

The Naughty App

Exam e16

Group 13 | ITSMAP | 28/09/16

Contents

[The Naughty App 1](#_Toc462834896)

[Design overview 1](#_Toc462834897)

[Components 1](#_Toc462834898)

[Risks 2](#_Toc462834899)

[User Stories 2](#_Toc462834900)

[User Story 1: 2](#_Toc462834901)

[User chats with contact 2](#_Toc462834902)

[User Story 2: 2](#_Toc462834903)

[User receives picture 2](#_Toc462834904)

[User Story 3: 2](#_Toc462834905)

[User navigates to picture location (pication) 2](#_Toc462834906)

# The Naughty App

The idea with “The Naughty App” is to create a SnapChat like application where you you can chat with your friends, send them pictures of places (more importantly, people) that you find interesting and then show on a map where the picture was taken.

* The Chat system will be a simple chat system just like facebook messenger (more simple) or like sms system.
* In the chat you will be able to attach a picture which will include coordinates
* The pictures you send and receive will be stored in a local database and can be retrieved so you can see where all the pictures were taken. Just for fun of course.

# Design overview

## Components

The app will consist of three activities.

* MainActivity
  + You can see a list of all your contacts and choose which to communicate with.
  + You can add new contacts.
  + You can navigate to the map activity.
* ChatActivity
  + You can write messages
  + You can take and send pictures
  + You can navigate to the MapActivity
* MapActivity
  + You can see the position of one picture (navigated to from ChatActivity)
  + You can see the position of all pictures ever send or received.
  + You can get a route to a picture

The app will have a DAO that will access the database.

The app will have a service that will pull an update on the current weather so you know when to go outside and when not to.

### Risks

To implement a chat system will probably prove to be a difficult task.

To ease this task, the [pubnub](https://www.pubnub.com/solutions/chat/) API has been taken into consideration. A chat system would probably include some kind of web hosted server, that would include all contacts and furthermore handle the messages and such.

But since the chat system is a core part of the program, it will see a fair amount work to complete the task.

# User Stories

## User Story 1:

### User chats with contact

1. User chooses contact 1 from list of contacts
2. User send text message saying “hi friend”.
3. User chooses to take a picture to send.
4. User clicks camera icon. Default camera app opens.
5. User takes picture and sends it to contact 1.
6. Picture is saved in local db.
7. Contact 1 can now see picture.

## User Story 2:

### User receives picture

1. User receives picture from contact 1.
2. User views picture.
3. User choses to view location of picture.
4. MapActivity opens.
5. Location of picture is now shown on map.

## User Story 3:

### User navigates to picture location (pication)

1. User opens MapActivity
2. User can now see all pictures send and received and their locations.
3. User chooses a picture.
4. User chooses to get a route.
5. Route is displayed .