Volunteer Organizer with Management Entity

**Jibran Tariq**

**14K – 3408**

**MS (SPM)**

**Supervisor**

**Sir Muhammad Shahzad**

****

**Department of Computer Science**

**National University of Computer & Emerging Sciences**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Software Design Specifications**  Volunteer Organizer with Management Entity  (VOWME)  Version: [1.2]   |  |  | | --- | --- | | Project Code | VOWME | | Supervisor | Mr. Muhammad Shahzad | | Co Supervisor |  | |  |  | | Project Team | Jibran Tariq (14K- 3408) | | Submission Date | 24-May-2017 | |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name of Person** | **Date** | **Description of change** |
| 1.0 | Jibran Tariq | 04th March 2017 | Official First Draft |
| 1.1 | Jibran Tariq | 25th April 2017 | Document Revisions Fixed formatting issues and usage inconsistencies |
| 1.2 | Jibran Tariq | 12th May 2017 | Review Changes Made corrections based on |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | |
| Mr. Muhammad Shahzad | | Supervisor |
|  | | Co- Supervisor |

Document Sign-Off

|  |  |  |
| --- | --- | --- |
| **Version** | **Sign-off Authority** | **Sign-off Date** |
| 1.2 | Mr. Muhammad Shahzad | D:\DegreesPDF(Final)\Signature.png |
|  |  |  |

**Table of Contents**

[1. Introduction 13](#_Toc483412260)

[1.1. Purpose of Document 13](#_Toc483412261)

[1.2. Intended Audience 13](#_Toc483412262)

[1.3. Document Convention 13](#_Toc483412263)

[2. Overall Design Description 14](#_Toc483412264)

[2.1. Project Background 14](#_Toc483412265)

[2.2. Project Scope 14](#_Toc483412266)

[2.2.1. Features 14](#_Toc483412267)

[2.3. Not in Scope 15](#_Toc483412269)

[2.4. Project Objectives 16](#_Toc483412270)

[2.5. Assumptions 16](#_Toc483412271)

[3. System Architecture 16](#_Toc483412272)

[3.1. System Level Architecture 16](#_Toc483412273)

[3.1.1. Architectural Vision 16](#_Toc483412275)

[3.1.2. Layer Definitions 17](#_Toc483412279)

[High Level System Layer Structure. 17](#_Toc483412282)

[3.1.3. Presentation Layer 18](#_Toc483412283)

[3.1.4. Application Layer 18](#_Toc483412285)

[3.1.5. Service Layer 18](#_Toc483412287)

[3.1.6. Data Storage Layer 18](#_Toc483412290)

[4. Producer-Consumer Relationships 19](#_Toc483412294)

[5. Detailed System Design 20](#_Toc483412296)

[5.1. Database Design 21](#_Toc483412299)

[5.1.1. ER Diagram 21](#_Toc483412300)

[5.2. Application Design 21](#_Toc483412302)

1. Introduction

* 1. Purpose of Document

The purpose of this document is to describe the design of the Volunteer Organizer with Management Entity for Final Year Project of Master in Software Project Management.

* 1. Intended Audience

The intended audience for this document includes all the stakeholders in the Volunteer Organizer with Management Entity project. The document will be used by TCF Rahbar Team as the specification from which to implement the working program code. The document will be used by Mr. Muhammad Shahzad of the Supervisor as a statement of what functionality will be delivered during the project. The document will be used as a deliverable for the academic portion of the Final Year Project of Master program of Software Project Management and graded for its quality.

* 1. Document Convention

Headings: Arial 16

Sub-headings: Arial 12

Text: Arial 10

Bold Text: Arial 11

1. Overall Design Description
   1. Project Background

People like to bring a positive change to the society, and they do that by volunteering for various projects and causes. For some, the act offers a chance to give something back to the community or make a difference to the lives of people around them – for others, it provides an opportunity to develop new skills or gain experience or knowledge that they didn’t have before.

It is important for us to encourage, support and engage people into voluntary activities. The support shouldn’t just be there because it’s for good causes but it is another step towards a better society. People who have a purpose outside of their everyday working role are more motivated, healthier and happier. So, volunteering for good causes is a great act whichever way you look at it.

* 1. Project Scope

The project is based on mobile as well as web application, which will develop to work on the Android platform and Server-side language.

The mobile interface will start from the register page where volunteer must provide basic details or can allow fetch information from 3rd party application like Facebook. After registration, system track volunteer’s location and based on location, system redirects to screen where list of registered campaign will be shown with characteristics like durations, location and short description and title of campaign. Complete details with register button should display when volunteer touches on the specific campaign. Volunteer can register into campaign by providing express interest on the specific campaign. System must generate notification to organizer. System will notify registered volunteer when new volunteer will get enroll in same campaign. Volunteer can send joining invitation request to peer volunteer with 120 characters note. Volunteer can accept or reject invitation request. System should provide details of peer volunteer to another volunteer when invitation request is accepted. Volunteer can provide feedback about campaign and volunteer for future actions.

The web application must utilize for creating campaigns, approval volunteer into campaign and monitor volunteer’s activities like invitation request to another volunteer. Organizer can change details of campaigns and invites peoples for campaigns.

* + 1. Features

Following are important features of the VowMe.

* **Provide location based Campaigns.**

System will provide campaigns based on volunteer’s location within range of 100 miles.

* **Notification of new registered volunteers.**

System will notify registered volunteers about new comers in campaigns.

* **Link peer volunteer within campaigns.**

Volunteer will able to send notification to another volunteer after registration in campaigns.

* **Follow up on campaigns.**

Organizer will monitor and send notification to volunteers about the activities and ask feedback on activities

* **Feedback about volunteers and campaigns.**

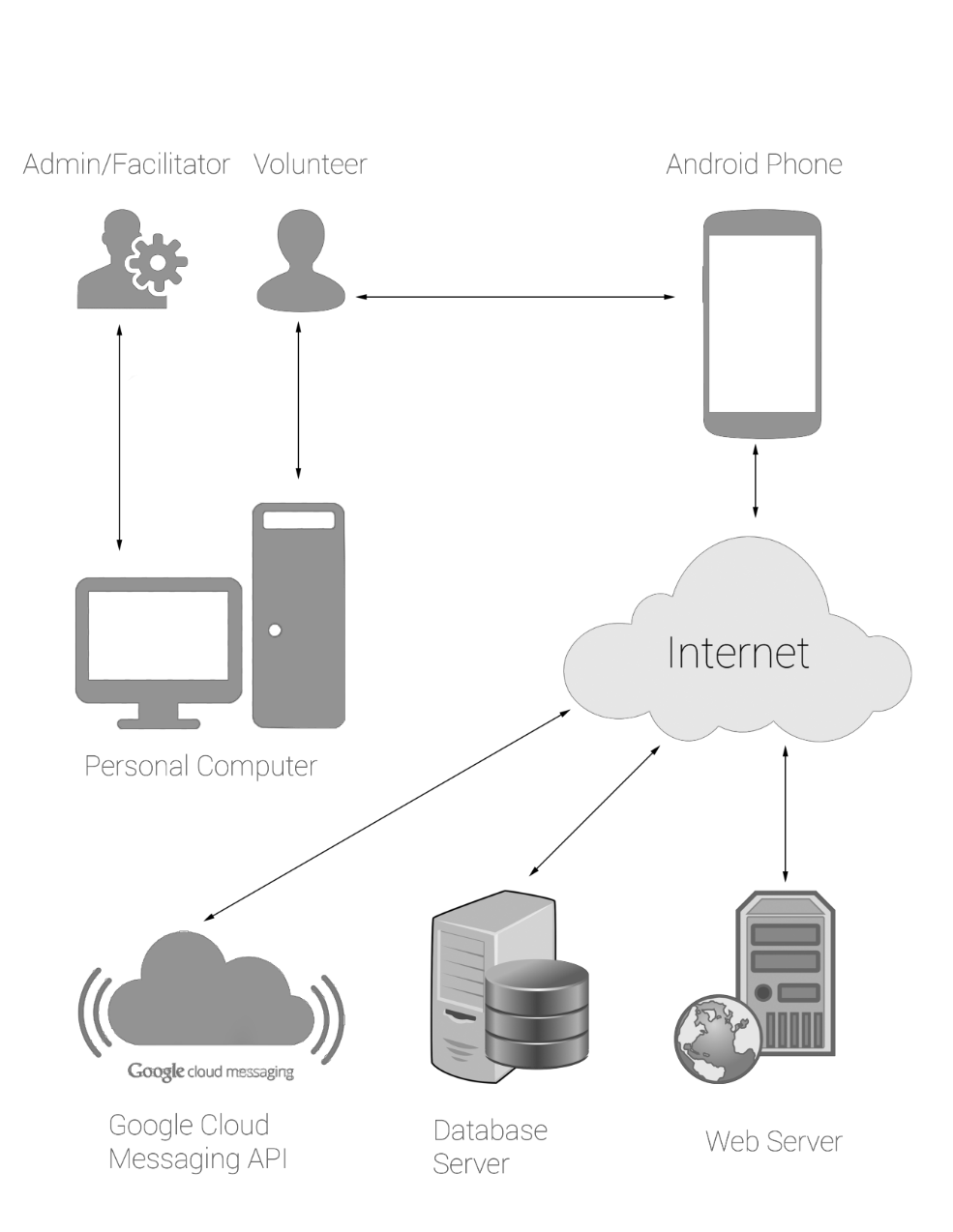
Volunteers will allow to provide feedback on activity of campaigns as well as about co-volunteers.

* **Send invitations to peoples.**

Organizer will able to send invitations to people via emails or Facebook.

* **Track Volunteer Hour**

Volunteers will able to track/manage volunteer hours to show their company.



* 1. Not in Scope

The stakeholders request some improvements which are completely out of the scope of what this team can do in the time frame allotted. These features are captured in this document as a reference to future development teams, as some of them are large enough to be a project themselves.

* Power failure backup for the dedicated server
* Verification of Volunteer Information.
* List current campaigns which are being used or being executed within city or area
  1. Project Objectives

This project will provide a way for volunteers to find registered volunteer campaign within their area, communicate more easily with other volunteers who are registered within same volunteer campaign. It will record/track volunteer hours of volunteers and will allow volunteers to provide feedback about campaigns and peers. Also, the application will provide a way for organizers to create the volunteer campaign, maintains volunteer’s basic information and send notifications related to campaign.

* 1. Assumptions

The following are assumption for the VOWME design.

* **Internet access:**

Users using the Internet site will have access to the Internet to view and use the system features. Users of the Android app will have access to the Internet to update any information in the database. For viewing purposes, it is necessary to have Internet access on the Android app.

* **Volunteer Integrity:**

System users will enter different entries, such as the number of registered campaigns they have offered. The system assumes that volunteers enter their activities accurately without the need for a procedure to verify the number of hours entered.

* **Android Device:**

VowMe users will have Android device to use the app on a phone. The Android device must be able to run Android software version 4.1.2 or higher.

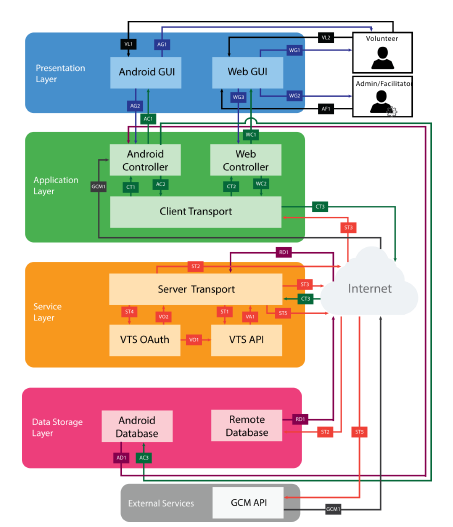
1. System Architecture
   1. System Level Architecture

This section describes the different guiding principles used to design the architecture of this system. These guiding principles serve as the basis for the rest of the document. This section also explains architectural vision.

* + 1. Architectural Vision

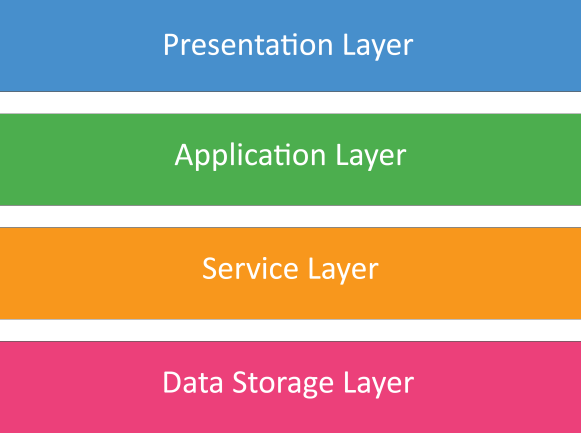
VowMe designed the architecture in such a way that all layers and components associated with the layers are simple and extensible. The architecture is designed in the way, so that in the future it leads to ease of development and modification.

The volunteer system will have four layers with several subsystems interacting with each other between and between layers. The four main layers are: presentation layer, application layer, service layer, and data storage layer. The presentation layer interacts with the user, obtains the input and displays the requested information. The application layer will gather these data and process it appropriately so that it is recovered and formatted correctly. The service layer allows Web and Android applications to access the volunteer system. The data storage layer stores the internal (Android) and external (remote) databases associated with the site and the application, respectively.



* + 1. Layer Definitions

This section describes the high-level layer structure of the system. It also defines in brief what each layer does and which subsystems are composed. The volunteer component will consist of four layers: presentation layer, application layer, service layer, and data storage layer. The figure below shows the layers whose system will be composed.



High Level System Layer Structure.

* + 1. Presentation Layer

The presentation layer includes the GUI of the web application, the user interface of the Android application. This layer is responsible for collecting information from the user and displaying the information processed to the user. The graphical interface of the Web application includes all the GUIs of the website that will allow the user to interact with the system via his computer. Similarly, the graphical interface of the Android application includes all the GUIs that will allow the user to interact with the via their Android smartphone. The layer under the presentation layer is Application Layer. The presentation layer will transmit the raw data collected from the user as input to the application layer. Depending on the task, the application layer will transmit the processed data to the presentation layer, which can then display the information to the user either via the Web GUI or in the Android GUI. Overall, the guideline used to define this layer was that it had to be a layer to handle all user interaction and display of information.

* + 1. Application Layer

The application layer is the next layer in our layer hierarchy. This layer is made up of Android Controller, web controller and customer transport subsystems. The application layer communicates with the presentation layer to get user input and events associated with user actions. Depending on the nature of the input, this layer must also request information from the database or store data in the database via the service layer. The Android controller will be responsible for managing all user-generated events via the Android GUI while the Web Controller will be responsible for managing all user-generated events through the Web GUI. The controller will then transmit the collected data to the service layer via the client Transport subsystem. Client transport acts as a bridge between the application layer and the service layer. Overall, the guideline used to define this layer was that there had to be a layer that separated the GUI Background processing and serves as a client-side communication bridge.

* + 1. Service Layer

The service layer is the next layer of our layer hierarchy. The service layer is composed of the Servers Transport, VTS API, and OAuth subsystems. This layer is responsible for all server-side processing and direct access to the database. The application layer transfers the data to the server layer through the transport subsystem of the server, which then uses the VA OAuth subsystem to verify the session before proceeding with processing. Once the session has been verified, the VTS API that hosts all input and data retrieval protocols accesses the database through Server Transport. Transporting the server acts as the primary communication line between the client and the server side to transfer the data to the VTS and VTS OAuth APIs or to return the JSON response to the application layer.

The guideline used to define this layer was that client-side processing should be separated from server-side processing. This will facilitate a highly cohesive system with low coupling. This will also make it easier to test components, as the client side and server-side code can be independently tested. Because the service layer cannot access databases, the application layer and presentation layer will be greatly simplified.

* + 1. Data Storage Layer

The next layer in our layer hierarchy is the data storage layer. This layer consists of Android and remote databases. The application layer communicates with the data storage layer via the service layer.

The Web application will communicate only with the remote database that will be implemented as a relational MySQL database. However, the Android app will communicate with the Android database that will be implemented as the relational database and the remote MySQL database. The presence of an Android database will allow the user to see his past activities.

Overall, the guideline used to define this layer was that there needed to be a layer that allowed the separation of the data from the program itself and allowed easy access to the data. Therefore, the sole responsibility of this layer is to effectively store the data and to allow secure access to the data.

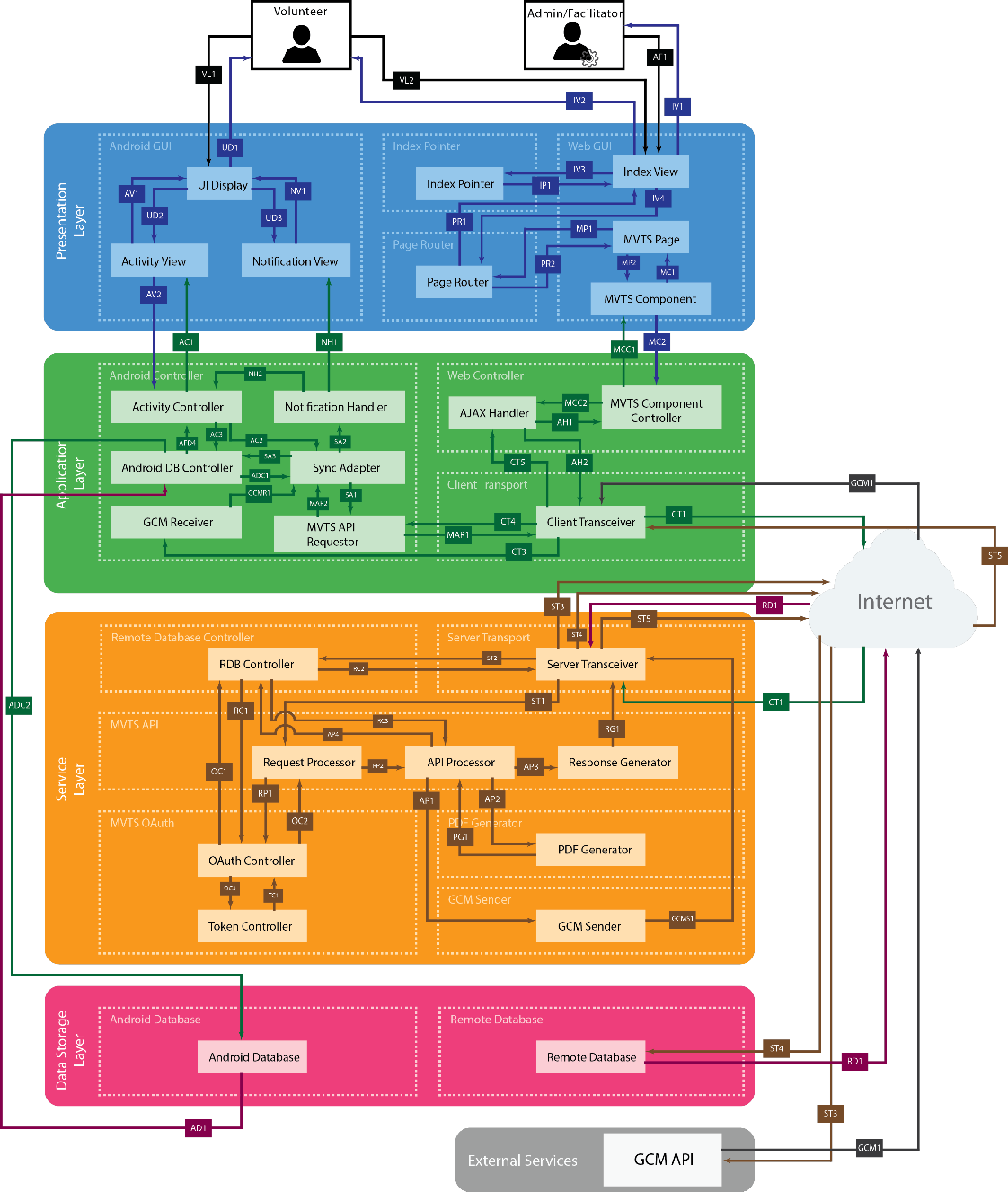
1. Producer-Consumer Relationships

The following matrix illustrates how data flows between different subsystems. From this matrix, we can see that the data flows are balanced because no subsystem consumes more than four elements and no subsystem produces more than five elements. The dispersion of the data elements can be seen, as they are not heavily grouped in one area of ​​the matrix. Based on this analysis, we can conclude that our architecture is modular where each subsystem is an integral part of the architecture and does not take on additional responsibilities.

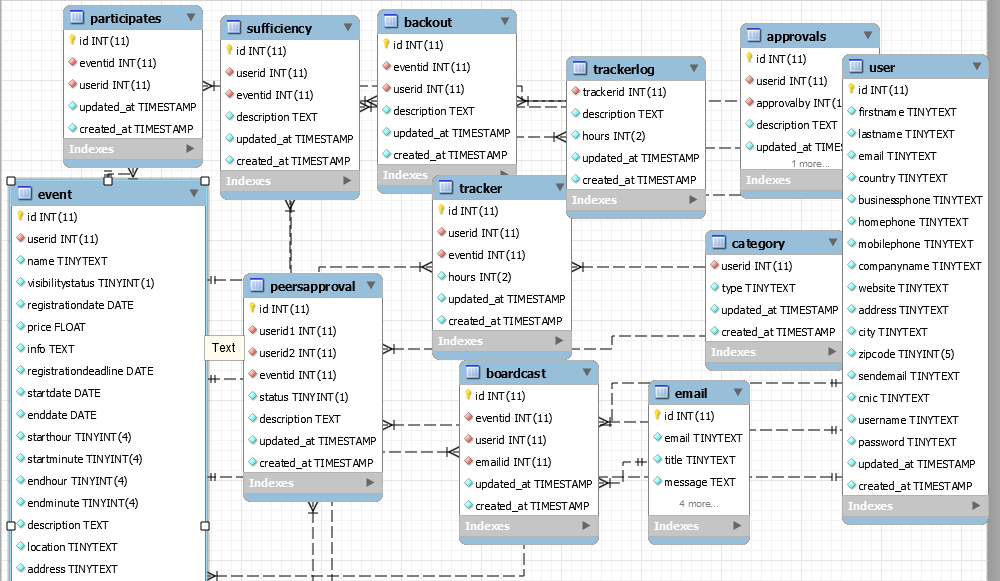
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Consumer** | Volunteer | Admin/Facilitator | Android GUI | Web GUI | Android  Controller | Web Controller | Client Transport | Server Transport | VTS OAuth | VTS API | Android Database | Remote Database | GCM API |
| **Producer** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Volunteer |  |  |  | VL1 | VL2 |  |  |  |  |  |  |  |  |  |
| Facilitator |  |  |  |  | AF1 |  |  |  |  |  |  |  |  |  |
| Android GUI |  | AG1 |  |  |  | AG2 |  |  |  |  |  |  |  |  |
| Web GUI |  | WG1 | WG2 |  |  |  | WG3 |  |  |  |  |  |  |  |
| Android Controller |  |  |  | AC1 |  |  | AC2 |  |  |  |  | AC3 |  |  |
| Web Controller |  |  |  |  | WC1 |  |  | WC2 |  |  |  |  |  |  |
| Client Transport |  |  |  |  | CT1 |  | CT2 |  | CT3 |  |  |  |  |  |
| Server Transport |  |  |  |  |  |  |  | ST3 |  | ST4 | ST1 |  | ST2 | ST5 |
| VTS OAuth |  |  |  |  |  |  |  |  |  |  | VO1 |  |  |  |
| VTS API |  |  |  |  |  |  |  |  | VA1 |  |  |  |  |  |
| Android Database |  |  |  |  |  | AD1 |  |  |  |  |  |  |  |  |
| Remote Database |  |  |  |  |  |  |  |  | RD1 |  |  |  |  |  |
| GCM API |  |  |  |  |  | GCM1 |  |  |  |  |  |  |  |  |

1. Detailed System Design

This detailed system design sections the various modules for the VowMe. These modules describe the more detailed details of the implementation of the system. This section will divide each subsystem of each layer into modules and describe the details of how these modules interact with each other or with external services. Since these modules are the smallest components of the system, it is essential that each module be tested to ensure that they are functioning properly. The most important components are built from the modules. Therefore, it is impossible to ensure that the most important components of the system function without first proving that the modules that make up them are functioning.



* 1. Database Design
     1. ER Diagram



* 1. Application Design

