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

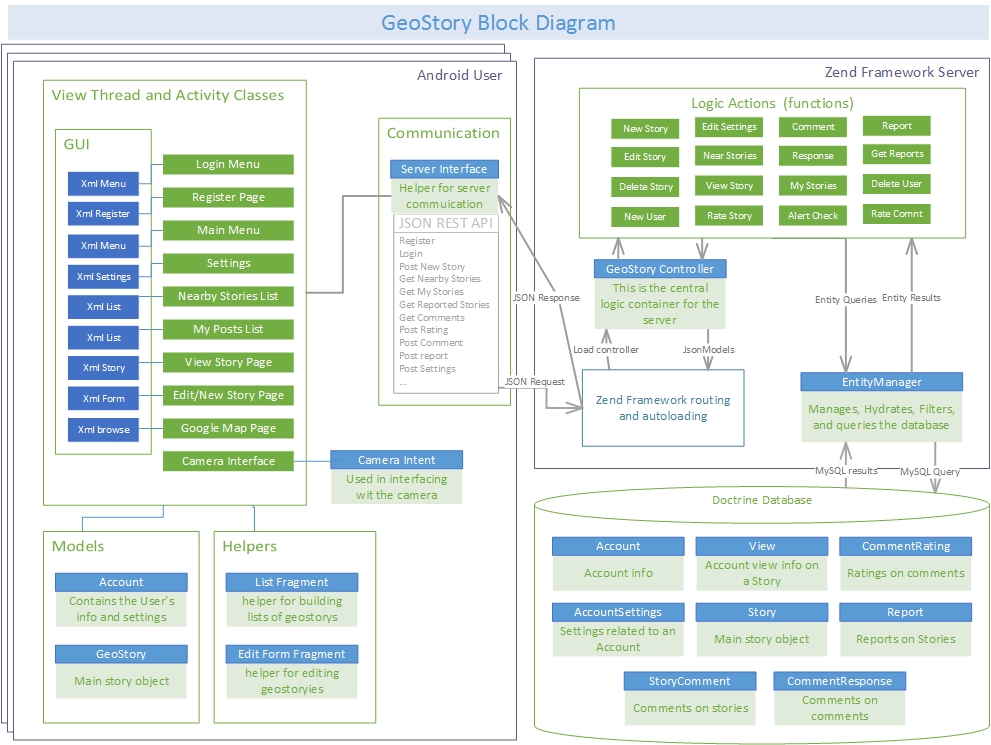
The system must be easily maintainable to keep up with new features and a growing user base.

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 1-8: Includes Login Menu, Register Page, Main Menu, Settings, Nearby Stories, MyPosts, View Story, Edit/New Story (These are described more in section 3.3> These 8 modules interact with each other through the use of intents, but also act as views. The geostory block diagram in the previous assignment shows the specific interaction between each of these modules, and are also stated in section 3.3.

### <Module 9: Server Interface Code> This module bridges the gap between the View Theads and the Server. Activities interact with it by building the async thread and registering. The server module also interacts with the server.

### <Module 10: Controller> This includes the logic and JSON parsing that is done between the app the server. This is very important to sort and interpret the data the is being sent to and from it.

### <Module 11: Database> This holds the queries that happen when a request is needed from the server. This module is the library holding all the information need (account info, Stories, ect…) that are needed to be queried (see diagram above.

## Concurrent Process

### Android View Thread Description

### Use-Case: User opens application, logs in, goes to settings page

### Uses one thread, where it is used to go through the different activities, such as described above in the use-case

### Server

### Use-Case: User opens application, logs in, creates new story.

### This Process can have two theads. When the application makes a call to the server, an asyncrounous thread is created to deal with the JSON and interactions between the application and the server.

## Data Decomposition

### <Login Menu> This class is the default opened if a user is not automatically logged in. This allows the user to log into their account, in which the server checks if there is an account with the given specications which sends them to <Stories Nearby> or go to <Register Page>.

### <Register Page> This class allows the user to register for an account, which will send the user data to the server, and promptly login the user, sending them to the <Main Screen>

### <Main Screen> This class has a list of actions the user can do (go to <Settings>, <Stories Nearby>, <My Posts>, <Edit/New Story>. There will also be text telling the user how many stories are nearby.

### <Settings> This class holds the various settings one can change, such as push notifications, ect…

### <Stories Nearby> This class gives a list of the stories that are nearby. This class can redirect to <Individual Story>.

### <My Posts> This class shows the lists of posts the user has made. A User can click on a post to go to <Individual Story>.

### <Individual Story> This class shows the current story, it’s textual representation, as well as a photo (optional). It will also show comments, and a rating. If it is the user’s post, they can redirect to <Edit/New Story> to edit such.

### <Edit/New Story> This class allows the user to edit/create a story. The user can send open the camera interface to take a photo, or choose one from a browser interface. When posted, the server fetches the story and saves it in the database.

## STATES

### <State/System 1 > Description

### <State/System 2> Description

# Dependency Description

## Intermodule Dependencies

## InterProcess Dependencies

## Data Dependencies

# 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