current group It uses a tab based system to switch between fragments.

All Known Implementing Classes:
android.content.ComponentCallbacks, android.content.ComponentCallbacks2, android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback,

android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator,
android.support.v4.app.TaskStackBuilder.SupportParentable,
android.support.v7.app.ActionBarDrawerToggle.DelegateProvider,
android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback,
android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2,
android.view.View.OnCreateContextMenuListener, android.view.Window.Callback,
android.view.Window.OnWindowDismissedCallback

Member Functions

- onCreateView ()
- onCreateView ()

10.13.1 Method Detail

10.13.1.a onCreateView ( )

*public android.view.View onCreateView(android.view.View parent,
java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)*

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory2

Overrides: onCreateView in class android.app.Activity

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.13.1.b onCreateView ( )

*public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)*

Standard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.St
This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Overrides: onCreateView in class android.app.Activity

Parameters:

- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.13.2 Method Summary

10.13.2.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.13.2.b Methods inherited from class android.support.v4.app.FragmentActivity

dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRequestPermissionsResult, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.13.2.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild, finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot,

isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemSelected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreateOptionsMenu, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsItemSelected, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState, onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOutside, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.13.2.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.13.2.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, send-

Broadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.13.2.f Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.13.2.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.13.3 Nested Class Summary

10.13.3.a Nested classes/interfaces inherited from class android.app.Activity

- **android.app.Activity.TranslucentConversionListener**

10.13.3.b Nested classes/interfaces inherited from class android.content.Context

- **android.content.Context.BindServiceFlags**
- **android.content.Context.CreatePackageOptions**
- **android.content.Context.ServiceName**

10.13.4 Feild Summary

10.13.5 Fields inherited from class android.app.Activity

```
DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL,
    DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, RESULT_CANCELED, RESULT_FIRST_USER,
    RESULT_OK
```

10.13.6 Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.13.7 Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.13.8 Generation

The documentation for this class was generated from the following file:

- GroupNavigationActivity

10.14 LoginActivity Class Reference

Class LoginActivity

class LoginActivity
 extends android.support.v7.app.AppCompatActivity
 implements android.app.LoaderManager.LoaderCallbacks<android.database.Cursor>

All Known Implementing Classes:
 android.app.LoaderManager.LoaderCallbacks<android.database.Cursor>,
android.content.ComponentCallbacks, android.content.ComponentCallbacks2,
android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback,
android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator,
android.support.v4.app.TaskStackBuilder.SupportParentable,
android.support.v7.app.ActionBarDrawerToggle.DelegateProvider,
android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback,
android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2,
android.view.View.OnCreateContextMenuListener, android.view.Window.Callback,
android.view.Window.OnWindowDismissedCallback

Member Functions

- onCreateLoader ()
- onCreateView ()

- onCreateView ()
- onLoaderReset ()
- onLoadFinished ()
- onRequestPerissionsResult ()

10.14.1 Method Detail

10.14.1.a onCreateLoader (int *i* android.os.Bundle *bundle*)

public android.content.Loader<android.database.Cursor> onCreateLoader(int i, android.os.Bundle bundle)

onCreateLoader in interface android.app.LoaderManager.LoaderCallbacks<android.database.Cursor>

Modifier and Type:
android.content.Loader<android.database.Cursor>

Specified by: onCreateLoader in interface android.app.LoaderManager.LoaderCallbacks<android.database.Cur

Parameters:

- i - The ID whose loader is to be created.
- bundle - Any arguments supplied by the caller.

Returns: Return a new Loader instance that is ready to start loading.

10.14.1.b onCreateView (android.view.View *parent* java.lang.String *name* android.content.Context *context* ndroid.util.AttributeSet *attrs*)

public android.view.View onCreateView(android.view.View parent, java.lang.String name, android.content.Context context, android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
android.content.Loader<android.database.Cursor>

Specified by: onCreateView in interface android.view.LayoutInflater.Factory2

Overrides: onCreateView in class android.app.Activity

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.14.1.c onCreateView (java.lang.String *name* android.content.Context *context* ndroid.util.AttributeSet *attrs*)

public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.St
This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
android.content.Loader<android.database.Cursor>

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Overrides: onCreateView in class android.app.Activity

Parameters:

- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.14.1.d onLoaderReset (android.content.Loader<android.database.Cursor> *cursorLoader*)

public void onLoaderReset(android.content.Loader<android.database.Cursor> cursorLoader)

Called when a previously created loader is being reset, and thus making its data unavailable. The application should at this point remove any references it has to the Loader's data.

Modifier and Type:
void>

Specified by: onLoaderReset in interface android.app.LoaderManager.LoaderCallbacks<android.database.Curs

Parameters:

- cursorLoader - The Loader that is being reset.

10.14.1.e onLoadFinished (android.content.Loader<android.database.Cursor> *cursorLoader* android.database.Cursor *cursor*)

public void onLoadFinished(android.content.Loader<android.database.Cursor> cursorLoader,
android.database.Cursor cursor)

Called when a previously created loader has finished its load. Note that normally an application is not allowed to commit fragment transactions while in this call, since it can happen after an activity's state is saved. See FragmentManager.openTransaction() for further discussion on this.

This function is guaranteed to be called prior to the release of the last data that was supplied for this Loader. At this point you should remove all use of the old data (since it will be released soon), but should not do your own release of the data since its Loader owns it and will take care of that. The Loader will take care of management of its data so you don't have to. In particular:

- The Loader will monitor for changes to the data, and report them to you through new calls here. You should not monitor the data yourself. For example, if the data is a Cursor and you place it in a CursorAdapter, use the CursorAdapter.CursorAdapter(android.content.Context, android.database.Cursor, int) constructor without passing in either CursorAdapter.FLAG_AUTO_REQUERY or CursorAdapter.FLAG_REGISTER_CONTENT_OBSERVER (that is, use 0 for the flags argument). This prevents the CursorAdapter from doing its own observing of the Cursor, which is not needed since when a change happens you will get a new Cursor throw another call here.

- The Loader will release the data once it knows the application is no longer using it. For example, if the data is a Cursor from a CursorLoader, you should not call close() on it yourself. If the Cursor is being placed in a CursorAdapter, you should use the CursorAdapter.swapCursor(android.database.Cursor) method so that the old Cursor is not closed.

Modifier and Type:
void

Specified by: onLoadFinished in interface android.app.LoaderManager.LoaderCallbacks<android.database.Curs

Parameters:

- cursorLoader - The Loader that has finished.

- cursor - The data generated by the Loader.

10.14.1.f onRequestPermissionsResult (android.content.Loader<android.database.Cursor> *cursorLoader* android.database.Cursor *cursor*)

public void onRequestPermissionsResult(int requestCode,
java.lang.String[] permissions,
int[] grantResults)

Callback received when a permissions request has been completed.

Modifier and Type:
void

Specified by: onRequestPermissionsResult in interface android.support.v4.app.ActivityCompat.OnRequestPerm

Overrides: onRequestPermissionsResult in class android.support.v4.app.FragmentActivity

Parameters:

- requestCode - The request code passed in Activity.requestPermissions(String[], int).

- permissions - The requested permissions. Never null.

- grantResults - The grant results for the corresponding permissions which is either PackageManager.PERMISSION_GRANTED or PackageManager.PERMISSION_DENIED. Never null.

10.14.2 Method Summary

10.14.2.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.14.2.b Methods inherited from class android.support.v4.app.FragmentActivity

dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.14.2.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild, finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot, isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemSelected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreateOptionsMenu, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsItemSelected, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState, onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOut-

side, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.14.2.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.14.2.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.14.2.f Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.14.2.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.14.3 Nested Class Summary

10.14.3.a Nested Classes

- UserLoginTask ()

10.14.3.b Nested classes/interfaces inherited from class android.content.Context

- android.app.Activity.TranslucentConversionListener

10.14.3.c Nested classes/interfaces inherited from class android.content.Context

- android.content.Context.BindServiceFlags
- android.content.Context.CreatePackageOptions
- android.content.Context.ServiceName

10.14.4 Feild Summary

10.14.5 Fields inherited from class android.app.Activity

```
DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL,
    DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, RESULT_CANCELED, RESULT_FIRST_USER,
    RESULT_OK
```

10.14.6 Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.14.7 Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.14.8 Generation

The documentation for this class was generated from the following file:

- LoginActivity

10.15 LoginController Class Reference

Class LoginController

class LoginActivity
extends java.lang.Object

10.15.1 Constructor Summary

10.15.1.a Constructor and Description

LoginController(android.content.Context app_context)

10.15.2 Method Summary

10.15.2.a Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.15.3 Constructor Detail

10.15.3.a LoginController

public LoginController(android.content.Context app_context)

10.15.4 Generation

The documentation for this class was generated from the following file:

- LoginController

10.16 MapFragment Class Reference

Class MapFragment

class MapFragment
extendsandroid.support.v4.app.Fragment

A simple Fragment subclass. Will Display a Google Map and place group members on it

Member Functions

- newInstance ()
- onCreate ()
- onCreateView ()

10.16.1 Method Detail

10.16.1.a newInstance (java.lang.String *text*)

public static MapFragment newInstance(java.lang.String text)

Use this factory method to create a new instance of this fragment using the provided parameters.

Modifier and Type:
static MapFragment

Parameters:

- text - Test text 1

Returns: A new instance of fragment MapFragment.

10.16.1.b onCreate (android.os.Bundle *savedInstanceState*)

public void onCreate(android.os.Bundle savedInstanceState)

Called to do initial creation of a fragment. This is called after Fragment.onAttach(Activity) and before Fragment.onCreateView(LayoutInflater, ViewGroup, Bundle).

Note that this can be called while the fragment's activity is still in the process of being created. As such, you can not rely on things like the activity's content view hierarchy being initialized at this point. If you want to do work once the activity itself is created, see Fragment.onActivityCreated(Bundle).

Modifier and Type:
void

Overrides: onCreate in class android.support.v4.app.Fragment

Parameters:

- savedInstanceState - If the fragment is being re-created from a previous saved state, this is the state.

10.16.1.c onCreateView (android.view.LayoutInflater *inflater* android.view.ViewGroup *container* android.os.Bundle *savedInstanceState*)

public void onCreate(android.os.Bundle savedInstanceState)

Called to have the fragment instantiate its user interface view. This is optional, and non-graphical fragments can return null (which is the default implementation). This will be called between Fragment.onCreate(Bundle) and Fragment.onActivityCreated(Bundle).

If you return a View from here, you will later be called in Fragment.onDestroyView() when the view is being released.

Modifier and Type:
android.view.View

Overrides: onCreate in class android.support.v4.app.Fragment

Parameters:

- inflater - The LayoutInflater object that can be used to inflate any views in the fragment
- container - If non-null, this is the parent view that the fragment's UI should be attached to. The fragment should not add the view itself, but this can be used to generate the LayoutParams of the view.
- savedInstanceState - If non-null, this fragment is being re-constructed from a previous saved state as given here.

Returns: Return the View for the fragment's UI, or null.

10.16.2 Method Summary

10.16.2.a Methods inherited from class android.support.v4.app.Fragment

dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap, getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition, getFragmentManager, getHost, getId, getLayoutInflater, getLoaderManager, getParentFragment, getReenterTransition, getResources, getRetainInstance, getReturnTransition, getSharedElementEnterTransition, getSharedElementReturnTransition, getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText, getUserVisibleHint, getView, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded, isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isVisible, onActivityCreated, onActivityResult, onAttach, onAttach, onConfigurationChanged, onContextItemSelected, onCreateAnimation, onCreateContextMenu, onCreateOptionsMenu, onDestroy, onDestroyOptionsMenu, onDestroyView, onDetach, onHiddenChanged, onInflate, onInflate, onLowMemory, onOptionsItemSelected, onOptionsMenuClosed, onPause, onPrepareOptionsMenu, onRequestPermissionsResult, onResume, onSaveInstanceState, onStart, onStop, onViewCreated, onViewStateRestored, registerForContextMenu, requestPermissions, setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap, setArguments, setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback, setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility, setReenterTransition, setRetainInstance, setReturnTransition, setSharedElementEnterTransition, setSharedElementReturnTransition, setTargetFragment, setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivityForResult, toString, unregisterForContextMenu

10.16.2.b Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

10.16.3 Nested Class Summary

10.16.3.a ested classes/interfaces inherited from class android.support.v4.app.Fragment

- android.support.v4.app.Fragment.InstantiationException
- android.support.v4.app.Fragment.SavedState

10.16.4 Generation

The documentation for this class was generated from the following file:

- MapFragment

10.17 MessagingFragment Class Reference

Class MessagingFragment

class MessagingFragment
extendsandroid.support.v4.app.Fragment

A simple Fragment subclass. Use the newInstance(java.lang.String) factory method to create an instance of this fragment. This Fragment will handle all messaging between user and group

All Known Implementing Classes:
android.content.ComponentCallbacks, android.view.View.OnCreateContextMenuListener

Member Functions

- newInstance ()
- onButtonPress ()
- onCreate ()
- onCreateView ()
- onDetach ()

10.17.1 Method Detail

10.17.1.a newInstance (java.lang.String *text*)

public static MessagingFragment newInstance(java.lang.String text)

Use this factory method to create a new instance of this fragment using the provided parameters.

Modifier and Type:
static MessagingFragment

Parameters:

- text - Test text 1

Returns: A new instance of fragment MessagingFragment.

10.17.1.b onButtonPress (android.net.Uri *uri*)

public void onButtonPressed(android.net.Uri uri)

Modifier and Type:
void

Parameters:

- uri

10.17.1.c onCreate (android.os.Bundle *savedInstanceState*)

public void onCreate(android.os.Bundle savedInstanceState)

Called to do initial creation of a fragment. This is called after Fragment.onAttach(Activity) and before Fragment.onCreateView(LayoutInflater, ViewGroup, Bundle).

Note that this can be called while the fragment's activity is still in the process of being created. As such, you can not rely on things like the activity's content view hierarchy being initialized at this point. If you want to do work once the activity itself is created, see Fragment.onActivityCreated(Bundle).

Modifier and Type:
void

Overrides: onCreate in class android.support.v4.app.Fragment

Parameters:

- savedInstanceState - If the fragment is being re-created from a previous saved state, this is the state.

10.17.1.d onCreateView (android.view.LayoutInflater *inflater* android.view.ViewGroup *container* android.os.Bundle *savedInstanceState*)

public android.view.View onCreateView(android.view.LayoutInflater inflater,
android.view.ViewGroup container,
android.os.Bundle savedInstanceState)

Called to have the fragment instantiate its user interface view. This is optional, and non-graphical fragments can return null (which is the default implementation). This will be called between Fragment.onCreate(Bundle) and Fragment.onActivityCreated(Bundle).

If you return a View from here, you will later be called in Fragment.onDestroyView() when the view is being released.

Modifier and Type:
void

Overrides: onCreateView in class android.support.v4.app.Fragment

Parameters:

- inflater - The LayoutInflater object that can be used to inflate any views in the fragment

- container - If non-null, this is the parent view that the fragment's UI should be attached to. The fragment should not add the view itself, but this can be used to generate the LayoutParams of the view.
- savedInstanceState - If non-null, this fragment is being re-constructed from a previous saved state as given here.

Returns: Return the View for the fragment's UI, or null.

10.17.1.e onDetach ( )

public void onDetach()

Called when the fragment is no longer attached to its activity. This is called after Fragment.onDestroy().

Modifier and Type:
void

Overrides: onCreateView in class android.support.v4.app.Fragment

10.17.2 Method Summary

10.17.2.a Methods inherited from class android.support.v4.app.Fragment

dump, equals, getActivity, getAllowEnterTransitionOverlap, getAllowReturnTransitionOverlap, getArguments, getChildFragmentManager, getContext, getEnterTransition, getExitTransition, getFragmentManager, getHost, getId, getLayoutInflater, getLoaderManager, getParentFragment, getReenterTransition, getResources, getRetainInstance, getReturnTransition, getSharedElementEnterTransition, getSharedElementReturnTransition, getString, getString, getTag, getTargetFragment, getTargetRequestCode, getText, getUserVisibleHint, getView, hashCode, hasOptionsMenu, instantiate, instantiate, isAdded, isDetached, isHidden, isInLayout, isMenuVisible, isRemoving, isResumed, isVisible, onActivityCreated, onActivityResult, onAttach, onAttach, onConfigurationChanged, onContextItemSelected, onCreateAnimation, onCreateContextMenu, onCreateOptionsMenu, onDestroy, onDestroyOptionsMenu, onDestroyView, onHiddenChanged, onInflate, onInflate, onLowMemory, onOptionsItemSelected, onOptionsMenuClosed, onPause, onPrepareOptionsMenu, onRequestPermissionsResult, onResume, onSaveInstanceState, onStart, onStop, onViewCreated, onViewStateRestored, registerForContextMenu, requestPermissions, setAllowEnterTransitionOverlap, setAllowReturnTransitionOverlap, setArguments, setEnterSharedElementCallback, setEnterTransition, setExitSharedElementCallback, setExitTransition, setHasOptionsMenu, setInitialSavedState, setMenuVisibility, setReenterTransition, setRetainInstance, setReturnTransition, setSharedElementEnterTransition, setSharedElementReturnTransition, setTargetFragment, setUserVisibleHint, shouldShowRequestPermissionRationale, startActivity, startActivityForResult, toString, unregisterForContextMenu

10.17.2.b Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

10.17.3 Nested Class Summary

10.17.3.a Nested classes/interfaces inherited from class android.support.v4.app.Fragment

- android.support.v4.app.Fragment.InstantiationException
- android.support.v4.app.Fragment.SavedState

10.17.4 Generation

The documentation for this class was generated from the following file:

- MessagingFragment

10.18 ParseBaseModel Class Reference

Class ParseBaseModel

class ParseBaseModel
extendsjava.lang.Object
implementsBaseModel ()

The base Parse implementation of models. Fully implements the

Member Functions

- getCreated ()
- getId ()
- getUpdated ()
- load ()
- loadInBackground ()
- save ()
- saveInBackground ()
- wasModified ()

10.18.1 Method Detail

10.18.1.a getCreated ( )

public java.util.Date getCreated()

Gets the object's creation timestamp.

Modifier and Type:
java.util.Date

Specified by:

Returns: The object's last update timestamp.

10.18.1.b getId ( )

public java.lang.String getId()

Gets this object's unique identifier.

Modifier and Type:
java.lang.String

Specified by: getId in BaseModel

Returns: The object's unique identifier.

10.18.1.c getUpdated( )

public java.util.Date getUpdated()

Gets the object's last updated timestamp.

Modifier and Type:
java.util.Date

Specified by: wasModified in BaseModel

Returns: Whether or not the object has unsaved changes.

10.18.1.d load( )

public void load()
throws com.parse.ParseException
Attempts to load the model from parse synchronously. This is a blocking function and thus should never be used on the main thread.

Modifier and Type:
void

Specified by: load in BaseModel

Throws: java.lang.Exception - If an exception is thrown by Parse, it will be passed on to this function's caller.
com.parse.ParseException.

10.18.1.e loadInBackground(BaseModel.LoadCallback *callback*)

public void loadInBackground(BaseModel.LoadCallback callback)

Attempts to save the model to Parse asynchronously and passes control back to the main thread by using the object passed as a parameter to this function.

Modifier and Type:
void

Specified by: loadInBackground in BaseModel

Parameters:

- callback - The callback object to pass control to once the operation is completed. If no object is provided (or null is given), then no callback will be executed.

10.18.1.f save( )

public void save()

throws com.parse.ParseException
Attempts to save the model to Parse synchronously. This is a blocking function and thus should never be used on the main thread.

Modifier and Type:
void

Specified by: save in BaseModel

Throws: java.lang.Exception - If an exception is thrown by Parse, it will be passed on to this function's caller. com.parse.ParseException

10.18.1.g saveInBackground(BaseModel.SaveCallback *callback*)

public void saveInBackground(BaseModel.SaveCallback callback)

Attempts to save the model to Parse asynchronously and passes control back to the main thread by using the object passed as a parameter to this function.

Modifier and Type:
void

Specified by: saveInBackground in BaseModel

Parameters:

- callback - The callback object to pass control to once the operation is completed. If no object is provided (or null is given), then no callback will be executed.

10.18.1.h wasModified( )

public java.lang.Boolean wasModified()

Gets whether or not the object has unsaved changes.
Modifier and Type:
java.lang.Boolean

Specified by: wasModified in BaseModel

Returns: Whether or not the object has unsaved changes.

10.18.2 Method Summary

10.18.2.a Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.18.3 Nested Class Summary

10.18.3.a Nested classes/interfaces inherited from class android.app.Activity

- BaseModel.LoadCallback
- BaseModel.SaveCallback

10.18.4 Constructor Summary

10.18.4.a Constructor and Description

ParseBaseModel(com.parse.ParseObject object)
The class constructor.

10.18.5 Generation

The documentation for this class was generated from the following file:

- ParseBaseModel

10.19 ParseGroupModel Class Reference

Class ParseGroupModel

class ParseGroupModel
extendsBaseModel ()
implementsGroupModel ()

Member Functions

- getGeneralLocation ()
- getGeneralDescription ()
- getGroupMembers ()

10.19.1 Method Detail

10.19.1.a getGeneralLocation ( )

public com.parse.ParseGeoPoint getGeneralLocation()

Gets the general location of the group.

Modifier and Type:
com.parse.ParseGeoPoint

Specified by: getGeneralLocation in GroupModel

Returns: The location of the group in the form of a ParseGeoPoint object.

10.19.1.b getGroupDescription ( )

public com.parse.ParseGeoPoint getGeneralLocation()

Gets the description of the current group.

Modifier and Type:
java.lang.String

Specified by: getGroupDescription in GroupModel

Returns: The description string attached to the group.

10.19.1.c getGroupMembers ( )

public com.parse.ParseUser getGroupMembers()

Gets the list of users associated with the current group.

Modifier and Type:
com.parse.ParseUser

Specified by: getGroupMembers in GroupModel

Returns: The list of users as ParseUserModel objects that belong to the group.

10.19.2 Method Summary

10.19.2.a Methods inherited from class com.bowtaps.crowdcontrol.model.ParseBaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.19.2.b Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.19.2.c Methods inherited from interface com.bowtaps.crowdcontrol.model.BaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.19.3 Nested Class Summary

10.19.3.a Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Base

- BaseModel.LoadCallback
- BaseModel.SaveCallback

10.19.4 Constructor Summary

10.19.4.a Constructor and Description

ParseGroupModel(com.parse.ParseObject object)
The class constructor.

10.19.5 Constructor Detail

10.19.5.a ParseGroupModel

The class constructor. Initializes the model from an existing ParseUser.
Parameters: object - The object to use as a handle.

10.19.6 Generation

The documentation for this class was generated from the following file:

- ParseGroupModel

10.20 ParseUserModel Class Reference

Class ParseUserModel

class ParseUserModel
extendsParseBaseModel ()
implementsUserModel ()

The Parse implementation for the @link UserModel interface.

Member Functions

- getAuthData ()
- getEmail ()
- getEmailVerified ()
- getObjectID ()
- getPhone ()
- getUsername ()
- getUsername ()
- setEmail ()
- setPhone ()

10.20.1 Method Detail

10.20.1.a getAuthData ( )

public java.lang.Object getAuthData()

Modifier and Type:
java.lang.Object

Specified by: getAuthData in UserModel

10.20.1.b getEmail ( )

public java.lang.String getEmail()

Gets the user's email.

Modifier and Type:
java.lang.String

Specified by: getEmail in UserModel

10.20.1.c getEmailVerified ( )

public java.lang.Boolean getEmailVerified()

Gets the user's email.

Modifier and Type:
java.lang.Boolean

Specified by: getEmailVerified in UserModel

10.20.1.d getObjectID ( )

public java.lang.String getObjectID()

Modifier and Type:
java.lang.String

Specified by: getObjectID in UserModel

10.20.1.e getPhone ( )

public java.lang.String getPhone()

Gets the user's phone number.

Modifier and Type:
java.lang.String

Specified by: getPhone in UserModel

10.20.1.f getUsername ( )

public java.lang.String getUsername()

Gets the user's username.

Modifier and Type:
java.lang.String

Specified by: getUsername in UserModel

10.20.1.g getUsername ( )

public java.lang.String getUsername()

Modifier and Type:
java.lang.String

Specified by: getUsername in UserModel

10.20.1.h setEmail (java.lang.String *email*)

public void setEmail(java.lang.String email))

Sets the user's email.

Modifier and Type:
void

Specified by: setEmail in UserModel

Parameters:

- email - The new email address to assign to the user.

10.20.1.i setPhone (java.lang.String *phone*)

public void setPhone(java.lang.String phone)

Sets the user's email.

Modifier and Type:
void

Specified by: setPhone in UserModel

Parameters:

- phone - The new phone number to assign to the user.

10.20.2 Method Summary

10.20.2.a Methods inherited from class com.bowtaps.crowdcontrol.model.ParseBaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.20.2.b Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.20.2.c Methods inherited from interface com.bowtaps.crowdcontrol.model.BaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.20.3 Nested Class Summary

10.20.3.a Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Base

- BaseModel.LoadCallback
- BaseModel.SaveCallback

10.20.4 Constructor Summary

10.20.4.a Constructor and Description

ParseUserModel(com.parse.ParseUser object)
The class constructor.

10.20.5 Constructor Detail

10.20.5.a ParseUserModel

public ParseUserModel(com.parse.ParseUser object)

The class constructor. Initializes the model from an existing ParseUser.

Parameters: object - The object to use as a handle.

10.20.6 Generation

The documentation for this class was generated from the following file:

- ParseUserModel

10.21 ParseUserProfileModel Class Reference

Class ParseUserProfileModel

class ParseUserProfileModel
extendsParseBaseModel ()
implementsUserModel ()

The Parse implementation for the @link UserModel interface.

Member Functions

- getDisplayName ()
- setDisplayName ()

10.21.1 Method Detail

10.21.1.a getDisplayName ( )

public java.lang.String getDisplayName()

Gets the user's display name.

Modifier and Type:
java.lang.String

Specified by: getDisplayName in UserProfileModel

Returns: The user's display name.

10.21.1.b setDisplayName (java.lang.String *displayName*)

public void setDisplayName(java.lang.String displayName)

Sets the user's display name.

Modifier and Type:
java.lang.Object

Specified by: setDisplayName in UserProfileModel

Parameters:

- displayName - The new display name for the user.

10.21.2 Method Summary

10.21.2.a Methods inherited from class com.bowtaps.crowdcontrol.model.ParseBaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.21.2.b Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.21.2.c Methods inherited from interface com.bowtaps.crowdcontrol.model.BaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.21.3 Nested Class Summary

10.21.3.a Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Base

- BaseModel.LoadCallback
- BaseModel.SaveCallback

10.21.4 Generation

The documentation for this class was generated from the following file:

- ParseUserProfileModel

10.22 SignupActivity Class Reference

Class SignupActivity

class SignupActivity
extendsandroid.support.v7.app.AppCompatActivity
implements android.app.LoaderManager.LoaderCallbacks<android.database.Cursor>, android.view.View.OnClick

A login screen that offers login via email/password.

All Known Implementing Classes:
android.app.LoaderManager.LoaderCallbacks<android.database.Cursor>,

android.content.ComponentCallbacks, android.content.ComponentCallbacks2, android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback, android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator, android.support.v4.app.TaskStackBuilder.SupportParentable, android.support.v7.app.ActionBarDrawerToggle.DelegateProvider, android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2, android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener, android.view.Window.Callback, android.view.Window.OnWindowDismissedCallback

Member Functions

- onClick ()
- onCreateLoader ()
- onCreateView ()
- onCreateView ()
- onLoaderReset ()
- onLoaderFinished ()
- onRequestPermissionResult ()

10.22.1 Method Detail

10.22.1.a onClick (android.view.View *view*)

public void onClick(android.view.View view)

Called when a view has been clicked.

Modifier and Type:
void

Specified by: onClick in interface android.view.View.OnClickListener

Parameters:

- view - The view that was clicked.

10.22.1.b onCreateLoader (int *i* android.os.Bundle *bundle*)

public android.content.Loader<android.database.Cursor> onCreateLoader(int i, android.os.Bundle bundle)

Instantiate and return a new Loader for the given ID.

Modifier and Type:
android.content.Loader<android.database.Cursor>

Specified by: onClick in interface android.view.View.OnClickListener

Parameters:

- i - The ID whose loader is to be created.
- bundle - Any arguments supplied by the caller.

Returns: Return a new Loader instance that is ready to start loading.

10.22.1.c onCreateView (android.view.View *parent* java.lang.String *name* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(android.view.View parent,
java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory2

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.22.1.d onCreateView (java.lang.String *name* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

standard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.St
This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Parameters:

- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.22.1.e onLoaderReset (android.content.Loader<android.database.Cursor> *cursorLoader*)

public void onLoaderReset(android.content.Loader<android.database.Cursor> cursorLoader)

Called when a previously created loader is being reset, and thus making its data unavailable. The application should at this point remove any references it has to the Loader's data.

Modifier and Type:
void

Specified by: onLoaderReset in interface android.app.LoaderManager.LoaderCallbacks<android.database.Curs

Parameters:

- cursorLoader - The Loader that is being reset.

10.22.1.f onLoadFinished (android.content.Loader<android.database.Cursor> *cursorLoader* android.database.Cursor *cursor*)

public void onLoadFinished(android.content.Loader<android.database.Cursor> cursorLoader, android.database.Cursor cursor)

Called when a previously created loader has finished its load. Note that normally an application is not allowed to commit fragment transactions while in this call, since it can happen after an activity's state is saved. See FragmentManager.openTransaction() for further discussion on this.

This function is guaranteed to be called prior to the release of the last data that was supplied for this Loader. At this point you should remove all use of the old data (since it will be released soon), but should not do your own release of the data since its Loader owns it and will take care of that. The Loader will take care of management of its data so you don't have to. In particular:

- The Loader will monitor for changes to the data, and report them to you through new calls here. You should not monitor the data yourself. For example, if the data is a Cursor and you place it in a CursorAdapter, use the CursorAdapter.CursorAdapter(android.content.Context, android.database.Cursor, int) constructor without passing in either CursorAdapter.FLAG_AUTO_REQUERY or CursorAdapter.FLAG_REGISTER_CONTENT_OBSERVER (that is, use 0 for the flags argument). This prevents the CursorAdapter from doing its own observing of the Cursor, which is not needed since when a change happens you will get a new Cursor throw another call here.
- the Loader will release the data once it knows the application is no longer using it. For example, if the data is a Cursor from a CursorLoader, you should not call close() on it yourself. If the Cursor is being placed in a CursorAdapter, you should use the CursorAdapter.swapCursor(android.database.Cursor) method so that the old Cursor is not closed.

Modifier and Type:
void

Specified by: onLoaderReset in interface android.app.LoaderManager.LoaderCallbacks<android.database.Curs

Parameters:

- cursorLoader - The Loader that has finished.
- cursor - The data generated by the Loader.

10.22.1.g onRequestPermissionsResult (int *requestCode* java.lang.String[] *permissions* int[] *grantResults*)

ublic void onRequestPermissionsResult(int requestCode,
java.lang.String[] permissions,
int[] grantResults)

Callback received when a permissions request has been completed.

Modifier and Type:
void

Specified by: nRequestPermissionsResult in interface android.support.v4.app.ActivityCompat.OnRequestPermi

Overrides: onRequestPermissionsResult in class android.support.v4.app.FragmentActivity

Parameters:

- requestCode - The request code passed in Activity.requestPermissions(String[], int).
- permissions - The requested permissions. Never null.
- grantResults - The grant results for the corresponding permissions which is either PackageManager.PERMISSION_GRANTED or PackageManager.PERMISSION_DENIED. Never null.

10.22.2 Method Summary

10.22.2.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.22.2.b Methods inherited from class android.support.v4.app.FragmentActivity

dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRequestPermissionsResult, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.22.2.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild,

finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot, isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemSelected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreateOptionsMenu, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsItemSelected, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState, onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOutside, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.22.2.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.22.2.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver,

getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStickyOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.22.2.f Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.22.2.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.22.3 Constructor Detail

10.22.3.a SignupActivity

public SignupActivity()

10.22.4 Nested Class Summary

10.22.4.a Nested classes

- SignupActivity.UserLoginTask - Represents an asynchronous login/registration task used to authenticate the user.

10.22.4.b Nested classes/interfaces inherited from class android.app.Activity

- android.app.Activity.TranslucentConversionListener

10.22.4.c Nested classes/interfaces inherited from class android.content.Context

- android.content.Context.BindServiceFlags
- android.content.Context.CreatePackageOptions
- android.content.Context.ServiceName

10.22.5 Feild Summary

10.22.6 Fields inherited from class android.app.Activity

```
DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL,
    DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, RESULT_CANCELED, RESULT_FIRST_USER,
    RESULT_OK
```

10.22.7 Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.22.8 Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.22.9 Generation

The documentation for this class was generated from the following file:

- SignupActivity

10.23 UserModel Class Reference

Class UserModel

class UserModel
extendsBaseModel ()

All Superinterfaces: BaseModel ()

All Known Implementing Classes: ParseUserModel ()

The Parse implementation for the @link UserModel interface.

Member Functions

- getAuthData ()
- getEmail ()
- getEmailVerified ()
- getObjectID ()
- getPhone ()
- getUsername ()
- getUsername ()
- setEmail ()
- setPhone ()

10.23.1 Method Detail

10.23.1.a getAuthData ( )

public java.lang.Object getAuthData()

Modifier and Type:
java.lang.Object

10.23.1.b getEmail ( )

public java.lang.String getEmail()

Gets the user's email.

Modifier and Type:
java.lang.String

Returns:
The user's email address.

10.23.1.c getEmailVerified ( )

public java.lang.Boolean getEmailVerified()

Gets the user's email.

Modifier and Type:
java.lang.Boolean

Returns: True if the user's email address has been verified, false if not.

10.23.1.d getObjectID ( )

public java.lang.String getObjectID()

Modifier and Type:
java.lang.String

10.23.1.e getPhone ( )

public java.lang.String getPhone()

Gets the user's phone number.

Modifier and Type:
java.lang.String

Returns: **The user's phone number.**

10.23.1.f getUsername ( )

public java.lang.String getUsername()

Gets the user's username.

Modifier and Type:
java.lang.String

10.23.1.g getUsername ( )

public java.lang.String getUsername()

ets the username for the user. This value should not be changed, as it is set at creation time and is a unique identifier for this object.

Modifier and Type:
java.lang.String

Returns : **The user's unique username.**

10.23.1.h setEmail (java.lang.String *email*)

public void setEmail(java.lang.String email))

Replaces the user's existing email address with a new one.

Modifier and Type:
void

Parameters:

- **email - The new email address to assign to the user.**

10.23.1.i setPhone (java.lang.String *phone*)

public void setPhone(java.lang.String phone)

Replaces the user's phone number with a new one

Modifier and Type:
void

Parameters:

- phone - The new phone number to assign to the user.

10.23.2 Method Summary

10.23.2.a Methods inherited from class com.bowtaps.crowdcontrol.model.BaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.23.3 Nested Class Summary

10.23.3.a Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Base

- BaseModel.LoadCallback
- BaseModel.SaveCallback

10.23.4 Generation

The documentation for this class was generated from the following file:

- UserModel

10.24 UserProfileModel Interface Reference

Interface UserProfileModel

class ParseUserProfileModel
extendsParseBaseModel ()
implementsUserModel ()

The interface for user profile models, providing access to public-facing user profile data, such as display name and profile image.

Member Functions

- getDisplayName()
- setDisplayName ()

10.24.1 Method Detail

10.24.1.a getDisplayName ( )

public java.lang.String getDisplayName()

Gets the display name of the current user. This value is not unique, is chosen by the user, should always be used to represent the user when presented to other users, and should never be used as an index key. This value can and should, however, be indexed for use in text searches. Additionally, the user's display name can be changed and thus should calls to this function should be performed relatively frequently.

Modifier and Type:
java.lang.String

Returns: String of the user's self-chosen display name.

10.24.1.b setDisplayName (java.lang.String *displayName*)

public void setDisplayName(java.lang.String displayName)

Sets the user's display name that will be seen by other users.

Modifier and Type:
java.lang.Object

Parameters:

- displayName - The new display name for the user.

10.24.2 Method Summary

10.24.2.a Methods inherited from interface com.bowtaps.crowdcontrol.model.BaseModel

getCreated, getId, getUpdated, load, loadInBackground, save, saveInBackground, wasModified

10.24.3 Nested Class Summary

10.24.3.a Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Base

- BaseModel.LoadCallback
- BaseModel.SaveCallback

10.24.4 Generation

The documentation for this class was generated from the following file:

- UserProfileModel

10.25 WelcomeActivity Class Reference

Class WelcomeActivityl

class WelcomeActivity
extendsandroid.support.v7.app.AppCompatActivity
implementsndroid.view.View.OnClickListener

All Known Implementing Classes:
android.content.ComponentCallbacks, android.content.ComponentCallbacks2,
android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback,
android.support.v4.app.ActivityCompatApi23.RequestPermissionsRequestCodeValidator,

android.support.v4.app.TaskStackBuilder.SupportParentable,
android.support.v7.app.ActionBarDrawerToggle.DelegateProvider,
android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback,
android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2,
android.view.View.OnClickListener, android.view.View.OnCreateContextMenuListener,
android.view.Window.Callback, android.view.Window.OnWindowDismissedCallback

Member Functions

- onClick ()
- onCreateOptionsMenu ()
- onCreateView ()
- onCreateView ()
- onOptionsItemSelected ()

10.25.1 Method Detail

10.25.1.a onClick (android.view.View *view*)

public void onClick(android.view.View view)

Called when a view has been clicked.

Modifier and Type:
void

Specified by: onClick in interface android.view.View.OnClickListener

Parameters:

- view - The view that was clicked.

10.25.1.b onCreateOptionsMenu (android.view.Menu *menu*)

public boolean onCreateOptionsMenu(android.view.Menu menu)

Initialize the contents of the Activity's standard options menu. You should place your menu items in to menu.

This is only called once, the first time the options menu is displayed. To update the menu every time it is displayed, see Activity.onPrepareOptionsMenu(android.view.Menu).

The default implementation populates the menu with standard system menu items. These are placed in the Menu.CATEGORY_SYSTEM group so that they will be correctly ordered with applicationdefined menu items. Deriving classes should always call through to the base implementation.

You can safely hold on to menu (and any items created from it), making modifications to it as desired, until the next time onCreateOptionsMenu() is called.

When you add items to the menu, you can implement the Activity's Activity.onOptionsItemSelected(android.view.Mer
method to handle them there.

Modifier and Type:
boolean

Overrides: onCreateOptionsMenu in class android.app.Activity

Parameters:

- menu - The options menu in which you place your items.

Returns: You must return true for the menu to be displayed; if you return false it will not be shown.

10.25.1.c onCreateView (android.view.View *parent* java.lang.String *nam* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(android.view.View parent,
java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory2.onCreateView(View, String, Context, AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.String). This implementation handles tags to embed fragments inside of the activity.

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory2

Overrides: onCreateOptionsMenu in class android.app.Activity

Parameters:

- parent - The parent that the created view will be placed in; note that this may be null.
- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.25.1.d onCreateView (java.lang.String *nam* android.content.Context *context* android.util.AttributeSet *attrs*)

public android.view.View onCreateView(java.lang.String name,
android.content.Context context,
android.util.AttributeSet attrs)

Standard implementation of LayoutInflater.Factory.onCreateView(java.lang.String, android.content.Context, android.util.AttributeSet) used when inflating with the LayoutInflater returned by Activity.getSystemService(java.lang.St
This implementation does nothing and is for pre-Build.VERSION_CODES.HONEYCOMB apps. Newer apps should use Activity.onCreateView(View, String, Context, AttributeSet).

Modifier and Type:
android.view.View

Specified by: onCreateView in interface android.view.LayoutInflater.Factory

Overrides: onCreateOptionsMenu in class android.app.Activity

Parameters:

- name - Tag name to be inflated.
- context - The context the view is being created in.
- attrs - Inflation attributes as specified in XML file.

Returns: View Newly created view. Return null for the default behavior.

10.25.2 Method Summary

10.25.2.a Methods inherited from class android.support.v7.app.AppCompatActivity

addContentView, getDelegate, getDrawerToggleDelegate, getMenuInflater, getSupportActionBar, getSupportParentActivityIntent, invalidateOptionsMenu, onConfigurationChanged, onContentChanged, onCreateSupportNavigateUpTaskStack, onMenuItemSelected, onMenuOpened, onPanelClosed, onPrepareSupportNavigateUpTaskStack, onSupportActionModeFinished, onSupportActionModeStarted, onSupportContentChanged, onSupportNavigateUp, onWindowStartingSupportActionMode, setContentView, setContentView, setContentView, setSupportActionBar, setSupportProgress, setSupportProgressBarIndeterminate, setSupportProgressBarIndeterminateVisibility, setSupportProgressBarVisibility, startSupportActionMode, supportInvalidateOptionsMenu, supportNavigateUpTo, supportRequestWindowFeature, supportShouldUpRecreateTask

10.25.2.b Methods inherited from class android.support.v4.app.FragmentActivity

dump, getLastCustomNonConfigurationInstance, getSupportFragmentManager, getSupportLoaderManager, onAttachFragment, onBackPressed, onCreatePanelMenu, onKeyDown, onLowMemory, onPreparePanel, onRequestPermissionsResult, onRetainCustomNonConfigurationInstance, onRetainNonConfigurationInstance, onStateNotSaved, setEnterSharedElementCallback, setExitSharedElementCallback, startActivityForResult, startActivityFromFragment, supportFinishAfterTransition, supportPostponeEnterTransition, supportStartPostponedEnterTransition, validateRequestPermissionsRequestCode

10.25.2.c Methods inherited from class android.app.Activity

canStartActivityForResult, closeContextMenu, closeOptionsMenu, convertFromTranslucent, convertToTranslucent, createPendingResult, dismissDialog, dispatchEnterAnimationComplete, dispatchGenericMotionEvent, dispatchKeyEvent, dispatchKeyShortcutEvent, dispatchPopulateAccessibilityEvent, dispatchTouchEvent, dispatchTrackballEvent, findViewById, finish, finishActivity, finishActivityFromChild, finishAffinity, finishAfterTransition, finishAndRemoveTask, finishFromChild, getActionBar, getActivityToken, getApplication, getCallingActivity, getCallingPackage, getChangingConfigurations, getComponentName, getContentScene, getContentTransitionManager, getCurrentFocus, getFragmentManager, getIntent, getLastNonConfigurationInstance, getLayoutInflater, getLoaderManager, getLocalClassName, getMediaController, getParent, getParentActivityIntent, getPreferences, getReferrer, getRequestedOrientation, getSearchEvent, getSystemService, getTaskId, getTitle, getTitleColor, getVoiceInteractor, getVolumeControlStream, getWindow, getWindowManager, hasWindowFocus, isBackgroundVisibleBehind, isChangingConfigurations, isChild, isDestroyed, isFinishing, isImmersive, isResumed, isTaskRoot, isVoiceInteraction, isVoiceInteractionRoot, managedQuery, managedQuery, moveTaskToBack, navigateUpTo, navigateUpToFromChild, onActionModeFinished, onActionModeStarted, onActivityReenter, onAttachedToWindow, onAttachFragment, onBackgroundVisibleBehindChanged, onContextItemS-

elected, onContextMenuClosed, onCreate, onCreateContextMenu, onCreateDescription, onCreateNavigateUpTaskStack, onCreatePanelView, onCreateThumbnail, onDetachedFromWindow, onEnterAnimationComplete, onGenericMotionEvent, onKeyLongPress, onKeyMultiple, onKeyShortcut, onKeyUp, onNavigateUp, onNavigateUpFromChild, onNewActivityOptions, onOptionsMenuClosed, onPostCreate, onPrepareNavigateUpTaskStack, onPrepareOptionsMenu, onProvideAssistContent, onProvideAssistData, onProvideReferrer, onRestoreInstanceState, onSaveInstanceState, onSearchRequested, onSearchRequested, onTouchEvent, onTrackballEvent, onTrimMemory, onUserInteraction, onVisibleBehindCanceled, onWindowAttributesChanged, onWindowDismissed, onWindowFocusChanged, onWindowStartingActionMode, onWindowStartingActionMode, openContextMenu, openOptionsMenu, overridePendingTransition, postponeEnterTransition, recreate, registerForContextMenu, releaseInstance, removeDialog, reportFullyDrawn, requestPermissions, requestVisibleBehind, requestWindowFeature, runOnUiThread, setActionBar, setContentTransitionManager, setDefaultKeyMode, setEnterSharedElementCallback, setExitSharedElementCallback, setFeatureDrawable, setFeatureDrawableAlpha, setFeatureDrawableResource, setFeatureDrawableUri, setFinishOnTouchOutside, setImmersive, setIntent, setMediaController, setPersistent, setProgress, setProgressBarIndeterminate, setProgressBarIndeterminateVisibility, setProgressBarVisibility, setRequestedOrientation, setResult, setResult, setSecondaryProgress, setTaskDescription, setTitle, setTitle, setTitleColor, setVisible, setVolumeControlStream, shouldShowRequestPermissionRationale, shouldUpRecreateTask, showAssist, showDialog, showDialog, showLockTaskEscapeMessage, startActionMode, startActionMode, startActivities, startActivities, startActivity, startActivity, startActivityAsCaller, startActivityAsUser, startActivityAsUser, startActivityForResult, startActivityForResult, startActivityForResultAsUser, startActivityForResultAsUser, startActivityFromChild, startActivityFromChild, startActivityFromFragment, startActivityFromFragment, startActivityIfNeeded, startActivityIfNeeded, startIntentSender, startIntentSender, startIntentSenderForResult, startIntentSenderForResult, startIntentSenderFromChild, startIntentSenderFromChild, startLockTask, startManagingCursor, startNextMatchingActivity, startNextMatchingActivity, startPostponedEnterTransition, startSearch, stopLockTask, stopManagingCursor, takeKeyEvents, triggerSearch, unregisterForContextMenu

10.25.2.d Methods inherited from class android.view.ContextThemeWrapper

applyOverrideConfiguration, getResources, getTheme, getThemeResId, setTheme

10.25.2.e Methods inherited from class android.content.ContextWrapper

bindService, bindServiceAsUser, checkCallingOrSelfPermission, checkCallingOrSelfUriPermission, checkCallingPermission, checkCallingUriPermission, checkPermission, checkPermission, checkSelfPermission, checkUriPermission, checkUriPermission, checkUriPermission, clearWallpaper, createApplicationContext, createConfigurationContext, createDisplayContext, createPackageContext, createPackageContextAsUser, databaseList, deleteDatabase, deleteFile, enforceCallingOrSelfPermission, enforceCallingOrSelfUriPermission, enforceCallingPermission, enforceCallingUriPermission, enforcePermission, enforceUriPermission, enforceUriPermission, fileList, getApplicationContext, getApplicationInfo, getAssets, getBaseContext, getBasePackageName, getCacheDir, getClassLoader, getCodeCacheDir, getContentResolver, getDatabasePath, getDir, getDisplayAdjustments, getExternalCacheDir, getExternalCacheDirs, getExternalFilesDir, getExternalFilesDirs, getExternalMediaDirs, getFilesDir, getFileStreamPath, getMainLooper, getNoBackupFilesDir, getObbDir, getObbDirs, getOpPackageName, getPackageCodePath, getPackageManager, getPackageName, getPackageResourcePath, getSharedPreferences, getSharedPrefsFile, getSystemServiceName, getUserId, getWallpaper, getWallpaperDesiredMinimumHeight, getWallpaperDesiredMinimumWidth, grantUriPermission, isRestricted, openFileInput, openFileOutput, openOrCreateDatabase, openOrCreateDatabase, peekWallpaper, registerReceiver, registerReceiver, registerReceiverAsUser, removeStickyBroadcast, removeStickyBroadcastAsUser, revokeUriPermission, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcast, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastAsUser, sendBroadcastMultiplePermissions, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcast, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendOrderedBroadcastAsUser, sendStickyBroadcast, sendStickyBroadcastAsUser, sendStick-

yOrderedBroadcast, sendStickyOrderedBroadcastAsUser, setWallpaper, setWallpaper, startActivitiesAsUser, startInstrumentation, startService, startServiceAsUser, stopService, stopServiceAsUser, unbindService, unregisterReceiver

10.25.2.f Methods inherited from class android.content.Context

getColor, getColorStateList, getDrawable, getString, getString, getSystemService, getText, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, obtainStyledAttributes, registerComponentCallbacks, unregisterComponentCallbacks

10.25.2.g Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

10.25.3 Nested Class Summary

10.25.3.a Nested classes/interfaces inherited from class android.app.Activity

- **android.app.Activity.TranslucentConversionListener**

10.25.3.b Nested classes/interfaces inherited from interface com.bowtaps.crowdcontrol.model.Bas

- **android.content.Context.BindServiceFlags**
- **android.content.Context.CreatePackageOptions**
- **android.content.Context.ServiceName**

10.25.4 Field Summary

10.25.4.a Fields inherited from class android.app.Activity

```
DEFAULT_KEYS_DIALER, DEFAULT_KEYS_DISABLE, DEFAULT_KEYS_SEARCH_GLOBAL,
    DEFAULT_KEYS_SEARCH_LOCAL, DEFAULT_KEYS_SHORTCUT, RESULT_CANCELED, RESULT_FIRST_USER,
    RESULT_OK
```

10.25.4.b Fields inherited from class android.content.Context

```
ACCESSIBILITY_SERVICE, ACCOUNT_SERVICE, ACTIVITY_SERVICE, ALARM_SERVICE, APP_OPS_SERVICE,
    APPWIDGET_SERVICE, AUDIO_SERVICE, BACKUP_SERVICE, BATTERY_SERVICE, BIND_ABOVE_CLIENT,
    BIND_ADJUST_WITH_ACTIVITY, BIND_ALLOW_OOM_MANAGEMENT, BIND_AUTO_CREATE,
    BIND_DEBUG_UNBIND, BIND_FOREGROUND_SERVICE, BIND_FOREGROUND_SERVICE_WHILE_AWAKE,
    BIND_IMPORTANT, BIND_NOT_FOREGROUND, BIND_NOT_VISIBLE, BIND_SHOWING_UI,
    BIND_TREAT_LIKE_ACTIVITY, BIND_VISIBLE, BIND_WAIVE_PRIORITY, BLUETOOTH_SERVICE,
    CAMERA_SERVICE, CAPTIONING_SERVICE, CARRIER_CONFIG_SERVICE, CLIPBOARD_SERVICE,
    CONNECTIVITY_SERVICE, CONSUMER_IR_SERVICE, CONTEXT_IGNORE_SECURITY,
    CONTEXT_INCLUDE_CODE, CONTEXT_REGISTER_PACKAGE, CONTEXT_RESTRICTED, COUNTRY_DETECTOR,
    DEVICE_IDLE_CONTROLLER, DEVICE_POLICY_SERVICE, DISPLAY_SERVICE, DOWNLOAD_SERVICE,
    DROPBOX_SERVICE, ETHERNET_SERVICE, FINGERPRINT_SERVICE, HDMI_CONTROL_SERVICE,
    INPUT_METHOD_SERVICE, INPUT_SERVICE, JOB_SCHEDULER_SERVICE, KEYGUARD_SERVICE,
    LAUNCHER_APPS_SERVICE, LAYOUT_INFLATER_SERVICE, LOCATION_SERVICE,
    MEDIA_PROJECTION_SERVICE, MEDIA_ROUTER_SERVICE, MEDIA_SESSION_SERVICE, MIDI_SERVICE,
    MODE_APPEND, MODE_ENABLE_WRITE_AHEAD_LOGGING, MODE_MULTI_PROCESS, MODE_PRIVATE,
    MODE_WORLD_READABLE, MODE_WORLD_WRITEABLE, NETWORK_POLICY_SERVICE,
    NETWORK_SCORE_SERVICE, NETWORK_STATS_SERVICE, NETWORKMANAGEMENT_SERVICE, NFC_SERVICE,
    NOTIFICATION_SERVICE, NSD_SERVICE, PERSISTENT_DATA_BLOCK_SERVICE, POWER_SERVICE,
    PRINT_SERVICE, RADIO_SERVICE, RESTRICTIONS_SERVICE, SEARCH_SERVICE, SENSOR_SERVICE,
    SERIAL_SERVICE, SIP_SERVICE, STATUS_BAR_SERVICE, STORAGE_SERVICE, TELECOM_SERVICE,
    TELEPHONY_SERVICE, TELEPHONY_SUBSCRIPTION_SERVICE, TEXT_SERVICES_MANAGER_SERVICE,
    TRUST_SERVICE, TV_INPUT_SERVICE, UI_MODE_SERVICE, UPDATE_LOCK_SERVICE,
    USAGE_STATS_SERVICE, USB_SERVICE, USER_SERVICE, VIBRATOR_SERVICE,
    VOICE_INTERACTION_MANAGER_SERVICE, WALLPAPER_SERVICE, WIFI_P2P_SERVICE,
    WIFI_PASSPOINT_SERVICE, WIFI_RTT_SERVICE, WIFI_SCANNING_SERVICE, WIFI_SERVICE,
    WINDOW_SERVICE
```

10.25.4.c Fields inherited from interface android.content.ComponentCallbacks2

```
TRIM_MEMORY_BACKGROUND, TRIM_MEMORY_COMPLETE, TRIM_MEMORY_MODERATE,
    TRIM_MEMORY_RUNNING_CRITICAL, TRIM_MEMORY_RUNNING_LOW, TRIM_MEMORY_RUNNING_MODERATE,
    TRIM_MEMORY_UI_HIDDEN
```

10.25.5 Generation

The documentation for this class was generated from the following file:

- WelcomeActivity

10.26 AppDelegate Class Reference

Class AppDelegate

public class AppDelegate

class AppDelegate: UIResponder, UIApplicationDelegate

The core delegate for the app. Contains pseudo-globa variables and functions that can be accessed from anywhere in the app

Member Functions

- instance ()
- window ()
- modelManager ()
- application) ()
- application ()
- applicationWillResignActive ()
- applicationDidEnterBackground ()
- applicationWillEnterForeground ()
- applicationDidBecomeActive ()
- applicationWillTerminate ()

10.26.1 Method Summary

10.26.1.a instance ( )

The main instance of this class for the entire application.

Modifier and Type:
static var instance

Declaration: Swift

static var instance: AppDelegate

10.26.1.b window ( )

The main instance of this class for the entire application.

Modifier and Type:
var window

Declaration: Swift

var window: UIWindow?

10.26.1.c modelManager ( )

Optional reference to the current model manager.

Modifier and Type:
var modelManager

Declaration: Swift

var modelManager: ModelManager?

10.26.1.d application ( )

application(_:didFinishLaunchingWithOptions:)
Modifier and Type:
class AppDelegate

Declaration: Swift

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.1.e application ( )

application(_:openURL:sourceApplication:annotation:)
Modifier and Type:
class AppDelegate

Declaration: **Swift**

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.1.f applicationWillResignActive ( )

applicationWillResignActive(_:)
Modifier and Type:
class AppDelegate

Declaration: **Swift**

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.1.g applicationDidEnterBackground ( )

applicationDidEnterBackground(_:)
Modifier and Type:
class AppDelegate

Declaration: **Swift**

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.1.h applicationWillEnterForeground ( )

applicationWillEnterForeground(_:)
Modifier and Type:
class AppDelegate

Declaration: **Swift**

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.1.i applicationDidBecomeActive ( )

applicationDidBecomeActive(_:)
Modifier and Type:
class AppDelegate

Declaration: **Swift**

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.1.j applicationWillTerminate ( )

applicationWillTerminate(_:)
Modifier and Type:
class AppDelegate

Declaration: Swift

class AppDelegate: UIResponder, UIApplicationDelegate

10.26.2 Generation

The documentation for this class was generated from the following file:

- AppDelegate

10.27 BaseModel Class Reference

Class BaseModel

public class BaseModel
protocol BaseModel

Base model protocol, which is to be implemented by all model classes in the project. Provides access to basic information about the model, such as:

- Unique identifier
- Creation timestamp
- Last updated timestamp
- Flag indicating modification since last save
- Methods for loading from and saving to storage

Member Functions

- id ()
- created ()
- updated ()
- modified ()
- load ()
- loadInBackground ()
- save ()
- saveInBackground ()

10.27.1 Method Summary

10.27.1.a id ( )

The ID of the object as determined by remote storage. This value is automatically generated when the object is stored and is usually determined by a remote server. It is valid for this value to be nil, and thus should be used with care and only if absolutely necessary.

Modifier and Type:
var id:

Declaration: Swift

var id: String get

10.27.1.b created ( )

Timestamp of when this object was first created and stored. This value may be automatically determined by the server if the model is using a remote machine for storage.

Modifier and Type:
var created:

Declaration: Swift

var created: NSDate get

10.27.1.c updated ( )

Timestamp of the last time this object was updated in storage. This value may be automatically determined by the server if the model is using a remote machine for storage.

Modifier and Type:
var created:

Declaration: Swift

var created: NSDate get

10.27.1.d modified ( )

Flag indicating whether or not this object has been modified since it was last pulled from or pushed to storage.

Modifier and Type:
var modified:

Declaration: Swift

var modified: Bool get

10.27.1.e load ( )

Loads this object from storage, either local or remote as determined by the implementation.

This is a blocking function and should be executed on a thread separate from the main thread. See loadInBackground(_:) for loading this object on a separate thread. This function will throw an exception if an error occurs.

Modifier and Type:
func

Declaration: Swift

func load() throws

10.27.1.f loadInBackground ( )

loadInBackground(_:) Loads this object from storage, either local or remote as determined by the implementation.

This function spawns a new thread and reloads this object from storage. After successful execution or if an error occurs, this function passes control back to the main thread by calling the closure that was optionally passed as an argument.

Modifier and Type:
var modified

Declaration: Swift

func loadInBackground(callback: ((object: BaseModel?, error: NSError?) -> Void)?)

Parameters:

- callback - Optional callback function to call after successful execution or if an error occurs. If nil is provided, then the callback will not be called.

10.27.1.g save ( )

Saves this object storage, either local or remote as determined by the implementation.

This is a blocking function and should be executed on a thread separate from the main thread. See saveInBackground(_:) for saving this object on a separate thread. This function will throw an exception if

Modifier and Type:
func

Declaration: Swift

save() throws

10.27.1.h save ( )

saveInBackground(_:)

Saves this object to storage, either local or remote as determined by the implementation.

This function spawns a new thread and sends this object to storage. After successful execution or if an error occurs, this function passes control back to the main thread by calling the closure that was optionally passed as an argument.

Modifier and Type:
func

Declaration: Swift

func saveInBackground(callback: ((object: BaseModel?, error: NSError?) -> Void)?)

Parameters:

- callback - Optional callback function to call after successful execution or if an error occurs. If nil is provided, then the callback will not be called.

10.27.2 Generation

The documentation for this class was generated from the following file:

- BaseModel

10.28 ChatViewController Class Reference

Class ChatViewController

public class ChatViewController

class ChatViewController: UITableViewController

Controller for the UITableView used to display a list of active conversations on the chat screen. Contains methods that fill the rows in the table with relevant content as well as making calls to the backend to load the current conversations

Member Functions

- conversations ()
- viewDidLoad ()
- numberOfSectionsInTableView ()
- tableView ()
- tableView ()

10.28.1 Method Summary

10.28.1.a conversations ( )

Array of ConversationModel objects to display in the table.

Modifier and Type:
var conversations:

Declaration: Swift

var conversations: [ConversationModel] = [

10.28.1.b viewDidLoad ( )

Modifier and Type:
class ChatViewController:

Declaration: Swift

class ChatViewController: UITableViewController

10.28.1.c numberOfSectionsInTableView ( )

Modifier and Type:
class ChatViewController:

Declaration: **Swift**

class ChatViewController: UITableViewController

10.28.1.d tableView ( )

tableView(_:numberOfRowsInSection:)
Modifier and Type:
class ChatViewController:

Declaration: **Swift**

class ChatViewController: UITableViewController

10.28.1.e tableView ( )

tableView(_:cellForRowAtIndexPath:)
Modifier and Type:
class ChatViewController:

Declaration: **Swift**

class ChatViewController: UITableViewController

10.28.2 Generation

The documentation for this class was generated from the following file:

- **ChatViewController**

10.29 ConversationModel Class Reference

Class ConversationModel

public class ConversationModel
class ConversationModel
Provides a simple model to be used as placeholder content for the UITableView.

Member Functions

- **name ()**
- **message ()**
- **time ()**
- **init ()**

10.29.1 Method Summary

10.29.1.a name ( )

Name of conversation

Modifier and Type:
let name:

Declaration: Swift

let name: String

10.29.1.b message ( )

Contents of the message

Modifier and Type:
let message:

Declaration: Swift

let message: String

10.29.1.c time ( )

Conversation timestamp

Modifier and Type:
let time:

Declaration: Swift

let time: String

10.29.1.d init ( )

Main constructor for the class. Initializes values using provided parameters. - Parameter name: The name of the conversation or person with whom the conversation is with. - Parameter message: The last message sent in the conversation. - Parameter time: The last time stamp a message was sent in this conversation.

Modifier and Type:
init

Declaration: Swift

init(name: String, message: String, time: String)

Parameters:

- name - The name of the conversation or person with whom the conversation is with.
- message - The last message sent in the conversation.

- time - The last time stamp a message was sent in this conversation.

10.29.2 Generation

The documentation for this class was generated from the following file:

- ConversationModel

10.30 GroupInfoviewController Class Reference

Class GroupInfoviewController

public class GroupInfoviewController

class GroupInfoViewController: UIViewController

Controller for manipulating the group info view, which displays basic group information and lists of members and group leaders.

Member Functions

- viewDidLoad ()
- didReceiveMemoryWarning ()
- onLeaveGroupButtonTapped ()

10.30.1 Method Summary

10.30.1.a viewDidLoad ( )

Override of super class

Modifier and Type:
override func

Declaration: Swift

override func viewDidLoad()

10.30.1.b didReceiveMemoryWarning ( )

Override of super class

Modifier and Type:
override func

Declaration: Swift

override func didReceiveMemoryWarning()

10.30.1.c onLeaveGroupButtonTapped ( )

Group leave button callback - Parameter sender: Button that called

Modifier and Type:
override func

Declaration: Swift

```
@IBAction func onLeaveGroupButtonTapped(sender: AnyObject)
```

Parameters:

- sender - Button that called

10.30.2 Generation

The documentation for this class was generated from the following file:

- GroupInfoviewController

10.31 GroupModel Class Reference

Class GroupModel

public class GroupModel
protocol GroupModel: BaseModel

Model protocol representing a group of users using the app. This model contains data accessible to all users of the app, including group name, group description, group leader, and members of the group. This model does not provide direct access to messages related to the group, nor does does it allow access to member locations, although it does provide a generic group location.

Member Functions

- generalLocation ()
- groupDescription ()
- groupName ()
- groupMembers ()
- groupLeader ()
- addGroupMember ()
- removeGroupMember ()

10.31.1 Method Summary

10.31.1.a generalLocation ( )

Group's general location, used for location filtering

Modifier and Type:
PFGeoPoint

Declaration: Swift

```
var generalLocation: PFGeoPoint get set
```

10.31.1.b groupDescription ( )

Group description set by the group leader during creation

Modifier and Type:
String

Declaration: Swift

var groupDescription: String get set

10.31.1.c groupName ( )

Group name set by the group leader during creation

Modifier and Type:
String

Declaration: Swift

var groupName: String get set

10.31.1.d groupMembers ( )

List of users who are members of the group. This property can only be modified by the addGroupMember(_:) and removeGroupMember(_:) methods.

Modifier and Type:
Array

Declaration: Swift

var groupMembers: [UserProfileModel] get

10.31.1.e groupLeader ( )

The user who is the designated leader of the group, usually the user who created the group.

Modifier and Type:
Array

Declaration: Swift

var groupLeader: UserProfileModel? get set

10.31.1.f addGroupMemeber ( )

addGroupMemeber(_:)
Method for adding a user as a member of the group. Model must be saved afterwards, as this method does not automatically save the model.

Modifier and Type:
Array

Declaration: func

func addGroupMember(member: UserProfileModel) -> Bool

10.31.1.g removeGroupMemeber ( )

removeGroupMemeber(_:)
Method for removing a user from the group. Model must be saved afterwards, as this method does not automatically save the model.

Modifier and Type:
Array

Declaration: func

func removeGroupMember(member: UserProfileModel) -> Bool

10.31.2 Generation

The documentation for this class was generated from the following file:

- GroupModel

10.32 GroupOverviewController Class Reference

Class GroupOverviewController

public class GroupOverviewController
class GroupOverviewController: UIViewController
Group Overview Controller contains the logic for the Group Overiew page.

Member Functions

- groupToDisplay ()
- groupLeader ()
- groupNameLabel ()
- groupLeaderLabel ()
- groupDescriptionLabel ()
- onRequestButtonTapped ()
- viewDidLoad ()

10.32.1 Method Summary

10.32.1.a groupToDisplay ( )

Group to display in current view

Modifier and Type:
var groupToDisplay:

Declaration: Swift

var groupToDisplay: GroupModel?

10.32.1.b groupLeader ( )

User display name of the Group Leader

Modifier and Type:
var groupLeader:

Declaration: Swift

var groupLeader: UserProfileModel?

10.32.1.c groupNameLabel ( )

UILabel for group name

Modifier and Type:
@IBOutlet weak var groupNameLabel:

Declaration: Swift

@IBOutlet weak var groupNameLabel: UILabel!

10.32.1.d groupLeaderLabel ( )

UILabel for group leader name

Modifier and Type:
@IBOutlet weak var groupLeaderLabel:

Declaration: Swift

@IBOutlet weak var groupLeaderLabel: UILabel!

10.32.1.e groupDescriptionLabel ( )

UILable for group description text

Modifier and Type:
@IBOutlet weak var groupDescriptionLabel:

Declaration: Swift

@IBOutlet weak var groupDescriptionLabel: UILabel!

10.32.1.f onRequestButtonTapped ( )

When user clicks on the group to join - Parameter sender: Caller of button

Modifier and Type:
@IBAction func onRequestButtonTapped

Declaration: Swift

```
@IBAction func onRequestButtonTapped(sender: AnyObject)
```

Parameters:

- sender - Caller of button

10.32.1.g viewDidLoad ( )

Overrides superclass then loads data from group object into view.

Modifier and Type:
override func

Declaration: Swift

```
override func viewDidLoad()
```

10.32.2 Generation

The documentation for this class was generated from the following file:

- GroupOverviewController

10.33 GroupTableController Class Reference

Class GroupTableController

public class GroupTableController
class GroupTableController: UITableViewController
Contains the logic for the Group table view

Member Functions

- groupTable ()
- groups ()
- SelectedGroup ()
- ViewDidLoad ()
- viewDidAppear ()
- rewindToGroupList ()
- numberOfSectionsInTableView ()
- tableView ()
- tableView ()
- tableView ()
- doRefresh ()
- prepareForSegue ()

10.33.1 Method Summary

10.33.1.a groupTable ( )

Modifier and Type:
class GroupTableController:

Declaration: **Swift**

class GroupTableController: UITableViewController

10.33.1.b groups ( )

Modifier and Type:
class GroupTableController:

Declaration: **Swift**

class GroupTableController: UITableViewController

10.33.1.c selectedGroup ( )

Modifier and Type:
class GroupTableController:

Declaration: **Swift**

class GroupTableController: UITableViewController

10.33.1.d viewDidLoad ( )

Modifier and Type:
class GroupTableController:

Declaration: **Swift**

class GroupTableController: UITableViewController

10.33.1.e viewDidAppear ( )

Modifier and Type:
class GroupTableController:

Declaration: **Swift**

class GroupTableController: UITableViewController

10.33.1.f rewindToGroupList ( )

Function for rewinding to this view, providing a nice target for screens to rewind to.

Parameter segue: The segue object communicated during transfer.

Modifier and Type:
@IBAction func rewindToGroupList

Declaration: Swift

@IBAction func rewindToGroupList(segue: UIStoryboardSegue)

Parameters:

- **seque - The segue object communicated during transfer.**

10.33.1.g numberOfSectionsInTableView ( )

Modifier and Type:
class GroupTableController:

Declaration: Swift

class GroupTableController: UITableViewController

10.33.1.h tableView ( )

tableView(_:numberOfRowsInSection:)

Modifier and Type:
class GroupTableController:

Declaration: Swift

class GroupTableController: UITableViewController

10.33.1.i tableView ( )

tableView(_:cellForRowAtIndexPath:)

Modifier and Type:
class GroupTableController:

Declaration: Swift

class GroupTableController: UITableViewController

10.33.1.j tableView ( )

tableView(_:didSelectRowAtIndexPath:)

Modifier and Type:
class GroupTableController:

Declaration: Swift

class GroupTableController: UITableViewController

10.33.1.k doRefresh ( )

Modifier and Type:
class GroupTableController:

Declaration: Swift

class GroupTableController: UITableViewController

10.33.1.l prepareForSegue ( )

prepareForSegue(_:sender:)

Modifier and Type:
class GroupTableController:

Declaration: Swift

class GroupTableController: UITableViewController

10.33.2 Generation

The documentation for this class was generated from the following file:

- GroupTableController

10.34 LocationModel Class Reference

Class LocationModel

public class LocationModel
protocol LocationModel: BaseModel

Base model protocol, which is to be implemented by all model classes in the project. Provides access to basic information about the model, such as:

Member Functions

- longitude ()
- latitude ()
- recipient ()
- sender ()

10.34.1 Method Summary

10.34.1.a longitude ( )

String holding the longitudinal coordinate

Modifier and Type:
String

Declaration: **Swift**

var longitude: String {get set}

10.34.1.b latitude ( )

String holding the latitudel coordinate

Modifier and Type:
String

Declaration: **Swift**

var latitude: String {get set}

10.34.1.c recipient ( )

UserProfileModel holding the recipient information when the locations are being sent to parse so that they can be sent encrypted to the intended recipient

Modifier and Type:
String

Declaration: **Swift**

var recipient: UserProfileModel {get}

10.34.1.d sender ( )

This is the Current users' UserProfileModel holding the reference for the storage on parse.

Modifier and Type:
String

Declaration: **Swift**

var sender: UserProfileModel {get}

10.34.2 Generation

The documentation for this class was generated from the following file:

- **LocationModel**

10.35 LoginViewController Class Reference

Class LoginViewController

public class LoginViewController

class LoginViewController: UIViewController, UITextFieldDelegate
Custom view controller class for handling user logins.

Custom UIViewController class for handling user logins. Processes requests, validates input, and passes data to backend for checking with the server, and properly handles both successful and unsuccessful login requests.

Member Functions

- scrollView ()
- emailField ()
- passwordField ()
- submitButton ()
- loginFormFields ()
- init ()
- deinit ()
- viewDidLoad ()
- registerForKeyboardNotifications ()
- deregisterFromKeyboardNotifications ()
- adjustForKeyboard ()
- submitButtonTapped ()
- textFieldShouldReturn ()
- submitForm ()

10.35.1 Method Summary

10.35.1.a scrollView ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.b emailField ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.c passwordField ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.d submitButton ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.e loginFormFields ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.f init ( )

init(coder:)

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.g deinit ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.h viewDidLoad ( )

Modifier and Type:
class LoginViewController:

Declaration: Swift

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.i registerForKeyboardNotifications ( )

Modifier and Type:
class LoginViewController:

Declaration: **Swift**

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.j deregisterFromKeyboardNotifications ( )

Modifier and Type:
class LoginViewController:

Declaration: **Swift**

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.k adjustForKeyboard ( )

adjustForKeyboard(_:)
Modifier and Type:
class LoginViewController:

Declaration: **Swift**

class LoginViewController: UIViewController, UITextFieldDelegate

10.35.1.l submitButtonTapped ( )

submitButtonTapped(_:) Callback executed when the submit button has been tapped.
Modifier and Type:
@IBAction func submitButtonTapped

Declaration: **Swift**

@IBAction func submitButtonTapped(sender: AnyObject)

10.35.1.m textFieldShouldReturn ( )

textFieldShouldReturn(_:) Callback executed when the return button is tapped on the keyboard.

Modifier and Type:
@IBAction func submitButtonTapped

Declaration: **Swift**

func textFieldShouldReturn(textField: UITextField) -> Bool

10.35.1.n submitForm ( )

Hide the keyboard, validate form input, and attempt to log in the user.

Modifier and Type:
func

Declaration: Swift

func submitForm()

10.35.2 Generation

The documentation for this class was generated from the following file:

- LoginViewController

10.36 MapViewController Class Reference

Class MapViewController

public class MapViewController

lass MapViewController: UIViewController, CLLocationManagerDelegate

Controller for the map view, which displays an interactive map that displays the user's current location and the location of others in the group

Member Functions

- manager ()
- map ()
- viewDidLoad ()
- didReceiveMemoryWarning ()

10.36.1 Method Summary

10.36.1.a manager ( )

The location manager that handles automatic location tracking and display on the map. - SeeAlso: CLLocationManager

Modifier and Type:
let manager

Declaration: Swift

let manager = CLLocationManager()

10.36.1.b map ( )

Outlet to Interface Builder for the map view - SeeAlso: MKMapView

Modifier and Type:
@IBOutlet weak var map:

Declaration: Swift

@IBOutlet weak var map: MKMapView!

10.36.1.c viewDidLoad ( )

Modifier and Type:
class MapViewController:

Declaration: Swift

class MapViewController: UIViewController, CLLocationManagerDelegate

10.36.1.d didReceiveMemoryWarning ( )

Modifier and Type:
class MapViewController:

Declaration: Swift

class MapViewController: UIViewController, CLLocationManagerDelegate

10.36.2 Generation

The documentation for this class was generated from the following file:

- MapViewController

10.37 ModelManager Class Reference

Class GroupModel

public class GroupModel
class PModelManager

A class dedicated for providing model functionality that does not belong in any individual model, such as logging users in, signing up new users, or getting the currrent user

Member Functions

- logInUser ()
- logInUserInBackground ()
- createUser ()
- createUserInBackground ()
- currentUser ()
- logOutCurrentUser ()
- fetchGroups ()
- fetchGroupsInBackground ()
- currentGroup ()
- setCurrentGroup ()
- fetchCurrentGroup ()
- fetchCurrentGroupInBackground ()

10.37.1 Method Summary

10.37.1.a logInUser ( )

logInUser(_:password:)

Compares the submitted username and password to storage and returns a UserModel object if the user was successfully logged in. Otherewise, it throws an exception.

Modifier and Type:
init

Declaration: Swift

func logInUser(username: String, password: String) throws -> UserModel

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.

Return Value: A user object if the login was successful.

10.37.1.b logInUserInBackground ( )

logInUserInBackground(_:password:callback:)

Compares the submitted username and password to storage on a separate thread, executing the given callback if successful. If login is unsuccessful, it throws an exception.

Modifier and Type:
init

Declaration: Swift

func logInUserInBackground(username: String, password: String, callback: ((user: UserModel?, error: NSError?) -> Void)?) -> Void

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.
- callback - An optional callback to call once the operation is complete.

10.37.1.c createUser ( )

createUser(_:email:password:)

Attempts to create a new user in the system, ensuring that the given username is unique. Also creates the corresponding user profile object. Both objects are then stored in the server.

This is a blocking function that can take several seconds to complete. If an operation fails, then an exception will be thrown.

Modifier and Type:
init

Declaration:
Swift

func createUser(username: String, email: String, password: String) throws -> UserModel

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.

Return Value:
The newly created UserModel if the operation is successful.

10.37.1.d createUserInBackground ( )

createUserInBackground(_:email:password:callback:)
Attempts to create a new user in the system, ensuring that the given username is unique. Also creates the corresponding UserProfileObject. Both objects are then stored in the server.

This is an asynchronous function that will pass control back to the main thread by executing the given callback prameter if it is not nil.

Modifier and Type:
init

Declaration:
Swift

func createUserInBackground(username: String, email: String, password: String, callback: ((user: UserModel?, error: NSError?) -> Void)?) -> Void

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.
- callback - The callback function that will be executed after the operation is complete, either successfully or unsuccessfully.

10.37.1.e currentUser ( )

Retrieves the currently logged-in user. If no user is logged in, returns nil.

Modifier and Type:
init

Declaration:
Swift

func createUserInBackground(username: String, email: String, password: String, callback: ((user: UserModel?, error: NSError?) -> Void)?) -> Void

10.37.1.f logOutCurrentUser ( )

Logs out the currently logged in user, removing them from any caches and returning whether or not the operation was successful.

Modifier and Type:
init

Declaration:
Swift

func logOutCurrentUser() -> Bool

Return Vales: true if the operation was successful and the user was successfully logged out, false if not.

10.37.1.g fetchGroupsInBackground ( )

etches all gropus in storage asynchronously.

This is an asynchronous function that will pass control back to the main thread by executing the given callback parameter if it is not nil.

Modifier and Type:
init

Declaration:
Swift

func fetchGroupsInBackground(callback: ((results: [GroupModel]?, error: NSError?) -> Void)?) -> Void

Return Vales: Array of all GroupModel objects in storage.

10.37.1.h fetchGroups ( )

fetchGroupsInBackground(_:) Fetches all groups in storage synchronously.

This is a blocking function that can take several seconds to complete. If an operation fails, then an exception will be thrown.

Modifier and Type:
init

Declaration:
Swift

func fetchGroups() throws -> [GroupModel]

Parameters:

- callback - The callback function that will be executed after the operation is complete, either successfully or unsuccessfully.

10.37.1.i currentGroup ( )

Fetches all groups in storage synchronously.

Gets the cached currently active group of which the current user is member, if any. If no user is logged in or the logged in user is not a member of any groups, this method will return nil. This function does not access storage in any way.

This method deals exclusively with cached values. In order to update the cached value, either set the cached value directly using setCurrentGroup(_:) or allowing it to be set automatically using fetchCurrentGroup() and fetchCurrentGroupInBackground(_:).

Modifier and Type:
init

Declaration:
Swift

func fetchGroups() throws -> [GroupModel]

Parameters:

- callback - The callback function that will be executed after the operation is complete, either successfully or unsuccessfully.

Returns Value:
The GroupModel of which the logged in user (if any) is a member of, or nil if no such group exists.

10.37.1.j setCurrentGroup ( )

setCurrentGroup(_:)
Sets the current cached value of the active group. Set the value to nil to indicate that there are no currently active groups. This function does not modify storage in anyway.

Modifier and Type:
init

Declaration:
Swift

func setCurrentGroup(group: GroupModel?)

Parameters:

- group - The current active GroupModel, or nil if no groups are currently active.

10.37.2 Generation

The documentation for this class was generated from the following file:

- ParseModelManager

10.38 ParseBaseModel Class Reference

Class ParseBaseMode

public class ParseBaseMode

class ParseBaseModel: BaseModel

The core Parse implementation for models. Implements the BaseModel protocol and provies functionality to access the basic information about the model, such as:

- **Unique identifier**
- **Creation timestamp**
- **Last updated timestamp**
- **Flag indicating modification since last save**
- **Methods for loading from and saving to storage**

Member Functions

- **parseObject ()**
- **init ()**
- **id ()**
- **created ()**
- **updated ()**
- **modified ()**
- **load ()**
- **loadInBackground ()**
- **save ()**
- **saveInBackground ()**

10.38.1 Method Summary

10.38.1.a parseObject ( )

Internal reference to the Parse API object representing this object in the remote database.

Modifier and Type:
let parseObject:

Declaration: Swift

```
let parseObject: PFObject
```

10.38.1.b init ( )

init(withParseObject:) Class constructor. Initializes the instance from a PFObject.

Modifier and Type:
withParseObject object:

Declaration: Swift

```
init(withParseObject object: PFObject)
```

Parameters:

- **withParseObject - The Parse object to tie this model to the Parse Parse database.**

10.38.1.c id ( )

init(withParseObject:) Read-only computed value representing the ID of the object as defined in the BaseModel protocol.

Modifier and Type:
var id:

Declaration: Swift

var id: String

10.38.1.d created ( )

init(withParseObject:) Read-only computed value for the timestamp when this object was initially created as defined in the BaseModel protocol.

Modifier and Type:
var created:

Declaration: Swift

var created: NSDate

10.38.1.e updated ( )

init(withParseObject:) Read-only computed value for the timestamp when this object was initially created as defined in the BaseModel protocol.

Modifier and Type:
var created:

Declaration: Swift

var created: NSDate

10.38.1.f modified ( )

init(withParseObject:) Read-only computed value indicating whether or not the data contained in this model is "dirty", which is to say that this model contains changes that have not been saved to the server.

Modifier and Type:
var modified:

Declaration: Swift

var modified: Bool

10.38.1.g load ( )

init(withParseObject:) Reloads this object from Parse as defined by the BaseModel protocol.

Modifier and Type:
func

Declaration: Swift

func load() throws

10.38.1.h loadInBackground ( )

init(withParseObject:) Reloads this object from Parse asynchronously as defined by the BaseModel protocol.

Modifier and Type:
func

Declaration: Swift

func loadInBackground(callback: ((object: BaseModel?, error: NSError?) -> Void)?)

10.38.1.i save ( )

init(withParseObject:) Reloads this object from Parse asynchronously as defined by the BaseModel protocol.

Modifier and Type:
func

Declaration: Swift

func save() throws

10.38.1.j saveInBackground ( )

init(withParseObject:) Saves this object to Parse as defined by the BaseModel protocol.

Modifier and Type:
func

Declaration: Swift

func saveInBackground(callback: ((object: BaseModel?, error: NSError?) -> Void)?)

10.38.2 Generation

The documentation for this class was generated from the following file:

- MapViewController

10.39 ParseGroupModel Class Reference

Class ParseGroupModel

public class ParseGroupModel

class ParseGroupModel: ParseBaseModel, GroupModel

The Parse implementation of the GroupModel protocol. Extends ParseBaseModel class and implements the GroupModel protocol and is designed to allow access to a group's information, including the group members, group name, and group description.

Member Functions

- init ()
- init ()
- generalLocation ()
- groupDescription ()
- groupName ()
- groupMembers ()
- groupLeader ()
- addGroupMember ()
- removeGroupMember ()
- load ()
- loadInBackground ()
- createGroup ()
- getAll ()
- getAllInBackground ()
- getGroupContainingUser ()
- getGroupContainingUserInBackground ()

10.39.1 Method Summary

10.39.1.a init ( )

Default class contructor. Creates a new entry in the database if saved.

Modifier and Type:
init

Declaration: Swift

init

10.39.1.b init ( )

init(withParseObject:)
Class constructor. Initializes the isntance from a PFObject.

Modifier and Type:
init

Declaration: Swift

init

Parameteres:

- withParseObject - The Parse object to tie this model to the Parse database.

10.39.1.c generalLocation ( )

PFGeoPoint Object to store the groups general location. This field is used for looking up groups as well as future support with finding ads by location

Modifier and Type:
var generalLocation:

Declaration: Swift

var generalLocation: PFGeoPoint

10.39.1.d groupDescription ( )

String to hold the description of the group.

Modifier and Type:
var groupDescription:

Declaration: Swift

var groupDescription: String

10.39.1.e groupName ( )

String to hold the Name of the group.

Modifier and Type:
var groupName:

Declaration: Swift

var groupName: String

10.39.1.f groupMembers ( )

An array of UserProfileModel objects to keep track of the members of the group.

Modifier and Type:
var groupMembers:

Declaration: Swift

var groupMembers: [UserProfileModel]

10.39.1.g groupLeader ( )

The UserProfileModel object who is the designated leader of the group.

Modifier and Type:
var groupMembers:

Declaration: Swift

var groupLeader: UserProfileModel?

10.39.1.h addGroupMember ( )

addGroupMember(_:)

Method for adding a user as a member of the group. Model must be saved afterwards, as this method does not automatically save the model.

Modifier and Type:
func addGroupMember

Declaration: Swift

func addGroupMember(member: UserProfileModel) -> Bool

10.39.1.i removeGroupMember ( )

Method for removing a user from the group. Model must be saved afterwards, as this method does not automatically save the model.

Modifier and Type:
func removeGroupMember

Declaration: Swift

func removeGroupMember(member: UserProfileModel) -> Bool

10.39.1.j load ( )

Loads this object from Parse storage synchronously. In addition to the normal functionality inherited from ParseBaseModel, this function also fetches and caches the users who are members of this group.

Modifier and Type:
override func

Declaration: Swift

override func load() throws

10.39.1.k loadInBackground ( )

loadInBackground(_:)
Loads this object from Parse storage asynchronously. In addition to the normal functionality inherited from ParseBaseModel, this function also fetches and caches the users who are members of this group.

Modifier and Type:
override override func

Declaration: Swift

override func loadInBackground(callback: ((object: BaseModel?, error: NSError?) -> Void)?)

10.39.1.l createGroup ( )

createGroup(_:description:)
Function to create a group if there is not one that exists

Modifier and Type:
override override func

Declaration: Swift

static func createGroup(name: String, description: String) -> ParseGroupModel

Parameters:

- name - String containing the group name
- description - String containing the description of the group

Return Value: Object of type ParseGroupModel that contains the information

10.39.1.m getAll ( )

Fetches all groups in storage synchronously.

This is a blocking function that can take several seconds to complete. If an operation fails, then an exception will be thrown.

Modifier and Type:
override static func

Declaration: Swift

static func getAll() throws -> [ParseGroupModel]

Return Value: Array of group models in storage.

10.39.1.n getAllInBackground ( )

Fetches all groups in storage synchronously.

Fetches all groups in Parse storage asynchronously, returning control to the main thread through the provided callback (if any).

Modifier and Type:
override static func

Declaration: Swift

static func getAllInBackground(callback: ((results: [ParseGroupModel]?, error: NSError?) -> Void)?)

Parameters:

- callback - Optional closure that will be called on completion or if an error is encountered.

10.39.1.o getGroupContainingUser ( )

Fetches the first group to which the provided user belongs, if any. If no group is found that contains the provided user, then nil is returned.

This is a blocking function and should be executed on a thread separate from the main thread. See getGroupContainingUserInBackground(_:callback:) for fetching on a separate thread. This function will throw an exception if an error occurs.

Modifier and Type:
override static func

Declaration: Swift

static func getGroupContainingUser(user: ParseUserProfileModel) throws -> ParseGroupModel?

Parameters:

- user - The user to search for.

Returns: Optional ParseGroupModel object that has user as a member, or nil if no such group could be found. This object will be fully loaded from storage such that a call to ParseGroupModel.load() is not necessary.

10.39.1.p getGroupContainingUserInBackground ( )

getGroupContainingUserInBackground(_:callback:)

Fetches the first group to which the provided user belongs, if any. If no group is found that contains the provided user, then nil is returned.

This function spawns a new thread for querying storage. After successful execution or if an error occurs, this function passes control back to the main thread by calling the closure that was optionally passed as an argument.

Modifier and Type:
override static func

Declaration: Swift

static func getGroupContainingUserInBackground(user: ParseUserProfileModel, callback: ((result: ParseGroupModel?, error: NSError?) -> Void)?)

Parameters:

- user - The user to search for.
- callback - Optional callback function to call after successful execution or if an error occurs. If nil is provided, then the callback will not be called.

Returns: Optional ParseGroupModel object that has user as a member, or nil if no such group could be found. This object will be fully loaded from storage such that a call to ParseGroupModel.load() is not necessary.

10.39.2 Generation

The documentation for this class was generated from the following file:

- ParseGroupModel

10.40 ParseModelManager Class Reference

Class ParseGroupModel

public class ParseGroupModel
class ParseModelManager: ModelManager

A Parse implementation of the ModelManager protocol. Class designed to query Parse for various models, including users and groups.

Member Functions

- logInUser ()
- logInUserInBackground ()
- createUser ()
- createUserInBackground ()
- currentUser ()
- logOutCurrentUser ()
- fetchGroups ()
- fetchGroupsInBackground ()
- currentGroup ()
- setCurrentGroup ()
- fetchCurrentGroup ()
- fetchCurrentGroupInBackground ()

10.40.1 Method Summary

10.40.1.a logInUser ( )

logInUser(_:password:)

Compares the submitted username and password to storage and returns a UserModel object if the user was successfully logged in. Otherewise, it throws an exception.

Modifier and Type:
init

Declaration: Swift

func logInUser(username: String, password: String) throws -> UserModel

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.

Return Value: A user object if the login was successful.

10.40.1.b logInUserInBackground ( )

logInUserInBackground(_:password:callback:)

Compares the submitted username and password to storage on a separate thread, executing the given callback if successful. If login is unsuccessful, it throws an exception.

Modifier and Type:
init

Declaration: Swift

func logInUserInBackground(username: String, password: String, callback: ((user: UserModel?, error: NSError?) -> Void)?) -> Void

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.
- callback - An optional callback to call once the operation is complete.

10.40.1.c createUser ( )

createUser(_:email:password:)

Attempts to create a new user in the system, ensuring that the given username is unique. Also creates the corresponding user profile object. Both objects are then stored in the server.

This is a blocking function that can take several seconds to complete. If an operation fails, then an exception will be thrown.

Modifier and Type:
init

Declaration:
Swift

func createUser(username: String, email: String, password: String) throws -> UserModel

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.

Return Value:
The newly created UserModel if the operation is successful.

10.40.1.d createUserInBackground ( )

createUserInBackground(_:email:password:callback:)

Attempts to create a new user in the system, ensuring that the given username is unique. Also creates the corresponding UserProfileObject. Both objects are then stored in the server.

This is an asynchronous function that will pass control back to the main thread by executing the given callback prameter if it is not nil.

Modifier and Type:
init

Declaration:
Swift

func createUserInBackground(username: String, email: String, password: String, callback: ((user: UserModel?, error: NSError?) -> Void)?) -> Void

Parameters:

- username - The username of the user to log in.
- password - The password to use to log in.
- callback - The callback function that will be executed after the operation is complete, either successfully or unsuccessfully.

10.40.1.e currentUser ( )

Retrieves the currently logged-in user. If no user is logged in, returns nil.

Modifier and Type:
init

Declaration:
Swift

func createUserInBackground(username: String, email: String, password: String, callback: ((user: UserModel?, error: NSError?) -> Void)?) -> Void

10.40.1.f logOutCurrentUser ( )

Logs out the currently logged in user, removing them from any caches and returning whether or not the operation was successful.

Modifier and Type:
init

Declaration:
Swift

func logOutCurrentUser() -> Bool

Return Vales: true if the operation was successful and the user was successfully logged out, false if not.

10.40.1.g fetchGroupsInBackground ( )

etches all gropus in storage asynchronously.

This is an asynchronous function that will pass control back to the main thread by executing the given callback parameter if it is not nil.

Modifier and Type:
init

Declaration:
Swift

func fetchGroupsInBackground(callback: ((results: [GroupModel]?, error: NSError?) -> Void)?) -> Void

Return Vales: Array of all GroupModel objects in storage.

10.40.1.h fetchGroups ( )

fetchGroupsInBackground(_:) Fetches all groups in storage synchronously.

This is a blocking function that can take several seconds to complete. If an operation fails, then an exception will be thrown.

Modifier and Type:
init

Declaration:
Swift

func fetchGroups() throws -> [GroupModel]

Parameters:

- callback - The callback function that will be executed after the operation is complete, either successfully or unsuccessfully.

10.40.1.i currentGroup ( )

Fetches all groups in storage synchronously.

Gets the cached currently active group of which the current user is member, if any. If no user is logged in or the logged in user is not a member of any groups, this method will return nil. This function does not access storage in any way.

This method deals exclusively with cached values. In order to update the cached value, either set the cached value directly using setCurrentGroup(_:) or allowing it to be set automatically using fetchCurrentGroup() and fetchCurrentGroupInBackground(_:).

Modifier and Type:
init

Declaration:
Swift

func fetchGroups() throws -> [GroupModel]

Parameters:

- callback - The callback function that will be executed after the operation is complete, either successfully or unsuccessfully.

Returns Value:
The GroupModel of which the logged in user (if any) is a member of, or nil if no such group exists.

10.40.1.j setCurrentGroup ( )

setCurrentGroup(_:)
Sets the current cached value of the active group. Set the value to nil to indicate that there are no currently active groups. This function does not modify storage in anyway.

Modifier and Type:
init

Declaration:
Swift

func setCurrentGroup(group: GroupModel?)

Parameters:

- group - The current active GroupModel, or nil if no groups are currently active.

10.40.2 Generation

The documentation for this class was generated from the following file:

- ParseModelManager

10.41 ParseUserModel Class Reference

Class ParseUserModel

public class ParseUserModel
class ParseUserModel: ParseBaseModel, UserModel
This class extends the ParseBaseModel class and implements the UserModel protocol and is the class to access the current user's information from Parse

Member Functions

- profile ()
- init ()
- username ()
- emailVerified ()
- email ()
- phone ()
- createFromSignUp ()

10.41.1 Method Summary

10.41.1.a profile ( )

The model corresponding to this user's public profile.

Modifier and Type:
init

Declaration: Swift

let profile: UserProfileModel

10.41.1.b init ( )

init(withParseUser:profile:)
Class constructor. Initializes the instance from a PFObject.

Modifier and Type:
init

Declaration: Swift

init(withParseUser user: PFUser, profile: UserProfileModel? = nil)

Parameters:

- withParseUser - The Parse user to tie this model to the Parse database.
- profile - An optional UserProfileModel to connect to this user. If no value is or nil is provided, this constructor will attempt to get the profile from the withParseUser model.

10.41.1.c username ( )

String containing the current user's username as defined in the UserModel protocol.

Modifier and Type:
var username:

Declaration: Swift

var username: String

10.41.1.d emailVerified ( )

Boolean to store if the user has verified their email with parse as defined by the UserModel protocol.

Modifier and Type:
var emailVerified:

Declaration: Swift

var emailVerified: Bool

Return Value:
true if their email has been verified, false if their email has not been verified.

10.41.1.e email ( )

String containing the current users email as defined in the UserModel protocol.

Modifier and Type:
var emailVerified:

Declaration: Swift

var email: String

10.41.1.f createFromSignUp ( )

createFromSignUp(_:password:)

Main function for creating a new user. Automatically creates a corresponding UserProfileModel and returns both in a tuple.

Modifier and Type:
var phone:

Declaration: Swift

var phone: String

10.41.1.g createFromSignUp ( )

createFromSignUp(_:password:)

Main function for creating a new user. Automatically creates a corresponding UserProfileModel and returns both in a tuple.

Modifier and Type:
var phone:

Declaration:
Swift

var phone: String

Parameters:

- username - The new user's username.
- password - The new user's password.

Declaration:
A tuple containing the newly created UserModel object and its corresponding UserProfileModel.

10.41.2 Generation

The documentation for this class was generated from the following file:

- ParseUserModel

10.42 ParseUserProfileModel Class Reference

Class ParseUserProfileModel

public class ParseUserProfileModel

class ParseUserProfileModel: ParseBaseModel, UserModel

This class extends the ParseBaseModel class and implements the UserProfileModel protocol and is the class to access a user's public profile information from Parse.

Member Functions

- init ()
- init ()
- displayName ()

10.42.1 Method Summary

10.42.1.a init ( )

Default class contructor. Creates a new entry in the database if saved.

Modifier and Type:
init

Declaration: Swift

init()

10.42.1.b init ( )

init(withParseObject:)

Default class contructor. Creates a new entry in the database if saved.

Modifier and Type:
init

Declaration: Swift

override init(withParseObject object: PFObject)

Parameters:

- withParseObject - The Parse object to tie this model to the Parse database.

10.42.1.c displayName ( )

String containing a user's display name as defined in the UserProfileModel protocol.

Modifier and Type:
var displayName

Declaration: Swift

var displayName: String

Parameters:

- withParseObject - The Parse object to tie this model to the Parse database.

10.42.2 Generation

The documentation for this class was generated from the following file:

- ParseUserProfileModel

10.43 SettingsViewController Class Reference

Class SettingsViewController

public class SettingsViewController

Member Functions

- logoutButton ()
- onLogoutTapped ()

10.43.1 Method Summary

10.43.1.a logoutButton ( )

10.43.1.b onLogoutTapped ( )

10.43.2 Generation

The documentation for this class was generated from the following file:

- SettingsViewController

10.44 SignupViewController Class Reference

Class SignupViewController

public class SignupViewController
class SignupViewController: UIViewController, UITextFieldDelegate
Custom view controller class for handling user signups.

Custom UIViewController class for handling user signups. Processes requests, validates input, and passes data to backend for checking with the server, and properly handles both successful and unsuccessful signup requests.

Member Functions

- scrollView ()
- facebookButton ()
- twitterButton ()
- emailButton ()
- nameField ()
- emailField ()

- **passwordField ()**
- **passwordConfirmField ()**
- **submitButton ()**
- **signupFormFields ()**
- **init ()**
- **deinit ()**
- **viewDidLoad ()**
- **viewDidAppear ()**
- **facebookButtonTapped ()**
- **twitterButtonTapped ()**
- **submitButtonTapped ()**
- **registerForKeyboardNotifications ()**
- **deregisterFromKeyboardNotifications ()**
- **adjustForKeyboard ()**
- **textFieldShouldReturn ()**
- **submitForm ()**
- **unwindIfLoggedIn ()**
- **rewindToWelcomeView ()**

10.44.1 Method Summary

10.44.1.a scrollView ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.b facebookButton ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.c twitterButton ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.d emailButton ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.e nameField ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.f emailField ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.g passwordField ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.h passwordConfirmField ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.i submitButton ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.j signupFormFields ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.k init ( )

init(coder:)

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.l deinit ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.m viewDidLoad ( )

Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.n viewDidAppear ( )

viewDidAppear(_:)
Modifier and Type:
class SignupViewController:

Declaration: **Swift**

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.o facebookButtonTapped ( )

facebookButtonTapped(_:)
Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.p twitterButtonTapped ( )

twitterButtonTapped(_:)

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.q submitButtonTapped ( )

submitButtonTapped(_:)

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.r registerForKeyboardNotifications ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.s deregisterFromKeyboardNotifications ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.t adjustForKeyboard ( )

adjustForKeyboard(_:)

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.u textFieldShouldReturn ( )

textFieldShouldReturn(_:)

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.v submitForm ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.w unwindIfLoggedIn ( )

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.1.x rewindToWelcomeView ( )

rewindToWelcomeView(_:)

Modifier and Type:
class SignupViewController:

Declaration: Swift

class SignupViewController: UIViewController, UITextFieldDelegate

10.44.2 Generation

The documentation for this class was generated from the following file:

- ParseUserProfileModel

10.45 UserModel Class Reference

Class UserModel

public class UserModel

class UserModel: BaseModel, UserModel

This class extends the BaseModel class and implements the UserModel protocol and is the class to access the current user's information from

Member Functions

- profile ()
- init ()
- username ()
- emailVerified ()
- email ()
- phone ()
- createFromSignUp ()

10.45.1 Method Summary

10.45.1.a profile ( )

The model corresponding to this user's public profile.

Modifier and Type:
init

Declaration: Swift

let profile: UserProfileModel

10.45.1.b init ( )

init(withUser:profile:)
Class constructor. Initializes the instance from a PFObject.

Modifier and Type:
init

Declaration: Swift

init(withUser user: PFUser, profile: UserProfileModel? = nil)

Parameters:

- withUser - The user to tie this model to the database.
- profile - An optional UserProfileModel to connect to this user. If no value is or nil is provided, this constructor will attempt to get the profile from the withUser model.

10.45.1.c username ( )

String containing the current user's username as defined in the UserModel protocol.

Modifier and Type:
var username:

Declaration: Swift

var username: String

10.45.1.d emailVerified ( )

Boolean to store if the user has verified their email with as defined by the UserModel protocol.

Modifier and Type:
var emailVerified:

Declaration: Swift

var emailVerified: Bool

Return Value:
true if their email has been verified, false if their email has not been verified.

10.45.1.e email ( )

String containing the current users email as defined in the UserModel protocol.

Modifier and Type:
var emailVerified:

Declaration: Swift

var email: String

10.45.1.f createFromSignUp ( )

createFromSignUp(_:password:)

Main function for creating a new user. Automatically creates a corresponding UserProfileModel and returns both in a tuple.

Modifier and Type:
var phone:

Declaration: Swift

var phone: String

10.45.1.g createFromSignUp ( )

createFromSignUp(_:password:)

Main function for creating a new user. Automatically creates a corresponding UserProfileModel and returns both in a tuple.

Modifier and Type:
var phone:

Declaration:
Swift

var phone: String

Parameters:

- username - The new user's username.
- password - The new user's password.

Declaration:
A tuple containing the newly created UserModel object and its corresponding UserProfileModel.

10.45.2 Generation

The documentation for this class was generated from the following file:

- UserModel

s

10.46 UserProfileModel Class Reference

Class UserProfileModel

public class UserProfileModel
class UserProfileModel: BaseModel, UserModel
This class extends the BaseModel class and implements the UserProfileModel protocol and is the class to access a user's public profile information from .

Member Functions

- init ()
- init ()
- displayName ()

10.46.1 Method Summary

10.46.1.a init ( )

Default class contructor. Creates a new entry in the database if saved.

Modifier and Type:
init

Declaration: Swift

```
init()
```

10.46.1.b init ( )

init(withObject:)

Default class contructor. Creates a new entry in the database if saved.

Modifier and Type:
init

Declaration: Swift

```
override init(withObject object: PFObject)
```

Parameters:

- withObject - The object to tie this model to the database.

10.46.1.c displayName ( )

String containing a user's display name as defined in the UserProfileModel protocol.

Modifier and Type:
var displayName

Declaration: Swift

```
var displayName: String
```

Parameters:

- withObject - The object to tie this model to the database.

10.46.2 Generation

The documentation for this class was generated from the following file:

- UserProfileModel

10.47 Waypoint Struct Reference

Struct Waypoint

Struct Waypoint

class Waypoint: BaseModell

Object to store group waypoints location and message

Member Functions

- waypointId ()
- longitude ()
- latitude ()
- message ()

10.47.1 Method Summary

10.47.1.a waypointId ( )

Waypoint's unique id to be generated on creation

Modifier and Type:
int

Declaration: Swift

```
var waypointId: Int
```

10.47.1.b longitude ( )

Waypoint longitude

Modifier and Type:
double

Declaration: Swift

```
var longitude: Double
```

10.47.1.c latitude ( )

Waypoint longitude

Modifier and Type:
double

Declaration: Swift

```
var latitude: Double
```

10.47.1.d message ( )

Message to display at location

Modifier and Type:
String

Declaration: **Swift**

var message: String

10.47.2 Generation

The documentation for this class was generated from the following file:

- WayPoint

10.48 Maint Class Reference

Class main

public class main

Cloud functions that allow for background functions with Parse.

Member Functions

- fetchGroupUpdates ()
- joinGroup ()
- leaveGroup ()
- fetchNotifications ()

10.48.1 Method Detail

10.48.1.a fetchGroupUpdates ( )

Parse.Cloud.define('fetchGroupUpdates', function(request, response)

This function allows for group updates.

Parameters:

- group - current group of the user.
- userProfile - current user calling the function
- timestamp - current timestamp

Returns: changes in the group status.

10.48.1.b joinGroup ( )

parse.Cloud.define('joinGroup', function(request, response)

This function allows for an user to join a group.

Parameters:

- group - current group of the user.
- userProfile - current user calling the function

Returns: user join group and group information.

10.48.1.c leaveGroup ( )

parse.Cloud.define('leaveGroup', function(request, response)

This function allows for an user to leave a group.

Parameters:

- group - current group of the user.
- userProfile - current user calling the function

Returns: removes the user from group information

10.48.1.d fetchNotifications ( )

parse.Cloud.define('IfetchNotifications', function(request, response)

This function fetches notification ipdates for the group.

Returns: new notifications for current group

10.48.2 Generation

The documentation for this class was generated from the following file:

- main

11

Business Plan

Crowd Control

Business Plan

BowTaps, LLC

Charles Bonn:	nick.bonn@bowtaps.com
Johnathan Ackerman:	johnny.ackerman@bowtaps.com
Daniel Andrus:	dan.andrus@bowtaps.com
Evan Hammer:	evan.hammer@bowtaps.com
Joseph Mowry:	joe.mowry@bowtaps.com

%sectionMetrics and Milestones

SDSMT SENIOR DESIGN
SOFTWARE DEVELOPMENT AGREEMENT

This Software Development Agreement (the "Agreement") is made between the SDSMT Computer Science

Senior Design Team: CrowdControl
("Student Group")

consisting of team members: Charles Bonn, Evan Hammer, Joseph Mowry, Daniel Andrus, Johnathan Ackerman,
("Student Names")

and Sponsor: Bowtaps (self),
("Company Name")

with address: 2326 Lance Street, Rapid City , SD 57702.

1 RECITALS

1. The Bowtaps team will be designing, implimenting, and distributing CrowdControl under the SDSMT Senior Design program.

NOW, THEREFORE, in consideration of the mutual covenants and promises herein contained, Bowtaps and Brian Butterfeild agree as follows:

2 EFFECTIVE DATE

This Agreement shall be effective as of 9/30/2015.

3 DEFINITIONS

1. "Software" shall mean the computer programs in machine readable object code and any subsequent error corrections or updates created by Bowtaps for CrowdControl pursuant to this Agreement.
2. "Acceptance Criteria" means the written technical and operational performance and functional criteria and documentation standards set out in the backlog.
3. "Acceptance Date" means the date for each Milestone when all Deliverables included in that Milestone have been accepted by BowTaps under the supervison of Brian Butterfeild in accordance with the Acceptance Criteria and this Agreement.
4. "Deliverable" means the product requirements specified in the backlog under the acceptance date.
5. "Delivery Date" shall mean, with respect to a particular sprint, the date on which BowTaps will evaluate all of the Deliverables for that sprint in accordance with the backlog and this Agreement.
6. "Documentation" means the documents, manuals and written materials (including end-user manuals) referenced, indicated or described in the project plan or otherwise developed pursuant to this Agreement.
7. "Milestone" means the completion and delivery of all of the Deliverables or other events which are included or described in backlog scheduled for developement and/or completion on a given target date; a Milestone will not be considered completed until the Acceptance Date has occurred with respect to all of the Deliverables for that Milestone.

4 DEVELOPMENT OF SOFTWARE

1. The BowTaps Team will use its best efforts to develop the Software described in backlog The Software development will be under the direction of Its members with the supervision of Brian Butterfeild. BowTaps will deliver the Software to the satisfaction of the course instructor that reasonable effort has been made to design and release CrowdControl as a mobile application. The Team understands that failure to deliver the Software is grounds for failing the course.
2. Brian Butterfeild understands that the Senior Design course's mission is education and advancement of knowledge, and, consequently, the development of Software must further that mission. The Senior Design Course does not guarantee specific results or any results, and the Software will be developed only on a best efforts basis. The Software created will be intened as a beta release for future refinement before the release of CrowdControl.
3. The Senior Design instructor will act as mediator for BowTaps to help guide twords a start up software engineering company

5 COMPENSATION

NONE. This is a company start up with the goals of releasing a mobile application and starting a software developement company.

6 CONSULTATION AND REPORTS

1. Sponsor's designated representative for consultation and communications with the BowTaps team shall be

   __________ Brian Butterfeild __________ or such other person as consultant(s) may from time to time designate to the BowTaps team.
2. During the Term of the Agreement, consultant's representatives may consult informally with course instructor regarding the project, both personally and by telephone. Access to work carried on in University facilities, if any, in the course of this Agreement shall be entirely under the control of University personnel but shall be made available on a reasonable basis.
3. BowTaps will submit written progress reports. At the conclusion of this Agreement, the BowTaps team shall submit a comprehensive final report in the form of the formal course documentation at the conclusion of the Senior Design II course.

7 CONFIDENTIAL INFORMATION

1. The parties may wish, from time to time, in connection with work contemplated under this Agreement, to disclose confidential information to each other ("Confidential Information"). Each party will use reasonable efforts to prevent the disclosure of any of the other party's Confidential Information to third parties for a period of three (3) years after the termination of this Agreement, provided that the recipient party's obligation shall not apply to information that:
   (a) is not disclosed in writing or reduced to writing and so marked with an appropriate confidentiality legend within thirty (30) days of disclosure;
   (b) is already in the recipient party's possession at the time of disclosure thereof;
   (c) is or later becomes part of the public domain through no fault of the recipient party;
   (d) is received from a third party having no obligations of confidentiality to the disclosing party;

   (e) is independently developed by the recipient party; or
   (f) is required by law or regulation to be disclosed.

2. In the event that information is required to be disclosed pursuant to subsection (6), the party required to make disclosure shall notify the other to allow that party to assert whatever exclusions or exemptions may be available to it under such law or regulation.

8 INTELLECTUAL PROPERTY RIGHTS

Intelectual Property created durind the development, testing, deployment, and updating of CrowdControl. Intelectual Property consists of any documents drafted, products designed, and code written and implimented by BowTaps The Intelectual Property belongs to the developement team, BowTaps, under the direction and guidance of SDSM&T and consultants.

9 WARRANTIES

The BowTaps Team represents and warrants to Sponsor that:

1. the Software is the original work of the BowTaps Team in each and all aspects;
2. the Software and its use do not infringe any copyright or trade secret rights of any third party.

No agreements will be made beyond items (1) and (2).

10 INDEMNITY

1. BowTaps is responsible for claims and damages, losses or expenses held against the BowTaps team.
2. NEITHER PARTY TO THIS AGREEMENT NOR THEIR AFFILIATED COMPANIES, NOR THE OFFICERS, AGENTS, STUDENTS AND EMPLOYEES OF ANY OF THE FOREGOING, SHALL BE LIABLE TO ANY OTHER PARTY HERETO IN ANY ACTION OR CLAIM FOR CONSEQUENTIAL OR SPECIAL DAMAGES, LOSS OF PROFITS, LOSS OF OPPORTUNITY, LOSS OF PRODUCT OR LOSS OF USE, WHETHER THE ACTION IN WHICH RECOVERY OF DAMAGES IS SOUGHT IS BASED ON CONTRACT TORT (INCLUDING SOLE, CONCURRENT OR OTHER NEGLIGENCE AND STRICT LIABILITY), STATUTE OR OTHERWISE. TO THE EXTENT PERMITTED BY LAW, ANY STATUTORY REMEDIES WHICH ARE INCONSISTENT WITH THE PROVISIONS OF THESE TERMS ARE WAIVED.

11 INDEPENDENT CONTRACTOR

For the purposes of this Agreement and all services to be provided hereunder, the parties shall be, and shall be deemed to be, independent contractors and not agents or employees of the other party. Neither party shall have authority to make any statements, representations or commitments of any kind, or to take any action which shall be binding on the other party, except as may be expressly provided for herein or authorized in writing.

12 TERM AND TERMINATION

1. This Agreement shall commence on the Effective Date and extend until the end of classes of the second semester of Senior Design (CSC 467), unless sooner terminated in accordance with the provisions of this Section ("Term").
2. This Agreement may be terminated by the written agreement of both parties.
3. In the event that either party shall be in default of its materials obligations under this Agreement and shall fail to remedy such default within thirty (30) days after receipt of written notice thereof, this Agreement shall terminate upon expiration of the thirty (30) day period.
4. Any provisions of this Agreement which by their nature extend beyond termination shall survive such termination.

13 GENERAL

1. This Agreement constitutes the entire and only agreement between the parties relating to the Senior Design Course, and all prior negotiations, representations, agreements and understandings are superseded hereby. No agreements altering or supplementing the terms hereof may be made except by means of a written document signed by the duly authorized representatives of the parties.
2. This Agreement shall be governed by, construed, and enforced in accordance with the internal laws of the State of South Dakota.

14 SIGNATURES

[illegible]	10 / 6 / 2015
Charles Bonn	Date
[illegible]	10 / 6 / 2015
Evan Hammer	Date
[illegible]	10 / 6 / 2015
Joseph Mowry	Date
[illegible]	10 / 6 / 2015
Daniel Andrus	Date
[illegible]	10 / 6 / 2015
Johnathan Ackerman	Date
[illegible]	10 / 6 / 2015
Brian Butterfeild	Date

A

Product Description

CrowdControl is a group management application that will be an application that has gps features, group messaging, group management features.

1 GPS Features

1.1 Group Members

The Group member gps features will allow for users to track other users in the same group as they are. This will be under user permission to allow other user to see there location.

1.2 Suggestions

The suggestion side of the GPS will take a user or group location and give even suggestions of places to go or things to do in the area of the group.

2 Group Messaging

Integrated group messaging on a single platform uniform to iOS and android.

3 Group Manangement Features

This will allow for members to join a group, add a member to a group, and leave a group.

4 Parse Features

Parse will be used to store user data and group data.

B

Sprint Reports

Sprint Report #1

Team Overview

Name

Crowd Control

Members

Johnathan Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, Joseph Mowry

Project Title

Crowd Control Group Management Mobile Application

Company

Bowtaps

Customer Overview

Customer Description

Bowtaps is a start up company based out of Rapid City, SD. Bowtaps plans on having their initial market presence with the mobile application Crowd Control.

Customer Problem

The design, creation, and marketing of the mobile application Crowd Control along with the creation of the company Bowtaps.

Customer

- GPS mapping of Members in the group
- Integrated group messaging
- Group management features (add/remove members)
- Intuitive UI
- Product testing
- Marketing plan and strategies
- Business plan
- End-user Documentation

Project Overview

The creation of Crowd Control, a mobile application on Android and iOS platforms for group management.

Phase 1

The design of the database and the basic design of the user interface.

Project Environment

Project Boundaries

- Crowd Control will be a free app available for download on the Android and iOS marketplaces.
- The product will be coded in Java (Android) , swift (iOS), and parse (back-end server).
- Source code will be kept in a private GitHub repository.
- Crowd Control will be planned on release by summer of 2016.

Project Context

- There will be 2 versions of the application (one for iOS and one for Android)
- Crowd Control will access a parse server
- Crowd Control will access GPS information

Deliverables

Phase 1

Deliverables will be UX design, database design and implementation.

Backlog

Phase 1

- Design UX
    1. Create groups
    2. Leave groups
    3. Group messaging
    4. Start page
- Database

    1. Design database schema
    2. Implement database on Parse

- Design application layers (MVC)
- Set up GitHub repository

Sprint Report

Work for this sprint included:

- Designs for Create Group page
- Design for Leave Group page
- Design for Group Messaging page
- Design for Start Page
- Design for Database Schema
- Database implementation
- Git Repository Initialization

Sprint Report #2

Team Overview

Name

Crowd Control

Members

Johnathan Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, Joseph Mowry

Project Title

Crowd Control Group Management Mobile Application

Company

Bowtaps

Work Summary

- Code UX
    1. Map Screen
    2. Group info Screen
    3. Group Messaging
    4. Start page
    5. Group Info UI
- Model
    1. User Model
    2. Communication layer
- Research on public/private key passing

Backlog

- Code UX
    1. Mapping features
    2. Messaging UI
- Model

    1. User Model
    2. Communication Layer
    3. Link back-end and front end

- Implement Cloud code
- Business Plan

Successes

Successes have been jumps in the code progress. Testing has been going well and progress has been made towards the end goal.

Issues and Changes

Some issues that have been ran into have been

- Public/Private key passing for increased security
- Differences between iOS and android coding standards not allowing for similar looks between operating systems.
- Testing of mapping features

Team Details

The team is going strong. With a busy semester, not all meeting times have worked out. But with a hard drive, we are working towards our goal of creating an app and starting our own business. We are still currently meeting with advisors to better our business plan and create marketing plans.

Sprint Report #3

Team Overview

Name

Crowd Control

Members

Johnathan Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, Joseph Mowry

Project Title

Crowd Control Group Management Mobile Application

Company

Bowtaps

Work Summary

- iOS
    1. Login
        (a) Create User
        (b) Facebook integration
    2. Mapping
    3. Working on Join Group
- Android
    1. Login
        (a) Create User
        (b) Facebook integration
    2. Mapping
    3. Working on Join Group
- Server
    1. Fixed Connection Issues
    2. User Connections Created

Backlog

- Messaging API
- Join Group Implementation
- Cloud Code
  1. Group Clean Up
  2. User Information Links
- Business Plan
  1. South Dakota Giant Vision
  2. SDSM&T Business Plan Competition

Success

Successes have been group team work towards the business plan competitions on the business side. On the development side was recreating some of the database to increase efficiency with parse. Logging in has been connected to Facebook accounts.

Issues and Changes

Some issues that have been ran into have been

- Public/Private key passing for increased security
- Server connection issues from table to table with group creation
- Changes in the database schema
- GUI updates to more modern standards.

Team Details

With business plan competitions, and the end of the semester, we have all been busy. We have come together to fix issues that where not planned for in the beginning, and furthered development of features in general.

The business plan and business plan competition are coming along well, and allowing us to focus more on the primary goals of the direction of the company, as well as development of Crowd Control.

Winter Sprint Report

Team Overview

Name

Crowd Control

Members

Johnathan Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, Joseph Mowry

Project Title

CrowdControl - Group Management Mobile Application

Company

Bowtaps

Deliverables

- iOS
    1. Login/Logout
        (a) Improved login/signup screens
        (b) Logout feature added
    2. Settings
        (a) Settings screen implemented
        (b) Logout functionality nested in the Settings screen
    3. Groups
        (a) Leaving/Joining a group implemented
        (b) Basic group operations
        (c) Detect if users are in a group
- Android
    1. Login
        (a) Automatic login on startup (from datastore)
        (b) Login to existing account via email address
    2. Settings
        (a) Page layout created and linked from GroupJoin page
        (b) Logout functionality implemented
    3. Groups

    (a) Leave button implemented
    (b) Tested adding/removing users from groups

- Misc/Transitional
    1. Further documented Android code to prepare for team merge
    2. Android code review with iOS team, to prepare for team merge

Remaining Backlog

Here are the incomplete items/features for this sprint:

- Android
    - Messaging (Sinch API)
    - GPS Location (backend models)
    - Persistent groups through local datastore
- iOS
    - Messaging (Sinch API)

Successes

- Android
    - Login through email
    - Settings page (layout and implementation)
    - Local Datastore (individual automatic login)
- iOS
    - Login/Logout
    - Settings page (layout and implementation)
    - Group functionality written

Issues and Changes

Some issues that we encountered include:

- Android
    - Issues
        * Tried to manually create queries in the Parse API. We were unaware of built-in methods to accomplish the tasks. This set us back a bit.

        - Encountered NullPointerException in the UserModel model. Had to change the structure to use an application global variable.
    - Changes
        - Further development on Settings is now added to the backlog
        - Sign out functionality is now added to the backlog
        - Leave Group functionality is now added to the backlog

- iOS
    - Issues
        - Unexpected complications with database design
        - Layout complications
        - Issues with the underlying data models
        - Parallel programming complications

- Misc/Transitional
    - iOS development will be postponed, in favor of an Android prototype. This is to ensure that Android will meet expectations for the design fair.
    - Team communication and long-distance coordination was difficult.
    - Holidays and vacations imepeded our ability to be productive.

Team Details

Our team fell behind in the first semester, and in an effort to mitigate this, we allocated work towards the Winter Sprint. From here, unsatisfactory progress was still met, and we decided on another large refactor.

For the remainder of our project development, the iOS team will halt development and assist the Android team, so that Bowtaps can guarantee a satisfactory product for the design fair in Spring 2016.

Finally, to hopefully achieve better group management, we have elected Daniel Andrus to serve as acting Scrum Master.

Sprint Report #4

Team Overview

Name

Crowd Control

Members

Johnathan Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, Joseph Mowry

Project Title

Crowd Control - Group Management Mobile Application

Company

Bowtaps

Backlog

The following items/features were assigned at the beginning of the sprint, and worked on throughout its duration. It is broken down by week as such:

Week 1

- Android
  - Begin implementing Sinch
  - Create location and messaging views and managers
  - Design models and manager classes for messaging and location
  -
- Cloud Code
  - Group data parsing started

Week 2

- Android
  - Broadcast/receive messages to/from all members in a group
  - Create a layout for messaging
  - Create a MapFragment to display a map
  - Created buttons overtop the MapFragment to correspond to syncing and homing locations
- Cloud code
  - Leaving and joining groups handled
  - Checking existing email upon login (validation)

Week 3

- Android
  - Retrieve locations of group members, place their locations on the map via pins
  - Update group settings and data when changed
  - Update Group members if someone leaves or joins a group
  - Group messaging unit tests
  - GPS Location unit tests
- Cloud Code
  - Returning group information upon changes
  - Functional Group update indicator complete
  - Basic group functionality implemented fully (login/logout, join/leave groups, update on change)

Documentation and Business Plan work was carried out through all weeks of the sprint, and is ongoing.

Deliverables

During this sprint, these are the items/features from the backlog that were successfully achieved:

- Android
  1. Group Messaging
     (a) Created a Layout
     (b) Used Sinch code to create a service
     (c) Implemented group messaging
     (d) Group messaging is working with no known bugs
  2. Location
     (a) Page layout created and linked from GroupJoin page
     (b) MapFragment has buttons for homing and syncing group locations
     (c) Retrieving the user's location on instantiation of the MapFragment
     (d) User and group locations implemented
  3. Group update service
     (a) Checks for updates in near real-time
     (b) Updates group settings when changed
     (c) Updates group members if someone leaves or joins
- Server (cloud code)
  1. Functional Group update indicator
  2. Returning group update information
  3. Join group function (created but not functioning)
  4. Leave group function (created but not functioning)

5. Check for Existing Email

- Misc/Transitional
  1. Business Plan filled out, also a version tailored towards the Governor's Giant Vision contest (converted to latex)
  2. Documentation done inside and outside of the source code files

Issues and Changes

Some issues that we encountered include:

- Android
  - Issues
    * Permissions to obtain contacts and locations from the device posed a challenge - still not handling the request gracefully
    * Had difficulty implementing a custom AlertDialogFragment that extends DialogFragment, inside of other fragments such as MapFragment, GroupInfoFragment, etc.
  - Changes
    * Added group update service - was not part of original backlog
    * Added user location homing on MapFragment - was not part of original backlog
- Server (cloud code)
  - Issues
    * Cloud functions improperly writing data.
  - Changes
    * Added join and leave cloud functions - was not part of original backlog

Remaining Backlog

The following items/features remain either incomplete or need improvement for this sprint, and will carry onto the next sprint:

- Android
  - Group messaging unit tests
  - GPS Location unit tests
- Cloud Code
  - Testing on Group Functions.
  - Completing Join and Leave functions

Team Details

Here are some auxiliary details about our workflow and division of responsibilities, during the sprint:

Dan and Johnny started off Sprint 4 by working on messaging-related features, while Joe and Evan worked on GPS and map features. Nick focused on the business plan, updating the existing documentation to use the updated layout, and was tasked with installing the Fabric SDK.

For week two of the sprint, Johnny focused on group messaging. Dan and Nick also worked together on cloud code, targeting the leaving/joining groups, and a group update service in Android. Evan wrote a LocationManager class and stubbed out methods, that Joe wrote an interface for and made a UI for in the MapFragment.

Week three continued with Joe and Evan working on various location features, while Dan and Johnny worked on the update service and messaging features respectively. Nick focused on cloud code and business plan/documentation writing.

Additionally, this was the first sprint in which we had Dan serve as acting Scrum Master, to aid in organization and appointment of responsibilities. Though he did officially take this role on during our Winter Sprint (Sprint 3.5), most of us were either working remotely or unavailable, thus we were unable to fully utilize this new organizational change until now.

C

Industrial Experience and Resumes

1 Resumes

Below are the resumes for the group members: Johnathon Ackerman, Daniel Andrus, Charles Bonn, Evan Hammer, and Joseph Mowry.

Johnathan Ackerman

605-877-1757
Johnathan.ackerman@mines.sdsmt.edu
GitHub profile https://github.com/Kiwii12

Education

South Dakota School of Mines and Technology

- **Computer Science Major**
- Start Date: Fall 2012
- Expected Graduation Date: **December 2016**
- Going for a Bachelor's Degree
- Enrolled Currently as a Senior

Central High School

- Graduated 2012

Programs

Team Projects

With Glut and C++, in teams of two, I have made the following:

-Pong (https://github.com/Kiwii12/CSC433_Program1_Pong)
-Solar System Model (https://github.com/Kiwii12/CSC433_Program3_SolarSystem)

In C++

-Simulated a B17 computer (https://github.com/Kiwii12/B17)

In Lisp

-Missionary Vs Cannibals (https://github.com/Kiwii12/missionaryVsCannibal)

Solo Projects

In C++

-WVX playlist creator (https://github.com/Kiwii12/WVX-Playlist-Creator)
-Basic Picture Editor (https://github.com/Kiwii12/Basic_Picture_Editor)

Skills

I have worked in the Operating Systems of Windows and Linux (Fedora and Ubunto)
I am very comfortable in **C++** and **Python.**
I am comfortable in **Android Studios**
I have also done work in SQL, HTML, Assembly, and PHP.

Goals

I wish to work with computer graphics, in virtual reality or augmented reality.

Work Experience

Pizza Ranch – 3 years, currently employed

- Rapid City, South Dakota, 57701
- 605-791-5255

DANIEL ANDRUS

Phone: (605) 269-1728
Email: danieleandrus@gmail.com
Twitter Handle: @deaboy100
Github Name: Deaboy

PROFILE

I am an undergraduate college student at the South Dakota School of Mines and Technology. I have a passion for video games and technology, and my career goal is to become a developer in the games industry, the mobile application industry, or the desktop application industry. I grew up in Los Angeles, California, then moved to South Dakota in Summer, 2010. I attended Black Hills State University for two years before transferring to South Dakota School of Mines and Technology, where I plan to graduate with a bachelors degree of computer science in May, 2016 and immediately begin working in software or game development.

EXPERIENCE

INTERN DEVELOPER, 7400 CIRCUITS — SUMMER 2015 - PRESENT

I held an internship at 7400 Circuits, a circuit board company located in Rapid City. Here I worked to improve an existing an iOS and Android game called *Trouble with Robots*. I also worked on a cross-platform desktop application that interacted via USB with a handheld game cartridge reader and writer that allows users to create and play Neo Geo Pocket and WonderSwan games on their handheld game devices.

SDSMT PROGRAMMING TEAM — 2014 - PRESENT

In fall 2014, I joined the SDSMT programming team and participated in the ACM regional Programming Competition where my team finished 14th in the region out of over 285 competing teams and 1st in the school.

SERVER ADMINISTRATOR, PROGRAMMER — 2010 - PRESENT

Since 2010, I have owned and operated a public game server for which I and another developer have written hundreds of lines of server software to help manage the community. Through this, I have become greatly acquainted with Linux, SSH, and managing small communities.

WEB DESIGNER AND DEVELOPER, BLACKHILLS.COM — 2013 - 2015

In May 2013, I started working for a local web development company as a full-time web developer. The job entailed designing and building websites of diverse sizes and varieties. Many sites were for small businesses located throughout the Black Hills, but a few were for large, high-traffic businesses such as BlackHillsNews.com and Sturgis.com.

INTERN, FTW INTERACTIVE (NOW RED SHED TECHNOLOGY) — SUMMER 2012

I held an internship at FTW Interactive, now known as Red Shed Technology where I worked with experienced developers on mobile app projects. I gained experience working with server and client communications and data processing.

SKILLS

- Programming in the **Java**, **C**, **C++**, **C#**, **PHP**, **Python**, **Objective-C**, and **Swift** programming languages.
- **OS X**, **iOS**, and **Android** development.
- Working with web technologies, including **HTML5**, **CSS**, **JavaScript**, and **PHP**.
- **Designing database systems** using **MySql**
- Working on **team projects**, **object-oriented program design**, and source control systems such as **Git** and **Subversion**

EDUCATION

Black Hills State University, Spearfish, SD — 2010-2012
South Dakota School of Mines and Technology, Rapid City, SD — 2012-2016

PERSONAL INFORMATION

I am good at math, am a fast learner, can pick up on new programming languages and standards quickly, and am a stickler for the proper usage of the word "literally". I can easily adapt to design patterns as well as programming paradigms and am perpetually learning the technologies and techniques employed in the software development, UX design, and games industries.

In my spare time, I enjoy playing and creating video games, creating YouTube videos, and learning more about the ever-changing technology industry. I love spending time with friends who enjoy similar things as I do. My career goals are to go into mobile application design and development, desktop application design and development, or game design and development. My ultimate personal goal with technology is to create applications that make people’s lives better.

C. Nicholas Bonn

2326 Lance Street, Rapid City SD 57702
(651) 503-2877 charlesnicholasbonn@gmail.com

Education:

South Dakota School of Mines and Technology, Rapid City, SD — Anticipated Graduation: May 2016
Bachelor of Science in Computer Science — Cumulative GPA: 2.5

Relevant Coursework:

Database	Software Engineering
Cyber Security	Graphic User Interface

Projects:

Crowd Control App – on-going senior design project
Description: a phone app designed to manage groups in a social setting, to track the members of the groups and ease social gatherings

Technical Skills:

Languages:
Proficient in: C/C++, Python, C#
Familiar with: Java, ARM Assembly, HTML/XML, Lisp, Qt Environment, Visual Basic

Other Technical Services:
Databases: SQL Server, MySQL
Platforms: Microsoft Windows (Active Directory), Mac OSX, and Linux

Work Experience:

Discover Program - Rapid City School District, Rapid City, SD — September 2009 – Current
Program Assistant

- Co-leader for after school and summer programs for elementary aged children
- Coordinate activities for 2nd and 3rd grade program
- Tutor children with their homework
- Mentor children and provide a positive environment for learning and activities

TMI Coatings Inc., Eagan, MN — May 2012 – August 2012
Summer Intern

- Traveled to potential clients in Midwest region to collect specifications for job bids
- Drove equipment and job supplies to job sites in the Midwest
- Assisted in shop preparing equipment and supplies
- Oversaw scanning and organization of job components into electronic storage database

Awards:

Butterfield Cup — May 2015
Award from local entrepreneurs to the best mobile app business plan, product and investor pitch

References:

Available upon request

Evan Paul Hammer

402 South St
Rapid City, SD 57701
Phone: 763-257-5060
E-mail: evan.hammer@mines.sdsmt.edu

Objective

Looking for a Full-Time opportunity in a competitive and leading edge company with a focus on intrapreneurship.

Education

South Dakota School Of Mines and Technology, Rapid City, SD **Expected Graduation:** May 2016
B.S. Computer Science; **GPA**: 2.9 August 2009 - Present
Activities:

- Member in Triangle Fraternity, a fraternity of Engineers, Architects and Scientists
- Member of SDSM&T's Society of Mining, Metallurgy, and Exploration Engineers

Experience

Software Developer May 2015 – Present
Golden West Telecommunications, Rapid City, SD

- Used mostly Python and JavaScript for development
- Mobile development with the use of Sencha Touch and Apache Cordova
- Proof of Concept work with SDK's and API's

Operator January 2014 – September 2014
Deadwood Biofuels, Rapid City, SD

- General shop cleaning
- Help with maintenance of equipment and Machines

Night Chaperone/Office Assistant September 2009 - July 2013
SDSM&T Youth Programs, Rapid City, SD

- Work with students attending the SDSM&T Engineering and Science camps.
- Teach the students about Engineering and Science
- Trained all the other chaperones and TA's
- Assisted in general office work

Skills and Interests

Leadership:

- Taught leadership skills to upcoming Boy Scout Leaders at a camp called Grey Wolf
- Eagle Scout

Computer Science:

- C,C++, Python, ARM Assembly, JavaScript, Lisp
- Experience with Native Mobile Development
- Experience with Cross-Platform Development and MVC
- Experience with Open GL
- Operating Systems: Windows, Linux, Mac OS
- Experience in Database Management - MySql, PostgreSql
- Experience with Git and Subversion

Awards:

- Butterfield Cup - 2015

Joseph Mowry

Skills

Computer Languages	C/C++, C#, ARM, SQL, HTML5, JavaScript, Java, Visual Basic, Python (3.X+)
Protocols & APIs	JSON, XML, .NET, REST
Databases	Microsoft SQL
Tools/Misc.	GitHub, Mercurial(Hg), Team Foundation Server, Android Studio, Visual Studio, Xamarin, LaTeX, SQL Server Management Studio

Organizations/Misc

- Educated in over four years of Spanish
- SDSM&T ACM Chapter Member
- SDSM&T Programming Team
- Attended the Black Hills Engineering Business Accelerator
- Awarded the Butterfield Cup for "Excellence in Software Engineering"

Work Experience

Period	**May 2015 — August 2015 (Full-Time)**	
Employer	**Innovative Systems**	Rapid City, SD
Job Title	**Software Developer (Intern)**	
Languages	**C#, SQL, Xamarin.Forms, .NET Framework**	
	Cross-platform mobile development (MVVM) in Xamarin Forms, C# back-end development/stored procedures in MSSQL	

Period	**May 2014 — August 2014 (Full-Time)**	
Employer	**Emit Technologies**	Sheridan, WY
Job Title	**Software Developer (Intern)**	
Languages	**C#, JavaScript, HTML, .NET Framework, SQL**	
	Front-end (web) development in C#, stored procedures in MSSQL,followed MVC development pattern	

Education

University	**South Dakota School of Mines & Technology**	
Major	**B.S. in Computer Science**	
GPA	**2.7**	
Grad Date	**Spring 2016**	(Projected)

2326 Lance Street, Rapid City, SD, 57702 ·
✉ joe.mowry92@gmail.com ☎ (605) 209-0208 ·
https://github.com/jmowry

2 ABET: Industrial Experience Reports

As a group we have attended the SD Engineering Accelerator. We have compeated in multiple business plan competitions including:

- Butterfield Cup
- SD Innovation Expo Business Plan Competition
- 2015 SD Mines CEO Student Business Plan Competition

We also have also have and regular meetings with SDSMT EIR's to help format our buisness plan and Crowd Control.

2.1 Johnathon Ackerman

I have had no Internship experience. However, before the project Crowd Control, I worked with C++, lisp, and python. I have worked with Visual Studios on Windows side, and Vim and G edit in Linux.

2.2 Daniel Andrus

I first learned the basics of web design and development in high school. After my second year of college, I obtained an internship with FTW Interactive (now known as Red Shed Technologies). Later, I hold a position as Web Developer for 2 years before becoming an intern software developer at 7400 Circuits.

My course experience has ranged from data structures, image processing, database design, web development, group projects, computer graphics (including 3D graphics), mobile app development, and even compression.

2.3 Charles Bonn

I currently have little internship experence. What industry experience i do have is HTML. In my personal/professional life i help manage a website and a minecraft server. Though this is work i have worked with HTML and C code. I have also worked with game code that is java based.

2.4 Evan Hammer

I am working for Golden West Telecommunications(GW), a rural telecommunications provider in the state of South Dakota. Since May of 2015 I have been a Software Developer for GW working on both mobile and back-end products. For the mobile side, I have been working with a product called Cordova that is wrapped with another product called Sencha Touch. Together these two products allow a developer to use JavaScript, HTML, CSS and more to produce a mobile application for Android, iOS and many other mobile platforms. I have also written the back-end for this app, using Python and a PostgreSQL Database creating a server-side API for the mobile application. While I am not working on the mobile application I have spent my time working on other in-house products using languages like Python and JavaScript. These projects have ranged from updating existing code to ground-up projects. Also as a Software Developer for GW, I have been tasked with creating some proof of concept work. This work has ranged from testing possible new services as well as testing new platforms for development. My work continues to grow and change as I continue to work for Golden West Telecommunications.

2.5 Joseph Mowry

In his pirior industry experience, Joseph specialized in C# development and database management. His employers gave him a solid footing in AGILE and Scrum methodologies, as well as general product development. Though his experience lies primarily on the Visual Studio/C# side of things, there is a large amount of skill overlap in Android Studio and Java that he can bring to the table for this project.

D

Acknowledgment

**As a special thanks we would like to thank Brian Butterfeild. His mentouring has made this project possable.
Another thanks goes to Dr. Logar, With out your soft engeneering class this would have never been possable.**

E

Supporting Materials

This document will contain several appendices used as a way to separate out major component details, logic details, or tables of information. Use of this structure will help keep the document clean, readable, and organized.