FRAPP System design document

1 - Introduction

The Frapp application is an android client to help you locate a certain room or building
by drawing annotations from the database onto a global map Making it easier for students to find their way around campus.

1.1 - Design goals

FRAPP is an application that uses Google online services and a local database. While the internet requirement is a concern, it was deemed a minor concern.
In order to achieve the following design goals we thought off a few things that could be a problem:

- Making sure users are able to perceive where they are and how close they are to the goal.
- Making sure users can easily navigate to their destination

1.2 - Definitions, acronyms and abbreviations

Building Annotation: Draw on set map, showing entrance(s)
Annotation: A point on the map with little information on title and description

1.3 - References

N/A

2 - Proposed system architecture

2.1 - Overview

The android client will show annotations on the map according to the name from our XML data. Which is parsed as a simple database, and then the user will be able to search for an interesting location.

The primary design is a client-server application that focuses on displaying information to the user.

2.2 - Software decomposition

2.2.1 - General

The application will be divided into three parts, one activity for searching for locations, one activity for displaying locations and additional information to help users find their way there and finally a basic configuration for selecting a database.

2.2.2 - Tiers

Application
Dataprovider

2.2.3 - Communication

Google maps requires access to internet activity

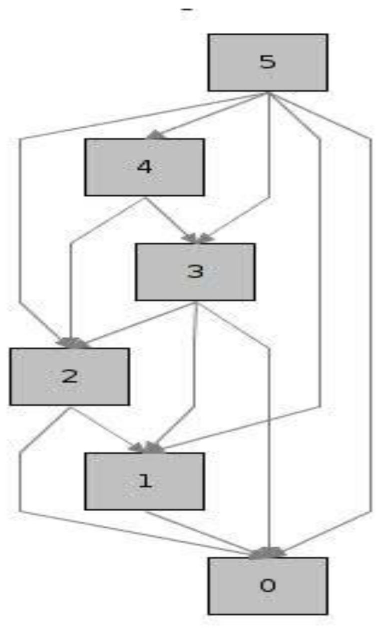
2.2.4 - Decomposition into subsystems

Activities
Data provider

2.2.5 - Layering

N/A

2.2.6 - Dependency analysis



0	com.chalmers.frapp.FRAPPPreferencesActivity com.chalmers.frapp.database.Entrance com.chalmers.frapp.database.Room
1	com.chalmers.frapp.database.Building
2	com.chalmers.frapp.database.LocationDatabase
3	com.chalmers.frapp.DestinationOverlay com.chalmers.frapp.database.Parser
4	com.chalmers.frapp.DisplayLocationActivity
5	com.chalmers.frapp.FindLocationActivity

2.3 - Concurrency issues

Javas GUI toolkit uses a separate thread to manage user events.
We don’t expect and issues with this.

2.4 - Persistent data management

The selected database setting will be stored using the default SharedPreferences class from the Android API.

2.5 - Access control and security

To be able to add content one has to be logged in. Otherwise it’s only possible to view. All traffic to the server will be encrypted and authorized by registered user authentication.

2.6 - Boundary conditions

N/A

2.7 – References

http://developer.android.com/about/dashboards/index.html
We choose android API 10 because it covers the largest part of the entire market