DukeGather Project Documentation

Author:	Shihao Li Shangxing Sun, Shengbin Wu,
Team:	+3s
Creation Date:	4/17/2016
Last Revised:	4/17/2016

Content

Overview

Purpose

This project documentation act as a communication medium between members of the development team. It serves as system information repository to be used by maintenance engineers. The project document provides information for management to plan, design and schedule the software development process. Additionally, some of the documents should tell users how to use and administer the system.

Scope

This project document covers requirements specification, system architecture, class diagram, as well as design details and implementation techniques.

Project

DukeGather is an Android mobile application that allows people to quickly form a group so that they can save money by carpooling to somewhere or safely walk home together. The idea of DukeGather is based on two situations as following. On the one hand, many crimes such as robbery and sexual assault happen among students when they are walking home alone at night or even in daylight. Students receive alert emails for crime report almost every month. According to National Center for Education Statistics, "in 2013, there were 27,600 criminal incidents against persons and property on campus at public and private 2-year and 4-year postsecondary institutions that were reported to police and security agencies".[1] A report from U.S. Department of Justice reports that "3.1% of undergraduates survived rape or attempted rape during a 6–7 month academic year".[2] On the other hand, taking a taxi, Uber or Lyft to somewhere far away from one's location is quite

expensive. Many students, especially international graduation students, do not have their own cars. However, they may need to go somewhere away from university frequently. For example, Chinese students may want to buy some Asian vegetables or eat in a Chinese restaurant away from their university. This will cost considerable money if there is only one student or two. Therefore, students could have saved money if they had downloaded such an application. Besides, DukeGather might reduce the rate of crimes since students can walk home together rather than alone. In conclusion, an application such as DukeGather is necessary and helpful. In order to design this application, several steps are needed.

DukeGather is designed on Android Studio platform and use firebase as its backend. The front end collects request and message from users and sends data to the backend. The real-time database then send data back to all the users at the same time. The users can also filter the specific data to show on the screen through a search function.

Requirement

Functional Requirements

1. The user should able to sign in by using email and password or gmail account
   1.1. If using email and password, the user should first be request to put in his email address. If it is already registered, the use is then noticed to put in password. If not, user need to put in username and password to finish registration.
2. After signing in, user will be automatically guided to post board. User can find post information include From, To, Date, Time, Number Limit of Group Member, Current Number of Members. User can join a group by press Join button on that group. After join, user can then enter the group freely by using Enter button.
   2.1. "From" field indicates the place where group members should gather to start the trip.
   2.2. "To" field indicates the place which should be the destination of the trip.
   2.3. "Date" field indicates the date on which group member should gather.
   2.4. "Time" field indicates the time when group member should meet.
   2.5. "Number Limit of Group Member" indicates the maximum number of group members.
   2.6. "Current Number of Members" indicate the current number of members who are in this group. Current number of members should not exceed number limit of group member.
   2.7. After join a group, user can will then be automatically guided to in group chatting interface. After join in, the Join button should disappear. User should then use Enter button to enter group chat.
   2.8. User should verify his email first to able to join a group.

3. User can enter Create New Post interface by pressing the Create New button on post board interface.
    3.1. User can create new post by putting in From, To, Date, Time, Number Limit of Group Member fields. These fields have the same meaning as above. User need to fill in all fields to be able to submit.
    3.2. User can finish creation by pressing POST button. User may also abort creation by pressing back button.
    3.3. Date and time should be picked use Date Dialog and Time Dialog to avoid invalid input.
    3.4. Number of people have a limitation of 10 to avoid invalid input.
    3.5. User should verify his email first to able to create new post.
4. User can enter Search interface by pressing the Search button on post board interface.
    4.1. User can put in one or more fields of From, To, Date, Time, to search for certain posts. If user input none of the field above and press search, it will work the same as refresh.
    4.2. Date and time should be picked use Date Dialog and Time Dialog to avoid invalid input.
5. User can refresh the post by using Refresh button in the menu of post board.
6. User can change the his/her profile setting by pressing the Setting button in the menu of post board.
    6.1. user can change his/her username, email, upload his/her photo and verify email at this interface.
    6.2. By pressing VERIFY EMAIL, the user will receive an email at his/her given email. After verifying, the user can then able to join a group or create new post.
7. In group chatting interface, user can send message, send pictures, check post details, edit post information, check group members, quit from group, and delete this post.
    7.1. user can type in the message box and send by pressing send button.

7.2. User can send pictures by pressing Send Picture button, and select pictures from
7.3. User can check group detail by pressing Post Detail Button. Any group member can check the detail.
7.4. User can edit group information by pressing Edit button. User will then be guided into Edit interface. User can change group information here. User can close the group and prevent others to further join by changing Open Status in Edit interface. Only owner (creator of the post) can use this function.
7.5. User can quit from the group by pressing Quit button.
7.6. User can check other member's information by pressing Members button. User will be guided to member interface and can check other member's username, email and photo there.

Non-functional Requirements

Organizational Requirements

The application shall use git as version control. The git repository should be on duke gitlab accessible by teaching assistant and the professor.

Security

The application shall keep the user account safe.

Ease of Use

The application shall be intuitive so no technical guide is required.

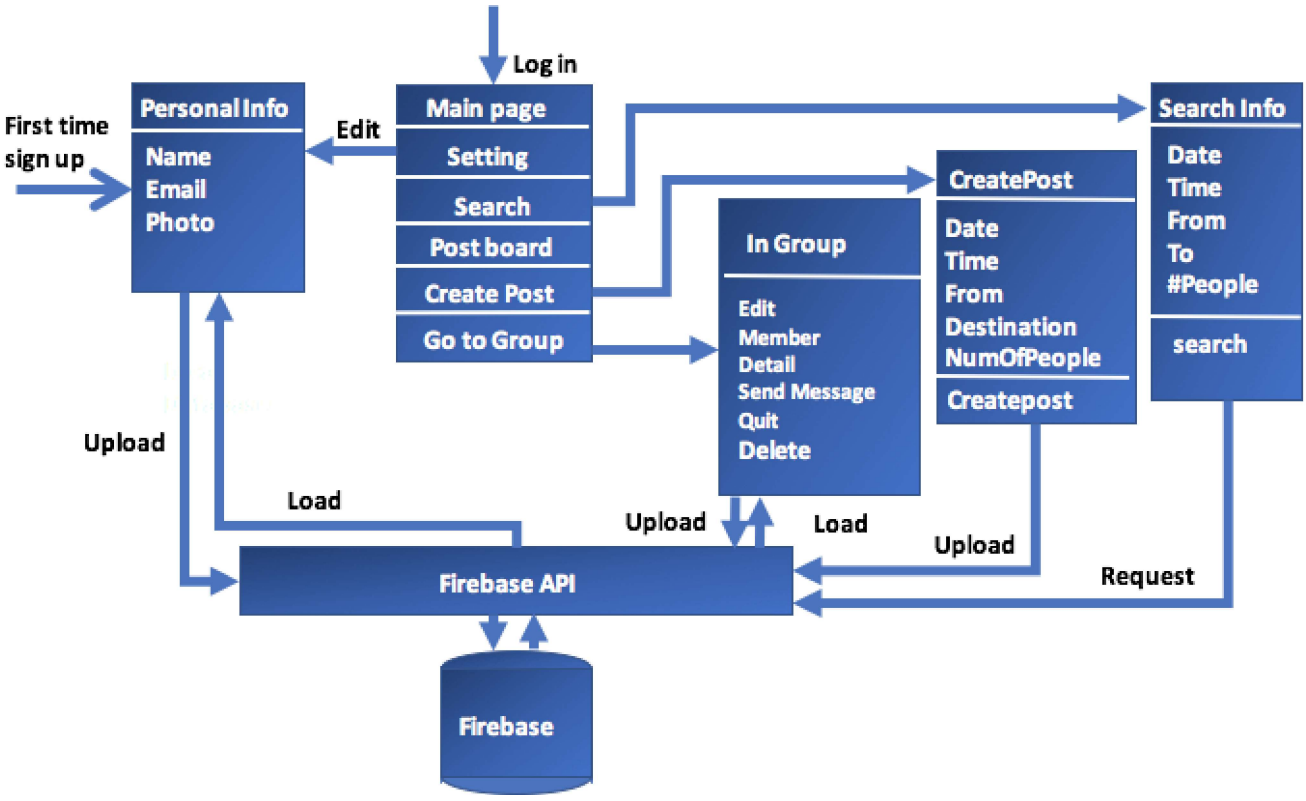
Reliability

The application should not crash when the users are doing some wrong operation with the application.

Release Requirements

The application is expected to be released by 04/21/2017.

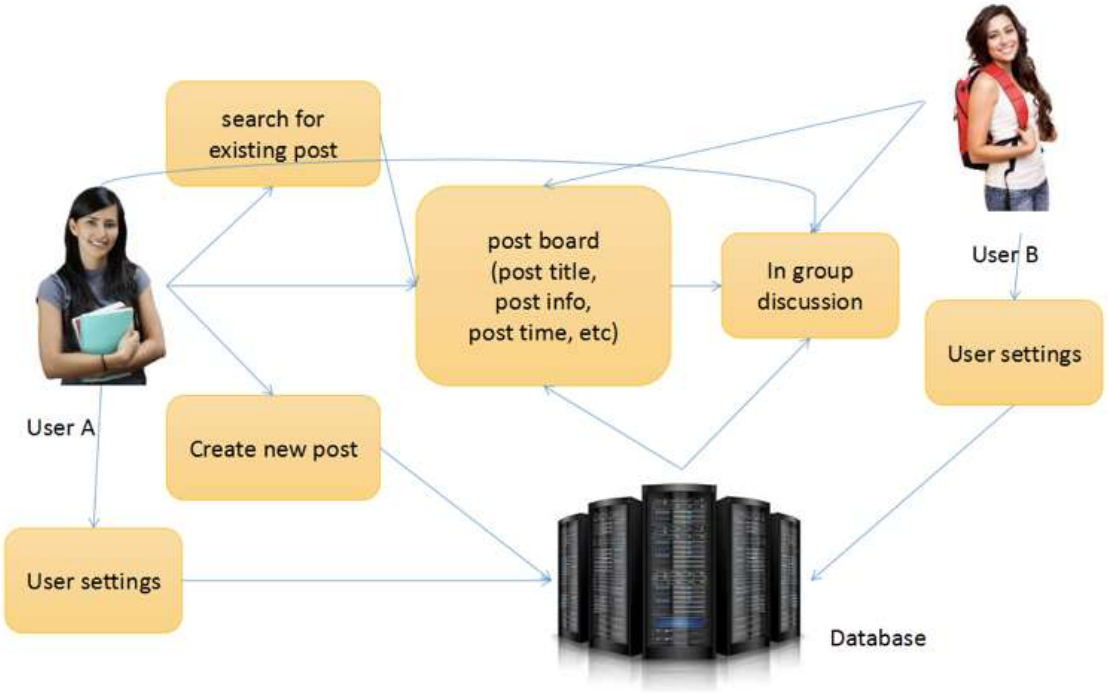
System Architecture



DukeGather's system architecture is shown above. In the front end, the mobile application allows people to store their personal information in the setting interface. It also allows users to create and post their own request to others and users can search for information that matches their need in the front end. Users can also chat with others in the ingroup interface. Firebase serves as a back end for the system. The application mainly uses Firebase's authentication, storage and real-time function. DukeGather implements the sign up and sign in function with the authentication function of Firebase. Personal picture is stored with the storage function. Update interface at the same time as post and send message is achieved by the real-time function.

Design

Use Case



	Search for Existing Post
Actor	User, Firebase
Description	User can search for existing post he/she needs by going to Search interface and type in one of these post information. Post information includes Time, Number of Passenger, From and Destination
Data	Post information
Stimulus	User clicks "Search Post" Button
Response	Skip to Search for Existing Post interface, after searching, skip to Post Board.
Commons	The Post Board after searching will only show posts which meet requirement

	Create New Post
Actor	User, Firebase
Description	User can type in post information to create a new post. Post information includes Date,Time, Number of Passenger, From and Destination.
Data	Post information, user information
Stimulus	User clicks "New Post" Button
Response	Skip to Create New Post interface, after creation, skip to Post Board
Commons	

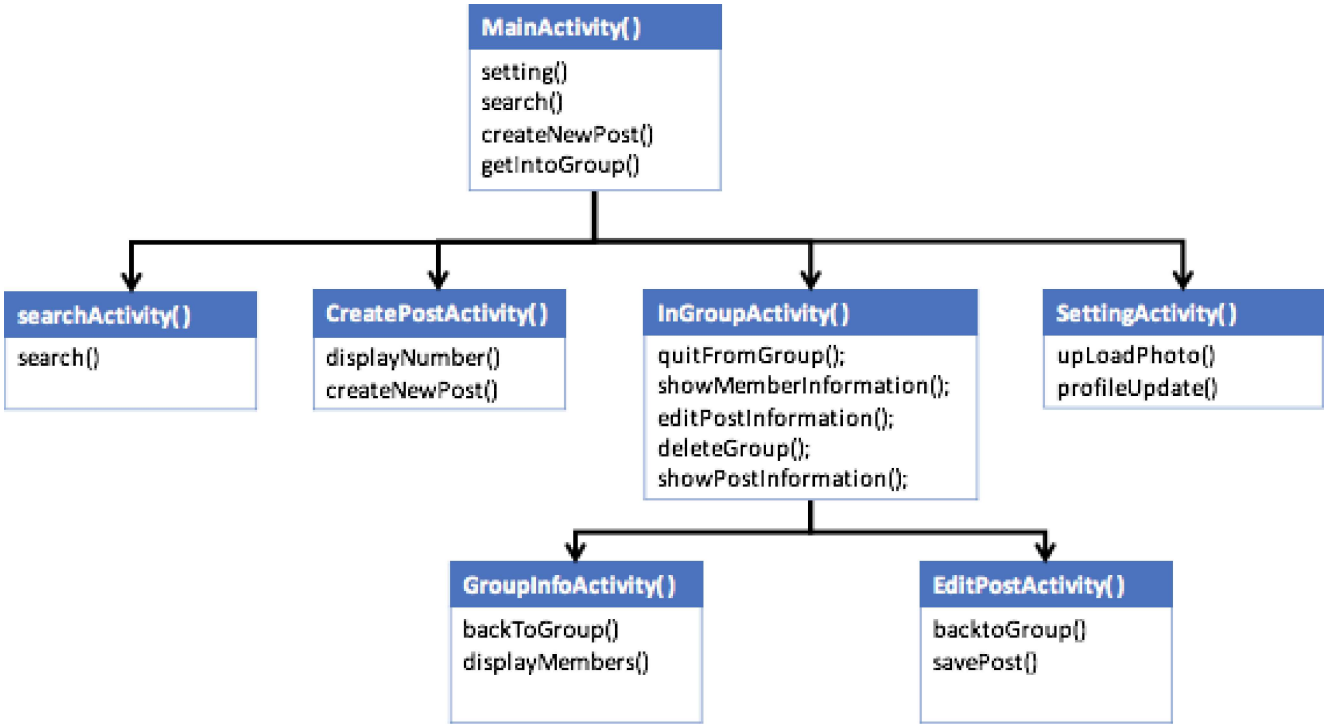
	User Setting
Actor	User, Firebase
Description	User can store their personal information including Name, Email, photo picture. User can change his/her information at this interview. Other user in the same group may be able to view his/her personal information.
Data	Name, Email, picture
Stimulus	User clicks" User Setting" Button
Response	Skip to User Setting interface, after setting, back to previous interview
Commons	

	In Group Discussion
Actor	User, Firebase, other Users
Description	User can see the details of the group. User may leave message for other users who are in the same group. User may also check other users' information. Owner may close or delete the group.
Data	Post information, additional information, user information
Stimulus	User clicks "Group Discussion" Button
Response	Skip to In Group Discussion interface. User can later quit to Post Board.
Commons	

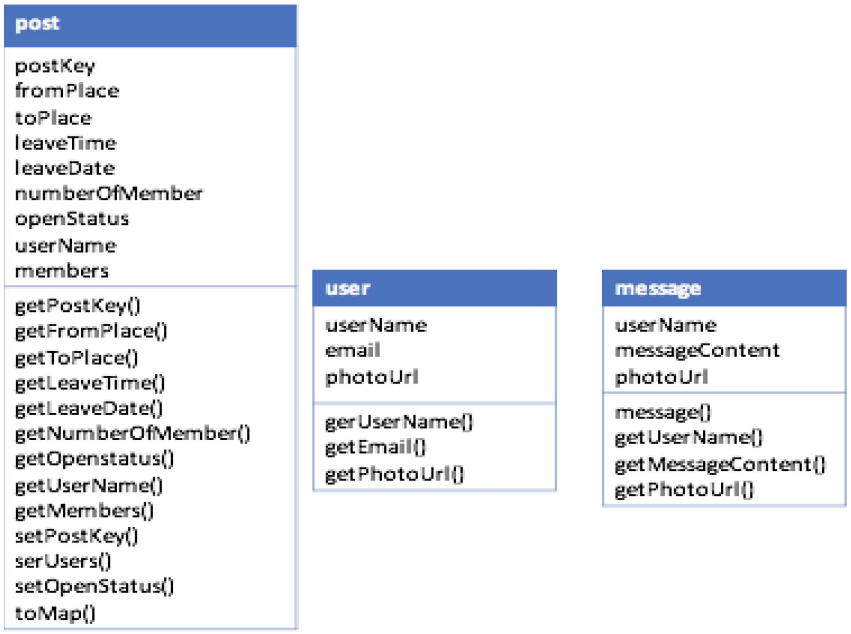
	Post Board
Actor	User, Firebase, other Users
Description	User can see the list of the group. User may join the group. User may also create new groups. User may search for certain groups.
Data	Post information
Stimulus	User automaticly enters after signing in
Response	User automaticly enters after signing in
Commons	

Class Diagram

Activity Class



Object Class



The Class Diagram is shown as above. The Properties and Methods of the application are shown in the diagram. The function of this requirement are fulfilled by this classes. There are five activities in the application, Main Activity, Search Activity, Create Post Activity, In Group Activity, Show Group Information Activity, Edit Post Activity, and Setting Activity. Each Activity is related to an User interface which is interacting directly with the users. Main Activity is the parent Activity of Search Activity, Create Post Activity, Setting Activity and In Group Activity. The In Group Activity is the parent activity of Show Group Information Activity and Edit Post Activity.

In this Application, three kinds of object are implemented, which include Post Object, message Object, and User Object. The Post Object is use to store the information of posts which is created by users. The Message Object is used to store the message sent by user in the group. While the User Object is used to store user's information.

Implementation

Project Organization

The Plus 3s Project is uploaded to GitLab at:
http://gitlab.oit.duke.edu/ECE651_S17_PROJECTS/Plus3s.git
The file in this repository is a mobile application project, which is created by Android Studio.

DownLoad,Build and Run

1. Environment Configuration

JDK Installation

Before downloading the project, you shall check if you have already installed Java Developer Kit, version 7 or greater in your computer. To check if user have JDK installed, user can type "java -version" in your terminal.

If the JDK is not available in your computer, or the version is lower than 7, please download the JDK from oracle.

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Android Studio Installation

Android Studio can be obtained from the website shown as below, https://developer.android.com/studio/index.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics

The setup wizard will set up Android Studio for you, You can go ahead and choose Standard Setup and accept all the licenses.

2. Project Download

After environment configuration, user can download our project by git clone or by downloading the zip file.
git clone http://gitlab.oit.duke.edu/ECE651_S17_PROJECTS/Plus3s.git

3. Project Import

After downloading the project, user can open the Android Studio and and open this project by the "open an existing Android Studio project " option.

4. Project build and run

The Android SDK version should be 25.

The Android Build Tools should be v25.0.2.

Android Support Repository should be v25.2.0

Gradle version should be 2.14.1

Android Plugin Version should be 2.2.3

User can run our project by Emulator in Android Studio or by their Phone.

Test

1. Purpose

This test plan describes the testing approach and schedule that will drive the testing of the Duke Gather Android Application. This Document includes: Test Methodology and Test Schedule.

2. Testing Methodology

Unit Test

Purpose: Unit test will be performed to test every unit of the application. The unit test will be conducted by feeding the unit input and validate the corresponding output from that unit.

Method: For the unit test, the test will be conducted manually for the first test when the relative code is finished. Then the tests based on JUnit will be ran to on the Java Virtual Machine on Android Studio or as instrumented tests on an Android device.

Metric: Function specification in user cases

Component Test

Purpose: Component test will be performed to test the components of the application. The component test will be conducted by interacting between these components and validated the behavior of these components. The Components includes ??User interfaces, local database and databases on server??.

Method: The approach for Components testing is to conduct a set of operations on our application manually by a tester.

Metric: user test cases

System Test
Purpose: System Test will conduct a series of test to check that components are compatible, can be interacted with each other correctly, and data can be transferred correctly between different interface.

Method: The approach for System test will be conducted manually by the system tester. Then the tests based on espresso will be ran to on the Java Virtual Machine on Android Studio or as instrumented tests on an Android device.

Metric: user test cases

3. Testing schedule

Unit Test

Method in search activity

Search method
Description:
Search certain post by putting in conditions. Then send a retrieve request to the database.
Input and Output:
1. Users fill all the conditions includes “from”, “to”, “time” and “number of people” and send request to database.
Output : a request of retrieving is sent to the database
2. Users fill none of these conditions and send request.
Output: The application should return error message to inform user.
Owner:
Shangxing Sun
When:

Finished by 4/10/2017
Result:
Passed, all bug fixed.

Method in CreatePost Activity

Createpost Method
Description:
Create a new post by filling with necessary information of the new post. Then push the new post to the database.
Input and output:
1.Fill all the text fields, and click post button.
Output: This method will check if the information is complete and the form in database is correct. If so, the post will be pushed to the database.
2. Fill some of the text fields and click the post button.
Output: The app will not allow to post incomplete information to database. The application should notify user and remain the same page.
Owner:
Shangxing Sun
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Increase number of people method
Description:
Increase the number of people by one. This method will appears in Createpost activity and Search activity. The most number of people is 20.
Input and output:
User click the “plus” icon
Output:
The number of people shown on the interface add one. However, the number should not exceed 20. If the number reach 20, it will not further increase. The default number of people is 1.
Owner:
Shangxing Sun

When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Decrease number of people method
Description:
Decrease the number of people by one. This method will appears in Createpost activity and Search activity. The least number of people is 1.
Input and output:
User click the “minus” icon
Output:
The number of people shown on the interface minus one. However, the number should not below 1. If the number reach 1, it will not further decrease. The default number of people is 1.
Owner:
Shangxing Sun
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Method in In Group Activity

Send message method
Description:
In the group interface, user can send a message to others by the input box for message. When click the “send” button the message will be send to the database.
Input and output:
Input: User types some words in text field and clicks send button
Output: check if the information is complete. If the massage is legal, it will be pushed to the database. After one user sends message, others check the interface on their app to see if there is a message pop out or it can be checked by display our database.
Owner:
Shengbin Wu
When:

Finished by 4/10/2017
Result:
Passed, all bug fixed.

Enrollment into group method
Description:
In the in group activity, user can enroll himself by clicking the enrollment button.
Input and Output:
input: Clicking the button of enrollment.
Output: The user's id will be add into the corresponding group table in the database. The user can then participate into group chat.
Owner:
Shengbin Wu
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Edit content
Description:
In the group activity, group member can edit the content of a group.
Input and Output:
Input: the content of of group, such as conditions includes "from", "to", "time" and "number of people" and send request to database.
Output: the content of the corresponding group table in the database will be changed.
Owner:
Shengbin Wu
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Quit from group method
Description:
In the in group activity, user can quite from this group by clicking the quit button.
Input and Output:

Input: Clicking the button of quit.
Output: the user's id will be delete from the corresponding group table in the database. The user can no longer read or write group chat in that group.
Owner:
Shengbin Wu
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Method in Setting Method

Setting method
Description:
Users can set up their information(include name, gender, e-mail, phone number) in this interface. Information will be checked if valid. If so, information will be pushed to the database.
Input and output:
1. User input valid information.
Output: Information will be checked before storage. If valid, Information will be stored in local database.
2. User input invalid information.
Output: Information will be checked before storage. If invalid, a warning toast message will be popped out.
Owner:
Shengbin Wu
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Component test

Interfacing test
Description:
Different interfaces can skip to each other with buttons.
Input and output:

1. Click the post button in main page, check if it skips to post page.
2. Click the set button in main page, check if it skips to setting page.
3. Click the search button in main page, check if it skips to search page.
4. Click the add button in main page, check if it skips to the ingroup page.
5. Click the back button in create post page, check if it skips to main page.
6. Click the post button in create post page, check if it remains at the create post page.
7. Click the back button in search page, check if it skips to main page.
8. Click the search button in search page, check if it remains at the create post page.
9. Click the back button in setting page, check if it skips to main page.
10. Click the save button in setting page, check if it skips to main page.
11. Click the discard button in setting page, check if it remains at the setting page.
12. Click the clean button in setting page, check if it remains at the setting page.
13. Click the back button in ingroup page, check if it skips to main page.
14. Click the send button in ingroup page, check if it remains at the setting page.

Owner:
Shihao Li
When:
Finished by 3/30/2017
Result:
Passed, all bug fixed.

Search activity test
Description:
Search certain post by putting in conditions. Then retrieve corresponding result from database.
Input and Output:
1. Users fill all the conditions includes “from”, “to”, “time” and “number of people” and send request to database.
Output : database correctly returns corresponding posts.

2. Users fill none of these conditions and send request.
Output: The application should return error message to inform user.
3. Users fill any of the conditions and send request to database.
Output: database correctly returns corresponding posts.
Owner:
Shangxing Sun
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Createpost Activity
Description:
Create a new post by filling with necessary information of the new post. Then push the new post to the database and display it in the main post interface.
Input and output:
1.Fill all the text fields, and click post button.
Output: The application will check if the information is complete and the form in database is correct. If so, database successfully stores the data and the interface automatically jumps back to main page. The new post can then be found on main page.
2. Fill some of the text fields and click the post button.
Output: The app will not allow to post incomplete information to database. The application should notify user and remain the same page.
Owner:
Shangxing Sun
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

Setting Activity
Description:
Users can set up their information(include name, gender, e-mail, phone number) in this interface. Information will be checked if valid. If so, information will be first stored in local database and then upload to firebase.

Input and output:
1. User input valid information.
Output: Information will be checked before storage. If valid, Information will be stored in local database and then upload to firebase. We can check from firebase that the information has been successfully received. The users can find the information they set once they click the change setting button.
2. User input invalid information.
Output: Information will be checked before storage.If not valid, either because of error input type or error input length, the application will notify the user. Invalid information will not be stored in either database.
Owner:
Shengbin Wu
When:
Finished by 4/10/2017
Result:
Passed, all bug fixed.

System Test

Description:
2 people open the app, they both click the set button in main page and set the personal information in setting page. After click the save button, they return to the main page. One person clicks the post button, then types the detailed information in create-post page. After that, the other person click the search button, and type in the information matched with the first person's request. Then check if the first person's post show in the main page. If this step success, the second person click the add button in the main page, then send message in the ingroup page. The first person can check if the message sent by the second person is shown on his or her app.
Owner:
Shihao Li, Shangxing Sun
When:
Finished by 4/20/2017
Result:
Under testing, no bugs remain until now.

Appendix-A Project Plan

DukeGather is an Android mobile App that people can form a team fast and conveniently so that they can save money by carpooling together to somewhere or students can walk home or walk to somewhere together safely.

Project Documentation Plan

The documentation of this project will include the following documents, requirement of this project, description of this project, system architecture, class diagram, user cases, development plan and user instruction. A appendix of README, and an User Instruction will also be provided. This documentation will serve as a supplementation of the code deliverable with all relevant and necessary information. The Project Documentation will be in the form of a .pdf document and will be delivered and presented on April 20th.

Task ID	Description	Owner	Start	Finish	Dependencies
Rqmts	The initial requirements document needs to be cleaned up, refined, and finalized.	ShihaoLi, Shengbin Wu	02/24	03/16	None
Project Dscrp	The overall project Description needs to be concise	Shangxin Sun	02/23	03/12	None
SysArch	The draft system architecture	ShihaoLi	03/09	03/16	Requirements, Sprint 1

	needs to be updated and more detailed				
Class Diag	The class diagram shall be updated and revised according to the feedback from sprint 1	Shengbin Wu	03/09	03/15	Requirements, System Architecture, Sprint 1
User Cases	The User cases need to be revised and finalized	Shangxin Sun	03/09	03/15	Requirements, Sprint 1
Devlop Plan	The development plan need to be revised and updated	Shangxin Sun	03/09	03/15	Requirements
User Instrt	The User instruction should be created when this mobile application is finished	Shengbin Wu	4/1	04/12	Sprint 3

Code Development

This is a complete project plan for the development of the DukeGather app, starting from the beginning of sprint 1. The entire work will take 3 sprint to be finished. The first sprint will take 14 day while the 2nd and 3rd sprint will take 21 days. The final deliverable will be pushed to Git on April 20th. We will push all the code into Gitlab along with complete documentation in the READ.ME on how to build, install and run the project.

Sprint 1

Sprint 1 will be delivered on March 9th and mainly focusing on building basic UI and local storage. Specifically, the first sprint will display all these UIs to user and will allow a user to store User Information into local storage and switch between these UI.

Task ID	Description	Owner	Start	Finish	Dependencies
Create Post UI	Create the New Post UI and associated Controller for the view.	Shangxing Sun	02/24	03/08	None
Main UI	Create the Main UI and associated with other views	Shihao Li	02/24	03/08	None
Search Post UI	Create the Search Post UI and associated controller for the view.	Shangxing Sun	02/24	03/08	None
User Setting UI	Create the User information setting and corresponding function	Shengbin Wu	02/24	03/08	None
Local Storage (Finished at time, but proved to be unnecessary,aborted)	Store user information into cell phone memory by using SharedPreference	Shengbin Wu	02/24	03/08	User Setting UI
In Group Discussi	Create In Group	Shengbin	02/24	03/08	None

on UI	Discussion UI and associated controller for the view	Wu			
Test and Debug	Test the corner and general cases and debug	All	03/08	03/09	Other works in sprint 1

Sprint 2

Sprint 2 will be delivered on April 1st and will realize basic function. User can post and receive from server and interact with other users. Messages and emails will automatically be sent to users to inform them about the important information of their group.

Task ID	Description	Owner	Start	Finish	Dependencies
Server and Database	Use Firebase to store posted data and retrieve data	Shangxing Sun, Shihao Li	3/10	4/10 (should be 3/31, finished behind schedule)	None (previously on Local Storage)
App Server Communication (later proved to be unnecessary, aborted)	Send and receive information with Json through HTTP	Shihao Li	3/10	4/10	None (previously on Local Storage)
Authentication	Login authentication for Duke student	Shengbin Wu	3/10	3/31	None (previously on Local Storage)
Message	Send email	Sheng	3/10	3/31	In Group

reminder and Email Sender (proved to be unnecessary later, aborted)	and message to users automatically if there are some news in their group	bin Wu			Discussion
Email Verification (new add feature)	Send email to users to verify his/her email address.	Shang xing Sun	4/1	4/10	Authenticat ion.
Test and Debug	Test the corner and general cases and debug	All	4/10	4/11 (should be 4/1, finishe d behind schedu le)	Other works in sprint 2

Sprint 3

Sprint 3 will be delivered on April 20th. All tests and debugging will be finished at that time. Some polishment on the UI will then be made to improve user experience.

Task ID	Description	Own er	Start	Finis h	Dependencie s
Google Map API (Time limited to implemen t)	Using Google Map API to show map in Search Post UI	All	4/1	4/18	Search and Post UI, Server and Database
Debug and Test	Test the corner and general cases and	All	4/1	4/19	Other works except polish in sprint 3

Polish	debug				
	Make a clean project, add needed comments and make the code clear and concise	All	4/1	4/20	Other works in sprint 3

Appendix-B User's Guide

DukeGather App

This app is build for gathering student's to call a duke safe ride.

Pre-requisites

- Android SDK v25
- Android Build Tools v25.0.2
- Android Support Repository v25.2.0

Getting Started

- This project uses the Gradle build system. To build this project, use the "gradle build" command or use "Import Project" in Android Studio.
- We recommend using Android studio to import this project, because we have not exported a .apk file that can be installed in your phone.
- Gradle version should be 2.14.1
- Android Plugin Version should be 2.2.3
- There are some bug reported on the newest version of gradle, so now we can just use the gradle version as mentioned above.

Overall structure

There are 7 main interfaces in our app, they are:

1. Post board (MainActivity), display all post groups, which can jump to search, create new post, in group chatting interface.
2. Search (SearchActivity), search for certain groups in post board, which can jump to post board.
3. Create new post (CreatePostActivity), create a new post in post board, which can jump to post board.
4. In group chatting (InGroupActivity), allow users to chat in the group, which can jump to edit post, group member.
5. Edit post (EditPostActivity), allow owner of the group to change group information.
6. Group member (GroupInfoActivity), allow group members to find information (email, name, photo) about other group

members.

7. Setting (SettingActivity), allow users to change his/her own name, email and upload his photo.

Key APIs into the back end

We are using Firebase as back end for our app. We also have used some APIs for images displaying

APIs includes:

- Authentication
  com.firebaseui:firebase-ui-auth:1.2.0
  com.google.firebase:firebase-auth
- Real time database
  com.google.firebase:firebase-database:10.2.1
  com.google.firebase:firebase-core:10.2.1
- Firebase storage
  com.google.firebase:firebase-storage:10.2.1
- Message Dependency
  com.google.firebase:firebase-messaging:10.2.1
- Image displaying
  com.github.bumptech.glide:glide:3.6.1

Functionality

1. **User setting**

New user shall set his/her name, email address before using this app. He/she can also upload his photo into database and further use it as icon. In this interface, there should be several text-boxes for user to input their information. This interface is not used for login or authentication. It is used to collect the information of user for further activity inside this app. User can also change their information mentioned above by entering the setting button in the main interface. User then need to verify his/her email before further join a group or create a group.(After verification, please sign out and sign in again to make verification work.) The information will save unto the Firebase.

This interface is showed as below.

2. Main Interface

- A search button for user to search relative events that the user want to join. When clicking the button, it will direct the user to the search interface.
- In the middle of this interface, there are some tags that show the feature you are searching. For example, if you search the start time and destination in the search interface, the tags “destination” and “time” will show on the main interface.
- On the top left of this interface, there is a setting button. It will direct the user to the User Setting interface.
- On the body of this interface, there is a list (main post board) of recent information sorted by the features you typed. At the tail of each post, there is a join button. When user click the button, user can join this group and can further press enter button. It will turn to the in-group chatting interface. User can scroll this list to see more posts when all the posts cannot be shown at the same time.

- At the button of the page, there is an add button. When the user cannot find a group that he/she wants to join, user can click this button to make to new post, which will take the user to the new post interface.

3. Search interface/New Post

- Type search information: From, To, Date,Time, Number pf Passengers
- Type post information: From, To, Date,Time, Number pf Passengers
- Button for start searching
- Button for posting
- Button for cancel

By touching the question mark on the top right corner of the board, the user can then enter the search menu. By touching the plus icon on the bottom of the board, the user can then enter the post menu.

These two menus have similar layout: five field to type and a button to click. In search menu, the user only need to type in at least one of them and click search button to start search. He can also type in all of them for more accurate search. In post menu, the user has to type in all four of them and click post button to create a new post.

There is also a cancel button on both menus. User can return board by clicking this button.

4. In group interface

- A list that can show the group members
- A button that can show the detail of group information
- A chat board that members can chat with others by posting message and pictures there
- A text-box and a “send” button for user to post message to the post board
- A button that can close this group
- A setting button that can change the detail of this group event
- A button for a group member to quit from the group

Once the user has joined and created a group, he/she can enter the in-group interface.

Inside the in-group interface, there will be a post board in which group members can chat with others by posting message there. Members should be able to interact with other in case of they want to make a change of their schedule. Group members make new massage to the chat board by typing in the text-box and click the

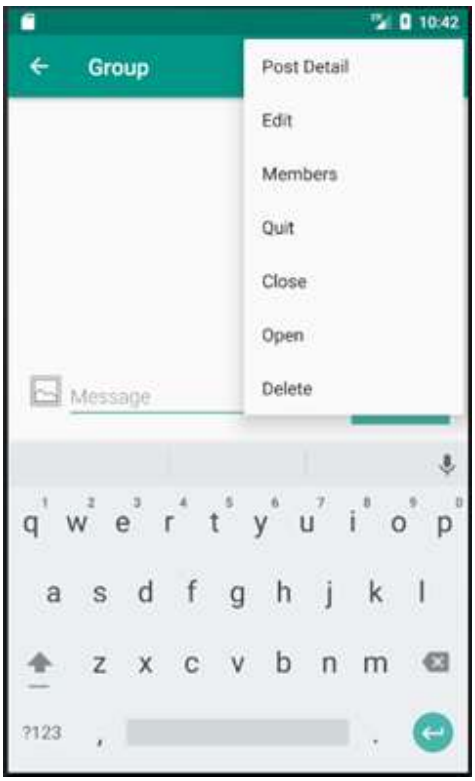
“send” button. They can also send pictures by pressing the figure button on the left.

User can jump to member interface from this interface to see the profile photos of all the group members in a list at this interface. User can find profile photos, name and email of other group members in this interface.

Group host can close the group by clicking the “close” button when the group do not want to accept more people in their group.

Group host can change the detail of their group event, such as leaving time, leaving location, group description and group member number by clicking the group setting button. It will jump to the interface of similar as the new post interface.

This interface is showed as below.



Appendix-C User story

Patrick is a student at Duke who always studies at the library until late at night. After finishing dinner at campus, he starts considering how to get back home tonight. He does not want to go back to his apartment alone because it is unsafe out there especially at midnight. He wants to find some companies, other students at campus, who can go back together with him. Then he opens his android app “DukeGather”, he first looks up in the main interface (an information board) of this app to see if there is anyone that want to leave campus at the similar departure time and location. He looks through the information board, then clicks the search button in the main interface. He enters the location and time that he wants to leave and his destination. After entering the “search” button, he gets some relative post.

Scenario A. Fortunately, he finds a post in main interface shows that a group of students want to leave campus at 11:30 pm near the Perkins Library and the destination of this group is not far from his. This group of students have already schedule a time and pick up location with the Duke safe ride inside the app. There is still space available for new group member. Then, Patrick joins this group. After joining this group, he gets into the group interface where he can see other group members and talk with them by posting message there. He enters his specific destination location in this interface. A group member asks in the group interface to change the leaving time because they have a meeting and could not leave until 11:40pm. After everyone agree with this request, they reschedule the time of leaving to 11:50pm with Duke safe ride.

Scenario B. Unfortunately, he doesn’t find a post about leaving from a place near Perkins library around 11:30 pm. Then, he decides to makes a new post to gather people go with him. He clicks the “new” button in the main interface. He gets into the new post interface, then he enters the leaving location, leaving time, upper group members number and a short description of this event inside

this interface. Then he submits his new post. After submitting, he can see his new post in the main interface and can also get into his group by clicking this post in the main interface. Once a new member joints into the group, he will get a notification from the app.