Design Document for GeoStories project

Group#F11

MemberName: Forrest Scott

MemberName: Joel Veencamp

MemberName: Alex Rinehart

MemberName: John Egannhouse

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 | 10/25/14 | FS | Initial Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 3](#_Toc84977455)

[1.1 Purpose 3](#_Toc84977456)

[1.2 Scope 3](#_Toc84977457)

[1.3 Definitions, Acronyms, Abbreviations 3](#_Toc84977458)

[1.4 Design Goals 3](#_Toc84977459)

[2 References 4](#_Toc84977460)

[3 Decomposition Description 5](#_Toc84977461)

[3.1 Module Decomposition 5](#_Toc84977462)

[3.2 Concurrent Process 5](#_Toc84977463)

[3.3 Data Decomposition 5](#_Toc84977464)

[3.4 STATES 5](#_Toc84977465)

[4 Dependency Description 6](#_Toc84977466)

[4.1 Intermodule Dependencies 6](#_Toc84977467)

[4.2 InterProcess Dependencies 6](#_Toc84977468)

[4.3 Data Dependencies 6](#_Toc84977469)

[5 Interface Description 7](#_Toc84977470)

[5.1 Module Interface 7](#_Toc84977471)

[5.2 Process Interface 7](#_Toc84977472)

[6 Detailed Design 8](#_Toc84977473)

[7 Design Rationale 9](#_Toc84977474)

[7.1 Design Issues 9](#_Toc84977475)

[7.2 <Issue 1> 9](#_Toc84977476)

[7.3 <Issue 1> 9](#_Toc84977477)

[8 Traceability 11](#_Toc84977478)

# Introduction

## Purpose

To explain the software architecture of the GeoStories project and to give the rationale behind design choices.

## Scope

This document covers both the Android GeoStories Application and the GeoStories web server.

## Definitions, Acronyms, Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| Zend Framework 2 | open source framework for developing web applications and services in PHP |
| Doctrine 2 | several PHP libraries primarily focused on database storage and object mapping |

## Design Goals

1. Reliability

The GeoStories project has many connections and data objects. The system must perform reliably in order to accomplish its core functions.

1. Security

It’s our responsibility to keep the user’s private data, including passwords, secure.

1. User Friendly

The GeoStoies project is a social application. In order to attract a user base of all ages, the system must be easily understood by an untrained user.

1. Maintainability

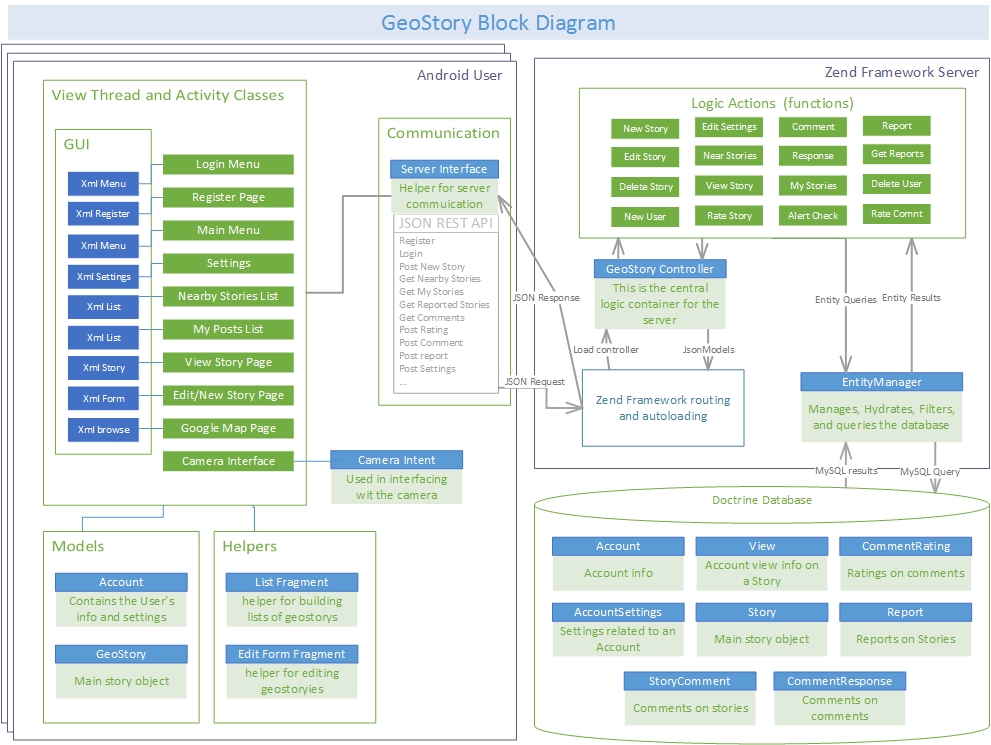
The system must be easily maintainable to keep up with new features and a growing user base.

1. Response Time

The system communicates through the web often. Because it’s a user based application, response times must be kept to a minimum to appease its user.

# References

(If any)



# Decomposition Description

## Module Decomposition

See previous figure “Geostory Block Diagram” for Block Diagram.

### <Module 1 > Description

### <Module 2> Description

### <Repeat for as many subsystems that you have>

## Concurrent Process

<Give an overall description of the different processes, threads, instantiations>

### Android View Thread Description

<Use collaboration diagram or sequence diagram to depict an use-case that uses this process(es)/thread(s). Also describe overall operation of the process. Also – describe what the threads do.>

### <Process 2> Description

<Repeat for as many subsystems that you have>

## Data Decomposition

### <Class 1> Description

### <Class 2> Description

## STATES

### <State/System 1 > Description

### <State/System 2> Description

# Dependency Description

## Intermodule Dependencies

### Create Post Module

### The create story model depends on the GeoStory class. When done creating story the Create Post Model must rely on the Servers JSON interface to send the message to the server “create story” asynchronous thread is used here. This module also depends on internet access to send the Story.

### 4.1.2 My Post Module

### In order for this module to populate it list it must make a call to the sever module to get posts created by the user and the server must send back all the posts created by this user from the database module. This module also depends on internet access.

### Notifications Nearby

### This module is another list view also dependent on a call to the server module to communicate with the database module. This module also depends on the Location model for the users coordinates to populate the Stories nearby. This module also depends on internet access.

### Map

#### The map module depends on the Stories Nearby model it then uses this data to populate a Google Map. This module also depends on internet access to update your location

### Settings

### The settings menu is not dependent on other modules, but module such as notifications rely on setting to whether they will function.

### Server Controller Module

### The server controller module depends client modules to send correct JSON requests. It also depends on the database module after it parses its instructions from the app

### Database Module

### The database module is only linked to the Server controller, it sololy functions off of calls from this module.

## InterProcess Dependencies

### Location process also depends on internet access to actively update.

### Maps

### Maps depend on the Google maps api to create a map, also using this module is making an implicated call to Stories nearby.

### Settings

### Modules such as notifications and stories nearby depend on this.

## Data Dependencies

### Server data dependency

### All user info and post sent to the server are store in the database.

### 4.3.1 App data

### The app may cache data on the device such as location and maps. It also stores user info such as login info.

# Interface Description

## Module Interface

### <Module 1> Interface

### <Module 2> Interface

## Process Interface

### <Process 1> Interface

### <Process 2> Interface

# Detailed Design

NOT REQUIRED <Java Docs to be used instead>

# Design Rationale

## Design Issues

## <Issue 1>

### Description

### Factors affecting Issue

### Alternatives and their pros and cons

### Resolution of Issue

## <Issue 1>

### Description

### Factors affecting Issue

### Alternatives and their pros and cons

### Resolution of Issue

# Traceability

|  |  |  |
| --- | --- | --- |
| **No** | **Use Case/ Non-functional Description** | **Subsystem/Module/classes that handles it** |
| 1 |  |  |
| 2 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

FEEL FREE TO ADD APPENDICES AS NEEDED. UPDATE TOC BEFORE SUBMITTING